

**FACULTY  
OF MATHEMATICS  
AND PHYSICS**  
Charles University

**DOCTORAL THESIS**

Jan Hajič jr.

**Optical Recognition  
of Handwritten Music Notation**

Institute of Formal and Applied Linguistics

Supervisor of the doctoral thesis: Pavel Pecina

Study programme: Computer Science

Study branch: Mathematical Linguistics

Prague 2019

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date .....

signature of the author



There is only space for one person as the thesis author, but at least here I can name some names without whom this text could not exist.

First, my thanks go to Pavel Pecina. The fact that he supported my work on this niche topic was as much as any grad student could ever ask for, and more – he took a sizeable risk as co-PI of the CEMI project by allocating quite a bit of money for my data acquisition, and supported me despite having absolutely no prior interest in the topic. Without his support, this thesis would never have been an option.

The next set of thanks goes to my amazing co-authors: Jorge Calvo-Zaragoza, Matthias Dorfer, and Alexander Pacha. A PhD might have the aura of individual achievement, but I couldn't do it without the teamwork and the inspiration that is only possible with a shared goal. The whole of our work was insanely greater than the sum of its parts.

Next, it is a pleasure to thank the MUSCIMA++ annotations team: Petra Čtveráčková, Petr Haas, Ondřej Horňas, Anna Laborová, Tereza Pinkasová, Adéla Venclová, and Michal Vokurka. Their diligence lives on!

A special shout-out goes to Xiaochuan Niu and especially Murat Akbacak, my mentors at Apple Inc., whose leadership and off-hand comments about taking ownership of my work, and working on something I want to have myself, were the fortunate inspiration for choosing this fruitful topic.

And then there are the people who kept me alive and sane throughout this time, sometimes at substantial costs to their own sanity: Ivana, Jakub, Eva, Olga, Vlastimil, and the rest of my family; to my colleagues at ÚFAL that make it such a nice workplace, to my amazing friend Veronika, whose wish is now finally fulfilled, and most dearly to Adéla, who suffered the incessant keyboard sounds during long nights of writing, who endured my long absences, the trials and tribulations of this lengthy process, and all this without her full support ever wavering.

Finally: this thesis is dedicated to my father, Jan Hajič. I could not have asked for a better role model – which extends far, far beyond this limited scope of a PhD.

It has been a blast.

O. A. M. D. G.

Title: Optical Recognition  
of Handwritten Music Notation

Author: Jan Hajič jr.

Institute: Institute of Formal and Applied Linguistics

Supervisor: Pavel Pecina, Institute of Formal and Applied Linguistics

Abstract: Optical Music Recognition (OMR) is the field of computationally reading music notation. This thesis presents, in the form of dissertation by publication, contributions to the theory, resources, and methods of OMR especially for handwritten notation. The main contributions are (1) the Music Notation Graph (MuNG) formalism for describing arbitrarily complex music notation using an oriented graph that can be unambiguously interpreted in terms of musical semantics, (2) the MUSCIMA++ dataset of musical manuscripts with MuNG as ground truth that can be used to train and evaluate OMR systems and subsystems from the image all the way to extracting the musical semantics encoded therein, and (3) a pipeline for performing OMR on musical manuscripts that relies on machine learning both for notation symbol detection and the notation assembly stage, and on properties of the inferred MuNG representation to deterministically extract the musical semantics. While the the OMR pipeline does not perform flawlessly, this is the first OMR system to perform at basic useful tasks over musical semantics extracted from handwritten music notation of arbitrary complexity.

Keywords: Optical Music Recognition Music Notation Graph Notation Assembly Object Detection Machine Learning Music Information Retrieval

# Contents

<b>I Preamble</b>	<b>3</b>
<b>1 Introduction</b>	<b>4</b>
<b>2 Reading Music</b>	<b>9</b>
2.1 Musical Semantics . . . . .	9
2.1.1 Pitch . . . . .	9
2.1.2 Duration . . . . .	11
2.1.3 Onset . . . . .	11
2.1.4 Strength and Timbre . . . . .	12
2.2 Music Notation . . . . .	13
2.2.1 Encoding Pitch . . . . .	13
2.2.2 Encoding Duration . . . . .	16
2.2.3 Encoding Onset . . . . .	17
<b>3 Optical Music Recognition</b>	<b>20</b>
3.1 Why is OMR difficult? . . . . .	23
3.2 An Overview of the State of the Art . . . . .	25
3.2.1 Methods . . . . .	26
3.2.2 Infrastructure of OMR . . . . .	34
3.2.3 Commercial Software . . . . .	36
<b>4 Contributions</b>	<b>37</b>
4.1 Music Notation Graph . . . . .	38
4.2 MUSCIMA++ . . . . .	42
4.3 The Recognition Pipeline . . . . .	44
4.3.1 Object Detection . . . . .	44
4.3.2 Notation Assembly and Semantics Inference . . . . .	51
4.3.3 Full Pipeline Results . . . . .	54
4.4 Auxilliary contributions . . . . .	56
4.4.1 Evaluation . . . . .	56
4.4.2 OMR Scientific Community . . . . .	58
<b>5 Conclusions</b>	<b>60</b>
<b>II Published Works</b>	<b>63</b>
<b>6 Theory and Resources</b>	<b>65</b>
6.1 Understanding Optical Music Recognition . . . . .	65
6.2 The MUSCIMA++ Dataset for Handwritten Optical Music Recognition . . . . .	99

6.3	Groundtruthing (not only) Music Notation with MUSCIMarker: a Practical Overview	108
6.4	Further Steps Towards a Standard Testbed for Optical Music Recognition	111
6.5	A Case for Intrinsic Evaluation of Optical Music Recognition	119
<b>7</b>	<b>Methods</b>	<b>122</b>
7.1	Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression	122
7.2	On the Potential of Fully Convolutional Neural Networks for Musical Symbol Detection	128
7.3	Towards Full-Pipeline Handwritten OMR with Musical Symbol Detection by U-Nets	131
7.4	A Baseline for General Music Object Detection with Deep Learning	140
7.5	How Current Optical Music Recognition Systems Are Becoming Useful for Digital Libraries	162
7.6	Handwritten Optical Music Recognition: A Working Prototype	168
	<b>Bibliography</b>	<b>171</b>
	<b>List of Figures</b>	<b>186</b>
	<b>List of Tables</b>	<b>187</b>
	<b>List of Abbreviations</b>	<b>188</b>
	<b>List of publications</b>	<b>189</b>
7.7	Publications used in this thesis	189
7.8	Other publications	190

**Part I**  
**Preamble**

# 1. Introduction

Optical Music Recognition (OMR) is the field of research that investigates how to computationally read music notation. Music notation is an established visual language that encodes music graphically; the role of OMR is to automatically understand this encoding and extract the encoded musical information from this graphical representation. This thesis presents contributions to Optical Music Recognition, with focus on musical manuscripts.



Figure 1.1: An example of a musical manuscript: a copy of G. B. Pergolesi's *Stabat Mater*, part X: *Fac, ut portem Christi mortem*.

Why should one attempt to do this?

In European culture, and wherever it has been able to reach, music notation is the primary way of transferring music from composer to performer, whether across a room or across centuries. The Common Western Music Notation (CWMN) writing system evolved over the course of the 17th and 18th centuries and has since been used to encode tens or hundreds of thousands of compositions, one of the major bodies of works that define European cultural heritage (like the manuscript in Fig. 1.1). It is daily in use by musicians ranging from children beginning to learn to seasoned professionals, by composers as well as performers, and reading music notation is one of the skills that belongs to a well-rounded general education. At a time when the digital domain tends to become the primary domain for manipulation and dissemination

of source materials, digitizing this body of cultural heritage becomes a requirement, lest it should fall by the wayside. Focusing on musical manuscripts is further justified by the fact that while there are probably more pages of printed music than of manuscripts, many more *compositions* are recorded only in manuscript form. Before the advent of personal computers and the proliferation of software such as Sibelius<sup>1</sup> or MuseScore<sup>2</sup> music typesetting was a very costly endeavor reserved for authors and compositions with practically assured chances of market success, or – in earlier times – with a particular printing privilege; therefore, most compositions never had the chance to be typeset.

Many digitization efforts have been undertaken by institutions holding large collections of music scores, such as the SLUB<sup>3</sup> in Dresden or the Bavarian State Library in Munich<sup>4</sup> or by organizations dedicated solely to facilitating access to scans and born-digital scores such as the IMSLP<sup>5</sup> and CPDL<sup>6</sup> projects. What these laudable digitization projects lack, however, is the capability to make digitally accessible not only an *image* of the music (which in and of itself is already extremely valuable!), but also its *musical content*: essentially, what the given music would sound like. Having digital access to the music encoded by music notation in the given documents would open up entirely new ways of interacting with the accumulated body of music scores, such as musical “full-text” search<sup>7</sup> re-typesetting old and contemporary manuscripts, creating full scores from collections where only parts for individual instruments survive – and vice versa, exporting parts for individual instruments from the full scores; cross-modal retrieval, digital musicology at scale and with access to music that has never been recorded, and cost-cutting tools for composers or music directors.

The process of *reading music* can be formulated as the process of correctly inferring the *notes* encoded graphically using the *music notation* visual language in a document that is commonly called the *score*. Notes are abstract musical objects that are determined by five properties – pitch (on a piano, which key to press), duration (how long to hold it), loudness and timbre (which are not encoded in music notation, aside from signs for some instrument-specific playing techniques), and the fifth property is onset: when should one press the given key, in relation to the start of the composition. Recovering the ⟨pitch, duration, onset⟩ triplets is sufficient to then create a practical representation for further processing in most of the envisioned applications downstream of OMR (such as searching for a piece based on a short melody); one widespread such representation is the MIDI file.<sup>8</sup> This is the first major

---

<sup>1</sup><https://www.avid.com/sibelius-ultimate>

<sup>2</sup><https://musescore.org>

<sup>3</sup><https://www.slub-dresden.de/en/collections/music/>

<sup>4</sup><https://www.bsb-muenchen.de/en/collections/music/>

<sup>5</sup><https://imslp.org>

<sup>6</sup><https://cpdl.org>

<sup>7</sup>The composition is often referred to in music and musicology as *musical text*; hence the term is indeed appropriate.

<sup>8</sup><https://www.midi.org>

part of the problem of Optical Music Recognition: extracting the *musical semantics*, defined as the set of these triplets.<sup>9</sup>

Apart from extracting the set of notes encoded by a music notation document, the second major task of OMR is recording *how* these notes were encoded: creating a digital representation of the *score* itself. This is a different objective: one may recover the musical semantics without explicitly recording information about how the semantics were encoded (e.g., one need not remember whether the stem of a half-note was oriented up, or down). Due to the nature of the music notation writing system, recovering the score itself requires a more complex representation than a set of triplets. Typical file formats for storing music notation are MusicXML,<sup>10</sup> or \*.mscz, \*.sib and other formats used by music notation editors.

These tasks can also be understood in terms of inverting the process by which music gets written down. The first stage of this process is conceptualizing a musical idea through the apparatus of musical notes; the second stage is deciding how to use the elements of music notation to best express the given structure of notes. This resulting combination of music notation elements is then embodied – using a pen in case of manuscripts, or using whatever physical process is preferred (movable type, copper plate engraving, digital typesetting, etc.). This logic of OMR as inverting the process of writing music is illustrated in Fig. 1.2.

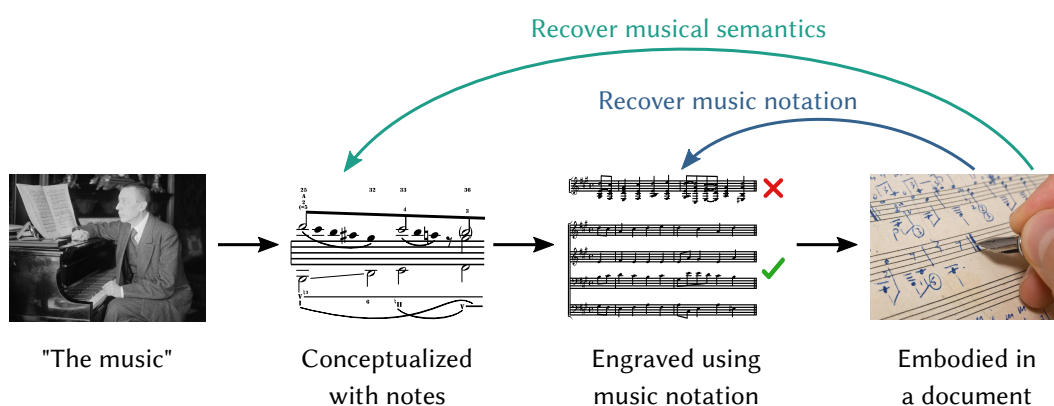


Figure 1.2: The two goals of OMR explained in terms of the process of writing music. We can recover *what* music was encoded, and we can also recover *how* it was encoded.

Both tasks of OMR pose significant challenges. Music notation does have a well-defined set of notation primitives (such as noteheads, stems, beams, ...) that form its alphabet, but some of these primitives may appear in various forms: beams can have wildly different angles, stems can be longer or shorter based on typesetting needs, etc. The sizes of primitives also differ wildly: from entire staff lines to augmentation dots. Some primitives cannot be disambiguated by their look alone, but their

<sup>9</sup>Technically, since it is possible for two notes to share all three properties, one needs to assign an ID as well, in order for the notes to formally be a set.

<sup>10</sup><https://www.musicxml.com>



relative positioning to other primitives must be taken into account: a dot below a notehead signifies a staccato articulation,<sup>11</sup> while a dot to the right of a notehead signifies prolonging the note’s base duration by half, and dots can also be part of repeat symbols or the f-clef symbol. More importantly, as opposed to the situation of the more popular Optical Character Recognition (OCR), in polyphonic music, multiple sequences can be recorded in parallel – and may share some graphical elements but not others. Inferring musical semantics requires taking into account symbols that are (unpredictably) far away from each other. The primitives can also be combined into complex composite shapes that have to be broken down into the original components in order to infer the musical semantics correctly. Processing handwritten music notation then adds a separate layer of complications, as all topological constraints that are part of music notation syntax are only approximately fulfilled. All these factors together mean that the tasks of OMR is, by virtue of properties inherent to music notation, substantially more difficult than OCR.

A further hindrance to OMR is that despite its intuitive appeal, the field is small, has had few resources and standards for reproducible OMR research, had little introductory literature for newcomers, and overall lacked internal cohesion. These challenges and issues have combined to make OMR a relatively immature field that provides few satisfactory solutions.

In this thesis, the task of automatically reading musical manuscripts is addressed. The first part of the contributions of this thesis are improvements to the “theory” of OMR: an extensive analysis of what OMR is and the taxonomy of the field, and, most importantly a general graph formalism for describing music notation that allows formulating the problem of musical manuscript recognition in a machine-learnable manner. The second significant part of the work was to prepare the resources necessary to enable actually addressing this problem, including MUSICMA++, the first extensive dataset that has ground truth appropriate for implementing and testing the full OMR pipeline. Third, an OMR pipeline that takes an image of handwritten music as input and outputs a MIDI file capturing the musical semantics encoded in the given score is built and evaluated both directly and in a retrieval setting.

The inherent variability of manuscripts also points directly towards using statistical methods that can deal with the corresponding uncertainties; we apply machine learning techniques that form the current state of the art in computer vision in general, which is specifically deep learning. Note, however, that this is not a thesis on developing deep learning methods: we are more concerned with adapting existing methods to the unique circumstance of the target domain, and, more importantly, at the same time we shape the target domain so as to allow these machine learning methods to be successfully applied; one could say that much of the value of our work lies in making the machine learning parts of the pipeline as simple as possible. With this in mind, providing an introduction and review of deep learning is not among the

---

<sup>11</sup>Technically, a staccato means shortening the note from its written duration, usually down to an audible minimum, but at the same time it has various expressive connotations.

goals of this thesis; we assume the reader is familiar with the concepts of machine learning and deep learning, and provide citations and brief explanations of specific models that we applied for the purposes of OMR. (We of course introduce the target domain of music notation, and review OMR itself.) For an overview of deep learning, we recommend the survey papers by Schmidhuber [2015] and LeCun et al. [2015], or the Deep Learning Book by Ian Goodfellow et al. [2016].

The thesis is structured as a *dissertation by publication*. In the first part, we introduce the topic of Optical Music Recognition (OMR), and describe and explain the contributions of the thesis. The substance of the thesis lies in the second part, which reproduces published works: six major peer-reviewed publications (two journal publications, out of which one is under review at the time of writing; four conference papers) and five complementary published works (that have also undergone a peer-review process, but are shorter contributions such as extended abstracts). Note that the main bibliography of the thesis only contains literature directly mentioned in the text outside of the published works; the published works carry their own bibliographies.<sup>12</sup>

---

<sup>12</sup>A full up-to-date bibliography of OMR is available at <https://github.com/OMR-Research/omr-research.github.io>; published as supplementary material to the manuscript “Understanding Optical Music Recognition” that is a part of this thesis in section 6.1

## 2. Reading Music

We have stated in the introduction (chapter 1) that “music notation is [...] a visual language that encodes music graphically; the role of OMR is to automatically extract the encoded musical information from this graphical representation”. In other words, the role of OMR is to automatically *read music*. What is this process of reading music? What is it that OMR is actually trying to achieve, in more specific terms that can be formalized for automation?

We proceed by introducing the two layers that take part in this process: the layer of musical semantics, which is the target of the process, and the layer of music notation, which is its input, and show how these relate to each other.

### 2.1 Musical Semantics

First, we define what we mean by this “encoded musical representation”.

The music that is encoded with Common Western Music Notation (CWMN) can be conceptualized as a structure of *notes in time*. This is not necessarily the only conceptualization of music, but it is the one that CWMN is designed to communicate. Notes form further structures, such as voices or phrases, but for the purposes of OMR, we restrict the conceptualization of musical semantics to a *set* of notes.

We must also clarify what “in time” means. Musical semantics are concerned not with wallclock time measured in seconds, but with *musical time*. Musical time is an abstract concept measured in units called *beats* that can be further subdivided into regular parts. Musical time is only projected into wallclock time during performance – primarily by means of *tempo*, which sets a certain baseline rate of beats per minute. However, in reality, the projection is wildly non-linear, and it is at the discretion of the performer; a linear projection would result in a robotic, boring performance.

A *note* is an abstract object that is defined by four properties: its *pitch*, *duration*, *strength*, and *timbre*. When a musician plays a note, these properties are translated into a fundamental frequency (pitch) maintained for a certain time (duration) with certain perceptual loudness (strength) and spectral characteristics (timbre). The notes are placed on the axis of *musical time* with a fifth parameter, their *onset*, which governs during performance when the note should start.

Next to notes, there are *rests*: periods of silence. They can be regarded as pseudo-notes that only have temporal parameters: onset and duration.

#### 2.1.1 Pitch

Formally, *pitch* is a categorical variable whose values are linearly ordered from lowest to highest. A piano-centric schematic of how notes are indexed by pitch is provided in Fig. 2.1. The distance between neighboring pitches is called the *semitone* and cor-

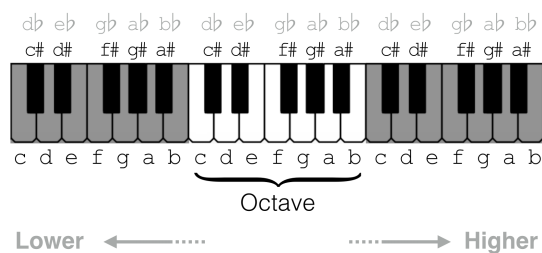


Figure 2.1: The values of pitch, illustrated on a piano keyboard. One octave period is indicated.

responds to neighboring keys. The values of pitch are expressed in terms of ⟨step, octave, accidental⟩ triplets:

- *Step*, which is traditionally one of A, B<sup>1</sup>, C, D, E, F, and G. The distance between neighboring steps is a whole tone (two semitones), except for B-C and E-F, which are only a semitone from each other.
- One period of this pattern of steps is called the *octave*, which is the second of the three determining parameters of pitch in post-medieval musical tradition. Octaves are usually numbered in the English-speaking tradition 1 to 8, from lowest to highest, with the “middle c” on a piano falling into octave 4. Although steps are traditionally named from A to G, octaves – somewhat confusingly – usually start with C and end with B, again especially where German pedagogical tradition is prevalent.
- The third pitch descriptor is the *accidental*. The accidental can shift the pitch, as determined by the step and octave, by one semitone upwards (indicated using a *sharp* sign: #), one semitone down (indicated using a *flat* sign: b), and sometimes by two semitones, indicated with a double sharp (an x-like symbol) and analogously a double flat (written usually as bb).

We emphasize again that pitch is an *abstract* object: it is not yet the fundamental frequency of the corresponding tone that would sound during performance. The relationship from pitch to a fundamental frequency during performance is governed by the *tuning* used. The middle A (A4) is first tied to a particular frequency; in modern times, this is usually 440 Hz.<sup>2</sup> The second component of projecting pitch to fundamental frequency is the *temperament*: the difference in frequency between individual semitones. In modern times, all the semitone distances between neighboring pitches are the same (this is called *equal temperament*); as indicated by Fig. 2.1, this implies

<sup>1</sup>In German tradition, which is also prevalent in the Czech Republic, B is called H.

<sup>2</sup>In practice, however, this is a much more complicated topic. Orchestras that include many wind instruments sometimes tune at 442 Hz, but in the baroque period, this standard varied: for most of music from the time of J. S. Bach or J. D. Zelenka, setting A4 at 415 Hz (roughly a semitone lower) is more appropriate, and for earlier music, many more tunings were used, both lower (390 Hz) and higher (467 Hz). Furthermore, current musicological and organological research suggests that much of Mozart’s or Beethoven’s music was originally performed with A4 = 430 Hz...

that, e.g., an F4 with a sharp (#) refers to the same frequency as a G4 with a flat (b): F4 and G4 are one tone away from each other, and the semitone upwards from F4 to F4# is the same as the one downwards from G4 to G4b. If a different temperament was used, e.g. the historical *meantone temperament*<sup>3</sup>, the distance from F4 to F4# would not be exactly half that of the distance from F4 to G4, and vice versa (the distance from G4 down to G4b would also be slightly smaller than the whole tone distance F4–G4 divided by two).<sup>4</sup>

The take-away from this explanation should be that pitch is an abstract concept expressed usually with a step, octave and accidental, that pitch of a note and frequency of a performed tone are two distinct concepts, and that music notation is concerned with encoding pitch, not frequency.

### 2.1.2 Duration

The duration of a note is theoretically any rational number, but in practice – such that is encoded using music notation – it also becomes a categorical variable. As stated above (section 2.1), duration is expressed in terms of *beats*, the basic units of musical time. In CWMN, these values are powers of 2, mostly negative: 8, 4, 2, 1, 1/2, 1/4, 1/8, 1/16, 1/32 and 1/64, rarely higher (1/128). The usual terminology calls a 4-beat note a *whole note*, a note that has a duration of 2 beats is called a *half-note*, a note that lasts for one beat is a *quarter note*, etc. (The most common grouping of beats into regular units called *measures* is 4, hence the name “whole” for the 4-beat note, as it lasts for the whole measure.) Additionally, music notation has ways to encode durations that do not fit into this sequence of powers of two, so that durations like 1/3 of a beat can be expressed as well. The usual durations one encounters (from a whole note to a 64th note) are usually depicted in music notation as shown in Fig. 2.2.

Duration applies to rests as well as to notes.

### 2.1.3 Onset

The onset of a note is the point in musical time at which the note should start being played. The elementary rule for determining the onset of a note is: once the previous note ends, the next note begins. In order to determine the onset of a note, one must therefore know the sequence into which the notes are organized. This sequence is

---

<sup>3</sup>There are very good musical reasons for this, not just chasing the nebulous concept of “authenticity”: meantone temperament provides very different timbre and character for different harmonies (although some become unusable), and these differences were actively exploited by composers of the time; when such compositions – especially those that feature the voice – are performed in equal temperament, they lose much of their dramatic power.

<sup>4</sup>On a keyboard instrument, when using meantone, one then has to choose whether to tune the given black key as the appropriate F4#, or G4b. Many so-called “well temperaments” were developed to allow some compromises in this respect; the title of J. S. Bach’s “Well-Tempered Clavier” refers to one of these many compromise options, not to the equal temperament of today. Pitch and tuning is a fascinating topic!



Figure 2.2: The types of notes according to duration. Two half-notes takes as long as one whole note, one half-note takes the same number of beats as two quarter-notes, etc.

called a *voice*. Assuming a composition with a single voice (monophonic music), the first note of the voice has an onset of 0, and the onset of the  $i$ -th note in the given voice is computed as  $\text{onset}(i - 1) + \text{duration}(i - 1)$ . This is the important concept of *precedence*. In the written score, notes are encoded by precedence from left to right according to the positions of the corresponding graphical notes. When more than one voice is present, in music that is called *polyphonic*, a simple left-to-right ordering is not sufficient; for correctly inferring precedence, one must correctly assign notes to voices.

The concept orthogonal to precedence is *simultaneity*: a simultaneity is a set of notes that shares the same onset. Simultaneity can happen within a voice, where all the participating notes share the same preceding note (or members of the preceding simultaneity); if all simultaneities with more than one member are such that the notes belong to the same voice, we call the music *homophonic*.

#### 2.1.4 Strength and Timbre

The note parameters of strength and timbre can be mostly left out for the purposes of OMR, as they are barely encoded in music notation. Strength is rudimentarily conveyed with dynamic markings such as *piano* or *forte*, and some symbols that convey how strength should change over a sequence of notes (*crescendo* for increasing, *decrescendo* for gradually decreasing strength); timbre is expressed with sporadic textual expressive markings or instrument-specific techniques. These two parameters are left for the most part at the discretion of the performer.

Importantly, for the motivating applications of OMR listed above, strength and timbre need not be a part of the OMR outputs. **Therefore, for the purposes of this thesis, we simplify the definition of a note to just the triplet**  $\langle \text{pitch}, \text{duration}, \text{onset} \rangle$ , and the task of recovering musical semantics then becomes the task of recovering the set of such triplets. This simplified representation is already rich enough to have the OMR system output files in the widely used MIDI format, which in turn serves as an input of many of the downstream applications of OMR – especially full-text search in music archives and other applications oriented towards digital libraries, more broadly curating and making accessible sheet music collections, and musicology overall.

## 2.2 Music Notation

Now that we have introduced the layer of *musical semantics*, we turn our attention to the music notation, specifically Common Western Music Notation (CWMN) visual language and its relationship to these semantics. We introduce music notation terminology factored into subsets of the music notation alphabet according to which aspect of the musical semantics defined above the given symbols help encode.

The elementary interface between the “written” layer and the “semantic” layer, or music notation and the notes that it encodes, are *noteheads*. The general rule is that **one notehead encodes one note**<sup>5</sup> Noteheads are round<sup>6</sup> objects with a fixed size.<sup>7</sup> They are the central and most frequent music notation primitives.

In the context of OMR, one must be careful to distinguish the abstract musical *note* object with the composite graphical objects that are often also called notes. These graphical objects serve as a useful pedagogical concept, but for the purposes of OMR they are not well-defined. Whenever we wish to refer to the graphical objects, we will explicitly use the term *graphical note*.

A reference example of actual music notation is given in Fig. 2.3.

Suppose we have correctly identified noteheads, and therefore we know how many notes are encoded in a given notation document. It remains to find their parameters that are unambiguously recorded by music notation: pitches, durations, and onsets. The nature of the music notation visual language is *featural*: these properties are encoded not using individual primitives, but using the *configurations* of these primitives. We describe which primitives participate in encoding which property, and explain how.

### 2.2.1 Encoding Pitch

Pitch, described as a ⟨step, octave, accidental⟩ triplet, is encoded using the following primitives:

- stafflines and spaces between them, which combine into staves,
- ledger lines,
- clefs (g-clef, f-clef, c-clef),
- accidentals (sharp, flat, double sharp, double flat, natural),
- measure separators (thin and thick barlines and barline groups).

---

<sup>5</sup>In polyphonic music, this rule may get broken: multiple voices on a single staff may share pitch and have a graphically compatible duration, in which case a notehead can encode two notes; this would be evident to the reader from the presence of two stems attached to a single notehead.

<sup>6</sup>Mostly. As is the case with nearly everything in music notation, exceptions abound. In this case, CWMN allows for different notehead shapes, especially in the latter half of the 20th century: these are most widely used to indicate certain playing techniques that affect timbre. Percussion parts are most prone to non-round noteheads. However, as these are still (relatively) rare, we do not consider non-round noteheads in this thesis.

<sup>7</sup>Except for *grace* noteheads: these encode ornamental notes that do not have a duration of their own.



Figure 2.3: An example of real-world notation, with the individual notation elements marked (not exhaustive). Taken from: Ernst Bloch, Baal Schem: Drei chassidische Stimmungen for violin and piano, II: Improvisation (Nigun).

Figure 2.4: The elements encoding pitch: three notes with the same pitch written in different ways. Noteheads red, for clarity.

Three notes with the same pitch are shown in Fig. 2.4.



Since the times of Guido of Arezzo (12th century), the *staff*, consisting of a number of equidistant parallel stafflines and the spaces between them, is the basic layout object in music notation. **Music is read left-to-right** per *system*; each system consists of a number of *staves* that are read concurrently; each staff consists of a certain number of *stafflines* and the *staffspaces* between those lines (plus two surrounding the outer stafflines of the staff). Overwhelmingly often, in post-1650 scores, staves are built from groups of five stafflines, with six staffspaces between and around them.<sup>8</sup> **Noteheads are placed on stafflines or into staffspaces.** Each notehead is positioned on exactly one staffline or staffspace (not both). Moving a notehead to the neighboring staffspace (or, vice versa, staffline) changes its pitch by a step (not necessarily by a semitone; see subsection 2.1.1).

*Ledger lines* are used when the pitch of a note is more extreme than can be recorded with the given limited number of stafflines and staffspaces. Ledger lines are short horizontal lines parallel to the staff that simulate the presence of a given number of additional stafflines for the given note; they are interpreted exactly as additional stafflines would be.

The *clef* symbols are also attached to stafflines (not staffspaces, however). **Clefs govern how positions on the staff are interpreted in terms of step and octave.** The *f-clef* denotes the staffline on which noteheads will be interpreted to encode notes with the pitch F3; overwhelmingly often, this would be the second staffline from the top. This clef is also known as the bass clef, as it is most often used to encode lower-sounding music, such as the left hand's part in piano literature or the bass part in vocal music. The *g-clef*, also known the violin clef, denotes the staffline corresponding to the pitch G4; this clef is overwhelmingly often placed on the fourth line from the top, and is used for music in the higher ranges (right hand on the piano, soprano and later also alto parts, the eponymous violin music, etc.). The *c-clef*, sometimes called the viola clef, denotes the staffline corresponding to the middle C (C4), and as its name suggests it is used mostly with alto or tenor instruments such as the viola. The c-clef usually appears in viola parts on the middle of the five standard stafflines, but in trombone, French horn, or cello music, it often appears also on the second line from the top, and in pre-1750 music it can appear on any of the stafflines.<sup>9</sup>

Clefs are most often present at the beginnings of staves, but it is perfectly valid to encounter clefs anywhere on a staff; all clefs are valid for interpreting notehead positions on the given staff only to the right of the given clef.

Thus, from the position of the clef, and the position of the notehead on the staff (in terms of which staffline, staffspace, or ledger line the notehead is placed on), one can assign to each notehead the step and octave parts of its pitch. What remains is the accidental. This is encoded by *accidental* primitives (again, one must distin-

---

<sup>8</sup>The widest exception today is modern plainchant notation with four stafflines, and single-line percussion parts.

<sup>9</sup>Surprisingly, in early music, the g-clef and the f-clef only appear more frequently in non-standard positions in French baroque scores.

guish between the semantic property of accidental and the subset of music notation primitives – sharps, flats, etc. – that encode this semantic property). These primitives are the *sharp* (modifies pitch upwards by semitone), the *flat* (downwards), the *double sharp* and *double flat* that modify pitch twice as much, and the *natural*, which cancels any accidental that might have been valid for the given note. Similarly to clefs, accidentals are also valid to the right of their horizontal position on the staff, but they have two flavors – *inline* accidentals, and *key signature* accidentals.

*Inline accidentals* apply to notes encoded by noteheads on the same staff position as the accidental by convention up until the end of the measure, denoted by the next barline or barline group.<sup>10</sup> Using this convention, if one wants to cancel an inline accidental at a given note (e.g., first encode F4#, then F4), one would use the *natural* sign to cancel, from then onward, the effect of the accidental that would apply at that point.

Accidentals in *key signatures* (which are simply groups of accidentals that are not associated with any specific notehead) differ from their inline counterparts in that their validity does *not* expire with the next barline: key signature accidentals stay valid unless overridden temporarily by an inline accidental, or permanently by a new key signature. They are usually re-stated at the beginning of each staff.

Note also that the inline accidentals and key signature changes imply that reading music is a *stateful* process: in order to correctly read the next note, one must remember some information about the previous notes and music notation elements that have already been read.

This analysis should now clarify why the notes corresponding to the three highlighted noteheads in Fig. 2.4 actually encode the same pitch.

## 2.2.2 Encoding Duration

Duration is encoded using:

- notehead type (full vs. empty),
- stem,
- flags and beams,
- augmentation dots (absent, one, two),
- tuples.

Recall that duration is a categorical variable that determines for how many beats (units of musical time) a note should be held. The values of duration are drawn mostly from powers of 2: 4 beats, 2 beats, 1, 1/2, 1/4, 1/8 and 1/16 (rarely is a note written shorter than 1/16th of a beat). This is all we need to care about in OMR; music notation does *not* encode the relationship of beats to wallclock time. The names for

---

<sup>10</sup>Before roughly 1670, the convention was to apply the inline accidental only to the next note on the given staff position, and this has become relevant again for atonal music in the 20th century.

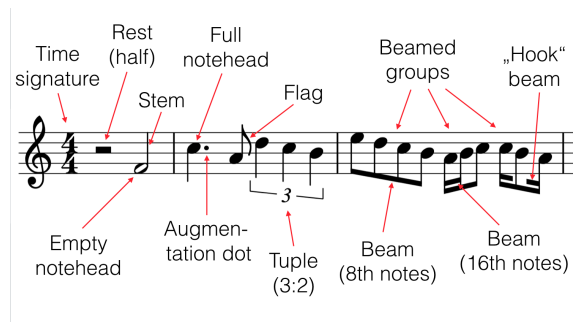


Figure 2.5: The elements encoding duration.

durations are somewhat misleading: a 4-beat note is a *whole* note, a 2-beat note is commonly called a *half* note, a 1-beat note is a *quarter* note, etc.

See Fig. 2.2 for how these duration values are prototypically encoded in CWMN. An empty notehead without a stem denotes a whole note; with a stem, it encodes a half note. A full notehead with a stem encodes a quarter note (1 beat); in CWMN, full noteheads are required to have stems. Smaller values are then encoded with *flags* or *beams*. There can be more than one flag (or beam) associated with a notehead; each associated flag or beam halves the duration of the encoded note (a note encoded with a full notehead, stem and no flag has a duration of 1 beat, with one flag it will have a duration of 1/2 beat, with two flags, 1/4 of a beat, etc.). Beams are used instead of flags prototypically when there is more than one consecutive note with sub-beat duration; they connect the respective graphical notes into *beamed groups*. For duration values from outside this row – especially when dividing a larger value into three equal parts instead of two – the *tuple* symbols are used. The elements that govern duration are depicted in Fig. 2.5

Beams and beamed groups are some of the most problematic elements of music notation for OMR. Individual beams can have wildly different lengths, thicknesses and angles; the way they are combined into groups relies on just a few rules, but can lead to visually complex structures with many overlapping elements, and especially in manuscripts, beamed groups are prime suspects where the topological constraints on printed music notation get violated. A moderately complex beamed group is depicted in Fig. 2.6; a tricky real-life example from an early music manuscript is depicted in Fig. 2.7

### 2.2.3 Encoding Onset

Onset is encoded by how notes are ordered by precedence and assigned to voices. Ordering notes by precedence within a voice is done simply by ordering the corresponding noteheads left-to-right within each staff (and ordering systems downwards). A simultaneity with more than one member within a voice is encoded by making all noteheads that correspond to its member notes share a single stem. (In case of whole notes, where no stem is used, the empty noteheads are either stacked



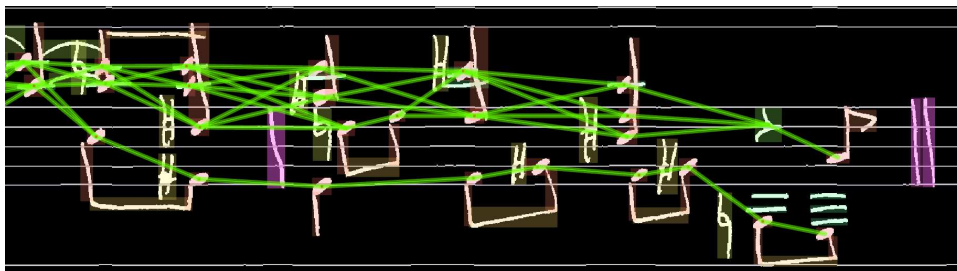
Figure 2.6: A somewhat complex beamed group. Note how it changes stem sizes, which are otherwise mostly fixed (at  $3.5 * \text{staffspace\_height}$ ). The first two and the last note have a 16th duration (two relevant beams), the third and fourth notes are 32nd notes (three beams).



Figure 2.7: A more complex beamed group situation in a 17th century violin manuscript (H. I. F. Biber, *Mystery sonata IV*).

on top of each other, or, if they lie on a neighboring staffline and staffspace, they are stacked very close to each other without the noteheads overlapping.)

In case of polyphonic music on a single staff, correctly finding the predecessor(s) of a note is a difficult problem to solve in general. When the number of voices in a staff is limited to two, most often the voices can be differentiated by stem direction (rests are then usually positioned significantly above their usual position for the top voice, and below their usual position for the bottom voice); however, when more than two voices are present on a staff, this approach fails. One can partly rely on the heuristic that whenever voices cross (a voice that is typically below the other has a note above the one that is currently in the upper voice), stem directions are set to reflect this. However, resolving precedence in polyphonic music in general is an open problem; examples of complex situations are given in Fig. 2.8. (Note also that in Fig. 2.7, there is a rare situation where the noteheads that are part of one beamed group are not necessarily encoding each other's predecessor notes.)



(a) Precedence in polyphonic guitar music: note the variable number of voices, some with chords.



(b) Precedence in complex pianoform music (green connecting lines, left to right). Notice the voice that is written across staves.

Figure 2.8: Examples of notation where precedence is complicated.

### 3. Optical Music Recognition

Armed with an understanding and a formal specification of musical semantics and how they are expressed using the music notation visual language, we can now talk about the field of Optical Music Recognition in more detail. We revisit the dichotomy of OMR goals and discuss its implications for OMR systems, describe the taxonomy of OMR systems according to the input they are designed to process, and proceed with an overview of the state of the art of the field.<sup>1</sup>

We have already indicated in chapter 1 that there is a fundamental difference in the objectives of the field: OMR for *replayability*, and *reprintability*, as remarked briefly by Miyao and Haralick [2000]. (Unfortunately, this distinction was not followed up on in subsequent OMR literature.) The terms themselves suggest where the dichotomy lies. Replayability aims to recover *what* is encoded, and it has its own set of applications such as – besides the eponymous replay – full-text search, musicological research, and curating of digital archives (for instance, detecting copies of the same music across multiple collections, which helps track how music evolved over time and geographical area).<sup>2</sup> Reprintability means the ability to re-typeset the given score digitally: recover *how* exactly music notation was used to encode the given set of notes.<sup>3</sup>

Of course, these two objectives are closely related, since they fundamentally require dealing with the same input. However, there are important differences. One lies in the output representation that is required. For replayability, this is typically some equivalent of MIDI: regardless of the file format, the underlying representation can be simply a list of the  $\langle \text{pitch}, \text{duration}, \text{onset} \rangle$  triplets (see section 2.1), for monophonic music perhaps better stored as a sequence of consecutive  $\langle \text{simultaneity-or-silence}, \text{duration} \rangle$  elements. This is not the case for reprintability, where one has to output a formal representation of the score itself, which is more complicated than merely the set of symbols and their positions – given the featural nature of music notation, one must also bear in mind how the symbols depend on each other. (For

---

<sup>1</sup>Much of this chapter is analogous to the content of the manuscript in section 6.1.

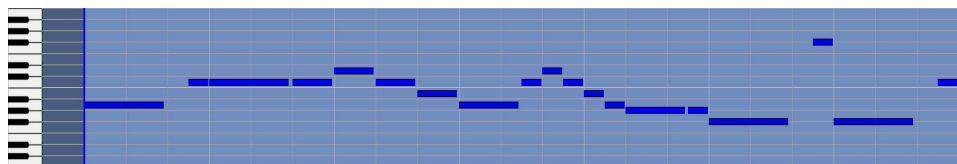
<sup>2</sup>Another such interesting application is part matching: many manuscript scores exist only as parts for individual instruments and not as orchestral scores, and sometimes these are assigned incorrectly: having access to the underlying musical semantics enables checking for incompatibility of parts assigned to one score, or unexpected compatibility of a part that was perhaps not considered in the context of the given composition – for instance, a part found in a different location.

<sup>3</sup>In the context of the previous footnote: while part matching requires replayability to make sure that the given parts are compatible, when one moves towards actually building scores from individual parts, this is a reprintability-oriented application: one needs to know at the very least the locations of measures, so that they can be correctly aligned to each other. In reality, this task is more difficult still, because in individual parts, consecutive measures where the instrument is not playing are usually condensed using either special markings (e.g., a pause of 13 measures denoted as two five-measure pauses and three single-measure pauses), or a different mark and the corresponding number. Again, this example illustrates how in conceptually straightforward applications of OMR run into the myriads of details and small optimizations of CWMN for readability that have evolved over the three hundred years of usage.





(a) Input: manuscript image.



(b) Replayable output: pitches, durations, onsets. Time is the horizontal axis, pitch is the vertical axis. This visualization is called a *piano roll*.



(c) Reprintable output: re-typesetting.



(d) Reprintable output: same music expressed differently

Figure 3.1: OMR for replayability and reprintability. The input (a) encodes the sequence of pitches, durations, and onsets (b), which can be expressed in different ways (c, d).

instance, in practical terms, a reprintability-oriented retypesetting application might require splitting a system in two – which requires adding the correct clef and key signature at the beginning of the new system, which is in turn derived from the position where the system is split.) Additionally, while a score has only one interpretation in terms of musical semantics, the same musical semantics can be encoded with many different scores (some of which are more readable than others); see Fig. 3.1. This already implies that more information about the document must be explicitly recorded if one wants a description of the score, rather than the semantics.

Orthogonally to their goals, OMR systems can be characterized by the types of input they are designed to process. The first major difference lies in the **input signal**: we differentiate *offline* OMR, which processes an image, from *online* OMR, which processes the temporal signal from a touch-based device (such as writing with a stylus on a tablet). The latter is in principle simpler because the pen strokes represent a very good natural segmentation heuristic; however, the former is more broadly applicable: while online OMR has its place whenever a composer or arranger is willing to use a device that records the trajectory information, it cannot deal with the stacks of sheet music that have already been written. An interesting

combination, however, is to use online OMR in ground truth acquisition, as tracing the already written notation is much faster and more natural to qualified annotators (who presumably themselves have ample experience with *writing* music notation), as done by Calvo Zaragoza et al. [2016a]. The second distinction based on input signal is whether the music in question is typeset, or handwritten, with obvious implications for symbol intra-class symbol variability. Third, one must specify what type of music notation a system is designed to process: CWMN, mensural notation, choral square notation, tablature (lute, modern guitar, North German organ...), etc.

A second major axis of classifying OMR systems by input is according to the **complexity of notation** they are able to process. This was described in depth by Byrd and Simonsen [2015]; we use a slightly different classification that nevertheless preserves the spirit of the original categories:

- *Monophonic*: each staff contains at most one voice; each simultaneity contains at most one note.
- *Homophonic*: each staff contains at most one voice; each simultaneity can contain more than one note.
- *Polyphonic*: each staff can contain multiple voices, but the staves can still be processed in isolation.
- *Pianoform*: staves contain multiple voices, and there is interaction between staves (e.g., cross-staff beaming).

The categories of [Byrd and Simonsen, 2015] did not differentiate between homophonic and polyphonic music, but rather between music on one staff and on multiple staves (monophonic – monophonic multi-staff – polyphonic multi-staff – pianoform). However, we believe the distinction between homophony and polyphony to be more important. First, separating music into staves is a relatively easy part of the problem (for instance, the OMR pipeline presented in this thesis handles assignment of symbols to staves successfully in variable manuscripts merely using heuristics). Second, the distinction between homophony and polyphony is important from the perspective of precedence: in homophonic scores, inferring precedence is near-trivial, while

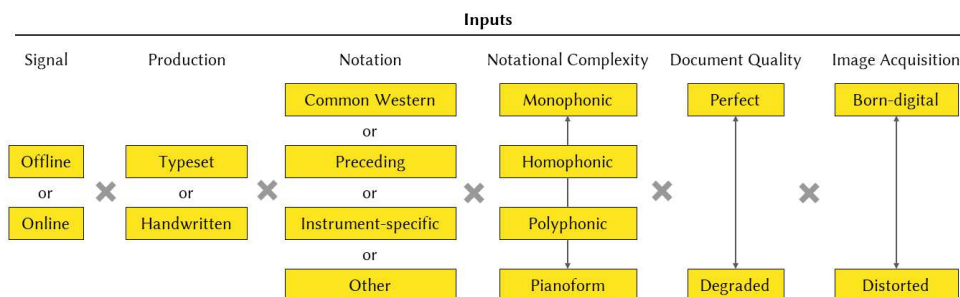


Figure 3.2: The basic ways of characterizing OMR inputs.



in polyphonic scores it becomes a difficult problem. Third, no end-to-end OMR systems are capable of modeling polyphonic output so far.

A third way of characterizing the inputs of OMR is by the image quality: both in terms of the underlying document, and in terms of the imaging process used to digitize the document. Problems that affect the underlying document are degradation over time (especially for archival materials) – most serious of which is bleedthrough – or outright damage to the material (stains and tears). The imaging process then ranges from high-quality scans from music libraries to mobile phone photos in sub-optimal lighting conditions.

An overview of the basic characterizations of OMR inputs is given in Fig. 3.2.

### 3.1 Why is OMR difficult?

OMR is still an open problem and satisfactory solutions are available only for limited sub-problems [Bainbridge and Bell, 2001, Rebelo et al., 2012, Novotný and Pokorný, 2015]. Besides the small size of the field and the accompanying non-technical challenges [Calvo Zaragoza et al., 2018], one reason why OMR is not solved to any satisfactory extent is its sheer difficulty [Byrd and Simonsen, 2015].<sup>4</sup>

There is a straightforward intuitive description of OMR as “Optical Character Recognition for music”. However, while appealing, this analogy is only accurate in terms of the purpose of both OMR and OCR. Given the content that it is trying to encode (see chapter 2), music notation has evolved into a very different writing system than the writing systems for natural languages: it is a *featural* system, where one must recover *configurations* of symbols in order to be able to output the well-defined musical semantics that OMR is, by definition of its domain, *expected* to produce. Compared to OCR, which has to output the sequence of graphical symbols (including whitespace) and this already can be presented as input for downstream applications in Natural Language Processing, OMR must, by virtue of the domain it operates on, perform additional steps in order to be considered useful. This is one fundamental reason why the analogy of OMR to OCR does not hold beyond a superficial similarity of purpose.

A further source of difficulty are the visual properties of music notation. According to [Byrd and Simonsen, 2015], music notation is probably the most complex writing system. The main reasons why the way CWMN is written makes OMR more difficult than OCR are:

- In order to correctly disambiguate individual symbols, and more generally in order to construct and interpret the symbol configurations correctly, both the horizontal and vertical dimensions are salient, in terms of both size and position.

---

<sup>4</sup>That OMR is a difficult problem is attested to by the fact that problems connected to the inherent properties of music notation have been called “really rotten” in a publication title, already in 1989 [Clarke et al., 1989]!

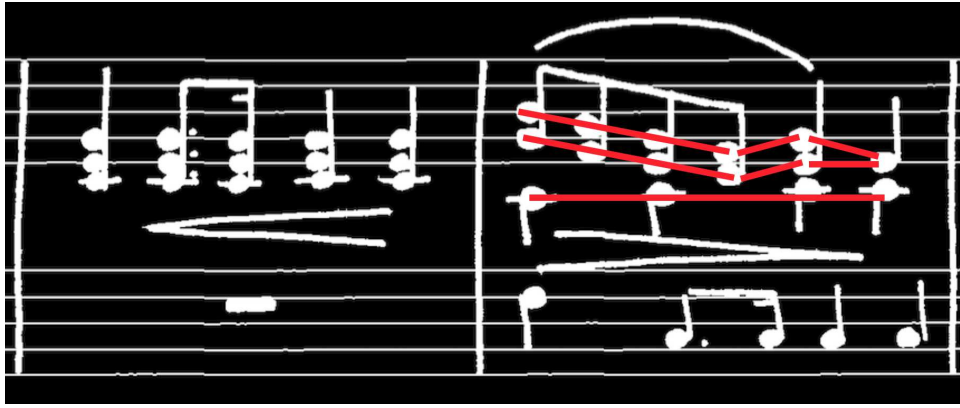
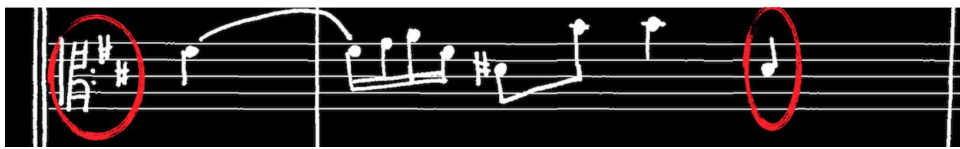
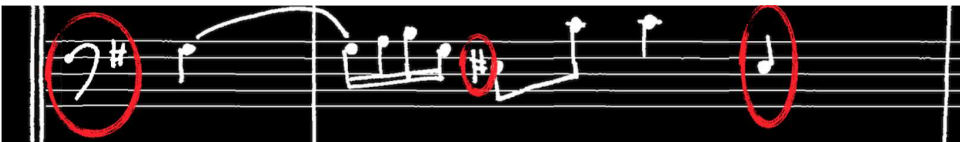


Figure 3.3: The presence of multiple voices (indicated with red lines) adds complications.



(a) The C-clef on the left influences how stafflines are interpreted with respect to the pitches they denote.

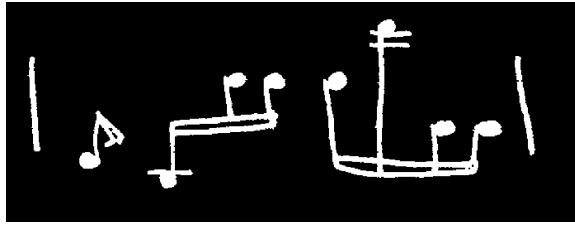


(b) A change of clef and key signature. Also, notice the sharp in the middle: it is valid up to the end of the measure.

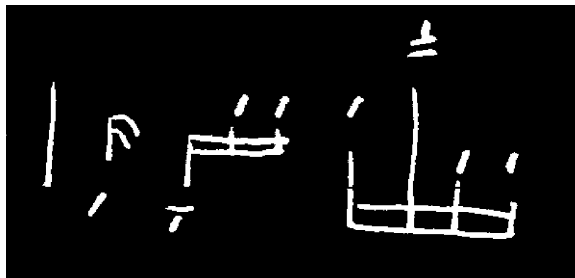
Figure 3.4: Long-distance relationships affecting pitch of the note on the right.

- Graphical complexity is increased due to the fact that many symbols overlap (especially stafflines) [Bainbridge and Carter, 1997], and by design composite graphical structures are built (esp. beamed groups – see Fig. 2.6).
- In handwritten music, besides vastly more varied symbol shapes, the variability of handwriting leads to a lack of reliable topological properties overall (Fig. 3.5) – symbols that should not touch start touching, and conversely gaps are left where symbols should touch or overlap.
- In polyphonic music, individual voices are written, in a sense, “over” each other (some symbols may be shared among multiple voices) – as opposed to OCR, where the ordering of the symbols is linear (Fig. 3.3).
- Recovering pitch and duration requires recovering long-distance relationships (Fig. 3.4).

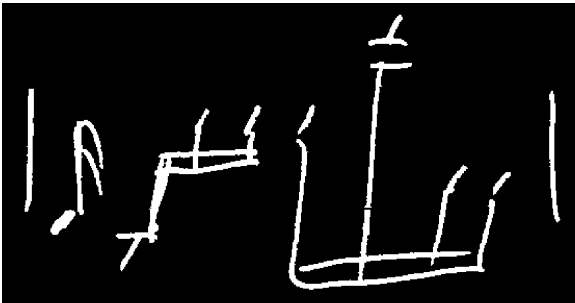
Finally, OMR has proven a long-term challenge also because of non-technical reasons: the field had until recently no natural shared publication venues, research was done with comparatively little regard to reusability, and infrastructure was lacking (both in terms of datasets, evaluation methodologies, and even appropriate file



(a) Nice handwriting that follows topological constraints according to ideal printed CWMN.



(b) Disjoint notation primitives.



(c) Very hasty handwriting. Some noteheads may be very hard to distinguish from the stem.

Figure 3.5: The variety of handwriting. Taken from the CVC-MUSCIMA dataset [Fornés et al., 2012].

formats), which made it difficult to *work* in OMR [Calvo Zaragoza et al., 2018].

### 3.2 An Overview of the State of the Art

We now introduce the state of the art on which the work in this thesis builds. The purpose of this section is to give the reader a good idea of the starting point of the thesis, in order to understand its contributions; detailed reviews of works related to the individual thesis contributions are parts of the corresponding published works.

What is the state of the art in Optical Music Recognition? While the “wishlist” of OMR applications exists from the earliest publications [Pruslin, 1966, Prerau, 1971, Fujinaga, 1988, Blostein and Baird, 1992] onwards, after more than 50 years of OMR research, few truly convincing results have materialized. The reasons for this state of affairs are several. First, despite its intuitive appeal, the field is small (some 500

publications to date), as it requires a combination of computer science expertise and relatively deep domain knowledge of music and music notation. Second, it follows that the field does not have too many resources and established methodologies. Most work on OMR has been focused into PhD theses [Fujinaga, 1996, Bainbridge, 1997, Fornés, 2009, Rebelo, 2012, Calvo Zaragoza, 2016], which is a form that offers little incentive for collaboration and establishing a research community that in turn establishes standards for evaluation and interoperability; therefore, it becomes difficult to build on previous work. Given the lack of standardized, practical evaluation methodologies [Byrd and Simonsen, 2015, Hajič jr. et al., 2016] and even the underlying understanding of *what* should be evaluated (see section 6.1), the field cannot in good conscience even say what “the state of the art in OMR” is.

Having said that, there are survey papers available for OMR. The first such substantial paper is by Blostein and Baird [1992], which is the first attempt to systematize the field. The key survey paper for OMR up until 2012 is [Rebelo et al., 2012], which systematizes the many approaches and contributions to OMR. The underlying terminology of the field and an analysis of its structure and needs has been done by [Byrd and Simonsen [2015]]; a smaller but nevertheless useful review paper for developments up to the start of this thesis has been written by [Novotný and Pokorný [2015]]. As a part of its contributions, the tutorial paper in section 6.1 systematizes the field from the perspective of its *output*, in addition to [Byrd and Simonsen [2015] characterizations by input and [Rebelo et al., 2012] by method. Further resources exist: a list of OMR datasets<sup>5</sup>, an OMR bibliography<sup>6</sup> and a video series that introduces OMR<sup>7,8</sup>. What the survey papers have in common is the assessment that a complete OMR system still lies in the future<sup>9</sup>.

With these limitations in mind, we turn to introduce the state of OMR, in terms of its methods and the available infrastructure.

### 3.2.1 Methods

In terms of methods, the problem is usually broken down into the following steps [Bainbridge and Bell, 2001, Fornés et al., 2006, Rebelo et al., 2012, Hankinson, 2014, Novotný and Pokorný, 2015]:

1. **Preprocessing.** This step involves image de-skewing, potentially binarization, and other steps that ensure the image is as normalized as possible for further processing.

---

<sup>5</sup><https://apacha.github.io/OMR-Datasets/>

<sup>6</sup>Originally maintained by [Fujinaga [2000]], recently updated and verified as part of the submitted manuscript in section 6.1: <https://github.com/OMR-Research/omr-research.github.io>

<sup>7</sup>Presented at the ISMIR 2018 conference as a tutorial: <https://www.youtube.com/playlist?list=PL1jvwDVNwQke-04UxzIzY4FM33bo1CGS0>

<sup>8</sup>The latter two resources were to a significant extent co-created by the thesis author.

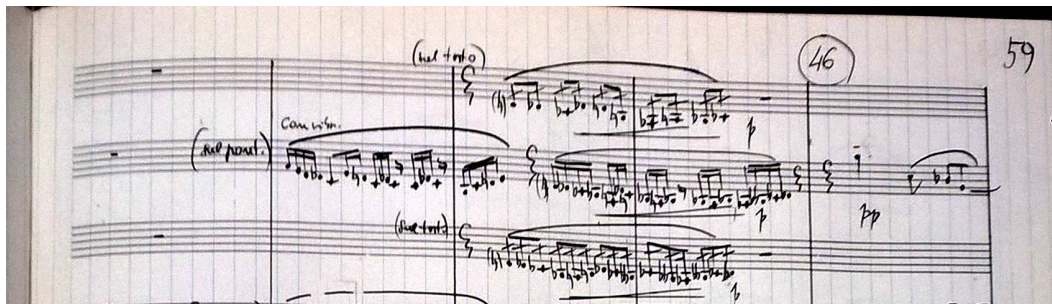
<sup>9</sup>Incidentally, this is the title of [Bainbridge, 1994], “A complete optical music recognition system: Looking to the future”.

2. **Staff detection and removal.** The staves (horizontal objects consisting, usually, of 5 equally-spaced lines) are the "spines" along which music is read, so detecting them provides basic information about the layout of the sheet music. They are often then removed from the image, as they are responsible for most of the object overlap and crossing; once staves are removed, segmentation can be done using some heuristics such as connected components. This is a step specific to processing music notation. The pipeline up to this step is depicted in Fig. 3.6.
3. **Object detection.** The individual notation objects are then detected, either in two steps (segmentation and classification) in earlier approaches, or detected directly in more recent works, using deep learning. In our view, the distinction from the previous step is mostly a practical issue, not one of principle – stafflines are also symbols that must be detected – but the methods for detecting stafflines have historically been distinct; this is due both to their distinct characteristics and the fact that most methods relied on finding and removing stafflines before the remaining objects could be found.
4. **Notation assembly and semantics inference.** Given the featural nature of music notation as a writing system, the relationships of the individual detected objects to each other must be added (such as: accidentals must be associated with the right note or grouped into a key signature, beams must be correctly assigned to noteheads, etc.) and the musical semantics can thus be inferred, by applying the rules of music notation.

The final step is to construct the output representation in the required format.

## Preprocessing

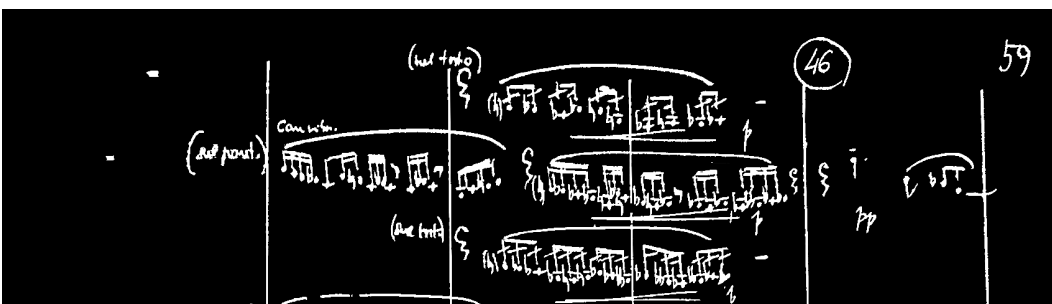
Preprocessing is a mostly practical stage that focuses on normalizing the input images in order for it to conform to the assumptions of the downstream parts of a given OMR system (e.g., de-skewing, so that staves are straight [Fujinaga, 1988]). The most important problem for OMR in this stage is binarization [Rebelo et al., 2012]: sorting out which pixels belong to the background, and which actually make up the notation. There is some evidence that with respect to binarization, sheet music does have some specifics [John Ashley et al., 2008], and consequently there have been attempts at OMR-specific binarization [Yoo et al., 2008, Pinto et al., 2010, 2011]. On the other hand, authors have attempted to bypass binarization, especially before staffline detection [Rebelo and Cardoso, 2013, Calvo Zaragoza et al., 2016b], as information may be lost with binarization that could help resolve symbol overlap or other ambiguities later. Other preprocessing requirements relate mostly to imperfections in the imaging process (e.g., uneven lighting, deformations of the paper; with mobile phone cameras, limited depth-of-field may lead to out-of-focus segments of the image) and the quality of the underlying document (degradation, stains, especially bleedthrough)



(a) Original image.



(b) Binarized image: notation pixels as foreground.



(c) After staff removal.

Figure 3.6: The standard OMR pipeline from the original image through image processing including binarization, and staff removal. While staff removal is technically part of symbol recognition, as stafflines are symbols as well, it has until very recently been considered practical to recognize stafflines separately.

[Byrd and Simonsen, 2015]. Other than possibly some specific binarization techniques, or rather estimating the optimal settings of general binarization techniques, preprocessing is not specific to OMR.

### Staff detection and removal

Staff detection and removal has seen a lot of activity, as it is a critical issue for OMR since its beginnings [Pruslin, 1966, Prerau, 1971, Fujinaga, 1988]. It is the only part of OMR where a competition has been organized [Fornés et al., 2011, Fornés et al., 2014], and an extensive dataset was created [Fornés et al., 2012].

There are practical motivations for considering staffline removal a separate step. Removing them significantly simplifies the topology of the foreground regions, to



the extend that connected components become a useful (if imperfect) heuristic for pruning the search space of possible segmentations [Fujinaga, 1996, Rebelo, 2012]. Furthermore, the vertical spacing of stafflines (staffspace height, measured in pixels) are the most important parameter that describes the scaling of the score: one can normalize scores by re-scaling to some fixed staffspace height (due to differences in staffline thickness relative to the height of a whole staff, using a sum of staffline and staffspace height is a more robust characteristic [Rebelo, 2012]).

Traditional staffline detection and removal methods exploit the fact that stafflines are by definition long and straight, or at least should be. The natural idea is to detect them by searching for peaks in horizontal projections [Pruslin, 1966, Prerau, 1971], notably also by Fujinaga [1988]. For imperfectly scanned scores, de-skewing can be used as a step that makes projections perform better [Fujinaga, 1996]. An alternative to horizontal projections is line tracking, where adjacent vertical runs of foreground that approximately match the prevalent staffline height are considered stafflines [Dalitz et al., 2008a]. A more general approach that also applies to grayscale images, not necessarily only to binarized inputs, was attempted by Cardoso et al. [2009] and Rebelo et al. [2013], [Rebelo and Cardoso, 2013], search for shortest “stable paths” through foreground areas from the left edge of the score to the right, also based on the assumption that the stafflines are the only extensive horizontal foreground objects. However, more recently, these results have been almost entirely superseded by convolutional networks [Calvo Zaragoza et al., 2017c], [Gallego and Calvo Zaragoza, 2017], achieving robust results: both significantly outperforming previous results on the CVC-MUSCIMA dataset used for the competition [Fornés et al., 2012], and being applicable to different types of scores as well.

Because errors during staff removal make further recognition complicated, especially by breaking symbols into multiple connected components with over-eager removal algorithms, some authors skip this stage. An interesting idea is tried by Sheridan and George, who instead *add* extra stafflines to annul differences between notes on stafflines and between stafflines [Sheridan and George, 2004], Pugin interprets the stafflines in a symbol’s bounding box to be part of that symbol for the purposes of recognition [Pugin, 2006a]. Furthermore, recent object detection methods using deep learning (such as presented in this thesis) have been found to not require staff *removal* at all [Pacha and Calvo Zaragoza, 2018, Hajič jr. et al., 2018a, Pacha et al., 2018b].

## Object Detection

Object detection, whether with or without staves removed, has been attempted mostly in two steps: a segmentation or localization step first, and a classification step next. While classification of musical symbols has produced near-perfect accuracy for both printed and handwritten musical symbols [Rebelo, 2012, Chanda et al., 2014, Wen et al., 2016], with baseline classification algorithms on raw pixels as features achieving close to 80 % accuracy [Calvo Zaragoza and Oncina, 2014], segmentation of hand-

written scores remains elusive. After stafflines are removed (and if they are removed well), one can start by using connected foreground components as object candidates. However, while this heuristic does prune the search space for possible segmentations to a great extent, it is still the case that (1) multiple notation objects are part of a single connected component (such as beamed groups), (2) a single object is split into multiple components (such as the f-clef).

It must also be noted that different authors use different “alphabets” of music notation symbols. Some OMR researchers decompose notation into individual primitives (notehead, stem, flag) [Coüasnon and Camillerapp, 1994, Bainbridge and Bell, 1997, Bellini et al., 2001, Bainbridge and Bell, 2003, Fornés, 2005], while others retain the graphical “note” as a single visual object, and beamed groups are decomposed into the beam(s) and the remaining notehead+stem combinations of “quarter-like notes” [Rebelo et al., 2010, Rebelo, 2012, Pham et al., 2015]; in some literature that chooses this decomposition, beams are unfortunately not included at all [Calvo Zaragoza and Oncina, 2014, Chanda et al., 2014].

Most segmentation approaches such as projections [Fujinaga, 1988, 1996, Bellini et al., 2001] rely on topological constraints (such as: the notehead and stem touch) that do not necessarily hold also in printed music, much less in manuscripts. In response, a fuzzy approach to topological constraints has been proposed in [Rossant and Bloch, 2006], and morphological skeletons have been proposed instead [Roach and Tatem, 1988, Ng et al., 1999, Luth, 2002] as a basis for handwritten OMR. Recently, however, general object detection methods based on deep learning (one of which is a part of this thesis) have brought previously unseen performance [Pacha et al., 2018b] that has since improved further.

## Notation Assembly

The locations and classes of symbols on the page becomes the input to the **notation assembly** stage. Recall that music notation is a featural writing system: the essence of notation assembly lies in inferring the symbol *configurations* from the individual symbols and their locations.

However, it is not clear what the output of this stage is, or rather: this output heavily depends on the assembly approach taken. This is because at this point in the recognition pipeline, one must start thinking about how to formally represent music notation. In replayability-oriented applications, one may decide that no explicit representation is needed [Shi et al., 2017, van der Wel and Ullrich, 2017, Calvo Zaragoza et al., 2017b, Calvo Zaragoza and Rizo, 2018]. However, in other cases, it is necessary to have a formal model of music *notation* in mind.

One such formalism are *context-free grammars*. This approach is rooted in the fact that music notation can be hierarchically decomposed, corresponding well to the notion of a non-terminal symbol. A page is split into systems, systems into staves, staves into measures, measures into notes, etc. Furthermore, there are strong visual syntactic rules for how to write valid music notation: e.g., every full notehead must



have an associated stem; the stem is supposed to touch the notehead on the rightmost point (if it is pointing upwards), or leftmost (if pointing down); the half-rest is on top of the middle staffline, the whole rest is positioned “hanging” from below the 2nd staffline from the top; an inline sharp is at the height of the notehead, etc., that invite this line of thinking: one can easily imagine generation with non-terminals such as `quarter_note -- > {notehead-full, stem}` with additional attributes to make sure that, e.g., stems point in the right direction.

Using context-free grammars has been first attempted already in 1982 [Alfio Andronico and Alberto Ciampa, 1982] and several times since [Coüasnon and Camilleri-app, 1994, Coüasnon and Rétif, 1995, Bainbridge and Bell, 2003, Szwoch, 2007], as it offers an elegant formalism with established inference algorithms. However, although it does to some extent simulate the human process of writing music (“I need to write a G4# quarter note” translates at the graphical level to “write a full notehead on 4th staffline, stem pointing upwards, sharp on the left of notehead”), the intuitively appealing top-down hierarchical decomposition of music notation into a tree structure is not necessarily an adequate representation of music notation itself: for instance, in the (relatively frequent) situation where two voices share a notehead, one either has to “invent” an overlapping notehead symbol so that the parse tree<sup>10</sup> remains a tree, or let subtrees share leaves. This is a problem for parsers, as they rely on a pre-computed segmentation of the input. In response, graph grammars have been used [Fahmy and Blostein, 1993, Baumann, 1995, Reed and Parker, 1996, Fahmy and Blostein, 1998]. The core idea of graph rewriting is also being used by the Audiveris open-source OMR system [Bitteur, 2004].<sup>11</sup> However, the interest in such unified formalisms (and notation assembly in general) seems to have waned after 2000, in exchange for increased focus on staff removal and symbol classification.

## Constructing the output representation

Once the score is fully described and the musical semantics are inferred, what remains is to store the given score in the desired output format. For replayability-oriented applications, this format is usually MIDI and the export is straightforward, from the list  $\langle \text{pitch, onset, duration} \rangle$  triplets that the OMR system has inferred. For reprintability-oriented applications, where a digital representation of the score itself is expected, the situation is more complex.

One group of music encoding formats are plaintext. The oldest of these formats,

---

<sup>10</sup>**TODO: explanation**

<sup>11</sup>A graph is also used by [Chen et al., 2015a]: a graph is built with edges directly connecting some notation primitives, but this was done for the purposes of preserving layout constraints when stretching and otherwise manipulating a score *without* fully recognizing it.

DARMS,<sup>12</sup> `**kern`,<sup>13</sup> the LilyPond<sup>14</sup> format for engraving, ABC,<sup>15</sup> or NIFF.<sup>16</sup> The problem with these formats – except for LilyPond – is that they are actually more suitable for replayability than for storing the score itself, as they prioritize saving the musical semantics and then having them rendered via sensible typesetting defaults, rather than to store what the score actually looked like. More complex, but also sufficiently powerful formats for storing music notation, are the XML-based MusicXML<sup>17</sup> and MEI.<sup>18</sup> These formats are the state of the art for describing the *score* itself (of course including the semantics).

The individual formats are each suitable for a different purpose: for instance, MIDI is most useful for interfacing different electronic audio devices, MEI is great for editorial work, LilyPond allows for excellent control of music engraving. Many of these have associated software tools that enable rendering the encoded music as a standard musical score, although some – notably MIDI – do not allow for a lossless round-trip.<sup>19</sup> Furthermore, evaluating against the more complex formats is notoriously problematic [Szwach, 2008, Hajič jr. et al., 2016].

As the notation assembly step should resolve any remaining ambiguity, constructing the output representation should remain an engineering task (even though it may still be complex), not a part of the OMR process per se. However, this still depends on having an appropriate formalism for notation assembly output.

## End-to-end OMR

With the advent of deep learning methods that require barely any feature engineering, a different approach than decomposing the problem into the standard pipeline can be taken: *end-to-end* recognition, where the intermediate stages of the process are not done explicitly and the corresponding intermediate results – especially the individual symbols and their locations – are never recorded. As the object detection subproblem is in principle hard (see section 3.1) in OMR, including issues with properly defining the set of symbols (see subsection 3.2.1), this approach is particularly appealing. It also widens the possibilities of using synthetic data generated on the fly during training. Recurrent networks offer the possibility of dealing with the long-range dependencies inherent in music notation, such as remembering which clef or key signature is valid for the particular location in the score.

---

<sup>12</sup><http://www.ccarh.org/publications/books/beyondmidi/online/darms/> – under the name Ford-Columbia music representation, was the output of the DO-RE-MI system of [Prerau 1971].)

<sup>13</sup><http://www.music-cog.ohio-state.edu/Humdrum/representations/kern.html>

<sup>14</sup><https://lilypond.org>

<sup>15</sup><http://abcnotation.com/>

<sup>16</sup><http://www.music-notation.info/en/formats/NIFF.html>

<sup>17</sup><http://www.musicxml.com/>

<sup>18</sup><https://music-encoding.org>

<sup>19</sup>That is: when converting a file from format A to B and then back from B to A, the result will not necessarily contain all the information of the original file in format A.

Already before the advent of deep learning, the end-to-end approach has been elegantly applied using Hidden Markov Models by Pugin [2006a], for the recognition of monophonic mensural notation printed with movable type. For monophonic music, this approach was presented first by Shi et al. [2017] as a side note for a recurrent-convolutional model; an encoder-decoder model was used by van der Wel and Ullrich [2017]. Calvo Zaragoza et al. [2017a] use a recurrent-convolutional model with Connectionist Temporal Classification loss. Unfortunately, no end-to-end models have so far been developed for polyphonic, much less pianoform music.

### Interactive OMR

For replayability-oriented applications where the OMR output is supposed to be used as performance material for musicians, no errors are tolerated, and therefore OMR outputs will always be held “under suspicion” until reviewed and cleared by a qualified editor [Raphael and Wang, 2011]. Since the application requires human intervention anyway, there is little reason to limit the intervention to the endpoint of the recognition process, especially since low-level errors early in the recognition pipeline can have severe implications [Bellini et al., 2007], it would be useful to catch these errors as they happen, saving subsequent editing effort. This line of thought leads to *interactive* OMR systems, where the user is invited to intervene along the pipeline. Fujinaga [1996] proposed an adaptive system that learned from user feedback over time; the ideas have been implemented in the Gamera framework [Droettboom et al., 2002]. More recently, this framework has been adapted into the Rodan online infrastructure that allows for arbitrary interactive pipeline steps in the browser [Hankinson, 2014]. Outside of the Gamera/Rodan effort, Church and Cuthbert [2014] created an interface to let users correct misrecognized rhythmic patterns using correct measures elsewhere in the score. In contrast to these post-editing approaches, Calvo Zaragoza et al. [2016a] combine the musical score image with the signal from pen-based “tracing” of the symbols, merging the offline and online modalities of OMR. Chen and Duan [2016] incorporate human guidance directly into the recognition process, by letting the user control what elements of notation are allowed, in order to avoid false positives for rare situations that the editor can rule out for the given page; the resulting CERES tool allows quick re-recognition and incorporates visual feedback. What has *not* been attempted yet is Interactive OMR guided by audio input, even though playing the music in question seems to be the fastest and most natural way of providing user feedback: after all, musical instruments are exactly the interfaces intended for the interpretation of the musical score. Closest is the work on tracking audio in sheet music [Dorfer et al. [2016b, 2018b]].

### Online OMR

With the advent of touch-operated devices, especially in the realm tablets, there has been interest in *online* OMR that takes as its input signal the trajectory of a pen

[Anstice et al., 1996, Miyao and Maruyama, 2004, Mitobe et al., 2004, Tsandilas, 2012, Calvo Zaragoza and Oncina, 2014, 2015, Calvo Zaragoza et al., 2016a, Calvo Zaragoza and Jose Oncina, 2017, Sober Mira et al., 2017]. The advantage of this approach is that much more information is available to the OMR system: individual pen strokes are an important pre-segmentation heuristic, the order in which strokes are done will also be predictive of their meaning [Calvo Zaragoza and Jose Oncina, 2017]. This approach cannot on the other hand deal with the already accumulated body of written works. However, an elegant idea is to use online OMR to speed up data acquisition for offline OMR [Calvo Zaragoza et al., 2016a]: the user traces notation that has already been written on a touch interface, and the system thus has multiple signals available. This is much faster than tracing the notation elements individually, and it might make it feasible for untrained annotators to quickly create in-domain datasets for specializing OMR systems for individual collections. The MuRET tool by Rizo et al. [2018] implements this process.

### Partial OMR

While transcription of individual scores for performance purposes will remain a major target application of OMR for musicians and composers, some application scenarios do not require full-scale automated transcription of sheet music – in response to specific needs of OMR users (musicians [Dorfer et al., 2016b] and musicologists [Hankinson et al., 2012], music pedagogy [Sébastien et al., 2012] and audiences and the general public [Ringwalt et al., 2015]), and as partial steps towards a complete OMR system. Hankinson et al. [2012] argue – very rightly so, we believe – that OMR should expand beyond the transcription application for individual users and towards retrieval from large collections. Retrieval has been a primary focus of the SIMSSA project [Fujinaga et al., 2014, Hankinson, 2014], producing e.g. the Liber Usualis project [Thompson et al., 2011]. Cross-modal retrieval using OMR has also been attempted by Damm et al. [2008] and Fremerey et al. [2009], later<sup>20</sup> by Balke et al. [2015], who specifically cites the low quality of OMR outputs as a bottleneck for this kind of application. In fact, the most successful sheet music retrieval system so far bypasses extracting musical semantics and instead learns a joint embedding space for image and spectrogram snippets [Dorfer et al., 2016a, 2018a].

### 3.2.2 Infrastructure of OMR

The individual steps of this pipeline have garnered the most attention in OMR literature, but three more important areas must be mentioned which underpin the overall state of the art: datasets, evaluation, and software.

Datasets have been scarce. There was no openly available dataset for object detection, for instance; much less for the full recognition pipeline. The only extensive

---

<sup>20</sup>By the same research group under Audiolabs Erlangen.

dataset that has been available is the CVC-MUSCIMA staff removal and writer identification collection of 1000 scores (and eleven distortions, for a total of 12 000 images) by [Fornés et al. \[2012\]](#). For symbol *classification* (not detection!), the HOMUS dataset [\[Calvo Zaragoza and Oncina, 2014\]](#) was the most extensive, with the advantage of providing inputs in both offline and online flavors, and also the only such dataset that was publicly available; even so, it contained only 32 different symbol classes (with the core alphabet of music notation, disregarding text, having more than 50 such classes).

During the last three years, the dataset situation has seen marked improvement. The first significant addition was the MUSCIMA++ dataset [\[Hajič jr. and Pecina, 2017c\]](#), which still remains the only dataset for full-pipeline recognition and for CWMN manuscripts and is one of the key contributions of this thesis, but for symbol detection and partial semantics inference, there is the much more extensive – though printed and synthetic – DeepScores [\[Tuggener et al., 2018\]](#). Third, the Capitan dataset of mensural notation has been made available [\[Pacha and Calvo Zaragoza, 2018\]](#) that supports symbol detection, although not semantics inference at this point.

While the dataset situation has gone from being a blocking issue to being a non-issue for the progress of the field, at least for the time being, evaluation remains a problem. While it is possible to evaluate individual sub-problems of the OMR pipeline and clear up-to-date methodologies exist [\[Fornés et al., 2012\]](#), [\[Rebelo, 2012\]](#), [\[Calvo Zaragoza and Oncina, 2014\]](#), [\[Pacha and Calvo Zaragoza, 2018\]](#), [\[Hajič jr. et al., 2018a\]](#), evaluating an OMR system as a whole is still problematic [\[Byrd and Simonsen, 2015\]](#), [\[Hajič jr. et al., 2016\]](#). The most extensive such effort was probably undertaken by [Bellini et al. \[2007\]](#), who directed annotators to manually label OMR mistakes according to a detailed list of possible errors; a different approach was undertaken by [Szwoch \[2008\]](#) and [\[Hajič jr. et al., 2016\]](#), which seeks to develop automated metrics for directly comparing files in the MusicXML output format. A significant part of the problem may be a lack of appropriate formalisms for describing what the distance between two music scores is, and a lack of appreciation for the many purposes for which OMR is used; the corresponding analysis and discussion is a part of this thesis (see sections [6.1](#), [6.5](#)): separate evaluation methodologies have to be applied for different tasks.

Finally, we mention software tools for OMR research. The situation here is actually better than in evaluation and datasets. The veritable Gamera system [\[MacMillan et al., 2001, 2002\]](#) has held reference implementations of various OMR methods, for instance staff removal up until the machine learning-based contributions of the last four years; it has been used to build also an OMR system for lute tabatures [\[Dalitz and Karsten, 2005\]](#) and Byzantine chant notation [\[Dalitz et al., 2008b\]](#). Aside from Gamera, there is the Audiveris generic open-source system [\[Bitteur, 2004\]](#) and the Aruspix system for processing early music prints [\[Pugin, 2006b\]](#), [\[Pugin et al., 2008\]](#). In recent years, the OMR pipeline has been marshalled by the SIMSSA project [\[Fujinaga et al., 2014\]](#) under the Rodan system [\[Hankinson, 2014\]](#), which is openly available.

Given the dominance of machine learning in computer vision, this includes interactive editors for creating ground truth: within the SIMSSA system, these are editors such as Pixel.js [Saleh et al., 2017] for creating pixelwise ground truth for binarization and staff detection, or Neume.js [Burlet et al., 2012] for manipulating recognition outputs of square notation of neumes. This thesis also contributes the MUSCIMarker editor for general object detection ground truth, including pixel-based masks – see section 6.3; recently, the MuReT tool was also released that facilitates also pen-based data acquisition [Rizo et al., 2018].

### 3.2.3 Commercial Software

Finally, we must touch on the available commercial software. The biggest players are PhotoScore<sup>21</sup> and SmartScore<sup>22</sup>, each integrated into one of the major commercial notation editors (PhotoScore in Finale, SmartScore in Sibelius). Given the sorry state of OMR evaluation and the “black box” nature of commercial software, it is not possible to measure their performance with more accuracy than anecdotal evidence. This anecdotal evidence suggests that at least for high-quality printed scans, the performance of all commercial software has improved significantly during the last several years, to the extent that they can now actually be used in practice. However, at the time of writing only PhotoScore offers manuscript recognition functionality, and it is rather bad. The Audiveris software is being gradually integrated into the MuseScore open-source notation editor<sup>23</sup>. For online OMR, the Neuratron NotateMe<sup>24</sup>, StaffPad<sup>25</sup> and the MyScript back-end service<sup>26</sup> are available, and again the estimates of their usefulness are at best anecdotal and uncertain.

---

<sup>21</sup><http://www.neuratron.com/photoscore.htm>

<sup>22</sup><http://www.musitek.com/index.html>

<sup>23</sup><https://www.musescore.com>

<sup>24</sup><https://www.neuratron.com/notateme.html>

<sup>25</sup><https://staffpad.com>

<sup>26</sup><https://developer.myscript.com/music>



## 4. Contributions

The thesis contributes to three areas of the field of Optical Music Recognition: theoretical advances (T), OMR resources (R), and methods (M). Each of these contributions is attested to by published works<sup>[1][2]</sup>

- (T1) Better definition of what Optical Music Recognition is; a detailed analysis of the field’s structure, objectives, and difficulties.  
(*Corresponding manuscript under review; see section 6.1.*)
- (T2) The Music Notation Graph (MuNG): an elegant formal description of the music notation visual language using a directed graph.  
[Hajič jr. and Pecina, 2017c]
- (R1) The MUSCIMA++ dataset, which is the first OMR dataset that supports full-pipeline recognition; it contains over 90 000 annotated objects and a similar number of their relationships as notation graphs (T2).  
[Hajič jr. and Pecina, 2017c]
- (R2) The mung software package for manipulating the MuNG representation of music notation (T2), musical semantics inference, and MIDI export.<sup>3</sup>  
[Hajič jr. and Pecina, 2017c]b, [Hajič jr. and Dorfer, 2017]
- (R3) The MUSCIMarker annotation and MuNG visualization tool.<sup>4</sup>  
[Hajič jr. and Pecina, 2017b], [Hajič jr. and Dorfer, 2017]
- (R4) The omreval corpus that can be used for testing OMR extrinsic evaluation metrics against human preferences; the corpus contains 100 human judgments per annotator, with a total of 15 annotators.  
[Hajič jr. et al., 2016]
- (R5) A tutorial on Optical Music Recognition presented at the ISMIR 2019 conference, available as a YouTube playlist.<sup>5</sup><sup>6</sup>
- (M1) Notehead detection with bounding box regression vs. semantic segmentation approaches.  
[Hajič jr. and Pecina, 2017a]

---

<sup>1</sup>In case of T1, a submitted journal manuscript under review at the time of thesis submission.

<sup>2</sup>The role of the thesis author in the individual published works is detailed in the respective sections in part III

<sup>3</sup><https://github.com/OMR-research/mung>

<sup>4</sup><https://github.com/OMR-research/MUSCIMarker>

<sup>5</sup><https://youtube.com/playlist?list=PL1jvwDVNwQke-04UxzlyY4FM33bo1CGS0>

<sup>6</sup>Note that ISMIR tutorials involve writing an extended abstract that undergoes review.

- (M2) General music notation object detection with U-Nets (semantic segmentation).  
[Hajič jr. et al., 2018a]
- (M3) Notation assembly using pairwise MuNG edge/non-edge classification.  
[Hajič jr. and Pecina, 2017c, Hajič jr. et al., 2018a]
- (M4) Full pipeline combining (M1-3) and (R2) that produces MIDI from musical manuscripts (R1), with a graphical interface with MUSCIMarker (R3).  
[Hajič jr. and Dorfer, 2017, Hajič jr. et al., 2018a]

Aside from the items listed above, the author of the thesis further contributed to the field by serving as one of the General Chairs of the 1st International Workshop on Reading Music Systems (WoRMS),<sup>7</sup> a satellite event of the ISMIR 2018 conference.<sup>8</sup>

In terms in which we have introduced OMR in chapter 3, this thesis focuses on **offline handwritten OMR**, general enough to cover both replayability and reprintability, although with focus especially on the former.<sup>9</sup> Given the state of the field when work on this thesis was starting, in order to address this task, a substantial effort was necessary before state-of-the-art computer vision methods could be adapted for the purpose of OMR. Most critically, since these rely on supervised learning, it was necessary to produce datasets. In the rest of this section, we describe the contributions in more detail and link them with the published work that forms the substance of this thesis.

**The key idea of this thesis is the Music Notation Graph (T2), which was used for designing and creating the MUSCIMA++ dataset (R1), which in turn enabled experiments leading to the recognition pipeline (M4).** We proceed by explaining this progression. The purpose of the following text is not to give the full technical details (these are given in the published works in the next part of the thesis); rather, it aims to explain the backbone of this thesis.

## 4.1 Music Notation Graph

We have already prepared ground for the idea of the Music Notation Graph (MuNG) through the discussion of notation assembly methods and their limitations in section 3.2.1. A shared property of all the context-free grammar approaches [Alfio Andronico and Alberto Ciampa, 1982, Coüasnon and Camillerapp, 1994, Coüasnon and Rétif, 1995, Bainbridge and Bell, 2003, Szwoch, 2007] and graph rewriting systems [Fahmy and Blostein, 1993, Baumann, 1995, Reed and Parker, 1996, Fahmy and

<sup>7</sup><https://sites.google.com/view/worms2018>

<sup>8</sup><https://https://ismir2018.ircam.fr/pages/events-at-a-glance.html>

<sup>9</sup>These “terms in which we have introduced OMR” have to some degree already existed in OMR literature previously, but a substantial effort was necessary in order to tie them into a coherent whole; this is under contribution T1 and subject of section 6.1



Blostein, 1998, Bittour, 2004] in OMR so far is that they infer non-terminal “invisible” symbols that correspond to the hierarchy of abstract notation concepts (note, measure, voice...). This derives from the context-free grammar approach of building constituency trees. However, while this hierarchical approach is certainly appealing, especially given that this is how one usually learns to think about music and music notation, is it the best one can do in OMR? We of course propose that the answer is – *no*.

Rather than what could be termed a “constituency graph” of the previous approaches, in an analogy to the Prague school of Computational Linguistics, we apply the notion of a *dependency graph*. Instead of grouping music notation primitives under composite nodes, we link them to each other. We call this formalism simply a **Music Notation Graph** (abbreviated as MuNG). The vertices of this graph are music notation primitives (*not* notes!); oriented edges may link the vertices. The idea of assembling the music notation primitives into a notation graph is illustrated in Fig. 4.1.

*The MuNG representation is first described in the publication **The MUSCIMA++ Dataset for Handwritten Optical Music Recognition** [Hajič jr. and Pecina, 2017c], reproduced in section 6.2.*

Once this dependency graph is built, we can exploit the straightforward relationship of noteheads to the abstract musical notes and the rules of reading music, as described in chapter 2, to deterministically infer the musical semantics.

*The implementation of this semantics inference and associated MIDI export is first referenced in the demo paper **Handwritten Optical Music Recognition: A Working Prototype** [Hajič jr. and Dorfer, 2017], reproduced in section 7.6 and available as the `mung.inference` software package.*

Central to how MuNG is specified is the principle that each notehead-type node (full notehead, empty notehead, and all rests) has as its neighbors (immediate or close) all the notation primitives relevant to the decoding of the corresponding note. We have stealthily structured the introduction to musical semantics and music notation (chapter 2) to make this principle natural. Recall that noteheads are the interface between music notation and the encoded notes: there is one note per notehead.<sup>10</sup> The musical semantics for each note are fully encoded through *configurations* of symbols associated with each notehead: the staves and ledger lines, clef, key signature and inline accidentals encode pitch; the notehead type, stem, flags or beams, augmen-

<sup>10</sup>The sole exception being notehead sharing across multiple voices; however, this is detectable from the presence of multiple stems. In case whole notes are shared, this is typeset as two consecutive empty noteheads significantly closer to each other than if they were to be played consecutively.

tation dots and tuples encode its duration. Each of these elements can be simply captured by linking the notehead to the given primitive. The precedence relationships that are necessary to compute the onsets of notes can be captured as precedence edges in the notation graph that link noteheads which should be interpreted as consecutive notes.

A significant advantage of this approach is that the separation between music *notation* as a visual language and musical *notes* and their semantics as abstract objects is retained: the notation graph is merely a description of the music notation on the page. The entire process of inferring semantics from the MuNG output of our notation assembly stage happens independently from the underlying image – and at the same time, all the information available in the score is fully disambiguated even *before* one starts thinking about the musical semantics. This separation makes it possible to deal with the image separately from the music encoded therein, and we conjecture that maintaining this principle is what allows formulating the OMR pipeline in terms of straightforward machine learning tasks (see section 4.3).

*This strict separation between the graphical level and the musical semantics, so that the score is described as a visual object without using any of the abstract musical concepts such as voice, and yet sufficiently to enable unambiguously inferring the semantics, is motivated by the thorough analysis of the internal structure of OMR that is part of the submitted manuscript **Understanding Optical Music Recognition** in section 6.1.*

The *disadvantage* of the dependency graph approach is that there are no tractable algorithms that we know of for generic graph inference from an image. However: it seems that these may not be required. First, for object detection, state-of-the-art generic models are capable of leveraging the neighborhood of an object to disambiguate it (such as the staccato dot, which is written below or above its corresponding notehead, vs. the augmentation dot, which positioned is to the right of a notehead or rest) *without* having explicit access to syntactic information (as indicated by Pacha et al. [2018b], Hajič jr. et al. [2018a] and especially by Pacha et al. [2018a]). With respect to notation assembly, we can make a strong independence assumption – that given the vertices of the graph (the music notation primitives), the edges of the graph are independent. This allows formulating the notation assembly as a binary classification problem over vertex pairs. On an average page of some 500–800 symbols, this would still amount to 250 000–640 000 decisions; however, in practice there are reasonable assumptions (such as the maximum distance between objects that may be related, and constraints on linked symbol classes) that help prune the space of decisions to an asymptotically linear instead of quadratic number of classifier runs. This already allows bringing the full power of current machine learning methods to bear on the assembly problem: already decision trees with simple features (bounding box relative distance and symbol class labels) achieve useful results, as described

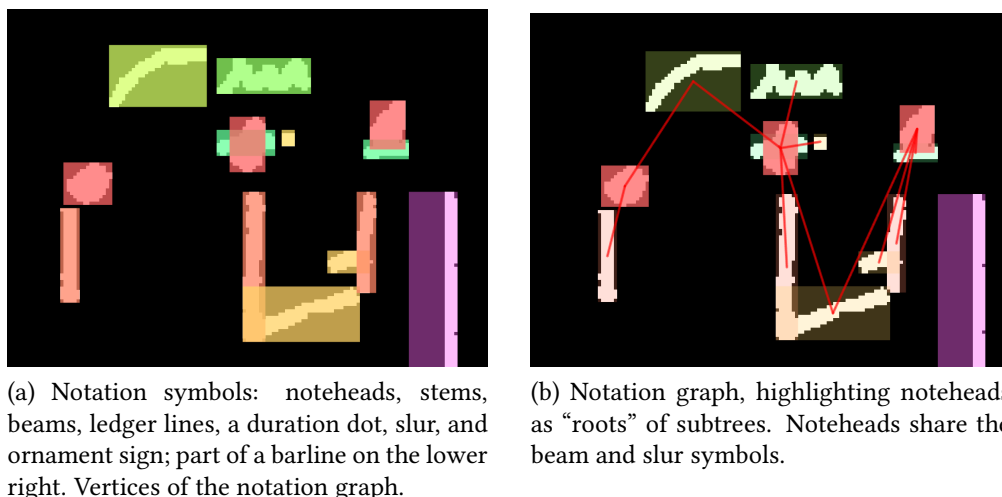


Figure 4.1: Visualizing the detected symbols and the assembled notation graph on top of staff removal output. Colors of symbol bounding boxes encode symbol classes (noteheads in red, stems in orange, ledger lines in green, etc.). Using the edges of the notation graph in (b), the pitch and duration of the notes encoded by the noteheads (highlighted) can be unambiguously inferred (stafflines removed for clarity, although for encoding pitch, we would need to establish the relationship of the noteheads to stafflines). Assuming the music is monophonic, onset can be inferred from the ordering of the noteheads and the notes’ durations.

below in section 4.3.2. Furthermore, ongoing experiments with factoring the MuNG inference process from detected objects into independent decisions about individual edges seems to also provide satisfactory results.<sup>11</sup>

While the idea of MuNG and the notehead-centric definition is hopefully clear and clearly motivated, there still remains a plethora of details to take care of: dealing with key signatures, time signatures, measure separators, etc. Further principles of MuNG definition are described in the publication [Hajič jr. and Pecina, 2017c] in section 6.2, and in full detail the definition is available online in the form of annotation guidelines for the MUSCIMA++ dataset,<sup>12</sup> which will be the subject of the following section. An open-source mung Python package for manipulating notation graphs, which also implements musical semantics inference and MIDI export from MuNG, is also made available.<sup>13</sup>

<sup>11</sup>These experiments are not part of the published works; the results are tentative at best, but very promising: over ground truth objects, pairwise classification using neural networks achieves near-perfect results with just a few elementary training tricks, and fine-tuned Faster R-CNN based detectors have recently achieved surprisingly good results; the assembly classifier is currently being tested over detection results.

<sup>12</sup><https://muscimarker.readthedocs.io/en/latest/instructions.html>

<sup>13</sup><https://github.com/OMR-Research/mung>

## 4.2 MUSCIMA++

Now that we have introduced the Music Notation Graph formalism that is one key to building a successful full recognition pipeline, we turn to the second such key: creating annotated resources for supervised learning.

Recall from subsection 3.2.2 that prior to the work in this thesis [Hajič jr. and Pecina, 2017c], there was no substantial openly available dataset for full-pipeline OMR, and the only dataset for object detection was a 3222-symbol collection by Rebelo [2012] that is not openly available, as it contains scores by contemporary composers that are under strict copyright restriction. The only datasets available were for the staff removal stage, with only the CVC-MUSCIMA dataset of Fornés et al. [2012] being large enough to support supervised machine learning (20 pages, each copied by hand by 50 different writers, for a total of 1000 pages), and the HOMUS dataset of Calvo Zaragoza and Oncina [2014] for symbol classification (not localization). It was therefore necessary to build a dataset for full-pipeline OMR.<sup>14</sup>

The MuNG formalism provides a clear definition of the ground truth. Three steps remained:

- Selecting the musical manuscripts to be annotated;
- Implementing an annotation interface;
- Managing the annotation work.

The criteria for selecting manuscripts were to cover as much notation complexity as possible while sacrificing types of input variability that can be simulated (image degradations), and secondarily, ease of annotation. In the end, the ideal collection turned out to be a subset of the 1000 pages of CVC-MUSCIMA. The binarized images with staves removed enabled significantly speeding up accurate annotations of the pixel-wise masks of individual symbols; at the same time, the 20 pages of music in CVC-MUSCIMA range from monophonic to pianoform music and include a relatively large number of different rare, yet important situations, such as cross-staff beaming, time signature, key signature and clef changes in the middle of a staff, complex beamed groups, non-standard tuples, and even the oft-mentioned notehead sharing between voices.<sup>15</sup> Furthermore, having been created by 50 different writers, there is a large variability in handwriting styles, ranging from elegant and clear to very hasty. In keeping with the original name, CVC-MUSCIMA, but denoting that the new dataset offers much richer options, we name our dataset **MUSCIMA++**.

---

<sup>14</sup>The need for such a resource is attested to by the fact that since its publication of [Hajič jr. and Pecina, 2017c] in late 2017 until the submission of this thesis, five papers unrelated to the author have cited this work. <https://scholar.google.com/scholar?oi=bibs&hl=en&cites=4140859639233316810>

<sup>15</sup>The authors of CVC-MUSCIMA, [Fornés et al. [2012]], have truly covered a *lot* of various annoying notation situations in only 20 pages of music!

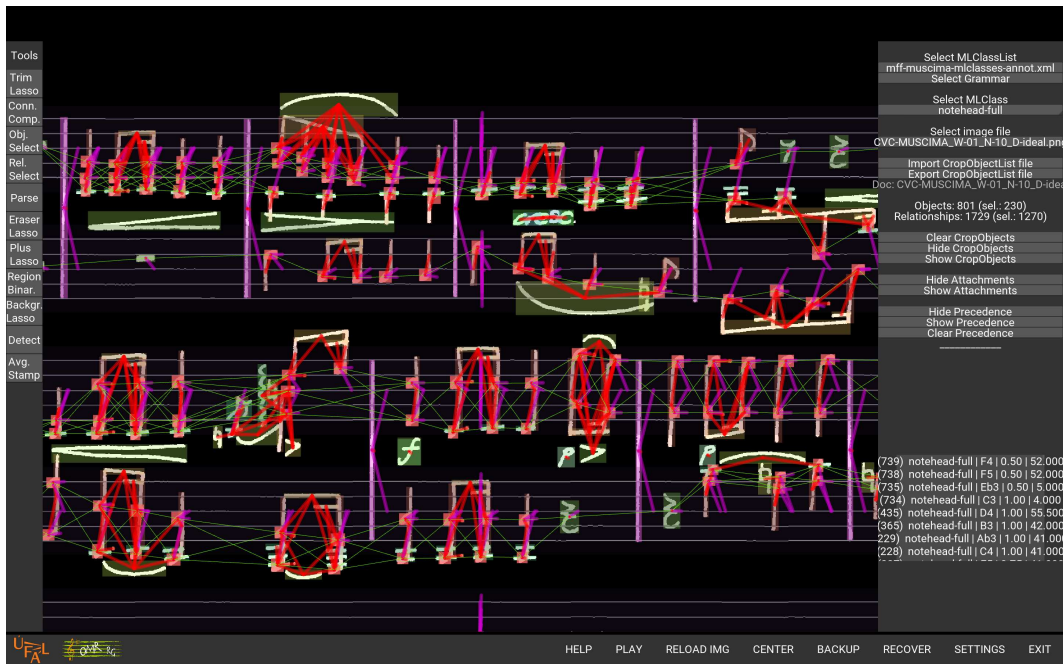


Figure 4.2: The interface of the MUSCIMarker tool. The notation graph in this example is taken from CVC-MUSCIMA image of page 10 by writer 01. Highlighted in red are manually added syntactic edges of the notation graph; in purple are automatically inferred edges pertaining to staff objects (stafflines, staffspaces and staves). Precedence edges are weakly visible in green.

*The MUSCIMA++ dataset is subject of the paper **The MUSCIMA++ Dataset for Handwritten Optical Music Recognition** [Hajič jr. and Pecina, 2017c], reproduced in section 6.2*

In order to actually create MUSCIMA++, an annotation graphical user interface had to be developed. There was no open-source software suitable for annotating dependency graphs over images; closest is probably the Aletheia document annotation software<sup>[16]</sup> but it still does not allow building graphs flexibly enough for the purposes of describing music notation. We created the **MUSCIMarker** tool, using the Kivy framework for the Python language<sup>[17]</sup> The interface is visualized in Fig. 4.2, with an example of a complex notation graph.

Managing the ground truth acquisition work itself involved recruiting and training qualified annotators (music or musicology students), performing quality control, maintaining and improving the annotation software (and its user documentation) based on annotator feedback, and postprocessing the outputs. In total, 7 annotators were working on the dataset<sup>[18]</sup> Each was to annotate one copy of each of the 20

<sup>16</sup><https://www.primaresearch.org/tools/Aletheia>

<sup>17</sup><https://github.com/OMR-research/MUSCIMarker>

<sup>18</sup>The choice of a self-contained cross-platform technology for MUSCIMarker proved fortunate: Windows, OS X and Linux operating systems were used by different annotators, including one in-

underlying pages in CVC-MUSCIMA, for a total of 140 pages annotated with MuNG ground truth. The pages were selected so that the handwriting of each writer appears at least 2 times and no more than 3 times. In total, there are 91 255 primitives in MUSCIMA++. Some 82 000 syntactic edges were manually marked (precedence edges and edges connecting symbols to their respective stafflines, staffspaces and staves were added automatically and corrected in postprocessing). This amounted to roughly to some 13 000 notation primitives and about 10 000 MuNG edges annotated per person. Annotators worked for a combined total of 400 hours, at an average speed of 4.3 symbols per minute, or one per 14 seconds; an average page of some 650 symbols took about 2 hours 45 minutes netto. Managing the annotation process (training annotators, distributing and collecting their work, and the first level of quality control) took an additional 150 hours, and the second, final round of quality control took an additional 80 hours.<sup>19</sup>

*The MUSCIMarker software can be referenced using the short paper **Groundtruthing (not only) Music Notation with MUSCIMarker: a Practical Overview** [Hajič jr. and Pecina, 2017b], reproduced in section 6.3.*

With the MUSCIMA++ dataset of binary images manually annotated with MuNG ground truth in hand, we may now proceed to build the recognition pipeline itself.

## 4.3 The Recognition Pipeline

The OMR pipeline in this thesis focuses on the later stages of the OMR pipeline: object detection, and notation assembly and semantics inference. We focus on a difficult setting in terms of processing manuscripts of arbitrary notation complexity, rather than on difficulties regarding image quality (which are of course in practice equally important, but not as inherent to the domain of music notation). The input images for our pipeline have already been binarized, and stafflines have been detected (and, if need be, removed). This is no more an entirely unreasonable expectation: convolutional networks have been shown to perform “layout analysis” (essentially, joint staffline detection and binarization: semantic segmentation into background, stafflines, and notation symbols) very well [Calvo Zaragoza et al., 2017a,c,d, Gallego and Calvo Zaragoza, 2017].

### 4.3.1 Object Detection

We start the work on the recognition pipeline by testing state-of-the-art object detection techniques. As detecting music notation primitives is a difficult problem (section 3.1) that does not conform well to general assumptions of object detection mod-

---

stance of Windows XP.

<sup>19</sup>Interestingly, using MUSCIMarker, the work was completed *under* budget.



els (there may be a few hundred different objects in each image, out of which perhaps a hundred can belong to the same class; the objects are very close together, some can be unpredictably small or large, etc.), we start with models that make as few assumptions as possible about the objects they are attempting to detect.

Seeing as noteheads are the most important object to detect reliably, we focus our efforts there. We start using the (then) state-of-the-art object detection method: Region Proposal Networks, specifically Faster R-CNN [Shaoqing Ren et al., 2015], for the purposes of OMR. Instead of using a fixed grid of anchor boxes, however, it uses as anchor *pixels* those pixels that belong to the morphological skeletons of the images in the dataset. Noteheads do have a more or less fixed size in a score; however, since we eventually want our method to generalize to other object classes, instead of pre-set anchor box sizes, we try to regress to the bounding box sizes directly, by using four ReLU output units that denote the top, left, bottom, and right offset of the (proposed) notehead's bounding box from the given skeleton pixel. (The target offsets for non-notehead skeleton pixels are set to 0, and in the model, the offset ReLU outputs are multiplied by the value of the "objectness" sigmoid output, so that in case of skeleton pixels with very low probability of being part of a notehead, the predictions for bounding box offsets do not actually induce a loss.) The training inputs and outputs to the network are depicted in Fig. 4.3.<sup>20</sup> The network is otherwise rather small (two convolutional/pooling layer blocks and two convolutional layers; enlarging the network does not improve results).

*Notehead detection with domain-adapted Region Proposal Networks is the subject of the short paper **Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression** [Hajič jr. and Pecina, 2017a] (section 7.1).*

This method achieves good recall of 0.97, but rather woeful precision of only 0.81.<sup>21</sup> In order to improve, a separate post-filtering step is trained: proposed noteheads are classified according to features such as their average proposed bounding box sizes, the ratio of skeleton pixels classified as noteheads within the proposed bounding box, etc. A random forest estimator is trained against the outputs of the model on the validation dataset. The post-filtering slightly impairs recall (0.96) but significantly increases precision (to 0.97). An example of the model's output is in Fig. 4.4.

A different approach is to use **fully convolutional networks** for semantic segmentation (assigning a label to each pixel) and a subsequent detection stage (such as peak picking, or thresholding and connected components). Specifically, we used

---

<sup>20</sup>This is loosely analogous to the YOLOv3 approach to bounding box regression [Redmon and Farhadi, 2018]; had YOLOv3 been published in 2015, we would have used it.

<sup>21</sup>Recall is computed as proportion of instances correctly detected out of all ground truth instances; precision is computed as the proportion of correctly detected instances out of all *detected* instances. In other words, recall penalizes false negatives, precision penalizes false positives. These standard metrics were first defined in [James W. Perry et al., 1955].

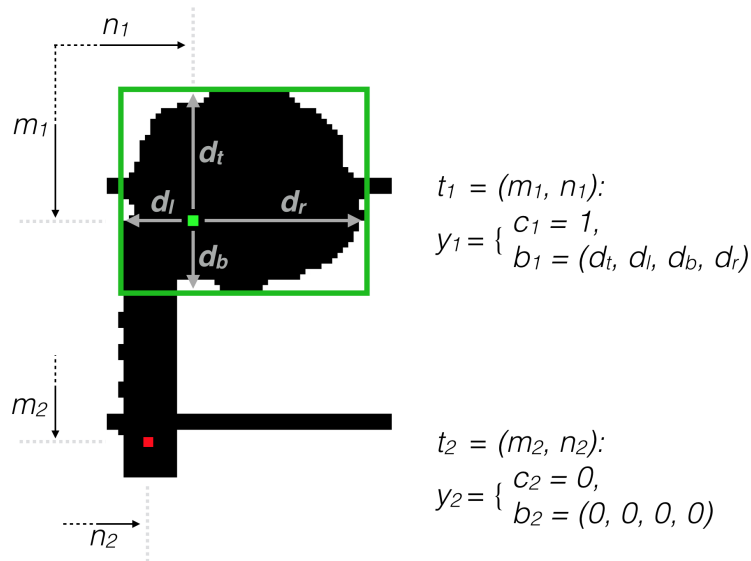


Figure 4.3: The design of the bounding box-based detector. It is learning for each anchor pixel, the green  $t_1$  and the red  $t_2$ , its class  $c$  (whether the anchor pixel is part of a notehead or not) and the offsets  $b$  of the corresponding bounding box – all 0 in case of pixel  $t_2$ , as it is not part of a notehead. (Figure taken from [Hajić jr. and Pecina, 2017a].)

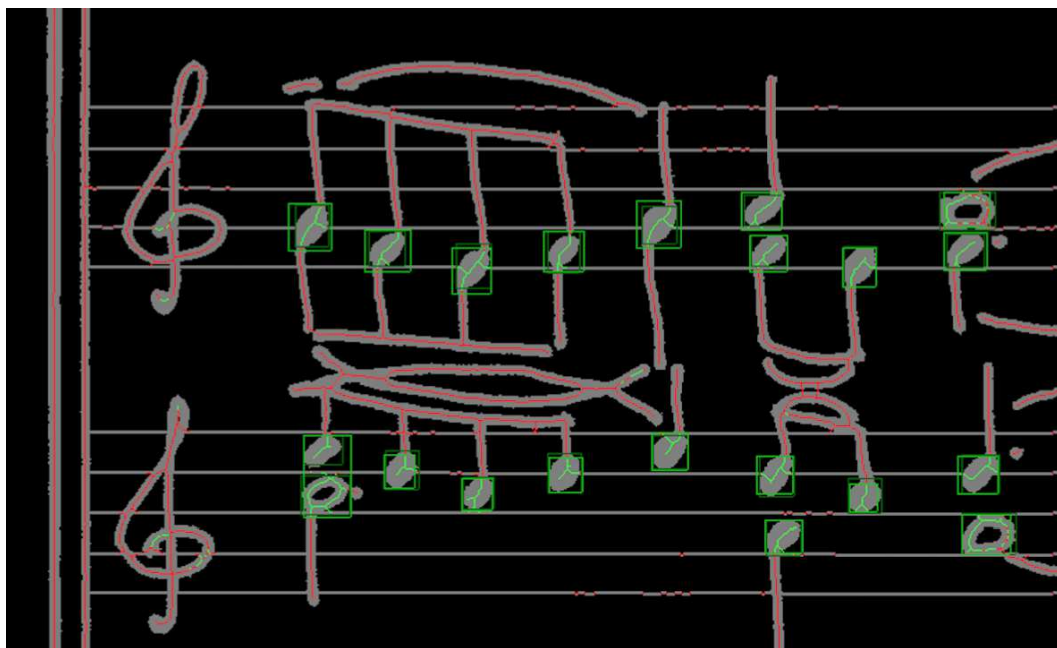


Figure 4.4: An example results of the RCNN-based detector. Note that while the network itself would detect some false positives in the G-clefs on the left side of the image, the post-filtering steps discards these candidates. On the other hand, an obvious error is caused by bounding box regression is the merging of bounding boxes for the leftmost bottom notes.



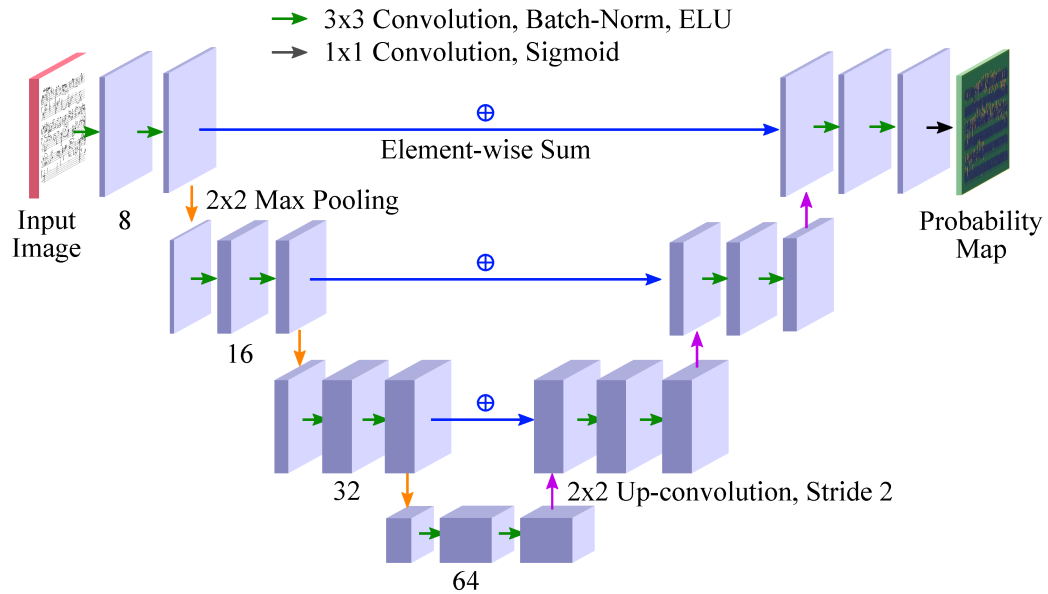


Figure 4.5: The U-Net architecture. Computation flows left-to-right; the “hourglass” shape is unrolled downwards with each 2x2 Max-Pooling layer (orange arrows); in the other direction are 2x2 up-convolution layers with a stride of 2. Blue arrows indicate the residual connections (implemented simply as elementwise sums) between blocks of corresponding sizes. (Figure taken from [Pacha et al., 2018b].)

the U-Net model [Ronneberger et al., 2015]. This model has an “hourglass” architecture reminiscent of autoencoders, but it is standard feedforward network; the output layer provides a value for each pixel (in this case, the probability of the given pixel belonging to the given symbol class). Given that the stages of the hourglass have the same size, residual connections are added between the corresponding stages. The network architecture is shown in Fig. 4.5. Without any post-filtering step, merely with thresholding at 0.5 and non-maxima suppression as the detection step on top of the probability map output by the model, the U-Net achieved on noteheads a recall of 0.97 and precision 0.99.

*Notehead detection with U-Nets is described in the short paper **On the Potential of Fully Convolutional Neural Networks for Musical Symbol Detection** [Dorfer et al., 2017] (section 7.2).*

Given this convincing advantage of the U-Net on noteheads, which are the most important object to detect, we chose to follow up on the fully convolutional model and build general object detection based on the U-Net architecture.

Noteheads are relatively easy to detect, relative to other symbols, because their appearance is very distinct (by design: they should be the first thing that attract the eye of a musician!), but some other symbols present tricky detection issues, especially in handwriting. Some fixed-size symbols such as clefs are visually quite complex and

very variable, and usually overlap with stafflines. Furthermore, clefs are by far not as frequent as noteheads, even though they are critical for decoding the semantics of all subsequent notes (as they define how stafflines are interpreted with respect to pitch). While these symbols at least have a (relatively) fixed size, there are others that have in principle a relatively simple shape (straight thick line), but their size is variable (stems, especially in music with wide chords), or even size and orientation (beams, which are probably the most variable symbol class, and slurs, which can theoretically be extremely complicated, although in practice they rarely have an inflection point). The choice of U-Nets is also motivated by the fact that they do not have any hyperparameters related to symbol sizes and locations besides the size of the receptive field of output pixels.

The major drawbacks of U-Nets for musical symbol detection is that these models only perform semantic segmentation, not object detection per se: a detector must be added on top of the output symbol probability map. Two simplest options are thresholding and connected component search, and thresholding and non-maxima suppression. Since non-maxima suppression is prone to leave false positives in long symbols such as stems or slurs, and in larger complex symbols such as clefs, we choose connected component search, with thresholding at 0.5<sup>22</sup>. Note that using a connected component detector implies that the model may merge objects from the same class that legitimately touch (such as some handwritten noteheads in chords) into a single symbol.

In a comparison to other general object detection models, Faster R-CNN [Shaoqing Ren et al., 2015] and RetinaNet [Lin et al., 2017], the advantages of U-Nets do result in better performance [Pacha et al., 2018b], as illustrated in Table 4.1<sup>23</sup>.

	mAP / w-mAP (%)		
	DeepScores	MUSCIMA++	Capitan
<b>Faster R-CNN</b>	19.6 / 14.4	3.9 / 7.9	15.2 / 23.2
<b>RetinaNet</b>	9.8 / 1.9	7.7 / 4.9	14.5 / 34.9
<b>U-Net</b>	24.8 / 17.4	16.6 / 23.3	17.4 / 26.0

Table 4.1: Results in terms of mAP (%) and w-mAP (%) with respect to the dataset and object detector model following the COCO evaluation protocol. (Table reproduced from [Pacha et al., 2018b].)

<sup>22</sup>Changing the threshold did not lead to improvements.

<sup>23</sup>The results are evaluated using Mean Average Precision (mAP) and Weighted Mean Average Precision (w-mAP), according to the object detection practices for the COCO dataset [Chen et al., 2015b]. The “mean” is taken over average precisions with true positives considered using different intersection-over-union thresholds: the most permissive is 0.5, and, using increments of 0.05, the cutoff for considering a detected object a true positive, the minimum intersection-over-union it must share with a ground truth object of the given class increases up to 0.95. In the weighted variant, the object classes are weighed by their support.

The comparison of generic object detectors across the available OMR datasets is subject of the paper *A Baseline for General Music Object Detection with Deep Learning* [Pacha et al., 2018b] (section 7.4).

The advantage disappears for the Capitan dataset [Pacha and Calvo Zaragoza, 2018] of mensural notation, which uses a different symbol alphabet: instead of decomposing the graphical notes into primitives, its symbol classes correspond to the entire note: longa, breve, semibreve, minima, semiminima, etc. Furthermore, Spanish white mensural notation (of which the Capitan dataset comprises) does not allow many of the situations that make CWMN recognition difficult, such as beamed groups and polyphony on one staff. Fig. 4.6 illustrates the advantage of U-Nets on MUSCIMA++<sup>24</sup>

We therefore choose U-Nets for the object detection stage of the full recognition pipeline. Since the objective in this paper is replayability, we restrict detection to only those classes that are relevant for extracting the musical semantics. (As there is a separate model trained for each class, however, this just means reducing the number of models). We employ further two tricks for improving detection performance.

First, we deal with class imbalances. As the training process samples a 256x512-pixel window for each data point [Ronneberger et al., 2015, Dorfer et al., 2017, Hajič jr. et al., 2018a], for relatively rare symbols, often the window contains no pixel of the given target class, and useful signal is drowned out by noise in the initial stages of learning. In order to avoid this effect, if the sampled window does not contain any pixels from the target class, we uniformly sample a different one up to five times. (If after five samples we still found no foreground target pixel, we use the last sampled window.) A second trick for training the detection of rare symbols is letting them share features: with the U-Net model, this only requires adding an output channel to the training data and the model.

A different trick is used to deal with symbols that exhibit complex shapes – again, especially clefs. Instead of training against their true masks, we train against the *convex hulls* of these masks. Since we are at this point mainly trying to detect the presence of the object in a particular location, this approximation does not lower the upper bound on detection performance. At the same time, it simplifies the job of the up-convolution part of the network, as it does not have to “fill in” blanks inside the complex symbols; it decreases the chances that a single detected symbol will form two connected components after thresholding due to false negative pixels in its thin parts, and most importantly, it saves us from dealing with symbols that are legitimately composed from several connected components (such as f-clefs and c-clefs) or written erroneously as disconnected. The convex hull trick is illustrated in Fig. 4.7.

<sup>24</sup>In [Pacha et al., 2018b], both the quantitative and qualitative results are further discussed.



Figure 4.6: Example results in a complex notational situation. (Selected classes. Figure taken from [Pacha et al., 2018b].)

*General object detection experiments with U-Nets, and the rest of the recognition pipeline, are subject of the paper **Towards Full-Pipeline Handwritten OMR with Musical Symbol Detection by U-Nets** [Dorfer et al., 2017] (section 7.3).*

The detection performance, reported simply as the detection f-score<sup>25</sup> for indi-

<sup>25</sup>The F-score is the harmonic mean of recall and precision. This way, it balances the need to avoid

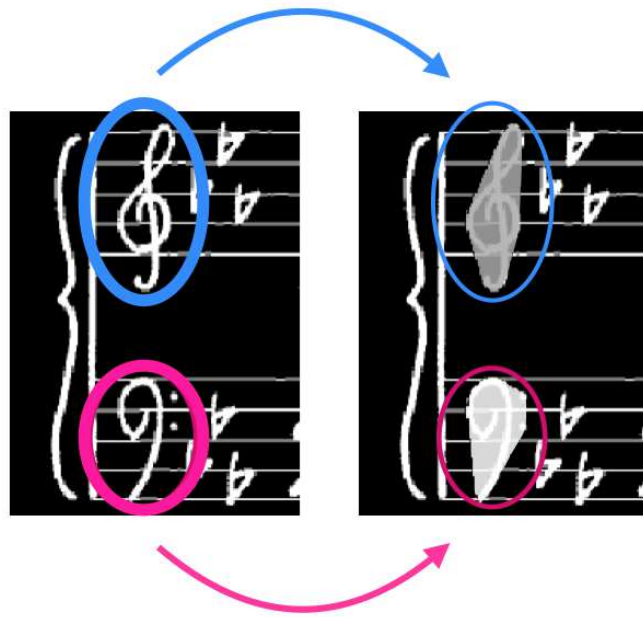


Figure 4.7: Modifying the targets for semantic segmentation training to convex hulls of the objects that we ultimately want to detect. Top: g-clef, bottom: f-clef.

vidual replayability-oriented object classes, is reported in Fig. 4.8. The greatest improvements using the training tricks came for clefs, which were the most problematic symbols for the “vanilla” training. They were also the group of symbols that benefited most from being grouped into a multichannel model.

### 4.3.2 Notation Assembly and Semantics Inference

Next, we build the notation assembly stage and infer musical semantics.

Under the MuNG formalism, notation assembly is the task of inferring the graph edges given the (detected) nodes. The simplest thing one can do is to decompose this task into decision about individual edges (or non-edges): frame the task as binary classification over node pairs, and assume the edges (and non-edges) are independent.

---

both false positives and false negatives, and penalizes systems that err too much to one side: a system with recall 1.0 and precision 0.1 will have an f-score of 0.18, while a system with recall 0.6 and precision 0.5 will have an f-score of 0.55.

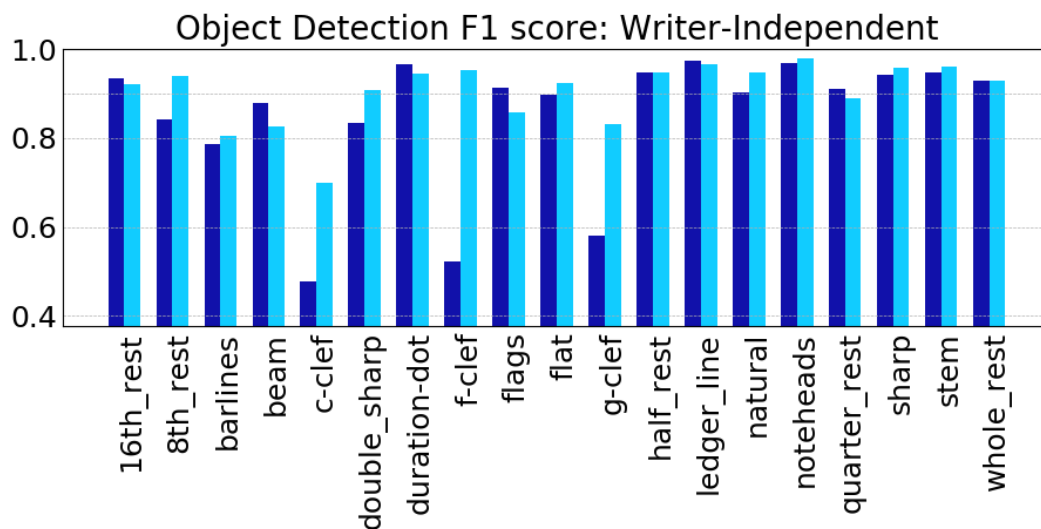


Figure 4.8: Detection f-score for symbols required for replayability: with “vanilla” U-Nets, and with tricks. Note the improvements especially for clefs: these are critical for pitch inference.

*The pairwise independent classification approach to notation assembly is first mentioned in the paper **The MUSCIMA++ Dataset for Handwritten Optical Music Recognition** [Hajič jr. and Pecina, 2017c], reproduced in section 6.2, as a baseline for graph assembly over ground truth objects. The assembly models are applied on top of object detection results in the article **Towards Full-Pipeline Handwritten OMR with Musical Symbol Detection by U-Nets** [Dorfer et al., 2017] (section 7.3).*

The independence assumption is of course an over-simplification: e.g., a notehead may be connected to a beam either above, or below its position, but not both.<sup>26</sup> However, breaking down the problem into independent decisions is a reasonable start – if the straightforward binary classification methods do not perform well, we at least will have a good understanding of what errors they systematically make and what kind of dependencies we should introduce into the model.

What simplifies the situation further is that edges leading from objects to stafflines, staffspaces and their containing staves can be even in manuscripts inferred near perfectly<sup>27</sup> using appropriate heuristics based simply on how the object overlaps with the stafflines and staffspaces. The only “magic number” that must be selected concerns the situation where a notehead has one part above a staffline and another part below the same staffline: if there is a large imbalance between these two parts, expressed as the ratio of the offset of the top of the notehead to the top of the staffline

<sup>26</sup>Unless the notehead in question also has stems in both directions, in the case of voices sharing a note, which is a corner case that breaks a lot of otherwise reasonable assumptions...

<sup>27</sup>Only two noteheads in MUSCIMA++ had their relationship to the staff objects inferred incorrectly, once the heuristics were completed.

vs. the offset of the bottom of the notehead to the bottom of the staffline, it should be considered connected to the corresponding staffspace rather than to the staffline it overlaps. If this ratio is smaller than 0.2 or greater than 0.8, the notehead should be assigned to the staffspace; if the imbalance is not as large, it should be considered to lie on the staffline it overlaps. The fact that the relationships of notation objects to staffs can be inferred so deterministically is one of the few surprisingly *easy* things in OMR.

Given that the average image in MUSCIMA++ contains about 650 notation objects, if we were to consider the quadratic amount of  $\langle \text{from}, \text{to} \rangle$  pairs in an image (recall that the MuNG has *oriented* edges, so the distinction between from and to is necessary), we would have to make over 400 000 decisions for each image. Fortunately, objects that are far from each other are quite certain to not be related. In MUSCIMA++, we found that if we only consider  $\langle \text{from}, \text{to} \rangle$  pairs within a distance of  $10 * \text{staffspace\_height} + \text{staffline\_height}$ , we only discard 52 out of the 82247 related symbol pairs (excluding the relationships to stafflines and staffspaces) [Hajič jr. and Pecina, 2017c]. This reduces the number of  $\langle \text{from}, \text{to} \rangle$  candidate pairs to a linear number, albeit with a significant multiplicative constant (about 8 – 15, based on the density of the handwriting).

For a  $\langle \text{from}, \text{to} \rangle$  candidate object pair: the simplest features we can use are their classes ( $class_f, class_t$ ), and relative position of their bounding boxes: given that the bounding box of the from object is  $B_f = (top_f, left_f, bottom_f, right_f)$ , and the bounding box of the to object is  $B_t = (top_t, left_t, bottom_t, right_t)$ , the offset features are  $(top_t - top_f, left_t - left_f, bottom_t - bottom_f, right_t - right_f)$ . If an edge leads from the from object to the to object, the target is 1, otherwise it is 0. As positive examples, we take all related objects in the training data, as negative examples, we take simply all pairs that are within the threshold distance, but are not related.

We then train a decision tree [Leo Breiman et al., 1984]. Given a sufficient maximum depth for the tree, the model picks up by itself on the constraints imposed by symbol classes (for instance, there can never be an edge leading from a stem to a notehead, only in the opposite direction, accidentals are never associated with rests, etc.). Already this simple model achieves an f-score of 0.92 on edges on the MUSCIMA++ test set [Hajič jr. and Pecina, 2017c, Hajič jr. et al., 2018a] (as there are many more non-edges than edges, reporting overall accuracy would be overly optimistic, and we care only about the positive class anyway).

Some egregious assembly errors are caused by the simplified pairwise model that does not take any other objects than the  $\langle \text{from}, \text{to} \rangle$  pair into account when making a decision. Especially (1) connecting a notehead to ledger lines both above and below the given notehead, and (2) connecting a notehead to beams both above it and below, unless it also has two related stems. However, these two can be relatively easily corrected. While these postprocessing heuristics based on “hard” constraints of music notation syntax did give quick improvement in these two specific cases, attempting other such patches did not improve overall results anymore.



We note that the MuNG formalism has allowed us to reach respectable assembly performance on handwritten music notation with gross oversimplifications of the rules of music notation, thanks to making straightforward machine learning techniques applicable.

A separate chapter are *precedence* edges. We admit that this aspect of the pipeline is somewhat underdeveloped: we simply order simultaneities linked to a staff left to right, and consider noteheads to belong to a simultaneity whenever they have an edge to a shared stem (this is the MuNG-based definition of what a chord is in music notation). This is probably the greatest limitation of our recognition pipeline.

All the results above matter little by themselves; what makes an OMR system interesting is its ability to infer the musical semantics. A key advantage of MuNG<sup>28</sup> is that once notation assembly is done, which happens without ever straying from the graphical layer of music notation into the layer of the semantics, one can infer the musical semantics unambiguously, using the rules for reading music laid out in section 2.2. The mung package implements this semantics inference and subsequent MIDI export.

The detection, assembly and semantics inference steps can also be run interactively from MUSCIMarker (with detection running over an http connection, so that the detection model can be run remotely as a service in case a given machine does not have the necessary computing power).

### 4.3.3 Full Pipeline Results

We evaluate the full pipeline first intrinsically, based on its ability to correctly retrieve the musical semantics. In order to do this, we first have to align the output MIDI with the MIDI corresponding to the ground truth pipeline. (Since not all notes are detected properly, it is not straightforward to directly compare the detected notes with the ground truth notes: especially duration errors propagate by influencing the onsets of all subsequent notes.) We use Dynamic Time Warping (DTW)<sup>29</sup> on sequences simultaneities, using the ratio of pitches shared as the inverse cost function. Within each simultaneity, the notes are aligned from lowest to highest using a second round of DTW. The reason for using DTW to find the alignment between the recognition result and the ground truth is that it naturally finds an optimal alignment that does not violate the precedence relationships in neither the ground truth, nor the OMR output.

Given this alignment, we can compute how well the semantics were recovered. Unfortunately, for durations, the results were less than convincing (an f-score of less than 0.6). **For pitch, the overall f-score was 0.81.** The breakdown of pitch recognition f-score by individual staves in the MUSCIMA++ test set is given in Fig. 4.9.

---

<sup>28</sup>Dare we say, its elegance?

<sup>29</sup>Dynamic Time Warping is a dynamic programming technique that finds the optimal monotonous alignment of two sequences given a cost function that assigns a cost to pairs of the sequence elements.



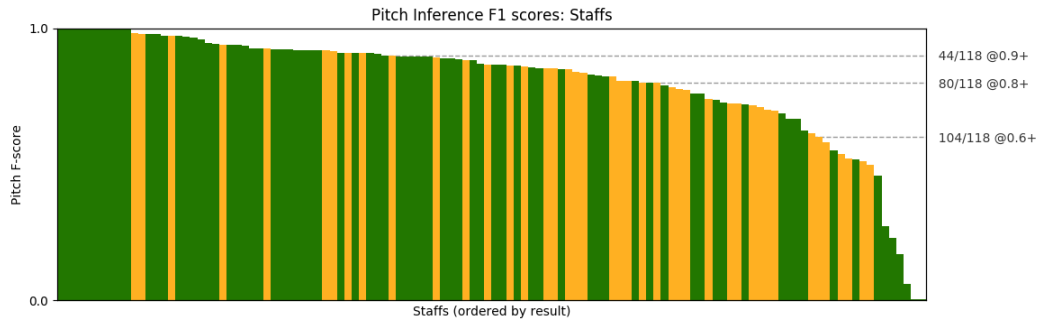


Figure 4.9: The pitch recognition f-score for the MUSCIMA++ writer-independent test set, broken down per individual staff. Monophonic staves in green, homophonic and polyphonic in yellow.

Since duration errors propagate into onset errors, f-score for onsets cannot really be computed; however, it is implicitly encoded in the inferred precedence edges. These are always correct in monophonic and homophonic notation to the extent to which the symbols corresponding to notes or rests are detected.

What can we do with a manuscript recognition system with this performance?

*The application scenario described below is part of the article **How Current Optical Music Recognition Systems Are Becoming Useful for Digital Libraries** [Hajič jr. et al., 2018b] (section 7.5).*

It turns out [Hajič jr. et al., 2018b] that it should be possible to use this system to retrieve musical manuscripts copies. Since transfer learning for object detection is an unresolved issue, we are limited to the CVC-MUSCIMA dataset (of which MUSCIMA++ is a subset). At least for demonstration purposes, we select a subset of 7 of the 20 pages that it as confusing as possible (the music is as similar as possible to each other), in order to not artificially inflate the scores, and run recognition for each of these across all 50 writers, for a total of 350 pages. This is a toy dataset (although for manuscripts, we unfortunately cannot do better right now); any OMR system worth its salt should be able to retrieve duplicate pages perfectly. We use the DTW alignment cost as the similarity function of the retrieval system.

In Table 4.2, we report results for retrieving OMR outputs using queries constructed from OMR outputs, and also *cross-modal* results: querying the database of MIDI files obtained through OMR using ground truth MIDI files and snippets. Aside from page queries, we also attempt to retrieve pages using only music from a single staff, where results are less than perfect (but the task is significantly more difficult); the results there are worse, especially in the cross-modal setting that is sensitive to design limitations of the OMR system (that may “cancel out” when using OMR outputs both as the query and as the database).

The implications of the retrieval experiments are: despite the many limitations of our OMR pipeline, already it is a method for extracting the musical semantics from

	MAP@1	MAP@10	MAP@49
Page queries, OMR2OMR	1.0	1.0	0.998
Page queries, cross-modal	1.0	1.0	0.998
Snippet queries, OMR2OMR	0.928	0.834	0.763
Snippet queries, cross-modal	0.606	0.610	0.577

Table 4.2: Results for page retrieval using page queries and snippet queries under two modalities: using OMR for creating the database and the query (OMR2OMR) or just for the database (cross-modal) and query with ground-truth MIDI. (Table reproduced from [Hajič jr. et al., 2018b].)

handwritten CWMN of arbitrary complexity that has potential real-world applications (identifying copies of manuscripts across archives) – to the extent to which training data will be available.

This concludes the main “thesis story”. Starting from a situation with no resources and viable methods to deal with musical manuscripts, our work builds and publishes a functioning (to the extent described above) OMR system for handwritten music notation of arbitrary complexity. However, we fully expect the performance of the recognition pipeline to be surpassed; the most durable contribution of the thesis is the idea (and implementation) of MuNG as a universal formalism for OMR.

## 4.4 Auxilliary contributions

Outside of the “main story” of the thesis, other aspects of OMR were also explored, most importantly evaluation and contributions to the functioning of the scientific community.

### 4.4.1 Evaluation

As stated in subsection 3.2.2, there is no (publicly available) way of meaningfully and automatically measuring the similarity (or distance) of a pair of music scores. For replayability-oriented applications, this is not necessarily a problem, but in the setting where OMR is supposed to produce a musical score, the need for such a metric is unavoidable [Hajič jr., 2018]. While OMR can move forward to some extent even in the absence of such a measure [Byrd and Simonsen, 2015], it is a major issue [Droettboom and Fujinaga, 2004, Padilla et al., 2014, Baoguang Shi et al., 2015].<sup>30</sup> The need for system-level evaluation at the level of a reconstructed score is most pressing when comparing against commercial systems: these do not output partial results, so evaluating on the level of individual symbols requires an impractically large amount of human effort [Bellini et al., 2007, Sapp, 2013], and is anyway too

<sup>30</sup>Additionally, this problem has come up as important in all the major discussions among current OMR researchers: at the ISMIR 2016 Unconference group, at the GREC 2017 workshop discussion group [Calvo Zaragoza et al., 2018], and at the WoRMS 2019 workshop.

time-consuming for iterative experimentation. While any way of comparing scores would be appreciated, OMR would benefit much more if the metric for comparing music scores could be automated, and, importantly for finally relating the state of the art in OMR research to commercial software, if it also operated on widespread representations of music scores.

To the best of our knowledge, the only effort to this end has already been undertaken by [Szwoch, 2008], who designs a top-down MusicXML comparison function. However, the method is not described in sufficient detail and source code was not made available. Furthermore, an automated metric for OMR evaluation needs itself to be evaluated: does it really rank as better systems that *should* be ranked better? In [Szwoch, 2008], the method was assessed against human judgment, the guidelines for this assesment were not described sufficiently to replicate the experiment.

In [Hajič jr. et al., 2016], we take a different approach. Instead of focusing on a method, we focus on gathering the human judgment resources against which OMR evaluation methods can be assessed. This “evaluating evaluation” approach is loosely modelled on analogous efforts in machine translation (MT) [Callison Burch et al., 2010, Bojar et al., 2011, Matouš Macháček and Ondřej Bojar, 2014], where results of human evaluations are used to validate proposed automated metrics. Good match against human preferences was the argument for adopting the now-widespread but at the same time relatively simple BLEU metric [Papineni et al., 2002].

*The work on evaluating automated OMR evaluation proposals is described in the paper **Further Steps Towards a Standard Testbed for Optical Music Recognition** [Hajič jr. et al., 2016] (section 6.4). A concise analysis that better delimits the needs of OMR evaluation is given in the short paper **A Case for Intrinsic Evaluation of Optical Music Recognition** [Hajič jr., 2018] (section 6.5).*

The data points we collect pairwise preference judgments: the annotators are shown two simulated OMR outputs and the target score, and they are asked to choose which of the OMR outputs they would prefer. The annotation interface is shown in Fig. 4.10. A total of 1500 such annotations was collected (a set of the same 100 examples for each annotator) and released as the omreval corpus.<sup>31</sup>


A proposed OMR evaluation metric can then be assessed according to how well it agrees with the human preferences using three standard measures of agreement: we used Spearman’s  $r$  [Charles Spearman, 1904], Pearson’s  $\rho$  [Pearson, 1896] and Kendall’s  $\tau$  [Kendall, 1938]. Four baseline MusicXML distance metrics are assessed in [Hajič jr. et al., 2016] in this manner: Levenshtein distance [Levenshtein, 1966] on the XML files, Levenshtein distance on the LilyPond<sup>32</sup> imports of the given MusicXML files, Tree Edit Distance [Zhang and Shasha, 1989] on the MusicXML files

<sup>31</sup><https://github.com/ufal/omreval>

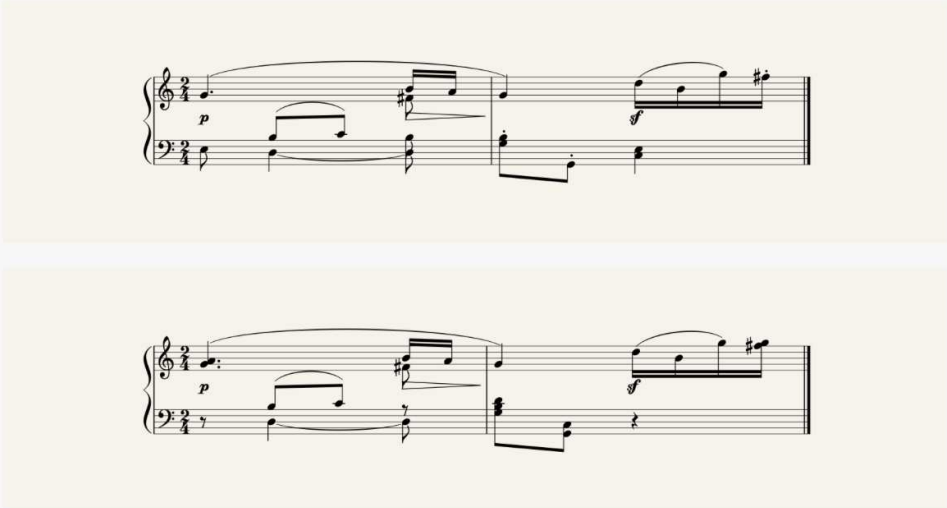
<sup>32</sup>A text-based format for representing music notation for the purposes of engraving, based on LaTeX: <https://lilypond.org>

## Which of these recognition results would you prefer?

**True score:**



**Prediction outputs**



The image displays a musical score in 2/4 time, labeled "True score". It features a piano (p) dynamic and a melody in the right hand with a bass line in the left hand. Below the true score are two "Prediction outputs" shown in orange boxes. The top prediction output is a slightly altered version of the true score, while the bottom prediction output is a more significant deviation, particularly in the bass line.

Figure 4.10: Collecting a data point for the omreval corpus: the annotators are asked to choose which of two simulated OMR outputs (orange, bottom) they would prefer, given the correct score (green, top).

directly, and Tree Edit Distance with the subtree of  $\langle \text{Note} \rangle$  nodes of the MusicXML document transformed into a positional encoding of the represented musical semantics. This last method was found to work best (see Table 4.3), but still significantly underperforms the maximum achievable w.r.t. inter-annotator agreement (see section 6.4 for details). However, the proposed baseline evaluation metrics are not the main contributions of [Hajič jr. et al., 2016]; the main contribution is the omreval corpus and the extensible and robust methodology for collecting it.

### 4.4.2 OMR Scientific Community

In the course of the work on this thesis, and especially as a function of the international collaboration established through this work, the author contributed significantly also to the coalescing of the field of OMR into an actual scientific community, with a shared publication venue, introductory materials for newcomers, centralized resources, and, similarly to the related Digital Libraries for Music community, a place as a part of the broader Music Information Retrieval community. Three elements were critical to building this (sense of) community:

<b>Metric</b>	$r_s$	$\hat{r}_s$	$\rho$	$\hat{\rho}$	$\tau$	$\hat{\tau}$
c14n	0.33	0.41	0.40	0.49	0.25	0.36
TED	0.46	0.58	0.40	0.50	0.35	0.51
TEDn	<b>0.57</b>	<b>0.70</b>	0.40	0.49	<b>0.43</b>	<b>0.63</b>
Ly	0.41	0.51	0.29	0.36	0.30	0.44

Table 4.3: The measures of agreement for some proposed evaluation metrics against the omreval corpus: Spearman’s  $r$ , Pearson’s  $\rho$  and Kendall’s  $\tau$ . Maximum achievable in the individual metrics with respect to the expected inter-annotator agreement indicated in columns with a hat.

- The GREC 2017 discussion group, organized by Alicia Fornés. The thesis author actively participated in this group and co-wrote its report [Calvo Zaragoza et al., 2018]. The outputs of this OMR roundtable established guidance for further community-building activities, namely:
- The 1st International Workshop on Reading Music Systems (WoRMS), which took place as a satellite event of the ISMIR 2019 conference in Paris. The thesis author served as one of the general chairs of the workshop.
- The tutorial “Optical Music Recognition for Dummies” selected for presentation at the ISMIR 2018 conference in Paris, which serves as extensive introductory material for newcomers to OMR. The tutorial is made available online.<sup>33</sup><sup>34</sup>

<sup>33</sup><https://youtube.com/playlist?list=PL1jvwDVNwQke-04UxzlzY4FM33bo1CGS0>

<sup>34</sup>The contribution of the thesis author to the tutorial is about 30 – 35 %.

## 5. Conclusions

Over the course of this thesis, the state of Optical Music Recognition (OMR) has advanced significantly. The state of the field in 2014 was such that both the underlying theoretical work in OMR was lacking, and the field was severely under-resourced, with no open dataset, standards, evaluation methodologies, etc., which one expects in less niche fields, and consequently there were few published works that dealt with manuscript recognition. This thesis has contributed to OMR in three major areas: the theoretical aspects of OMR, open resources for OMR research, and applied machine learning for OMR to build an image-to-MIDI pipeline for handwritten scores of arbitrary complexity.

Theoretical advances have been made first in creating a better definition of OMR and an analysis of the field’s internal structure, and, second, in proposing a formal description of music notation documents as dependency graphs. The Music Notation Graph (MuNG) formalism allows accurate notation assembly in arbitrarily complex handwritten notation by making state-of-the-art machine learning methods applicable. The formalism is conceptually simple, universal, avoids the mixing of music notation and musical semantics, and while the presented object detection methods will most likely be surpassed in the very near future with improvements in applying generic models for OMR, the fact that the MuNG-based assembly performs well, is easy to set up, and has supporting software (esp. MIDI inference), the dependency graph-based methodology is in a good position to become part of the OMR mainstream, especially for manuscript recognition. Additionally, an introduction to OMR in the form of a conference tutorial was made publicly available.

The open resources that have been contributed are the MUSCIMA++ dataset, which is the first extensive OMR dataset with systematic annotations for full-pipeline recognition, the mung software package for manipulating the Music Notation Graph representation including MIDI export, the MUSCIMarker annotation tool, and the omreval corpus for systematic testing of OMR extrinsic evaluation metrics against human preferences. The most important of these is the MUSCIMA++ dataset, which is the first extensive dataset of handwritten music notation annotated with ground truth that allows training and evaluating supervised full-pipeline OMR systems.

Finally, OMR methods presented in this thesis comprise a machine learning-based pipeline that starts with an image and extracts the musical semantics (note pitches, onsets and durations) as a MIDI file. The key symbol detection step is tackled using fully convolutional neural networks (U-Nets), which have outperformed comparable object detection networks. The notation graph formalism then allows formulating notation assembly as a straightforward binary classification task. The resulting pipeline has been evaluated also on a small manuscript retrieval task, and found promising for manuscript copy retrieval. This is the first time a machine learning-

based full pipeline has been developed for handwritten OMR.

Much of the work in this thesis has been achieved through robust international collaboration in the OMR community,<sup>1</sup> which has historically been a rare occurrence. The thesis author was one of the chairs of the 1st International Workshop on Reading Music Systems, an attempt to create an OMR-centered publication venue related to the Music Information Retrieval and Digital Libraries and Musicology communities; already in its first year, despite the tiny size of the field, it has attracted twelve contributions and more than 25 participants.<sup>2</sup> Further community resources were created through the established international collaboration: the tutorial “Optical Music Recognition for Dummies” presented at the ISMIR 2018 conference and made available as a video playlist,<sup>3</sup> a website<sup>4</sup> and repository<sup>5</sup> for keeping track of OMR advances and publicizing them, etc.

What are the challenges that OMR is facing now? What are the current opportunities for research advances?

One major remaining challenge is evaluation: there is still no way to meaningfully measure the similarity between two scores, especially in terms of edit distance. The MuNG formalism offers one possible solution, if an appropriate graph alignment algorithm is developed. Of course, further datasets, especially some that would better correspond to real-world use-cases on top of the existing datasets focused on general challenges of OMR, would be helpful to the field, but datasets are no longer a bottleneck.

In terms of methods, there are two clear major challenges. One is to create sufficiently generic musical object detection methods, so that the amount of task-specific manual annotation necessary to apply the machine learning-based OMR techniques is reduced (and, ideally, eliminated entirely). This should be, to some extent, possible: Common Western Music Notation (CWMN) remains in principle – and in practice for human readers – the same writing system, regardless of whether it is printed or handwritten, whether it is born-digital or written on a parchment. Models that take advantage of its ideal topology may be able to generalize across input conditions; however, these would probably require more complex top-down Bayesian modeling, such as the hierarchical model of [Lake et al., [2013]]. Alternately, training data could perhaps be adapted to fit images from a target archive using style transfer [Leon A. Gatys et al., [2016]], at least with respect to the document quality and imaging process. An interesting modeling challenge specific to the MuNG formalism is to find an objective function that would allow jointly learning object detection and assembly. The second major challenge for OMR methods is to find a model for end-to-end recognition of polyphonic music and the corresponding algorithms for inference.

Overall, we believe this thesis has moved the needle of Optical Music Recognition

---

<sup>1</sup>As evidenced by the wealth of co-authors of the publications that form the backbone of this thesis.

<sup>2</sup><https://sites.google.com/view/worms2018/proceedings>

<sup>3</sup><https://youtube.com/playlist?list=PL1jvwDVNwQke-04UxzIzY4FM33bo1CGS0>

<sup>4</sup><https://omr-research.net/>

<sup>5</sup><https://github.com/OMR-research>



on CWMN manuscripts to the extent that there is a clear path to practical solutions to the problem of computationally reading handwritten music notation. We are looking forward to seeing how others will utilize our contributions to move the field forward.

**Part II**

**Published Works**

Here we present the individual publication that underlie the claims made in this dissertation. The publications are grouped in two sections: one on OMR resources and theory, which are closely interconnected, another on OMR methods. In a short preface to each of the article, we briefly describe the contributions of the articles towards the dissertation, and detail the contributions of the dissertation author to the articles.

## 6. Theory and Resources

We first list articles related to the theoretical improvements (terminology, taxonomy, deeper understanding) of OMR and resources that this thesis contributes to the field.

### 6.1 Understanding Optical Music Recognition

Jorge Calvo-Zaragoza, Jan Hajič jr. and Alexander Pacha. Understanding Optical Music Recognition. *Manuscript under review*.

The article *Understanding Optical Music Recognition* introduces Optical Music Recognition: gives a clear definition and analyzes the structure of the problem, and proposes a taxonomy of OMR inputs and outputs (applications). Note that the paper organizes previous work on OMR by *application* rather than by method, as methods up to 2011 have been thoroughly reviewed in [Rebelo et al., 2012] and later contributions are nearly all based on deep learning and reviewed in the corresponding publications in the methods section of this thesis.

The contribution of the thesis author and the co-authors cannot be described in other terms than equal; the article is a result of year-long weekly discussions. Of the original theoretical contributions in the article, the thesis author formulated the analysis of what OMR is in terms of inverting the process how music notation is created, and elaborated the structure of replayability and reprintability (sections 2, 3, 4).

The article is currently under review in a journal.

# Understanding Optical Music Recognition

JORGE CALVO-ZARAGOZA\*, University of Alicante, Spain

JAN HAJIČ, JR.\*, Charles University, Czech Republic

ALEXANDER PACHA\*, TU Wien, Austria

For over 50 years, researchers have been trying to teach computers to read music notation, referred to as Optical Music Recognition (OMR). However, this field is still difficult to access for new researchers, especially those without a significant musical background: few introductory materials are available, and furthermore the field has struggled with defining itself and building a shared terminology. In this tutorial, we address these shortcomings by (1) providing a robust definition of OMR and its relationship to related fields, (2) analyzing how OMR inverts the music encoding process to recover the musical notation and the musical semantics from documents, (3) proposing a taxonomy of OMR, with most notably a novel taxonomy of applications. Additionally, we discuss how deep learning affects modern OMR research, as opposed to the traditional pipeline. Based on this work, the reader should be able to attain a basic understanding of OMR: its objectives, its inherent structure, its relationship to other fields, the state of the art, and the research opportunities it affords.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Information systems** → Music retrieval; • **Applied computing** → **Document analysis**; **Graphics recognition and interpretation**; *Sound and music computing*; *Digital libraries and archives*.

Additional Key Words and Phrases: Optical Music Recognition, Music Notation, Music Scores

## ACM Reference Format:

Jorge Calvo-Zaragoza, Jan Hajič, jr., and Alexander Pacha. 2019. Understanding Optical Music Recognition. *ACM Comput. Surv.* 1, 1, Article 1 (January 2019), 34 pages. <https://doi.org/0000001.0000001>

## 1 INTRODUCTION

Music notation refers to a group of writing systems with which a wide range of music can be visually encoded so that musicians can later perform it. In this way, it is an essential tool for preserving a musical composition, facilitating permanence of the otherwise ephemeral phenomenon of music. In a broad, intuitive sense, it works in the same way that written text may serve as a precursor for speech. In the same way that Optical Character Recognition (OCR) technology has enabled the automatic processing of written texts, reading music notation also invites automation. In an analogy to OCR, the field of Optical Music Recognition (OMR) covers the automation of this task of “reading” in the context of music. However, while musicians can read and interpret very complex music scores even in real time, there is still no computer system that is capable of doing so with success.

\*Equal contribution

---

Authors' addresses: Jorge Calvo-Zaragoza, University of Alicante, Carretera San Vicente del Raspeig, Alicante, 03690, Spain, [jcalvo@dlsi.ua.es](mailto:jcalvo@dlsi.ua.es); Jan Hajič, jr. Charles University, Prague, Czech Republic, [hajicj@ufal.mff.cuni.cz](mailto:hajicj@ufal.mff.cuni.cz); Alexander Pacha, TU Wien, Institute of Information Systems Engineering, Favoritenstraße 9-11, Vienna, 1040, Austria, [alexander.pacha@tuwien.ac.at](mailto:alexander.pacha@tuwien.ac.at).

---

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2019 Copyright held by the owner/author(s).

0360-0300/2019/1-ART1

<https://doi.org/0000001.0000001>

We argue that besides the technical challenges, one reason for this state of affairs is also that OMR has not defined its goals with sufficient rigor to formulate its motivating applications clearly, in terms of inputs and outputs. Work on OMR is thus fragmented, and it is hard for a would-be researcher, and even harder for external stakeholders such as librarians, musicologists, composers, and musicians, to understand and follow up on the aggregated state of the art. The individual contributions are formulated with relatively little regard to each other, although less than 500 works on OMR have been published to date. This makes it hard to combine the numerous contributions and use previous work from other researchers, leading to frequent “reinventions of the wheel.” The field, therefore, has been relatively opaque for newcomers, despite its clear, intuitive appeal.

One reason for the unsatisfactory state of affairs was a lack of practical OMR solutions: when one is hard-pressed to solve basic subproblems like staff detection or symbol classification, it seems far-fetched to define applications and chain subsystems. However, some of these traditional OMR sub-steps, which do have a clear definition and evaluation methodologies, have recently seen great progress, moving from the category of “hard” problems to “close to solved,” or at least clearly solvable [70, 118]. Therefore, the breadth of OMR applications that have long populated merely the introductory sections of articles now comes within practical reach. As the field garners more interest within the document recognition and music information retrieval communities [1, 11, 34, 50, 78, 83, 92, 114, 135], we see further need to clarify how OMR talks about itself.

The primary contribution of this paper is to clearly define what OMR is, what problems it seeks to solve and why. Readers should be able to fully understand what OMR is, even without prior knowledge of music notation. OMR is, unfortunately, a somewhat opaque field due to the fusion of the music-centric and document-centric perspectives. Even for researchers in the field, it is difficult to clearly relate their work to the field, as illustrated in Section 2.

Many authors think of OMR also notoriously difficult to evaluate [84]. However, we show that this clarity also disentangles OMR tasks which are genuinely hard to evaluate, such as full re-typesetting of the score, from those where established methodologies can be applied straightforwardly, such as searching scenarios.

Furthermore, the separation between music notation as a visual language and music as the information it encodes is sometimes not made clear, which leads to a confusing terminology. The way we formulate OMR should provide a framework of thought in which this distinction becomes obvious.

In order to be a proper tutorial on OMR, this paper addresses certain shortcomings in the current literature, specifically by providing:

- A robust definition of what OMR is, and a thorough analysis of its inherent structure;
- Terminological clarifications that should make the field more accessible and easier to survey;
- A review of OMR uses and applications; well-defined in terms of inputs and outputs, and—as much as possible—recommended evaluation methodologies;
- A brief discussion of how OMR was traditionally approached and how modern machine learning techniques (namely deep learning) affects current and future research;
- As supplementary material, an extensive, extensible, accessible and up-to-date bibliography of OMR (see [Appendix A: OMR Bibliography](#)).<sup>1</sup>

The novelty of this paper thus lies in collecting and systematizing the fragments found in the existing literature, all in order to make OMR more approachable, easier to collaborate on, and—hopefully—progress faster.

---

<sup>1</sup><https://github.com/OMR-Research/omr-research.github.io>

## 2 WHAT IS OPTICAL MUSIC RECOGNITION?

So far, the literature on OMR does not really share a common definition of what OMR is. Most authors agree on some intuitive understanding, which can be sketched out as “computers reading music.” But until now, no rigorous analysis of this question has been carried out, as most of the literature on the field focuses on providing solutions—or, more accurately, solutions to certain subproblems—that are usually justified by a certain envisioned application or by referencing a review paper that elaborates on common motivations, with [132] being the most prominent one. However, even these review papers [7, 22, 111, 132] focus almost exclusively on technical OMR solutions and avoid elaborating the scope of the research.

A critical review of the scientific literature reveals a wide variety of definitions for OMR (see [Appendix B: List of OMR definitions and descriptions from published works](#)) with two extremes: On one end, the proposed definitions are clearly motivated by the (sub)problem which the authors sought to solve (e.g., “transforming images of music scores into MIDI files”) which leads to a definition that is too narrow and does not to capture the full spectrum of OMR. On the other end, there are some definitions that are so generic that they fail to outline what OMR actually is and what it tries to achieve. An obvious example would be to define OMR as “OCR for music.” This definition is overly vague, and the authors are—as likewise in many other papers—particularly unspecific when it comes to clarifying what it actually includes and what is not included. We have observed that the problem statements and definitions in these papers are commonly adapted to fit the provided solution or to demonstrate the relevance to a particular target audience, e.g., computer vision, music information retrieval, document analysis, digital humanities, or artificial intelligence.

While people rely on their intuition to compensate for this lack of accuracy, we would rather prefer to put an umbrella over OMR and name its essence by proposing the following definition.

**Definition 1.** Optical Music Recognition is a field of research that investigates how to computationally read music notation in documents.

The first claim of this definition is that OMR is a “*research field*.” In the published literature, many authors refer to OMR as “task” or “process,” which is insufficient, as OMR cannot be properly formalized in terms of unique inputs and outputs (as discussed in Section 6). OMR must, therefore, be considered something bigger, like the embracing research field, which investigates how to provide a computer with the ability to read music notation. Within this research field, several tasks can be formulated with specific, unambiguous input/output pairs.

The term “*computationally*” distinguishes OMR from the musicological and paleographic studies of how to decode a particular notation system. It also excludes studying how humans read music. OMR does not study the music notation systems themselves—rather, it builds upon this knowledge, with the goal that a computer should be able to read the music notation as well.

The last part of the definition “*reading music notation in documents*” tries to define OMR in a concise, clear, specific, and inclusive way. To fully understand this part of the definition, the next section clarifies what kind of information is captured in a music notation document and outlines the process by which it gets generated. The subsequent section then elaborates on how OMR attempts to invert this process to read and recover the encoded information.

It should be noted, that the output of OMR is omitted intentionally from its definition, as different tasks require different outputs (see Section 6) and specifying any particular output representation would make the definition unnecessarily restrictive.

To conclude this section, Fig. 1 illustrates how various definitions of OMR in the literature relate to our proposed definition and are captured by it. A full list of the formulations that have appeared in OMR papers so far can be found in [Appendix B: List of OMR definitions and descriptions from published works](#).



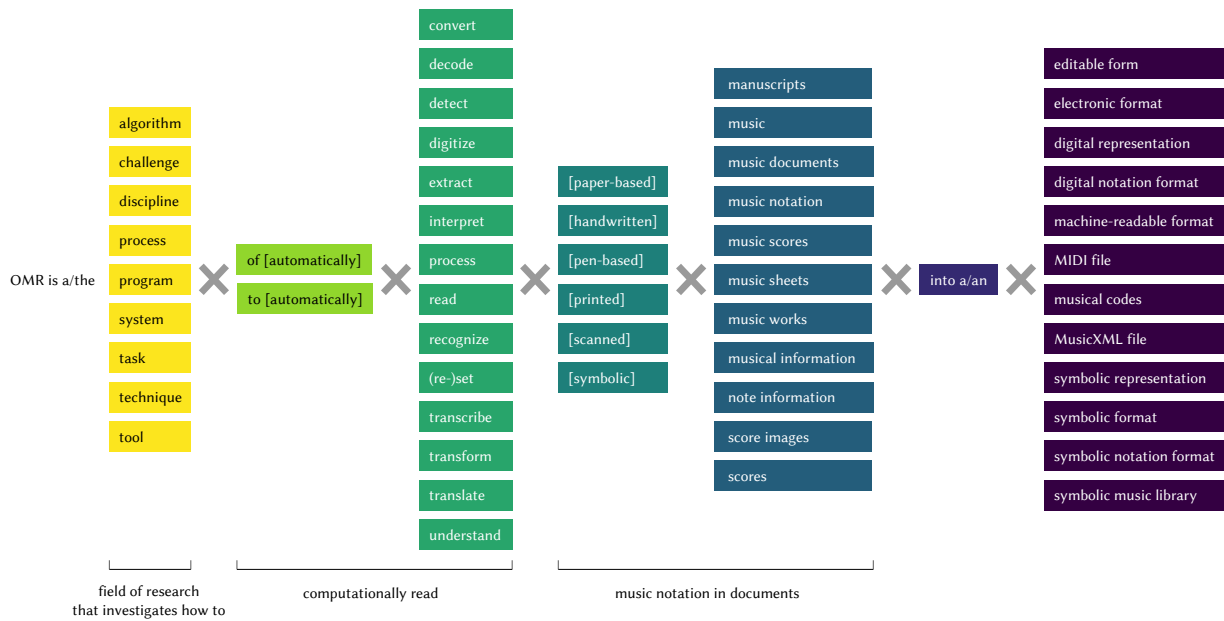


Fig. 1. How OMR tends to be defined or described and how our proposed definition relates to them. For example: “OMR is the challenge of (automatically) converting (handwritten) scores into a digital representation.”

### 3 FROM “MUSIC” TO A DOCUMENT

Music can be conceptualized as a structure of *notes in time*. This is not necessarily the only way to conceptualize music,<sup>2</sup> but it is the only one that has a consistent, broadly accepted visual language used to transmit it in writing, so it is the conceptualization we consider for the purposes of OMR. A note is a musical object that is defined by four parameters: *pitch*, *duration*, *loudness*, and *timbre*. Additionally, it has an *onset*: a placement onto the axis of time, which in music does not mean wall-clock time, but is measured in relative units called beats.<sup>3</sup> Periods of musical time during which no note is supposed to be played are marked by rests, which only have an onset and a duration. Notes and rests are grouped hierarchically into phrases, voices, and other musical units that can have logical relationships to one another. This structure is a vital part of music—it is essential to work it out for making a composition comprehensible.

In order to record this “conceptualization of music” visually, for it to be performed over and over in (roughly) the same way, at least at the relatively coarse level of notes, multiple music notation systems have evolved. A music notation system is a visual language that encodes music into a graphical form and enriches it with information on *how to perform* it (e.g., bowing marks, fingerings or articulations).<sup>4</sup> To do that, it defines a set of symbols as its alphabet and specific rules for how to position these symbols to capture a musical idea. Note that all music notation systems entail a certain loss of information as they are designed to preserve the most relevant properties

<sup>2</sup>As evidenced by either very early music (plainchant) or some later twentieth century compositional styles (mostly spectralism).

<sup>3</sup>Musical time is projected onto wall-clock time with an underlying tempo, which can further be stretched and compressed by the performer. Strictly speaking, the notion of beats might not be entirely applicable to some very early music and some contemporary music, where the rhythmic pulse is not clearly defined. However, the notation used to express such music usually does have beats.

<sup>4</sup>Feist [57] refers to notation whimsically as a “haphazard Frankenstein soup of tangentially related alphabets and hieroglyphics via which music is occasionally discussed amongst its wonkier creators.”

of the composition very accurately, especially the pitches, durations, and onsets of notes, while under-specifying or even intentionally omitting other aspects. Tempo could be one of these aspects, where the composer might have expressed precise metronomic indication, given a verbal hint, or stated nothing at all. It is therefore considered the responsibility of the performer to fill those gaps appropriately. We consider this as a natural boundary of OMR: it ends where musicians start to disagree over the same piece of music.

Arguably the most frequently used notation system is *Common Western Music Notation* (CWMN, also known as modern staff notation), which has developed during the seventeenth century from its mensural notation predecessors and stabilized at the beginning of the nineteenth century. There have been attempts to supersede it in the avant-garde and postmodern movements, but so far, these have not produced workable alternatives. Apart from CWMN, there exist a wealth of modern tablature scores for guitar, used i.a., to write down popular music, as well as a significant body of historical musical manuscripts that are using earlier notation systems (e.g., mensural notations, quadratic notation for plainchant, early organum, or a wealth of tablature notations for lutes).

Once a music notation system is selected for writing down a piece of music, it is still a challenging task to engrave<sup>5</sup> the music because a single set of notes can be expressed in many ways. For example, one must make sure that the stem directions mark voices consistently and appropriate clefs are used, in order to make the music as readable as possible [57, 79, 89, 143]. These decisions not only affect the visual appearance but also help to preserve the logical structure (see Fig. 2). Afterwards, it can be embodied in a document, whether physically or digitally.

To summarize, music can be formalized as a structured assembly of notes, enriched through additional instructions for the performer, that are encoded visually using a music notational language and embodied in a medium such as paper (see Fig. 3). Once this embodiment is digitized, OMR can be understood in terms of inverting this process.

#### 4 INVERTING THE MUSIC ENCODING PROCESS

OMR starts after a musical composition has been expressed visually with music notation in a document.<sup>6</sup> The music notation document serves as a medium, designed to encode and transmit a musical idea from the composer to the performer, enabling the recovery and interpretation of that envisioned music by reading through it. The performer would:

- (1) *Read the visual signal* to determine what symbols are present and what is their configuration,
- (2) Use this information to *parse and decode the notes and their accompanying instructions* (e.g., indications of which technique to use), and
- (3) Apply musical intuition, prior knowledge, and taste to *interpret the music* and fill in the remaining parameters which music notation did not capture.

---

<sup>5</sup>Normally, music engraving is defined as the process of drawing or typesetting music notation with a high quality for mechanical reproduction. However, we use the term to refer to “planning the page”: selecting music notation elements and planning their layout to most appropriately capture the music, before it is physically (or digitally) written on the page. This is a loose analogy to the actual engraving process, where the publisher would carefully prepare the printing plates from soft metal, and use them to produce many copies of the music; in our case, this “printing process” might not be very accurate, e.g., in manuscripts. The engraving process involves complex decisions [24] that can affect only a local area, like spacings between objects but can also have global effects, like where to insert a page break to make it convenient for the musician to turn the page.

<sup>6</sup>While OMR mainly works with a complete image or document, it is also possible to perform online OMR with the temporal signal as it is being generated, e.g., by capturing the stylus input on an electronic tablet device, which also results in a document.

M. M. ♩ = 108

(a)

(b)

Fig. 2. Excerpt of Robert Schumann’s “Von fremden Ländern und Menschen” (Engl. “Of foreign countries and people”), Op. 15 for piano. Properly engraved (a), it has two staves for the left and the right hand with three visible voices, a key signature and phrase markings to assist the musician. In a poor engraving of the same music (b), that logical structure is lost, and it becomes painfully hard to read and comprehend the music, although these two versions contain the same notes.

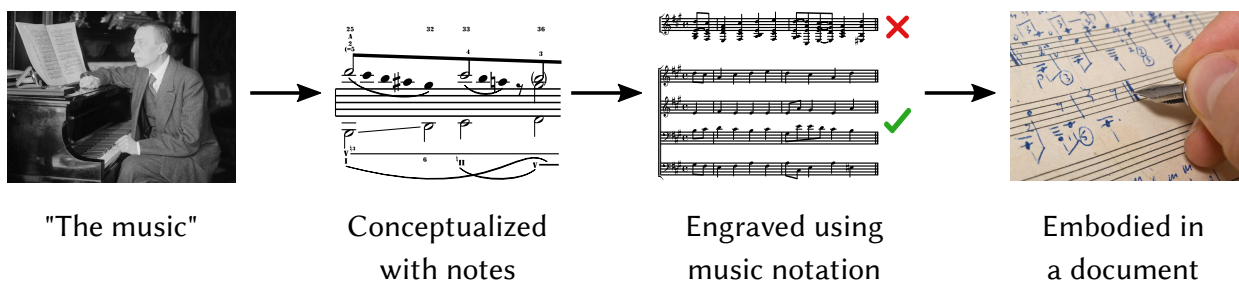


Fig. 3. How music is typically expressed and embodied (written down).

Note that the third step is clearly outside of OMR since it needs to deal with information that is not written into the music document—and where human performers start to disagree, although they are reading the very same piece of music [98].<sup>7</sup>

Coming back to our definition of OMR, based on the stages of the writing/reading process we outlined above, there are two fundamental ways to interpret the term “read” in *reading music notation* as illustrated in Fig. 4. We may wish to:

- A *Recover music notation* and information from the engraving process, i.e., what elements were selected to express the given piece of music and how were they laid out? This corresponds to stage (1) and does not necessarily require specific musical knowledge, but it does require an output representation that is capable of storing music notation, e.g., MusicXML or MEI, which can be quite complex.

<sup>7</sup>Analogously, speech synthesis is not considered a part of optical character recognition. However, there exists expressive performance rendering software that attempts to simulate more authentic playback, addressing step (3) in our analysis. More information can be found in [36].

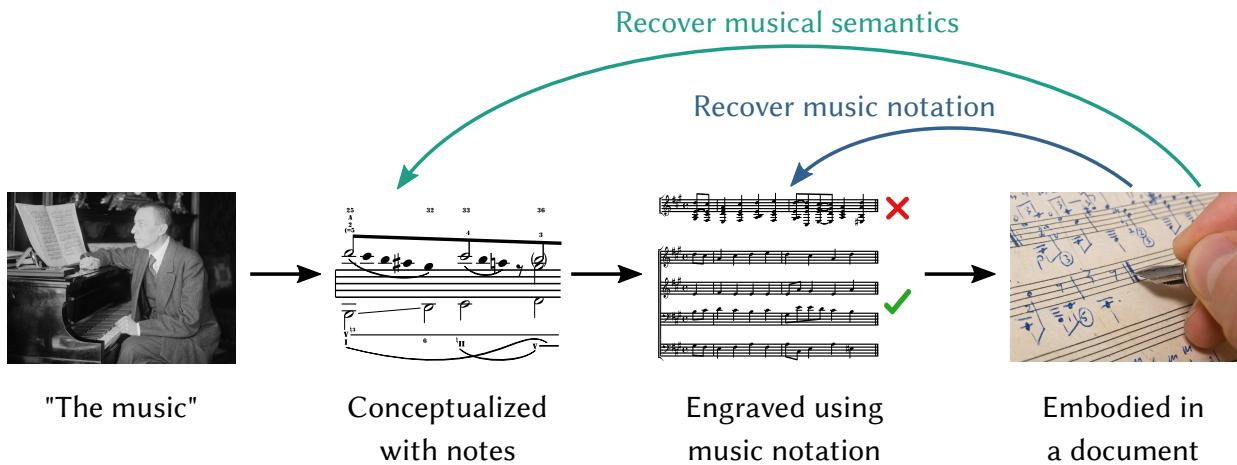


Fig. 4. How “reading” music can be interpreted as the operations of inverting the encoding process.

B *Recover musical semantics*, which we define as the notes, represented by their pitches, velocities, onsets, and durations. This corresponds to stage (2) in the analysis above—we use the term “semantics” to refer only to the information that can be unambiguously inferred from the music notation document. In practical terms, MIDI would be an appropriate output representation for this goal.

This is a fundamental distinction that dictates further system choices, as we discuss in the next sections. Note that counter-intuitively, going backwards through this process just one step (A - recover music notation) might be in fact more difficult than going back two steps (B - recover musical semantics) directly. This is because music notation contains a logical structure and more information than simply the notes. Skipping the explicit description of music notation allows bypassing this complexity.

There is, of course, a close relationship between recovering music notation and musical semantics. A single system may even attempt to solve both at the same time because once the full score with all its notational details is recovered, the musical semantics can be inferred unambiguously. Keep in mind, that the other direction does not necessarily work: if only the musical semantics are restored from a document without the engraving information that describes how the notes were arranged, those notes may still be typeset using meaningful engraving defaults, but the result is probably much harder to comprehend (see Fig. 2b for such an example).

#### 4.1 Alternative Names

*Optical Music Recognition* is a well-established term, and we do not seek to establish a new one. We just notice a lack of precision in its definition. Therefore, it is no wonder that people have been interpreting it in many different ways to the extent, that even the optical detection of lip motion for identifying the musical genre of a singer [53] has been called OMR. Alternative names that might not exhibit this vagueness are Optical Music Notation Recognition, Optical Score Recognition<sup>8</sup>, or Optical Music Score Recognition. While the prefix “Optical” is not compulsory, it could still prove beneficial in highlighting the visual characteristics and help distinguish it from techniques that work on audio recordings.

<sup>8</sup>which is similar to the German equivalent “Optische Notenerkennung”

## 5 RELATION TO OTHER FIELDS

Now that we have thoroughly described what *Optical Music Recognition* is, we briefly set it in context of other disciplines, both scientific and general fields of human endeavors.

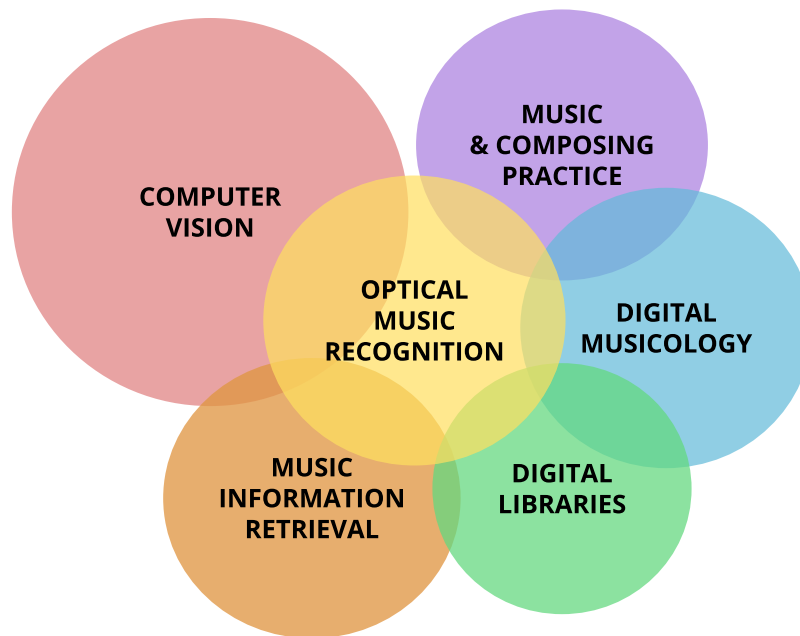


Fig. 5. Optical Music Recognition with its most important related fields, methods, and applications.

Figure 5 lays out the various key areas that are relevant for OMR, both as its tools and the “consumers” of its outputs. From a technical point of view, OMR can be considered a subfield of computer vision and document analysis, with deep learning acting as a catalyst that opens up promising novel approaches. Within the context of Music Information Retrieval (MIR), OMR should enable the application of MIR algorithms that rely on symbolic data and audio inputs (through rendering the recognized scores). It furthermore can enrich digital music score libraries and make them much more searchable and accessible, which broadens the scope of digital musicology to compositions for which we only have the written score (which is probably the majority of Western musical heritage). Finally, OMR has practical implications for composers, conductors, and the performers themselves, as it cuts down the costs of digitizing scores, and therefore bring the benefits of digital formats to their everyday practice.

### 5.1 Optical Music Recognition vs. Text Recognition

One must also address the obvious question: why should OMR be singled out besides Optical Character Recognition (OCR) and Handwritten Text Recognition (HTR), given that they are tightly linked [18], and OMR has been called “OCR for music” frequently [25, 26, 68, 80, 93, 94, 109, 128, 129, 147]?<sup>9</sup> What is the justification of talking specifically about music notation and what differentiates it from other graphics recognition challenges? What are the special considerations in OMR that one does not encounter in other writing systems?

A part of the justification lies in the properties of music notation as a *featural* writing system. While its alphabet consists of well-defined primitives (e.g., stems, noteheads, or flags) that have a clear interpretation, it is only in their configuration—how they are placed and arranged on the

<sup>9</sup>Even the English Wikipedia article on OMR has been calling it “Music OCR” for over 13 years.



Fig. 6. How the translation of the graphical concept of a note into a pitch is affected by the clef and accidentals. The effective pitch is written above each note. Accidentals immediately before a note propagate to other notes within the same measure, but not to the next measure. Accidentals at the beginning of a measure indicate a new key signature that affects all subsequent notes.

staves, and with respect to each other—that specifies what notes should be played. The properties of music notation that make it a challenge for computational reading have been discussed exhaustively by Byrd and Simonsen [29]; we hypothesize that these difficulties are ultimately caused by this featural nature of music notation.

Another major reason for considering the field of OMR distinct from text recognition is the application domain itself—music. When processing a document of music notation, there is a natural requirement to recover its musical semantics (see Section 4, setting B) as well, as opposed to text recognition, which typically does not have to go beyond recognizing letters or words and ordering them correctly. There is no proper equivalent of this interpretation step in text recognition since there is no definite answer to *how a symbol configuration (=words) should be further interpreted*; therefore, one generally leaves interpretation to humans or to other well-defined tasks from the Natural Language Processing field. However, given that music is overwhelmingly often conceptualized as notes, and notes are well-defined objects that can be inferred from the score, OMR is, not unreasonably, asked to produce this additional level of outputs that text recognition does not. Perhaps the simplest example to illustrate this difference is given by the concept of the pitch of the notes (see Fig. 6). While graphically a note lies on a specific vertical position of the staff, other objects, such as the clefs and accidentals determine its musical pitch. It is therefore insufficient for the OMR to provide just the results in terms of positions, but it also has to take the context into account, in order to convert positions (graphical concept) into pitches (musical concept). In this regard, OMR is more ambitious than text recognition, since there is an additional interpretation step specifically for music that has no good analogy in other natural languages.

The character set poses another significant challenge, compared to text recognition. Although writing systems like Chinese have extraordinarily complex character sets, the set of primitives for OMR spans a much greater range of sizes, ranging from small elements like a *dot* to big elements spanning an entire page like the *brace*. Many of the primitives may appear at various scales and rotations like *beams* or have a nearly unrestricted appearance like *slurs* that are only defined as more-or-less smooth curves that may be interrupted anywhere. Finally, in contrast to text recognition, music notation involves ubiquitous two-dimensional spatial relationships, which are salient for the symbols' interpretation. Some of these properties are illustrated in Fig. 7.

Furthermore, Byrd and Simonsen [29] argue that because of the vague limits of what one may want to express using music notation, its syntactic rules can be expected to be bent accordingly; this happens to such an extent that Homenda et al. [90] argued that there is no universal definition of music notation at all. Figure 7 actually contains an instance of such rule-breaking: while one would expect all notes in one chord to share the same duration, the chord on the bottom left contains a mix of white and black noteheads, corresponding to half- and quarter-notes. At the same time, however, the musical intent is yet another: the two quarter-notes in the middle of the chord are actually played as eighth notes, to add to the rich sonority of the fortissimo chord on the first



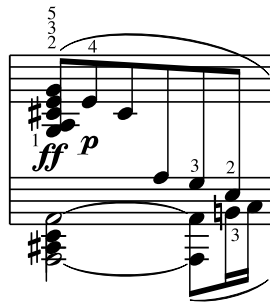


Fig. 7. This excerpt by Ludwig van Beethoven, Piano Sonata op. 2 no. 2, Largo appassionato, m. 31 illustrates some properties of the music notation that distinguish it from other types of writing systems: a wide range of primitive sizes, the same primitives appearing at different scales and rotations, and the ubiquitous two-dimensional spatial relationships.

beat.<sup>10</sup> We believe this example succinctly illustrates the intricacies of the relationship between musical comprehension and music notation. This last difference between a written quarter and interpreted eighth note is, however, beyond what one may expect OMR to do, but it serves as further evidence that the domain of music presents its own difficulties, compared to the domains where text recognition normally operates.

## 5.2 Optical Music Recognition vs. Other Graphics Recognition Challenges

Apart from text, documents can contain a wide range of other graphical information, such as engineering drawings, floor plans, mathematical expressions, comics, maps, patents, diagrams, charts or tables [44, 58]. Recognizing any of these comes with its own set of challenges, e.g., comics combine text and other visual information in order to narrate a story, which makes recovering the correct reading order a non-trivial endeavor. Similarly, the arrangement of symbols in engineering drawing and floor plans can be very complex with rather arbitrary shapes. Even tasks that are seemingly easy, such as the recognition of tables, must not be underestimated and are still subject to ongoing research [131, 144]. The hardest aspects of OMR are much closer to these challenges than to text recognition: the ubiquitous two-dimensionality, long-distance spatial relationships, and the permissive way of how individual elements can be arranged and appear at different scales and rotations.

One thing that makes CWMN more complex than many graphics recognition challenges like mathematical formulae recognition is the complex typographical alignment of objects [7, 29], that is dictated by the content, e.g., each space between multiple notes of the same length should be equal. This complexity is often driven by interactions between individual objects that force other elements to move around, breaking the principal horizontal alignment of simultaneous events (see Fig. 8, 9 and 10).

Apart from the typographical challenges, OMR also has an extremely complex semantic, with many implicit rules. To handle this complexity, researchers have started a long time ago to leverage the rules that govern music notation and formulate them into grammars [4, 123]. For instance, the fact that the note durations (in each notated voice) have to sum up to the length of a measure has been integrated into OMR as a post-processing step [120]. Fujinaga [67] even states that music notation can be recognized by an LL(k) grammar. Nevertheless, the following citation from Blostein and Baird [22] (p.425) is still mostly true:

<sup>10</sup>This effect would be especially prominent on the Hammerklavier instruments prevalent around the time Beethoven was composing this sonata.



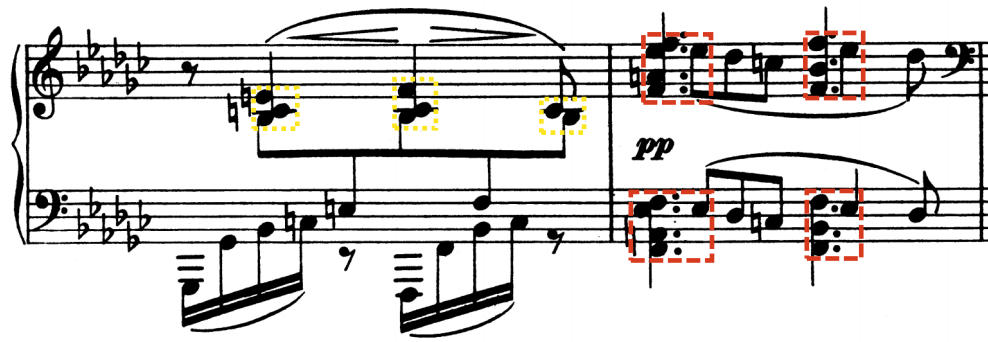


Fig. 8. Brahms Intermezzo, Op. 117 no. 1. Adjacent notes of the chords in the first bar in the top staff are shifted to the right to avoid overlappings (yellow dotted boxes). The moving eighths in the second bar are forced even further to the right, although being played simultaneously with the chord (red dashed boxes).



Fig. 9. Sample from the CVC-MUSCIMA dataset [60] with the same bar transcribed by two different writers. The first three notes and the second three notes form a chord and should be played simultaneously (see right figure) but is sometimes horizontally spelled out (see left figure) left is sometimes used in violin scores.



Fig. 10. Sample from the Songbook of Romeo & Julia by Gerard Presgurvic [124] with uneven spacing between multiple sixteenth notes of the same length in the middle voice to align the notes with the lyrics.

“Various methods have been suggested for extending grammatical methods which were developed for one-dimensional languages. While many authors suggest using grammars for music notation, their ideas are only illustrated by small grammars that capture a tiny subset of music notation.” [22] (p.425; sec. 7 - Syntactic Methods).

There has been progress on enlarging the subset of music notation captured by these grammars, most notably in the DMOS system [49], but there are still no tractable 2-D parsing algorithms that are powerful enough for recognizing music notation without relying on fragile segmentation heuristics. It is not clear whether current parsers used to recognize mathematical expressions [3]

are applicable to music notation or simply have not been applied yet—at least we are not aware of any such works.

## 6 A TAXONOMY OF OMR

Now that we have progressed in our effort to define *Optical Music Recognition*, we can turn our attention to systematizing the field with respect to motivating applications, subtasks, and their interfaces. We reiterate that our objective is not to review the methods by which others have attempted to reach the goals of their OMR work; rather, we are proposing a taxonomy of the field's goals themselves. Our motivation is to find natural groups of OMR applications and tasks for which we can expect, among other things, shared evaluation protocols. The need for such systematization has long been felt [23, 30], but subsequent reviews [111, 132] have focused almost entirely on technical solutions.

### 6.1 OMR Inputs

The taxonomy of *inputs* of OMR systems is generally established. The first fundamental difference can be drawn between *offline* and *online*<sup>11</sup> OMR: offline OMR operates on a static image, while online OMR operates on a time series of user-interactions, typically pen positions that were captured from a touch interface [31, 72, 73, 150]. Online OMR is generally considered easier since the decomposition into strokes provides a high-quality over-segmentation essentially for free. Offline OMR can be further subdivided by the engraving mechanism that has been used, which can be either *typeset* by a machine, often inaccurately referred to as *printed*<sup>12</sup>, or *handwritten* by a human, with an intermediate, yet common scenario of handwritten notation on pre-printed staff paper.

Importantly, music can be written down in many different notation systems that can be seen as different languages to express musical concepts (see Fig. 11). CWMN is probably the most prominent one. Before CWMN was established, other notations such as mensural or neumes preceded it, so we refer to them as *early notations*. Although this may seem like a tangential issue, the recognition of manuscripts in ancient notations has motivated a large number of works in OMR that facilitate the preservation and analysis of the cultural heritage, as well as enabling digital musicological research of early music at scale [50, 51, 69, 158]. Another category of notations that are still being actively used today are *instrument-specific notations*, such as tablature for string instruments or percussion notation. The final category captures all *other notations* including, e.g., modern graphic notation, braille music or numbered notation that are only rarely used and for which the existing body of music is much smaller than for the other notations.

To get an idea of how versatile music can be expressed visually, the Standard Music Font Layout [148] currently lists over 2440 recommended characters, plus several hundred optional glyphs.

Byrd and Simonsen [29] further characterize OMR inputs by the *complexity* of the notated music itself, ranging from simple monophonic music to “pianoform.” They use both the presence of multiple staves as well as the number of notated voices inside a single staff as a dimension of notational complexity. In contrast, we do not see the number of staves as a driver of complexity since a page typically contains many staves and a decision on how to group them into systems has to be made anyway. Additionally, we explicitly add a category for *homophonic* music that only has a single logical voice, even though that voice may contain chords with multiple notes being played simultaneously. The reason for singling out homophonic music is that inferring onsets becomes

<sup>11</sup>Although it might sound ambiguous, the term online recognition has been used systematically in the handwritten recognition community. Sometimes, this scenario is also referred to as pen-based recognition.

<sup>12</sup>Handwritten manuscripts can also be printed out, if they were scanned previously, therefore we prefer the word typeset.

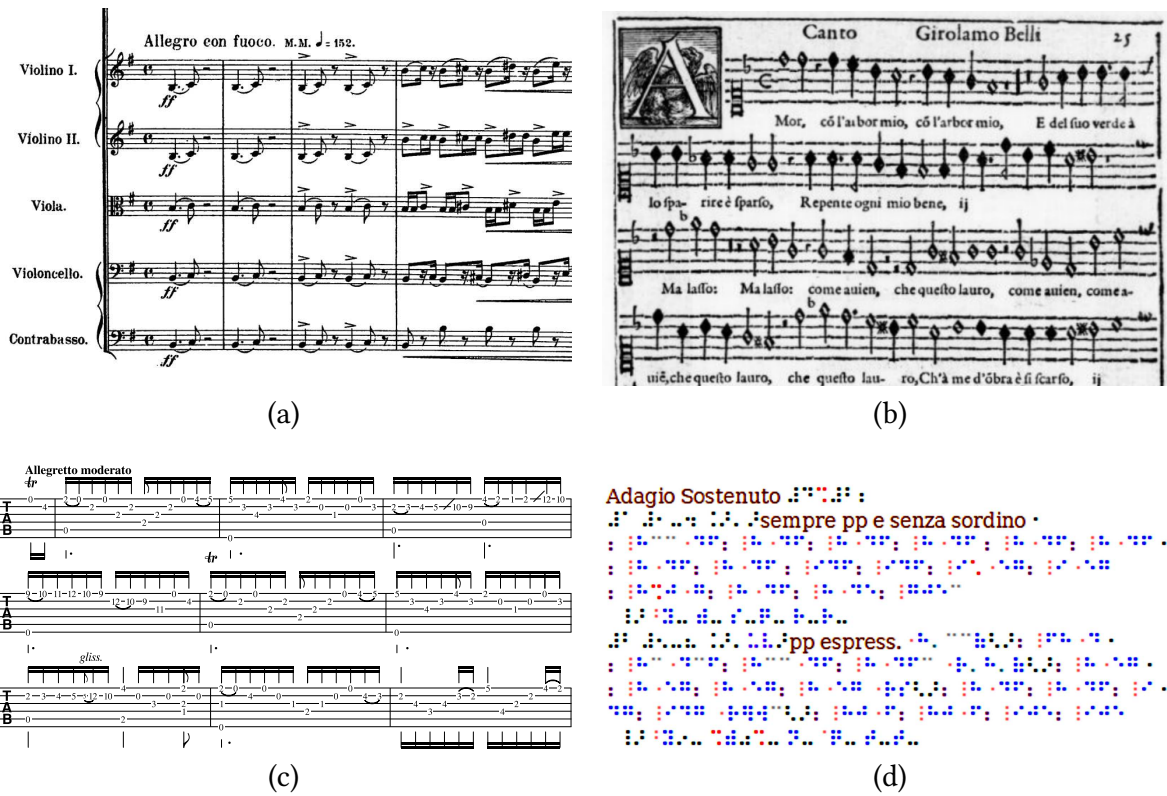


Fig. 11. Examples of scores written in various notations: (a) Common Western Music Notation (Dvorak Symphony No.9, IV), (b) White Mensural Notation (Belli [121]), (c) Tablature (Regondi, Etude No.10) and (d) Braille (Beethoven, Sonata No.14 Op.27 No.2).

trivial once notes are grouped into chords, as opposed to polyphonic music with multiple logical voices: one can simply read them left-to-right without having to do a voice assignment.

Therefore, we propose the following four categories (see Fig. 12):

- (a) *Monophonic*: only one note (per staff) is played at a time.
- (b) *Homophonic*: multiple notes can occur at the same time to build up a chord, but only as a single voice.
- (c) *Polyphonic*: multiple voices can appear in a single staff.
- (d) *Pianoform*: scores with multiple staves and multiple voices that exhibit significant structural interactions. They can be much more complex than polyphonic scores and cannot be disassembled into a series of monophonic scores, such as in polyphonic renaissance vocal part books. This term was coined by Byrd and Simonsen [29].

This complexity of the encoded music has significant implications on the model design since the various levels translate into different sets of constraints on the output. It cannot simply be adjusted or simulated like the visual complexity by applying an image operation on a perfect image [95], because it represents an intrinsic property of the music.

Finally, as with other digital document processing, OMR inputs can be classified according to their image quality which is determined by two independent factors: the underlying *document quality*, and the *digital imaging acquisition* mode. The underlying document quality is a continuum on a scale from perfect or nearly flawless (e.g., if the document was born-digital and printed) to heavily degraded or defaced documents (e.g., ancient manuscripts that deteriorated over time and exhibit faded ink, ink blots, stains, or bleedthrough) [29]. The image acquisition mode is also a continuum

(a) Monophonic

(b) Homophonic

(c) Polyphonic

(d) Pianoform

Fig. 12. Examples of the four categories of music notation complexity.

that can reach from born-digital images, over scans of varying quality to low-quality, distorted photos that originate from camera-based scenarios with handheld cameras, such as smartphones [2, 160].

## 6.2 OMR Outputs

The taxonomy of OMR *outputs*, on the other hand, has not been treated as systematically in the OMR literature. Lists of potential or hypothetical applications are typically given in introductory sections [22, 38, 67, 111]. While this may not seem like a serious issue, it makes it hard to categorize different works and compare their results with each other, because one often ends up comparing apples to oranges [7].

The need for a more principled treatment is probably best illustrated by the unsatisfactory state of OMR evaluation. As pointed out by [29, 81, 84], there is still no good way at the moment of how to measure and compare the performance of OMR systems. The lack of such evaluation methods is best illustrated by the way how OMR literature presents the state of the field: Some consider it a mature area that works well (at least for typeset music) [5, 12, 61, 62, 134]. Others describe their systems with reports of very high accuracies of up to nearly 100% [33, 91, 99, 104, 110, 122, 145, 160, 161], giving an impression of success; however, many of these numbers are symbol detection scores on a small corpus with a limited vocabulary that are not straightforward to interpret in terms of

actual usefulness, since they do not generalize [19, 29]<sup>13</sup>. The existence of commercial applications [71, 106–108, 112, 130, 149] is also sometimes used to support the claim that OMR “works” [13]. On the other hand, many researchers think otherwise [19, 28, 40, 46, 82, 83, 109, 118, 132, 133], emphasizing that OMR does not provide satisfactory solutions in general—not even for typeset music. Some indirect evidence of this can be gleaned from the fact that even for high-quality scans of typeset music, only a few projects rely on OMR,<sup>14</sup> while other projects still prefer to crowdsource the manual transcription instead of using systems for the automatic recognition [78], or at least crowdsource the correction of the errors produced by OMR systems [141]. Given the long-standing absence of OMR evaluation standards, this ambivalence is not surprising. However, a scientific field should be able to communicate its results in comprehensible terms to external stakeholders—something OMR is currently unable to do.

We feel that to a great extent this confusion stems from the fact that the question “Does OMR work?” is an overly vague question. As our analysis in Section 2 shows, OMR is not a monolithic problem—therefore, asking about the “state of OMR” is *under-specified*. “Does OMR work?” must be followed by “... as a tool for X,” where X is some application, in order for such questions to be answerable. There is, again, evidence for this in the OMR literature. OMR systems have been properly evaluated in retrieval scenarios [1, 10, 66] or in the context of digitally replicating a musicological study [83]. It has, in fact, been explicitly asserted [81] that evaluation methodologies are only missing for a limited subset of OMR applications. Specifically, there is no known meaningful edit distance between two scores (whatever their underlying representation).

At the same time, the granularity at which we define the various tasks should not be too fine, otherwise one risks entering a different swamp: instead of no evaluation at all, each individual work is evaluated on the merits of a narrowly defined (and often merely hypothetical) application scenario, which also leads to incomparable contributions. In fact, this risk has already been illustrated on the subtask of symbol detection, which seems like a well-defined problem where the comparison should be trivial. In 2018, multiple music notation object detection papers have been published [82, 116, 117, 152], but each reported results in a different way while presenting a good argument for choosing that kind of evaluation, so significant effort was necessary in order to make these contributions directly comparable [119]. A compromise is therefore necessary between fully specifying the question of whether OMR “works” by asking for a specific application scenario, and on the other hand retaining sufficiently general categories of such tasks.

Having put forward the reasoning for why systematizing the field of OMR with respect to its outputs is desirable, we proceed to do so. For defining meaningful categories of outputs for OMR, we come back to the fundamentals of how OMR inverts the music encoding process to recover the musical semantics and musical notation (see Section 2). These two prongs of reading musical documents roughly correspond to two broad areas of OMR applications [105] that overlap to a certain extent:

- *Replayability*: recovering the encoded music itself in terms of pitch, velocity, onset, and duration. This application area sees OMR as a component inside a bigger music processing pipeline that enables the system to operate on music notation documents as just another input. Notice, that readability by humans is not required for these applications, as long as the computer can process and “play” the symbolic data.

<sup>13</sup>The problem of incomparable results has already been noted in the very first review of OMR in 1972 by Kassler [96] when he reviewed the first two OMR theses by Pruslin [126] and Prerau [123].

<sup>14</sup>Some users of the Choral Public Domain Library (CPDL) project use commercial applications such as SharpEye or PhotoScore Ultimate: <http://forums.cpdll.org/phpBB3/viewtopic.php?f=9&t=9392>



- *Structured Encoding*: recovering the music along with the information on how it was encoded using elements of music notation. This avenue is oriented towards providing the score for music performance, which requires a (lossless) re-encoding of the score and assumes that humans read the OMR output directly. Recovering the musical semantics might not in fact be strictly necessary, but in practice, one often wishes to obtain that information too, in order to enable digitally manipulating the music in a way that would be easiest done with the semantics being recovered (e.g., transposing a part to make it suitable for another instrument).

In other words, the output of an application that targets replayability is typically processed by a machine, whereas humans usually demand the complete recognition of the structured encoding to allow for a readable output (see Fig. 2).

While the distinction between replayability and structured encoding is already useful, there are other reasons that make it interesting to read musical notation from a document. For example, to search for specific content or to draw paleographic conclusions about the document itself. Therefore, we need to broaden the scope of OMR to actually capture these applications. We realized that some use-cases require much less comprehension of the input and music notation than others. To account for this, we propose the following four categories that demand an increasing level of comprehension: *Document Metadata Extraction*, *Search*, *Replayability*, and *Structured Encoding* (see Fig. 13).

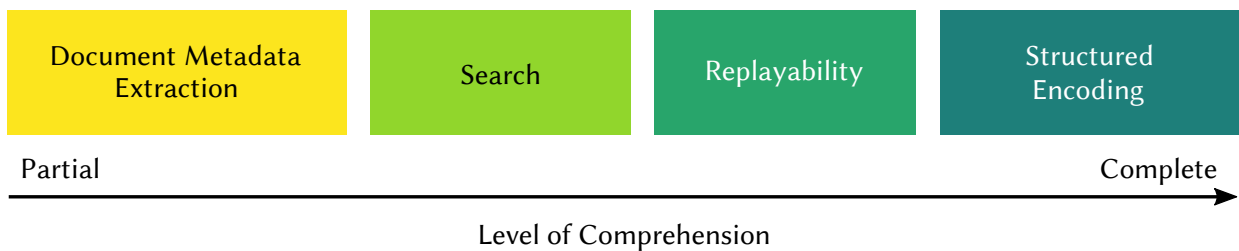


Fig. 13. Taxonomy of four categories of OMR applications that require an increasing level of comprehension, starting with metadata extraction where a minimal understanding might be sufficient, up to structured encoding that requires a complete understanding of music notation with all its intricacies.

Depending on the goal, applications differ quite drastically in terms of requirements—foremost in the choice of output representation. Furthermore, this taxonomy allows us to use different evaluation strategies.

**6.2.1 Document Metadata Extraction.** The first application area requires only a partial understanding of the entire document and attempts to answer specific questions about it. These can be very primitive ones, like whether a document contains music scores or not, but the questions can also be more elaborate, for example:

- In which period was the piece written in?
- What notation was used?
- How many instruments are depicted?
- Are two segments written by the same copyist?

All of the aforementioned tasks entail a different level of underlying computational complexity. However, we are not organizing applications according to their difficulty but instead by the type of answer they provide. In that sense, all of these tasks can be formulated as classification or regression problems, for which the output is either a discrete category or a continuous value, respectively.

**Definition 2.** Document metadata extraction refers to a class of Optical Music Recognition applications that answer questions about the music notation document.

The output representation for document metadata extraction tasks are scalar values or category labels, and if not, its structure is determined by the user, not by the properties of the domain. Again, this does not imply that extracting the target values is necessarily easy, but that the difficulties are not related to the output representation, as is the case for other uses.

Although this type of application has not been very popular in the OMR literature, there are some works that approach this scenario. In [9] and [118] the authors describe systems, that classify images whether they depict music scores or not. While the former one used a basic computer vision approach with a Hough transform and run-length ratios, the latter uses a deep convolutional neural network. Such systems can come in handy if one has to automatically classify a very large number of documents [114]. Perhaps the most prominent application is identifying the writer of a document [63, 64, 77, 139] (which can be different from the composer). This task was one of the main motivations behind the construction of the CVC-MUSCIMA dataset [60] and was featured in the ICDAR 2011 Music Score Competition [59].

The document metadata extraction scenario has the advantage of its unequivocal evaluation protocols. Tasks are formulated regarding either classification or regression, and these have well-defined metrics such as accuracy, f-measure or mean squared error.

**6.2.2 Search.** Nowadays we have access to a vast amount of musical documents. Libraries and communities have taken considerable efforts to catalog and digitize music scores, by scanning them and freely providing users access to them, e.g., IMSLP [125], SLUB [140], DIAMM [20] or CPDL [113], to name a few. Here is a fast growing interest in automated methods which would allow users to search for relevant musical content inside these sources systematically. Unfortunately, searching for specific content often remains elusive, because many projects only provide the images and manually entered metadata. We capture all applications that enable such lookups under the category *Search*. Examples of search questions could be:

- Do I have this piece of music in my library?
- On which page can I find this melody?
- Where does this sequence of notes (e.g., a theme) repeat itself?
- Was a melody copied from another composition?
- Find the same measure in different editions for comparing them.

**Definition 3.** Search refers to a class of Optical Music Recognition applications that, given a collection of sheet music and a musical query, compute the relevance of individual items of the collection with respect to the given query.

Applications from this class share a direct analogy with keyword spotting (KWS) in the text domain [74] and a common formulation: the input is a query, as well as the collection of documents where to look for it; the output is the selection of elements from that collection that match the query. However, “where” is a loose concept and can refer to a complete music piece, a page, or in the most specific cases, a particular bounding-box or even a pixel-level location. In the context of OMR, the musical query must convey musical semantics (as opposed to general search queries, e.g., by title or composer; hence the term “musical” query in Definition 3). The musical query is typically represented in a symbolic way, interpretable unambiguously by the computer (similar to query-by-string in KWS), yet it is also interesting to consider queries that involve other modalities, such as image queries (query-by-example in KWS) or audio queries (query-by-humming in audio information retrieval or query-by-speech in KWS). Additionally, it makes sense to establish different domain-specific types of matching, as it is useful to perform searches restricted to specific music concepts such as melodies, sequences of intervals, or contours, in addition to exact matching.



A direct approach for search within music collections is to use OMR technology to transform the documents into symbolic pieces of information, over which classical content-based or symbolic retrieval methods can be used [1, 14, 47, 52, 55, 88, 97, 151]. The problem is that these transformations require a more comprehensive understanding of the processed documents (see Sections 6.2.3 and 6.2.4 below). To avoid the need for an accurate symbol-by-symbol transcription, search applications can resort to other methods to determine whether (or how likely) a given query is in a document or not. For instance, in cross-modal settings, where one searches a database of sheet music using a MIDI file [10, 66] or a melodic fragment that is given by the user on the fly [1], OMR can be used as a hash function. When the queries and documents are both projected into the search space by the same OMR system, some limitations of the system may even cancel out (e.g., ignoring key signatures), so that retrieval performance might deteriorate less than one would expect. Unfortunately, if either the query or the database contains the true musical semantics, such errors do become critical [83].

A few more works have also approached the direct search of music content without the need to convert the documents into a symbolic format first. Examples comprise the works by [100] dealing with a query-by-example task in the CVC-MUSCIMA dataset, and by [35], considering a classical query-by-string formulation over early handwritten scores. In the cross-modal setting, the audio-sheet music retrieval contributions of [54] are an example of a system that explicitly attempts to gain only the minimum level of comprehension of music notation necessary for performing its retrieval job.

Search systems usually retrieve not just a single result but all those that match the input query, typically sorted by confidence. This setting can re-use general information retrieval methodologies for evaluating performance [87, 101], such as precision and recall, as well as encompassing metrics like average precision and mean average precision.

**6.2.3 *Replayability.*** Replayability applications are concerned with reconstructing the notes encoded in the music notation document. Notice that producing an actual audio file is not considered to be part of OMR, despite being one of the most frequent use-cases of OMR. In any case, OMR can enable these applications by recovering the pitches, velocities, onsets, and durations of notes. This symbolic representation, usually stored as a MIDI file, is already a very useful abstraction of the music itself and allows for plugging in a vast range of computational tools such as:

- synthesis software to produce an audio representation of the composition
- music information retrieval tools that operate on symbolic data
- tools that perform large-scale music-theoretical analysis
- creativity-focused applications [162]

**Definition 4.** Replayability refers to a class of Optical Music Recognition applications that recover sufficient information to create an audible version of the written music.

Producing a MIDI (or an equivalent) representation is one key goal for OMR—at least for the foreseeable future since MIDI is a representation of music that has a long tradition of computational processing for a vast variety of purposes. Many applications have been envisioned, that only require replayability, such as applications, that can sight-read the scores to assist practicing musicians or provide missing accompaniment.

Replayability is also a major concern for digital musicology. Historically, the majority of compositions has probably never been recorded, and therefore is only available in written form as scores; of these, most compositions have also never been typeset, since typesetting has been a very expensive endeavor, reserved essentially either for works with assured commercial success, or composers with substantial backing by wealthy patrons. Given the price of manual transcription, it is prohibitive to transcribe large historical archives. OMR that produces MIDI, especially if it can do

so for manuscripts, is probably the only tool that could open up the vast amount of compositions to quantitative musicological research, which, in turn, could perhaps finally start answering broad questions about the evolutions of the average musical styles, instead of just relying on the works of the relatively few well-known composers.

Systems designed for the goal of replayability traditionally seek first to obtain the structured encoding of the score (see Section 6.2.4), from which the sequences of notes can be straightforwardly retrieved [82]. However, if the specific goal is to obtain something equivalent to a MIDI representation, it is possible to simplify the recognition and ignore many of the elements of musical notation, as demonstrated by numerous research projects [16, 65, 90, 91, 102, 116, 138]. An even clearer example of this distinction can be observed in the works of Shi et al. [146] as well as van der Wel and Ullrich [157]; both focus only on obtaining the sequence of note pairs (duration, pitch) that are depicted in single-staff images, regardless of how these notes were actually expressed in the document. Another instance of a replay-oriented application is the Gocen system [5] that reads handwritten notes with a specially designed device with the goal of producing a musical performance while ignoring the majority of music notation syntax.

Once a system is able to arrive at a MIDI-like representation, evaluating the results is a matter of comparing sets of pitch-onset-duration-triplets. Velocities may optionally be compared too, once the note-by-note correspondence has been established, but can be seen as secondary for many applications. Note, however, that even on the level of describing music as configurations of pitch-velocity-onset-duration-quadruples, MIDI is a further simplification that is heavily influenced by its origin as a digital representation of performance, rather than of a composition: the most obvious inadequacy of MIDI is its inability to distinguish pitches that sound equivalent but are named differently, e.g., F-sharp and G-flat.<sup>15</sup>

Multiple similarity metrics for comparing MIDI files have been proposed during the Symbolic Melodic Similarity track of the Music Information Retrieval Evaluation eXchange (MIREX),<sup>16</sup> e.g., by determining the local alignment between the geometric representations of the melodies [153–156]. Other options could be multi-pitch estimation evaluation metrics [17], Dynamic Time Warping [54], or edit distances between two time-ordered sequences of pitch-duration pairs [33, 163].

**6.2.4 Structured Encoding.** It can be reasonably stated that digitizing music scores for “human consumption” and score manipulation tasks that a *vollkommener Capellmeister*<sup>17</sup> [103] routinely performs, such as part exporting, merging, or transposing for available instruments is the original motivation of OMR ever since it started [6, 67, 123, 126] and the one that appeals to the widest audience. Given that typesetting music is troublesome and time-consuming, OMR technology represents an attractive alternative to obtain a digital version of music scores on which these operations can be performed efficiently with the assistance of the computer.

This brings us to our fourth and last category that requires the highest level of comprehension, called structured encoding. Structured encoding aims to recognize the entire music score while retaining all the engraving information available to a human reader. Since there is no viable alternative to music notation, the system has to fully transcribe the document into a structured digital format with the ultimate goal of keeping the same musical information that could be retrieved from the physical score itself.

<sup>15</sup>This is the combined heritage of equal temperament, where these two pitches do correspond to the same fundamental frequency, and of the origins of MIDI in genres dominated by fretted and keyboard instruments.

<sup>16</sup> [https://www.music-ir.org/mirex/wiki/MIREX\\_HOME](https://www.music-ir.org/mirex/wiki/MIREX_HOME)

<sup>17</sup>roughly translated from German as “ideal conductor”

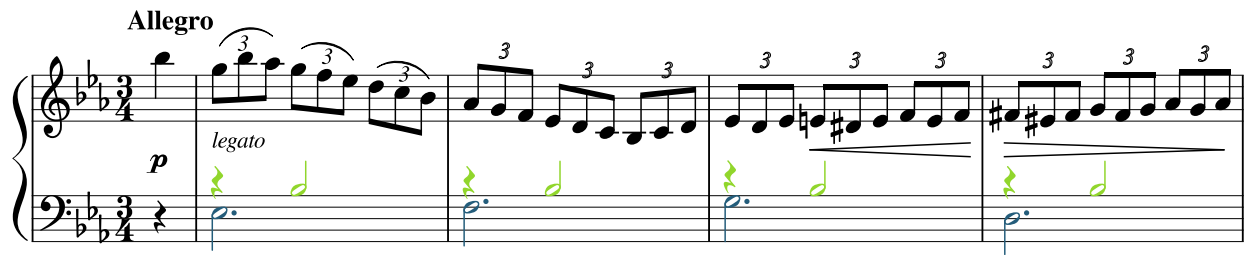


Fig. 14. Beginning of Franz Schubert, Impromptu D.899 No. 2 with omitted thirds starting in the second measure of the top staff (gray) and a color-coding of the two distinct voices in the second staff (green and blue).

**Definition 5.** Structured Encoding refers to a class of Optical Music Recognition applications that fully decode the musical content, along with the information of 'how' it was encoded by means of music notation.

Note that the difference between replayability and structured encoding can seem vague: for instance, imagine a system that detects all notes and all other symbols and exports them into a MusicXML file. The result, however, is not the structured encoding unless the system also attempts to preserve the information on how the scores were laid out. That does not mean it has to store the bounding box and exact location of every single symbol, but the engraving information that *conveys musical semantics*, like whether the stem of a note went up or down. To illustrate this, consider the following musical snippet in Fig. 14. If a system like the one described in [33] recognized this, it would remain restricted to replayability. Not because of the current limitations to monophonic, single-staff music, but due to the selected output representation, which does not store engraving information such as the simplifications that start in the second measure of the top staff (the grayed out 3s that would be omitted in the printing) or the stem directions of the notes in the bottom staff (green and blue) that depict two different voices. In summary, any system discarding engraving information that conveys musical semantics cannot reach, by definition, the structured encoding goal.

To help understand, why structured encoding poses such a difficult challenge, we would like to avail ourselves of the intuitive comparison given by Donald Byrd<sup>18</sup>: representing music as time-stamped events (e.g., with MIDI) is similar to storing a piece of writing in a plain text file; whereas representing music with music notation (e.g., with MusicXML) is similar to a structured description like an HTML website. By analogy, obtaining the structured encoding from the image of a music score can be as challenging as recovering the HTML source code from the screenshot of a website.

Since this use-case appeals to the widest audience, it has seen development both from the scientific research community and commercial vendors. Notable products that attempt full structured encoding include SmartScore [106], Capella Scan [37], PhotoScore [108] as well as the open-source application Audiveris [21]. While the projects described in many scientific publications seem to be striving for structured encoding to enable interesting applications such as the preservation of the cultural heritage [39], music renotation [41], or transcriptions between different music notation languages [135], we are not aware of any systems in academia that would actually produce structured encoding.

A major stumbling block for structured encoding applications has for a long time been the lack of practical formats for representing music notation that would be powerful enough to retain the

<sup>18</sup>[http://music.informatics.indiana.edu/don\\_notation.html](http://music.informatics.indiana.edu/don_notation.html)

information from the input score, and at the same time be a natural endpoint for OMR. This is illustrated by papers that propose OMR-specific representations, both before the emergence of MusicXML [75, 76] as a viable interchange format [105] and after [86]. At the same time, however, even without regard for OMR, there are ongoing efforts to improve music notation file formats: further development of MusicXML has moved into the W3C Music Notation Community Group<sup>19</sup>, and there is an ongoing effort in the development of the Music Encoding Initiative format [137], best illustrated by the annual Music Encoding Conference.<sup>20</sup> Supporting the whole spectrum of music notation situations that arise in a reasonably-sized archive is already a difficult task. This can be evidenced by the extensive catalog of requirements for music notation formats that Byrd and Isaacson [27] list for a multi-purpose digital archive of music scores. Incidentally, the same paper also mentions support for syntactically incorrect scores among the requirements, which is one of the major problems that OMR has with outputting to existing formats directly. Although these formats are becoming more precise and descriptive, they are not designed to store information about how the content was automatically recognized from the document. This kind of information is actually relevant for systems' evaluation, as it allows, for example, determining if a pitch was misclassified because of either a wrongly detected position in the staff or a wrongly detected clef.

The imperfections of representation standards for music notation is also reflected in a lack of evaluation standards for structured encoding. Given the ground truth representation of a score and the output of a recognition system, there is currently no automatic method that is capable of reliably computing how well the recognition system performed. Ideally, such a method would be rigorously described and evaluated, have a public implementation, and give meaningful results. Within the traditional OMR pipeline, the partial steps (such as symbol detection) can use rather general evaluation metrics. However, when OMR is applied for getting the structured encoding of the score, no evaluation metric is available, or at least generally accepted, partially because of the lack of a standard representation for OMR output, as mentioned earlier. The notion of “edit cost” or “recognition gain” that defines success in terms of how much time a human editor saves by using an OMR system is yet more problematic, as it depends on the editor and on the specific toolchain [19].

There is no reason why a proper evaluation should not be possible since there is only a finite amount of information that a music document retains, which can be exhaustively enumerated. It follows that we should be able to measure what proportion of this information our systems recovered correctly. The rationale why this is still such a hard problem is because there is no underlying *formal model of music notation*. Such a model could support structured encoding evaluation by being:

- *Comprehensive*: integrating naturally both the “reprintability” and “replayability” level (also called graphical and semantical level in the literature), by being capable of describing the various corner cases (which implies extensibility);
- *Useful*: enabling tractable inference (at least approximate) and an adequate distance function; and
- *Sufficiently supported* through open-source software.

The existing XML formats for encoding music notation are inadequate representations for OMR. For example, the XML tree structure is unsuitable, as evidenced by the frequent need for referencing the XML elements across arbitrarily distant subtrees. Historically, context-free grammars have been the most explored avenue for a unified formal description of music notation, both with an explicit grammar [4, 49] and implicitly using a modified stack automaton [8]: this feels natural, given that music notation has strict syntactic rules and hierarchical structures that invite such

<sup>19</sup><https://www.w3.org/community/music-notation/>

<sup>20</sup><https://music-encoding.org/conference/past.html>

descriptions. The 2-D nature of music notation also inspired graph grammars [56] and attributed graph grammars [15]. Recently, modeling music notation as a directed acyclic graph has been proposed as an alternative [82, 86]. However, none of these formalisms has yet been adopted: the notation graph is too recent and does not have sufficient software and community support, and the older grammar-based approaches lack up-to-date open-source implementations altogether (and are insufficiently detailed in the respective publications for re-implementation). Without an appropriate formalism and the corresponding tooling, the evaluation of structured encoding can hardly hope to move beyond ad-hoc methods.

Hajič [81] argues that a good OMR evaluation metric should be intrinsic<sup>21</sup> and independent of a certain use-case. The benefits would be the independence from the selected score editing toolchain as well as the music notation format and a clearly interpretable automatic metric for guiding OMR development (which could ideally be used as a differentiable loss function for training full-pipeline end-to-end machine learning-based systems). This question is still one of the major issues in the field.

## 7 APPROACHES TO OMR

In order to complete our journey through the landscape of *Optical Music Recognition*, we have yet to visit the arena of OMR techniques. These have recently undergone a paradigm shift towards machine learning that has brought about a need to revisit the way that OMR methods have traditionally been systematized. As opposed to OMR applications, the vocabulary of OMR methods and subtasks already exists [132] and only needs to be updated to reflect the new reality of the field.

As mentioned before, obtaining the structured encoding of the scores has been the main motivation to develop the OMR field. Given the difficulty of such objective, the process was usually approached by dividing it into smaller stages that could represent challenges within reach with the available technologies and resources. Over the years, the pipeline described by Bainbridge and Bell [7], refined by Rebelo et al. in 2012 [132] became the de-facto standard. That pipeline is traditionally organized into the following four blocks, sometimes with slightly varying names and scopes of the individual stages:

- (1) *Preprocessing*: Standard techniques to ease further steps, e.g., contrast enhancement, binarization, skew-correction or noise removal. Additionally, the layout should be analyzed to allow subsequent steps to focus on actual content and ignore the background.
- (2) *Music Object Detection*: Finding and classifying all relevant symbols or glyphs in the image.
- (3) *Notation Assembly*: Recovering the music notation semantics from the detected and classified symbols. The output is a symbolic representation of the symbols and their relationships, typically as a graph.
- (4) *Encoding*: Encoding the music into any output format unambiguously, e.g., into MIDI for playback or MusicXML/MEI for further editing in a music notation program.

With the appearance of deep learning in OMR, many steps that traditionally produced suboptimal results, such as the staff-line removal or symbol classification have seen drastic improvements [70, 118] and are nowadays considered solved or at least clearly solvable. This caused some steps to become obsolete or collapse into a single (bigger) stage. For instance, the music object detection stage was traditionally separated into a segmentation stage and classification stage. Since staff lines make it hard to separate isolated symbols through connected component analysis, they typically were removed first, using a separate method. However, deep learning models with convolutional

<sup>21</sup>Extrinsic evaluation means evaluating the system in an application context: “How good is this system for purpose X?” Intrinsic evaluation attempts to evaluate a system without reference to a specific use-case, asking how much of the encoded information has been recovered. In the case of OMR, this essentially reduces evaluation to error counting.



neural networks have been shown to be able to deal with the music object detection stage holistically without having to remove staff lines at all. In addition to the performance gains, a compelling advantage is the capability of these models to train them in a single step by merely providing pairs of images and positions of the music objects to be found, eliminating the preprocessing step altogether. A baseline of competing approaches on several datasets containing both handwritten and typeset music can be found in the work of Pacha et al. [119].

The recent advances also diversified the way of how OMR is approached altogether: there are alternative pipelines with their own ongoing research that attempt to face the whole process in a single step. This holistic paradigm, also referred to as end-to-end systems, has been dominating the current state of the art in other tasks such as text, speech, or mathematical formula recognition [45, 48, 163]. However, due to the complexity of how musical semantics are inferred from the image, it is difficult (for now) to formulate it as a learnable optimization problem. While end-to-end systems for OMR do exist, they are still limited to a subset of music notation, at best. Pugin pioneered this approach utilizing hidden Markov models for the recognition of typeset mensural notation [127], and some recent works have considered deep recurrent neural networks for monophonic music written in both typeset [32, 33, 146, 157] and handwritten [13] modern notation. Unfortunately, polyphonic and pianoform scores are currently out of reach for end-to-end models—not just that the results would be disappointing, there is simply no appropriate model formulation. Therefore, even when only trying to produce the “notes” (semantics), one may choose to recover some of the engraving decisions explicitly as well, relying on the rules of inferring musical semantics as in the last stages of the traditional pipeline.

Along with the paradigm shift towards machine learning—which nowadays can be considered widely established—several public datasets have emerged, such as MUSCIMA++ [86], DeepScores [152] or Camera-PrIMuS [32].<sup>22</sup> There are also significant efforts to develop tools by which training data for OMR systems can be obtained including MUSCIMarker [85], Pixel.js [142], and MuRET [135].

On the other hand, while the machine learning paradigm has undeniably brought significant progress, it has shifted the costs onto data acquisition. This means that while the machine learning paradigm is more general and delivers state-of-the-art results when appropriate data is available, it does not necessarily drive down the costs of applying OMR. Still, we would say—tentatively—that once these resources are spent, the chances of OMR yielding useful results for the specific use-case are higher compared to earlier paradigms.

Tangentially to the way of dealing with the process itself, there has been continuous research on interactive systems for years. The idea behind such systems is based on the insight that OMR systems might always make some errors, and if no errors can be tolerated, the user is essential to correct the output. These systems attempt to incorporate user feedback into the OMR process in a more efficient way than just post-processing system output. Most notably is the interactive system developed by Chen et al. [42, 43], where the user directly interacts with the OMR system by specifying which constraints to take into account while visually recognizing the scores. The user can then iteratively add or remove constraints before re-recognizing individual measures until he is satisfied. The most powerful feature of interactive systems is probably the displaying of recognition results, superimposed on top of the original image, which allows to quickly spot errors [21, 37, 135, 159].

---

<sup>22</sup>A full list of all available datasets can be found at <https://apacha.github.io/OMR-Datasets/>

## 8 CONCLUSIONS

In this article, we have first addressed what *Optical Music Recognition* is and proposed to define it as research field that investigates how to computationally read music notation in documents—a definition that should adequately delimit the field, and set it in relation to other fields such as OCR, graphics recognition, computer vision, and fields that await OMR results. We furthermore analyzed in depth the inverse relation of OMR to the process of writing down a musical composition and highlighted the relevance of engraving music properly—something that must also be recognized to ensure readability for humans. The investigation of what OMR is, revealed why this seemingly easy task of reading music notation has turned out to be such a hard problem: besides the technical difficulties associated with document analysis, many fundamental challenges arise from the way how music is expressed and captured in music notation. By providing a sound, concise and inclusive definition, we capture how the field sees and talks about itself.

We have then reviewed and improved the taxonomy of OMR, which should help systematize the current and future contributions to the field. While the inputs of OMR systems have been described systematically and established throughout the field, a taxonomy of OMR outputs and applications has not been proposed before. An overview of this taxonomy is given in Fig. 15.

Finally, we have also updated the general breakdown of OMR systems into separate subtasks in order to reflect the paradigm shift towards machine learning methods and discussed alternative paradigms such as end-to-end systems and interactive scenarios.

One of the key points we wanted to stress is the internal diversity of the field: OMR is not a monolithic task. As analyzed in Section 4, it enables various use-cases that require fundamentally different system designs, as discussed in Section 6.2. So before creating an OMR system, one should be clear about the goals and the associated challenges.

The sensitivity to errors is another relevant issue that needs to be taken into account. As long as errors are inevitable [43, 50], it is important to consider the impact of those errors to the envisioned application. If someone wants to transcribe a score with an OMR system, but the effort needed for correcting the errors is greater than the effort for directly entering the notes into a music notation program, such an OMR system would obviously be useless [19]. Existing literature on error-tolerance is inconclusive: while we tend to believe that users—especially practicing musicians—would not tolerate false recognitions [136], we also see systems that can handle a substantial amount of OMR errors [1, 50, 83] and still produce meaningful results, e.g., when searching in a large database of scores. Therefore, it cannot be decided in advance how severe errors are, as it is always the end user who sets the extent of tolerable errors.

The reader should now comprehend the spectrum of what OMR might do, understand the challenges that reading music notation entails, and have a solid basis for further exploring the field on his own—in other words, be equipped to address the issues described in the next section.

### 8.1 Open Issues and Perspectives for Future Research

We conclude this paper by listing major open problems in Optical Music Recognition that significantly impede its progress and usefulness. While some of them are technical challenges, there are also many non-technical issues:

- *Legal aspects:* Written music is the intellectual property of the composer and its allowed uses are defined by the respective publisher. Recognizing and sharing music scores can be seen as copyright infringement, like digitizing books without permission. To avoid this dispute, many databases such as IMSLP only store music scores whose copyright protection has expired. So an OMR dataset is either limited to old scores or one enters a legal gray area if not paying close attention to the respective license of every piece stored therein.



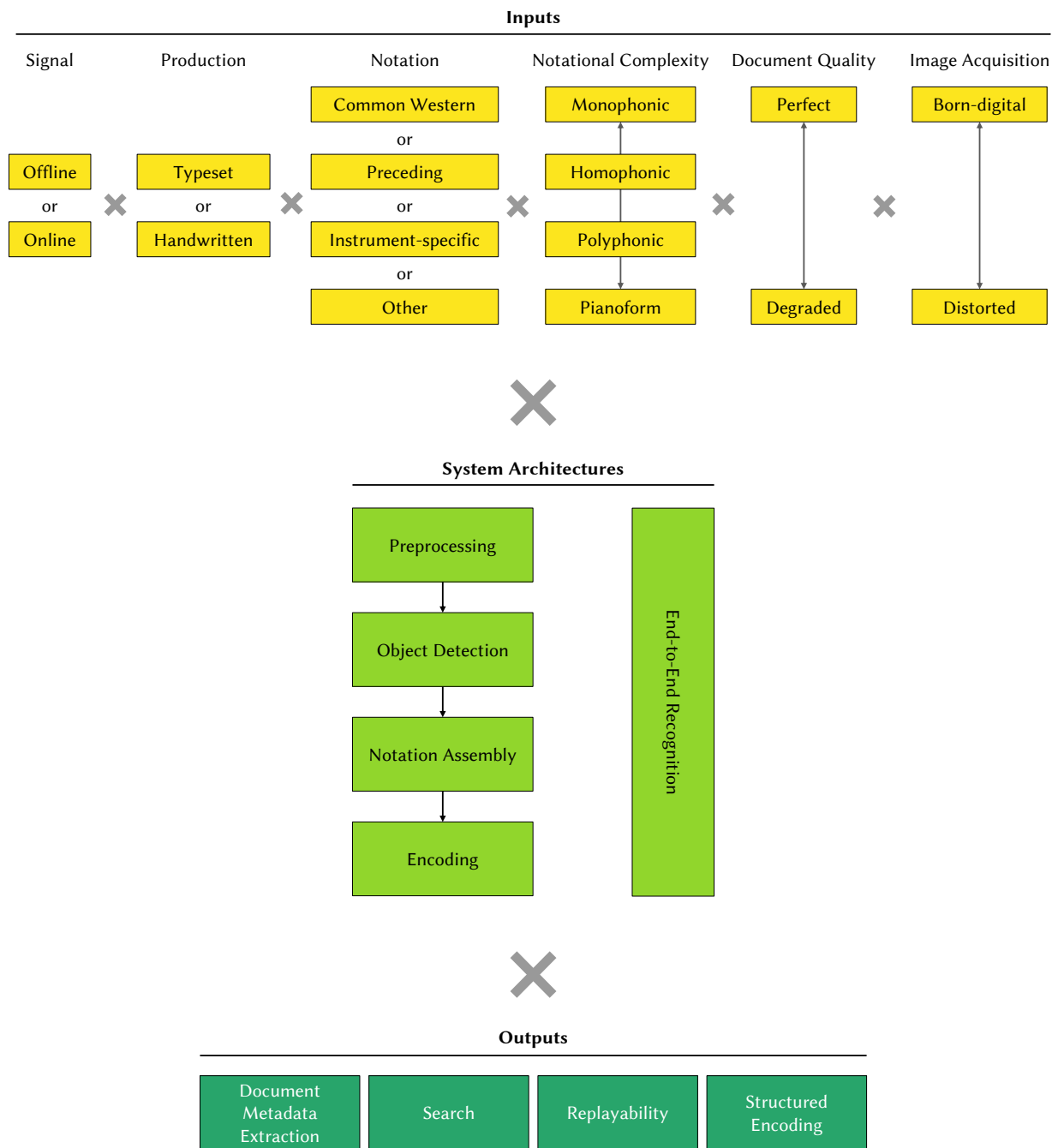


Fig. 15. An overview of the taxonomy of OMR inputs, architectures, and outputs. A fairly simple OMR system could, for example, read high-quality scans (offline) of well-preserved documents that contain typeset, monophonic, mensural notation, process it in a tradition pipeline and output the results in a MIDI file to achieve replayability. An extremely complex system, on the other hand, would allow images (offline) of handwritten music in common western notation from degraded documents as input and strive to recognize the full structured encoding in an end-to-end system.

- *Stable community*: For decades, OMR research was conducted by just a few individuals that worked distributedly and mostly uncoordinated. Most OMR researchers joined the field with minor contributions but left again soon afterward. Furthermore, due to a lack of dedicated venues, researchers rarely met in person [30]. This unstable setting and researchers that were not paying sufficient attention to reproducibility led to the same problems being solved over and over again [115].
- *Lack of standards representations*: There exist no standard representation formats for OMR outputs, especially not for structured encoding, and virtually every system comes with its own internal representation and output format, even for intermediate steps. This causes incompatibilities between different systems and makes it very hard to replace subcomponents. Work on underlying formalisms for describing music notation can also potentially have a wide impact, especially if done in collaboration with the relevant communities (W3C Community Group on Music Notation, Music Encoding Initiative).
- *Evaluation*: Due to the lack of standards for outputting OMR results, evaluating them is currently in an equally unsatisfactory state. An ideal evaluation method would be rigorously described and verified, have a public implementation, give meaningful results, and not rely on a particular use-case, thus only intrinsically evaluating the system [81].

On the technical side, there are also many interesting avenues, where future research is needed, including:

- *Music Object Detection*: recent work has shown that the music object detection stage can be addressed in one step with deep neural networks. However, the accuracy is still far from optimal, which is especially detrimental to the following stages of the pipeline that are based on these results. In order to improve the detection performance, it might be interesting to develop models that are specific to the type of inputs that OMR works on: large images with a high quantity of densely packed objects of various sizes from a vast vocabulary.
- *Semantical reconstruction*: merely detecting the music objects in the document does not represent a complete music notation recognition system, and so the music object detection stage must be complemented with the semantical reconstruction. Traditionally, this stage is addressed by hand-crafted heuristics that either hardly generalize or do not cover the full spectrum of music notation. Machine learning-based semantical reconstruction represents an unexplored line of research that deserves further consideration.
- *Structured encoding research*: despite being the main motivation for OMR in many cases, there is a lack of scientific research and open systems that actually pursue the objective of retrieving the full structure encoding of the input.
- *Full end-to-end systems*: end-to-end systems are accountable for major advances in machine learning tasks such as text recognition, speech recognition, or machine translation. The state of the art of these fields is based on recurrent neural networks. For design reasons, these networks currently deal only with one-dimensional output sequences. This fits the aforementioned tasks quite naturally since their outputs are mainly composed of word sequences. However, its application for music notation—except for simple monophonic scores—is not so straightforward, and it is unknown how to formulate an end-to-end learning process for the recognition of fully-fledged music notation in documents.
- *Statistical modeling*: most machine learning algorithms are based on statistical models that are able to provide a probability distribution over the set of possible recognition hypotheses. When it comes to recognizing, we are typically interested in the best hypothesis—the one that is proposed as an answer—forgetting the probability given to such hypothesis by the model. However, it could be interesting to be able to exploit this uncertainty. For example, in

the standard decomposition of stages in OMR systems, the semantic reconstruction stage could benefit from having a set of hypotheses about the objects detected in the previous stage, instead of single proposals. Then, the semantic reconstruction algorithm could establish relationships that are more logical a priori, although the objects involved have a lower probability according to the object detector. These types of approaches have not been deeply explored in the OMR field. Statistical modeling could also be useful so that the system provides its certainty about the output. Then, the end user might have a certain notion about the accuracy that has been obtained for the given input.

- *Generalizing systems*: A pressing issue is generalizing from training datasets to various real-world collections because the costs for data acquisition are still significant and currently represent a bottleneck for applying state-of-the-art machine learning models in stakeholders' workflows. However, music notation follows the same underlying rules, regardless of graphical differences such as whether it is typeset or handwritten. Can one leverage a typeset sheet music dataset to train for handwritten notation? Given that typeset notation can be synthetically generated, this would open several opportunities to train handwritten systems without the effort of getting labeled data manually. Although it seems more difficult to transfer knowledge across different kinds of music notation, a system that recognizes some specific music notation could be somehow useful for the recognition of shared elements in other styles as well, e.g., across the various mensural notation systems.
- *Interactive systems*: Interactive systems are based on the idea of including users in the recognition process, given that they are necessary if there is no tolerance for errors—something that at the moment can only be ensured by human verification. This paradigm reformulates the objective of the system, which is no longer improving accuracy but reducing the effort—usually measured as time—that the users invest in aiding the machine to achieve that perfect result. This aid can be provided in many different ways: error corrections that then feed back into the system, or manually activating and deactivating constraints on the content to be recognized. However, since user effort is the most valuable resource, there is still a need to reformulate the problem based on this concept, which also includes aspects related to human-computer interfaces. The conventional interfaces of computers are designed to enter text (keyboard) or perform very specific actions (mouse); therefore, it would be interesting to study the use of more ergonomic interfaces to work with musical notation, such as an electronic pen or a MIDI piano, in the context of interactive OMR systems.

We hope that these lists demonstrate that OMR still provides many interesting challenges that await future research.

## ACKNOWLEDGMENTS

The authors would like to thank David Rizo and Horst Eidenberger for their valuable feedback and helpful comments on the manuscript.

## REFERENCES

- [1] Sanu Pulimootil Achankunju. 2018. Music Search Engine from Noisy OMR Data. In *1st International Workshop on Reading Music Systems*. Paris, France, 23–24.
- [2] Julia Adamska, Mateusz Piecuch, Mateusz Podgórski, Piotr Walkiewicz, and Ewa Lukasik. 2015. Mobile System for Optical Music Recognition and Music Sound Generation. In *Computer Information Systems and Industrial Management*. Cham, 571–582.
- [3] Francisco Álvaro, Joan-Andreu Sánchez, and José-Miguel Benedí. 2016. An integrated grammar-based approach for mathematical expression recognition. *Pattern Recognition* 51 (2016), 135–147.
- [4] Alfio Andronico and Alberto Ciampa. 1982. On Automatic Pattern Recognition and Acquisition of Printed Music. In *International Computer Music Conference*. Venice, Italy.

- [5] Tetsuaki Baba, Yuya Kikukawa, Toshiki Yoshiike, Tatsuhiko Suzuki, Rika Shoji, Kumiko Kushiyama, and Makoto Aoki. 2012. Gocen: A Handwritten Notational Interface for Musical Performance and Learning Music. In *ACM SIGGRAPH 2012 Emerging Technologies*. New York, USA, 9–9.
- [6] David Bainbridge and Tim Bell. 1997. Dealing with superimposed objects in optical music recognition. In *6th International Conference on Image Processing and its Applications*. 756–760.
- [7] David Bainbridge and Tim Bell. 2001. The Challenge of Optical Music Recognition. *Computers and the Humanities* 35, 2 (2001), 95–121.
- [8] David Bainbridge and Tim Bell. 2003. A music notation construction engine for optical music recognition. *Software: Practice and Experience* 33, 2 (2003), 173–200.
- [9] David Bainbridge and Tim Bell. 2006. Identifying music documents in a collection of images. In *7th International Conference on Music Information Retrieval*. Victoria, Canada, 47–52.
- [10] Stefan Balke, Sanu Pulimootil Achankunju, and Meinard Müller. 2015. Matching Musical Themes based on noisy OCR and OMR input. In *International Conference on Acoustics, Speech and Signal Processing*. 703–707.
- [11] Arnau Baró, Pau Riba, Jorge Calvo-Zaragoza, and Alicia Fornés. 2017. Optical Music Recognition by Recurrent Neural Networks. In *14th International Conference on Document Analysis and Recognition*. IEEE, Kyoto, Japan, 25–26.
- [12] Arnau Baró, Pau Riba, and Alicia Fornés. 2016. Towards the recognition of compound music notes in handwritten music scores. In *15th International Conference on Frontiers in Handwriting Recognition*. 465–470.
- [13] Arnau Baró, Pau Riba, and Alicia Fornés. 2018. A Starting Point for Handwritten Music Recognition. In *1st International Workshop on Reading Music Systems*. Paris, France, 5–6.
- [14] Louis W. G. Barton. 2002. The NEUMES Project: digital transcription of medieval chant manuscripts. In *2nd International Conference on Web Delivering of Music*. 211–218.
- [15] Stephan Baumann. 1995. A Simplified Attributed Graph Grammar for High-Level Music Recognition. In *3rd International Conference on Document Analysis and Recognition*. 1080–1083.
- [16] Stephan Baumann and Andreas Dengel. 1992. *Transforming Printed Piano Music into MIDI*. World Scientific, 363–372.
- [17] Mert Bay, Andreas F. Ehmann, and J. Stephen Downie. 2009. Evaluation of Multiple-F0 Estimation and Tracking Systems. In *10th International Society for Music Information Retrieval Conference*. Kobe, Japan, 315–320.
- [18] Pierfrancesco Bellini, Ivan Bruno, and Paolo Nesi. 2001. Optical music sheet segmentation. In *1st International Conference on WEB Delivering of Music*. 183–190.
- [19] Pierfrancesco Bellini, Ivan Bruno, and Paolo Nesi. 2007. Assessing Optical Music Recognition Tools. *Computer Music Journal* 31, 1 (2007), 68–93.
- [20] Margaret Bent and Andrew Wathey. 1998. Digital Image Archive of Medieval Music. <https://www.diamm.ac.uk/>
- [21] Hervé Bitteur. 2004. Audiveris. <https://github.com/audiveris>
- [22] Dorothea Blostein and Henry S. Baird. 1992. *A Critical Survey of Music Image Analysis*. Springer Berlin Heidelberg, 405–434.
- [23] Dorothea Blostein and Nicholas Paul Carter. 1992. *Recognition of Music Notation: SSPR'90 Working Group Report*. Springer Berlin Heidelberg, 573–574.
- [24] Dorothea Blostein and Lippold Haken. 1991. Justification of Printed Music. *Commun. ACM* 34, 3 (1991), 88–99.
- [25] John Ashley Burgoyne, Johanna Devaney, Laurent Pugin, and Ichiro Fujinaga. 2008. Enhanced Bleedthrough Correction for Early Music Documents with Recto-Verso Registration. In *9th International Conference on Music Information Retrieval*. Philadelphia, PA, 407–412.
- [26] John Ashley Burgoyne, Ichiro Fujinaga, and J. Stephen Downie. 2015. *Music Information Retrieval*. Wiley Blackwell, 213–228.
- [27] Donald Byrd and Eric Isaacson. 2016. *A Music Representation Requirement Specification for Academia*. Technical Report. Indiana University, Bloomington.
- [28] Donald Byrd and Megan Schindele. 2006. Prospects for Improving OMR with Multiple Recognizers. In *7th International Conference on Music Information Retrieval*. 41–46.
- [29] Donald Byrd and Jakob Grue Simonsen. 2015. Towards a Standard Testbed for Optical Music Recognition: Definitions, Metrics, and Page Images. *Journal of New Music Research* 44, 3 (2015), 169–195.
- [30] Jorge Calvo-Zaragoza, Jan jr. Hajič, and Alexander Pacha. 2018. Discussion Group Summary: Optical Music Recognition. In *Graphics Recognition, Current Trends and Evolutions (Lecture Notes in Computer Science)*. 152–157.
- [31] Jorge Calvo-Zaragoza and Jose Oncina. 2014. Recognition of Pen-Based Music Notation: The HOMUS Dataset. In *22nd International Conference on Pattern Recognition*. 3038–3043.
- [32] Jorge Calvo-Zaragoza and David Rizo. 2018. Camera-PriMuS: Neural End-to-End Optical Music Recognition on Realistic Monophonic Scores. In *19th International Society for Music Information Retrieval Conference*. Paris, France, 248–255.
- [33] Jorge Calvo-Zaragoza and David Rizo. 2018. End-to-End Neural Optical Music Recognition of Monophonic Scores. *Applied Sciences* 4 (2018).

- [34] Jorge Calvo-Zaragoza, Alejandro Toselli, and Enrique Vidal. 2017. Handwritten Music Recognition for Mensural Notation: Formulation, Data and Baseline Results. In *14th International Conference on Document Analysis and Recognition*. Kyoto, Japan, 1081–1086.
- [35] Jorge Calvo-Zaragoza, Alejandro H. Toselli, and Enrique Vidal. 2018. Probabilistic Music-Symbol Spotting in Handwritten Scores. In *16th International Conference on Frontiers in Handwriting Recognition*. Niagara Falls, USA, 558–563.
- [36] Carlos E. Cancino-Chacón, Maarten Grachten, Werner Goebel, and Gerhard Widmer. 2018. Computational Models of Expressive Music Performance: A Comprehensive and Critical Review. *Frontiers in Digital Humanities* 5 (2018), 25.
- [37] capella-software AG. 1996. Capella Scan. <https://www.capella-software.com>
- [38] Nicholas Paul Carter. 1992. *A New Edition of Walton's Façade Using Automatic Score Recognition*. World Scientific, 352–362.
- [39] Gen-Fang Chen and Jia-Shing Sheu. 2014. An optical music recognition system for traditional Chinese Kunqu Opera scores written in Gong-Che Notation. *EURASIP Journal on Audio, Speech, and Music Processing* 2014, 1 (2014), 7.
- [40] Liang Chen and Kun Duan. 2016. MIDI-assisted egocentric optical music recognition. In *Winter Conference on Applications of Computer Vision*.
- [41] Liang Chen, Rong Jin, and Christopher Raphael. 2015. Renotation from Optical Music Recognition. In *Mathematics and Computation in Music*. Cham, 16–26.
- [42] Liang Chen, Rong Jin, and Christopher Raphael. 2017. Human-Guided Recognition of Music Score Images. In *4th International Workshop on Digital Libraries for Musicology*.
- [43] Liang Chen and Christopher Raphael. 2018. Optical Music Recognition and Human-in-the-loop Computation. In *1st International Workshop on Reading Music Systems*. Paris, France, 11–12.
- [44] Atul K. Chhabra. 1998. Graphic symbol recognition: An overview. In *Graphics Recognition Algorithms and Systems*. Berlin, Heidelberg, 68–79.
- [45] Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. 2018. State-of-the-Art Speech Recognition with Sequence-to-Sequence Models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 4774–4778.
- [46] Kwon-Young Choi, Bertrand Couasnon, Yann Riquebourg, and Richard Zanibbi. 2017. Bootstrapping Samples of Accidentals in Dense Piano Scores for CNN-Based Detection. In *14th International Conference on Document Analysis and Recognition*. IAPR TC10 (Technical Committee on Graphics Recognition), Kyoto, Japan.
- [47] G. Sayeed Choudhury, M. Droetboom, Tim DiLauro, Ichiro Fujinaga, and Brian Harrington. 2000. Optical Music Recognition System within a Large-Scale Digitization Project. In *1st International Symposium on Music Information Retrieval*.
- [48] Arindam Chowdhury and Lovekesh Vig. 2018. An Efficient End-to-End Neural Model for Handwritten Text Recognition. In *29th British Machine Vision Conference*.
- [49] Bertrand Couasnon and Jean Camillerapp. 1994. Using Grammars To Segment and Recognize Music Scores. In *International Association for Pattern Recognition Workshop on Document Analysis Systems*. Kaiserslautern, Germany, 15–27.
- [50] Tim Crawford, Golnaz Badkobeh, and David Lewis. 2018. Searching Page-Images of Early Music Scanned with OMR: A Scalable Solution Using Minimal Absent Words. In *19th International Society for Music Information Retrieval Conference*. Paris, France, 233–239.
- [51] Christoph Dalitz, Georgios K. Michalakis, and Christine Pranzas. 2008. Optical recognition of psaltic Byzantine chant notation. *International Journal of Document Analysis and Recognition* 11, 3 (2008), 143–158.
- [52] Jürgen Diet. 2018. Innovative MIR Applications at the Bayerische Staatsbibliothek. In *5th International Conference on Digital Libraries for Musicology*. Paris, France.
- [53] Ing-Jr Ding, Chih-Ta Yen, Che-Wei Chang, and He-Zhong Lin. 2014. Optical music recognition of the singer using formant frequency estimation of vocal fold vibration and lip motion with interpolated GMM classifiers. *Journal of Vibroengineering* 16, 5 (2014), 2572–2581.
- [54] Matthias Dorfer, Jan jr. Hajić, Andreas Arzt, Harald Frostel, and Gerhard Widmer. 2018. Learning Audio-Sheet Music Correspondences for Cross-Modal Retrieval and Piece Identification. *Transactions of the International Society for Music Information Retrieval* 1, 1 (2018), 22–33.
- [55] Matthew J. Dovey. 2004. Overview of the OMRAS project: Online music retrieval and searching. *Journal of the American Society for Information Science and Technology* 55, 12 (2004), 1100–1107.
- [56] Hoda M. Fahmy and Dorothea Blostein. 1993. A graph grammar programming style for recognition of music notation. *Machine Vision and Applications* 6, 2 (1993), 83–99.
- [57] Jonathan Feist. 2017. *Berklee Contemporary Music Notation*. Berklee Press.



- [58] Alicia Fornés and Lamiroy Bart (Eds.). 2018. *Graphics Recognition, Current Trends and Evolutions*. Lecture Notes in Computer Science, Vol. 11009. Springer International Publishing.
- [59] Alicia Fornés, Anjan Dutta, Albert Gordo, and Josep Lladós. 2011. The ICDAR 2011 Music Scores Competition: Staff Removal and Writer Identification. In *International Conference on Document Analysis and Recognition*. 1511–1515.
- [60] Alicia Fornés, Anjan Dutta, Albert Gordo, and Josep Lladós. 2012. CVC-MUSCIMA: A Ground-truth of Handwritten Music Score Images for Writer Identification and Staff Removal. *International Journal on Document Analysis and Recognition* 15, 3 (2012), 243–251.
- [61] Alicia Fornés, Josep Lladós, and Gemma Sánchez. 2006. Primitive Segmentation in Old Handwritten Music Scores. In *Graphics Recognition. Ten Years Review and Future Perspectives*. Berlin, Heidelberg, 279–290.
- [62] Alicia Fornés, Josep Lladós, and Gemma Sánchez. 2008. Old Handwritten Musical Symbol Classification by a Dynamic Time Warping Based Method. In *Graphics Recognition. Recent Advances and New Opportunities*. Berlin, Heidelberg, 51–60.
- [63] Alicia Fornés, Josep Lladós, Gemma Sánchez, and Horst Bunke. 2008. Writer Identification in Old Handwritten Music Scores. In *8th International Workshop on Document Analysis Systems*. Nara, Japan, 347–353.
- [64] Alicia Fornés, Josep Lladós, Gemma Sánchez, and Horst Bunke. 2009. On the Use of Textural Features for Writer Identification in Old Handwritten Music Scores. *10th International Conference on Document Analysis and Recognition* (2009), 996–1000.
- [65] Stavroula-Evita Fotinea, George Giakoupis, Aggelos Livens, Stylianos Bakamidis, and George Carayannis. 2000. An Optical Notation Recognition System for Printed Music Based on Template Matching and High Level Reasoning. In *RIA0 '00 Content-Based Multimedia Information Access*. Paris, France, 1006–1014.
- [66] Christian Fremerey, Meinard Müller, Frank Kurth, and Michael Clausen. 2008. Automatic Mapping of Scanned Sheet Music to Audio Recordings. In *9th International Conference on Music Information Retrieval*. 413–418.
- [67] Ichiro Fujinaga. 1988. *Optical Music Recognition using Projections*. Master's thesis. McGill University.
- [68] Ichiro Fujinaga and Andrew Hankinson. 2014. SIMSSA: Single Interface for Music Score Searching and Analysis. *Journal of the Japanese Society for Sonic Arts* 6, 3 (2014), 25–30.
- [69] Ichiro Fujinaga, Andrew Hankinson, and Julie E. Cumming. 2014. Introduction to SIMSSA (Single Interface for Music Score Searching and Analysis). In *1st International Workshop on Digital Libraries for Musicology*. ACM, 1–3.
- [70] Antonio-Javier Gallego and Jorge Calvo-Zaragoza. 2017. Staff-line removal with selectional auto-encoders. *Expert Systems with Applications* 89 (2017), 138–148.
- [71] Gear Up AB. 2017. iSeeNotes. <http://www.iseenotes.com/>
- [72] Susan E. George. 2003. Online Pen-Based Recognition of Music Notation with Artificial Neural Networks. *Computer Music Journal* 27, 2 (2003), 70–79.
- [73] Susan E. George. 2004. Wavelets for Dealing with Super-Imposed Objects in Recognition of Music Notation. In *Visual Perception of Music Notation: On-Line and Off Line Recognition*. IRM Press, Hershey, PA, 78–107.
- [74] Angelos P. Giotis, Giorgos Sfikas, Basilis Gatos, and Christophoros Nikou. 2017. A survey of document image word spotting techniques. *Pattern Recognition* 68 (2017), 310–332.
- [75] Michael Good. 2001. *MusicXML: An Internet-Friendly Format for Sheet Music*. Technical Report. Recordare LLC.
- [76] Michael Good and Geri Actor. 2003. Using MusicXML for file interchange. In *Third International Conference on WEB Delivering of Music*. 153.
- [77] Albert Gordo, Alicia Fornés, and Ernest Valveny. 2013. Writer identification in handwritten musical scores with bags of notes. *Pattern Recognition* 46, 5 (2013), 1337–1345.
- [78] Mark Gotham, Peter Jonas, Bruno Bower, William Bosworth, Daniel Rootham, and Leigh VanHandel. 2018. Scores of Scores: An Openscore Project to Encode and Share Sheet Music. In *5th International Conference on Digital Libraries for Musicology*. Paris, France, 87–95.
- [79] Elaine Gould. 2011. *Behind Bars*. Faber Music.
- [80] Gianmarco Gozzi. 2010. *OMRjX: A framework for piano scores optical music recognition*. Master's thesis. Politecnico di Milano.
- [81] Jan jr. Hajič. 2018. A Case for Intrinsic Evaluation of Optical Music Recognition. In *1st International Workshop on Reading Music Systems*. Paris, France, 15–16.
- [82] Jan jr. Hajič, Matthias Dorfer, Gerhard Widmer, and Pavel Pecina. 2018. Towards Full-Pipeline Handwritten OMR with Musical Symbol Detection by U-Nets. In *19th International Society for Music Information Retrieval Conference*. Paris, France, 225–232.
- [83] Jan jr. Hajič, Marta Kolárová, Alexander Pacha, and Jorge Calvo-Zaragoza. 2018. How Current Optical Music Recognition Systems Are Becoming Useful for Digital Libraries. In *5th International Conference on Digital Libraries for Musicology*. Paris, France, 57–61.
- [84] Jan jr. Hajič, Jiří Novotný, Pavel Pecina, and Jaroslav Pokorný. 2016. Further Steps towards a Standard Testbed for Optical Music Recognition. In *17th International Society for Music Information Retrieval Conference*. New York

- University, New York, USA, 157–163.
- [85] Jan jr. Hajič and Pavel Pecina. 2017. Groundtruthing (Not Only) Music Notation with MUSICMarker: A Practical Overview. In *14th International Conference on Document Analysis and Recognition*. Kyoto, Japan, 47–48.
- [86] Jan jr. Hajič and Pavel Pecina. 2017. The MUSCIMA++ Dataset for Handwritten Optical Music Recognition. In *14th International Conference on Document Analysis and Recognition*. Kyoto, Japan, 39–46.
- [87] Donna Harman. 2011. *Information Retrieval Evaluation* (1st ed.). Morgan & Claypool Publishers.
- [88] Kate Helsen, Jennifer Bain, Ichiro Fujinaga, Andrew Hankinson, and Debra Lacoste. 2014. Optical music recognition and manuscript chant sources. *Early Music* 42, 4 (2014), 555–558.
- [89] George Heussenstamm. 1987. *The Norton Manual of Music Notation*. W. W. Norton & Company.
- [90] Władysław Homenda. 1996. Automatic recognition of printed music and its conversion into playable music data. *Control and Cybernetics* 25, 2 (1996), 353–367.
- [91] Yu-Hui Huang, Xuanli Chen, Serafina Beck, David Burn, and Luc Van Gool. 2015. Automatic Handwritten Mensural Notation Interpreter: From Manuscript to MIDI Performance. In *16th International Society for Music Information Retrieval Conference*. Málaga, Spain, 79–85.
- [92] José Manuel Iñesta, Pedro J. Ponce de León, David Rizo, José Oncina, Luisa Micó, Juan Ramón Rico-Juan, Carlos Pérez-Sancho, and Antonio Pertusa. 2018. HISPAMUS: Handwritten Spanish Music Heritage Preservation by Automatic Transcription. In *1st International Workshop on Reading Music Systems*. Paris, France, 17–18.
- [93] Linn Saxrud Johansen. 2009. *Optical Music Recognition*. Master’s thesis. University of Oslo.
- [94] Graham Jones, Bee Ong, Ivan Bruno, and Kia Ng. 2008. Optical music imaging: music document digitisation, recognition, evaluation, and restoration. In *Interactive multimedia music technologies*. IGI Global, 50–79.
- [95] Nicholas Journet, Muriel Visani, Boris Mansencal, Kieu Van-Cuong, and Antoine Billy. 2017. DocCreator: A New Software for Creating Synthetic Ground-Truthed Document Images. *Journal of imaging* 3, 4 (2017), 62.
- [96] Michael Kassler. 1972. Optical Character-Recognition of Printed Music : A Review of Two Dissertations. Automatic Recognition of Sheet Music by Dennis Howard Pruslin ; Computer Pattern Recognition of Standard Engraved Music Notation by David Stewart Prerau. *Perspectives of New Music* 11, 1 (1972), 250–254.
- [97] Klaus Keil and Jennifer A. Ward. 2017. Applications of RISM data in digital libraries and digital musicology. *International Journal on Digital Libraries* (2017).
- [98] Daniel Lopresti and George Nagy. 2002. *Issues in Ground-Truthing Graphic Documents*. Springer Berlin Heidelberg, Ontario, Canada, 46–67.
- [99] Nawapon Luangnapa, Thongchai Silpavarangkura, Chakarida Nukoolkit, and Pornchai Mongkolnam. 2012. Optical music recognition on android platform. In *International Conference on Advances in Information Technology*. Springer, 106–115.
- [100] Rakesh Malik, Partha Pratim Roy, Umapada Pal, and Fumitaka Kimura. 2013. Handwritten Musical Document Retrieval Using Music-Score Spotting. In *12th International Conference on Document Analysis and Recognition*. 832–836.
- [101] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.
- [102] T. Matsushima, I. Sonomoto, T. Harada, K. Kanamori, and S. Ohteru. 1985. Automated high speed recognition of printed music (WABOT-2 vision system). In *International Conference on Advanced Robotics*. 477–482.
- [103] Johann Mattheson. 1739. *Der vollkommene Capellmeister*. Herold, Christian, Hamburg.
- [104] Apurva A. Mehta and Malay S. Bhatt. 2015. Optical Music Notes Recognition for Printed Piano Music Score Sheet. In *International Conference on Computer Communication and Informatics*. Coimbatore, India.
- [105] Hidetoshi Miyao and Robert Martin Haralick. 2000. Format of ground truth data used in the evaluation of the results of an optical music recognition system. In *4th International Workshop on Document Analysis Systems*. Brasil, 497–506.
- [106] Musitek. 2017. SmartScore X2. <http://www.musitek.com/smartscore-pro.html>
- [107] Neuratron. 2015. NotateMe. <http://www.neuratron.com/notateme.html>
- [108] Neuratron. 2018. PhotoScore 2018. <http://www.neuratron.com/photoscore.htm>
- [109] Kia Ng, Alex McLean, and Alan Marsden. 2014. Big data optical music recognition with multi images and multi recognisers. In *EVA London 2014 on Electronic Visualisation and the Arts*. BCS, 215–218.
- [110] Tam Nguyen and Gueesang Lee. 2015. A Lightweight and Effective Music Score Recognition on Mobile Phones. *Journal of Information Processing Systems* 11, 3 (2015), 438–449.
- [111] Jiri Novotný and Jaroslav Pokorný. 2015. Introduction to Optical Music Recognition: Overview and Practical Challenges. In *Annual International Workshop on Databases, TExts, Specifications and Objects*. 65–76.
- [112] Organum. 2016. PlayScore. <http://www.playscore.co/>
- [113] Rafael Ornes. 1998. Choral Public Domain Library. <http://cpdl.org>
- [114] Tuula Pääkkönen, Jukka Kervinen, and Kimmo Kettunen. 2018. Digitisation and Digital Library Presentation System – Sheet Music to the Mix. In *1st International Workshop on Reading Music Systems*. Paris, France, 21–22.



- [115] Alexander Pacha. 2018. Advancing OMR as a Community: Best Practices for Reproducible Research. In *1st International Workshop on Reading Music Systems*. Paris, France, 19–20.
- [116] Alexander Pacha and Jorge Calvo-Zaragoza. 2018. Optical Music Recognition in Mensural Notation with Region-Based Convolutional Neural Networks. In *19th International Society for Music Information Retrieval Conference*. Paris, France, 240–247.
- [117] Alexander Pacha, Kwon-Young Choi, Bertrand Couïason, Yann Riquebourg, Richard Zanibbi, and Horst Eidenberger. 2018. Handwritten Music Object Detection: Open Issues and Baseline Results. In *13th International Workshop on Document Analysis Systems*. 163–168.
- [118] Alexander Pacha and Horst Eidenberger. 2017. Towards a Universal Music Symbol Classifier. In *14th International Conference on Document Analysis and Recognition*. IAPR TC10 (Technical Committee on Graphics Recognition), Kyoto, Japan, 35–36.
- [119] Alexander Pacha, Jan jr. Hajič, and Jorge Calvo-Zaragoza. 2018. A Baseline for General Music Object Detection with Deep Learning. *Applied Sciences* 8, 9 (2018), 1488–1508.
- [120] Victor Padilla, Alan Marsden, Alex McLean, and Kia Ng. 2014. Improving OMR for Digital Music Libraries with Multiple Recognisers and Multiple Sources. In *1st International Workshop on Digital Libraries for Musicology*. London, United Kingdom, 1–8.
- [121] Emilia Parada-Cabaleiro, Anton Batliner, Alice Baird, and Björn Schuller. 2017. The SEILS dataset: Symbolically Encoded Scores in ModernAncient Notation for Computational Musicology. In *18th International Society for Music Information Retrieval Conference*. Suzhou, China.
- [122] Viet-Khoi Pham, Hai-Dang Nguyen, and Minh-Triet Tran. 2015. Virtual Music Teacher for New Music Learners with Optical Music Recognition. In *International Conference on Learning and Collaboration Technologies*. Springer, 415–426.
- [123] David S. Prerau. 1971. Computer pattern recognition of printed music. In *Fall Joint Computer Conference*. 153–162.
- [124] Gérard Presgurvic. 2005. Songbook Romeo & Julia. <https://www.musicalvienna.at/de/souvenirs/12/ANDERE-MUSICALS/10/Songbook-Romeo-und-Julia>
- [125] Project Petrucci LLC. 2006. International Music Score Library Project. <http://imslp.org/>
- [126] Denis Pruslin. 1966. *Automatic recognition of sheet music*. Ph.D. Dissertation. Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.
- [127] Laurent Pugin. 2006. Optical Music Recognition of Early Typographic Prints using Hidden Markov Models. In *7th International Conference on Music Information Retrieval*. Victoria, Canada, 53–56.
- [128] Laurent Pugin, John Ashley Burgoyne, and Ichiro Fujinaga. 2007. Reducing Costs for Digitising Early Music with Dynamic Adaptation. In *Research and Advanced Technology for Digital Libraries*. Berlin, Heidelberg, 471–474.
- [129] Laurent Pugin and Tim Crawford. 2013. Evaluating OMR on the Early Music Online Collection. In *14th International Society for Music Information Retrieval Conference*. Curitiba, Brazil, 439–444.
- [130] Gene Ragan. 2017. KompApp. <http://kompapp.com/>
- [131] Sheikh Faisal Rashid, Abdullah Akmal, Muhammad Adnan, Ali Adnan Aslam, and Andreas Dengel. 2017. Table Recognition in Heterogeneous Documents Using Machine Learning. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. 777–782.
- [132] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R.S. Marcal, Carlos Guedes, and Jamie dos Santos Cardoso. 2012. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval* 1, 3 (2012), 173–190.
- [133] Pau Riba, Alicia Fornés, and Josep Lladós. 2017. Towards the alignment of handwritten music scores. In *Graphic Recognition. Current Trends and Challenges (Lecture Notes in Computer Science)*. 103–116.
- [134] Adrià Rico Blanes and Alicia Fornés Bisquerra. 2017. Camera-Based Optical Music Recognition Using a Convolutional Neural Network. In *14th International Conference on Document Analysis and Recognition*. IEEE, Kyoto, Japan, 27–28.
- [135] David Rizo, Jorge Calvo-Zaragoza, and José M. Iñesta. 2018. MuRET: A Music Recognition, Encoding, and Transcription Tool. In *5th International Conference on Digital Libraries for Musicology*. Paris, France, 52–56.
- [136] Heinz Roggenkemper and Ryan Roggenkemper. 2018. How can Machine Learning make Optical Music Recognition more relevant for practicing musicians?. In *1st International Workshop on Reading Music Systems*. Paris, France, 25–26.
- [137] Perry Roland. 2002. The music encoding initiative (MEI). In *1st International Conference on Musical Applications Using XML*. 55–59.
- [138] Florence Rossant and Isabelle Bloch. 2004. A fuzzy model for optical recognition of musical scores. *Fuzzy Sets and Systems* 141, 2 (2004), 165–201.
- [139] Partha Pratim Roy, Ayan Kumar Bhunia, and Umapada Pal. 2017. HMM-based writer identification in music score documents without staff-line removal. *Expert Systems with Applications* 89 (2017), 222–240.
- [140] Sächsische Landesbibliothek. 2007. Staats- und Universitätsbibliothek Dresden. <https://www.slub-dresden.de>
- [141] Charalampos Saitis, Andrew Hankinson, and Ichiro Fujinaga. 2014. Correcting Large-Scale OMR Data with Crowdsourcing. In *1st International Workshop on Digital Libraries for Musicology*. ACM, 1–3.

- [142] Zeyad Saleh, Ke Zhang, Jorge Calvo-Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. 2017. Pixel.js: Web-Based Pixel Classification Correction Platform for Ground Truth Creation. In *14th International Conference on Document Analysis and Recognition*. Kyoto, Japan, 39–40.
- [143] Eleanor Selfridge-Field. 1997. *Beyond MIDI: The Handbook of Musical Codes*. MIT Press, Cambridge, MA, USA.
- [144] Asif Shahab, Faisal Shafait, Thomas Kieninger, and Andreas Dengel. 2010. An Open Approach Towards the Benchmarking of Table Structure Recognition Systems. In *9th International Workshop on Document Analysis Systems*. Boston, Massachusetts, USA, 113–120.
- [145] Muhammad Sharif, Quratul-Ain Arshad, Mudassar Raza, and Wazir Zada Khan. 2009. [COMSCAN]: An Optical Music Recognition System. In *7th International Conference on Frontiers of Information Technology*. ACM, 34.
- [146] Baoguang Shi, Xiang Bai, and Cong Yao. 2017. An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 11 (2017), 2298–2304.
- [147] Mahmood Sotoodeh, Farshad Tajeripour, Sadegh Teimori, and Kirk Jorgensen. 2017. A music symbols recognition method using pattern matching along with integrated projection and morphological operation techniques. *Multimedia Tools and Applications* (2017).
- [148] Daniel Spreadbury and Robert Piéchaud. 2015. Standard Music Font Layout (SMuFL). In *First International Conference on Technologies for Music Notation and Representation - TENOR2015*. Paris, France, 146–153.
- [149] StaffPad Ltd. 2017. StaffPad. <http://www.staffpad.net/>
- [150] Gabriel Taubman. 2005. *MusicHand : A Handwritten Music Recognition System*. Technical Report. Brown University.
- [151] Jessica Thompson, Andrew Hankinson, and Ichiro Fujinaga. 2011. Searching the Liber Usualis: Using CouchDB and ElasticSearch to Query Graphical Music Documents. In *12th International Society for Music Information Retrieval Conference*.
- [152] Lukas Tuggener, Isamil Elezi, Jürgen Schmidhuber, Marcello Pelillo, and Stadelmann Thilo. 2018. DeepScores - A Dataset for Segmentation, Detection and Classification of Tiny Objects. In *24th International Conference on Pattern Recognition*. Beijing, China.
- [153] Julián Urbano. 2013. *MIREX 2013 Symbolic Melodic Similarity: A Geometric Model supported with Hybrid Sequence Alignment*. Technical Report. Music Information Retrieval Evaluation eXchange.
- [154] Julián Urbano, Juan Lloréns, Jorge Morato, and Sonia Sánchez-Cuadrado. 2010. *MIREX 2010 Symbolic Melodic Similarity: Local Alignment with Geometric Representations*. Technical Report. Music Information Retrieval Evaluation eXchange.
- [155] Julián Urbano, Juan Lloréns, Jorge Morato, and Sonia Sánchez-Cuadrado. 2011. *MIREX 2011 Symbolic Melodic Similarity: Sequence Alignment with Geometric Representations*. Technical Report. Music Information Retrieval Evaluation eXchange.
- [156] Julián Urbano, Juan Lloréns, Jorge Morato, and Sonia Sánchez-Cuadrado. 2012. *MIREX 2012 Symbolic Melodic Similarity: Hybrid Sequence Alignment with Geometric Representations*. Technical Report. Music Information Retrieval Evaluation eXchange.
- [157] Eelco van der Wel and Karen Ullrich. 2017. Optical Music Recognition with Convolutional Sequence-to-Sequence Models. In *18th International Society for Music Information Retrieval Conference*. Suzhou, China.
- [158] Gabriel Vigliensoni, John Ashley Burgoyne, Andrew Hankinson, and Ichiro Fujinaga. 2011. Automatic Pitch Detection in Printed Square Notation. In *12th International Society for Music Information Retrieval Conference*. Miami, Florida, 423–428.
- [159] Gabriel Vigliensoni, Jorge Calvo-Zaragoza, and Ichiro Fujinaga. 2018. Developing an environment for teaching computers to read music. In *1st International Workshop on Reading Music Systems*. Paris, France, 27–28.
- [160] Quang Nhat Vo, Guee Sang Lee, Soo Hyung Kim, and Hyung Jeong Yang. 2017. Recognition of Music Scores with Non-Linear Distortions in Mobile Devices. *Multimedia Tools and Applications* (2017).
- [161] Matthias Wallner. 2014. *A System for Optical Music Recognition and Audio Synthesis*. Master’s thesis. TU Wien.
- [162] Gus G. Xia and Roger B. Dannenberg. 2017. Improvised Duet Interaction: Learning Improvisation Techniques for Automatic Accompaniment. In *New Interfaces for Musical Expression*. Aalborg University Copenhagen, Denmark.
- [163] Jianshu Zhang, Jun Du, Shiliang Zhang, Dan Liu, Yulong Hu, Jinshui Hu, Si Wei, and Lirong Dai. 2017. Watch, Attend and Parse: An End-to-end Neural Network Based Approach to Handwritten Mathematical Expression Recognition. *Pattern Recognition* (2017).

## 6.2 The MUSCIMA++ Dataset for Handwritten Optical Music Recognition

Jan Hajič jr. and Pavel Pecina. The MUSCIMA++ Dataset for Handwritten Optical Music Recognition. *14th International Conference on Document Analysis and Recognition*, pages 39–46, Kyoto, Japan, 2017. ISBN 978-1-5386-3586-5, ISSN 2379-2140, doi: 10.1109/ICDAR.2017.16.

The article *The MUSCIMA++ Dataset for Handwritten Optical Music Recognition* introduces the first dataset for full-pipeline OMR, and most importantly introduces the key concept of the *Music Notation Graph* (MuNG), an adequate and learnable representation that allows formulating the latter half of the OMR pipeline as a (1) relatively simple machine learning problem, (2) deterministic exploitation of the rules of music notation.

The thesis author designed the MuNG representation, managed the annotation process and wrote the text of the article. The co-author Pavel Pecina contributed to the final text of the article with his comments. The contribution of the dissertation author is about 95% of the article.

# The MUSCIMA++ Dataset for Handwritten Optical Music Recognition

Jan Hajic Jr.

Institute of Formal and Applied Linguistics  
Charles University  
Email: hajicj@ufal.mff.cuni.cz

Pavel Pecina

Institute of Formal and Applied Linguistics  
Charles University  
Email: pecina@ufal.mff.cuni.cz

**Abstract**—Optical Music Recognition (OMR) promises to make accessible the content of large amounts of musical documents, an important component of cultural heritage. However, the field does not have an adequate dataset and ground truth for benchmarking OMR systems, which has been a major obstacle to measurable progress. Furthermore, machine learning methods for OMR require training data. We design and collect MUSCIMA++, a new dataset for OMR. Ground truth in MUSCIMA++ is a *notation graph*, which our analysis shows to be a necessary and sufficient representation of music notation. Building on the CVC-MUSCIMA dataset for staffline removal, the MUSCIMA++ dataset v1.0 consists of 140 pages of handwritten music, with 91254 manually annotated notation symbols and 82247 explicitly marked relationships between symbol pairs. The dataset allows training and directly evaluating models for symbol classification, symbol localization, and notation graph assembly, and musical content extraction, both in isolation and jointly. Open-source tools are provided for manipulating the dataset, visualizing the data and annotating more, and the data is made available under an open license.

## I. INTRODUCTION: WHAT DATASET?

Optical Music Recognition (OMR) is a field of document analysis that aims to automatically read music. Music notation encodes musical information in a graphical form; OMR backtracks through this process to extract the musical information from its graphical representation. OMR can be likened to OCR for the music notation writing system; however, it is more difficult [1], and remains an open problem [2], [3]. The intricacies of Common western music notation (CWMN<sup>1</sup>) have been thoroughly discussed since early attempts at OMR, notably by Byrd [4], [5].

One of the most persistent hindrances to OMR progress is a **lack of datasets**. These are necessary to provide ground truth for evaluating OMR systems [1], [5]–[8], to enable fair, replicable comparison among academic and commercial systems. Furthermore, especially for handwritten notation, supervised machine learning methods have often been used that require training data [9]–[12].

We use the term *dataset* in the following sense:  $\mathcal{D} = \langle (x_i, y_i) \forall i = 1 \dots n \rangle$ . Given a set of inputs  $x_i$  (in our case, images of sheet music), the dataset records the desired outputs

<sup>1</sup>We assume the reader is familiar with CWMN. In case a refresher is needed, we recommend chapter 2 of “Music Notation by Computer” [4], by Donald Byrd. A comprehensive list of music notation terminology is maintained on Wikipedia: [https://en.wikipedia.org/wiki/List\\_of\\_musical\\_symbols](https://en.wikipedia.org/wiki/List_of_musical_symbols)

$y_i$  – ground truth. The quality of OMR systems can then be measured by how closely they approximate the ground truth, although defining this approximation for the variety of representations of music is very much an open problem [2], [5]–[7], [13].

For printed music notation, the lack of datasets can be bypassed by generating music in representations such as LilyPond<sup>2</sup> or MEL<sup>3</sup> and capturing intermediate steps of the rendering process. However, for handwritten music, no satisfactory synthetic data generator exists so far, and an extensive annotation effort cannot be avoided. Therefore, to best utilize our resources available for creating a dataset, we create a dataset of **handwritten notation**.

To build a dataset of handwritten music, we need to decide:

- What should the ground truth  $y_i$  be for an image  $x_i$ ?
- What sheet music do we choose as data points?

The definition of ground truth must **reflect what OMR does**. Miyao and Haralick [14] group OMR applications into two broad groups: those that require replayability, and those that need reprintability. *Replayability* entails recovering pitches and durations of individual notes and organizing them in time by note onset. *Reprintability* is the ability to take OMR results as the input to music typesetting software and obtain a result that encodes this music in the same way as it was encoded in the input sequence. Reprintability implies replayability, but not vice versa, as one musical sequence can be encoded by different musical scores; e.g. MIDI is a good representation for replayability, but not reprintability (see Fig. 1).

The selection of musical score images in the dataset should **cover the known “dimensions of difficulty”** [5], to allow for assessing OMR systems with respect to increasingly complex inputs.

In the rest of the article, we reason what the ground truth for OMR should be (II-A) and what kinds of musical score images the dataset should contain (II-B); we scavenge existing OMR datasets for work already done that would satisfy these design choices (III); finally, we describe the MUSCIMA++ dataset (IV), establish simple baselines (V); and provide some concluding remarks (VI).

The main contributions of this work are:

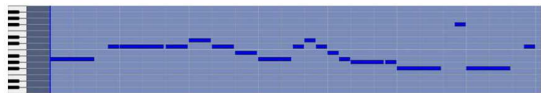
<sup>2</sup><http://www.lilypond.org>

<sup>3</sup><http://www.music-encoding.org>





(a) Input: manuscript image.



(b) Replayable output: pitches, durations, onsets. Time is the horizontal axis, pitch is the vertical axis. This visualization is called a *piano roll*.



(c) Reprintable output: re-typesetting.



(d) Reprintable output: same music expressed differently

Fig. 1: OMR for replayability and reprintability. The input (a) encodes the sequence of pitches, durations, and onsets (b), which can be expressed in different ways (c, d).

- MUSCIMA++<sup>4</sup> – an extensive dataset of handwritten musical symbols and their relationships,<sup>5</sup>
- A *notation graph* ground truth definition and implementation that de-couples the graphical expression of music and musical semantics, while recording sufficient information to bridge this gap, and also helps understanding the problem space of OMR;
- Open-source tools for processing the data including inferring pitches and durations, visualizing it, and annotating more.

MUSCIMA++ enables training and evaluating models for symbol localization, classification, and arguably its most innovative aspect for OMR is that it enables directly solving music notation reconstruction, in a way that explicitly considers the need to infer musical semantics.

## II. DESIGNING A DATASET FOR OMR

In this section, we discuss the key design concerns introduced above: an appropriate ground truth for OMR, and the choice of data.

### A. Ground Truth

The ground truth over a dataset is the desired output of a system solving a task. Therefore, in order to design the ground truth for the dataset, we need to understand how OMR can be expressed in terms of inputs and outputs. OMR solutions are usually pipelines with four major stages [1], [2]:

- 1) Image preprocessing: enhancement, binarization, scaling;

<sup>4</sup>Standing for MUsic SCore IMAges, credit for abbreviation to [15]

<sup>5</sup>Available from: <http://hdl.handle.net/11372/LRT-2372>

TABLE I: OMR Pipeline as inputs and outputs

Sub-task	Input	Output
Image Processing	Score image	“Cleaned” image
Binarization	“Cleaned” image	Binary image
Staff ID & removal	Binary image	Stafflines list
Symbol localization	(Staff-less) image	Symbol regions
Symbol classification	Symbol regions	Symbol labels
Notation assembly	Symbol regs. & labels	<b>Notation graph</b>
Infer pitch/duration	Notation graph	Pitch/duration attrs.
Output conversion	Notation graph + attrs.	MusicXML, MIDI, ...

- 2) Music symbol recognition: staffline identification and removal, localization and classification of other symbols;
- 3) Musical notation reconstruction: recovering the logical structure of the score;
- 4) Final representation construction: depending on the output requirements, usually inferring pitch and duration (MusicXML, MEI, MIDI, LilyPond, etc.).

The key problems of OMR reside in stages 2 and 3: finding individual musical symbols on the page, and recovering their relationships. The inputs and outputs of the individual pipeline stages and sub-tasks is summarized in Table I. While end-to-end OMR that bypasses some sections of this pipeline is an attractive option (see [16]), these should still be compared against more orthodox solutions.

The input of **music symbol recognition** is a “cleaned” and usually binarized image. The output of this stage is a list of musical symbols recording their locations on the page, and their types (e.g., c-clef, beam, sharp). Usually, there are three sub-tasks: staffline identification and removal, symbol localization (in binary images, synonymous with foreground segmentation), and symbol classification [2]. Stafflines are typically handled as a separate step [17], due to them being rather a layout element than a character-like symbol.

In turn, the list of locations and classes of symbols on the page is the input to the **music notation reconstruction** stage. At this stage, it is necessary to recover the *relationships* among the individual musical symbols. These relationships enable inferring the “musical content” (most importantly, pitch and duration information – what to play, and when): there is a 1:1 relationship between a notehead notation primitive and a note musical object, of which pitch and duration are properties, and the other symbols that relate – directly or indirectly – to a notehead, such as stems, stafflines, beams, accidentals, or clefs, inform the reader’s decision to assign the pitch and duration.

The result of OMR stage 3 naturally forms a graph. The symbols from the previous stage become vertices of the graph, with the symbol classes and locations being the vertex attributes, and the relationships between symbols assume the role of edges. Graphs have been explicitly used for assembly of music notation primitives e.g. by [18], [19], and grammar-based approaches (e.g., [20]–[23]) lend themselves to a graph representation as well, by recording the parse tree(s). An example of symbol recognition and notation reconstruction



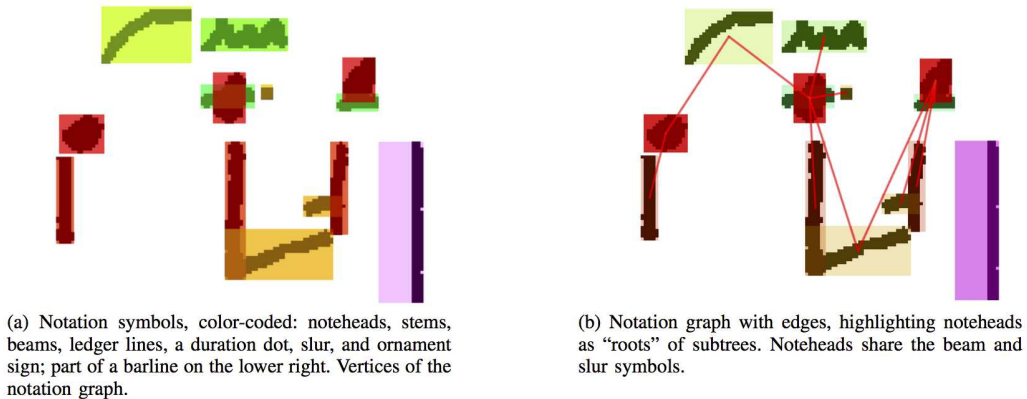


Fig. 2: Visualizing the list of symbols and the notation graph over staff removal output. The notation graph in (b) allows unambiguously inferring pitch and duration (stafflines removed for clarity, although for encoding pitch, we would need to establish the relationship of the noteheads to stafflines).

output over the same snippet of a musical score is given in Figure 2.

A key observation for ground truth design is that **the notation graph records information both necessary and sufficient for both replayability and reprintability**, and thus makes a good ground truth for an OMR dataset.

1) *Necessary*: Before the notation graph is constructed in stage 3, there is not enough information extracted for the output to be either replayable or reprintable. No finite alphabet can be designed so that its symbols could be interpreted in isolation: recognizing a note is not enough to determine its pitch: one needs it to relate to the stafflines, clefs, key signatures, etc.

2) *Sufficient*: The process of reading musical scores is such that stage 3 output is the point where the OMR system has extracted *all* the useful information – signal – from the input image, resolving all *ambiguities*; the system is therefore properly “free” to forget about the input image. All that remains in order to project the written page to the corresponding point in the space of musical note<sup>6</sup> configurations in time is to follow the rules for reading music, which can be expressed in terms of querying the graph to infer additional properties of the nodes representing noteheads – essentially, a graph transformation. This implies that creating the desired representation in stage 4 is only a technical task: implementing conversion to the desired output format (which can nevertheless still be a very complex piece of software).<sup>7</sup> This observation also implies that

<sup>6</sup>A musical note object, as opposed to the written note, is characterized in music theory by four attributes: pitch, duration, loudness, and timbre, of which OMR needs to recover pitch and duration; the musical score additionally encodes the onsets of notes in musical time.

<sup>7</sup>The representation used to record the dataset is not necessarily best for experiments – but experiment-specific output representations (such as a MIDI file for replayability-only experiments) are *unambiguously obtainable* from the notation graph.

an OMR system that can recover the notation graph does *not* have to explicitly recover pitch and duration.

### B. Choice of data

The dataset should enable evaluating handwritten OMR with respect to the “challenge space” of OMR. In their state-of-the-art analysis of the difficulties of OMR, Byrd and Simonsen [5] identify three axes along which musical score images become less or more challenging inputs for an OMR system: *Notation complexity*, *Image quality*, and *Tightness of spacing*.

The dataset should also contain a wide variety of musical *symbols*, including less frequent items such as tremolos or glissandi, to enable differentiating systems also according to the breath of their vocabulary.

The axis of **notation complexity** is structured by [5] into four levels. Level 1, single-staff single-voice music, tests an “OMR minimum”: the recognition of individual symbols for a single sequence of notes. Level 2, single-staff multi-voice music, tests the ability to deal with multiple sequences of notes in parallel, so e.g. rhythmical constraints based on time signatures [24] are harder to use. Level 3, multi-staff single-voice music, tests high-level segmentation into systems and staves. Level 4, pianoform music, then presents the most complex, combined challenge, as piano music has exploited the rules of CWMN to their fullest [5] and sometimes beyond. The dataset should contain a choice of musical scores representing all these levels.

On the other hand, difficulties relating to **image quality** – deformations, noise, and document degradations – do not have to be represented in the dataset. Their descriptions in [5] essentially define how to simulate them; many morphological distortions have already been implemented for staff removal data [15], [25].



The **tightness of spacing** in [5] refers to default horizontal and vertical distances between symbols.<sup>8</sup> As spacing tightens, assumptions about relative notation spacing may cease to hold: Byrd and Simonsen give an example where the augmentation dot of a preceding note can be easily confused with a staccato dot of its following note (see [5], Fig. 21). In handwritten music, variability in spacing is superseded by the variability of handwriting itself. Handwritten music gives no topological guarantees: by definition straight lines, such as stems, become curved, noteheads and stems do not touch, accidentals and noteheads *do* touch, etc. – see Fig. 3. The various styles of handwriting should be represented in the dataset as broadly as possible.

### III. EXISTING DATASETS

We describe the available datasets and discuss how they correspond to the requirements of Section II. Reviewing Table I, the subtasks at stages 2 and 3 of the OMR pipeline are (a) staffline removal, (b) symbol localization, (c) symbol classification, and (d) symbol assembly.

For **staff removal** in handwritten music, the premier dataset is CVC-MUSCIMA [15], consisting of 1000 handwritten scores (20 pages of music, each copied by hand by 50 musicians). The state-of-the-art for staff removal has been established with a competition using CVC-MUSCIMA [17]. The dataset fulfills the requirements for a good choice of data: the 20 pages include scores of all 4 levels of complexity, and a wide array of music notation symbols (including tremolos, glissandi, grace notes, or trills), and handwriting style varies greatly among the 50 writers, including topological inconsistencies, as illustrated in Fig. 3. Importantly, CVC-MUSCIMA is freely available for download under a CC-BY-NC-SA 4.0 license.<sup>9</sup>

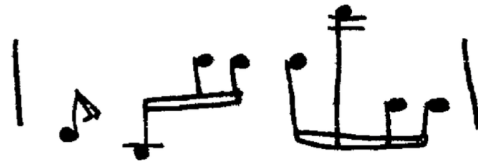
The most extensive dataset for handwritten **symbol classification** is the HOMUS dataset of Calvo-Zaragoza and Oncina [11], which provides 15200 handwritten musical symbols (100 writers, 32 symbol classes, and 4 versions of a symbol per writer per class, with 8 for note-type symbols). HOMUS data is recorded from a touchscreen device, so it can be used for online as well as offline recognition. However, the dataset only contains isolated symbols, not their positions on a page. While it might be possible to synthesize handwritten music pages from the HOMUS symbols, such a synthetic dataset will be rather limited, as HOMUS does not contain beamed groups and chords. For **symbol localization**, we are only aware of a dataset of 3222 handwritten symbols by Rebelo et al. [26], and for **notation reconstruction**, we are not aware of a dataset that provides ground truth for recovering the relationships among handwritten musical symbols.

### IV. THE MUSCIMA++ DATASET

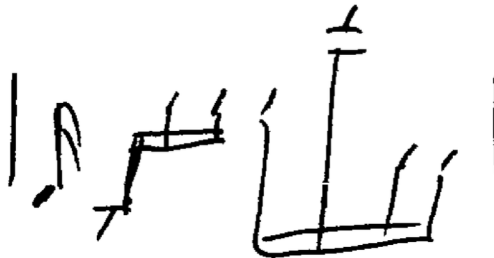
Our main source of musical score images is the CVC-MUSCIMA dataset, described in subsection III. The annotator

<sup>8</sup>We find *adherence to topological standards* to be a more general term that describes this particular class of difficulties.

<sup>9</sup>[http://www.cvc.uab.es/cvcmuscima/index\\_database.html](http://www.cvc.uab.es/cvcmuscima/index_database.html)



(a) Writer 9: beamed groups, nice handwriting.



(b) Writer 22: Disjoint primitives and deformed noteheads. Some noteheads will be very hard to distinguish from the stem.

Fig. 3: Variety of handwriting styles in CVC-MUSCIMA.

team consisted of three professional and four advanced amateur musicians. Each annotator marked one of the 50 versions for each of the 20 CVC-MUSCIMA pages. We selected the 140 out of 1000 pages of CVC-MUSCIMA so that all of the 50 writers are represented as equally as possible: 2 or 3 pages are annotated from each writer, thus fulfilling the same choice-of-data requirements (notation complexity, handwriting style) as CVC-MUSCIMA itself.

There is a total of 91254 symbols (excluding staff objects, which are already given in the CVC-MUSCIMA ground truth) marked in the 140 annotated pages of music, of 107 distinct symbol classes. There are 82247 relationships between pairs of symbols. The total number of *notes* encoded in the dataset is 23349. The set of symbol classes consists of both notation primitives, such as noteheads or beams, and higher-level notation objects, such as key signatures or time signatures. (Given the decomposition of notes into primitives, the equivalent number in terms of HOMUS symbols would be  $\approx 57\,000$ .) The choice of symbols and relationship policies is described in subsec. IV-A. The frequencies of the most significant symbols are described in Table II.

#### A. MUSCIMA++ ground truth

Our ground truth is a **graph of musical symbols and their relationships**, with unlabeled directed edges.<sup>10</sup> For each vertex (symbol), we annotated:

- its **label** (notehead, sharp, g-clef, etc.),
- its **bounding box** with respect to the image,
- its **mask**: exactly which pixels in the bounding box belong to this symbol.

<sup>10</sup>The complete annotation guidelines detailing what the symbol set is and how to deal with individual notations are available online: <https://muscimarker.readthedocs.io/en/latest/instructions.html>

TABLE II: Symbol frequencies in MUSCIMA++

Symbol	Count	Symbol (cont.)	Count
stem	21416	16th_flag	495
notehead-full	21356	16th_rest	436
ledger_line	6847	g-clef	401
beam	6587	grace-notehead-full	348
thin_barline	3332	f-clef	285
measure_separator	2854	other_text	271
slur	2601	hairpin-decr.	268
8th_flag	2198	repeat-dot	263
duration-dot	2074	tuple	244
sharp	2071	hairpin-cresc.	233
notehead-empty	1648	half_rest	216
staccato-dot	1388	accent	201
8th_rest	1134	other-dot	197
flat	1112	time_signature	192
natural	1089	staff_grouping	191
quarter_rest	804	c-clef	190
tie	704	trill	179
key_signature	695	<i>All letters</i>	<i>4072</i>
dynamics_text	681	<i>All numerals</i>	<i>594</i>

These are a superset of the primitive attributes in [14]. Annotating the mask enables us to build an accurate model of actual symbol shapes.

**We do not define a note symbol.** The concept of a note on paper [6], [11], [26] is ambiguous: they consist of multiple primitives (notehead and stem and beams or flags), but at the same time, multiple notes can *share* these primitives, including noteheads. Furthermore, it is not clear what primitives constitute a note. If we follow musical semantics, should e.g. an accidental be considered a part of the note, because it directly influences its pitch? It is more elegant to annotate *how the “note” musical objects are expressed*, and if need be, use the relationships among the primitives to construct the somewhat arbitrary “note” written symbols when necessary.

Instead of trying to categorize symbols as low- or high-level [5], [6], [13] according to whether they carry semantics or not (which is a dubious proposition: musical semantics arise from *configurations* of symbols, as music notation is mostly a featural writing system, where the individual symbols encode separate well-defined aspects of musical semantics but make very limited sense in isolation), we express the dichotomy through the rules for forming relationships. This leads to “layered” annotation. E.g., a 3/4 time signature is annotated using three symbols: a `numeral_3`, `numeral_4`, and a `time_signature` symbol that has outgoing relationships to both numerals. An example of this structure for is given in Figure 4. We take care to define relationships so that the result is a Directed Acyclic Graph (DAG). There is no theoretical limit on the maximum oriented path length, but in practice, it is rarely longer than 3. We break down symbols that consist of multiple connected components when these components can be used in syntactically valid music notation in different configurations to encode distinct musical semantics: an empty

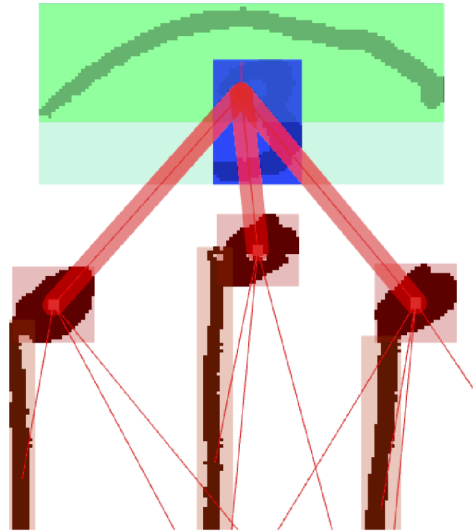


Fig. 4: Two-layer annotation of a triplet. The symbols `numeral_3` (in blue), `tuple_bracket/line`, and the three noteheads that form the triplet are highlighted. The tuple symbol itself, to which the noteheads are connected, is the lighter rectangle encompassing its two components; it has relationships leading to both of them (not highlighted).

notehead may show up with a stem, without one, with multiple stems when two voices share pitch,<sup>11</sup> or it may share stem with others, so we define these as separate symbols. An f-clef dot should not exist without the rest of the clef, and vice versa, so we define the `f-clef` as a single symbol; however, a single repeat may have a variable number of repeat dots, based on how many staves it is spanning, so we define a `repeat-dot` separately.

### B. MUSCIMA++ software tools

In order to make using the dataset easier, we provide two open-source software tools. The `musicma` Python 3 package<sup>12</sup> implements the MUSCIMA++ data model, which can parse the dataset and enables manipulating the data further (such as assembling the related primitives into notes, to provide a comparison to the existing datasets with different symbol sets), and implements extracting pitch, duration and onset data from the notation graph, thus enabling exporting MIDI and thus multimodal OMR experiments, even if so far only on synthesized audio. Second, we provide the `MUSCIMarker` application<sup>13</sup> used for creating the dataset, which can also visualize the data.

### C. Annotation process and quality control

The annotators worked on symbols-only CVC-MUSCIMA images, which allowed for more efficient annotation. The

<sup>11</sup>As seen in page 20 of CVC-MUSCIMA.

<sup>12</sup><https://github.com/hajicj/musicma>

<sup>13</sup><https://github.com/hajicj/MUSCIMarker>



interface used to add symbols consists of two tools: foreground lasso selection, and connected component selection, and our MUSCIMarker software also supports editing the objects’ masks in-place.

After an annotator completed an image, we checked for correctness. Automated validation of the submitted relationships was implemented in MUSCIMarker, however, manual checks and manually correcting mistakes found in auto-validation was still needed, as the validation was just an advisory voice to highlight questionably annotated symbols. After collecting annotations for all 140 images, we performed a second quality control round, this time with further automated checks. We checked for disconnected symbols, and for symbols with suspiciously sparse masks (a symbol was deemed suspicious if more than 7 % of the foreground pixels in its bounding box were not marked as part of any symbol at all). We also fixed other clearly wrong markings (e.g., if a significant amount of stem-only pixels was marked as part of a beam).

The average speed overall was 4.3 symbols per minute, or one per 14 seconds: an average page of about 650 symbols took about  $2\frac{3}{4}$  hours. Annotating the dataset using the process detailed above took roughly 400 hours of work; the “quality control” correctness checks and managing the annotation process took an additional 150. The second, more complete round of quality control took roughly 80 hours.

#### D. Inter-annotator agreement

In order to assess the trustworthiness of the annotations, all annotators were given the same image to annotate, and we measured inter-annotator agreement both before and after quality control (QC) was applied, and we also measured how many changes were made in QC. Given that the expected level of true ambiguity in our ground truth is relatively low, we can interpret disagreement between annotators as evidence of inaccuracies. At the same time, a comparison of annotations *after* quality control gives the upper limit on achievable per-pixel accuracy.

1) *Computing agreement*: To compute agreement, we align the annotated object sets against each other, and compute the macro-averaged per-pixel f-score over the aligned object pairs. Alignment was done in a greedy fashion. For symbol sets  $S, T$ , we first align each  $t \in T$  to the  $s \in S$  with the highest pairwise f-score  $F(s, t)$ , then vice versa align each  $s \in S$  to the  $t \in T$  with the highest pairwise f-score. Taking the intersection, we then get symbol pairs  $s, t$  such that they are each other’s “best friends” in terms of f-score. The symbols that have no such a counterpart are left out of the alignment. Furthermore, symbol pairs that are not labeled with the same symbol class are removed from the alignment as well. When there are multiple such “best friend” candidates, we prefer aligning those that have the same symbol class. Objects that have no counterpart contribute 0 to both precision and recall.

2) *Agreement results*: The resulting f-scores are summarized in Table III. We measured inter-annotator agreement both before quality control (noQC-noQC) and after (withQC-withQC), and we also measured the extent to which quality

TABLE III: Inter-annotator agreement

Setting	macro-avg. f-score
noQC-noQC (inter-annot.)	0.89
noQC-withQC (self)	0.93
<b>withQC-withQC (inter-annot.)</b>	<b>0.97</b>

control changed the originally submitted annotations (noQC-withQC), averaged over the 7 annotators. Ideally, the post-QC measurements reflect the level of genuine disagreement among the annotators about how to lead the boundaries of objects in intersections and the inconsistency of QC, while the pre-QC measurements also measures the extent of actual mistakes that were fixed in QC.

Legitimate sources of disagreement lie in unclear symbol boundaries in intersections, and illegible handwriting. However, even after quality control, there were 689 – 691 objects in the image and 613 – 637 relationships, depending on which annotator we asked. This highlights the limits of both the annotation guidelines and QC: the ground truth is probably not entirely unambiguous, so various annotations of the same notation passed QC, and the QC process itself is not free from human error. At the same time, as seen in Table III, the two-round quality control process apparently removed nearly 4/5 of all disagreements, bringing the withQC inter-annotator f-score of 0.97 from a noQC f-score of 0.89. On average, QC introduced *less* change than what the original differences between individual annotators were. This suggests that the withQC results are somewhere in the “center” of the space of submitted annotations, and therefore the quality control process probably really leads to more accurate annotation instead of merely distorting the results in its own way.

#### V. BASELINE EXPERIMENTS

MUSCIMA++ allows developing and evaluating OMR systems on symbol recognition and notation reconstruction sub-tasks, both in isolation and jointly:

- **Symbol classification**: use the bounding boxes and symbol masks as inputs, symbol labels as outputs. Use primitive relationships to generate a ground truth of composite symbols, for compatibility with datasets of [11] or [2].
- **Symbol localization**: use the pages (or sub-regions) as inputs; the corresponding list of bounding boxes (and optionally, masks) is the output.
- **Primitives assembly**: use the bounding boxes/masks and labels as inputs, adjacency matrix as output.

Convincing baselines for handwritten musical symbol classification have already been established in [11]. We therefore focus on musical symbol localization and primitives assembly, for which MUSCIMA++ is a key contribution.

##### A. Symbol localization/segmentation

We examine a basic heuristics: skeleton graphs (SGs). Although we do not expect this baseline to be particularly strong, it could prove useful as an *oversegmentation* step, an initialization of other segmentation algorithms, and it

should illuminate what are the serious challenges posed by handwritten notation.

The **skeleton graph** (SG)  $G$  is derived from the morphological skeleton  $S$  of the binary image. Each endpoint (skeleton pixel with at most one 8-connected neighbor in  $S$ ) and junction (set of neighboring skeleton pixels with more than 2 neighbors in  $S$ ) forms a vertex of the skeleton graph, and every vertex pair  $u, v \in G$  such that there is an 8-connected path  $p \subset S$  from  $u$  to  $v$ , on which no other vertex  $v'$  lies, forms an edge  $e$  in  $G$ . When computing  $S$ , we smooth the foreground boundary by first dilating the image with a 3x3 square structuring element, then eroding it with a 5x5 diamond. (However, evaluation metrics are computed against the unsmoothed input image.)

We compute the oversegmentation on the binary images after staff removal.

To assess the usefulness of a given oversegmentation, we want to compute the *upper bound of segmentation performance*, assuming that the proposed superpixels will not be further subdivided: if we use the given oversegmentation, how much information do we inevitably lose? This is expressed well with *area under the precision-recall curve* (AUC-PR).

This inevitable loss of information is going to happen when a superpixel spans multiple symbols, and is not a subset of any one of them. For instance, the skeleton graph might not have a vertex at the boundary of two symbols  $s_1, s_2$ , so the edge is either “sticking out” of whichever symbol we assign it to, and – as SG edges do *not* overlap, except for junction vertices – its pixels are missing from whichever  $s_1, s_2$  we do *not* assign it to.

Because symbols can (and do) overlap arbitrarily, the oversegmentation setting is atypical in that it is a one-to-many alignment: one proposed superpixel can legitimately be a part of multiple symbols, which implies that assigning a superpixel to one symbol does not preclude assigning it to any other symbol. This enables us to treat symbols independently.

For each ground truth symbol  $s$  and its intersection  $I(s, S)$  with the image skeleton  $S$ , we can find: (A) the maximum-recall assignment  $A_r(s) = \cup_{e_{s,1}, \dots, e_{s,i}}$  of SG edges  $e_{s,1}, \dots, e_{s,i} \in E$  such that  $\forall e \in A_r(s) : e \subset I(s, S)$ ; (B) the maximum-precision assignment  $A_p(s) = \cup_{e_{s,1}, \dots, e_{s,j}}$  such that  $\forall x \in I(s, S) : x \in A_p(s)$ . The size of  $A_r(s)$  relative to the size of  $I(s, S)$  gives us maximum recall  $rec^+(s, E)$  at precision 1.0, and the size of  $I(s, S)$  relative to  $A_p(s)$  gives us maximum precision  $prec^+(s, E)$  at recall 1.0, *given the oversegmentation*  $E$  derived from the skeleton graph. We can then compute a lower bound on AUC-PR as  $rec^+(s, E) + (1 - rec^+(s, E)) * prec^+(s, E)$ . We use macro-averaging over symbols, as larger symbols are not necessarily more important (in fact, noteheads are most important, and they are some of the smallest symbols).

1) *Results.*: The average AUC-PR lower bound over all symbols in the dataset is 0.767, with average  $rec^+(s, E) = 0.548$  and  $prec^+(s, E) = 0.649$ .

We also measured “hard” recall: the proportion of ground truth symbols that have at least one “dedicated” SG edge

(nonzero  $rec + (s, E)$ ), so that they can be at least found (even if not particularly accurately) without “using up” the edge and compromising the ability to find another symbol. This proportion of objects with at least one skeleton graph edge that is a subset of  $I(s, S)$ , is, however, only 0.67, and unfortunately this disproportionately affects the most important symbols: there are 10121 out of 21356 full noteheads with  $rec^+(s, E) = 0$ , 194/348 such grace noteheads, and 12205/21416 stems. (However, when we measured hard recall for CCs directly, it was just 0.37.<sup>14</sup>)

## B. Notation graph construction

For primitives assembly, we establish a binary classification baseline *given gold-standard symbol segmentation and classification* for deciding whether oriented symbol pairs are related. As positive instances, we extract all 82247 symbol pairs connected by a relationship; as negative instances, we extract for each symbol all symbol within a threshold distance  $d_{neg}$ , set to 200 pixels (only 52 out of 82247 related symbol pairs are further away). As features for an oriented symbol pair  $u, v$ , we use their respective symbol classes, and the relative positions of their bounding boxes.

We used a decision tree classifier.<sup>15</sup> Using a random 80:20 training–test split, we obtained an f-score of 0.92 on recovering the 82247 positive instances. Note that this was achieved even without syntactic constraints (e.g.: “At least one stem per full notehead.”). Most frequent problems were in recovering notehead–beam relationships: about 1 in 10 notehead–beam relationships was a false negative. This result suggests that the primary difficulty in notation graph reconstruction will be dealing with symbol detection errors.

## VI. CONCLUSION

In MUSCIMA++, we provide an OMR dataset of handwritten music that allows training and benchmarking OMR systems tackling the symbol recognition and notation reconstruction stages of the OMR pipeline. Building on the CVC-MUSCIMA staff removal ground truth, we provide ground truth for symbol localization, classification, and notation graph construction, which is the step that performs ambiguity resolution necessary for inferring pitch and duration.

However, some requirements discussed in Sec. II, are not yet fully implemented. While stafflines, staves, and the relationships of noteheads to the staff symbols can be found automatically, it is not clear how accurately precedence can

<sup>14</sup>Note that skeleton graph oversegmentation will always perform at least as well as the connected components (CCs) heuristic. The skeleton of each connected component is also a connected component in the skeleton image, so if the given CC corresponds to a symbol (or is part of a multi-CC symbol), all edges in the skeleton of this CC will be assigned to  $A_r(s)$  and there will be no edge from this CC which will be in  $A_p(s)$  and not in  $A_r(s)$ . In fact, SG oversegmentation may lead to a better AUC. CC oversegmentation fails when one connected component consists of multiple symbols. However, the skeleton graph of the CC may consist of multiple edges, and some of these may be unrelated to one or more of the ground truth symbols, thereby not appearing in  $A_p(s, E)$  and improving – at least –  $prec^+(s, E)$ .

<sup>15</sup>We used the `scikit-learn` implementation, setting maximum tree depth to 20 and minimum number of instances per leaf to 10.



be inferred. Second, while the variety of handwriting collected by Fornés et al. [15] is impressive, it is all *contemporary* – whereas the application domain of handwritten OMR is also in early music, where different handwriting styles have been used. The dataset should also be re-encoded in a standard format. From the available musical score encodings, the Music Encoding Initiative (MEI<sup>16</sup>) is a format that can theoretically represent the notation graph and all its vertex attributes.

Finally, evaluation procedures over the notation graph need to be established. We are confident that the conceptual clarity of the MUSCIMA++ ground truth definition will simplify this task, although the relationship of simple metrics such as adjacency matrix f-score to semantical correctness of the output needs to be explored.

In spite of its imperfections, the MUSCIMA++ dataset is the most complete and extensive dataset for OMR to date. Together with the provided software, it should enable the OMR field to establish a more robust basis for comparing systems and measuring progress. Although evaluation procedures will need to be developed for the notation graph, we believe the fine-grained annotation will enable automatically evaluating at least the stage 2 and stage 3 tasks, in isolation and jointly, with a methodology close to those suggested in [5], [6], or [13]. Finally, it can also serve as the training data for extending the machine learning paradigm of OMR described by Calvo-Zaragoza et al. [12] to symbol recognition and notation assembly tasks.

We hope that the MUSCIMA++ dataset will be useful to the broad OMR community.

#### ACKNOWLEDGMENT

First of all, we thank our annotators for their dedicated work. We are also thankful to Alicia Fornés of CVC UAB<sup>17</sup>, who generously decided to share the CVC-MUSICMA dataset under the CC-BY-NC-SA 4.0 license, thus enabling us to share the MUSCIMA++ dataset in the same open manner as well.

This work is supported by the Czech Science Foundation, grant number P103/12/G084, the Charles University Grant Agency grants number 1444217 and 170217, and SVV project 260 453.

#### REFERENCES

- [1] D. Bainbridge and T. Bell, "The challenge of optical music recognition," *Computers and the Humanities*, vol. 35, pp. 95–121, 2001.
- [2] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R. S. Marcal, Carlos Guedes, and Jaime S. Cardoso, "Optical Music Recognition: State-of-the-Art and Open Issues," *Int J Multimed Info Retr*, vol. 1, no. 3, pp. 173–190, Mar 2012.
- [3] Jiří Novotný and Jaroslav Pokorný, "Introduction to Optical Music Recognition: Overview and Practical Challenges," *DATESO 2015 Proceedings of the 15th annual international workshop*, 2015.
- [4] D. Byrd, "Music Notation by Computer," Ph.D. dissertation, 1984.
- [5] Donald Byrd and Jakob Grue Simonsen, "Towards a Standard Testbed for Optical Music Recognition: Definitions, Metrics, and Page Images," *Journal of New Music Research*, vol. 44, no. 3, pp. 169–195, 2015.
- [6] Michael Droettboom and Ichiro Fujinaga, "Symbol-level groundtruthing environment for OMR," *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pp. 497–500, 2004.
- [7] J. Hajič jr., J. Novotný, P. Pecina, and J. Pokorný, "Further Steps towards a Standard Testbed for Optical Music Recognition," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, M. Mandel, J. Devaney, D. Turnbull, and G. Tzanetakis, Eds., New York University. New York, USA: New York University, 2016, pp. 157–163.
- [8] Arnau Baro, Pau Riba, and Alicia Fornés, "Towards the Recognition of Compound Music Notes in Handwritten Music Scores," in *15th International Conference on Frontiers in Handwriting Recognition, ICFHR 2016, Shenzhen, China, October 23-26, 2016*. IEEE Computer Society, 2016, pp. 465–470.
- [9] M. V. Stuckelberg and D. Doermann, "On musical score recognition using probabilistic reasoning," *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR '99 (Cat. No. PR00318)*, no. Did, pp. 115–118, 1999.
- [10] A. Rebelo, F. Paszkiewicz, C. Guedes, A. R. S. Marcal, and J. S. Cardoso, "A Method for Music Symbols Extraction based on Musical Rules," *Proceedings of BRIDGES*, no. 1, pp. 81–88, 2011.
- [11] Jorge Calvo-Zaragoza and Jose Oncina, "Recognition of Pen-Based Music Notation: The HOMUS Dataset," *22nd International Conference on Pattern Recognition*, Aug 2014.
- [12] J. Calvo Zaragoza, G. Vigiensoni, and I. Fujinaga, "A machine learning framework for the categorization of elements in images of musical documents," in *Third International Conference on Technologies for Music Notation and Representation*. A Coruña: University of A Coruña, 2017.
- [13] Pierfrancesco Bellini, Ivan Bruno, and Paolo Nesi, "Assessing Optical Music Recognition Tools," *Computer Music Journal*, vol. 31, no. 1, pp. 68–93, Mar 2007.
- [14] H. Miyao and R. M. Haralick, "Format of Ground Truth Data Used in the Evaluation of the Results of an Optical Music Recognition System," in *IAPR workshop on document analysis systems*, 2000, p. 497506.
- [15] A. Fornés, A. Dutta, A. Gordo, and J. Lladó, "CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal," *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 15, no. 3, pp. 243–251, 2012.
- [16] Baoguang Shi, Xiang Bai, and Cong Yao, "An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition," *CoRR*, vol. abs/1507.05717, 2015.
- [17] A. Fornés, A. Dutta, A. Gordo, and J. Lladó, "The ICDAR 2011 music scores competition: Staff removal and writer identification," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 1511–1515.
- [18] K. T. Reed and J. R. Parker, "Automatic computer recognition of printed music," *Proceedings - International Conference on Pattern Recognition*, vol. 3, pp. 803–807, 1996.
- [19] Liang Chen, Rong Jin, and Christopher Raphael, "Renotation from Optical Music Recognition," in *Mathematics and Computation in Music*. Springer Science + Business Media, 2015, pp. 16–26.
- [20] I. Fujinaga, "Optical Music Recognition using Projections," Master's thesis, 1988.
- [21] B. Coüasnon and J. Camillerapp, "Using Grammars To Segment and Recognize Music Scores," *Pattern Recognition*, pp. 15–27, October 1994.
- [22] D. Bainbridge and T. Bell, "A music notation construction engine for optical music recognition," *Software - Practice and Experience*, vol. 33, no. 2, pp. 173–200, 2003.
- [23] M. Szwach, "Guido: A musical score recognition system," *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 2, no. 3, pp. 809–813, 2007.
- [24] Ana Rebelo, Andre Marcal, and Jaime S. Cardoso, "Global constraints for syntactic consistency in OMR: an ongoing approach," in *Proceedings of the International Conference on Image Analysis and Recognition (ICIAR)*, 2013.
- [25] Christoph Dalitz, Michael Droettboom, Bastian Pranzas, and Ichiro Fujinaga, "A Comparative Study of Staff Removal Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 753–766, May 2008.
- [26] A. Rebelo, G. Capela, and J. S. Cardoso, "Optical recognition of music symbols," *International Journal on Document Analysis and Recognition*, vol. 13, pp. 19–31, 2010.

<sup>16</sup><http://www.music-encoding.org>

<sup>17</sup><http://www.cvc.uab.es/people/afornes/>

### 6.3 Groundtruthing (not only) Music Notation with MUSCIMarker: a Practical Overview

Jan Hajič jr. and Pavel Pecina. Groundtruthing (not only) Music Notation with MUSCIMarker: a Practical Overview. *14th IAPR International Conference on Document Analysis and Recognition / GREC*, Kyoto, Japan, pages 47–48, 2017. ISBN 978-1-5386-3586-5, doi: 10.1109/ICDAR.2017.271

The short workshop paper *Groundtruthing (not only) Music Notation with MUSCIMarker: a Practical Overview* introduces the MUSCIMarker software that was used to create the MUSCIMA++ dataset and is made openly available from GitHub<sup>1</sup>; the paper serves as the reference paper for the MUSCIMarker tool. The co-author Pavel Pecina contributed to the final text of the article with his comments. The contribution of the dissertation author is about 95% of the article.

---

<sup>1</sup><https://www.github.com/OMR-research/MUSCIMarker>



# Groundtruthing (not only) Music Notation with MUSCIMarker: a Practical Overview

Jan Hajič jr.

Institute of Formal and Applied Linguistics  
Charles University  
Email: hajicj@ufal.mff.cuni.cz

Pavel Pecina

Institute of Formal and Applied Linguistics  
Charles University  
Email: pecina@ufal.mff.cuni.cz

**Abstract**—Dataset creation for graphics recognition, especially for hand-drawn inputs, is often an expensive and time-consuming undertaking. The MUSCIMarker tool used for creating the MUSCIMA++ dataset for Optical Music Recognition (OMR) led to efficient use of annotation resources, and it provides enough flexibility to be applicable to creating datasets for other graphics recognition tasks where the ground truth can be represented similarly. First, we describe the MUSCIMA++ ground truth to define the range of tasks for which using MUSCIMarker to annotate ground truth is applicable. We then describe the MUSCIMarker tool itself, discuss its strong and weak points, and share practical experience with the tool from creating the MUSCIMA++ dataset.

## I. INTRODUCTION

Optical Music Recognition (OMR) is a field of graphics recognition that aims to automatically read music. Music notation encodes music with configurations of graphical primitives; OMR extracts the musical information back from the page. OMR can be likened to OCR for the music notation writing system; however, it is more difficult [1], [2]. One of the major roadblocks to OMR progress is the lack of publicly available datasets with ground truth [1]–[3]. The CVC-MUSCIMA dataset [4] provides ground truth for staff removal, the HOMUS dataset [5] provides ground truth for symbol classification, and only very recently the MUSCIMA++ dataset [6] has ground truth for symbol *localization* as well, and all these datasets are limited to contemporary handwriting, normalized staff size, and binary images. Therefore, it is reasonable to expect that OMR researchers will need to create datasets of their own.

The MUSCIMarker open-source annotation tool that was used to create MUSCIMA++ may prove useful for further groundtruthing efforts. Furthermore, the ground truth representation and the annotation tool are general enough to support the creation of different datasets beyond OMR. This work focuses on the practical aspects of dataset creation with MUSCIMarker. It should serve as a guide for anyone interested in datasets for OMR, and anyone who might contemplate creating an analogous dataset for a different task. We hope to invite feedback and suggestions for further development both of the ground truth and the software, with respect to the needs of the OMR community, but also the broader graphics recognition community as well, as the software may be useful e.g. for datasets of mathematical drawings or diagrams.

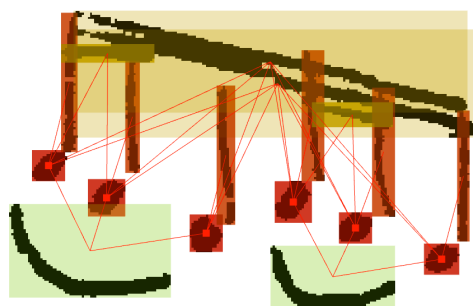


Fig. 1: A group of six notes that illustrates MUSCIMA++ notation graph. The rectangles are vertices, the lines represent edges (all outgoing from noteheads). Notice e.g. the two short beams in darker yellow, which are only relevant for notes 1 and 2, and 4 and 5, while all notes share the two long beams.

## II. THE NOTATION GRAPH GROUND TRUTH

The ground truth of MUSCIMA++ is a **notation graph**. It is a directed acyclic graph (DAG), its vertices are music notation primitives, and its edges connect primitives which are related to one another. OMR-specific is the definition of the alphabet of vertex classes and constraints on subgraphs, which required careful design in order to be useful for OMR focused both on the goals of replayability and reprintability [7].<sup>1</sup> An example is given in fig. 1. The data model for the annotation, the primitives alphabet, and their grammar through classes is implemented by the `muscima` Python package.<sup>2</sup>

Graphs have of course been explored before for representing music notation, e.g. [8], parse trees are implied whenever there is a mention of formal grammars (e.g., [9]–[12]), and the Audiveris software<sup>3</sup> uses a „Symbol Interpretation Graph” as its internal representation. However, to the best of our knowledge, no one before MUSCIMA++ has yet undertaken the effort to provide notation graph ground truth. Furthermore, our notation graph is based on *dependency* grammars, rather

<sup>1</sup>In full: <https://muscimarker.readthedocs.io/en/develop/instructions.html>

<sup>2</sup><https://github.com/hajicj/muscima>

<sup>3</sup><https://github.com/Audiveris/audiveris>

than *constituency* grammars, which are more general, e.g. allowing non-tree structures, which make the notation graph easier to query for musical information (pitches and durations of the encoded notes), and in general allow describing a broader range of visual structures.

### III. THE MUSCIMARKER ANNOTATION TOOL

The MUSCIMarker annotation tool<sup>4</sup> is implemented in Python 2.7, using the Kivy<sup>5</sup> framework. While Python is not the first choice for GUI applications, it is reliably cross-platform, and we anticipate tight integration of machine learning-based components automating portions of the annotation workflow, which can be expected to be Python. Kivy provides an elegant event/observer model that naturally enables triggering e.g. consistency checks or backups. MUSCIMarker is designed to work offline, for use e.g. in public transport.

While a tool for marking regions in images is in principle trivial, MUSCIMarker takes advantage of the nature of the inputs and the ground truth, and implements “tricks” that greatly improved annotator productivity. To speed up adding accurate objects with binary inputs, it provides auto-cropping and connected component search; for adding graph edges, parsing is available for primitive groups (permitted edges are defined in a “dependency grammar file”, which lists the allowed edges based on their starting and ending vertex classes and cardinalities.) Automated recognition functionality is being added concurrently with experimental progress. MUSCIMarker also provides extensive user action logging capability, which helped analyze productivity bottlenecks and prioritize features for development. Quality control tools were developed to address the most time-consuming problems: validating the notation graph, and searching for very small or sparse symbols.

Annotators marked on average 7.5 primitives+edges per minute overall, with the fastest at 10.5. (The author, with intimate knowledge of the software and ground truth, set the upper limit at 15.6.) [6] The dataset was done under budget.

MUSCIMarker still has important drawbacks that need to be addressed. Adding vertices is only optimized for binary images, edges cannot be differentiated into classes of their own, and the development documentation is rather poor.

### IV. DISCUSSION AND CONCLUSIONS

The toolchain used to create the MUSCIMA++ dataset is designed generically enough that it may serve the creators of other datasets with comparable graph-based ground truth. Through the experience with MUSCIMA++, it has been optimized for annotator productivity. Despite its limitations, it has been instrumental to efficient dataset creation, and we hope it will be a useful tool for addressing outstanding dataset needs across the OMR and broader graphics recognition community.

<sup>4</sup><https://github.com/hajicj/MUSCIMarker>,  
Documentation at <https://muscimarker.readthedocs.org>  
<sup>5</sup>[www.kivy.org](http://www.kivy.org)



Fig. 2: MUSCIMarker interface. Tool selection on the left; controls on the right. Highlighted relationships have been selected. (Last bar of from MUSCIMA++ image W-35\_N-0.8.)

### ACKNOWLEDGMENTS

This work is supported by the Czech Science Foundation, grant P103/12/G084, the Charles University Grant Agency, grants 1444217 and 170217, and by SVV project 260 453. We also thank our annotators for their substantial feedback.

### REFERENCES

- [1] D. Bainbridge and T. Bell, “The challenge of optical music recognition,” *Computers and the Humanities*, vol. 35, pp. 95–121, 2001.
- [2] Donald Byrd and Jakob Grue Simonsen, “Towards a Standard Testbed for Optical Music Recognition: Definitions, Metrics, and Page Images,” *Journal of New Music Research*, vol. 44, no. 3, pp. 169–195, 2015. [Online]. Available: <http://dx.doi.org/10.1080/09298215.2015.1045424>
- [3] Michael Droettboom and Ichiro Fujinaga, “Symbol-level groundtruthing environment for OMR,” *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pp. 497–500, 2004.
- [4] A. Fornés, A. Dutta, A. Gordo, and J. Llad, “CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal,” *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 15, no. 3, pp. 243–251, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s10032-011-0168-2>
- [5] Jorge Calvo-Zaragoza and Jose Oncina, “Recognition of Pen-Based Music Notation: The HOMUS Dataset,” *22nd International Conference on Pattern Recognition*, Aug 2014. [Online]. Available: <http://dx.doi.org/10.1109/ICPR.2014.524>
- [6] J. Hajič, jr. and P. Pecina, “In Search of a Dataset for Handwritten Optical Music Recognition: Introducing MUSCIMA++,” *ArXiv e-prints*, Mar. 2017.
- [7] H. Miyao and R. M. Haralick, “Format of Ground Truth Data Used in the Evaluation of the Results of an Optical Music Recognition System,” in *IAPR workshop on document analysis systems*, 2000, p. 497506.
- [8] K. T. Reed and J. R. Parker, “Automatic computer recognition of printed music,” *Proceedings - International Conference on Pattern Recognition*, vol. 3, pp. 803–807, 1996.
- [9] I. Fujinaga, “Optical Music Recognition using Projections,” Master’s thesis, 1988.
- [10] B. Coüasnon and J. Camillerapp, “Using Grammars To Segment and Recognize Music Scores,” *Pattern Recognition*, pp. 15–27, October 1994.
- [11] D. Bainbridge and T. Bell, “A music notation construction engine for optical music recognition,” *Software - Practice and Experience*, vol. 33, no. 2, pp. 173–200, 2003.
- [12] M. Szwoch, “Guido: A musical score recognition system,” *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 2, no. 3, pp. 809–813, 2007.

## 6.4 Further Steps Towards a Standard Testbed for Optical Music Recognition

Jan Hajič jr., Jiří Novotný, Pavel Pecina and Jaroslav Pokorný: Further Steps towards a Standard Testbed for Optical Music Recognition. *Proceedings of the 17th International Society for Music Information Retrieval Conference*, pages 157–163, New York, USA, 2016. ISBN 978-0-692-75506-8.

The paper **Further Steps Towards a Standard Testbed for Optical Music Recognition** [Hajič jr. et al., 2016] responds to the lack of established automated evaluation metrics for OMR. In order to establish such an automated metric, one needs to ascertain that the results of the computation actually correspond to the quality of the output. Inspired by the data-driven methodology used for establishing the BLEU metric in machine translation, the article uses the following methodology: collect a corpus of human judgments, and then assess the adequacy of proposed automated OMR evaluation metrics according to how they agree with the human judgments. This omreval corpus was collected, its reliability was analyzed, and several baseline OMR evaluation metrics for comparing MusicXML files – the closest to an interchange format for music notation – were tested against the omreval corpus. As a companion to the evaluation methodology, a subset of the CVC-MUSCIMA dataset was annotated with symbol locations and MusicXML encodings, as a prototype of a multi-layer test corpus for OMR itself.

In this article, the thesis author designed and performed did all the work and writing, except for section 4.1, Symbol-level ground truth, which was done by co-author Jiří Novotný. The co-authors Pavel Pecina and Jaroslav Pokorný contributed to the final text of the article with their comments. The contribution of the dissertation author is about 80% of the article.

# FURTHER STEPS TOWARDS A STANDARD TESTBED FOR OPTICAL MUSIC RECOGNITION

Jan Hajič jr.<sup>1</sup>      Jiří Novotný<sup>2</sup>      Pavel Pecina<sup>1</sup>      Jaroslav Pokorný<sup>2</sup>

<sup>1</sup> Charles University, Institute of Formal and Applied Linguistics, Czech Republic

<sup>2</sup> Charles University, Department of Software Engineering, Czech Republic

hajicj@ufal.mff.cuni.cz, novotny@ksi.mff.cuni.cz

## ABSTRACT

Evaluating Optical Music Recognition (OMR) is notoriously difficult and automated end-to-end OMR evaluation metrics are not available to guide development. In “Towards a Standard Testbed for Optical Music Recognition: Definitions, Metrics, and Page Images”, Byrd and Simonsen recently stress that a benchmarking standard is needed in the OMR community, both with regards to data and evaluation metrics. We build on their analysis and definitions and present a prototype of an OMR benchmark. We do not, however, presume to present a complete solution to the complex problem of OMR benchmarking. Our contributions are: (a) an attempt to define a multi-level OMR benchmark dataset and a practical prototype implementation for both printed and handwritten scores, (b) a corpus-based methodology for assessing automated evaluation metrics, and an underlying corpus of over 1000 qualified relative cost-to-correct judgments. We then assess several straightforward automated MusicXML evaluation metrics against this corpus to establish a baseline over which further metrics can improve.

## 1. INTRODUCTION

Optical Music Recognition (OMR) suffers from a lack of evaluation standards and benchmark datasets. There is presently no publicly available way of comparing various OMR tools and assessing their performance. While it has been argued that OMR can go far even in the absence of such standards [7], the lack of benchmarks and difficulty of evaluation has been noted on multiple occasions [2, 16, 21]. The need for end-to-end system evaluation (at the final level of OMR when musical content is reconstructed and made available for further processing), is most pressing when comparing against commercial systems such as PhotoScore,<sup>1</sup> SmartScore<sup>2</sup> or SharpEye<sup>3</sup>:

<sup>1</sup> <http://www.neuratron.com/photoscore.htm>

<sup>2</sup> <http://www.musitek.com/index.html>

<sup>3</sup> <http://www.visiv.co.uk>



© Jan Hajič jr., Jiří Novotný, Pavel Pecina, Jaroslav Pokorný. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Jan Hajič jr., Jiří Novotný, Pavel Pecina, Jaroslav Pokorný. “Further Steps towards a Standard Testbed for Optical Music Recognition”, 17th International Society for Music Information Retrieval Conference, 2016.

these typically perform as “black boxes”, so evaluating on the level of individual symbols requires a large amount of human effort for assessing symbols and their locations, as done by Bellini et al. [18] or Sapp [19].

OMR systems have varying goals, which should be reflected in evaluation. Helping speed up transcription should be measured by some cost-to-correct metric; a hypothetical automated score interpretation system could require accurate MIDI, but does not need to resolve all slurs and other symbols; digitizing archive scores for retrieval should be measured by retrieval accuracy; etc. We focus on evaluating transcription, as it is most sensitive to errors and most lacking in evaluation metrics.

Some OMR subtasks (binarization, staff identification and removal, symbol localization and classification) have natural ways of evaluating, but the end-to-end task does not: it is difficult to say how good a semantic representation (e.g., MusicXML) is. Manually evaluating system outputs is costly, slow and difficult to replicate; and aside from Knopke and Byrd [12], Szwoch [20] and Padilla et al. [21], we know of no attempts to even define an automatic OMR evaluation metric, much less define a methodology for assessing how well it actually evaluates.

Our contribution does not presume to define an entire evaluation standard. Instead, we propose a robust, cumulative, data-driven methodology for *creating* one. We collect human preference data that can serve as a gold standard for comparing MusicXML automated evaluation metrics, mirroring how the BLEU metric and its derivatives has been established as an evaluation metric for the similarly elusive task of assessing machine translation based on agreement with human judgements [17]. This “evaluating evaluation” approach is inspired by the Metrics track of the Workshop of Statistical Machine Translation competition (WMT) [3, 5, 14]. To collect cost-to-correct estimates for various notation errors, we generate a set of synthetically distorted “recognition outputs” from a set of equally synthetic “true scores”. Then, annotators are shown examples consisting of a true score and a pair of the distorted scores, and they are asked to choose the simulated recognition output that would take them less time to correct.

Additionally, we provide an OMR benchmark dataset prototype with ground truth at the symbol and end-to-end levels.

The main contributions of our work are:

- A corpus-based “evaluating evaluation” methodol-



ogy that enables iteratively improving, refining and fine-tuning automated OMR evaluation metrics.

- A corpus of 1230 human preference judgments as gold-standard data for this methodology, and assessments of example MusicXML evaluation metrics against this corpus.
- Definitions of ground truths that can be applied to Common Western Music Notation (CWMN) scores.
- MUSCIMA++. A prototype benchmark with multiple levels of ground truth that extends a subset of the CVC-MUSCIMA dataset [9], with 3191 annotated notation primitives.

The rest of this paper is organized as follows: in Sec. 2, we review the state-of-the-art on OMR evaluation and datasets; in Sec. 3, we describe the human judgment data for developing automated evaluation metrics and demonstrate how it can help metric development. In Sec. 4, we present the prototype benchmark and finally, in Sec. 5, we summarize our findings and suggest further steps to take.<sup>4</sup>

## 2. RELATED WORK

The problem of evaluating OMR and creating a standard benchmark has been discussed before [7, 10, 16, 18, 20] and it has been argued that evaluating OMR is a problem as difficult as OMR itself. Jones et al. [10] suggest that in order to automatically measure and evaluate the performance of OMR systems, we need (a) a standard dataset and standard terminology, (b) a definition of a set of rules and metrics, and (c) definitions of different ratios for each kind of errors. The authors noted that distributors of commercial OMR software often claim the accuracy of their system is about 90 %, but provide no information about how that value was estimated.

Bellini et al. [18] manually assess results of OMR systems at two levels of symbol recognition: low-level, where only the presence and positioning of a symbol is assessed, and high-level, where the semantic aspects such as pitch and duration are evaluated as well. At the former level, mistaking a beamed group of 32nds for 16ths is a minor error; at the latter it is much more serious. They defined a detailed set of rules for counting symbols as recognized, missed and confused symbols. The symbol set used in [18] is quite rich: 56 symbols. They also define *recognition gain*, based on the idea that an OMR system is at its best when it minimizes the time needed for correction as opposed to transcribing from scratch, and stress *verification cost*: how much it takes to verify whether an OMR output is correct.

An extensive theoretical contribution towards benchmarking OMR has been made recently by Byrd and Simonson [7]. They review existing work on evaluating OMR systems and clearly formulate the main issues related to evaluation. They argue that the complexity of CWMN is the main reason why OMR is inevitably problematic, and

suggest the following stratification into levels of difficulty:

1. Music on one staff, strictly monophonic,
2. Music on one staff, polyphonic,
3. Music on multiple staves, but each strictly monophonic, with no interaction between them,
4. “Pianoform”: music on multiple staves, one or more having multiple voices, and with significant interaction between and/or within staves.

They provide 34 pages of sheet music that cover the various sources of difficulty. However, the data does not include handwritten music and no ground truth for this corpus is provided.

Automatically evaluating MusicXML has been attempted most significantly by Szwoch [20], who proposes a metric based on a top-down MusicXML node matching algorithm and reports agreement with human annotators, but how agreement was assessed is not made clear, no implementation of the metric is provided and the description of the evaluation metric itself is quite minimal. Due to the complex nature of MusicXML (e.g., the same score can be correctly represented by different MusicXML files), Szwoch also suggests a different representation may be better than comparing two MusicXML files directly.

More recently, evaluating OMR with MusicXML outputs has been done by Padilla et al. [21]. While they provide an implementation, there is no comparison against gold-standard data. (This is understandable, as the paper [21] is focused on recognition, not evaluation.) Aligning MusicXML files has also been explored by Knopke and Byrd [12] in a similar system-combination setting, although not for the purposes of evaluation. They however make an important observation: stems are often mistaken for barlines, so the obvious simplification of first aligning measures is not straightforward to make.

No publicly available OMR dataset has ground truth for end-to-end recognition. The CVC-MUSCIMA dataset for staffline identification and removal and writer identification by Fornés et al. [9] is most extensive, with 1000 handwritten scores (50 musicians copying a shared set of 20 scores) and a version with staves removed, which is promising for automatically applying ground truth annotations across the 50 versions of the same score. Fornés et al. [8] have also made available a dataset of 2128 clefs and 1970 accidentals.

The HOMUS musical symbol collection for online recognition [11] consists of 15200 samples (100 musicians, 32 symbol classes, 4-8 samples per class per musician) of individual handwritten musical symbols. The dataset can be used for both online and offline symbol classification.

A further dataset of 3222 handwritten and 2521 printed music symbols is available upon request [1]. Bellini et al. [18] use 7 selected images for their OMR assessment; unfortunately, they do not provide a clear description of the database and its ground truth, and no more information is publicly available. Another staffline removal dataset is Dalitz’s database,<sup>5</sup> consisting of 32 music pages that cov-

<sup>4</sup> All our data, scripts and other supplementary materials are available at <https://github.com/ufal/omreval> as a git repository, in order to make it easier for others to contribute towards establishing a benchmark.

<sup>5</sup> <http://music-staves.sourceforge.net>

ers a wide range of music types (CWMN, lute tablature, chant, mensural notation) and music fonts. Dalitz et al. [6] define several types of distortion in order to test the robustness of the different staff removal algorithms, simulating both image degradation and page deformations. These have also been used to augment CVC-MUSCIMA.

There are also large sources such as the Mutopia project<sup>6</sup> with transcriptions to LilyPond and KernScores<sup>7</sup> with HumDrum. The IMSLP database<sup>8</sup> holds mostly printed scores, but manuscripts as well; however, as opposed to Mutopia and KernScores, IMSLP generally only provides PDF files and no transcription of their musical content, except for some MIDI recordings.

### 3. EVALUATING EVALUATION

OMR lacks an automated evaluation metric that could guide development and reduce the price of conducting evaluations. However, an automated metric for OMR evaluation needs itself to be evaluated: does it really rank as better systems that *should* be ranked better?

Assuming that the judgment of (qualified) annotators is considered the gold standard, the following methodology then can be used to assess an automated metric:

1. Collect a corpus of annotator judgments to define the expected gold-standard behavior,
2. Measure the agreement between a proposed metric and this gold standard.

This approach is inspired by machine translation (MT), a field where comparing outputs is also notoriously difficult: the WMT competition has an evaluation track [5, 14], where automated MT metrics are evaluated against human-collected evaluation results, and there is ongoing research [3, 15] to design a better metric than the current standards such as BLEU [17] or Meteor [13]. This methodology is nothing surprising; in principle, one could machine-learn a metric given enough gold-standard data. However: how to best design the gold-standard data and collection procedure, so that it encompasses what we in the end want our application (OMR) to do? How to measure the quality of such a corpus – given a collection of human judgments, how much of a gold standard is it?

In this section, we describe a data collection scheme for human judgments of OMR quality that should lead to comparing automated metrics.

#### 3.1 Test case corpus

We collect a corpus  $C$  of *test cases*. Each test case  $c_1 \dots c_N$  is a triplet of music scores: an “ideal” score  $I_i$  and two “mangled” versions,  $P_i^{(1)}$  and  $P_i^{(2)}$ , which we call *system outputs*. We asked our  $K$  annotators  $a_1 \dots a_K$  to choose the less mangled version, formalized as assigning  $r_a(c_i) = -1$  if they preferred  $P_i^{(1)}$  over  $P_i^{(2)}$ , and  $+1$  for the opposite preference. The term we use is to “rank” the predictions. When assessing an evaluation metric against

this corpus, the test case rankings then constrain the space of well-behaved metrics.<sup>9</sup>

The exact formulation of the question follows the “cost-to-correct” model of evaluation of [18]:

“Which of the two system outputs would take you less effort to change to the ideal score?”

##### 3.1.1 What is in the test case corpus?

We created 8 ideal scores and derived 34 “system outputs” from them by introducing a variety of mistakes in a notation editor. Creating the system outputs manually instead of using OMR outputs has the obvious disadvantage that the distribution of error types does not reflect the current OMR state-of-the-art. On the other hand, once OMR systems change, the distribution of corpus errors becomes obsolete anyway. Also, we create errors for which we can assume the annotators have a reasonably accurate estimate of their own correction speed, as opposed to OMR outputs that often contain strange and syntactically incorrect notation, such as isolated stems. Nevertheless, when more annotation manpower becomes available, the corpus should be extended with a set of actual OMR outputs.

The ideal scores (and thus the derived system outputs) range from a single whole note to a “pianoform” fragment or a multi-staff example. The distortions were crafted to cover errors on individual notes (wrong pitch, extra accidental, key signature or clef error, etc.: micro-errors on the semantic level in the sense of [16, 18]), systematic errors within the context of a full musical fragment (wrong beaming, swapping slurs for ties, confusing staccato dots for noteheads, etc.), short two-part examples to measure the tradeoff between large-scale layout mistakes and localized mistakes (e.g., a four-bar two-part segment, as a perfect concatenation of the two parts into one vs. in two parts, but with wrong notes) and longer examples that constrain the metric to behave sensibly at larger scales.

Each pair of system outputs derived from the same ideal score forms a test case; there are 82 in total. We also include 18 control examples, where one of the system outputs is identical to the ideal score. A total of 15 annotators participated in the annotation, of whom 13 completed all 100 examples; however, as the annotations were voluntary, only 2 completed the task twice for measuring intra-annotator agreement.

##### 3.1.2 Collection Strategy

While Bellini et al. [18] define how to count individual errors at the level of musical symbols, assign some cost to each kind of error (miss, add, fault, etc.) and define the overall cost as composed of those individual costs, our methodology does not assume that the same type of error has the same cost in a different *context*, or that the overall cost can be computed from the individual costs: for instance, a sequence of notes shifted by one step can be in

<sup>6</sup> <http://www.mutopiaproject.org>

<sup>7</sup> <http://humdrum.ccarh.org>

<sup>8</sup> <http://imslp.org>

<sup>9</sup> We borrow the term “test case” from the software development practice of unit testing: test cases verify that the program (in our case the evaluation metric) behaves as expected on a set of inputs chosen to cover various standard and corner cases.



most editors corrected simultaneously (so, e.g., clef errors might not be too bad, because the entire part can be transposed together).

Two design decisions of the annotation task merit further explanation: why we ask annotators to compare examples instead of rating difficulty, and why we disallow equality.

**Ranking.** The practice of ranking or picking the best from a set of possible examples is inspired by machine translation: Callison-Burch et al. have shown that people are better able to agree on which proposed translation is better than on how good or bad individual translations are [4]. Furthermore, ranking does not require introducing a cost metric in the first place. Even a simple 1-2-3-4-5 scale has this problem: how much effort is a “1” on that scale? How long should the scale be? What would the relationship be between short and long examples?

Furthermore, this annotation scheme is fast-paced. The annotators were able to do all the 100 available comparisons within 1 hour. Rankings also make it straightforward to compare automated evaluation metrics that output values from different ranges: just count how often the metric agrees with gold-standard ranks using some measure of monotonicity, such as Spearman’s rank correlation coefficient.

**No equality.** It is also not always clear which output would take less time to edit; some errors genuinely are equally bad (sharp vs. flat). These are also important constraints on evaluation metrics: the costs associated with each should not be too different from each other. However, allowing annotators to explicitly mark equality risks overuse, and annotators using *underqualified* judgment. For this first experiment, therefore, we elected not to grant that option; we then interpret disagreement as a sign of uncertainty and annotator uncertainty as a symptom of this genuine tie.

### 3.2 How gold is the standard?

All annotators ranked the control cases correctly, except for one instance. However, this only accounts for elementary annotator failure and does not give us a better idea of systematic error present in the experimental setup. In other words, we want to ask the question: if all annotators are performing to the best of their ability, **what level of uncertainty should be expected under the given annotation scheme?** (For the following measurements, the control cases have been excluded.)

Normally, inter-annotator agreement is measured: if the task is well-defined, i.e., if a gold standard *can* exist, the annotators will tend to agree with each other towards that standard. However, usual agreement metrics such as Cohen’s  $\kappa$  or Krippendorff’s  $\alpha$  require computing *expected agreement*, which is difficult when we do have a subset of examples on which we do *not* expect annotators to agree but cannot *a priori* identify them. We therefore start by defining a simple agreement metric  $L$ . Recall:

- $C$  stands for the corpus, which consists of  $N$  examples  $c_1 \dots c_N$ ,

- $A$  is the set of  $K$  annotators  $a_1 \dots a_K$ ,  $a, b \in A$ ;
- $r_a$  is the *ranking function* of an annotator  $a$  that assigns +1 or -1 to each example in  $c$ ,

$$L(a, b) = \frac{1}{N} \sum_{c \in C} \frac{|r_a(c) + r_b(c)|}{2}$$

This is simply the proportion of cases on which  $a$  and  $b$  agree: if they disagree,  $r_a(c) + r_b(c) = 0$ . However, we expect the annotators to disagree on the genuinely uncertain cases, so some disagreements are not as serious as others. To take the existence of legitimate disagreement into account, we modify  $L(a, b)$  to weigh the examples according to how certain the other annotators  $A \setminus \{a, b\}$  are about the given example. We define weighed agreement  $L_w(a, b)$ :

$$L_w(a, b) = \frac{1}{N} \sum_{c \in C} w^{(-a, b)}(c) \frac{|r_a(c) + r_b(c)|}{2}$$

where  $w^{(-a, b)}$  is defined for an example  $c$  as:

$$w^{(-a, b)}(c) = \frac{1}{K-2} \left| \sum_{a' \in A \setminus \{a, b\}} r_{a'}(c) \right|$$

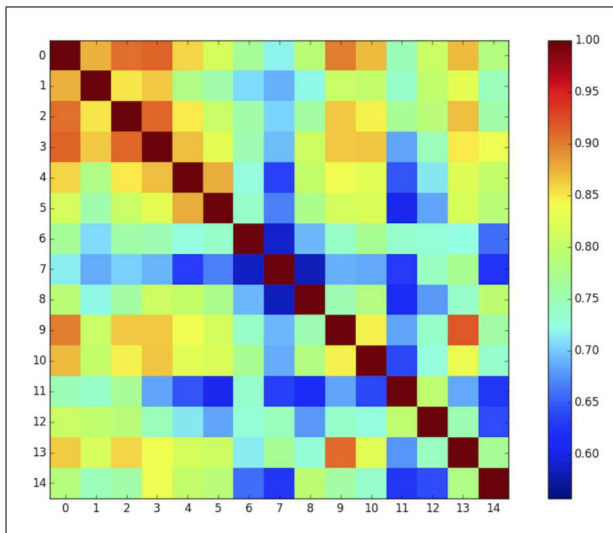
This way, it does not matter if  $a$  and  $b$  disagree on cases where no one else agrees either, but if they disagree on an example where there is strong consensus, it should bring the overall agreement down. Note that while maximum achievable  $L(a, b)$  is 1 for perfectly agreeing annotators (i.e., all the sum terms equal to 1), because  $w(c) \leq 1$ , the maximum achievable  $L_w(a, b)$  will be less than 1, and furthermore depends on the choice of  $a$  and  $b$ : if we take notoriously disagreeing annotators away from the picture, the weights will increase overall. Therefore, we finally adjust  $L_w(a, b)$  to the proportion of maximum achievable  $L_w(a, b)$  for the given  $(a, b)$  pair, which is almost the same as  $L_w(a, a)$  with the exception that  $b$  *must also be excluded from computing the weights*. We denote this maximum as  $L_w^*(a, b)$ , and the adjusted metric  $\hat{L}_w$  is then:

$$\hat{L}_w(a, b) = L_w(a, b) / L_w^*(a, b)$$

This metric says: “What proportion of achievable weighed agreement has been actually achieved?” The upper bound of  $\hat{L}_w$  is therefore 1.0 again; the lower bound is agreement between two randomly generated annotators, with the humans providing the consensus.

The resulting pairwise agreements, with the lower bound established by averaging over 10 random annotators, are visualized in Fig. 1. The baseline agreement  $\hat{L}_w$  between random annotators weighed by the full human consensus was close to 0.5, as expected. There seems to be one group of annotators relatively in agreement (green and above, which means adjusted agreement over 0.8), and then several individuals who disagree with everyone – including among themselves (lines 6, 7, 8, 11, 12, 14).

Interestingly, most of these “lone wolves” reported significant experience with notation editors, while the group more in agreement not as much. We suspect this is because



**Figure 1.** Weighed pairwise agreement. The cell  $[a, b]$  represents  $\hat{L}_w(a, b)$ . The scale goes from the average random agreement (ca. 0.55) up to 1.

with increasing notation editor experience, users develop a personal editing style that makes certain actions easier than others by learning a *subset* of the “tricks” available with the given editing tools – but each user learns a different subset, so agreement on the relative editing cost suffers. To the contrary, inexperienced users might not have spent enough time with the editor to develop these habits.

### 3.3 Assessing some metrics

We illustrate how the test case ranking methodology helps analyze these rather trivial automated MusicXML evaluation metrics:

1. Levenshtein distance of XML canonization (**c14n**),
2. Tree edit distance (**TED**),
3. Tree edit distance with `<note>` flattening (**TEDn**),
4. Convert to LilyPond + Levenshtein distance (**Ly**).

**c14n.** Canonize the MusicXML file formatting and measure Levenshtein distance. This is used as a trivial baseline.

**TED.** Measure Tree Edit Distance on the MusicXML nodes. Some nodes that control auxiliary and MIDI information (`work`, `defaults`, `credit`, and `duration`) are ignored. Replacement, insertion, and deletion all have a cost of 1.

**TEDn.** Tree Edit Distance with special handling of `note` elements. We noticed that many errors of TED are due to the fact that while deleting a note is easy in an editor, the edit distance is higher because the `note` element has many sub-nodes. We therefore encode the notes into strings consisting of one position per `pitch`, `stem`, `voice`, and `type`. Deletion cost is fixed at 1, insertion cost is 1 for non-note nodes, and  $1 + \text{length of code}$  for notes. Replacement cost between notes is the edit distance between their codes; replacement between a note and non-note costs  $1 + \text{length of code}$ ; between non-notes costs 1.

Metric	$r_s$	$\hat{r}_s$	$\rho$	$\hat{\rho}$	$\tau$	$\hat{\tau}$
c14n	0.33	0.41	0.40	0.49	0.25	0.36
TED	0.46	0.58	0.40	0.50	0.35	0.51
TEDn	<b>0.57</b>	<b>0.70</b>	0.40	0.49	<b>0.43</b>	<b>0.63</b>
Ly	0.41	0.51	0.29	0.36	0.30	0.44

**Table 1.** Measures of agreement for some proposed evaluation metrics.

**Ly.** The LilyPond<sup>10</sup> file format is another possible representation of a musical score. It encodes music scores in its own LaTeX-like language. The first bar of the “Twinkle, twinkle” melody would be represented as `d' 8 [ d' 8 ] a' 8 [ a' 8 ] b' 8 [ b' 8 ] a' 4 |` This representation is much more amenable to string edit distance. The **Ly** metric is Levenshtein distance on the LilyPond import of the MusicXML system output files, with all whitespace normalized.

For comparing the metrics against our gold-standard data, we use nonparametric approaches such as Spearman’s  $r_s$  and Kendall’s  $\tau$ , as these evaluate monotonicity without assuming anything about mapping values of the evaluation metric to the  $[-1, 1]$  range of preferences. To reflect the “small-difference-for-uncertain-cases” requirement, however, we use Pearson’s  $\rho$  as well [14]. For each way of assessing a metric, its maximum achievable with the given data should be also estimated, by computing how the metric evaluates the consensus of one group of annotators against another. We randomly choose 100 splits of 8 vs 7 annotators, compute the average preferences for the two groups in a split and measure the correlations between the average preferences. The expected upper bounds and standard deviations estimated this way are:

- $r_s^* = 0.814$ , with standard dev. 0.040
- $\rho^* = 0.816$ , with standard dev. 0.040
- $\tau^* = 0.69$ , with standard dev. 0.045

We then define  $\hat{r}_s$  as  $\frac{r_s}{r_s^*}$ , etc. Given a cost metric  $\mathcal{L}$ , we get for each example  $c_i = (I_i, P_i^{(1)}, P_i^{(2)})$  the cost difference  $\ell(c_i) = \mathcal{L}(I_i, P_i^{(1)}) - \mathcal{L}(I_i, P_i^{(2)})$  and pair it with the gold-standard consensus  $r(c_i)$  to get pairwise inputs for the agreement metrics.

The agreement of the individual metrics is summarized in Table 1. When developing the metrics, we did *not* use the gold-standard data against which metric performance is measured here; we used only our own intuition about how the test cases should come out.

## 4. BENCHMARK DATASET PROTOTYPE

A benchmark dataset should have ground truth at levels corresponding to the standard OMR processing stages, so that sub-systems such as staff removal, or symbol localization can be compared with respect to the end-to-end pipeline they are a part of. We also suspect handwritten music will remain an open problem much longer than printed music. Therefore, we chose to extend the **CVC-MUSCIMA** dataset instead of Byrd and Simonsen’s pro-

<sup>10</sup><http://www.lilypond.org>



posed test bed [7] because of the extensive handwritten data collection effort that has been completed by Fornés et al. and because ground truth for staff removal and binarization is already present. At the same time, CVC-MUSCIMA covers all the levels of notational complexity from [7], as well as a variety of notation symbols, including complex tuples, less common time signatures (5/4), C-clefs and some symbols that could very well expose the differences between purely symbol-based and more syntax-aware methods (e.g., tremolo marks, easily confused for beams). We have currently annotated symbols in printed scores only, with the perspective of annotating the handwritten scores automatically or semi-automatically.

We selected a subset of scores that covers the various levels of notational complexity: single-part monophonic music (F01), multi-part monophonic music (F03, F16), and pianoform music, primarily based on chords (F10) and polyphony (F08), with interaction between staves.

#### 4.1 Symbol-level ground truth

Symbols are represented as bounding boxes, labeled by symbol class. In line with the low-level and high-level symbols discussed by [7], we differentiate symbols at the level of *primitives* and the level of *signs*. The relationship between primitives and signs can be one-to-one (e.g., clefs), many-to-one (composite signs: e.g. notehead, stem, and flag form a note), one-to-many (disambiguation: e.g., a sharp primitive can be part of a key signature, accidental, or an ornament accidental), and many-to-many (the same beam participates in multiple beamed notes, but each beamed note also has a stem and notehead). We include individual numerals and letters as notation primitives, and their disambiguation (tuplet, time signature, dynamics...) as signs.

We currently define 52 primitives plus letters and numerals, and 53 signs. Each symbol can be linked to a MusicXML counterpart.<sup>11</sup> There are several groups of symbols:

- Note elements (noteheads, stems, beams, rests...)
- Notation elements (slurs, dots, ornaments...)
- Part default (clefs, time and key signatures...)
- Layout elements (staves, brackets, braces...)
- Numerals and text.

We have so far annotated the primitive level. There are 3191 primitives marked in the 5 scores. Annotation took about 24 hours of work in a custom editor.

#### 4.2 End-to-end ground truth

We use MusicXML as the target representation, as it is supported by most OMR/notation software, actively maintained and developed and available under a sufficiently permissive license. We obtain the MusicXML data by manually transcribing the music and postprocessing to ensure each symbol has a MusicXML equivalent. Postprocessing mostly consists of filling in default barlines and correcting

<sup>11</sup> The full lists of symbol classes are available in the repository at <https://github.com/ufal/omreval> under `muscima++/data/Symbolic/specification`.

staff grouping information. Using the MuseScore notation editor, transcription took about 3.5 hours.

## 5. CONCLUSIONS AND FUTURE WORK

We proposed a corpus-based approach to assessing automated end-to-end OMR evaluation metrics and illustrated the methodology on several potential metrics. A gold standard annotation scheme based on assessment of relative cost-to-correct of synthetic “system outputs” was described that avoids pre-defining any cost metric, and the resulting corpus of 1230 human judgments was analyzed for inter-annotator agreement, taking into account the possibility that the compared system outputs may not be clearly comparable. This preference-based setup avoids the need to pre-define any notion of cost, requires little annotator training, and it is straightforward to assess an evaluation metric against this preference data.

Our results suggest that the central assumption of a single ground truth for preferences among a set of system outputs is weaker with increasing annotator experience. To make the methodology more robust, we recommend:

- Explicitly control for experience level; do not assume that more annotator experience is better.
- Measure actual cost-to-correct (in time and interface operations) through a notation editor, to verify how much human *estimation* of this cost can be relied on.
- Develop models for computing expected agreement for data where the annotations may legitimately be randomized (the “equally bad” cases). Once expected agreement can be computed, we can use more standard agreement metrics.

The usefulness of the test case corpus for developing automated evaluation metrics was clear: the TEDn metric that outperformed the others by a large margin was developed through analyzing the shortcomings of the TED metric on individual test cases (before the gold-standard data had been collected). As Szwoch [20] suggested, modifying the representation helped. However, if enough human judgments are collected, it should even be possible to sidestep the difficulties of hand-crafting an evaluation metric through machine learning; we can for instance try learning the insertion, deletion, and replacement costs for individual MusicXML node types.

An OMR environment where different systems can be meaningfully compared, claims of commercial vendors are verifiable and progress can be measured is in the best interest of the OMR community. We believe our work, both on evaluation and on a dataset, constitutes a significant step in this direction.

## 6. ACKNOWLEDGMENTS

This research is supported by the Czech Science Foundation, grant number P103/12/G084. We would also like to thank our annotators from the Janáček Academy of Music and Performing Arts in Brno and elsewhere, and Alicia Fornés for providing additional background and material for the CVC-MUSCIMA dataset.

## 7. REFERENCES

- [1] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R. S. Marcal, Carlos Guedes, and Jaime S. Cardoso. Optical Music Recognition: State-of-the-Art and Open Issues. *Int J Multimed Info Retr*, 1(3):173–190, Mar 2012.
- [2] Baoguang Shi, Xiang Bai, and Cong Yao. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. *CoRR*, abs/1507.05717, 2015.
- [3] Ondřej Bojar, Miloš Ercegovčević, Martin Popel, and Omar F. Zaidan. A Grain of Salt for the WMT Manual Evaluation. *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 1–11, 2011.
- [4] Chris Callison Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. (Meta-) Evaluation of Machine Translation. *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, 2007.
- [5] Chris Callison Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar F. Zaidan. Findings of the 2010 Joint Workshop on Statistical Machine Translation and Metrics for Machine Translation. *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics/MATR*, pages 17–53, 2010.
- [6] Christoph Dalitz, Michael Droettboom, Bastian Pranzas, and Ichiro Fujinaga. A Comparative Study of Staff Removal Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):753–766, May 2008.
- [7] Donald Byrd and Jakob Grue Simonsen. Towards a Standard Testbed for Optical Music Recognition: Definitions, Metrics, and Page Images. *Journal of New Music Research*, 44(3):169–195, 2015.
- [8] Alicia Fornés, Josep Lladós, Gemma Sánchez, and Horst Bunke. Writer Identification in Old Handwritten Music Scores. *Proceedings of Eighth IAPR International Workshop on Document Analysis Systems*, pages 347–353, 2008.
- [9] Alicia Fornés, Anjan Dutta, Albert Gordo, and Josep Lladós. CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal. *International Journal on Document Analysis and Recognition (IJ DAR)*, 15(3):243–251, 2012.
- [10] Graham Jones, Bee Ong, Ivan Bruno, and Kia Ng. Optical Music Imaging: Music Document Digitisation, Recognition, Evaluation, and Restoration. *Interactive Multimedia Music Technologies*, pages 50–79, 2008.
- [11] Jorge Calvo-Zaragoza and Jose Oncina. Recognition of Pen-Based Music Notation: The HOMUS Dataset. *22nd International Conference on Pattern Recognition*, Aug 2014.
- [12] Ian Knopke and Donald Byrd. Towards Musicdiff : A Foundation for Improved Optical Music Recognition Using Multiple Recognizers. *International Society for Music Information Retrieval Conference*, 2007.
- [13] Alon Lavie and Abhaya Agarwal. Meteor: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, 2007.
- [14] Matouš Macháček and Ondřej Bojar. Results of the WMT14 Metrics Shared Task. *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 293–301, 2014.
- [15] Matouš Macháček and Ondřej Bojar. Evaluating Machine Translation Quality Using Short Segments Annotations. *The Prague Bulletin of Mathematical Linguistics*, 103(1), Jan 2015.
- [16] Michael Droettboom and Ichiro Fujinaga. Microlevel groundtruthing environment for OMR. *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 497–500, 2004.
- [17] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, 2002.
- [18] Pierfrancesco Bellini, Ivan Bruno, and Paolo Nesi. Assessing Optical Music Recognition Tools. *Computer Music Journal*, 31(1):68–93, Mar 2007.
- [19] Craig Sapp. OMR Comparison of SmartScore and SharpEye. <https://ccrma.stanford.edu/~craig/mro-compare-beethoven>, 2013.
- [20] Mariusz Szwoch. Using MusicXML to Evaluate Accuracy of OMR Systems. *Proceedings of the 5th International Conference on Diagrammatic Representation and Inference*, pages 419–422, 2008.
- [21] Victor Padilla, Alan Marsden, Alex McLean, and Kia Ng. Improving OMR for Digital Music Libraries with Multiple Recognisers and Multiple Sources. *Proceedings of the 1st International Workshop on Digital Libraries for Musicology - DLfM '14*, pages 1–8, 2014.

## 6.5 A Case for Intrinsic Evaluation of Optical Music Recognition

Jan Hajič jr.: *A Case for Intrinsic Evaluation of Optical Music Recognition. Proceedings of the 1st International Workshop on Reading Music Systems*, Paris, France, pp. 15–16, 2018.

The short position paper *A Case for Intrinsic Evaluation of Optical Music Recognition* complements the previous article with a better understanding of the problems in OMR evaluation, providing a better delineation of the contribution of the previous article, but otherwise this is not a substantial work. The contribution of the thesis author is 100% of the article.



# A Case for Intrinsic Evaluation of Optical Music Recognition

Jan Hajič jr.

Institute of Formal and Applied Linguistics

Charles University

Email: hajicj@ufal.mff.cuni.cz

**Abstract**—Evaluating Optical Music Recognition (OMR) has long been an acknowledged sore spot of the field. This short position paper attempts to bring some clarity to what are actually open problems in OMR evaluation: a closer look reveals that the main problem is finding an edit distance between some practical representations of music scores. While estimating these editing costs in the transcription use-case of OMR is difficult, I argue that the problems with modeling the subsequent editing workflow can be de-coupled from general OMR system development using an intrinsic evaluation approach, and sketch out how to do this.

## I. WE NEED A MUSIC SCORE EDIT DISTANCE

Optical Music Recognition (OMR) has a known problem with evaluation [1]–[3]. We can approach OMR evaluation from two angles: extrinsic and intrinsic. By *extrinsic*, we mean evaluation in application contexts: how well does an OMR system address a specific need (such as retrieval, transcription, playback, ...)? *Intrinsic* evaluation asks a different question: how much of the information encoded by the music score has a given OMR system recovered? An example of extrinsic OMR evaluation can be found, e.g., in [4], where OMR is evaluated in the context of a cross-modal retrieval system; (partial) intrinsic evaluation is done i.a. in [5], where pitches and durations of recognized notes are counted against ground truth data. In this short position paper, I assess what the outstanding problems in evaluating OMR are, and propose intrinsic evaluation as a sensible way forward for OMR research.

The major problem in OMR evaluation is that given a ground truth encoding of a score and the output of a recognition system, there is no automatic method capable of reliably computing how well the recognition system performs that would (1) be rigorously described and evaluated, (2) have a public implementation, (3) give meaningful results. Other applications such as retrieval or extracing MIDI can be evaluated using more general methodologies. E.g., when using OMR to retrieve music scores, there is little domain-specific to defining success compared to retrieving other documents; any time MIDI output is required, metrics used to evaluate multi-f0 estimation can be adapted; score following has well-defined evaluation metrics at different levels of granularity as well. Within the traditional OMR pipeline [6], the partial steps (such as symbol detection) also can use more general evaluation metrics. However, when OMR is applied to re-

typesetting music (which is arguably its original motivation), no evaluation metric is available.

In fact, computing an “edit distance” between a ground truth representation of a full music score and OMR output may be the only evaluation scenario where satisfactory measures are not available. The notion of “edit cost” [7] or “recognition gain” [8] that defines success in terms of how much time a human editor saves by using an OMR system is yet more problematic, as it depends on the specific toolchain used.

What can be done? One can try and implement such a metric. However, because cost-to-correct depends on the toolchains music editors use to work with OMR outputs, developing extrinsic evaluation metrics of OMR for transcription would require user studies at a scale which is not feasible for the few active OMR researchers. For these reasons, we argue it would be helpful for OMR development to have an *intrinsic* evaluation metric. After all, why address individual concerns that OMR users may have when full-pipeline OMR does have the potential to address *all* the application scenarios of OMR, as it attempts to extract *all* the information available from a music score?

## II. MUSIC NOTATION FORMATS ARE PROBLEMATIC

A part of the edit distance problem lies in the ways music notation is stored digitally. MusicXML or MEI, which represent current best practices in open-source formats of digital representation of music scores, have some properties that make it difficult to compute a useful edit distance between two such files (useful in the sense that it would measure either the amount of errors that an OMR system made, or the actual difficulty of changing one score to the other). Furthermore, the formats can encode the same score in multiple ways – e.g., MusicXML stores scores either measure-wise, or voice-wise.

Next, both formats are designed top-down, as trees that represent in their nodes both abstract concepts like a voice or note and graphical entities such as stems or beams. This implies that they cannot represent partial recognition results, and cannot encode syntactically incorrect notation. Furthermore, while the hierarchical structure mostly reflects the abstract structures of music such as voices and measures, it does not reflect the structure of music *notation*: local changes in the score can lead to several changes in the encoding that occur far apart, and vice versa. This is an inherent limitation of their tree structure.

The LilyPond format is impractical for anything but attempts at end-to-end OMR, as it hides much of the graphical representation in its engraving engine, and has so many ways of representing the same music that it is hard to meaningfully compare LilyPond files. The MuNG format [3] does to some extent overcome this locality problem by assuming a directed acyclic graph instead of a tree structure, but it is limited to OMR ground truth and lacks conversions to other formats than MIDI.

The lesson here is that one should not bind intrinsic OMR evaluation to specific notation formats. After all, these formats change much faster than music notation itself. Rather, an evaluation metric should focus on inherent properties of music notation.

### III. ARGUING FOR INTRINSIC EVALUATION

Intrinsic evaluation of OMR systems means to answer the question “*How good is this system?*” without having to add, “*for this specific purpose?*” – thus de-coupling research of OMR methods from their individual use-cases, including the problematic score transcription. After all, music notation is the same regardless of whether it is being recognized for the purpose of searching a database or for producing a digital edition of the score.

There is no reason why this should not be possible: there is a finite amount of information that a music document carries, which can be exhaustively enumerated. It follows that we should be able to measure what proportion of this information our systems recover correctly. The benefit of intrinsic evaluation would be shedding the burden of accounting for score editing toolchains, independence on problematic music notation formats used in broad practice, and a clearly interpretable automatic metric for guiding OMR development (and potentially usable as a differentiable loss function for training full-pipeline end-to-end machine learning-based systems).

### IV. A ROADMAP

What would such an intrinsic evaluation metric measure? At the fullest, we expect two classes of outputs from an OMR system. First, a digital re-encoding of the score itself — creating a digital document that would convey exactly the same to a reader as the original. Second, recovering the semantic musical information: primarily the pitches, durations and onsets of notes (the minimum to build a MIDI representation of the given composition).

A thorough definition of error types in OMR was done by Bellini et al. [8]. They ask human evaluators to count errors for individual symbol types, and what they call “high-level” mistakes: pitch and duration attributes of note symbols. This seems like a good starting point from which to develop an automated intrinsic OMR evaluation metric.

The reason why [8] do not automate error-counting was a (then) lack of ground truth data. This has now been alleviated by the DeepScores dataset [9] at the low level, and MUSCIMA++ dataset [3] at both levels. The other step to automating the metric of [8] is aligning the recognition output

and the ground truth score. At the graphical level, where the outputs are in principle symbol and their relationships, success can be measured using some graph similarity metric. At the semantic level, distance on lists of (onset, duration, pitch) triplets would be conditioned on some optimal alignment; DTW seems like a possible starting point for tractably finding this alignment, as it harshly penalizes ordering errors, which are rather critical due to the sequential nature of music. Given that noteheads can be thought of as carriers of the semantic information within the graphical level, the graph alignment function can also be used to directly find corresponding semantic triplets.

### V. FINALLY

I hope this short paper will inspire discussion on the merits of intrinsic evaluation of OMR (I am especially keen to find out how I am wrong!), and perhaps nudge along the musical score edit distance problem that has been a thorn in the side of OMR research for the duration of its existence.

### ACKNOWLEDGMENTS

This work is supported by the Czech Science Foundation, grant P103/12/G084, the Charles University Grant Agency, grants 1444217 and 170217, and by SVV project 260 453.

### REFERENCES

- [1] M. Szwoch, “Using MusicXML to Evaluate Accuracy of OMR Systems,” *Proceedings of the 5th International Conference on Diagrammatic Representation and Inference*, pp. 419–422, 2008. [Online]. Available: [http://dx.doi.org/10.1007/978-3-540-87730-1\\_53](http://dx.doi.org/10.1007/978-3-540-87730-1_53)
- [2] Donald Byrd and Jakob Grue Simonsen, “Towards a Standard Testbed for Optical Music Recognition: Definitions, Metrics, and Page Images,” *Journal of New Music Research*, vol. 44, no. 3, pp. 169–195, 2015. [Online]. Available: <http://dx.doi.org/10.1080/09298215.2015.1045424>
- [3] J. Hajič jr. and P. Pecina, “The MUSCIMA++ Dataset for Handwritten Optical Music Recognition,” in *14th International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 13 - 15, 2017*, Dept. of Computer Science and Intelligent Systems, Graduate School of Engineering, Osaka Prefecture University. New York, USA: IEEE Computer Society, 2017, pp. 39–46.
- [4] S. Balke, S. P. Achankunju, and M. Müller, “Matching Musical Themes based on noisy OCR and OMR input,” pp. 703–707, 2015.
- [5] Victor Padilla, Alan Marsden, Alex McLean, and Kia Ng, “Improving OMR for Digital Music Libraries with Multiple Recognisers and Multiple Sources,” *Proceedings of the 1st International Workshop on Digital Libraries for Musicology - DLfM '14*, pp. 1–8, 2014. [Online]. Available: <http://dx.doi.org/10.1145/2660168.2660175>
- [6] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R. S. Marcal, Carlos Guedes, and Jaime S. Cardoso, “Optical Music Recognition: State-of-the-Art and Open Issues,” *Int J Multimed Info Retr*, vol. 1, no. 3, pp. 173–190, Mar 2012. [Online]. Available: <http://dx.doi.org/10.1007/s13735-012-0004-6>
- [7] J. Hajič jr., J. Novotný, P. Pecina, and J. Pokorný, “Further Steps towards a Standard Testbed for Optical Music Recognition,” in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, M. Mandel, J. Devaney, D. Turnbull, and G. Tzanetakis, Eds., New York University. New York, USA: New York University, 2016, pp. 157–163. [Online]. Available: [https://18798-presscdn-pagely.netdna-ssl.com/ismir2016/wp-content/uploads/sites/2294/2016/07/289\\_Paper.pdf](https://18798-presscdn-pagely.netdna-ssl.com/ismir2016/wp-content/uploads/sites/2294/2016/07/289_Paper.pdf)
- [8] Pierfrancesco Bellini, Ivan Bruno, and Paolo Nesi, “Assessing Optical Music Recognition Tools,” *Computer Music Journal*, vol. 31, no. 1, pp. 68–93, Mar 2007. [Online]. Available: <http://dx.doi.org/10.1162/comj.2007.31.1.68>
- [9] Lukas Tuggener, Ismail Elezi, Jürgen Schmidhuber, Marcello Pelillo, and Thilo Stadelmann, “DeepScores - A Dataset for Segmentation, Detection and Classification of Tiny Objects,” *CoRR*, vol. abs/1804.00525, 2018. [Online]. Available: <http://arxiv.org/abs/1804.00525>

# 7. Methods

## 7.1 Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression

Jan Hajič jr. and Pavel Pecina. Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression. *CoRR*, 2017. arXiv:1708.01806

(Note: The short paper was presented as a *poster* at the Digital Libraries for Music 2017 workshop in Shanghai, China; originally, it was submitted as a short paper, but was shifted to the poster section from the main program as too technical for oral presentation to the broader workshop audience. DLFM posters are not listed in the proceedings, hence the arXiv citation.)

The short paper *Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression* starts the effort to detect music notation objects, and chooses to focus on noteheads, arguing that OMR can be re-formulated to a significant extent as (1) detecting noteheads, (2) assigning properties to noteheads based on their local neighborhood, and therefore noteheads are the key and (almost) only object that has to be detected directly from the page. Then-state of the art object detection with Region Proposal Networks, specifically R-CNN, was adapted for the purposes of OMR. The dissertation author did all the work on the paper; the co-author contributed to the final text of the article with his comments. The contribution of the dissertation author is about 95% of the article.

# Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression

Jan Hajič jr.

Institute of Formal and Applied Linguistics  
Faculty of Mathematics and Physics  
Charles University  
Email: hajicj@ufal.mff.cuni.cz

Pavel Pecina

Institute of Formal and Applied Linguistics  
Faculty of Mathematics and Physics  
Charles University  
Email: pecina@ufal.mff.cuni.cz

**Abstract**—Noteheads are the interface between the written score and music. Each notehead on the page signifies one note to be played, and detecting noteheads is thus an unavoidable step for Optical Music Recognition. Noteheads are clearly distinct objects; however, the variety of music notation handwriting makes noteheads harder to identify, and while handwritten music notation symbol *classification* is a well-studied task, symbol *detection* has usually been limited to heuristics and rule-based systems instead of machine learning methods better suited to deal with the uncertainties in handwriting. We present ongoing work on a simple notehead detector using convolutional neural networks for pixel classification and bounding box regression that achieves a detection f-score of 0.97 on binary score images in the MUSCIMA++ dataset, does not require staff removal, and is applicable to a variety of handwriting styles and levels of musical complexity.

## I. INTRODUCTION

Optical Music Recognition (OMR) attempts to extract musical information from its written representation, the musical score. Musical information in Western music means an arrangement of *notes* in musical time.<sup>1</sup> There are many ways in which music notation may encode an arrangement of notes, but an elementary rule is that one note is encoded by one *notehead*.<sup>2</sup>

Given the key role noteheads play, detecting them – whether implicitly or explicitly – is unavoidable for OMR. At the same time, if one is concerned only with replayability and not with re-printing the input, noteheads are one of the few music notation symbols that truly need detecting (i.e., recovering their existence and location) in the score: most of the remaining musical information can then be framed in terms of classifying the noteheads with respect to their properties such as pitch or duration; bringing one to the simpler territory of music notation symbol classification.

Music notation defines noteheads so that they are quickly discernible, and from printed music, detecting noteheads has been done using segmentation heuristics such as projections

<sup>1</sup>Here, the term “note” signifies the *musical* object defined by its pitch, duration, strength, timbre, and onset; not the written objects: quarter-note, half-note, etc.

<sup>2</sup>An exception would be “repeat” and “tremolo” signs in orchestral notation. Trills and ornaments only seem like exceptions if one thinks in terms of MIDI; from a musician’s perspective, they simply encode some special execution of what is conceptually one note.



Fig. 1: The variety of noteheads and handwriting styles: full, empty, and grace noteheads.

[1], [2] or morphological operators [3], [4]. However, in handwritten music, noteheads can take on a variety of shapes and sizes, as illustrated by fig. 1, and handwriting often breaks the rules of music notation topology: noteheads may overlap (or separate from) symbols against the rules. Robust notehead detection in handwritten music thus invites machine learning.

**Our contribution is a simple handwritten notehead detector**, which achieves a detection performance of 0.97 on binary images across scores of various levels of musical complexity and handwriting styles. At the heart of the detector is a small convolutional neural network based on the RCNN family of models, specifically Faster R-CNN [?]. Within the traditional OMR pipeline as described by Rebelo et al. [5], our work falls within the symbol recognition stage, specifically as a crucial subset of the “isolating primitive elements” and jointly “symbol classification” steps; however, it does *not* require staff removal. In the following sections, we describe the detector in detail, demonstrate its performance in an experimental setting, describe its relationship to previous work, and discuss its limitations and how they can be overcome.

## II. NOTEHEAD DETECTOR

The notehead detection model consists of three components: a **target pixel generator** that determines which regions the detector should attend to, a **detection network** that jointly decides whether the target pixel belongs to a notehead and predicts the corresponding bounding box, and an additional **proposal filter** that operates on the combined predictions of the detection network and decides whether the proposed bounding boxes really correspond to a notehead.



### A. Target Pixel Generator

The target pixel generator takes a binary score image and outputs a set of  $X, t$  pairs, where  $X$  is the input for the detection network corresponding to the location  $t = (m, n)$  of a *target pixel*. From training and validation data, it also outputs  $y = (c, b)$  for training the detection network. The class  $c$  is 1 if  $t$  lies in a notehead and 0 otherwise;  $b$  encodes the bounding box of the corresponding notehead relative to  $t$  if  $c = 1$  (all values in  $b$  are non-negative; they are interpreted as distance from  $t$  to the top edge of the bounding box of the notehead, to its left edge, etc.); if  $c = 0$ ,  $b$  is set to  $(0, 0, 0, 0)$ . The network outputs are described by Fig. 2.

The detection network input  $X$  is a patch of the image centered on  $t$ . The patch must have sufficient size to capture enough context around a given target pixel, to give the network a chance to implicitly operate with rules of music notation, e.g. to react to the presence of a stem or a beam in certain positions. We set the patch size to  $101 \times 101$  (derived from  $1.2 * \text{staff\_height}$ ), and downscale to  $51 \times 51$  for speed.

At runtime, we use all pixels of the morphological skeleton<sup>3</sup> as target pixels  $t$ . If one correctly classifies the skeleton pixels and then dilates these classes back over the score, we found over 97 % of individual foreground pixels classified correctly (measured on the MUSCIMA++ dataset [6]), making the skeleton a near-lossless compression of the score to about 10 % of the original foreground pixels.

For *training*, we randomly choose  $k$  target pixels for each musical symbol in the training set, from the subset of the skeleton pixels that lies within the given symbol. In non-notehead symbols, we forbid extracting skeleton pixels that are shared with overlapping noteheads: we simply want to know whether a given pixel  $t$  belongs to a notehead or not. (This is most pronounced in ledger lines crossing noteheads.) Setting  $k > 1$  did not improve detection performance; we suspect this is because all  $X$ s from a symbol  $S$  are highly correlated and therefore do not give the network much new information.

### B. Detection Network

The detection network handles most of the “heavy lifting”. It is a small convolutional network with two outputs: a binary classifier (notehead-or-not) and a bounding box regressor. The inputs to the network are the patches  $X_1 \dots X_N$  extracted by the target pixel generator; the ground truth for training are the class and bounding box information  $y_1 = (c_1, b_1), \dots, y_N$  described in II-A. This follows the architecture of Faster R-CNN [7], but as our inputs are not the natural images on which VGG16 [8] was trained, we train our own convolutional stack. (See Sec. IV for a more thorough comparison.)

Our network has four convolutional layers, with the first two followed by max-pooling with pool size  $2 \times 2$ . Dropout is set to 0.25 after the max-pooling layers and 0.125 after the remaining convolutional layers. The output of the fourth convolutional layer is then densely connected to the classification output and the bounding box regression outputs. The convolutional

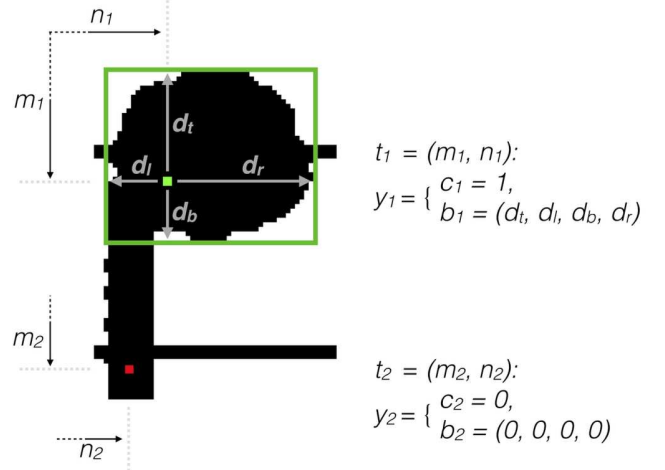


Fig. 2: The outputs which the detection network is learning for each target pixel, the green  $t_1$  and red  $t_2$ : its class  $c$ , and the position  $b$  of the target pixel inside the notehead’s bounding box – set to all zeros when  $t$  does not belong to a notehead, as seen in  $y_2$ .

layers use *tanh* activation rather than ReLU: we found that this made learning converge faster, although we are still unsure why. Details are given in table ??.

The classification output uses cross-entropy loss; the bounding box regression output uses mean squared error loss, weighted at 0.02 of the classification loss. The network was implemented using the Keras library [9].

### C. Notehead Proposal Filter

The detection network outputs correspond to individual target pixels, selected by the generator; we now combine these results into noteheads.

We take the union of all bounding boxes output by the detection network for target pixels with predicted class  $c = 1$ , and we use bounding boxes of the connected components of this union as notehead proposals. We then train a classifier of notehead proposals. This classifier can take into account all the network’s decisions, as well as other global information; however, it can only fix false positives, – if the network misses a notehead completely, the filter cannot find it. However, the detection network in the described setting achieves good recall and has more trouble with precision, so such filtering is appropriate.

The features we extract for proposal filtering from each proposal region  $B_1, \dots, B_j$  are:

- $h(B)$ , the height of  $B$ ,
- the ratio  $\hat{h}(B)$  of  $h(B)$  to the average height of  $B_1, \dots, B_j$ ,
- $w(B)$ , the width of  $B$ , and analogously  $\hat{w}(B)$ ,
- area  $a(B) = h(B) * w(B)$ , analogously  $\hat{a}(B)$ ,
- the no. of foreground pixels in  $B$ :  $N_{fg}(B)$ , and the proportion  $p_{fg}(B) = N_{fg}(B)/a(B)$ ,

<sup>3</sup>As implemented by the *skimage* Python library.



- $N^+(B)$ , the no. of positively classified target pixels  $t^+ \in B$ ,
- $p^+(B)$ , the proportion of such target pixels to all in  $t^+ \in B$ ,
- the ratio  $\hat{N}^+(B)$ ,
- equiv. for non-notehead pixels  $t^- \in B$ :  $N^-(B)$ ,  $p^-(B)$ ,  $\hat{N}^-(B)$ ,
- "soft" sum of noteheadedness:  $S^+(B) = \sum_{t^+ \in B} P(+ | t)$ ,
- again, the ratio to the average  $S^+(B)$  in the image:  $\hat{S}^+(B)$ .
- $l(B)$ : how much to the left in the input image  $B$  is.

The ratio features ( $\hat{w}(B)$ , etc.) are designed to simulate invariance to individual handwriting styles. Also, beyond features based on detection network outputs, the left-ness  $l(B)$  is used to find false positives in clefs.

For training the proposal filter, we consider correct each notehead proposal that has Intersection-over-Union (IoU) with a true notehead above 0.5. We use a Random Forest with 300 estimators, a maximum depth of 8, and a minimum of 3 samples in each leaf.

### III. EXPERIMENTS

We now describe the experimental setup in which the detector was tested: the dataset, evaluation procedure, and experimental results.

#### A. Data

For experiments, we use the MUSCIMA++ dataset of Hajič jr. and Pecina [6], based on the underlying images of CVC-MUSCIMA by Fornés et al. [10]. The dataset contains 140 binary images. There are 20 pages of music, each as transcribed by 7 of the 50 writers of CVC-MUSCIMA; all the 50 CVC-MUSCIMA writers are represented in MUSCIMA++. The scores all use the same staffline and staffspace heights (see [10] for details). We use a test set that contains one of each of the 20 pages, chosen so that no page by the writers of the test set pages is seen in the training set (we first want to see how the system generalizes to unseen handwriting style, rather than unseen notational situations). When extracting ground truth, we did not differentiate between different noteheads (full, empty, grace-note).

We used the first 100 of the remaining 120 images as the training set for the detection network, and the other 20 as the validation set. As we are training on only one sample target pixel per musical symbol, this amounted to 65015 training instances. Using the Adam optimizer, training converged after 8-9 epochs.

The outputs of the network on the dev set were then included for training the notehead proposal filter, together with the first 50 images from the training set. (The dev set better approximates the inputs to the proposal filter at runtime conditions, when the detection network runs on images never seen in training.)

Layer	Dropout	Activation	Size
conv1	–	tanh	32 filters 5x5
pool1	0.25	–	pool 2x2
conv2	–	tanh	64 filters 3x3
pool2	0.25	–	pool 2x2
conv3	0.125	tanh	64 filters 3x3
conv4	0.125	tanh	64 filters 3x3
clf.	–	sigmoid	1
bb. reg.	–	ReLU	4

TABLE I: Detection Network Architecture. (Both the classification and bounding box regression output layers are densely connected to conv4.)

#### B. Evaluation Procedure

We evaluate notehead detection recall and precision. A notehead prediction that has IoU over 0.5 with a true notehead is a hit. Furthermore, we count each predicted notehead that completely contains a true notehead. This non-standard way of counting hits was chosen because in some cases, the bounding box regression produced bounding boxes that were symmetrically "around" the true notehead, but slightly too large, to the extent that it set IoU too low due to the predicted notehead's contribution to the union term. However, a symmetrically larger bounding box (when oversized only to the limited extent present in the model outputs) does not impede recovering the notehead's relationship to other musical symbols, e.g., stafflines, and this adjustment should therefore give the reader a better grasp of the detector's actual useful performance.<sup>4</sup>

#### C. Results

**On average, the detector achieves a recall of 0.96 and precision of 0.97.** Among the test set, there were two images where recall fell to around 0.9: 0.87 for CVC-MUSCIMA image W-12\_N-19 (writer 12, page 19), and 0.91 for W-29\_N-10, due to the detection network's errors on empty noteheads in the middle of chords, full noteheads in chords with a handwriting style where the notehead is essentially just a thickening or straight extension of the stem, and certain grace notes; there are also problems with whole notes on the "wrong" side of the stem in W-39\_N-20. Aside from these situations, the detector rarely misses a note.<sup>5</sup>

The detection network itself has an average *pixel-wise* recall on the positive class of 0.94 (again, the average is lowered mostly by the three problematic images, rather than evenly distributed errors), but precision only 0.78 (even though most of the false positives are skeleton pixels in the close vicinity of actual noteheads). The *notehead detection* recall without post-filtering is 0.97 and precision is 0.81. As the false positives are clearly a much greater problem than false negatives, preliminary results to this effect on the development set motivated work on the post-filtering step. The post-filter increases detection precision by 0.16, eliminating over 84 %

<sup>4</sup>Technically, this adjustment moved recall upwards by 3 - 5 % across all images.

<sup>5</sup>Visualizations of results for the test set images are available online: <https://drive.google.com/open?id=0B9l5xUyYe-f8Y2FQWTZxc09PaEE>

of all the false positives, while only introducing 1 % more false negatives.

#### IV. RELATED WORK

Given that there was little publicly available ground truth for notehead detection until recently [6], it is hard to compare results to previous work directly. A noteworthy approach on the same CVC-MUSCIMA handwritten data was taken by Baro et al. [4]. They achieve a notehead detection f-score of 0.64 based on handcrafted rules alone, without any machine learning. This is an indication that contemporary handwritten music will need a machine-learning approach rather than the projection-based heuristics that have been used in printed music [1], [2] and applied to handwritten early music scores with recall 0.99 and precision 0.75 [3].

Convolutional neural networks have been previously successfully applied to music scores by Calvo-Zaragoza et al., for segmentation into staffline, notation, and text regions [11] or binarization [12], with convincing results that generalize over various input modes.

Our detector is inspired by the RCNN family of models, especially Faster R-CNN [7]. RCNNs were motivated by the fact that detection can be decomposed into region proposals and classification, with models such as VGG16 [8] for natural image classification obtaining near-human performance. However, the pre-trained image classification nets are too slow for a trivial sliding window approach. RCNNs use a sparse grid of proposal regions with pre-defined sizes and shapes, and train bounding box regression to locate the object of interest within the proposal region. (Faster R-CNN trains bounding box regression directly on top of the high-quality image classification features.) When combining predictions to obtain detection outputs, RCNN models then apply non-maximum suppression on the detection probability landscape obtained from predictions for each of the pre-defined proposal regions.

Together with [7], we apply joint classification and bounding box regression, but our approach differs from RCNNs in four aspects. We do not use a fixed proposal grid but generate proposal regions dynamically from the input image. Second, we cannot reuse VGG16 [8] or other powerful pre-trained image classification models for feature extraction, since they not trained on music notation data; however, because our input space is much simpler, we can train the convolutional layer stack directly. Third, we use a separate classifier to take advantage of the network outputs for related proposal regions, combining the network's "votes" on multiple closely related inputs more generally than simply non-maxima suppression. A final subtle distinction is that we are not just looking for a notehead anywhere in the proposal region; we want the *center pixel* of the region to be part of the notehead, constraining bounding box regression outputs roughly to the average size of a notehead even with a much larger input patch.

#### V. DISCUSSION AND CONCLUSIONS

We proposed an accurate notehead detector from simple image processing and machine learning components. However, ongoing work on the detector will need to address several limitations.

Our system requires binary images. The detection network can be trained on augmented grayscale data, and given the track record of convolutional networks, one would expect good performance; however, an alternate target pixel selection mechanism is needed.

A second problem is slow runtime: over a minute per MUSCIMA++ test set image on a consumer CPU. This can be mitigated by first downsampling the skeleton, and then informing the choice of more target pixels by the results, directing the network to focus only on "hopeful" areas where it has detected a notehead.

While the binary nature of the task is appealing, the network is in fact forced to lump different symbols together. This is more pronounced in the negative class, where the variety of shapes is larger. Saliency maps for the last convolutional layer suggest that most of its filters relate to the presence of a stem; forcing the network to discriminate among more classes might force convolutional filters to distinguish specifically between noteheads and similar objects.

The final issue is generalizing past the high-quality scans of CVC-MUSCIMA images. In preliminary experiments on an early music page with low-quality binarization and some blurring and deformation, the detector gets f-score 0.85 without and 0.79 with post-filtering. The more fragile stages of the detector are at fault – a low-quality skeleton, and the correspondingly uncertain inputs for the post-filtering classifier. (The post-filtering classifier is fragile in the sense that its features are directly derived from the combined outputs of the detection network, and thus it is "conditioned" for a certain level of network performance.)

These limitations suggest a way forward: more efficient target pixel selection, applicable to grayscale images; data augmentation to simulate more real-world conditions; a more robust post-filtering step, ideally trainable jointly with the detection network; and extending the detection network to multiple output classes (which, when combined with using previous outputs in the vicinity of a given target pixel as an input to the network, can also incorporate music notation syntax more explicitly).

The simple detector has proven to be quite powerful, resistant to changes in handwriting style and most notation complexity, showcasing the potential of quite simple neural networks (and the value of a dataset). In spite of the limitations, we find this an encouraging result for offline handwritten OMR.

#### ACKNOWLEDGMENTS

This work is supported by the Czech Science Foundation, grant number P103/12/G084, the Charles University Grant Agency, grants number 1444217 and 170217, and by SVV project number 260 453.

## REFERENCES

- [1] I. Fujinaga, "Optical Music Recognition using Projections," Master's thesis, 1988.
- [2] P. Bellini, I. Bruno, and P. Nesi, "Optical music sheet segmentation," in *Proceedings First International Conference on WEB Delivering of Music. WEDELMUSIC 2001*. Institute of Electrical & Electronics Engineers (IEEE), 2001, pp. 183–190. [Online]. Available: [papers2://publication/uuid/FEF468FA-2244-48DF-94C6-64246D675F15](https://publication/uuid/FEF468FA-2244-48DF-94C6-64246D675F15)
- [3] A. Fornés, "Primitive Segmentation in Old Handwritten Music Scores," pp. 279–290, 2006.
- [4] Arnau Baro, Pau Riba, and Alicia Fornés, "Towards the Recognition of Compound Music Notes in Handwritten Music Scores," in *15th International Conference on Frontiers in Handwriting Recognition, ICFHR 2016, Shenzhen, China, October 23-26, 2016*. IEEE Computer Society, 2016, pp. 465–470. [Online]. Available: <http://dx.doi.org/10.1109/ICFHR.2016.0092>
- [5] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R. S. Marcal, Carlos Guedes, and Jaime S. Cardoso, "Optical Music Recognition: State-of-the-Art and Open Issues," *Int J Multimed Info Retr*, vol. 1, no. 3, pp. 173–190, Mar 2012. [Online]. Available: <http://dx.doi.org/10.1007/s13735-012-0004-6>
- [6] J. Hajić, jr. and P. Pecina, "In Search of a Dataset for Handwritten Optical Music Recognition: Introducing MUSCIMA++," *ArXiv e-prints*, Mar. 2017.
- [7] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, pp. 91–99. [Online]. Available: <http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks>
- [8] Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [9] François Chollet, "Keras," <https://github.com/fchollet/keras>, 2017. [Online]. Available: <https://github.com/fchollet/keras>
- [10] A. Fornés, A. Dutta, A. Gordo, and J. Llads, "CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal," *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 15, no. 3, pp. 243–251, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s10032-011-0168-2>
- [11] J. Calvo Zaragoza, A. Pertusa, and J. Oncina, "Staff-line detection and removal using a convolutional neural network," *Machine Vision and Applications*, pp. 1–10, 2017. [Online]. Available: <http://dx.doi.org/10.1007/s00138-017-0844-4>
- [12] J. Calvo Zaragoza, G. Vigliensoni, and I. Fujinaga, "A machine learning framework for the categorization of elements in images of musical documents," in *Third International Conference on Technologies for Music Notation and Representation*. A Coruña: University of A Coruña, 2017. [Online]. Available: <http://grfia.dlsi.ua.es/repositori/grfia/pubs/360/tenor-unified-categorization.pdf>



## 7.2 On the Potential of Fully Convolutional Neural Networks for Musical Symbol Detection

Matthias Dorfer, Jan Hajič jr. and Gerhard Widmer. On the Potential of Fully Convolutional Neural Networks for Musical Symbol Detection. *14th International Conference on Document Analysis and Recognition / GREC*, Kyoto, Japan, pp. 53–54, 2017. ISBN 978-1-5386-3586-5, doi: 10.1109/ICDAR.2017.274.

The short paper *On the Potential of Fully Convolutional Neural Networks for Musical Symbol Detection* uses a different model that addresses some of the problems of the R-CNN based approach of the previous paper in generalizing to other symbol classes<sup>1</sup> and speed: by doing away with all bounding box-related hyperparameters: the fully convolutional network processes the entire image in a single shot.

In this article, the dissertation author contributed the dataset and wrote most of the text of the article; the first author used a previous implementation of FCNs on the MUSCIMA++ dataset. The contribution of the dissertation author to the article is about 30–40%.

---

<sup>1</sup>Unpublished experiments.

# On the Potential of Fully Convolutional Neural Networks for Musical Symbol Detection

Matthias Dorfer<sup>1</sup>, Jan Hajič, jr.<sup>2</sup> and Gerhard Widmer<sup>1</sup>

**Abstract**—Musical symbol detection on the page is an outstanding Optical Music Recognition (OMR) subproblem. We propose using a fully convolutional segmentation network to produce high-quality pixel-wise symbol probability masks. Experiments on notehead detection show a very promising detection f-score of 0.98 with elementary detection methods.

## I. INTRODUCTION

Optical Music Recognition (OMR), the process of automatically extracting musical information from its visual encoding, music notation, is a long-standing graphics recognition problem, with potential for composers, musicians, and music scholars. It is usually done by (1) preprocessing and binarization, (2) staffline removal, (3) symbol detection (localization and classification), and (4) notation reconstruction [10]. While binarization and staff removal have recently been successfully tackled with convolutional neural networks (CNNs) [3], [7], and symbol classification achieves good results as well [4], [12], symbol localization remains a bottleneck usually addressed without data-driven methods, e.g. with projections [6], [2], or Kalman Filters [1].

Our contribution is applying fully convolutional segmentation networks (FCSN) for joint musical symbol segmentation and classification. These output a pixel-wise symbol probability mask, which is then passed to a detection decision-making step such as non-maxima suppression. FCSNs have recently been successfully applied to staff removal [7]. However, staff removal requires only the staffline mask; to detect symbols, we need to further process the segmentation mask to make decisions about individual symbols. Furthermore, stafflines do not have much visual variety (even though they come in varied surroundings) while, especially in handwritten music, symbols can take on wildly different shapes and sizes. We have begun by examining FCSN performance on notehead detection (noteheads are the critical 1:1 interface from the written score to the encoded music: one note per one notehead), with surprisingly good performance and potential to generalize to other notation symbols.

## II. METHODS

We detect noteheads in images of sheet music in two steps: (1) Predict a pixel-wise notehead probability map

<sup>1</sup>Department of Computational Perception, Johannes Kepler University, 4040 Linz, AT. matthias.dorfer at jku.at

<sup>2</sup>Institute of Formal and Applied Linguistics, Charles University, 118 00 Prague, CZ. hajicj at ufal.mff.cuni.cz Work done while at CP JKU Linz.

by applying a FCSN in sliding window fashion on half-overlapping tiles; (2) based on this probability map we localize noteheads by searching for peaks in these maps.

### A. Fully Convolutional Note Segmentation

Our segmentation network is heavily inspired by the U-Net architecture [13] originally introduced for biomedical image segmentation and successful in various other applications. In our case the network takes images of sheet music as input and is trained to predict a pixel-wise probability for the target structure at hand. In particular, we want the network to assign high probability to notehead pixels (foreground) and low probability to the rest (background). Figure 1 details the architecture used for our experiments. Compared to the

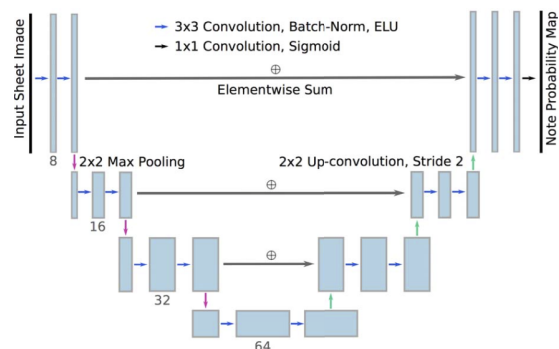


Fig. 1: Fully Convolutional Note Segmentation Network.

original U-Net architecture, we apply some minor modifications. Firstly, we use  $3 \times 3$  convolutions followed by batch normalization and Exponential Linear Units (ELUs). We also replaced the concatenation of feature maps with *element-wise sum skip connections*, which performs similarly but reduces computation time, as the number of feature maps is halved. The network output is computed as a single feature map and post-processed with the sigmoid nonlinearity yielding a pixel-wise note probability map  $P$ .

For training, we give a set of  $N$  score images  $X = \{X_1, \dots, X_N\}$  along with a corresponding set of manually labeled binary mask images  $Y = \{Y_1, \dots, Y_N\}$  for the noteheads. In each epoch, we randomly choose a  $256 \times 512$  window from each  $X_i$  and the corresponding crop from  $Y_i$ . Optimization is then performed by minimizing the mean over the pixel-wise binary cross entropy  $\sum_x \log(p(x))$  between ground truth and network prediction. Updates are computed using ADAM with an initial learning rate of 0.001 and a batch size of 2.



## B. Notehead Localization

Given the (high quality) probability maps produced by the network, we threshold the masks at 0.7 and search for local maxima. In addition, we suppress neighboring local peaks within a region of 10 pixels of a larger maximum. The peaks surviving this suppression are considered centroids of detected noteheads.

## III. EXPERIMENTS AND PRELIMINARY RESULTS

We report preliminary notehead segmentation and recognition results on the MUSCIMA++ dataset of handwritten scores [8]: 140 binarized pages from CVC-MUSCIMA [5], with 91,255 music notation primitives of 107 distinct types, out of which 23,352 are noteheads. We evaluate our proposal on two different scenarios: *writer-dependent*, where the writers of the test set are also represented in the train set, and *writer-independent*, where this is not the case. We train using 100 of the train set images and use 20 images as a validation set. The test sets also comprise 20 sheet music images. We report the Dice coefficient, a measure of overlap between ground truth and the predicted segmentation, after applying an threshold of 0.7; to evaluate notehead detection, we report precision, recall and F-score. Table I summarizes our results for the test sets. Our results slightly outperform the present state-of-the-art on this dataset [9] and the model generalizes well also to unseen writers.<sup>1</sup>

Test Set	Precision	Recall	F-Score	Dice Coef.
writer dependent	0.984	0.970	0.977	83.1
writer independent	0.986	0.970	0.978	83.3

TABLE I: Notehead recognition performance.

## IV. DISCUSSION AND OUTLOOK

In this work we have investigated the potential of FCSNs for musical symbol detection. Besides very promising performance on notehead recognition, we outline some further advantages of the proposed model:

(1) As the particular network architecture requires only one forward pass to produce the note probability map, it is computationally much more efficient than approaches based on proposal regions or pixel classification.

(2) The generality of the model makes the recognition system adaptable to arbitrary musical primitives. Besides staffline removal [7], we have tried detecting bar lines with similarly promising results.

(3) If our system generalizes to handwritten scores when trained on printed sheet music, we would have access to much larger repositories of automatically generated training data.

FCSNs seem to be a promising direction to address the segmentation stage of OMR. However, it is still not a complete detection solution: the network merely provides a symbol probability mask, and it still needs to be decided where

<sup>1</sup>A sample output probability mask and detection result is given as supplementary material.

the actual symbols are. While the probability mask is good enough to apply straightforward non-maxima suppression or thresholding and connected components search, we have encountered some double detections in empty noteheads, and other symbols may easily overlap: noteheads in chords in printed music, or parallel beams in handwritten music. Furthermore, there are symbols that consist of multiple connected components (F-clef, measure-repeat, 16th tremolo, sometimes 8th and higher rests in low-quality images), which the detection heuristics cannot find. It would be ideal to incorporate this functionality directly into the network; however, this requires modifying the model. Nevertheless, the speed, generality, and performance of FCSNs promises very good results whenever training data is available.

## ACKNOWLEDGMENT

J. Hajič acknowledges support by the Czech Science Foundation, grant number P103/12/G084, the Charles University Grant Agency, grants number 1444217 and 170217, and by SVV project number 260 453.

## REFERENCES

- [1] V. d'Andecy, J. Camillerapp, and I. Leplumey, "Kalman filtering for segment detection: application to music scores analysis," in *Proceedings of 12th International Conference on Pattern Recognition*. IEEE Comput. Soc. Press.
- [2] P. Bellini, I. Bruno, and P. Nesi. "Optical music sheet segmentation," *Proceedings First International Conference on WEB Delivering of Music. WEDELMUSIC 2001*. Institute of Electrical & Electronics Engineers (IEEE), pp. 183–190, 2001.
- [3] J. Calvo Zaragoza, G. Vigiensoni, and I. Fujinaga, "A machine learning framework for the categorization of elements in images of musical documents," in *Third International Conference on Technologies for Music Notation and Representation*. A Coruña: Univ. of A Coruña, 2017.
- [4] S. Chanda, D. Das, U. Pal, and F. Kimura. 2014. "Offline Handwritten Musical Symbol Recognition, 2014 14th International Conference on Frontiers in Handwriting Recognition pp. 405–410, 2014.
- [5] A. Fornés, A. Dutta, A. Gordo, and J. Llads, "CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal," *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 15, no. 3, pp. 243–251, 2012.
- [6] I. Fujinaga, "Optical Music Recognition using Projections," Msc. thesis, 1988.
- [7] Antonio-Javier Gallego and Jorge Calvo Zaragoza, "Staff-line removal with selectional auto-encoders," *Expert Systems with Applications*, vol. 89, pp. 138 – 148, 2017.
- [8] J. Hajič, jr. and P. Pecina, "In Search of a Dataset for Handwritten Optical Music Recognition: Introducing MUSCIMA++," *ArXiv e-prints*, CoRR 1703.04824, Mar. 2017.
- [9] J. Hajič jr. and P. Pecina, "Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression," Tech. Rep., Aug 2017. [Online].
- [10] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R. S. Marcal, Carlos Guedes, and Jaime S. Cardoso, "Optical Music Recognition: State-of-the-Art and Open Issues," *Int J Multimed Info Retr*, vol. 1, no. 3, pp. 173–190, Mar 2012.
- [11] Cuihong Wen, Ana Rebelo, Jing Zhang, and Jaime Cardoso. 2015. "A new optical music recognition system based on combined neural network." *Pattern Recognition Letters* 58:1 – 7., 2015.
- [12] Cuihong Wen, Jing Zhang, Ana Rebelo, and Fanyong Cheng. "A Directed Acyclic Graph-Large Margin Distribution Machine Model for Music Symbol Classification" *PLOS ONE* 11(3):e0149688. 2016.
- [13] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, Cham, 2015.

## 7.3 Towards Full-Pipeline Handwritten OMR with Musical Symbol Detection by U-Nets

Jan Hajič jr., Matthias Dorfer, Gerhard Widmer, Pavel Pecina. Towards Full-Pipeline Handwritten OMR with Musical Symbol Detection by U-Nets. *Proceedings of the 19th Conference of the International Society for Music Information Retrieval*, pages 225–232, Paris, France, 2018. ISBN 978-2-9540351-2-3.

The article *Towards Full-Pipeline Handwritten OMR with Musical Symbol Detection by U-Nets* is the cornerstone of the methods section of the thesis, as it builds the full recognition pipeline to MIDI. The pipeline proceeds in three steps: object detection, notation graph assembly, and semantics inference (with MIDI export). Based on the previous two notehead detection approaches, the fully convolutional models (U-Nets) were chosen as a general object detection model, and improved with domain-specific training tricks (convex hull training and multichannel outputs) to boost performance. A baseline notation assembly model was chosen using decision trees, which for each plausible object pair decide whether there should be a notation graph edge connecting the given two objects. Semantics inference from the notation graph is then deterministic. The article then evaluates both the object detection with U-Nets, and the full pipeline: directly the accuracy of the inferred semantics, and in a (small) retrieval context.

In this article, the co-author Matthias Dorfer has trained the object detection models designed earlier, implemented the detector evaluation procedure, and designed the multichannel training trick. The dissertation author has re-implemented the U-Net training procedure and replicated the object detection results<sup>2</sup>, authored the convex hull training trick, designed and implemented the notation assembly step and semantics inference and MIDI export from the MuNG representation, performed the full-pipeline evaluations, and wrote most of the text except for most of section 3 and 4. The co-authors Gerhard Widmer and Pavel Pecina contributed comments and improvements to the text. The contribution of the dissertation author to the article is roughly 65–70%.

---

<sup>2</sup>The original implementation used the theano framework, the re-implementation was done in PyTorch, resulting in significantly simplified deployment and some future-proofing, as theano development has been discontinued.

# TOWARDS FULL-PIPELINE HANDWRITTEN OMR WITH MUSICAL SYMBOL DETECTION BY U-NETS

Jan Hajič jr.<sup>1</sup>    Matthias Dorfer<sup>2</sup>    Gerhard Widmer<sup>2</sup>    Pavel Pecina<sup>1</sup>

<sup>1</sup> Institute of Formal and Applied Linguistics, Charles University

<sup>2</sup> Institute of Computational Perception, Johannes Kepler University

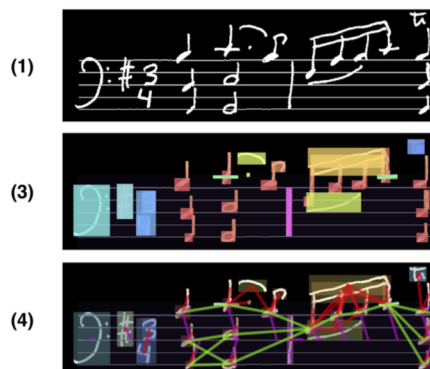
hajicj@ufal.mff.cuni.cz

## ABSTRACT

Detecting music notation symbols is the most immediate unsolved subproblem in Optical Music Recognition for musical manuscripts. We show that a U-Net architecture for semantic segmentation combined with a trivial detector already establishes a high baseline for this task, and we propose tricks that further improve detection performance: training against convex hulls of symbol masks, and multichannel output models that enable feature sharing for semantically related symbols. The latter is helpful especially for clefs, which have severe impacts on the overall OMR result. We then integrate the networks into an OMR pipeline by applying a subsequent notation assembly stage, establishing a new baseline result for pitch inference in handwritten music at an f-score of 0.81. Given the automatically inferred pitches we run retrieval experiments on handwritten scores, providing first empirical evidence that utilizing the powerful image processing models brings content-based search in large musical manuscript archives within reach.

## 1. INTRODUCTION

Optical Music Recognition (OMR), the field of automatically reading music notation from images, has long held the significant promise for music information retrieval of making a great diversity of music available for further processing. More compositions have probably been written than recorded, and more have remained in manuscript form rather than being typeset; this is not restricted to the tens of thousands of manuscripts from before the age of recordings, but holds also for contemporary music, where many manuscripts have been left unperformed for reasons unrelated to their musical quality. Making the content of such manuscript collections accessible digitally and searchable is one of the long-held promises of OMR, and at the same time OMR is reported to be the bottleneck there [17]. On printed music or simpler early music notation, this has been attempted by the PROBADO



**Figure 1.** OMR pipeline in this work. Top-down: (1) input score, (3) symbol detection output, (4) notation assembly output. Obtaining MIDI from output of notation assembly stage (for evaluating pitch accuracy and retrieval performance) is then deterministic. Our work focuses on the symbol detection step (1) → (3); notation reconstruction is done only with a simple baseline.

[17, 28] or SIMSSA/Liber Usualis [3] projects. However, for manuscripts, results are not forthcoming.

The usual approach to OMR is to break down the problem into a four-step pipeline: (1) preprocessing and binarization, (2) staffline removal, (3) symbol detection (localization and classification), and (4) notation reconstruction [2]. Once stage (4) is done, the musical content — pitch, duration, and onsets — can be inferred, and the score itself can be encoded in a digital format such as MIDI, MEI<sup>1</sup> or MusicXML. We term OMR systems based on explicitly modeling these stages *Full-Pipeline OMR*.

Binarization and staff removal have been successfully tackled with convolutional neural networks (CNNs) [4, 11], formulated as semantic segmentation. Symbol classification achieves good results as well [12, 13, 33]. However, detecting the symbols on a full page remains the next major bottleneck for handwritten OMR. As CNNs have not been applied to this task yet, they are a natural choice.

Full-Pipeline OMR is not necessarily the only viable approach: recently, *end-to-end OMR* systems have been proposed. [16, 24]. However, they have so far been limited to short excerpts of monophonic music, and it is not clear how to generalize their output design from MIDI equivalents to

<sup>1</sup> <http://music-encoding.org/>





lossless structured encoding such as MEI or MusicXML, so full-pipeline approaches remain justified.

Our work mainly addresses step (3) of the pipeline, applied in the context of a baseline full-pipeline system, as depicted in Fig. 1. We skip stage (2): we treat staves as any other object, since we jointly segment and classify and do not therefore have to remove them in order to obtain a more reasonable pre-segmentation. We claim the following contributions:

**(1) U-Nets used for musical symbol detection.** Applying fully convolutional networks, specifically the U-Net architecture [38], for musical symbol segmentation and classification, without the need for staffline removal. We apply improvements in the training setup that help overcome OMR-specific issues. The results in Sec. 5 show that the improvements one expects from deep learning in computer vision are indeed present.

**(2) Full-Pipeline Handwritten OMR Baseline for Pitch Accuracy and Retrieval.** We combine our stage (3) symbol detection results with a baseline stage (4) system for notation assembly and pitch inference. This OMR system already achieves promising pitch-based retrieval results on handwritten music notation; to the best of our knowledge, its pitch inference f-score of 0.81 is the first reported result of its kind, and it is the first published full-pipeline OMR system to demonstrably perform a useful task well on handwritten music.

## 2. RELATED WORK

**U-Nets.** U-Nets [38] are fully convolutional networks shaped like an autoencoder that introduce skip-connections between corresponding layers of the downsampling and upsampling halves of the model (see Fig. 2). For each pixel, they output a probability of belonging to a specific class. U-Nets are meant for semantic segmentation, not instance segmentation/object detection, which means that they require an ad-hoc detector on top of the pixel-wise output. On the other hand, this formulation avoids domain-specific hyperparameters such as choosing R-CNN anchor box sizes, is agnostic towards the shapes of the objects we are looking for, and does not assume any implicit priors on their sizes. This promises that the same hyperparameter settings can be used for all the visually disparate classes (the one neuralgic point being the choice of receptive field size). Furthermore, U-Nets process the entire image in a single shot — which is a considerable advantage, as music notation often contains upwards of 500 symbols on a single page. A disadvantage of U-Nets (as well as most CNNs) is their sensitivity to the training data distribution, including the digital imaging process. Because of the variability of musical manuscripts, it is likely real-world applications will require case-specific training data, and data augmentation would therefore be used to mitigate this sensitivity; fortunately, fully convolutional networks are known to respond well to data augmentation over sheet music [30] as well as over other application scenarios [9, 23]. Therefore, we consider this choice reasonable, at the very least to establish a strong baseline for handwritten musical symbol

detection with deep learning.

**Object Detection CNNs.** A standard architecture for object detection is the Regional CNN (R-CNN) family, most notably Faster R-CNN [40] and Mask R-CNN [26]. These networks output probabilities of an object’s presence in each one of a pre-defined sets of anchor boxes, and make the bounding box predictions more accurate with regression. In comparison, the U-Net architecture may have an advantage in dealing with musical symbols that have significantly varying extents, such as beams or stems, as it does not require specifying the appropriate anchor box sizes, and it is significantly faster, requiring only one pass of the network (the detector then requires one connected component search). Furthermore, Faster R-CNN does not output pixel masks, which are useful for archival- and musicology-oriented applications downstream of OMR, such as handwriting-based authorship attribution. Mask R-CNN, admittedly, does not have this limitation, but still requires the same bounding box setup.

Another option is the YOLO architecture [25], specifically the most recent version YOLOv3 [36], which predicts bounding boxes and confidence degrees without the need to specify anchor boxes. A similar approach was proposed in [22], achieving a notehead detection f-score of 0.97, but only with a post-filtering step.

**Convolutional Networks in OMR.** Convolutional networks have been applied in OMR to symbol classification [33], indicating that they can in principle handle the variability of music notation symbols, but not yet in also finding the symbols on the page. Fully convolutional networks have been successfully applied to staff removal [4], and to resolving the document to a background, staff, text, and symbol layers [11]. However, these are semantic segmentation tasks; whereas we need to make decisions about individual symbols. The potential of U-Nets for symbol detection was preliminarily demonstrated on noteheads [22, 31], but compared to other symbol classes, noteheads are “easy targets”, as they look different from other elements, have constant size, and appear only in one pose (as opposed to, e.g., beams).

**OMR Symbol Detection.** Localizing symbols on the page has been previously addressed with heuristics rather than machine learning, e.g. with projections [8, 18], Kalman Filters [14], Line Adjacency Graphs [37], or other combinations of low-level image features [39]. On handwritten music, due to its variability, more complex heuristics such as the algorithm of [1] that consists of 14 interdependent steps have been applied.

**OMR for Content-Based Retrieval.** The idea of using imperfect OMR for retrieval is not new, although originally OMR was attempted in the context of transcribing individual scores. In the PROBADO project [17, 28], an off-the-shelf OMR system was applied to printed Common Western Music Notation (CWMN) scores, allowing retrieval and measure-level score following in a database of 1200 printed scores. The Liber Usualis project at SIMSSA is another such project, on square plainchant notation; it operates at a more fine-grained level that allows for ex-

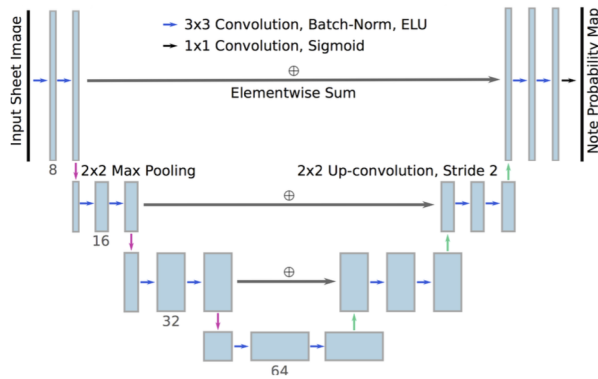


Figure 2. Baseline U-Net model architecture.

ample accurate motif retrieval [3]. However, for CWMN manuscripts, we are not aware of similar experiments.

### 3. MODEL

For all experiments, we use as a basis the same fully convolutional network architecture [38] as shown in Figure 2. There are three down-sampling blocks and three corresponding up-sampling blocks. Each down-sampling block consists of two convolutional layers with batch normalization using the same number of filters; down-sampling is done through 2x2 Max Pooling. After each downsizing step, we use twice the number of filters. The output layer uses sigmoid activation; otherwise, ELU nonlinearity is used. Additionally, we add element-wise-sum residual connections between symmetric layers of the encoder and decoder part of the network.

In the rest of this section, we propose modifications for both architecture and training strategy for symbol detection in handwritten sheet music.

#### 3.1 Convex Hull Segmentation Targets

Our first proposal is to use the convex hull region of individual symbols as a target for training instead of the original segmentation masks. Figure 3 shows an example of the modified training targets. This simple adaptation is an elegant way of dealing with symbols such as f-clefs or c-clefs, which by definition consist of multiple components. As we employ a connected components detector for recognizing the symbols in our experiments in Section 4 we circumvent the need for treating these symbol classes in any special way. This advantage also holds “pre-emptively” for complex symbols which for example contain “holes” and might break up into multiple components after imperfect automatic segmentation, or may be disconnected due to handwriting style (e.g., flats).

#### 3.2 Multichannel Training

Our second proposal is to train multichannel U-Nets predicting the segmentation simultaneously for multiple symbol classes. This design choice has two advantages over



Figure 3. Training on convex hulls circumvents detection problems for symbols consisting of multiple connected components (see f-clef).

training separate detectors for each class. Firstly, at runtime we can predict the segmentation for multiple symbols with a single forward pass of the network. Furthermore, by simultaneously training on multiple symbols at the same time, we allow the model to share low-level feature maps for a certain symbol group (i.e., noteheads, beams and stems), and on the other hand force the model to learn upper-layer features that discriminate well between the various symbols, which – because the capacity of the model stays fixed, and the output layer only uses 1x1 convolutions – could lead to more descriptive representations of the image. In other words, due to the strong correlations across classes induced by music notation syntax, whatever features are learned for one output channel will at the same time be relevant for a different channel; the 1x1 convolution will simply weigh them differently.

However, this setup presents an optimization problem due to imbalanced classes: both in terms of how many foreground pixels there are (i.e. beams vs. duration dots), and with respect to how often they occur on an “average” page of sheet music (noteheads vs. clefs). We address the first issue by splitting the multichannel model into groups of symbols with roughly similar amount of foreground pixels across the dataset. To overcome the second issue, as the training setup operates on randomly chosen windows of the input image (see Sec. 4), we use oversampling: when drawing the random window when a training batch is being built, we check whether the window contains at least one pixel of the target class, and we retry up to five times if there is none. If no target class pixel is found in five tries, we concede and use the last sampled window, even though no pixel of target class is in it. (As opposed to this oversampling, adjusting the weights of the output channels did not lead to improvements.)

Furthermore, if model capacity becomes a limiting factor, we can opt out of sharing the up-sampling part of the model and keep a separate “decoder” for each output channel. This is a compromise that retains some of the speed, space and feature-sharing advantages, but at the same time does not so severely restrict the capacity of the model.

### 4. EXPERIMENTAL SETUP

We restrict ourselves to the subset of symbol classes that are necessary for pitch inference and basic duration inference (we currently do not detect tuplets – detecting handwritten digits is straightforward enough, the difficulty with tuplets lies in the notation assembly stage). Already this



selection contains symbols with heterogeneous appearance: constant-size, trivial shape (specifically, noteheads, ledger lines, whole and half rests, duration dots), constant-size, non-trivial shape (clefs, flags, accidentals, quarter-, 8th- and 16th rests), and symbols that have simple shapes, but varying extent (stems, beams, barlines).<sup>2</sup> We assume binary inputs, not least because large-scale OMR symbol detection ground truth is only available for binary images; however, binarization can be done with the same model.

**Dataset.** We use the MUSCIMA++ dataset, version 1.0 [20]. This is the only publicly available dataset of handwritten music notation with ground truth for symbol detection at a scale that is feasible for machine learning. The dataset contains over 90 000 manually annotated symbols with pixel masks. We use the designated writer-independent test set from MUSCIMA++.

**Training Details.** We set the network input size to a  $256 \times 512$  window and randomly sample crops of this size as training samples. We train all our models using the Adam update rule with an initial learning rate of 0.001 [27] and a batch size of 2 (with the  $256 \times 512$  input window, this is equivalent to batches of a single  $512 \times 512$  image of [38]). After there is no update on the validation loss for 25 epochs, we divide the learning rate by 5 and continue training from the previously best model. This procedure is repeated two times.

## 5. RESULTS

As there is no work to which we can compare directly, we first gather at least related OMR solutions, in order to provide whatever context we can for the reader. Then, we report results for symbol detection, and evaluate it in context of downstream tasks: pitch inference in a baseline full-pipeline OMR scenario, as well as first experiments applying our models in retrieval settings.

### 5.1 Comparison to Existing Systems

Comparison to existing systems is hard, because there are few symbol detection results reported, and even fewer full-pipeline OMR results. Direct comparison is not possible, as the MUSCIMA++ dataset we use has been released only very recently, and previous OMR pipelines (see Sec. 2) generally do not have publicly available code. Furthermore, earlier literature on OMR rarely provides evaluation scores, most of previous work on OMR has (sensibly) focused on printed music rather than manuscripts, and there are few established evaluation practices in OMR anyway [15, 21]. We do our best to at least gather literature where some results on related tasks are given, in order to provide context for our work.

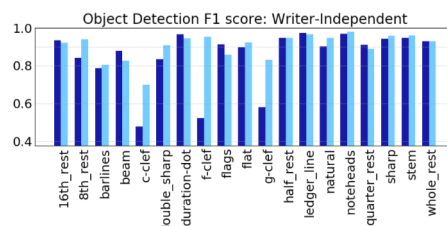
**Pitch accuracy, printed music.** In printed music, results for pitch accuracy have been consistently very good, when reported. Already in [32], the GAMUT system is said to correctly recover 96 % of pitches in printed music.

<sup>2</sup> There are also notation symbols that can have non-trivial shape and varying extent, such as slurs or hairpins; however, these are not required for neither pitch, nor duration inference, and we therefore leave them out.

The complex fuzzy system of [39] achieves near-perfect pitch accuracy (98.7 %). Similarly, the CANTOR system evaluated in [5] achieves 98 % semantic accuracy — this time, including polyphonic music. On printed square notation, [19] achieves 95 % pitch accuracy. A combination of systems in [42] achieves over 85 % joint pitch and duration accuracy.

**Symbol detection, handwritten music.** The most extensive evaluation of symbol detection in handwritten music has been carried out in [1]. Using a complex combination of robust heuristics for segmentation and machine learning for classification, they achieve an average symbol detection f-score of 0.75. These results seem ripe to be surpassed with CNNs: in [31], 98 % handwritten notehead detection accuracy has been reported. For staff detection, a similar architecture has been used in [4] with over 97 % pixel-wise f-score, and similar results are available with a ConvNet pixel classification approach for semantic segmentation into background, text, staves, and notation symbols [11]. At the same time, [33] reports symbol *classification* (without localization) accuracy over 98 %, indicating that CNNs are well capable of generalizing over the variety of handwritten musical symbols. However, we are *not* aware of pitch accuracy results reported on handwritten CWMN scores.

**OMR for Retrieval.** For retrieval, it is even harder to find comparable results, since evaluation metrics for retrieval depend on the test collection, and there is no such established collection for OMR. Using the open-source Audiveris<sup>3</sup> OMR software, [7] matches 9803 printed monophonic fragments from *A Dictionary of Musical Themes* to their electronic counterparts, using a comparable DTW alignment that also (mostly) ignores note duration, reporting a top-1 accuracy of 0.44; however, the collection of themes is a difficult one, since it often contains very similar melodies.



**Figure 4.** Results for binary segmentation models for individual symbols. Blue: baseline training with mask output; green: training with convex hulls.

### 5.2 Symbol Detection

We report detection f-scores for the chosen subset of symbols. Aggregating the results is not too meaningful: some rare symbols have an outsized impact on downstream processing (clefs). In Fig. 4, we show the baseline results and compare them to the convex hull setup. Training against

<sup>3</sup> <https://github.com/audiveris/audiveris>

Method	c-clef	g-clef	f-clef
single channel – no convex hull	0.48	0.58	0.52
single channel	0.70	0.83	<b>0.95</b>
multi-channel – all	0.16	0.37	0.49
multi-decoder – clefs, oversampling	<b>0.77</b>	<b>0.96</b>	<b>0.93</b>

**Table 1.** Comparison of detection performance (F-score) of clefs using different segmentation strategies.

convex hulls of objects does address the issue of detecting otherwise disjoint symbols using connected components; otherwise it achieves mixed results.

Compressing the detector with multichannel training without a loss of performance was possible on correlated sub-groups of symbols that bypass the class imbalance problem, such as training together noteheads, stems, beams, and flags; the results worsened when all classes were trained at once. The clefs were most affected by all the changes to the model described in Section 3: improved by convex hull training, neglected when the multichannel model was trained to predict all symbol classes at once, and then drastically improved again when trained as a group with separate decoders and the oversampling strategy. Table 1 summarizes the results for clef detection. Clefs are critical for useful OMR, since they affect the pitch inferred from all subsequent noteheads.

## 6. APPLICATION SCENARIO: FULL-PIPELINE HANDWRITTEN OMR IN RETRIEVAL

We now explore the utility of the symbol detectors within an OMR pipeline. It is known in OMR that low-level errors can lead to effects on recognition of wildly different magnitudes [15, 35]; in the presence of detection errors, one should therefore see how severely they impact downstream applications. We choose a *retrieval* scenario as the application context for evaluating symbol detection. As opposed to applications where we produce the transcribed score [15, 21, 41], this is straightforward to evaluate.

To verify that our symbol detection approach can yield useful results in an application context, we add a simple notation assembly and pitch inference system on top of the symbol detection results. We choose *retrieval* as the most feasible application of handwritten OMR: there are music manuscript archives with thousands of scores that contain manual copies, and matching them cannot be done without their musical content.

For inferring pitch, we must re-introduce stafflines. However, we can safely assume they have been detected correctly: both [4] and our replication of their experiments with stafflines on this dataset exhibit extremely few errors, and these can be filtered away with a trivial projection heuristic such as that of [18].

### 6.1 Notation Assembly and Music Inference

Symbol detection alone is not sufficient for decoding musical information: meaningful units are *configurations* of

symbols rather than the symbols themselves [6, 20]. The notation assembly stage is the step where these configurations are recovered (step (4) in the OMR pipeline: see 1). In the MUSCIMA++ dataset, they are represented as an oriented graph; once this graph is recovered, one can perform deterministic pitch inference.<sup>4</sup>

Symbol detection outputs vertices of the notation graph; we therefore need to recover graph edges. Replicating the baseline established in [20], we train a binary classifier over ordered symbol pairs. While this classifier achieves an f-score of 0.92, it makes embarrassing errors: noteheads connected to irrelevant ledger lines in chords, to beams that belong to an entirely different staff, and sometimes to multiple adjacent stems. We discard these obviously wrong edges using straightforward heuristics. We also discard detected objects that are entirely contained within another detected object. The last step is recovering *precedence* edges: we just order rest and noteheads on each staff left-to-right; noteheads connected to the same stem are considered simultaneous, but actual polyphony is ignored.

Once the pitches, durations, and onsets are inferred for the detected noteheads, we then export them as a MIDI file. MIDI is appropriate for retrieval, since it presents straightforward ways of computing similarity. This file then can serve as both the query and the database key for the given score. To compute the similarity of two MIDI files, we align them using Dynamic Time Warping [29] (DTW) over sequences of time frames that contain onsets. The DTW score function for a pair of frames is 1 minus the Dice coefficient of the onset pitch sets in the frames. Then, we match individual pitches within the frame sets that are aligned by DTW and measure the f-score of predicted pitches. DTW is used as the similarity function in [7]; however, we do not reduce polyphonic music to its upper pitch envelope.

### 6.2 Results

We now report how the full-pipeline baseline on top of the object detection U-Nets predicts pitches, and how it can be used to retrieve related scores.

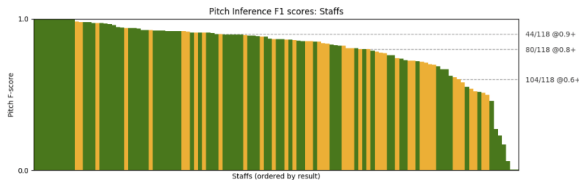
**Pitch accuracy.** We use the DTW alignment to directly evaluate pitch classification.<sup>5</sup> Performing DTW on the inference outputs for page images, we achieve a (micro-)average F-score of only 0.59. Rather than due to errors in symbol detection, this is mostly due to the polyphony de-synchronization effects of bad duration inference; indeed, on (mostly) monophonic music, pitch F-score jumps to 0.78. In order to bypass de-synchronization problems that in fact obscure correct pitch recognition, we split the scores into individual staves (118 in total) and evaluate pitch accuracy on these. The results for the test set staves are reported in Fig. 5. On average, we obtain pitch F-score 0.81, with 0.83 for monophonic staves (and ignoring clef errors, 0.88).

Finally, we evaluate our detector in the context of a retrieval application. We run experiments both on gold-

<sup>4</sup> A proof-of-concept implementation: <https://github.com/hajicj/muscima>.

<sup>5</sup> We could evaluate duration classification as well, but due to errors by the notation assembly baseline, this is too low to be worth reporting.





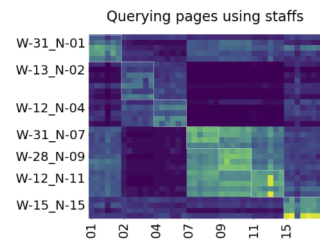
**Figure 5.** Pitch F-score after DTW alignments on the 118 individual staves in the writer-independent test set, ordered by result. Monophonic staves (darker green) predictably score better than staves with multiple voices or chords (yellow). We found no clear relationship between pitch accuracy and handwriting style.

standard MIDI retrieval and duplicate score retrieval, using the predicted scores; since the similarity metric is pitch f-score, all retrieval experiments work in both directions. Experiments with ground truth MIDI correspond to cross-modal retrieval, where the modalities are a symbolic representation, and the score projected into the MIDI modality using the OMR system; queries with predictions correspond to a simpler scenario where we are querying scores with scores, using the OMR system as a hash function.

**Retrieving gold MIDI with scores.** Given how small the test set is, retrieving the correct ground truth page — and even staff — should be near-perfect. For staff-to-staff retrieval,  $\text{Prec}@1$  is 0.93; for page-to-page and staff-to-page retrieval, this is 1.0, indicating that with our U-Net object detection stage, retrieving gold-standard MIDI using handwritten scores (and vice versa, as the similarity metric is symmetrical) is feasible.

**Retrieving scores with scores.** The next scenario is to run retrieval not against the ground truth, but against MIDI predicted from different versions of the test set scores. While errors related to differences in handwriting get compounded, the rest of the pipeline imposes consistent limitations on both the database and query recognition outputs and may make the *same* errors on both query and database scores, making the task actually easier. Therefore, we select a *confuse-retrieval* subset of 7 scores from MUSCIMA++ that are as similar to each other as possible: mostly monophonic, and with 0 – 2 sharps. Some of these pieces are musically closely related. For these experiments, our database consists of recognition outputs computed from all *confuse-retrieval* pages in the *training* subset of MUSCIMA++. Queries are taken from predictions on the writer-independent test set: we use both the 7 entire pages and individual staves (34 of those).

The system achieves perfect  $\text{Prec}@1$  when pages are used as queries, and 0.94 when using staff queries (2 staff queries did not return the right piece as the top result). The retrieval scores are plotted in Fig. 6. We checked this score also with ground truth queries; this system made only 2 errors as well, but in different queries, which we take as circumstantial evidence that the ground truth MIDI has different issues when matching against a predicted MIDI than a different prediction. When measuring MAP with the cutoff  $k=6$  (as there are 7 versions of each page in MUSCIMA++



**Figure 6.** Pitch f-score between predictions on test set staves and (predictions on) training set pages. Notice the pages 07, 09 and 11: these are three movements from J. S. Bach’s Cello suite no. 1, which contain musically highly related material.

and one of them is used for querying), it drops to 0.86.

## 7. DISCUSSION & CONCLUSIONS

We consider our work a successful step towards enabling applications of hitherto problematic handwritten OMR. The retrieval scenario results are an indication that U-Nets are a workable solution to the handwritten symbol detection bottleneck in the context of full-pipeline OMR. (Here, we must re-state that these results should *not* be interpreted as more than supporting evidence that our object detection method is viable for such scenarios!)

However, U-Nets are still in principle limited by the size of the receptive field: for instance the middle of a long stem looks exactly the same as a barline. We could further leverage syntactic properties of music notation: e.g., the self-attention layer of [34] allows building up the final output from partial recognition results. Fragmenting of long symbols could be overcome with instance segmentation embeddings [10].

To the best of our knowledge, this is also the first time OMR was done with a machine-learning method for notation assembly. We in fact consider this the most interesting line of follow-up work. Recovering the notation graph itself seems like the next bottleneck, especially for duration inference. The non-independent nature of the edges poses an interesting structured prediction challenge, and one could also work towards models that jointly detect symbols and recover their relationships.

Despite their limitations, U-Nets can be used to detect handwritten music notation symbols. They establish a new CNN-based baseline for the object detection task, and we believe the results in pitch inference and a proof-of-concept retrieval scenario indicate that a significant step has been taken towards full-pipeline OMR systems, so that the content of musical manuscripts can become accessible digitally.

## 8. ACKNOWLEDGMENTS

Jan Hajič jr. and Pavel Pecina acknowledge support by the Czech Science Foundation grant no. P103/12/G084, Charles University Grant Agency grants 1444217 and 170217, and by SVV project 260 453.

## 9. REFERENCES

- [1] Ana Rebelo. *Robust Optical Recognition of Handwritten Musical Scores based on Domain Knowledge*. PhD thesis, 2012.
- [2] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R. S. Marcal, Carlos Guedes, and Jaime S. Cardoso. Optical Music Recognition: State-of-the-Art and Open Issues. *Int J Multimed Info Retr*, 1(3):173–190, Mar 2012.
- [3] Andrew Hankinson, John Ashley Burgoyne, Gabriel Vigliensoni, Alastair Porter, Jessica Thompson, Wendy Liu, Remi Chiu, and Ichiro Fujinaga. Digital Document Image Retrieval Using Optical Music Recognition. In *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, Mosteiro S.Bento Da Vitória, Porto, Portugal, October 8-12, 2012*, pages 577–582. FEUP Edições, 2012.
- [4] Antonio-Javier Gallego and Jorge Calvo Zaragoza. Staff-line removal with selectional auto-encoders. *Expert Systems with Applications*, 89:138–148, 2017.
- [5] David Bainbridge. Extensible optical music recognition. page 112, 1997.
- [6] David Bainbridge and Tim Bell. A music notation construction engine for optical music recognition. *Software - Practice and Experience*, 33(2):173–200, 2003.
- [7] Stefan Balke, Sanu Pulimootil Achankunju, and Meinard Müller. Matching Musical Themes based on noisy OCR and OMR input. pages 703–707, 2015.
- [8] P. Bellini, I. Bruno, and P. Nesi. Optical music sheet segmentation. In *Proceedings First International Conference on WEB Delivering of Music. WEDELMUSIC 2001*, pages 183–190. Institute of Electrical & Electronics Engineers (IEEE), 2001.
- [9] Avi Ben Cohen, Idit Diamant, Eyal Klang, Michal Amitai, and Hayit Greenspan. Fully Convolutional Network for Liver Segmentation and Lesions Detection. In Gustavo Carneiro, Diana Mateus, Loïc Peter, Andrew Bradley, João Manuel R. S. Tavares, Vasileios Belagiannis, João Paulo Papa, Jacinto C. Nascimento, Marco Loog, Zhi Lu, Jaime S. Cardoso, and Julien Cornebise, editors, *Deep Learning and Data Labeling for Medical Applications*, pages 77–85, Cham, 2016. Springer International Publishing.
- [10] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic Instance Segmentation with a Discriminative Loss Function. *CoRR*, abs/1708.02551, 2017.
- [11] Jorge Calvo Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. A machine learning framework for the categorization of elements in images of musical documents. In *Third International Conference on Technologies for Music Notation and Representation*, A Coruña, 2017. University of A Coruña.
- [12] Sukalpa Chanda, Debleena Das, Umapada Pal, and Fumitaka Kimura. Offline Hand-Written Musical Symbol Recognition. *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 405–410, sep 2014.
- [13] Cuihong Wen, Jing Zhang, Ana Rebelo, and Fanyong Cheng. A Directed Acyclic Graph-Large Margin Distribution Machine Model for Music Symbol Classification. *PLOS ONE*, 11(3):e0149688, mar 2016.
- [14] V.P. d’Andecy, J. Camillerapp, and I. Leplumey. Kalman filtering for segment detection: application to music scores analysis. In *Proceedings of 12th International Conference on Pattern Recognition*. IEEE Comput. Soc. Press, 1994.
- [15] Donald Byrd and Jakob Grue Simonsen. Towards a Standard Testbed for Optical Music Recognition: Definitions, Metrics, and Page Images. *Journal of New Music Research*, 44(3):169–195, 2015.
- [16] Eelco van der Wel and Karen Ullrich. Optical Music Recognition with Convolutional Sequence-to-Sequence Models. *CoRR*, abs/1707.04877, 2017.
- [17] Christian Fremerey, Meinard Müller, Frank Kurth, and Michael Clausen. Automatic mapping of scanned sheet music to audio recordings. *Proceedings of the International Conference on Music Information Retrieval*, pages 413–418, 2008.
- [18] Ichiro Fujinaga. Optical Music Recognition using Projections. Master’s thesis, 1988.
- [19] Gabriel Vigliensoni, John Ashley Burgoyne, Andrew Hankinson, and Ichiro Fujinaga. Automatic Pitch Detection in Printed Square Notation. In *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011*, pages 423–428. University of Miami, 2011.
- [20] Jan Hajič jr. and Pavel Pecina. The MUSCIMA++ Dataset for Handwritten Optical Music Recognition. In *14th International Conference on Document Analysis and Recognition*, pages 39–46, New York, USA, November 2017. Dept. of Computer Science and Intelligent Systems, Graduate School of Engineering, Osaka Prefecture University, IEEE Computer Society.
- [21] Jan Hajič jr., Jiří Novotný, Pavel Pecina, and Jaroslav Pokorný. Further Steps towards a Standard Testbed for Optical Music Recognition. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*, pages 157–163, New York, USA, 2016. New York University, New York University.
- [22] Jan Hajič Jr. and Pavel Pecina. Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression. *CoRR*, abs/1708.01806, 2017.



- [23] Jesus Munoz Bulnes, Carlos Fernandez, Ignacio Parra, David Fernandez Llorca, and Miguel A. Sotelo. Deep fully convolutional networks with random data augmentation for enhanced generalization in road detection. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, oct 2017.
- [24] Jorge Calvo-Zaragoza, Jose J. Valero-Mas, and Antonio Pertusa. End-to-End Optical Music Recognition Using Neural Networks. In *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, pages 472–477, 2017.
- [25] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *CoRR*, abs/1506.02640, 2015.
- [26] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.
- [27] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR) (arXiv:1412.6980)*, 2015.
- [28] F. Kurth, M. Müller, C. Fremerey, Y. Chang, and M. Clausen. Automated synchronization of scanned sheet music with audio recordings. *Proc. ISMIR, Vienna, AT*, pages 261–266, 2007.
- [29] Lawrence R. Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice Hall signal processing series. Prentice Hall, 1993.
- [30] Matthias Dorfer, Andreas Arzt, and Gerhard Widmer. Learning Audio-Sheet Music Correspondences for Score Identification and Offline Alignment. In *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, pages 115–122, 2017.
- [31] Matthias Dorfer, jr. Jan Hajič, and Gerhard Widmer. On the Potential of Fully Convolutional Neural Networks for Musical Symbol Detection. In *Proceedings of the 12th IAPR International Workshop on Graphics Recognition*, pages 53–54, New York, USA, 2017. IAPR TC10 (Technical Committee on Graphics Recognition), IEEE Computer Society.
- [32] Michael Droettboom and Ichiro Fujinaga. Symbol-level groundtruthing environment for OMR. *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 497–500, 2004.
- [33] Alexander Pacha and Horst Eidenberger. Towards a Universal Music Symbol Classifier. In *Proceedings of the 12th International Workshop on Graphics Recognition*, 2017.
- [34] N. Parmar, A. Vaswani, J. Uszkoreit, Ł. Kaiser, N. Shazeer, and A. Ku. Image Transformer. *ArXiv e-prints*, February 2018.
- [35] Pierfrancesco Bellini, Ivan Bruno, and Paolo Nesi. Assessing Optical Music Recognition Tools. *Computer Music Journal*, 31(1):68–93, Mar 2007.
- [36] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. Technical report, 2018.
- [37] K. T. Reed and J. R. Parker. Automatic computer recognition of printed music. *Proceedings - International Conference on Pattern Recognition*, 3:803–807, 1996.
- [38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III*, pages 234–241, Cham, 2015. Springer International Publishing.
- [39] Florence Rossant and Isabelle Bloch. Robust and Adaptive OMR System Including Fuzzy Modeling, Fusion of Musical Rules, and Possible Error Detection. *EURASIP Journal on Advances in Signal Processing*, 2007(1):081541, 2007.
- [40] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015.
- [41] Mariusz Szwoch. Using MusicXML to Evaluate Accuracy of OMR Systems. *Proceedings of the 5th International Conference on Diagrammatic Representation and Inference*, pages 419–422, 2008.
- [42] Victor Padilla, Alan Marsden, Alex McLean, and Kia Ng. Improving OMR for Digital Music Libraries with Multiple Recognisers and Multiple Sources. *Proceedings of the 1st International Workshop on Digital Libraries for Musicology - DLfM '14*, pages 1–8, 2014.



## 7.4 A Baseline for General Music Object Detection with Deep Learning

Alexander Pacha, Jan Hajič jr. and Jorge Calvo-Zaragoza. A Baseline for General Music Object Detection with Deep Learning. *Applied Sciences*, Vol. 8, No. 9, pages 1488–1488, Basel, Switzerland, 2018. ISSN 2076-3417.

The article *A Baseline for General Music Object Detection with Deep Learning* compares three general object detection models with each other across three different OMR datasets that have object detection ground truth. The models used are Faster R-CNN (two-step detection), RetinaNet (one-step detection), and the U-Net models from the previous publication [Hajič jr., 2018]. The datasets in question are the Capitan dataset of mensural notation, the DeepScores dataset with synthetic printed modern notation, and the MUSCIMA++ dataset with handwritten modern notation. The paper therefore establishes a relatively robust and state-of-the-art baseline for music notation object detection, using the general MSCOCO evaluation methodology with (Weighted) Mean Average Precision. Notably, the U-Net model outperformed the other baselines, even without applying the training tricks from the previous paper.

In this paper, the dissertation author contributed the U-Net experiments on all three datasets and most of the text in the Results and Conclusions sections, with contributions to the text throughout. The contribution of the dissertation author to this article is about 25–30%; the purpose of including the article in the dissertation is to provide supporting evidence that the U-Net approach developed over the previous two articles has been selected well, in that it is competitive against other baseline options.

Article

# A Baseline for General Music Object Detection with Deep Learning

Alexander Pacha <sup>1,\*</sup> , Jan Hajič, Jr. <sup>2</sup> and Jorge Calvo-Zaragoza <sup>3</sup>

<sup>1</sup> Institute for Visual Computing and Human-Centered Technology, TU Wien, 1040 Wien, Austria

<sup>2</sup> Institute of Formal and Applied Linguistics, Charles University, 116 36 Staré Město, Czech Republic; hajicj@ufal.mff.cuni.cz

<sup>3</sup> PRHLT Research Center, Universitat Politècnica de València, 46022 València, Spain; jcalvo@upv.es

\* Correspondence: alexander.pacha@tuwien.ac.at

Received: 31 July 2018; Accepted: 26 August 2018; Published: 29 August 2018



**Abstract:** Deep learning is bringing breakthroughs to many computer vision subfields including Optical Music Recognition (OMR), which has seen a series of improvements to musical symbol detection achieved by using generic deep learning models. However, so far, each such proposal has been based on a specific dataset and different evaluation criteria, which made it difficult to quantify the new deep learning-based state-of-the-art and assess the relative merits of these detection models on music scores. In this paper, a baseline for general detection of musical symbols with deep learning is presented. We consider three datasets of heterogeneous typology but with the same annotation format, three neural models of different nature, and establish their performance in terms of a common evaluation standard. The experimental results confirm that the direct music object detection with deep learning is indeed promising, but at the same time illustrates some of the domain-specific shortcomings of the general detectors. A qualitative comparison then suggests avenues for OMR improvement, based both on properties of the detection model and how the datasets are defined. To the best of our knowledge, this is the first time that competing music object detection systems from the machine learning paradigm are directly compared to each other. We hope that this work will serve as a reference to measure the progress of future developments of OMR in music object detection.

**Keywords:** optical music recognition; deep learning; object detection; music scores

---

## 1. Introduction

Optical Music Recognition (OMR) is the field of research that investigates how to computationally read music notation in documents. Having accurate OMR technology would enable fully integrating written music into the ecosystem of digital music processing. In recent years, diverse initiatives have been launched to digitize musical heritage in the written form, such as the The Digital Image Archive of Medieval Music project [1] on the academic side, or at the same time the crowd-sourced International Music Score Library Project (IMSLP) repository of public-domain and openly available music [2] which has grown to become a primary provider of sheet music worldwide. However, making not only the digital images of all these compositions, but also their structured representation accessible at scale, as attempted e.g., by the Single Interface for Music Score Searching and Analysis (SIMSSA) project [3], would constitute a breakthrough in interacting with written music, and making it accessible to both the professional and the general public in previously unseen ways: content-based search in large sheet music libraries including cross-modal retrieval, digital musicology at scale and with access to structured representations of music that only exists in written form, renotation of early notation to modern notation, manuscript transcription and part-matching to directly cut costs of music directors

and composers. These (and more) applications have been envisioned in OMR literature for a long time [4,5]; however, results have not been forthcoming [6].

In order to be able to apply Music Information Retrieval (MIR) algorithms on music scores and enable this wide range of applications, it is first necessary to bring them into this symbolic, machine-readable format. Manually creating such symbolic representations by means of specialized music typesetting software is an expensive effort, and constitutes the bottleneck to digitally encoding music at large scales—which is, in turn, a bottleneck both for digital musicology, subsequent MIR applications, and music accessibility.

OMR is expected to provide the enabling technology for scalable structured encoding. From this perspective, OMR can be seen as the key to diversifying the available symbolic music sources in reasonable time and cost. Crucially, OMR has seen a shift in paradigms in the last few years, mainly triggered by advances in the field of computer vision and machine learning through deep learning [7–10]. This development is further fueled by the availability of large annotated datasets (e.g., MUSCIMA++, DeepScores) and sufficient computational power to work with such large datasets. This new paradigm, combined with a better understanding of the challenges [11,12], allow approaching the problem of OMR somewhat differently.

The entire process of OMR can be broken down into the following steps [6,13–15]:

1. **Preprocessing:** Standard techniques to ease further steps, e.g., contrast enhancement, binarization, skew-correction or noise removal. Additionally, the layout should be analyzed to allow subsequent steps to focus on actual content and ignore the background.
2. **Music Object Detection:** This step is responsible for finding and classifying all relevant symbols or glyphs in the image. Note that music object detection is sometimes referred to as music symbol recognition, but we use the former term because of its relation to “object detection”, which is commonly used in computer vision to refer to the very same localization and classification task in (natural) images, answering the question “What is where in this image?”.
3. **Relational Understanding:** From the detected and classified symbols, a music notational graph (MuNG) can be constructed that holds both the symbols and their relationships to each other. Note that, for a complete and unambiguous reconstruction, two kinds of relations are necessary: a logical relationship (e.g., between a notehead and a stem) and a temporal relationship to guarantee the correct order of the symbols. The graph formulation essentially re-casts the notation reconstruction algorithms like that of [16] as a problem of recovering binary labels over symbol pairs, therefore also making it amenable to machine learning approaches. Again, other works sometimes refer to the stage after object detection as semantical reconstruction. Note that, in this approach, this stage only attempts to reconstruct the relations between symbols and a large part of the semantics is assigned in the encoding stage.
4. **Encoding:** Given a complete music notation graph, the music can be encoded into any output format unambiguously, e.g., into MIDI for playback or MusicXML/MEI for further editing in a music notation program. Keep in mind that this step potentially has to deal with the subtleties of music notation, such as omitted symbols.

Currently, the hardest challenge of this pipeline is posed by the music object detection step. Unfortunately, it is unclear to what extent deep learning has been successful in addressing this stage. Existing studies that focus on music notation objects are dispersed and not comparable with each other in terms of the used algorithms, datasets, and metrics, which has so far made a fair comparison impossible. However, there is no good reason for this state of affairs: music object detection can borrow standard evaluation from generic object detection settings, and the deep learning models are similarly domain-agnostic. Therefore, this work aims to fill an obvious gap: provide a direct comparison between the different general deep learning models for object detection that were recently proposed for the task of music object detection, across the available musical symbol datasets, and thus establish a clear state-of-the-art baseline.



We evaluate three competing approaches on three distinct datasets containing both handwritten and typeset music. To compare the different approaches on common ground, we propose a standard bounding-box based data model, usable with multiple OMR datasets, and use an up-to-date standard for evaluating object detection, namely the *Common Objects in Context* (COCO) evaluation protocol [17]. All scripts for obtaining the test-bed, preprocessing the data and evaluating the results are being made publicly available [18].

To the best of our knowledge, this marks the first time that music object detection methods based on machine learning are directly compared against each other. Bellini et al. [19] evaluated a number of commercial OMR applications in 2007, but it was done manually, making it difficult to replicate, and, more importantly, the systems have no published descriptions, which means the comparison has limited value for guiding future developments. The evaluation methodology in [19] also does not correspond to current object detection evaluation protocols.

## 2. Background on Music Object Detection

Traditionally, OMR has been approached by workflows composed of several stages, as outlined in the previous section. In addition, these stages were further subdivided into smaller steps. Inside of the music object detection stage, the key step used to be the staff-line detection and removal [20]. Although staves are essential for the understanding of music notation, their presence hindered the isolation of musical primitives using classical algorithms such as connected-components analysis. That is why, for many years, much research was devoted to improving staff-line removal [21]. Currently, thanks to the use of deep neural networks, the staff-line removal can be considered a solved problem, with selectional auto-encoders outperforming all previously existing methods given a sufficient amount of training data [22]. However, even with an ideal staff-line removal algorithm, isolating musical symbols by means of connected components remains problematic, since multiple primitives could be connected to each other (e.g., a beam group can be a single connected component that includes several heads, stems, and beams) or a single unit can have multiple disconnected parts (e.g., a fermata, voltas, f-clef). The second case is particularly severe in the context of handwritten notation, where symbols can be written with such a high variability (e.g., detached noteheads) that modeling all possible appearances becomes intractable.

Recently, it has been shown that the use of region-based machine learning models is an alternative that can deal with the stage of music object detection holistically. These models have been widely developed in the computer vision community, attaining high performance in detecting objects in images by using convolutional neural networks. In addition to the performance, a compelling advantage is that these models can be trained in an end-to-end manner, that is, by merely providing pairs of images and positions where the objects to be detected are located; these models, therefore, make it possible to bypass several stages of the classical OMR workflow by directly detecting symbols in music score images.

Pacha et al. [23] presented the first work that considered region-based convolutional neural networks for the task of music object detection. They proposed a sliding-window based approach, that cuts the image in a context-sensitive way into smaller chunks that contain no more than one staff and ran a Faster R-CNN detector to obtain the positions and classes of all symbols in the cropped image. While the evaluation is limited to the detection performance on small image chunks instead of the entire images, the extension of this approach to full pages of handwritten music scores, written in mensural notation, is reported to yield promising results [24].

Hajič jr. et al. [25] use a different approach: instead of applying an object detection model directly, they use a semantic segmentation model and a subsequent detection stage. More specifically, the semantic segmentation is done with the U-Net architecture [26]. The overall detection problem is broken down into a set of binary pixel classification problems and subsequently uses a connected components detector to arrive at the final detection proposals. The object detection results are reported in terms of F-scores, broken down by symbol class with no aggregate result, and the experiments are

done only for a subset of the symbol classes available in the MUSCIMA++ dataset; on the other hand, the notation reconstruction step is subsequently applied, and the object detection is evaluated in terms of the subsequent MIDI inference.

The Deep Watershed Detector proposed by Tuggener et al. [27] is another attempt to solve music object detection by training a convolutional neural network to learn a custom energy function that is used in a watershed transformation to perform semantic segmentation of an entire score. They evaluate their approach on the DeepScores and the MUSCIMA++ dataset. While the results for some classes are promising, e.g., it works exceptionally well on small objects such as staccato dots, the algorithm generally struggles with rare classes, overlapping symbols, and accurate bounding box regression. Unfortunately, no overall results of the detection performance are given by the authors.

As discussed above, while these studies use standard object detection models, they used completely different datasets, vocabularies, and metrics for the reported results. A major part of the motivation for this paper is to evaluate these advances in music object detection in a consistent manner, so that future advances have a clear, up-to-date formulation and baseline.

### 3. Task Formulation

We formulate the task of object detection in images in the following way. Given an image, a variable-length list of 6-tuples  $(y_1, x_1, y_2, x_2, c, s)$  is obtained, where  $y_1, x_1$  and  $y_2, x_2$  denote the coordinates of the top-left and bottom-right corners, respectively, of a predicted bounding box,  $c$  is the category assigned to the object therein, and  $s$  is the confidence score given by the model to such a prediction. In the specific case of music object detection, the categories correspond to the music-notation primitives that are considered relevant to the user, depending on the specific OMR task. Note that the requirements may vary depending on both the input music notation and the pursued application: the interesting primitives for replayability may differ from the interesting ones for getting a structured encoding of the music.

The main reason to formulate the music object detection as bounding box retrieval is that it provides a direct relationship between the detection results and the entities to be recognized in the music score image. It has already been discussed in Section 2 that the traditional segmentation step based on connected components can produce both super-symbols (a single component that gathers several symbols) and sub-symbols (a single symbol separated into several parts), which increases the complexity of post-processing considerably. Similarly, a pixel-wise categorization (known as semantic segmentation in the computer vision community) might avoid predicting super-symbols, yet the problem with sub-symbols remains. In addition, a pixel-level annotation provides ambiguities that are difficult to handle when nearby or touching pixels are labeled in the same way while belonging to different entities (for example, multiple noteheads in a chord).

Furthermore, the prediction with bounding boxes provides an implicit grouping. Thus, detecting isolated entities directly, along with their positions in the image, is the kind of information that the following stages of the OMR workflow might need, in which detected symbols are grouped to reconstruct the actual music notation. Therefore, once objects have been detected, the image is no longer relevant, since the bounding boxes are sufficient representatives of the graphical information that needs to be recovered from the music score image. For example, bounding box dimensions have long been used as features for symbol classification in pipelines where this step is separate [4]; they are suitable for filtering false positives [28]; in the dependency graph approach of MUSCIMA++, bounding boxes already provide useful features for the reconstruction step [14]; and they could be also used to model terminals of a music notation grammar for the reconstruction stage [29].

In addition to the above, the reality with music documents is that the stylistic and graphical differences amongst different manuscripts is very pronounced, especially in the case of handwritten notation. That means it is advisable to build ground-truth data for each type of manuscript with which to train the recognition models, as is happening in other similar domains such as text recognition [30]. We believe that annotating images at the bounding box level is less expensive than building a dataset



to train a traditional multi-stage system, in which each stage needs its own ground truth. Furthermore, this level of annotation represents a good trade-off between effort and accuracy in comparison to other current approaches in computer vision that include pixel-wise labeling [31]. Although these fine-grained annotations could eventually lead to better localization results, the required initial effort for building ground-truth data is much higher, which is especially detrimental when dealing with a new type of music manuscript.

## 4. Experimental Setup

### 4.1. Object Detection Models

The objective of this work is to provide a good baseline for the music object detection task, and so we consider three neural models of different nature for performing the experiments. While we do want our detectors to be as accurate as possible, we primarily wish to exemplify the different deep learning approaches to object detection. We believe that this is more interesting from the point of view of some reference results, and can help to draw more interesting conclusions. Thus, we use Faster R-CNN as a representative of two-stage detectors, RetinaNet as a representative of one-stage detectors, and U-Nets as a representative of models based on pixel-level segmentation. Figure 1 overviews the general operation of these types of detectors.

#### 4.1.1. Faster R-CNN

Faster Region-based Convolutional Neural Network (Faster R-CNN) [32] is the evolution of the first convolutional network schemes for object detection R-CNN [33] and Fast R-CNN [34]. Faster R-CNN belongs to the class of two-stage detectors, with the first stage generating a sparse set of region proposals that are classified and further refined in the second stage.

While the previous R-CNN schemes used an external mechanism for generating the proposals, such as Selective Search [35] or EdgeBoxes [36], Faster R-CNN attempts to learn the object proposal stage directly from the data employing a region proposal network. The whole process can be carried out efficiently because the convolutional features are shared between both stages, and therefore computing the region proposals does not represent a bottleneck. This also increases the efficiency to train such a network.

The details for training this model followed the recommendations given in the work of Pacha et al. [23]. That is, an Inception-ResNet-V2 [37] is used for the feature extraction stage, initialized with pre-trained weights from ImageNet (as provided by TensorFlow Object Detection API [38]). Input images are rescaled so that the longest edge is no longer than 1000 pixels. A clustering of symbol bounding box shapes is done for each dataset, in order to establish an appropriate set of bounding box shapes to predict, therefore providing appropriate hyperparameters for the object proposal stage.

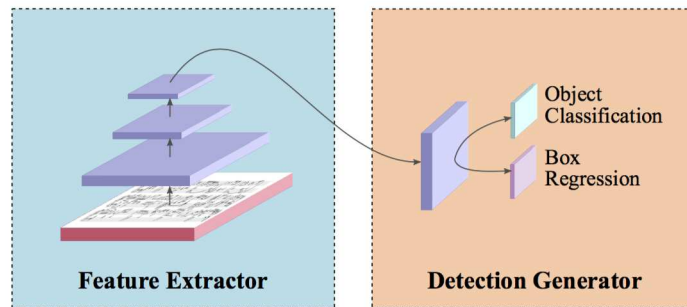
#### 4.1.2. RetinaNet

The RetinaNet [39] belongs to the family of one-stage detectors that are built on convolutional neural networks. Other prominent representatives are OverFeat [40], Single Shot Detector (SSD) [41] or You Look Only Once (YOLO) [42]. These one-shot detectors create a dense set of proposals along a grid and directly classify and refine those proposals. As opposed to the two-stage detectors, they have to handle a large number of background samples, which potentially can dominate the learning signal.

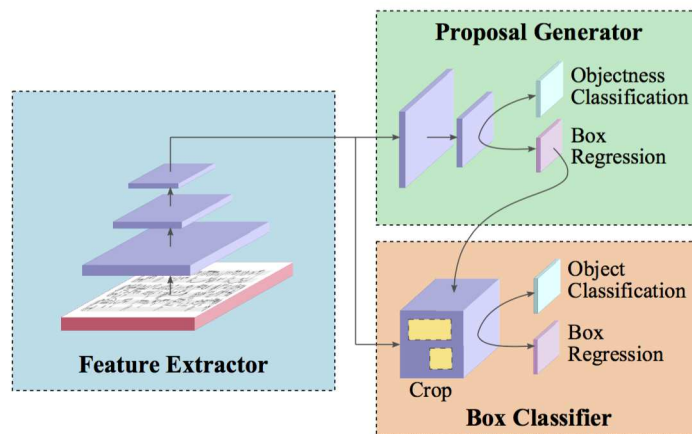
The RetinaNet [39] is an adaptation of a Residual Network [43] with lateral connections to create features on multiple scales [44]. Small convolutional subnetworks perform classification and bounding box regression on each output layer. RetinaNet was proposed along with the focal loss function, which tries to overcome the hard object-background imbalance issue by dynamically shifting weight to increase the contribution of hard negative examples and decreasing the contribution of easy positives.

The configuration of the network model requires setting several hyperparameters. We specifically checked four different back-ends for feature extraction, namely: ResNet50 [43], MobileNet128 [45],

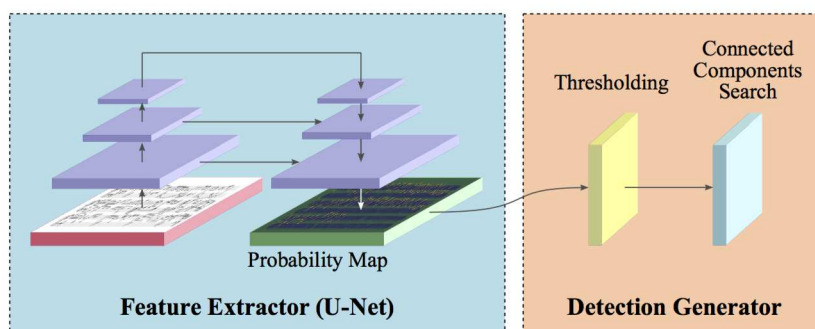
DenseNet121 [46], and a highly simplified version of the DenseNet. Various anchor dimension settings were also examined: the ResNet50 feature extractor performed best in preliminary experiments and was subsequently chosen. The negative overlap threshold was set to 40%, so every box with lower Intersection over Union (IoU) counts as background; similarly, the positive overlap threshold was set to 50%, and every box with a higher IoU is treated as foreground; boxes in between are omitted from the training signal.



(a) Basic architecture of a one-stage detector.



(b) Basic architecture of a two-stage detector.

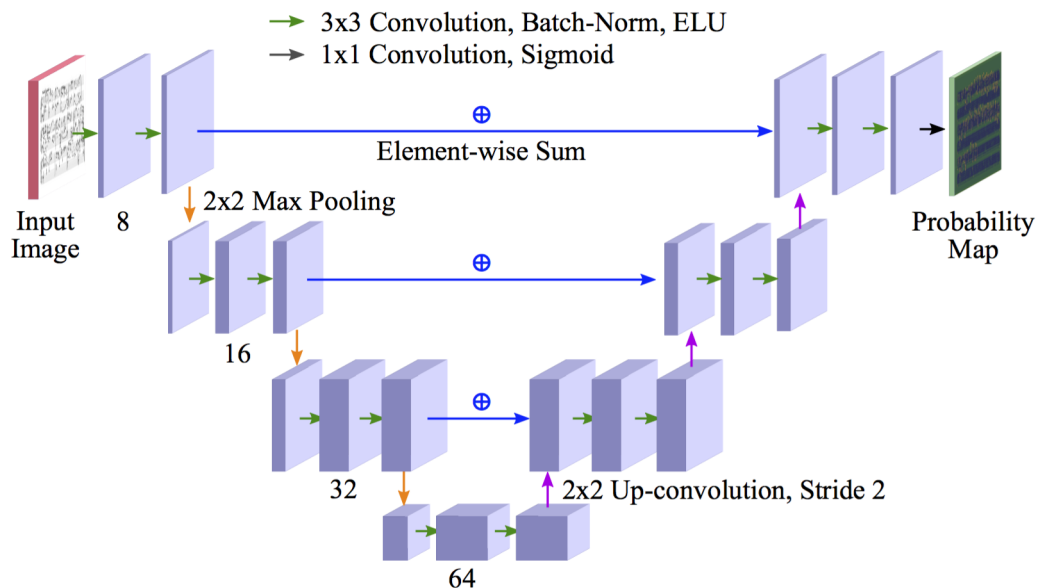


(c) Basic architecture of the U-Net detector.

Figure 1. Basic architectures of the considered types of object detectors.

#### 4.1.3. U-Net

The U-Net [26] is a model for performing semantic segmentation that assigns each pixel of the input image to a certain class. It can be extended to perform object detection, as defined in Section 3. The U-Net architecture combines three key elements: standard 2D convolutions, the “hourglass” architecture inspired by auto-encoders, and residual connections from ResNets [43]. As no other operations than convolutions and element-wise sums of corresponding layers in the “hourglass” are used, the U-Net can in parallel assign a label—or a numerical value, or a probability distribution—to each pixel of an arbitrarily large image. The architecture is depicted in Figure 2.



**Figure 2.** The U-Net architecture, with computation flowing left-to-right; the “hourglass” is unrolled downwards. Green arrows indicate 2D convolution with  $3 \times 3$  kernels, downward orange arrows indicate  $2 \times 2$  Max-Pooling, upward purple arrows indicate  $2 \times 2$  up-convolution, and blue arrows indicate element-wise sums that form the residual connections between corresponding parts of the two “hourglass” halves.

In order to generate the binary pixel mask training data from the bounding box ground truth, we set all pixels within the bounding boxes of a given symbol class to 1, resulting in rectangular foreground regions for each symbol instance (despite the fact that the symbols themselves are *not* rectangles).

One drawback of U-Nets is that they were initially designed for semantic segmentation: based on the pixel-wise outputs (such as a probability map), one needs to add a detector stage to actually perform object detection. However, if we thus decide on the detector in advance, we can manipulate the output masks on which we train the behavior of this detector. In the case of music notation, for symbols that may consist of multiple connected components or have complex shapes (the f-clef is an example that combines both), this can be attenuated by training on masks computed from their convex hulls rather than directly from their pixels [25]. Fortunately, as a side effect of using bounding box data in this paper to generate the rectangular pixel-wise masks, we are in essence already getting crude approximations of convex hulls. Note, however, that the bounding box data model thus forces the model to classify background pixels to belong to the symbol, which might otherwise be some way off; this is pronounced especially with beams that are slanted or close to each other.

By not considering the bounding boxes themselves at all during training, U-Nets avoid questions of granularity and the corresponding anchor box hyperparameters, which is a welcome property given the variability of musical symbol shapes—both inter-class and in some cases intra-class. On the



other hand, the arbitrary detector step, of course, introduces its own hyperparameters: the masking threshold, and the pixel merging strategy. One can consider the pixel-wise labels as a very fine-grained over-segmentation; the detector then acts as the over-segment merging step. The only architectural hyperparameter one has to set is the size of the receptive field of an output pixel, which is defined implicitly through the number of convolutional and max-pooling layers and their filter sizes; if we fix the size of the network, we can also trade off the receptive field size and resolution by downscaling the images.

**Model specifics** We follow the architecture of [26] and our U-Nets have four “depth” levels, as depicted in Figure 2. The final layer that produces the probability map uses  $1 \times 1$  convolutions with just one filter, with a sigmoid activation. (This is an efficient implementation of computing a weighted combination of the convolutional features for each pixel from the second-to-last layer.)

**Training setup** To go from bounding box ground truth to labels for each pixel, we render the rectangles specified by the bounding box ground truth as foreground. Each image is downscaled with a factor of 0.5. Training is not performed on entire images; instead, in each epoch, we uniformly sample a random  $256 \times 512$  window from each training image (corresponding to a  $512 \times 1024$  window from the original image). If this window contains no foreground pixel for the given class, we re-sample up to 5 times; this is a general way of slightly oversampling rare classes.

For each symbol class, one U-Net is trained with exactly the same setup. We use cross-entropy loss, using the Adam optimizer with the default parameters suggested in [47]. Batch size is set to 2. We use a learning rate attenuation schedule: starting from 0.001, if the validation loss does not improve for 50 epochs, we multiply the learning rate by 0.2, a process that is repeated five times. Again, none of these steps are domain specific.

Detection is then performed independently for each symbol class: in this setup, the fact that a pixel is classified as belonging to, e.g., a barline, does not preclude it from also being classified as a stem pixel (note that certain music notation symbols indeed overlap to a great extent, e.g., noteheads and ledger lines). As opposed to [25], we do not experiment with multi-channel outputs, as this is a step that already requires domain-specific knowledge. For the detection stage, we use simple thresholding at 50% and a connected component detector, this time following the setup of [25]. The detector does not output any natural confidence score, so we add a placeholder value of 1 for each detected foreground region.

#### 4.2. Datasets

As we are considering generic object detection methods, we can evaluate all of them across a range of OMR datasets for symbol detection [48]. As a side-effect of this evaluation, we also obtain a notion of the difficulty of these datasets for object detection in general. Each dataset contains a different kind of typography, adding to the breadth of the baselines we establish.

- **DeepScores:** DeepScores [49] is a very large synthetic dataset of music scores in Common Western Modern Notation (CWMN), consisting of 300,000 images along with their ground-truth annotations for performing symbol classification, image segmentation, and object detection. It is based on a large collection of freely available MusicXML files from MuseScore [50] that were converted into Lilypond files and digitally rendered into images using five different fonts to obtain a higher visual variability. The first version of this dataset only has annotations for a limited vocabulary that is missing essential glyphs, such as stems, beams, barlines, ledger lines or slurs. The second version, which is currently under development, contains these missing annotations and has been made available to us by the original authors. This set contains only 100 pages, but has full annotations for all relevant music symbols.
- **MUSCIMA++:** MUSCIMA++ [14] is a dataset of handwritten music that has over 90,000 manually annotated handwritten musical symbols in CWMN. The dataset is built on the CVC-MUSCIMA dataset for staff removal [51]. The ground truth is defined as a notation graph: in addition to the individual symbols, their relationships are annotated as well, so that the semantics (pitch,



duration, and onset) can be inferred and the full OMR pipeline can be trained on the dataset. However, in this paper, we only focus on symbol detection, equivalent to recovering the vertices of the notation graph.

- **Capitan:** Capitan consists of 46 fully-annotated pages in Spanish mensural notation from the 16th–18th century. The manuscripts represent sacred music, composed for vocal interpretation. The compositions were written in music books by different copyists of that time. To preserve the integrity of the physical sources, images of the manuscripts were taken with a camera instead of scanning them in a flatbed scanner, leading to suboptimal conditions in some cases. The corpus is based on the dataset used in the work of Pacha and Calvo-Zaragoza [24]. However, the refined version used in this work is focused on obtaining a diplomatic transcript, keeping the information of how symbols were written in the source as intact as possible. That is why there is a higher number of categories, since now symbols that have the same meaning—for example, a minima with the stem pointing up or down—are considered as different categories.

An overview of the corpora considered is given in Table 1, while we show some patches extracted from their images in Figure 3. As can be observed, the characteristics of the different corpora are quite heterogeneous, which is interesting for drawing generalizable conclusions from our experiments.

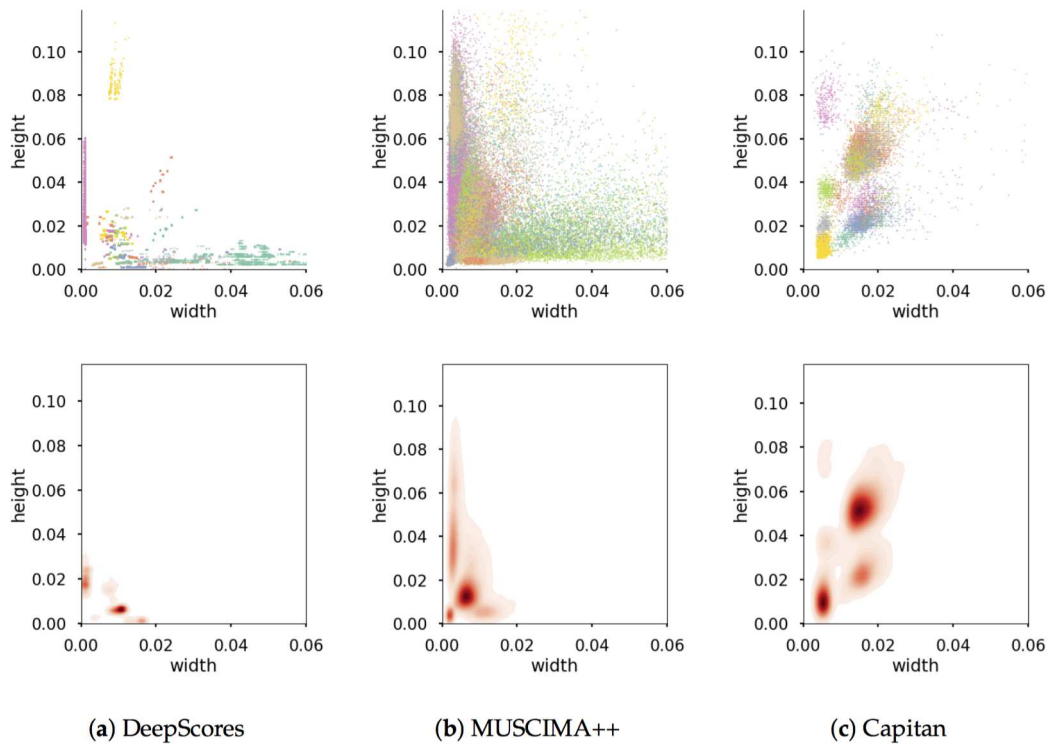
**Table 1.** Overview of the considered datasets.

Dataset	Notation	Engraving	Images	Categories	Scores	Symbols
DeepScores	CWMN	Printed	Binary	39	100	87,703
MUSCIMA++	CWMN	Handwritten	Binary	107	140	91,254
Capitan	Mensural	Handwritten	Color	56	46	11,242

It is important to mention the variability in the aspects of the bounding boxes of the elements within these datasets. This variability appears not only amongst elements of different classes but also, especially in the case of handwritten notation, amongst elements of the same class. To illustrate this scenario, Figure 4 shows the different shapes of the boxes to be recognized in each dataset. The majority of objects in the DeepScores dataset are very tiny. The MUSCIMA++ dataset shows a greater variation in aspect ratios with one dominant cluster, the noteheads. In addition, the Capitan dataset contains a significant number of bigger objects, compared to the other two datasets with distinct clusters.



**Figure 3.** Samples of notation from the considered datasets.



**Figure 4.** Scatter plot (**top**) row and density plot (**bottom**) row of the normalized object sizes for the considered corpora to illustrate the challenges of each dataset (best viewed in color). Each point in the top row depicts one instance from the dataset with the color encoding the respective class. The width and height of a sample are reported as the fraction of the full image size.

To evaluate the models in the different corpora, we followed a fixed partitioning scheme for training, validating, and testing. Therefore, the experiments are reproducible, and future results will be directly comparable. Specifically, 60% of the available data is used for training, to learn the values of the neural models; 20% for validation and hyperparameter optimization; and 20% for testing and computing the final evaluation metrics.

#### 4.3. Evaluation

As stated in Section 3, our formulation expects models to provide a set of detection proposals, each of which consists of a bounding box and the recognized class of the object therein. The models are also expected to provide a score of their confidence for each proposal. A bounding box proposal  $B_p$  is considered a positive sample if it overlaps with the ground-truth bounding box  $B_g$  according to the Intersection over Union (IoU) criterion

$$\frac{\text{area}(B_p \cap B_g)}{\text{area}(B_p \cup B_g)}$$

exceeding a certain threshold ( $t_{\text{IoU}}$ ). If the predicted category matches the actual category of the object, it is considered a true positive (TP), being otherwise a false positive (FP). Additional detections of the same object are considered as false positives as well. Those ground-truth objects for which the model makes no proposal are considered false negatives (FN). From these values, precision (P) and recall (R) metrics can be computed as

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}.$$



$P$  measures how reliable detections are (ratio of correct detections), whereas  $R$  measures the ability of the model to detect symbols (ratio of detected symbols).

Object detection can be seen as a retrieval task, in which bounding boxes are ordered by their associated scores. Then,  $P$  and  $R$  can be computed as previously described from the top  $k$  predictions. However, different values of  $P$  and  $R$  are obtained by varying the parameter  $k$ . To obtain a single metric encompassing the performance of the model, the average precision (AP) can be computed, which is defined as the area under the precision–recall curve for all possible values of  $k$ .

A single AP value is obtained independently for each class, and then the mean AP (mAP) is computed as the average across all classes. Since our problem is highly unbalanced with respect to the number of objects of each class, we also compute the weighted mAP (w-mAP), in which the mean value is weighted according to the frequency of each class. The difference between mAP and w-mAP gives a quick idea of how the evaluated models deal with the rare classes.

When  $t_{IoU}$  is set to 50%, the described evaluation protocol matches the *PASCAL Visual Object Classes (VOC)* challenge [52]. The accuracy of the localization is especially important for OMR, as objects are often packed densely. Failing to locate them correctly heavily affects the subsequent recognition. To account for this, we average mAP and w-mAP over different values of  $t_{IoU}$ , ranging from 50% to 95% by steps of 5%. This evaluation protocol is taken from the COCO challenge [17], and it is expected to provide figures that are more sensitive to precise symbol localization.

## 5. Results

The aggregate detection performance of the individual models over each of the datasets is reported in Table 2, presenting both mAP and w-mAP as defined for the COCO challenge [17]. These results should serve as the baseline for further music object detection research. Generally, it can be observed that the results are still very far from the optimal. The evaluated models struggle most with the MUSCIMA++ dataset, with the U-Net performing best at around 16% mAP and 33% w-mAP. It might be that the comparison is not entirely fair since the U-Net was specially designed for this dataset. However, U-Net outperforms the rest of the models in the case of DeepScores as well, where it attains around 24% in both mAP and w-mAP, leaving Faster R-CNN and RetinaNet below 20% and 10%, respectively, in both metrics. Concerning the Capitan dataset, all models behave quite similarly, except for the superior performance from RetinaNet regarding the w-mAP metric.

**Table 2.** Results in terms of mAP (%) and w-mAP (%) with respect to the dataset and object detector model following the COCO evaluation protocol.

	mAP (%)			w-mAP (%)		
	DeepScores	MUSCIMA++	Capitan	DeepScores	MUSCIMA++	Capitan
<b>Faster R-CNN</b>	19.6	3.9	15.2	14.4	7.9	23.2
<b>RetinaNet</b>	9.8	7.7	14.5	1.9	4.9	34.9
<b>U-Net</b>	24.8	16.6	17.4	23.3	33.6	26.0

In general, Faster R-CNN performs better than RetinaNet. However, it is especially sensitive to the selection of hyperparameters that regulate the shape and scale of the objects to be detected. The high variability in the bounding box shapes shown in Figure 4 might explain why Faster R-CNN is far from offering the performance it demonstrates for detecting objects in natural images. Compared to previous works that reported 80% mAP for snippets [23] and 76% mAP for full pages [24], a few differences need to be pointed out to understand the large difference between the numbers: the experiments from this work used less training data due to a stricter dataset split, the vocabulary of the Capitan dataset became larger and the final results are computed following the strict COCO evaluation protocol as opposed to reporting the PASCAL VOC metrics [52].

In the case of RetinaNet, an in-depth analysis of its operation reveals that it is not capable of detecting small objects. This explains the noticeable discrepancy between their mAP and w-mAP in DeepScores, where the noteheads—small objects—are the most represented category. Note that Faster R-CNN also exhibits this behavior on the DeepScores dataset, where more frequent symbols are also more problematic for the model than the more rare symbols.

In practical settings, inference speed, and in some situations (re-)training speed, can offset small differences in detection performance. We give a rough comparison when running the experiments on a standard consumer PC, equipped with a GTX 1080 graphics card:

- **Faster R-CNN:** Training time: 8–12 h; inference time: 20–50 s per image,
- **RetinaNet:** Training time: 1–2 h; inference time: less than 1 s per image,
- **U-Net:** Training time: 2–3 h per symbol class; inference time: 40–80 s per image, or about 0.8 s per symbol class.

In this comparison, the RetinaNet has a clear advantage: if one were to find a way to improve its accuracy to an acceptable level, it would be a clear champion for interactive OMR or online recognition settings. U-Nets, on the other hand, are impractical for situations where frequent re-training is needed: unless one has a cluster of graphical processing units (GPUs), training even the minimum 30+ classes that are necessary for pitch and duration inference would take several days.

### Qualitative Results

To illustrate the differences in performance, we show samples of detector outputs across the three datasets for some selected classes. Figure 5 shows how the detectors fare with the born-digital printed music of DeepScores. As the rendered symbols have relatively little variability, this sample allows for comparing the strengths and weaknesses of the models' designs, especially with respect to music notation data.

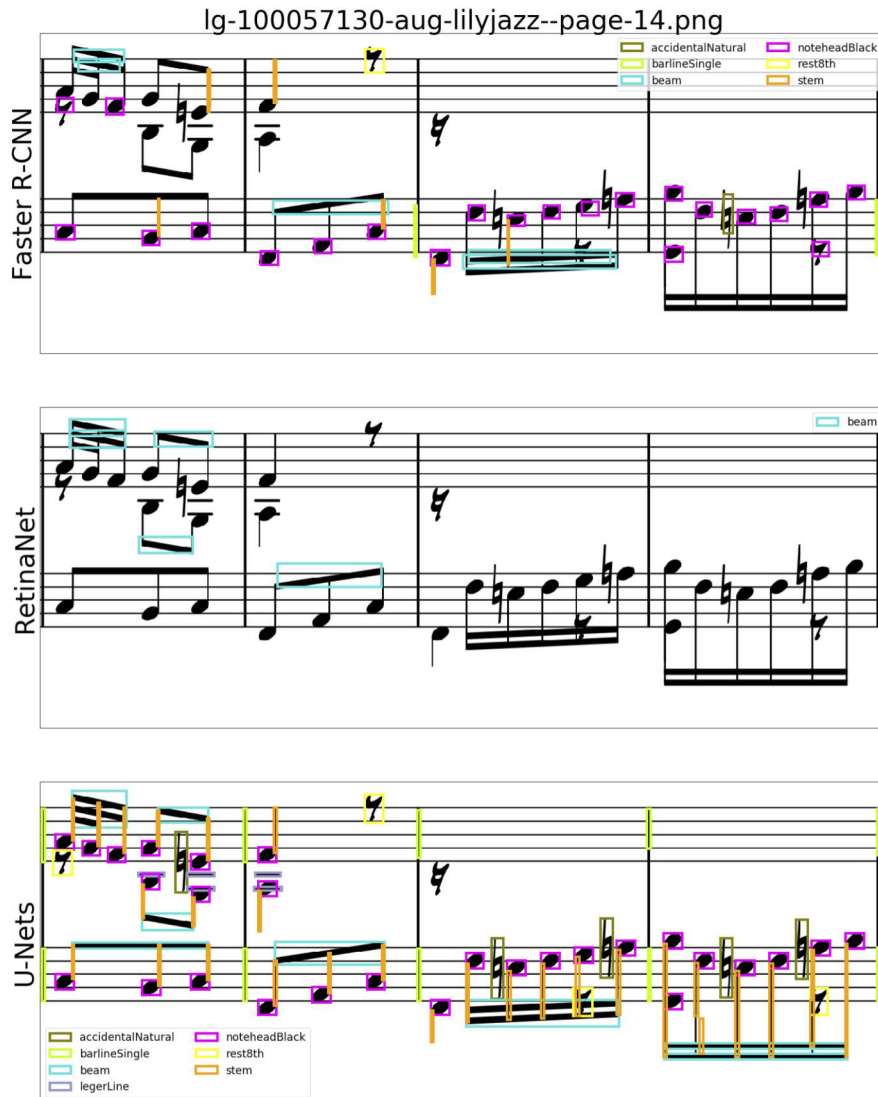
The Faster R-CNN model (Figure 5 top) has trouble with symbols that are bunched together closely, especially in the upper left corner. This may be due to too few available proposals in a given region. On the other hand, it can distinguish slanted parallel beams (first and third measure). The RetinaNet (Figure 5 middle) is unable to deal with symbols smaller than the beams and does not even find all of them. The U-Nets (Figure 5 bottom) shine in this specific example, perhaps a bit more than the quantitative results suggest: they also recover the heavily overlapping eighth rest in the third and fourth measures. On the other hand, the inherent limitation of the connected component detector causes beams with overlapping bounding boxes to get lumped together. If one were to choose an image with dense chords, noteheads within a chord would also invariably get merged into one.

Detection performance on the MUSCIMA++ dataset (Figure 6) displays a similar pattern. The RetinaNet again cannot detect anything but the large objects; Faster R-CNN again seems to run out of proposals in cluttered regions, or perhaps proposals get inadvertently merged into one due to insufficient feature map resolution. U-Nets are lucky in this image: the descending thirds in the first measure are just far enough from each other so that they get detected separately; if they were as close to each other as the bottom two noteheads on the third and fourth beat of the second measure of the sample, they would get merged into one. Beams, even though their bounding boxes do not necessarily overlap (bottom staff, second measure), again get merged, and there are false positive beams in hairpins.

On the Capitan dataset, the situation changes, as illustrated in Figure 7. We hypothesize that the main driver for this difference is the change in symbol class definition: instead of using notation primitives such as noteheads or stems, the Capitan dataset uses composite symbols such as *note.quarter-up*, *note.beamedLeft1*. This discrepancy in defining music notation objects has persisted throughout the literature on music object detection [19].



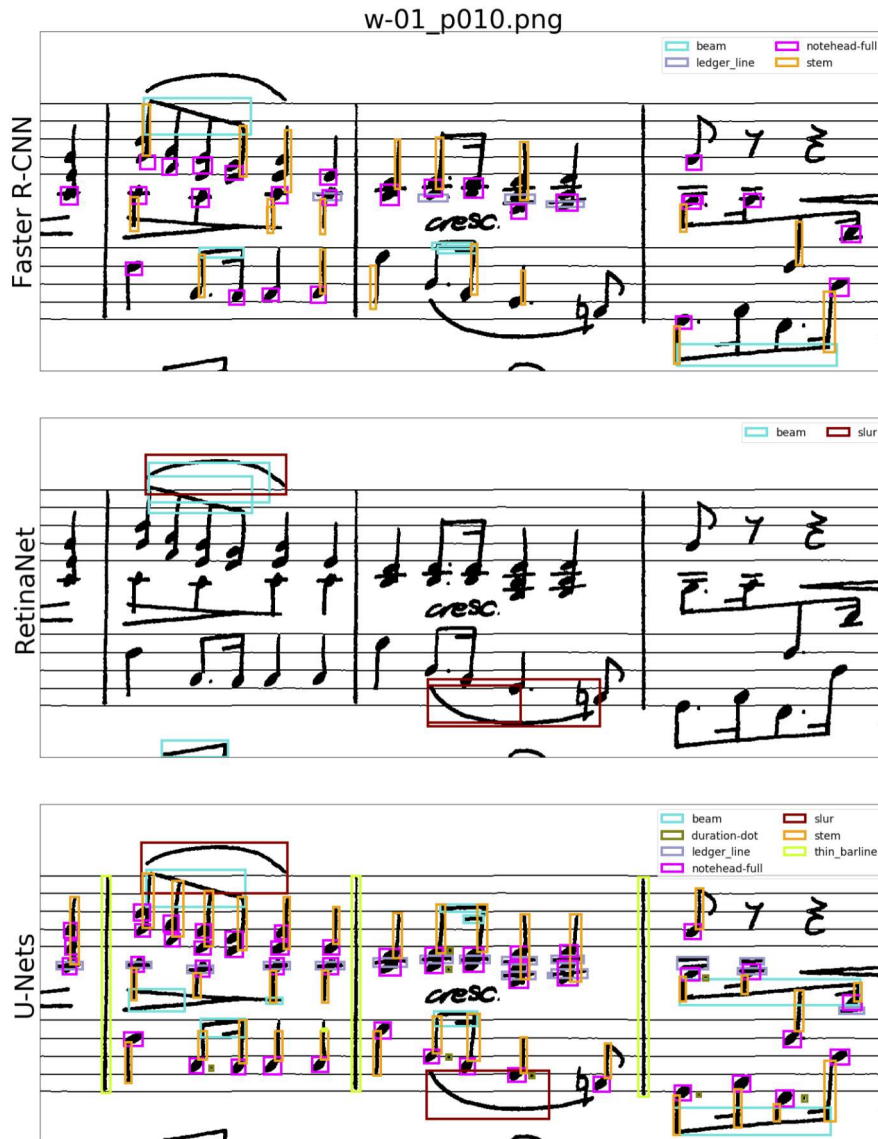
## Detection results sample: DeepScores



**Figure 5.** Detection sample on some selected classes from the DeepScores dataset. (top–bottom): Faster R-CNN, RetinaNet, and U-Nets detection results.

This presents a problem for the U-Nets: the most prominent feature of a note, whether facing down or up, is the notehead. As the symbols are processed independently, there is a risk that noteheads will be detected as instances of all applicable objects according to the notehead type. If one looks at the U-Nets' output (Figure 7 bottom), e.g., the middle of the second staff on the second page, eighth notes get classified as quarter notes, and half-note stems fool the quarter-note detector into false positives. In addition, as the symbols get larger, the U-Net runs into one of its inherent risks concerning the connected components detector: symbol fragmentation. As the pixels of symbols that are easily classified tend to be on their extremes, the system may become less certain in their centers, and the symbol falls apart after thresholding the U-Net output probability map. We have observed this behavior on barlines and long stems on the MUSCIMA++ dataset as well. This breakup produces many false positives (in Figure 7, especially for quarter notes).

## Detection results sample: MUSCIMA++



**Figure 6.** Detection sample on some selected classes from the MUSCIMA++ dataset. (top–bottom): Faster R-CNN, RetinaNet, and U-Nets detection results.

On the other hand, while Faster R-CNN still struggles—although to a much smaller extent—with false negatives, RetinaNet does not face too small symbols anymore, and learns well: when symbol class frequencies are used to weight the result, it outperforms both contenders by a large margin. It falls into none of the U-Nets’ traps.

What can we say regarding the datasets?

For DeepScores, our results seem to confirm the intentions of the dataset authors: the main difficulty of the dataset is the large number of tiny objects [49]. While Faster R-CNN does outperform the same baseline architecture of [49] (which, according to the authors, does not detect anything at all), it does still encounter the limitations that they expected of this class of models. The single-shot RetinaNet detector runs into even worse trouble (and thus the authors of [49] were probably right to not use single-shot detection at all).



12642.png

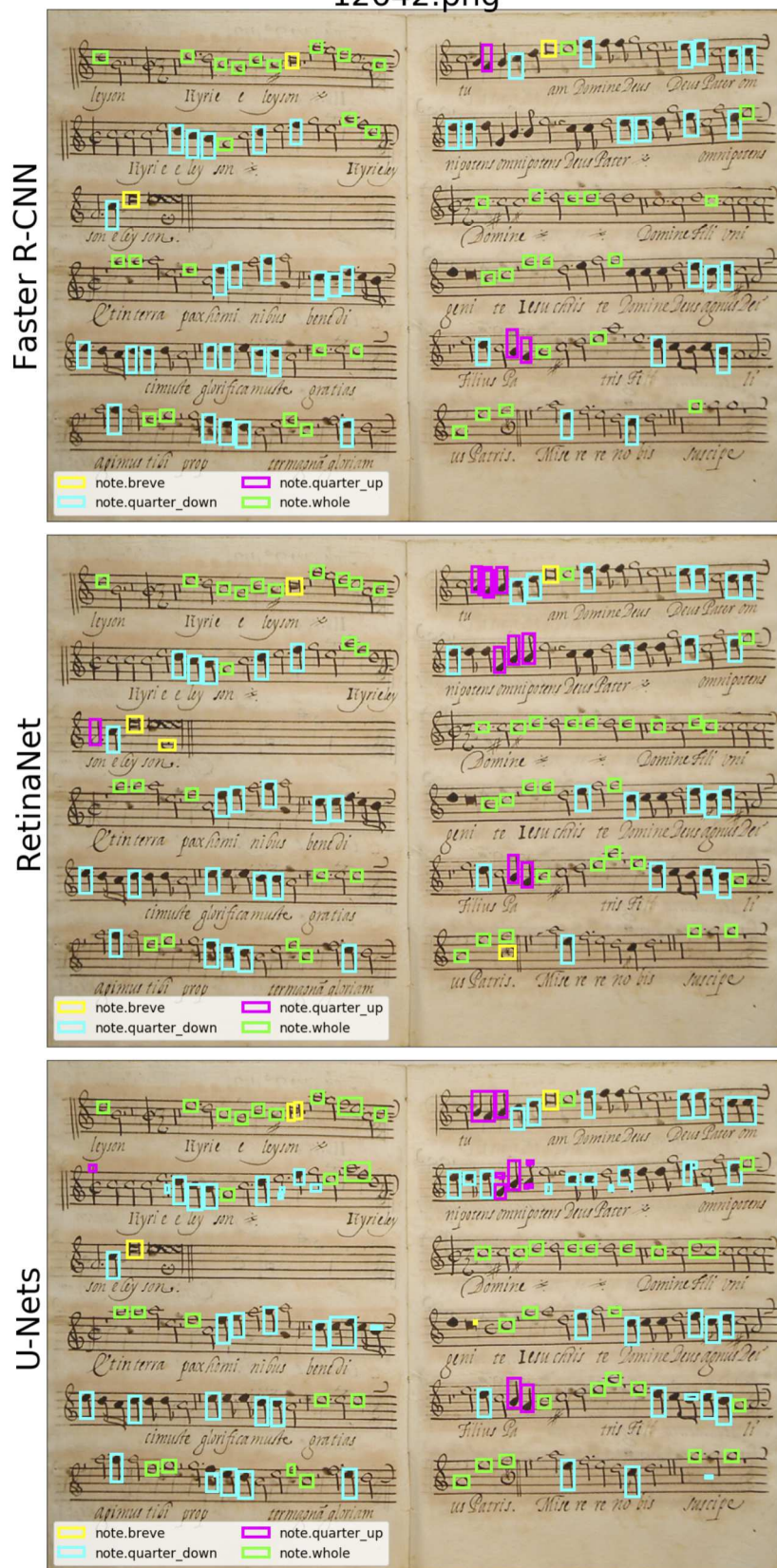


Figure 7. Detection sample on some selected classes from the Capitan dataset. (top–bottom): Faster R-CNN, RetinaNet, and U-Nets detection results.

The Capitan dataset seems to present a more straightforward object detection challenge. The close relationship of the composite object classes does not seem to be a problem for standard detectors; semantic segmentation, however, struggles.

From the perspective of music object detection, the MUSCIMA++ dataset has turned out to be essentially a more difficult version of DeepScores: the ground truth is defined at the level of notation primitives, the music contained in the datasets has similar complexity, but MUSCIMA++ is handwritten, which makes the shapes more variable, and topological features such as corners less reliable.

## 6. Conclusions

In this work, we establish a baseline for detecting music notation objects with deep learning models for generic object detection. Experiments were performed over three diverse major OMR datasets: the synthesized DeepScores dataset of born-digital modern notation, the MUSCIMA++ dataset of handwritten modern notation with varying degrees of writing quality, and the Capitan dataset that contains mensural notation which is also handwritten, but of consistently high quality. Three types of neural models have been evaluated, namely the two-stage Faster R-CNN detector, the one-stage RetinaNet detector, and the U-Net detection mechanism that combines flexible semantic segmentation with a connected component detector. The choice of experimental setup and evaluation in this paper can serve as a basis for further music object detection experiments that will, therefore, be directly comparable to these baselines and will enable drawing conclusions and model design recommendations from these direct comparisons.

Based on the quantitative and qualitative results in this paper, can we already formulate tentative practical recommendations for choosing a certain detection approach over another? We are well aware that three datasets may not be enough to draw such general conclusions; however, it is the most comprehensive experimentation that the current state of the OMR concerning available data allows. The suggestions should, therefore, be treated as tentative suggestions for further targeted investigations rather than fully-fledged conclusions.

U-Nets, except for merging nearby symbols of the same class, do not seem to have a problem with the recall. Because they process symbol classes independently and do not reduce the output features resolution, they cannot run into the same (hypothesized) problems as Faster R-CNN, which has a limited number of region proposals for any single region of the image that the symbols in effect compete for. The number of available proposals depends on a hyperparameter setting that might be difficult to set appropriately for areas densely populated of ground truth objects. Furthermore, the proposal merging step (such as non-maximum suppression) may also lead to false negatives in cluttered environments. None of these disadvantages concern the U-Nets.

On the other hand, while these properties are ideal for very cluttered data where symbol classes are set to notation primitives, the design drawbacks of U-Nets do appear when the symbol vocabulary consists of composite symbols; conversely, this is where the cluttering that presumably hinders the bounding box-based models ceases to be an important factor, and the relative strength of these models—the ability to consider a particular region as a whole—becomes more relevant because composite symbols share visual elements that correspond to the primitives. The choice of a musical symbol detection model, therefore, seems to be based on the way the detection ground truth is defined.

Now that a deep learning baseline for music object detection has been established, where can subsequent research be heading?

First, one can use the first insights gained from comparing the models over various datasets to improve the music object detectors themselves. The weak point of U-Nets seems to be settings with composite objects; experiments with composites built from MUSCIMA++ primitives by leveraging their syntactic relationships would be a logical step to investigate this. In order for U-Nets to improve on datasets with composite symbols (which are cheaper to annotate, as they generally contain fewer symbol instances, and therefore more likely to be encountered during various music digitization



efforts), a combination of the pixel-wise approach, which deals very well with highly cluttered areas or occlusion, and combined properties of the resulting pixel groups can be a viable avenue, while also perhaps alleviating the problem of parallel beams. In [28], a YOLOv3-like approach has been used to detect noteheads with joint pixel classification and bounding box regression. A post-filtering step then significantly improved precision, which is a much bigger problem for U-Nets than recall. The Deep Watershed Detector used by [27] exhibits a similar combination.

For improving the Faster R-CNN results on music notation data, we would need a better understanding of the relationship between anchor hyperparameters and expected symbol density. The inability of the RetinaNet to detect small symbols is disappointing and merits further investigation, as it persisted regardless of various anchor hyperparameter settings. An idea to test the hypothesis of some minimum detectable absolute symbol size would be to upscale the image until the objects of interest reach sufficient size, and run detection on windows of the upscaled image that fit into GPU memory. The speed of this model both in training and inference would make it an attractive choice for interactive OMR, which is now probably the most viable approach towards building OMR systems that can best support creating digital editions of music, such as the Ceres system [53] or the Pixel.js editor [54].

More can also be done in terms of evaluation to make the baseline more informative regarding the outputs expected from OMR downstream. While music object detection is a critical step in OMR pipelines, it is not the final step; for evaluating a detector as part of an OMR system, one should be able to attribute downstream errors, e.g., in pitch or duration inference, to detection errors or uncertainties. For instance, Ref. [25] uses several ways of evaluating MIDI inferred on top of the object detection results, using a baseline reconstruction model. Furthermore, the graph model of MUSCIMA++ offers hope that the edges can serve as “conduits” from higher-level errors to their lower-level causes, but, so far, we are not aware of any method that would allow combining such structured gradient flows with the object detection architectures.

Then, there are exciting challenges of transfer learning. Modern notation follows the same underlying rules, regardless of whether it is printed or handwritten: can one leverage a printed music dataset to train for handwritten object detection? At least between DeepScores and MUSCIMA++, many symbol classes can be directly mapped onto each other—experiments in this direction should be possible. In this context, the effect of image deformations and other, perhaps more realistic data augmentation can be explored.

Finally, while it is obvious that merely detecting the musical elements in score images does not represent a complete OMR system, we believe that addressing music object detection in a generic machine learning manner brings a series of changes that are quite interesting for the development of the OMR field. Except for the few attempts at end-to-end OMR that are so far limited to monophonic output [7,8,55], all OMR systems are explicitly detecting music objects at some point in their recognition pipeline. Generic deep learning approaches may have the potential to decouple object detection from actual knowledge of music notation itself—nevertheless, users now need to be aware of how these systems learn and design them accordingly. The proposed general machine learning approach can then be used by all of them, regardless of the musical notation system (except for hyperparameter tuning and cookbook-style model choice recommendations), as opposed to approaches that exploit specific characteristics of how the music notation system works to build segmentation heuristics. Then, as the music object detection stage is done, image processing can in principle be forgotten: the only remaining link to the original image is the bounding box and potentially pixel mask features associated with the detected objects. The remaining stages—notation reconstruction and exporting an output representation—then, in turn, do not require computer vision knowledge (while now requiring, of course, some understanding of how music notation stores content). On the other hand, one can utilize the syntactic regularities of music notation to improve the object detection stage (and perhaps perform detection and relational understanding jointly). Incorporating the graph structure, and further prior knowledge about the properties of music notation (such as expected voice leading), into a

differentiable loss function that can be optimized by the neural network learning process, represents an interesting avenue for future research. Both approaches, therefore, open up the possibility for experts from different areas to establish a synergy that pushes the development of the OMR field from both perspectives.

**Author Contributions:** A.P., J.H. and J.C.-Z. all contributed equally.

**Funding:** The authors wish to thank the TU Wien Bibliothek for the financial support through its Open Access Funding Program. The second author additionally acknowledges support by the Czech Science Foundation Grant No. P103/12/G084, Charles University Grant Agency grants 1444217 and 170217, and by SVV project 260 453. The third author additionally acknowledges the support from the Spanish Ministerio de Ciencia, Innovación y Universidades through a Juan de la Cierva Formación grant (Ref. FJCI-2016-27873).

**Conflicts of Interest:** The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

## Abbreviations

The following abbreviations and acronyms are used in this manuscript:

OMR	Optical Music Recognition
IMSLP	International Music Score Library Project
SIMSSA	Single Interface for Music Score Searching and Analysis
MIR	Music Information Retrieval
MuNG	Music Notational Graph
MIDI	Musical Instrument Digital Interface
MEI	Music Encoding Initiative
MUSCIMA	Music Score Images
COCO	Common Objects in Context
PASCAL	Pattern Analysis, Statistical Modelling and Computational Learning
VOC	Visual Object Classes
R-CNN	Region-based Convolutional Neural Network
API	Application Programming Interface
SSD	Single Shot Detector
YOLO	You Only Look Once
CWMN	Common Western Modern Notation
IoU	Intersection over Union
mAP	Mean Average Precision
GPU	Graphics Processing Unit
ELU	Exponential Linear Unit

## References

1. Craig-McFeely, J. Digital Image Archive of Medieval Music: The evolution of a digital resource. *Digit. Med.* **2008**, *3*. [[CrossRef](#)]
2. The International Music Score Library Project. Available online: <http://imslp.org/> (accessed on 28 August 2018).
3. Fujinaga, I.; Hankinson, A.; Cumming, J.E. Introduction to SIMSSA (Single Interface for Music Score Searching and Analysis). In Proceedings of the 1st International Workshop on Digital Libraries for Musicology, London, UK, 12 September 2014; pp. 1–3.
4. Fujinaga, I. Optical Music Recognition Using Projections. Master's Thesis, McGill University, Montreal, QC, Canada, 1988.
5. Blostein, D.; Baird, H.S. A Critical Survey of Music Image Analysis. In *Structured Document Image Analysis*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 405–434.
6. Pacha, A.; Eidenberger, H. Towards Self-Learning Optical Music Recognition. In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 18–21 December 2017; pp. 795–800.



7. Shi, B.; Bai, X.; Yao, C. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *11*, 2298–2304. [[CrossRef](#)] [[PubMed](#)]
8. Van der Wel, E.; Ullrich, K. Optical Music Recognition with Convolutional Sequence-to-Sequence Models. In Proceedings of the 18th International Society for Music Information Retrieval Conference, Suzhou, China, 23–27 October 2017.
9. Choi, K.Y.; Coüasnon, B.; Ricquebourg, Y.; Zanibbi, R. Bootstrapping Samples of Accidentals in Dense Piano Scores for CNN-Based Detection. In Proceedings of the 12th IAPR International Workshop on Graphics Recognition, Kyoto, Japan, 9–10 November 2017.
10. Calvo-Zaragoza, J.; Rizo, D. End-to-End Neural Optical Music Recognition of Monophonic Scores. *Appl. Sci.* **2018**, *8*, 606. [[CrossRef](#)]
11. Byrd, D.; Simonsen, J.G. Towards a Standard Testbed for Optical Music Recognition: Definitions, Metrics, and Page Images. *J. New Music Res.* **2015**, *44*, 169–195. [[CrossRef](#)]
12. Hajič jr., J.; Novotný, J.; Pecina, P.; Pokorný, J. Further Steps towards a Standard Testbed for Optical Music Recognition. In Proceedings of the 17th International Society for Music Information Retrieval Conference, New York, NY, USA, 7–11 August 2016; Mandel, M., Devaney, J., Turnbull, D., Tzanetakis, G., Eds.; New York University: New York, NY, USA, 2016; pp. 157–163.
13. Rebelo, A.; Fujinaga, I.; Paszkiewicz, F.; Marcal, A.R.; Guedes, C.; Cardoso, J.S. Optical music recognition: state-of-the-art and open issues. *Int. J. Multimed. Inf. Retr.* **2012**, *1*, 173–190. [[CrossRef](#)]
14. Hajič, J.J.; Pecina, P. The MUSCIMA++ Dataset for Handwritten Optical Music Recognition. In Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition, Kyoto, Japan, 10–15 November 2017.
15. Calvo-Zaragoza, J.; Castellanos, F.J.; Vigiensoni, G.; Fujinaga, I. Deep Neural Networks for Document Processing of Music Score Images. *Appl. Sci.* **2018**, *8*, 654. [[CrossRef](#)]
16. Bainbridge, D.; Bell, T. A music notation construction engine for optical music recognition. *Software* **2003**, *33*, 173–200. [[CrossRef](#)]
17. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755.
18. Music Object Detection Repository on Github. Available online: <http://github.com/apacha/MusicObjectDetection> (accessed on 28 August 2018).
19. Bellini, P.; Bruno, I.; Nesi, P. Assessing Optical Music Recognition Tools. *Comput. Music J.* **2007**, *31*, 68–93. [[CrossRef](#)]
20. Dalitz, C.; Droettboom, M.; Pranzas, B.; Fujinaga, I. A Comparative Study of Staff Removal Algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 753–766. [[CrossRef](#)] [[PubMed](#)]
21. Fornés, A.; Dutta, A.; Gordo, A.; Lladós, J. The 2012 Music Scores Competitions: Staff Removal and Writer Identification. In *Graphics Recognition, Proceedings of the 9th International Workshop, Seoul, Korea, 15–16 September 2011*; Kwon, Y.B., Ogier, J.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 173–186.
22. Gallego, A.J.; Calvo-Zaragoza, J. Staff-line removal with selectional auto-encoders. *Expert Syst. Appl.* **2017**, *89*, 138–148. [[CrossRef](#)]
23. Pacha, A.; Choi, K.Y.; Coüasnon, B.; Ricquebourg, Y.; Zanibbi, R.; Eidenberger, H. Handwritten Music Object Detection: Open Issues and Baseline Results. In Proceedings of the 2018 13th IAPR Workshop on Document Analysis Systems (DAS), Vienna, Austria, 24–27 April 2018.
24. Pacha, A.; Calvo-Zaragoza, J. Optical Music Recognition in Mensural Notation with Region-Based Convolutional Neural Networks. In Proceedings of the 19th International Society for Music Information Retrieval Conference, Paris, France, 23–27 September 2018.
25. Hajič jr., J.; Dorfer, M.; Widmer, G.; Pecina, P. Towards Full-Pipeline Handwritten OMR with Musical Symbol Detection by U-Nets. In Proceedings of the 19th International Society for Music Information Retrieval Conference, Paris, France, 23–27 September 2018.
26. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention, Proceedings of the 18th International Conference, Munich, Germany, 5–9 October 2015*; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.

27. Tuggener, L.; Elezi, I.; Schmidhuber, J.; Stadelmann, T. Deep Watershed Detector for Music Object Recognition. In Proceedings of the 19th International Society for Music Information Retrieval Conference, Paris, France, 23–27 September 2018.
28. Hajič, J.; Pecina, P. Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression. *arXiv* **2017**, arXiv:1708.01806.
29. Coüason, B.; Brisset, P.; Stéphan, I. Using Logic Programming Languages For Optical Music Recognition. In Proceedings of the Third International Conference on the Practical Application of Prolog, Paris, France, 3–6 April 1995.
30. Villegas, M.; Sánchez, J.A.; Vidal, E. Optical modelling and language modelling trade-off for Handwritten Text Recognition. In Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Nancy, France, 23–26 August 2015; pp. 831–835.
31. Chen, L.; Hermans, A.; Papandreou, G.; Schroff, F.; Wang, P.; Adam, H. MaskLab: Instance Segmentation by Refining Object Detection with Semantic and Direction Features. *arXiv* **2017**, arXiv:1712.04837.
32. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems 28*; Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA 2015; pp. 91–99.
33. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2014; pp. 580–587.
34. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
35. Uijlings, J.R.R.; van de Sande, K.E.A.; Gevers, T.; Smeulders, A.W.M. Selective Search for Object Recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [[CrossRef](#)]
36. Zitnick, L.; Dollar, P. Edge Boxes: Locating Object Proposals from Edges. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014.
37. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
38. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S.; et al. Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
39. Lin, T.; Goyal, P.; Girshick, R.B.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *arXiv* **2017**, arXiv:1708.02002.
40. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv* **2013**, arXiv:1312.6229.
41. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37.
42. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
43. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, CA, USA, 26 June–1 July 2016; pp. 770–778.
44. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
45. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
46. Huang, G.; Liu, Z.; Weinberger, K.Q. Densely Connected Convolutional Networks. *arXiv* **2017**, arXiv:1608.06993.



47. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
48. The OMR datasets project on Github. Available online: <http://apacha.github.io/OMR-Datasets/> (accessed on 28 August 2018).
49. Tuggener, L.; Elezi, I.; Schmidhuber, J.; Pelillo, M.; Thilo, S. DeepScores—A Dataset for Segmentation, Detection and Classification of Tiny Objects. In Proceedings of the 24th International Conference on Pattern Recognition, Beijing, China, 20–28 August 2018.
50. MuseScore. The free and open-source score writer. Available online: <http://musescore.org> (accessed on 28 August 2018).
51. Fornés, A.; Dutta, A.; Gordo, A.; Lladós, J. CVC-MUSCIMA: A ground truth of handwritten music score images for writer identification and staff removal. *Int. J. Doc. Anal. Recognit.* **2012**, *15*, 243–251. [[CrossRef](#)]
52. Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [[CrossRef](#)]
53. Chen, L.; Jin, R.; Raphael, C. Human-Guided Recognition of Music Score Images. In Proceedings of the 4th International Workshop on Digital Libraries for Musicology, Shanghai, China, 28 October 2017.
54. Saleh, Z.; Zhang, K.; Calvo-Zaragoza, J.; Vigliensoni, G.; Fujinaga, I. Pixel.js: Web-Based Pixel Classification Correction Platform for Ground Truth Creation. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 10–15 November 2017; pp. 39–40.
55. Calvo-Zaragoza, J.; Rizo, D. Camera-PrIMuS: Neural End-to-End Optical Music Recognition on Realistic Monophonic Scores. In Proceedings of the 19th International Society for Music Information Retrieval Conference, Paris, France, 23–27 September 2018.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

## 7.5 How Current Optical Music Recognition Systems Are Becoming Useful for Digital Libraries

Jan Hajič jr., Marta Kolářová, Alexander Pacha, Jorge Calvo-Zaragoza. How current optical music recognition systems are becoming useful for digital libraries. *Proceedings of the 5th International Conference on Digital Libraries for Musicology*, pages 57–61, Paris, France, 2018. ISBN 978-1-4503-6522-2.

The paper **How Current Optical Music Recognition Systems Are Becoming Useful for Digital Libraries**. updates general expectations of applicability of state-of-the-art OMR systems within the context of digital libraries: the performance of OMR systems in a retrieval setting, symbolic music similarity setting, and in replicating a prototypical digital musicology study is showcased and discussed, including its limitations.

The dissertation author designed the structure of the paper, the experiments, and evaluation methodologies, carried out the retrieval experiments, carried out the coding part of digital musicology replication section, and wrote most of the text except for section 5. The co-author Marta Kolářová selected the digital musicology case study and prepared the corresponding data; co-author Jorge Calvo-Zaragoza contributed the model and experiments for section 5 (symbolic music similarity), and co-author Alexander Pacha contributed to the text of the article. The contribution of the dissertation author to this article is about 65–70%.

# How Current Optical Music Recognition Systems Are Becoming Useful for Digital Libraries

Jan Hajič jr.  
Charles University, Czech Republic  
hajicj@ufal.mff.cuni.cz

Alexander Pacha  
TU Wien, Austria  
alexander.pacha@tuwien.ac.at

Marta Kolárová  
Charles University, Czech Republic  
marta.kolarova@ff.cuni.cz

Jorge Calvo-Zaragoza  
Universitat Politècnica de València, Spain  
jcalvo@prhlt.upv.es

## ABSTRACT

Optical Music Recognition (OMR) promises to make large collections of sheet music searchable by their musical content. It would open up novel ways of accessing the vast amount of written music that has never been recorded before. For a long time, OMR was not living up to that promise, as its performance was simply not good enough, especially on handwritten music or under non-ideal image conditions. However, OMR has recently seen a number of improvements, mainly due to the advances in machine learning. In this work, we take an OMR system based on the traditional pipeline and an end-to-end system, which represent the current state of the art, and illustrate in proof-of-concept experiments their applicability in retrieval settings. We also provide an example of a musicological study that can be replicated with OMR outputs at much lower costs. Taken together, this indicates that in some settings, current OMR can be used as a general tool for enriching digital libraries.

## CCS CONCEPTS

• **Information systems** → **Music retrieval**; *Image search*; • **Applied computing** → **Digital libraries and archives**; *Document searching*; **Graphics recognition and interpretation**;

## KEYWORDS

Optical Music Recognition, Music Information Retrieval, Symbolic Music Search, Music Digital Libraries, Digital Musicology

### ACM Reference Format:

Jan Hajič jr., Marta Kolárová, Alexander Pacha, and Jorge Calvo-Zaragoza. 2018. How Current Optical Music Recognition Systems Are Becoming Useful for Digital Libraries. In *Proceedings of Digital Libraries for Musicology (DLfM'18)*. ACM, New York, NY, USA, Article 4, 5 pages. <https://doi.org/10.1145/3273024.3273034>

## 1 INTRODUCTION

Optical Music Recognition (OMR), the field of computationally reading music notation in documents, is long known to hold significant promise for music libraries. The ability to search in vast archives

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).  
DLfM '18, September 28, 2018, Paris, France  
© 2018 Copyright is held by the owner/author(s).  
ACM ISBN 978-1-4503-6522-2/18/09.  
<https://doi.org/10.1145/3273024.3273034>

of musical manuscripts using their content rather than solely their metadata would open entirely new avenues of large-scale research in digital musicology. A large number of compositions have never been recorded or digitized before; most of them probably exist only as manuscripts, since typesetting music has historically been a costly endeavor. As OMR is a cost-effective alternative to arrive at a structured encoding, it is, therefore, a key to significantly diversify the digitally available sources both to the general and professional audience. This applies to works from the 20<sup>th</sup> and 21<sup>st</sup> centuries as well: many are currently collecting dust in composers' private collections because there are insufficient resources to typeset them.

OMR is also known not to work very well [5, 19], and existing methods are rarely applicable beyond specific collections of scores. However, we believe recent OMR advances (e.g., [3, 16]) warrant revisiting this assertion. The contribution of this paper is to provide evidence for digital librarians and musicologists that current approaches to OMR can make it applicable in the following downstream scenarios: content-based retrieval, especially at the page level, of handwritten scores; melodic similarity matching; and digital musicology studies based on data aggregation.

Furthermore, the current OMR state of the art relies purely on supervised machine learning. Therefore, rather than demonstrating the use of an OMR system within a specific project (e.g., [6, 8]), our paper can be interpreted to set general expectations on the performance of the given OMR methods across analogous application scenarios, as long as comparable training data is available. The OMR methods are independent from specific use-cases, to the point where one can follow a "cookbook" to apply them to a new collection; costs are mainly shifted onto manual supervised data acquisition, which is a standardized, predictable task that does not require competitive computer vision expertise.

## 2 RELATED WORK

The PROBADO project [7], the Levy Collection [6], the OMRAS project [8], the digital version of the Liber Usualis [1, 22] within the SIMSSA project [12], the RISM project<sup>1</sup>, the RILM project<sup>2</sup> and more recently PatternFinder [13] reflect the ongoing effort to create digital libraries of a large body of music and enable searching and indexing those collections. These systems feature powerful engines to evaluate a range of queries in an extensive database of symbolic music, e.g., searching by melody, by interval or looking

<sup>1</sup><http://opac.rism.info/metaopac/>

<sup>2</sup><http://www.rilm.org/>

up a particular note-sequence with optional wildcards. This power is enabled once a symbolic representation of the music is available – and without OMR, obtaining this representation has to be done manually, which is expensive and time-consuming.

Attempts have been made to use generic OMR to extract the requisite symbolic representation of music directly from musical score images, but the OMR component proved to be a weak point. In [10, 11], the authors describe how to match scanned sheet music to audio recordings automatically with an OMR algorithm doing the initial sheet music transcription. However, the evaluated algorithms produced such a large number of errors, that a subsequent correction was required before being able to match the OMR output to the audio representation. Similarly, in [2] the authors describe how to match musical themes from multiple sources using OMR, but the OMR output contained too many errors, and the authors had to resort to a drastically simplified representation, practically discarding note durations, clefs, and even absolute pitches.

OMR accuracy has been a significant bottleneck in the further development of similar applications to the extent that more success has been achieved by retrieving raw score images rather than their structured encodings [18] – but this does not provide the structured encodings that enable further processing and research.

### 3 OMR SYSTEMS

We showcase two OMR systems representing the current state of the art for obtaining the musical content from sheet music images. Both systems output a MIDI representation corresponding to the music score in the input image. First, we use a traditional full-pipeline system [16], applied to retrieval. Second, we use an end-to-end system [3]. Both systems are based on supervised learning with generic neural networks.

The full-pipeline approach (FP-OMR) builds off of the traditional OMR stages [20]. However, the detection method (U-Nets for semantic segmentation and a Connected Components detector [16]) jointly performs segmentation and classification on the input image directly, without removing the staff lines. Notation assembly is also performed with a machine learning method, as the Notation Graph representation [15] allows decomposing this step into a series of local decisions. MIDI is then inferred deterministically from the notation graph.<sup>3</sup> The advantages of this system are its applicability to arbitrarily complex music (given corresponding training data), the possibility of exporting the results in a rich representation such as MEI from the Notation Graph,<sup>4</sup> and its ability to operate on manuscripts, since the statistical methods can deal with the topological uncertainties of handwriting. Its disadvantages are that the symbol detection network is sensitive to low-level properties of the training dataset, thus requiring separate training sets for every source of data, and that the notation assembly model is currently underdeveloped.

The end-to-end approach (E2E-OMR) uses a convolutional recurrent neural network (CRNN) that is capable of providing the sequence of music symbols from the image of a single staff [3]. The term end-to-end signifies that the model is trained to directly produce the correct sequence of musical events, *without* providing

geometric information of where each symbol is located. Although this reduces the effort when creating the ground-truth data, the CRNN design is so far inherently limited to single-staff, monophonic music. The system has only been trained on born-digital printed scores, but with artificial distortions to simulate more realistic score images.

## 4 RETRIEVAL EXPERIMENTS

We define several retrieval tasks over a small test collection, evaluated with common retrieval metrics. The similarity between two MIDI files is computed using Dynamic Time Warping over the pitch sequences (discarding durations, which are still too unreliable), similar to [2]. We assume a human user will verify retrieved items from a ranked list and stop when the first non-duplicate score is returned. For this, we return Mean Average Precision (MAP@ $k$ ), where  $k$  is the number of duplicates for a given page in the collection (in our case, MAP@49).

As the retrieval collection, we use CVC-MUSCIMA [9]. This dataset contains 20 distinct pages of music, each copied by 50 people, for a total of 1000 images. Since the individual pieces exhibit a significant amount of variability, using the entire collection would make the problem extremely easy. For that reason, we select a *confuse-retrieval* subset of 7 pages. While decreasing the collection size would typically improve retrieval performance, in this case, the remaining 13 pieces are so distinctive that including them would make the collection *less* challenging. One advantage of this dataset is that we know in advance how many copies of each page exist in the database, so the experiments in this section can thus be seen as indicators of the general ability of the OMR system to deal with retrieving manuscripts with different handwriting styles.

We prepare all the MIDI representations of the score images in this section with the full-pipeline system (FP-OMR), as it is capable of dealing with entire pages instead of just individual staves. We investigate page retrieval when querying with full pages (e.g., searching for copies of a piece) or just with snippets (searching for pages using individual staves).

### 4.1 Page Queries

Musical manuscripts were often manually copied; in large collections and across collections, there may be duplicates of the same music that are accidentally kept as a separate composition. One might want to discover such copies automatically. This is the first retrieval task we simulate.

In principle, this task is easy once OMR systems achieve results somewhat above a random baseline. The collection is quite small – 350 pages in total. In the MIDI representation, pages are long sequences in a very sparse space, so any minimally robust similarity score should yield good results.<sup>5</sup> Page retrieval is therefore a natural starting point for demonstrating whether OMR systems are useful for anything at all: if an OMR system fails on this task, it can hardly be expected to be useful anywhere else. So far, we are not aware of any OMR system that can handle handwritten music scores even with remote success.

<sup>3</sup>Implemented in <https://github.com/hajicj/muscima>.  
<sup>4</sup>Theoretically. Only MIDI export is currently implemented.  
<sup>5</sup>One does not even necessarily have to use the musical content of the scores to match them, given a smart enough algorithm dealing with the score images. However, “smart enough” may be daunting, as one would have to contend with different handwriting styles, segmentations of scores into staves and pages, etc.

<sup>3</sup>Implemented in <https://github.com/hajicj/muscima>.

<sup>4</sup>Theoretically. Only MIDI export is currently implemented.



	MAP@1	MAP@10	MAP@49
Page queries, OMR2OMR	1.0	1.0	0.998
Page queries, cross-modal	1.0	1.0	0.998
Snippet queries, OMR2OMR	0.928	0.834	0.763
Snippet queries, cross-modal	0.606	0.610	0.577

**Table 1: Results for page retrieval using page queries and snippet queries under two modalities: using OMR for creating the database and the query (OMR2OMR) or just for the database (cross-modal) and query with ground-truth MIDI.**

## 4.2 Snippet Queries

One might want to search not only using entire pieces, but also with shorter segments. We imagine musicologists, e.g., tracing the genealogy of a musical thought throughout a substantial body of work, or looking for musical citations across a geographic area. Here, the query is much shorter, and therefore OMR mistakes matter proportionally more.

## 4.3 Evaluation and Results

Both tasks are evaluated in two modalities: when the database and the query are created using the same OMR system (OMR2OMR), and when only the database is created by the OMR system and the queries are taken from the ground truth MIDI (cross-modal: simulating searching a sheet music database with, e.g., a keyboard capture sequence). If both the query and the database are processed with the same OMR system, some of the system's limitations may cancel out (e.g., ignoring key signatures), whereas when querying a sheet music database with MIDI from a different source, these limitations come to light.

The retrieval results are shown in Table 1. The FP-OMR system can deal with manuscripts of CVC-MUSCIMA well enough to retrieve copies of the same score reliably. When snippets are used as queries, the applicability of the system would depend on the specific scenario; the results in Table 1, row 3 indicate that the OMR system will be better suited in situations that require precision rather than recall. In the cross-modal setting, the simplifications made by [16] render the system practically useless at the granularity of individual staves.

## 5 SYMBOLIC MUSIC SIMILARITY

Besides content-based retrieval, one may have various other reasons to compute similarity over symbolic representations of music [13, 14, 17, 21]. As we cannot realistically evaluate OMR systems in all these settings, we can instead try to measure how the errors made by OMR systems influence the behavior of the standard symbolic similarity metrics.



**Figure 1: Sample from PrIMuS dataset, synthetically distorted to resemble non-ideal sheet conditions.**

Query	Spearman		Pearson	
	$M_1$	$M_2$	$M_1$	$M_2$
OMR2OMR	94.0	96.9	96.4	97.0
cross-modal	93.8	97.1	97.0	97.1

**Table 2: Average Spearman's and Pearson's correlation coefficients (in %) for the similarity between the original MIDI file and the MIDI file generated by the OMR system.  $M_1$  and  $M_2$  refers to *ShapeH* and *Time* symbolic similarity functions, respectively, from Urbano's *MelodyShape* library.**

We use Urbano's *MelodyShape* library<sup>6</sup> as the battery of standard metrics, available for measuring symbolic music similarity. Specifically, we consider *ShapeH* ( $M_1$ ) and *Time* ( $M_2$ ) similarity functions [24], as these ranked top in previous editions of the MIREX Symbolic Melodic Similarity challenge.<sup>7</sup>

The data used for this experiment corresponds to a subset of the PrIMuS dataset [4], which contains synthetically rendered scores of real music incipits from the RISM database. An incipit is the opening sequence of a song and can be used for the identification of a musical work. Therefore, they represent suitable musical elements for showcasing OMR-based search. We specifically consider the partition of images that have been distorted to resemble difficult conditions that might appear in some real cases [3]. An example from this collection is shown in Fig. 1.

The experiment considers the similarity between an incipit that acts as a query, and each sample of two sets of 1500 incipits: the real (ground truth) MIDI files and the MIDI files generated by the E2E-OMR system. For evaluation, Spearman's and Pearson's correlation coefficients are computed between the similarities obtained from the same query in both datasets. While Spearman's coefficient measures only whether the relationship is monotonous, Pearson's coefficient also measures if the relationship is linearly correlated. The higher these correlation coefficients are, the more smoothly an OMR system can substitute human input, to provide MIDI in applications where the given similarity function is used. A total of 1000 such queries were made, and the averaged coefficients are reported. In addition, we study the same two modalities as before: in the first one, the query is the MIDI output of the OMR system that read the image (OMR2OMR); in the other one, the query is taken from the ground-truth MIDI representation (cross-modal).

The results of this experiment are provided in Table 2. In most cases, the correlation coefficients are higher than 95 %. Reflecting the observations in [24], OMR errors perturb  $M_1$  more with respect to the rank-aware Spearman's correlation. Considering the high figures of the Pearson's coefficient, reorderings caused by OMR mistakes are likely to occur for samples that are very similar anyway. With respect to the  $M_2$  metric, fewer reorderings are observed, while rank-unaware correlations remain the same.

<sup>6</sup><https://github.com/julian-urbano/MelodyShape>

<sup>7</sup>See [http://www.music-ir.org/mirex/wiki/2015:Symbolic\\_Melodic\\_Similarity\\_Results](http://www.music-ir.org/mirex/wiki/2015:Symbolic_Melodic_Similarity_Results) for further details.

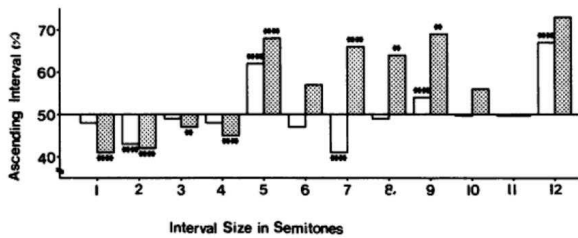


Figure 2: Original figure from [25] summarizing their quantitative results. Note that this figure compares data for two stylistically different datasets: western music (white), and folk tunes (gray). The authors were looking for how the difference between ascending/descending interval distributions could be used to distinguish melodies originating from the music of the respective styles.

## 6 CASE STUDY FOR DIGITAL MUSICOLOGY

So far, we have shown the extent to which current state-of-the-art OMR can enhance a digital library’s indexing and search. In this section, we illustrate how OMR systems can be useful to musicologists when working with such an enriched collection.

As a model for a musicological investigation that could plausibly benefit from OMR, we use the work of Vos and Troost [25]. Its authors propose characterizing the Western classical music genre based on the joint distribution of interval sizes and direction (ascending vs. descending), and compare these against a corpus of non-artificial music: both quantitatively and in a perceptual experiment.<sup>8</sup> We re-trace the quantitative portion of their work, showing that in this data aggregation scenario, the OMR systems would lead the researcher to propose the same hypothesis while obviating the need for manual data entry and checking. The errors that OMR introduces are offset by the vastly greater scale at which data aggregation is enabled, compared to manual data entry.

The quantitative findings of [25] are summarized in Fig. 2 and reproduced in Fig. 3. We compute the same distribution from the PrIMuS set of incipits [4]. This dataset is stylistically the same as the Dictionary of Musical Themes (DMT), although it is not limited to themes. The ascending/descending interval distributions are shown in Fig. 3. We found that for no interval size the balance between its ascending and descending instances was significantly different between the ground truth MIDI and OMR outputs (two-tailed binomial test at levels 0.05 and 0.01, following [25]).

Comparing the figures 2 and 3, one could discover the same trends. There are meaningful differences for the fifth and the octave, which may be because the DMT only contains prominent melodies, while PrIMuS data contains all incipits, including those from middle voices. However, our point is rather that one can see the same trend both in the ground truth MIDI and OMR outputs, indicating that the manual labor of data acquisition in [25] can be avoided using OMR without substantially putting the conclusions into question.

<sup>8</sup>This study has been cited over 180 times and is used, e.g., to illustrate the functionality of the MIDIToolbox software [23].

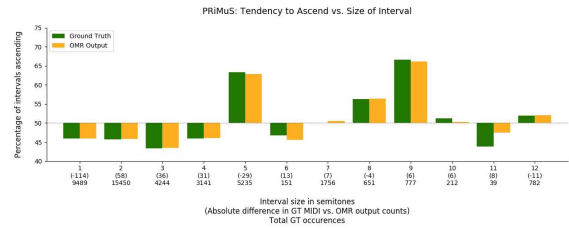


Figure 3: Ascending vs. descending tendency by interval size in semitones, comparing ground truth MIDI and OMR outputs on the monophonic PrIMuS dataset. The dataset is monostylistic; colors differentiate the method of MIDI acquisition. We observe comparable tendencies to the Composer data (white bars in Fig. 2).

## 7 CONCLUSIONS

We have attempted to show on several scenarios that recent advances in OMR state of the art have, to an extent, made OMR a more relevant technology. We believe these advances, especially given the underlying generic machine learning methodology, have implications for designing and enriching digital collections of sheet music. Being aware of these advances can be valuable for various stakeholders such as librarians and musicologists.

The showcased methods still have inherent limitations. Chiefly, learning does not transfer easily between datasets. The currently best-performing methods require re-training for each archive, even though the notation style may be the same. This implies that for every use-case, manual annotations will be necessary, and it is difficult to estimate in advance how much data will be enough. Furthermore, the systems are still not accurate enough to provide functionality such as playback or structured encoding. Beyond sufficient accuracy, further concerns also remain before “full-text” search in music can be done at a truly massive scale – efficient representations of music notation and its indexing, multimedia linking (such as lyrics alignment), and user interface design.

Overall, we conclude that the current state of the art in OMR enables (1) adding content-based similarity and retrieval functionality to music score image databases, especially for use-cases that do not require fine granularity, (2) applications based on symbolic melodic similarity, (3) research in digital musicology that builds on aggregating massive amounts of data and quantitative conclusions. The experiments in this paper should be considered as supporting evidence for these conclusions. We hope that the interested reader will find the reported capabilities of state-of-the-art OMR worth considering.

## ACKNOWLEDGMENTS

Jan Hajić jr. acknowledges support by the Czech Science Foundation grant no. P103/12/G084, Charles University Grant Agency grants 1444217 and 170217, and by SVV project 260 453; Marta Kolárová is supported by Charles University Grant Agency grant 1444217.

## REFERENCES

- [1] Andrew Hankinson, John Ashley Burgoyne, Gabriel Vigliensoni, Alastair Porter, Jessica Thompson, Wendy Liu, Remi Chiu, and Ichiro Fujinaga. 2012. Digital Document Image Retrieval Using Optical Music Recognition. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Fabien Gouyon, Perfecto Herrera, Luis Gustavo Martins, and Meinard Müller (Eds.). 577–582.
- [2] Stefan Balke, Sanu Pulimootil Achankunju, and Meinard Müller. 2015. Matching Musical Themes based on noisy OCR and OMR input. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*: 2015-August (2015), 703–707. DOI : <http://dx.doi.org/10.1109/ICASSP.2015.7178060>
- [3] Jorge Calvo-Zaragoza and David Rizo. 2018. Camera-PrIMuS: Neural End-to-End Optical Music Recognition on Realistic Monophonic Scores. In *19th International Society for Music Information Retrieval Conference*. (in press).
- [4] Jorge Calvo-Zaragoza and David Rizo. 2018. End-to-End Neural Optical Music Recognition of Monophonic Scores. *Applied Sciences* 4 (2018). DOI : <http://dx.doi.org/10.3390/app8040606>
- [5] Liang Chen and Kun Duan. 2016. MIDI-assisted egocentric optical music recognition. In 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016* (2016). DOI : <http://dx.doi.org/10.1109/WACV.2016.7477714> cited By 0; Conference of IEEE Winter Conference on Applications of Computer Vision, WACV 2016 ; Conference Date: 7 March 2016 Through 10 March 2016; Conference Code:121834.
- [6] G. Sayeed Choudhury, M. Droetboom, Tim DiLauro, Ichiro Fujinaga, and Brian Harrington. 2000. Optical Music Recognition System within a Large-Scale Digitization Project. In *1st International Symposium on Music Information Retrieval*. 1–6.
- [7] Jürgen Diet and Frank Kurth. 2007. The Probado Music Repository at the Bavarian State Library.. In *ISMIR*. 501–504.
- [8] Matthew J. Dovey. 2004. Overview of the OMRAS project: Online music retrieval and searching. *Journal of the American Society for Information Science and Technology* 55, 12 (2004), 1100–1107. DOI : <http://dx.doi.org/10.1002/asi.20063>
- [9] Alicia Fornés, Anjan Dutta, Albert Gordo, and Josep Lladós. 2012. CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal. *International Journal on Document Analysis and Recognition (IJ DAR)* 15, 3 (2012), 243–251. DOI : <http://dx.doi.org/10.1007/s10032-011-0168-2>
- [10] C. Fremerey, D. Damm, F. Kurth, and M. Clausen. 2009. Handling Scanned Sheet Music and Audio Recordings in Digital Music Libraries. In *Proceedings of the International Conference on Acoustics NAG/DAGA*. 1–2.
- [11] Christian Fremerey, Meinard Müller, Frank Kurth, and Michael Clausen. 2008. Automatic mapping of scanned sheet music to audio recordings. In *Proceedings of the 9th International Society for Music Information Retrieval Conference (ISMIR 2008)*. 413–418. [http://ismir2008.ismir.net/papers/ISMIR2008\\_116.pdf](http://ismir2008.ismir.net/papers/ISMIR2008_116.pdf)
- [12] Ichiro Fujinaga, Andrew Hankinson, and Julie E. Cumming. 2014. Introduction to SIMSSA (Single Interface for Music Score Searching and Analysis). In *Proceedings of the 1st International Workshop on Digital Libraries for Musicology*. ACM, 1–3. DOI : <http://dx.doi.org/10.1145/2660168.2660184>
- [13] David Garfinkle, Claire Arthur, Peter Schubert, Julie Cumming, and Ichiro Fujinaga. 2017. PatternFinder: Content-Based Music Retrieval with Music21. In *Proceedings of the 4th International Workshop on Digital Libraries for Musicology (DLfM '17)*. ACM, New York, NY, USA, 5–8. DOI : <http://dx.doi.org/10.1145/3144749.3144751>
- [14] Joe George and Lior Shamir. 2014. Computer analysis of similarities between albums in popular music. *Pattern Recognition Letters* 45 (2014), 78–84.
- [15] Jan jr. Hajič and Pavel Pecina. 2017. The MUSCIMA++ Dataset for Handwritten Optical Music Recognition. *Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition* (2017).
- [16] Jan Hajič jr., Matthias Dorfer, Gerhard Widmer, and Pavel Pecina. 2018. Towards Full-Pipeline Handwritten OMR with Musical Symbol Detection by U-Nets. In *19th International Society for Music Information Retrieval Conference*. (in press).
- [17] Alan Marsden. 2012. Interrogating Melodic Similarity: A Definitive Phenomenon or the Product of Interpretation? *Journal of New Music Research* 41, 4 (2012), 323–335.
- [18] Matthias Dorfer, Andreas Arzt, and Gerhard Widmer. 2017. Learning Audio-Sheet Music Correspondences for Score Identification and Offline Alignment. In *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, Sally Jo Cunningham, Zhiyao Duan, Xiao Hu, and Douglas Turnbull (Eds.). 115–122. [https://ismir2017.smcnus.org/wp-content/uploads/2017/10/32\\_Paper.pdf](https://ismir2017.smcnus.org/wp-content/uploads/2017/10/32_Paper.pdf)
- [19] Alexander Pacha and Horst Eidenberger. 2017. Towards Self-Learning Optical Music Recognition. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 795–800. DOI : <http://dx.doi.org/10.1109/ICMLA.2017.00-60>
- [20] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R.S. Marcal, Carlos Guedes, and Jaime S. Cardoso. 2012. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval* 1, 3 (March 2012), 173–190. DOI : <http://dx.doi.org/10.1007/s13735-012-0004-6>
- [21] David Rizo. 2010. *Symbolic music comparison with tree data structures*. Ph.D. Dissertation. Universidad de Alicante.
- [22] Jessica Thompson, Andrew Hankinson, and Ichiro Fujinaga. 2011. Searching the Liber Usualis: Using CouchDB and ElasticSearch to Query Graphical Music Documents. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*. <http://ismir2011.ismir.net/latebreaking/LB-10.pdf>
- [23] P. Toiviainen and T. Eerola. 2016. MIDI toolbox 1.1. <https://github.com/miditoolbox/>. (2016).
- [24] Julián Urbano. 2013. *MIREX 2013 Symbolic Melodic Similarity: A Geometric Model supported with Hybrid Sequence Alignment*. Technical Report. Music Information Retrieval Evaluation eXchange.
- [25] Piet G. Vos and Jim M. Troost. 1989. Ascending and Descending Melodic Intervals: Statistical Findings and their Perceptual Relevance. *Music Perception* 6, 4 (1989), 383–396.

## 7.6 Handwritten Optical Music Recognition: A Working Prototype

Jan Hajič jr. and Matthias Dorfer. Handwritten Optical Music Recognition: a Working Prototype. *Extended Abstracts for the Late-Breaking Demo Session of the 18th International Society for Music Information Retrieval Conference*. Suzhou, China, 2017.

**Handwritten Optical Music Recognition: A Working Prototype.** This short paper accompanied a demo in which the OMR pipeline was integrated into the MUS-CIMarker software. The paper underlies thesis contribution (M4).

The contribution of the dissertation author is about 60% of the article if one takes into account that the co-author provided the trained models for object detection; if one takes merely the software demo aspect of the article into account, the contribution of the dissertation author rises to about 90%.



# PROTOTYPING FULL-PIPELINE OPTICAL MUSIC RECOGNITION WITH MUSCIMARKER

**Jan Hajič jr.**

Charles University  
Institute of Formal and Applied Linguistics  
hajicj@ufal.mff.cuni.cz

**Matthias Dorfer**

Johannes Kepler Universität  
Institute of Computational Perception  
matthias.dorfer@jku.at

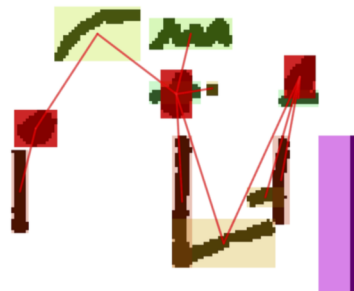
## ABSTRACT

We present MUSCIMarker, an open-source workbench for developing Optical Music Recognition (OMR) systems from image preprocessing to MIDI export. It is built around the notation graph data model of the MUSCIMA++ dataset for full-pipeline OMR. The system is transparent and interactive, enabling the user to visualize, validate and edit results of individual OMR stages. It is platform-independent, written purely in Python, and can work offline. We demonstrate its value with a prototype OMR system for musical manuscripts that implements the recognition pipeline, up to playing the recognition outputs through MIDI. The audience will interact with the program and can test an integrated OMR system prototype.

## 1. INTRODUCING MUSCIMARKER

The task of recovering symbolic musical information such as MIDI from the image of the written score has been addressed by the field of Optical Music Recognition (OMR) for half a century. Much more music has been written than has been recorded, which makes OMR important to making a large portion of musical heritage accessible to both the professional and the general public. However, OMR has not yet been able to deliver satisfactory solutions for anything but high-quality scans of printed music – much less for manuscripts, even though more pieces probably remain in handwritten form than have been typeset, both from long-dead and contemporary composers. We can therefore expect the development of OMR systems to continue for some time. OMR systems also need diagnosing with respect to the final output, as well as in the individual stages of the typical recognition pipeline (see [1]). At the same time, although one dataset is now available that provides ground truth sufficient for experiments on the full OMR pipeline (MUSCIMA++ [5]), the variability of sheet music means further data acquisition is still necessary.

To address these needs, we present MUSCIMarker: a workbench for OMR groundtruthing and prototyping.



**Figure 1.** Notation graph: notation symbols are vertices; edges encode which symbols have to be interpreted in relationship to each other. Staff symbols have been removed for clarity.

### 1.1 Data model

OMR aims to recover the “musical content”, which usually means the piano roll representation. In terms of notes, this can be thought of in terms of recovering *notes in time*: their pitches, durations, and onsets. These properties can be inferred deterministically from the *notation graph* of the musical score [5]. The vertices of this graph are the individual symbols (with properties such as symbol class and position on the page), the edges are the logical relationships between symbols. Each encoded note is represented by a notehead-class vertex, and its pitch, duration and onset can be decoded from the relationships of the notehead to other symbols that affect these parameters, such as stems, stafflines, or accidentals. A simplified example of a notation graph is in Figure 1.

Because the data model for music scores is an open problem, MUSCIMarker also allows customizing it.

### 1.2 MUSCIMarker Features

The core functionality of MUSCIMarker is manipulating the notation graph for a given musical score. Advances in OMR can then be integrated into MUSCIMarker as “automated helpers”.

Its closest “cousin” is probably the Aletheia system for OCR groundtruthing [4]. However, music notation needs a richer data model than the document layout and glyph model that Aletheia provides. The related gamera<sup>1</sup> OMR



<sup>1</sup><http://gamera.informatik.hsnr.de>

toolkit does not at all provide the interactive functionality necessary for groundtruthing and correcting errors of automated subsystems integrated into the tool.

MUSCIMarker implements the following feature sets:

- Efficient tools for creating and editing the notation graph, incl. interactive binarization,
- User activity tracking, both for optimizing “bottle-necks” in usability and for assessing experimental subsystems in terms of error correction times,
- Automated notation graph validation.

To obtain the piano roll representation, MUSCIMarker then implements pitch/duration/onset inference for the particular definition of the notation graph used in MUSCIMA++, and can play back MIDI for the user.

On top of this, we have trained prototype OMR subsystems that perform symbol detection and notation graph reconstruction from symbols, so that in principle a full OMR pipeline is available (even though these prototypes’ performance leaves something to be desired).

All functionality is available for the audience to try.

## 2. PROTOTYPE OMR SYSTEM OVERVIEW

Recent progress in image processing methods using convolutional networks and the release of the new MUSCIMA++ dataset [5] have together made building a viable OMR solution for handwritten musical scores a more realistic proposition, although progress has so far been mainly focused on staffline removal [2, 3]. We include a prototype of an OMR solution integrated into MUSCIMarker.

We build the notation graph in two steps: finding the vertices of the notation graph, which means detecting notation symbols, and recovering their relationships, the edges of the graph.

**Symbol detection** has models trained for symbols that participate in pitches, durations, and onsets. For each symbol, the detector is a U-net (see Figure 2) [6]; each class is trained independently. Staffline detection is handled in exactly the same manner as other symbols, and staffline removal is not necessary for detecting other classes of symbols. The FCN outputs a probability mask; we threshold this mask at 0.3 and then detect symbols using connected components. The detector needs a GPU to achieve reasonable speed, and thus runs remotely.

**Relationships** between symbols are then inferred. There are two types of relationships: *attachment* and *precedence*. Attachments are syntactic relationships between primitives, such as that between a notehead and a stem or staffline, which are necessary for obtaining pitch and duration; precedence edges are used to compute onsets. A manual parser is provided that simply adds all edges among a manually selected subset of vertices that are possible based on the symbols’ classes. For attachments, a probabilistic parser is also trained, so that one can work with larger selections; the probabilistic parser is a binary classifier that classifies each (ordered) pair of symbols according to the difference between their bounding boxes and the symbol classes.

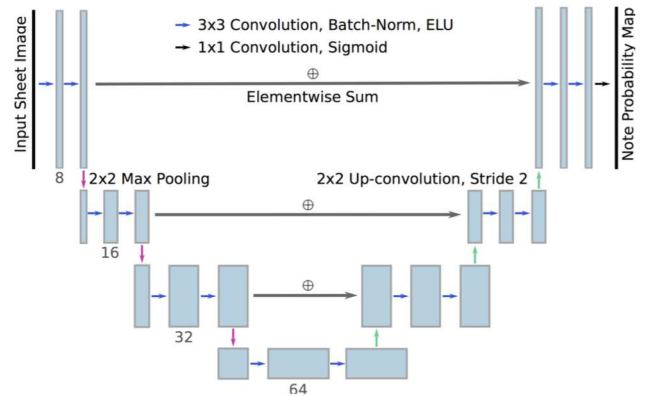


Figure 2. Detector network architecture.

## 3. CLOSING REMARKS

We hope that the presented MUSCIMarker tool for OMR prototyping and groundtruthing will be valuable to OMR researchers, and by implication to future OMR users.<sup>2</sup> We look forward to audience feedback, to guide further development so that we come closer to bringing musical manuscripts into the digital fold.

## 4. REFERENCES

- [1] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R. S. Marcal, Carlos Guedes, and Jaime S. Cardoso. Optical Music Recognition: State-of-the-Art and Open Issues. *Int J Multimed Info Retr*, 1(3):173–190, Mar 2012.
- [2] Antonio-Javier Gallego and Jorge Calvo Zaragoza. Staff-line removal with selectional auto-encoders. *Expert Systems with Applications*, 89:138 – 148, 2017.
- [3] Jorge Calvo Zaragoza, Antonio Pertusa, and Jose Oncina. Staff-line detection and removal using a convolutional neural network. *Machine Vision and Applications*, pages 1–10, 2017.
- [4] Christian Clausner, Stefan Pletschacher, and Apostolos Antonacopoulos. Aletheia - An Advanced Document Layout and Text Ground-Truthing System for Production Environments. In *2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18-21, 2011*, pages 48–52. IEEE Computer Society, 2011.
- [5] Jan Hajič, jr. and Pavel Pecina. In Search of a Dataset for Handwritten Optical Music Recognition: Introducing MUSCIMA++. *ArXiv e-prints*, March 2017.
- [6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III*, pages 234–241, Cham, 2015. Springer International Publishing.

<sup>2</sup> <https://github.com/hajicj/MUSCIMarker>



# Bibliography

- Alfio Andronico and Alberto Ciampa. On Automatic Pattern Recognition and Acquisition of Printed Music. In *International Computer Music Conference*, Venice, Italy, 1982. Michigan Publishing. URL <http://hdl.handle.net/2027/spo.bbp2372.1982.024>.
- Jamie Anstice, Tim Bell, Andy Cockburn, and Martin Setchell. The design of a pen-based musical input system. In *6th Australian Conference on Computer-Human Interaction*, pages 260–267, 1996. doi: 10.1109/OZCHI.1996.560019.
- David Bainbridge. A complete optical music recognition system: Looking to the future. Technical report, University of Canterbury, 1994. URL <https://ir.canterbury.ac.nz/handle/10092/14874>.
- David Bainbridge. *Extensible optical music recognition*. PhD thesis, University of Canterbury, 1997. URL <http://hdl.handle.net/10092/9420>.
- David Bainbridge and Tim Bell. Dealing with superimposed objects in optical music recognition. In *6th International Conference on Image Processing and its Applications*, number 443, pages 756–760, 1997. ISBN 0 85296 692 X. doi: 10.1049/cp:19970997.
- David Bainbridge and Tim Bell. The Challenge of Optical Music Recognition. *Computers and the Humanities*, 35(2):95–121, 2001. ISSN 1572-8412. doi: 10.1023/A:1002485918032.
- David Bainbridge and Tim Bell. A music notation construction engine for optical music recognition. *Software: Practice and Experience*, 33(2):173–200, 2003. ISSN 1097-024X. doi: 10.1002/spe.502.
- David Bainbridge and Nicholas Paul Carter. Automatic reading of music notation. In H. Bunke and P. Wang, editors, *Handbook of Character Recognition and Document Image Analysis*, pages 583–603. World Scientific, Singapore, 1997. doi: 10.1142/9789812830968\_0022.
- Stefan Balke, Sanu Pulimootil Achankunju, and Meinard Müller. Matching Musical Themes based on noisy OCR and OMR input. In *International Conference on Acoustics, Speech and Signal Processing*, pages 703–707. Institute of Electrical and Electronics Engineers Inc., 2015. ISBN 9781467369978. doi: 10.1109/ICASSP.2015.7178060.
- Baoguang Shi, Xiang Bai, and Cong Yao. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. *CoRR*, abs/1507.05717, 2015. URL <http://arxiv.org/abs/1507.05717>.

- Stephan Baumann. A Simplified Attributed Graph Grammar for High-Level Music Recognition. In *3rd International Conference on Document Analysis and Recognition*, pages 1080–1083. IEEE, 1995. ISBN 0-8186-7128-9. doi: 10.1109/ICDAR.1995.602096.
- Pierfrancesco Bellini, Ivan Bruno, and Paolo Nesi. Optical music sheet segmentation. In *1st International Conference on WEB Delivering of Music*, pages 183–190. Institute of Electrical & Electronics Engineers (IEEE), 2001. ISBN 0769512844. doi: 10.1109/wdm.2001.990175.
- Pierfrancesco Bellini, Ivan Bruno, and Paolo Nesi. Assessing Optical Music Recognition Tools. *Computer Music Journal*, 31(1):68–93, 2007. doi: 10.1162/comj.2007.31.1.68.
- Hervé Bitteur. Audiveris, 2004. URL <https://github.com/audiveris>.
- Dorothea Blostein and Henry S. Baird. *A Critical Survey of Music Image Analysis*, pages 405–434. Springer Berlin Heidelberg, 1992. ISBN 978-3-642-77281-8. doi: 10.1007/978-3-642-77281-8\_19.
- Ondřej Bojar, Miloš Ercegovčević, Martin Popel, and Omar F. Zaidan. A Grain of Salt for the WMT Manual Evaluation. *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 1–11, 2011. URL <http://dl.acm.org/citation.cfm?id=2132960.2132962>.
- Gregory Burlet, Alastair Porter, Andrew Hankinson, and Ichiro Fujinaga. Neon.js: Neume Editor Online. In *13th International Society for Music Information Retrieval Conference*, pages 121–126, Porto, Portugal, 2012. URL [http://ismir2012.ismir.net/event/papers/121\\_ISMIR\\_2012.pdf](http://ismir2012.ismir.net/event/papers/121_ISMIR_2012.pdf).
- Donald Byrd and Jakob Grue Simonsen. Towards a Standard Testbed for Optical Music Recognition: Definitions, Metrics, and Page Images. *Journal of New Music Research*, 44(3):169–195, 2015. ISSN 0929-8215. doi: 10.1080/09298215.2015.1045424.
- Chris Callison Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar F. Zaidan. Findings of the 2010 Joint Workshop on Statistical Machine Translation and Metrics for Machine Translation. *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53, 2010. URL <http://dl.acm.org/citation.cfm?id=1868850.1868853>.
- Jorge Calvo Zaragoza. *Pattern Recognition for Music Notation*. PhD thesis, 2016.
- Jorge Calvo Zaragoza and Jose Oncina. Recognition of pen-based music notation with finite-state machines. *Expert Systems with Applications*, 72:395–406, 2017. ISSN 0957-4174. doi: 10.1016/j.eswa.2016.10.041.



- Jorge Calvo Zaragoza and Jose Oncina. Recognition of Pen-Based Music Notation: The HOMUS Dataset. In *22nd International Conference on Pattern Recognition*, pages 3038–3043. Institute of Electrical & Electronics Engineers (IEEE), 2014. doi: 10.1109/ICPR.2014.524.
- Jorge Calvo Zaragoza and Jose Oncina. Clustering of strokes from pen-based music notation: An experimental study. *Lecture Notes in Computer Science*, 9117:633–640, 2015. ISSN 0302-9743. doi: 10.1007/978-3-319-19390-8\_71.
- Jorge Calvo Zaragoza and David Rizo. End-to-End Neural Optical Music Recognition of Monophonic Scores. *Applied Sciences*, (4), 2018. ISSN 2076-3417. doi: 10.3390/app8040606. URL <http://www.mdpi.com/2076-3417/8/4/606>.
- Jorge Calvo Zaragoza, David Rizo, and José Manuel Iñesta. Two (note) heads are better than one: pen-based multimodal interaction with music scores. In J. et al. Devaney, editor, *17th International Society for Music Information Retrieval Conference*, pages 509–514, New York City, 2016a. ISBN 978-0-692-75506-8. URL [https://wp.nyu.edu/ismir2016/wp-content/uploads/sites/2294/2016/07/006\\_Paper.pdf](https://wp.nyu.edu/ismir2016/wp-content/uploads/sites/2294/2016/07/006_Paper.pdf).
- Jorge Calvo Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. Document Analysis for Music Scores via Machine Learning. In *3rd International workshop on Digital Libraries for Musicology*, pages 37–40, New York, USA, 2016b. ACM, ACM. ISBN 978-1-4503-4751-8. doi: 10.1145/2970044.2970047.
- Jorge Calvo Zaragoza, Antonio Pertusa, and Jose Oncina. Staff-line detection and removal using a convolutional neural network. *Machine Vision and Applications*, pages 1–10, 2017a. ISSN 1432-1769. doi: 10.1007/s00138-017-0844-4.
- Jorge Calvo Zaragoza, Jose J. Valero Mas, and Antonio Pertusa. End-to-end Optical Music Recognition using Neural Networks. In *18th International Society for Music Information Retrieval Conference*, Suzhou, China, 2017b. ISBN 978-981-11-5179-8. URL [https://ismir2017.smcnus.org/wp-content/uploads/2017/10/34\\_Paper.pdf](https://ismir2017.smcnus.org/wp-content/uploads/2017/10/34_Paper.pdf).
- Jorge Calvo Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. One-step detection of background, staff lines, and symbols in medieval music manuscripts with convolutional neural networks. In *18th International Society for Music Information Retrieval Conference*, Suzhou, China, 2017c. ISBN 978-981-11-5179-8. URL [https://ismir2017.smcnus.org/wp-content/uploads/2017/10/162\\_Paper.pdf](https://ismir2017.smcnus.org/wp-content/uploads/2017/10/162_Paper.pdf).
- Jorge Calvo Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. Staff-Line Detection on Grayscale Images with Pixel Classification. In Luís A. Alexandre, José Salvador Sánchez, and João M. F. Rodrigues, editors, *Pattern Recognition and Image Analysis*, pages 279–286, Cham, 2017d. Springer International Publishing.

ISBN 978-3-319-58838-4. URL [https://link.springer.com/chapter/10.1007%2F978-3-319-58838-4\\_31](https://link.springer.com/chapter/10.1007%2F978-3-319-58838-4_31).

Jorge Calvo Zaragoza, Jan Hajič jr., and Alexander Pacha. Discussion Group Summary: Optical Music Recognition. In Alicia Fornés and Lamiroy Bart, editors, *Graphics Recognition, Current Trends and Evolutions*, Lecture Notes in Computer Science, pages 152–157. Springer International Publishing, 2018. ISBN 978-3-030-02283-9. doi: 10.1007/978-3-030-02284-6\_12.

Jamie dos Santos Cardoso, Artur Capela, Ana Rebelo, Carlos Guedes, and Joaquim Pinto da Costa. Staff Detection with Stable Paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1134–1139, 2009. ISSN 0162-8828. doi: 10.1109/TPAMI.2009.34.

Sukalpa Chanda, Debleena Das, Umapada Pal, and Fumitaka Kimura. Offline Hand-Written Musical Symbol Recognition. *14th International Conference on Frontiers in Handwriting Recognition*, pages 405–410, 2014. doi: 10.1109/ICFHR.2014.74. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6981053>.

Charles Spearman. The proof and measurement of association between two things. volume 15, page 72–101, 1904.

Liang Chen and Kun Duan. MIDI-assisted egocentric optical music recognition. In *Winter Conference on Applications of Computer Vision*. Institute of Electrical and Electronics Engineers Inc., 2016. ISBN 9781509006410. doi: 10.1109/WACV.2016.7477714.

Liang Chen, Rong Jin, and Christopher Raphael. Renotation from Optical Music Recognition. In *Mathematics and Computation in Music*, pages 16–26, Cham, 2015a. Springer International Publishing. doi: 10.1007/978-3-319-20603-5\_2.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015b.

Maura Church and Michael Scott Cuthbert. Improving Rhythmic Transcriptions via Probability Models Applied Post-OMR. In Hsin-Min Wang, Yi-Hsuan Yang, and Jin Ha Lee, editors, *15th International Society for Music Information Retrieval Conference*, pages 643–648, 2014. URL [http://www.terasoft.com.tw/conf/ismir2014/proceedings/T116\\_357\\_Paper.pdf](http://www.terasoft.com.tw/conf/ismir2014/proceedings/T116_357_Paper.pdf).

Alastair T. Clarke, B. Malcom Brown, and M. P. Thorne. Coping with some really rotten problems in automatic music recognition. *Microprocessing and Microprogramming*, 27(1):547–550, 1989. ISSN 0165-6074. doi: 10.1016/0165-6074(89)90108-7. URL <http://www.sciencedirect.com/science/article/>

- [pii/0165607489901087](#). Fifteenth EUROMICRO Symposium on Microprocessing and Microprogramming.
- Bertrand Couasnon and Jean Camillerapp. Using Grammars To Segment and Recognize Music Scores. In *International Association for Pattern Recognition Workshop on Document Analysis Systems*, pages 15–27, Kaiserslautern, Germany, 1994. URL <ftp://ftp.idsa.prd.fr/local/IMADOC/couasnon/Articles/das94.ps>.
- Bertrand Couasnon and Bernard Rétif. Using a grammar for a reliable full score recognition system. In *International Computer Music Conference*, pages 187–194, 1995. URL <https://pdfs.semanticscholar.org/3b97/949f436f929ed11ee76358e07fa1a61d2e01.pdf>.
- Christoph Dalitz and Thomas Karsten. Using the Gamera framework for building a lute tablature recognition system. In *6th International Conference on Music Information Retrieval*, pages 478–481, London, UK, 2005. URL <http://ismir2005.ismir.net/proceedings/2012.pdf>.
- Christoph Dalitz, Michael Droettboom, Bastian Pranzas, and Ichiro Fujinaga. A Comparative Study of Staff Removal Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):753–766, 2008a. ISSN 0162-8828. doi: 10.1109/tpami.2007.70749.
- Christoph Dalitz, Georgios K. Michalakis, and Christine Pranzas. Optical recognition of psaltic Byzantine chant notation. *International Journal of Document Analysis and Recognition*, 11(3):143–158, 2008b. ISSN 1433-2825. doi: 10.1007/s10032-008-0074-4. URL <https://doi.org/10.1007/s10032-008-0074-4>.
- David Damm, Christian Fremerey, Frank Kurth, Meinard Müller, and Michael Clausen. Multimodal Presentation and Browsing of Music. In *10th International Conference on Multimodal Interfaces*, pages 205–208, Chania, Greece, 2008. ACM. ISBN 978-1-60558-198-9. doi: 10.1145/1452392.1452436. URL <http://doi.acm.org/10.1145/1452392.1452436>.
- Matthias Dorfer, Andreas Arzt, and Gerhard Widmer. Towards End-to-End Audio-Sheet-Music Retrieval. *Computing Research Repository*, abs/1612.05070, 2016a. URL <http://arxiv.org/abs/1612.05070>.
- Matthias Dorfer, Andreas Arzt, and Gerhard Widmer. Towards Score Following In Sheet Music Images. In Michael I. Mandel, Johanna Devaney, Douglas Turnbull, and George Tzanetakis, editors, *17th International Society for Music Information Retrieval Conference*, pages 789–795, 2016b. ISBN 978-0-692-75506-8. URL [https://wp.nyu.edu/ismir2016/wp-content/uploads/sites/2294/2016/07/027\\_Paper.pdf](https://wp.nyu.edu/ismir2016/wp-content/uploads/sites/2294/2016/07/027_Paper.pdf).

Matthias Dorfer, Jan Hajič jr., and Gerhard Widmer. On the Potential of Fully Convolutional Neural Networks for Musical Symbol Detection. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 02, pages 53–54, New York, USA, Nov 2017. IAPR TC10 (Technical Committee on Graphics Recognition), IEEE Computer Society. ISBN 978-1-5386-3586-5. doi: 10.1109/ICDAR.2017.274.

Matthias Dorfer, Jan Hajič jr., Andreas Arzt, Harald Frostel, and Gerhard Widmer. Learning Audio–Sheet Music Correspondences for Cross-Modal Retrieval and Piece Identification. *Transactions of the International Society for Music Information Retrieval*, 1(1):22–33, 2018a. doi: 10.5334/tismir.12.

Matthias Dorfer, Florian Henkel, and Gerhard Widmer. Learning To Listen, Read And Follow: Score Following As A Reinforcement Learning Game. In *19th International Society for Music Information Retrieval Conference*, pages 784–791, Paris, France, 2018b. ISBN 978-2-9540351-2-3. URL [http://ismir2018.ircam.fr/doc/pdfs/45\\_Paper.pdf](http://ismir2018.ircam.fr/doc/pdfs/45_Paper.pdf).

Michael Droettboom and Ichiro Fujinaga. Symbol-level groundtruthing environment for OMR. In *5th International Conference on Music Information Retrieval*, pages 497–500, 2004. URL <http://ismir2004.ismir.net/proceedings/p090-page-497-paper117.pdf>.

Michael Droettboom, Ichiro Fujinaga, Karl MacMillan, G. Sayeed Chouhury, Tim DiLauro, Mark Patton, and Teal Anderson. Using the Gamera framework for the recognition of cultural heritage materials. In *Joint Conference on Digital Libraries*, pages 12–17, London, UK, 2002. URL <http://droettboom.com/papers/p74-droettboom.pdf>.

Hoda M. Fahmy and Dorothea Blostein. Graph Grammar Processing of Uncertain Data. In *Advances in Structural and Syntactic Pattern Recognition*, pages 373–382. World Scientific, 1993. doi: 10.1142/9789812797919\_0031.

Hoda M. Fahmy and Dorothea Blostein. A graph-rewriting paradigm for discrete relaxation: Application to sheet-music recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 12(6):763–799, 1998. doi: 10.1142/S0218001498000439.

Alicia Fornés. Analysis of Old Handwritten Musical Scores. Master’s thesis, Universitat Autònoma de Barcelona, 2005. URL [http://www.cvc.uab.es/~afornes/publi/AFornes\\_Master.pdf](http://www.cvc.uab.es/~afornes/publi/AFornes_Master.pdf).

Alicia Fornés. *Writer Identification by a Combination of Graphical Features in the Framework of Old Handwritten Music Scores*. PhD thesis, Universitat Autònoma de Barcelona, 2009. URL <http://www.cvc.uab.es/~afornes/publi/PhDAliciaFornes.pdf>.



- Alicia Fornés, Josep Lladós, and Gemma Sánchez. Primitive Segmentation in Old Handwritten Music Scores. In Wenying Liu and Josep Lladós, editors, *Graphics Recognition. Ten Years Review and Future Perspectives*, pages 279–290, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-34712-5. doi: 10.1007/11767978\_25.
- Alicia Fornés, Anjan Dutta, Albert Gordo, and Josep Lladós. The ICDAR 2011 Music Scores Competition: Staff Removal and Writer Identification. In *International Conference on Document Analysis and Recognition*, pages 1511–1515, 2011. doi: 10.1109/ICDAR.2011.300.
- Alicia Fornés, Anjan Dutta, Albert Gordo, and Josep Lladós. CVC-MUSCIMA: A Ground-truth of Handwritten Music Score Images for Writer Identification and Staff Removal. *International Journal on Document Analysis and Recognition*, 15(3): 243–251, 2012. ISSN 1433-2825. doi: 10.1007/s10032-011-0168-2.
- Alicia Fornés, Van Cuong Kieu, Muriel Visani, Nicholas Journet, and Anjan Dutta. The ICDAR/GREC 2013 Music Scores Competition: Staff Removal. In Bart Lamiroy and Jean-Marc Ogier, editors, *Graphics Recognition. Current Trends and Challenges*, pages 207–220, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. ISBN 978-3-662-44854-0. URL [https://link.springer.com/chapter/10.1007/978-3-662-44854-0\\_16](https://link.springer.com/chapter/10.1007/978-3-662-44854-0_16).
- Christian Fremerey, David Damm, Frank Kurth, and Michael Clausen. Handling Scanned Sheet Music and Audio Recordings in Digital Music Libraries. In *International Conference on Acoustics NAG/DAGA*, pages 1–2, 2009. URL [https://www.audiolabs-erlangen.de/content/05-fau/professor/00-mueller/03-publications/2009\\_FremereyDaMuKuCl\\_ScanAudio\\_DAGA.pdf](https://www.audiolabs-erlangen.de/content/05-fau/professor/00-mueller/03-publications/2009_FremereyDaMuKuCl_ScanAudio_DAGA.pdf).
- Ichiro Fujinaga. Optical Music Recognition using Projections. Master’s thesis, McGill University, 1988. URL [https://www.researchgate.net/profile/Ichiro\\_Fujinaga/publication/38435306\\_Optical\\_music\\_recognition\\_using\\_projections/links/546ca7980cf24b753c628c6e.pdf](https://www.researchgate.net/profile/Ichiro_Fujinaga/publication/38435306_Optical_music_recognition_using_projections/links/546ca7980cf24b753c628c6e.pdf).
- Ichiro Fujinaga. Exemplar-based learning in adaptive optical music recognition system. In *International Computer Music Conference*, pages 55–56, Hong Kong, 1996. ISBN 962-85092-1-7. URL <http://hdl.handle.net/2027/spo.bbp2372.1996.015>.
- Ichiro Fujinaga. Optical Music Recognition Bibliography. Website, 2000. URL <http://www.music.mcgill.ca/~ich/research/omr/omrbib.html>.
- Ichiro Fujinaga, Andrew Hankinson, and Julie E. Cumming. Introduction to SIMSSA (Single Interface for Music Score Searching and Analysis). In *1st International Workshop on Digital Libraries for Musicology*, pages 1–3. ACM, 2014. doi: 10.1145/2660168.2660184.

- Antonio-Javier Gallego and Jorge Calvo Zaragoza. Staff-line removal with selectional auto-encoders. *Expert Systems with Applications*, 89:138–148, 2017. ISSN 0957-4174. doi: 10.1016/j.eswa.2017.07.002. URL <http://www.sciencedirect.com/science/article/pii/S0957417417304712>.
- Jan Hajič jr. A Case for Intrinsic Evaluation of Optical Music Recognition. In Jorge Calvo-Zaragoza, Jan Hajič jr., and Alexander Pacha, editors, *1st International Workshop on Reading Music Systems*, pages 15–16, Paris, France, 2018. URL <https://sites.google.com/view/worms2018/proceedings>.
- Jan Hajič jr. and Matthias Dorfer. Handwritten Optical Music Recognition: a Working Prototype. In *Extended abstracts for the Late-Breaking Demo Session of the 18th International Society for Music Information Retrieval Conference*, Suzhou, China, 2017. URL <https://ismir2017.smcnus.org/lbds/Hajic2017.pdf>.
- Jan Hajič jr. and Pavel Pecina. Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression. *Computing Research Repository*, abs/1708.01806, 2017a. URL <http://arxiv.org/abs/1708.01806>.
- Jan Hajič jr. and Pavel Pecina. Groundtruthing (Not Only) Music Notation with MUSICMarker: A Practical Overview. In *14th International Conference on Document Analysis and Recognition*, pages 47–48, Kyoto, Japan, 2017b. doi: 10.1109/ICDAR.2017.271.
- Jan Hajič jr. and Pavel Pecina. The MUSCIMA++ Dataset for Handwritten Optical Music Recognition. In *14th International Conference on Document Analysis and Recognition*, pages 39–46, Kyoto, Japan, 2017c. doi: 10.1109/ICDAR.2017.16.
- Jan Hajič jr., Jiří Novotný, Pavel Pecina, and Jaroslav Pokorný. Further Steps towards a Standard Testbed for Optical Music Recognition. In Michael Mandel, Johanna Devaney, Douglas Turnbull, and George Tzanetakis, editors, *17th International Society for Music Information Retrieval Conference*, pages 157–163, New York, USA, 2016. New York University, New York University. ISBN 978-0-692-75506-8. URL <https://wp.nyu.edu/ismir2016/event/proceedings/>.
- Jan Hajič jr., Matthias Dorfer, Gerhard Widmer, and Pavel Pecina. Towards Full-Pipeline Handwritten OMR with Musical Symbol Detection by U-Nets. In *19th International Society for Music Information Retrieval Conference*, pages 225–232, Paris, France, 2018a. ISBN 978-2-9540351-2-3. URL [http://ismir2018.ircam.fr/doc/pdfs/175\\_Paper.pdf](http://ismir2018.ircam.fr/doc/pdfs/175_Paper.pdf).
- Jan Hajič jr., Marta Kolárová, Alexander Pacha, and Jorge Calvo Zaragoza. How Current Optical Music Recognition Systems Are Becoming Useful for Digital Libraries. In *5th International Conference on Digital Libraries for Musicology*, pages 57–61, Paris, France, 2018b. ACM. ISBN 978-1-4503-6522-2. doi: 10.1145/3273024.3273034. URL <http://doi.acm.org/10.1145/3273024.3273034>.

- Andrew Hankinson. *Optical music recognition infrastructure for large-scale music document analysis*. PhD thesis, McGill University, 2014. URL <http://digitool.library.mcgill.ca/webclient/DeliveryManager?pid=130291>.
- Andrew Hankinson, John Ashley Burgoyne, Gabriel Vigliensoni, Alastair Porter, Jessica Thompson, Wendy Liu, Remi Chiu, and Ichiro Fujinaga. Digital Document Image Retrieval Using Optical Music Recognition. In Fabien Gouyon, Perfecto Herrera, Luis Gustavo Martins, and Meinard Müller, editors, *13th International Society for Music Information Retrieval Conference*, pages 577–582, 2012. URL <http://ismir2012.ismir.net/event/papers/577-ismir-2012.pdf>.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- James W. Perry, Allen Kent, and Madeline M. Berry. Machine literature searching X. Machine language: factors underlying its design and development. *American Documentation*, 6(4):242–254, October 1955. doi: 10.1002/asi.5090060411. URL <https://doi.org/10.1002/asi.5090060411>.
- John Ashley, Burgoyne Laurent, Pugin Greg, and Eustace Ichiro Fujinaga. A Comparative Survey of Image Binarisation Algorithms for Optical Recognition on Degraded Musical Sources. *ACM SIGSOFT Software Engineering Notes*, 24(1985):1994–1997, 2008.
- M. G. Kendall. A NEW MEASURE OF RANK CORRELATION. *Biometrika*, 30(1-2): 81–93, 1938.
- Brenden M Lake, Ruslan R Salakhutdinov, and Josh Tenenbaum. One-shot learning by inverting a compositional causal process. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2526–2534. Curran Associates, Inc., 2013. URL <http://papers.nips.cc/paper/5128-one-shot-learning-by-inverting-a-compositional-causal-process.pdf>.
- Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436 EP –, 05 2015. URL <https://doi.org/10.1038/nature14539>.
- Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984. ISBN 0-534-98053-8.
- Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image Style Transfer Using Convolutional Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2414–2423. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.265. URL <https://doi.org/10.1109/CVPR.2016.265>.

- V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, February 1966.
- Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. *CoRR*, abs/1708.02002, 2017. URL <http://arxiv.org/abs/1708.02002>.
- Nailja Luth. Automatic Identification of Music Notations. In *2nd International Conference on WEB Delivering of Music*, 2002. ISBN 0769518621. doi: 10.1109/WDM.2002.1176212.
- Karl MacMillan, Michael Droettboom, and Ichiro Fujinaga. Gamera: A structured document recognition application development environment. In *2nd International Symposium on Music Information Retrieval*, pages 15–16, Bloomington, IN, 2001. URL <https://jscholarship.library.jhu.edu/handle/1774.2/44376>.
- Karl MacMillan, Michael Droettboom, and Ichiro Fujinaga. Gamera: Optical music recognition in a new shell. In *International Computer Music Conference*, pages 482–485, 2002. URL <http://www.music.mcgill.ca/~ich/research/icmc02/icmc2002.gamera.pdf>.
- Matouš Macháček and Ondřej Bojar. Results of the WMT14 Metrics Shared Task. *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 293–301, 2014.
- Youichi Mitobe, Hidetoshi Miyao, and Minoru Maruyama. A fast HMM algorithm based on stroke lengths for on-line recognition of handwritten music scores. In *9th International Workshop on Frontiers in Handwriting Recognition*, pages 521–526, 2004. doi: 10.1109/IWFHR.2004.2.
- Hidetoshi Miyao and Robert Martin Haralick. Format of ground truth data used in the evaluation of the results of an optical music recognition system. In *4th International Workshop on Document Analysis Systems*, pages 497–506, Brasil, 2000. URL [https://www.researchgate.net/profile/Robert\\_Haralick/publication/242138660\\_Format\\_of\\_Ground\\_Truth\\_Data\\_Used\\_in\\_the\\_Evaluation\\_of\\_the\\_Results\\_of\\_an\\_Optical\\_Music\\_Recognition\\_System/links/0046353bac1589cc3f000000.pdf](https://www.researchgate.net/profile/Robert_Haralick/publication/242138660_Format_of_Ground_Truth_Data_Used_in_the_Evaluation_of_the_Results_of_an_Optical_Music_Recognition_System/links/0046353bac1589cc3f000000.pdf).
- Hidetoshi Miyao and Minoru Maruyama. An online handwritten music score recognition system. In *17th International Conference on Pattern Recognition*. Institute of Electrical & Electronics Engineers (IEEE), 2004. doi: 10.1109/icpr.2004.1334164.
- Kia Ng, David Cooper, Ewan Stefani, Roger Boyle, and Nick Bailey. Embracing the Composer : Optical Recognition of Handwrtnen Manuscripts. In *International Computer Music Conference*, pages 500–503, 1999. URL <https://ci.nii.ac.jp/naid/10011612045/en/>.



- Jiri Novotný and Jaroslav Pokorný. Introduction to Optical Music Recognition: Overview and Practical Challenges. In Pokorný J. Necasky M., Moravec P., editor, *Annual International Workshop on DAtabases, TExtS, Specifications and Objects*, pages 65–76. CEUR-WS, 2015. URL <http://ceur-ws.org/Vol-1343/paper6.pdf>.
- Alexander Pacha and Jorge Calvo Zaragoza. Optical Music Recognition in Mensural Notation with Region-Based Convolutional Neural Networks. In *19th International Society for Music Information Retrieval Conference*, pages 240–247, Paris, France, 2018. ISBN 978-2-9540351-2-3. URL [http://ismir2018.ircam.fr/doc/pdfs/32\\_Paper.pdf](http://ismir2018.ircam.fr/doc/pdfs/32_Paper.pdf).
- Alexander Pacha, Kwon-Young Choi, Bertrand Coüasnon, Yann Ricquebourg, Richard Zanibbi, and Horst Eidenberger. Handwritten Music Object Detection: Open Issues and Baseline Results. In *13th International Workshop on Document Analysis Systems*, pages 163–168, 2018a. doi: 10.1109/DAS.2018.51.
- Alexander Pacha, Jan Hajič jr., and Jorge Calvo Zaragoza. A Baseline for General Music Object Detection with Deep Learning. *Applied Sciences*, 8(9):1488–1508, 2018b. ISSN 2076-3417. doi: 10.3390/app8091488. URL <http://www.mdpi.com/2076-3417/8/9/1488>.
- Victor Padilla, Alan Marsden, Alex McLean, and Kia Ng. Improving OMR for Digital Music Libraries with Multiple Recognisers and Multiple Sources. In *1st International Workshop on Digital Libraries for Musicology*, pages 1–8, London, United Kingdom, 2014. ACM. ISBN 978-1-4503-3002-2. doi: 10.1145/2660168.2660175.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, 2002. doi: 10.3115/1073083.1073135. URL <http://dx.doi.org/10.3115/1073083.1073135>.
- K. Pearson. Mathematical Contributions to the Theory of Evolution. III. Regression, Heredity, and Panmixia. *Philosophical Transactions of the Royal Society of London Series A*, 187:253–318, 1896. doi: 10.1098/rsta.1896.0007.
- Viet-Khoi Pham, Hai-Dang Nguyen, and Minh-Triet Tran. Virtual Music Teacher for New Music Learners with Optical Music Recognition. In *International Conference on Learning and Collaboration Technologies*, pages 415–426. Springer, 2015. doi: 10.1007/978-3-319-20609-7\_39.
- Telmo Pinto, Ana Rebelo, Gilson Giraldo, and Jamie dos Santos Cardoso. Content Aware Music Score Binarization. Technical report, Universidade do Porto, Portugal, 2010. URL <http://www.inescporto.pt/~jsc/publications/conferences/2010TPintoACCV.pdf>.

- Telmo Pinto, Ana Rebelo, Gilson Giraldi, and Jamie dos Santos Cardoso. Music Score Binarization Based on Domain Knowledge. In Jordi Vitrià, João Miguel Sanches, and Mario Hernández, editors, *Pattern Recognition and Image Analysis*, pages 700–708. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-21257-4. doi: 10.1007/978-3-642-21257-4.87.
- David S. Prerau. Computer pattern recognition of printed music. In *Fall Joint Computer Conference*, pages 153–162, 1971.
- Denis Pruslin. *Automatic recognition of sheet music*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1966.
- Laurent Pugin. Optical Music Recognition of Early Typographic Prints using Hidden Markov Models. In *7th International Conference on Music Information Retrieval*, pages 53–56, Victoria, Canada, 2006a. URL [http://ismir2006.ismir.net/PAPERS/ISMIR06152\\_Paper.pdf](http://ismir2006.ismir.net/PAPERS/ISMIR06152_Paper.pdf).
- Laurent Pugin. Aruspix: an Automatic Source-Comparison System. *Computing in Musicology*, 14:49–59, 2006b. ISSN 1057-9478. URL <https://dialnet.unirioja.es/servlet/articulo?codigo=3476563>.
- Laurent Pugin, Jason Hockman, John Ashley Burgoyne, and Ichiro Fujinaga. Gamera versus Aruspix – Two Optical Music Recognition Approaches. In *9th International Conference on Music Information Retrieval*, 2008. URL [http://ismir2008.ismir.net/papers/ISMIR2008\\_247.pdf](http://ismir2008.ismir.net/papers/ISMIR2008_247.pdf).
- Christopher Raphael and Jingya Wang. New Approaches to Optical Music Recognition. In Anssi Klapuri and Colby Leider, editors, *12th International Society for Music Information Retrieval Conference*, pages 305–310, Miami, Florida, 2011. University of Miami. URL <http://ismir2011.ismir.net/papers/OS3-3.pdf>.
- Ana Rebelo. *Robust Optical Recognition of Handwritten Musical Scores based on Domain Knowledge*. PhD thesis, University of Porto, 2012. URL <http://www.inescporto.pt/~arebelo/arebeloThesis.pdf>.
- Ana Rebelo and Jamie dos Santos Cardoso. Staff Line Detection and Removal in the Grayscale Domain. In *12th International Conference on Document Analysis and Recognition*, pages 57–61, 2013. doi: 10.1109/ICDAR.2013.20.
- Ana Rebelo, G. Capela, and Jamie dos Santos Cardoso. Optical recognition of music symbols. *International Journal on Document Analysis and Recognition*, 13(1):19–31, 2010. ISSN 1433-2825. doi: 10.1007/s10032-009-0100-1.
- Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R.S. Marcal, Carlos Guedes, and Jamie dos Santos Cardoso. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 1(3):173–190, 2012. doi: 10.1007/s13735-012-0004-6.

- Ana Rebelo, André Marçal, and Jamie dos Santos Cardoso. Global constraints for syntactic consistency in OMR: an ongoing approach. In *International Conference on Image Analysis and Recognition*, 2013. URL <http://www.inescporto.pt/~jsc/publications/conferences/2013ARebeloICIAIAR.pdf>
- Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. *arXiv*, 2018.
- K. Todd Reed and J. R. Parker. Automatic Computer Recognition of Printed Music. In *13th International Conference on Pattern Recognition*, pages 803–807, 1996. ISBN 081867282X. doi: 10.1109/ICPR.1996.547279.
- Dan Ringwalt, Roger Dannenberg, and Andrew Russell. Optical Music Recognition for Interactive Score Display. In Edgar Berdahl and Jesse T. Allison, editors, *International Conference on New Interfaces for Musical Expression*, pages 95–98, Baton Rouge, Louisiana, USA, 2015. The School of Music and the Center for Computation and Technology (CCT), Louisiana State University. ISBN 978-0-692-49547-6. URL <http://dl.acm.org/citation.cfm?id=2993778.2993805>.
- David Rizo, Jorge Calvo Zaragoza, and José M. Iñesta. MuRET: A Music Recognition, Encoding, and Transcription Tool. In *5th International Conference on Digital Libraries for Musicology*, pages 52–56, Paris, France, 2018. ACM. ISBN 978-1-4503-6522-2. doi: 10.1145/3273024.3273029. URL <http://doi.acm.org/10.1145/3273024.3273029>.
- JW W Roach and J E Tatem. Using domain knowledge in low-level visual processing to interpret handwritten music: an experiment. *Pattern Recognition*, 21(1):33–44, 1988. ISSN 0031-3203. doi: 10.1016/0031-3203(88)90069-6. URL <http://www.sciencedirect.com/science/article/pii/0031320388900696>.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4. doi: 10.1007/978-3-319-24574-4\_28. URL [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28).
- Florence Rossant and Isabelle Bloch. Robust and Adaptive OMR System Including Fuzzy Modeling, Fusion of Musical Rules, and Possible Error Detection. *EURASIP Journal on Advances in Signal Processing*, 2007(1):081541, 2006. ISSN 1687-6180. doi: 10.1155/2007/81541.
- Zeyad Saleh, Ke Zhang, Jorge Calvo Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. Pixel.js: Web-Based Pixel Classification Correction Platform for Ground

- Truth Creation. In *14th International Conference on Document Analysis and Recognition*, pages 39–40, Kyoto, Japan, 2017. doi: 10.1109/ICDAR.2017.267.
- Craig Sapp. OMR Comparison of SmartScore and SharpEye, 2013. URL <https://ccrma.stanford.edu/~craig/mro-compare-beethoven>.
- Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.
- Véronique Sébastien, Henri Ralambondrainy, Olivier Sébastien, and Noël Conruyt. Score Analyzer: Automatically Determining Scores Difficulty Level for Instrumental e-Learning. In Fabien Gouyon, Perfecto Herrera, Luis Gustavo Martins, and Meinard Müller, editors, *13th International Society for Music Information Retrieval Conference*, pages 571–576, Porto, Portugal, 2012. URL <http://ismir2012.ismir.net/event/papers/571-ismir-2012.pdf>.
- Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015.
- Scott Sheridan and Susan E. George. Defacing Music Scores for Improved Recognition. In *2nd Australian Undergraduate Students' Computing Conference*, pages 142–148, 2004. URL <https://sites.google.com/site/theauscc/auscc04/papers/sheridan-auscc04.pdf>.
- Baoguang Shi, Xiang Bai, and Cong Yao. An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304, 2017. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2646371.
- Javier Sober Mira, Jorge Calvo Zaragoza, David Rizo, and José Manuel Iñesta. Pen-Based Music Document Transcription. In *14th International Conference on Document Analysis and Recognition*, pages 21–22, Kyoto, Japan, 2017. IEEE. doi: 10.1109/ICDAR.2017.258.
- Mariusz Szwoch. Guido: A Musical Score Recognition System. In *9th International Conference on Document Analysis and Recognition*, pages 809–813, 2007. doi: 10.1109/ICDAR.2007.4377027.
- Mariusz Szwoch. Using MusicXML to Evaluate Accuracy of OMR Systems. In *International Conference on Theory and Application of Diagrams*, pages 419–422, Herrsching, Germany, 2008. Springer, Springer-Verlag. ISBN 978-3-540-87729-5. doi: 10.1007/978-3-540-87730-1\_53.



- Jessica Thompson, Andrew Hankinson, and Ichiro Fujinaga. Searching the Liber Usualis: Using CouchDB and ElasticSearch to Query Graphical Music Documents. In *12th International Society for Music Information Retrieval Conference*, 2011. URL <http://ismir2011.ismir.net/latebreaking/LB-10.pdf>.
- Theophanis Tsandilas. Interpreting Strokes on Paper with a Mobile Assistant. In *25th Annual ACM Symposium on User Interface Software and Technology*, pages 299–308, Cambridge, Massachusetts, USA, 2012. ACM. ISBN 978-1-4503-1580-7. doi: 10.1145/2380116.2380155.
- Lukas Tuggener, Isamil Elezi, Jürgen Schmidhuber, Marcello Pelillo, and Stadelmann Thilo. DeepScores - A Dataset for Segmentation, Detection and Classification of Tiny Objects. In *24th International Conference on Pattern Recognition*, Beijing, China, 2018. doi: 10.21256/zhaw-4255. URL <https://arxiv.org/abs/1804.00525>.
- Eelco van der Wel and Karen Ullrich. Optical Music Recognition with Convolutional Sequence-to-Sequence Models. In *18th International Society for Music Information Retrieval Conference*, Suzhou, China, 2017. ISBN 978-981-11-5179-8. URL <https://arxiv.org/abs/1707.04877>.
- Cuihong Wen, Jing Zhang, Ana Rebelo, and Fanyong Cheng. A Directed Acyclic Graph-Large Margin Distribution Machine Model for Music Symbol Classification. *PLoS ONE*, 11(3):1–11, 2016. doi: 10.1371/journal.pone.0149688.
- JaeMyeong Yoo, Nguyen Dinh Toan, DeokJai Choi, HyukRo Park, and Gueesang Lee. Advanced Binarization Method for Music Score Recognition Using Local Thresholds. In *8th International Conference on Computer and Information Technology Workshops*, pages 417–420, 2008. doi: 10.1109/CIT.2008.Workshops.101.
- Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6):1245–1262, 1989.

# List of Figures

1.1	An example of a musical manuscript: a copy of G. B. Pergolesi's Stabat Mater, part X: Fac, ut portem Christi mortem.	4
1.2	The two goals of OMR explained in terms of the process of writing music.	6
2.1	The values of pitch, illustrated on a piano keyboard. One octave period is indicated.	10
2.2	The types of notes according to duration.	12
2.3	Example page of music notation.	14
2.4	The elements encoding pitch	14
2.5	The elements encoding duration.	17
2.6	A somewhat complex beamed group	18
2.7	A more complex beamed group situation in a 17th century violin manuscript (H. I. F. Biber, Mystery sonata IV).	18
2.8	Examples of notation where precedence is complicated.	19
3.1	OMR for replayability and reprintability.	21
3.2	The basic ways of characterizing OMR inputs.	22
3.3	The presence of multiple voices (indicated with red lines) adds complications.	24
3.4	Long-distance relationships affecting pitch of the note on the right.	24
3.5	The variety of handwriting	25
3.6	The standard OMR pipeline up to staff removal.	28
4.1	Visualizing the detected symbols and the assembled notation graph on top of staff removal output	41
4.2	Interface of the MUSCIMarker tool	43
4.3	The design of the bounding box-based detector	46
4.4	An example results of the RCNN-based detector	46
4.5	The U-Net architecture	47
4.6	Example results in a complex notational situation	50
4.7	Convex hull trick for training U-Nets for complex symbols	51
4.8	The detection f-score for symbols required for replayability	52
4.9	Pitch recognition f-score per staff	55
4.10	Collecting a data point for the omreval corpus	58

# List of Tables

4.1	Comparison of generic object detection methods on OMR datasets.	48
4.2	The results for page retrieval using page queries and snippet queries under two modalities . . . . .	56
4.3	Measures of agreement for some proposed evaluation metrics against the omreval corpus . . . . .	59

# List of Abbreviations

CWMN: Common Western Music Notation

OMR: Optical Music Recognition

MIDI: Musical Instrument Digital Interface

MuNG: Music Notation Graph

MUSCIMA: Music Score Images

DTW: Dynamic Time Warping

R-CNN: Region-Proposal Convolutional Neural Network

TED: Tree Edit Distance



# List of publications

## 7.7 Publications used in this thesis

Jan Hajič jr., Jiří Novotný, Pavel Pecina and Jaroslav Pokorný: Further Steps towards a Standard Testbed for Optical Music Recognition. *Proceedings of the 17th International Society for Music Information Retrieval Conference*, pages 157–163, New York, USA, 2016. ISBN 978-0-692-75506-8.

*Cited by (excluding self-citations, according to Google Scholar): 4*

Jan Hajič jr. and Pavel Pecina. Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression. *CoRR*, 2017. arXiv:1708.01806

*Cited by (excluding self-citations, according to Google Scholar): 4*

Jan Hajič jr. and Pavel Pecina. The MUSCIMA++ Dataset for Handwritten Optical Music Recognition. *14th International Conference on Document Analysis and Recognition*, pages 39–46, Kyoto, Japan, 2017. ISBN 978-1-5386-3586-5, ISSN 2379-2140, doi: 10.1109/ICDAR.2017.16.

*Cited by (excluding self-citations, according to Google Scholar): 8*

Jan Hajič jr. and Pavel Pecina. Groundtruthing (not only) Music Notation with MUSCIMarker: a Practical Overview. *14th IAPR International Conference on Document Analysis and Recognition / GREC*, Kyoto, Japan, pages 47–48, 2017. ISBN 978-1-5386-3586-5, doi: 10.1109/ICDAR.2017.271

Matthias Dorfer, Jan Hajič jr. and Gerhard Widmer. On the Potential of Fully Convolutional Neural Networks for Musical Symbol Detection. *14th International Conference on Document Analysis and Recognition / GREC*, Kyoto, Japan, pp. 53–54, 2017. ISBN 978-1-5386-3586-5, doi: 10.1109/ICDAR.2017.274.

*Cited by (excluding self-citations, according to Google Scholar): 1*

Jan Hajič jr. and Matthias Dorfer. Handwritten Optical Music Recognition: a Working Prototype. *Extended Abstracts for the Late-Breaking Demo Session of the 18th International Society for Music Information Retrieval Conference*. Suzhou, China, 2017.

*Cited by (excluding self-citations, according to Google Scholar): 3*

Alexander Pacha, Jan Hajič jr. and Jorge Calvo-Zaragoza. A Baseline for General Music Object Detection with Deep Learning. *Applied Sciences*, Vol. 8, No. 9, pages 1488–1488, Basel, Switzerland, 2018. ISSN 2076-3417.

*Cited by (excluding self-citations, according to Google Scholar): 1*

Jan Hajič jr.: A Case for Intrinsic Evaluation of Optical Music Recognition. *Proceed-*

*ings of the 1st International Workshop on Reading Music Systems*, Paris, France, pp. 15–16, 2018.

Jan Hajič jr., Matthias Dorfer, Gerhard Widmer, Pavel Pecina. Towards Full-Pipeline Handwritten OMR with Musical Symbol Detection by U-Nets. *Proceedings of the 19th Conference of the International Society for Music Information Retrieval*, pages 225–232, Paris, France, 2018. ISBN 978-2-9540351-2-3.

*Cited by (excluding self-citations, according to Google Scholar): 2*

Jan Hajič jr., Marta Kolárová, Alexander Pacha, Jorge Calvo-Zaragoza. How current optical music recognition systems are becoming useful for digital libraries. *Proceedings of the 5th International Conference on Digital Libraries for Musicology*, pages 57–61, Paris, France, 2018. ISBN 978-1-4503-6522-2.

Jorge Calvo-Zaragoza, Jan Hajič jr. and Alexander Pacha. Understanding Optical Music Recognition. *Manuscript under review*.

## 7.8 Other publications

Kateřina Veselovská, Jan Hajič jr. and Jana Šindlerová. Creating annotated resources for polarity classification in Czech. *Empirical Methods in Natural Language Processing – Proceedings of the Conference on Natural Language Processing 2012*, Wien, Austria, ISBN 3-85027-005-X, pp. 296-304, 2012

*Cited by (excluding self-citations, according to Google Scholar): 17*

Kateřina Veselovská and Jan Hajič jr.: Why Words Alone Are Not Enough: Error Analysis of Lexicon-based Polarity Classifier for Czech. *Proceedings of the 6th International Joint Conference on Natural Language Processing*, Copyright © Asian Federation of Natural Language Processing, Nagoya, Japan, ISBN 978-4-9907348-0-0, pp. 1-5, 2013

*Cited by (excluding self-citations, according to Google Scholar): 4*

Kateřina Veselovská and Jan Hajič jr. Developing Sentiment Annotator in UIMA – the Unstructured Management Architecture for Data Mining Applications. *ITAT 2013: Information Technologies - Applications and Theory (Workshops, Posters, and Tutorials)*, Donovaly, Slovakia, pp. 5–10, 2013. ISBN 978-1490952086

Kateřina Veselovská, Jan Hajič jr. and Jana Šindlerová. Subjectivity Lexicon for Czech: Implementation and Improvements. *Journal for Language Technology and Computational Linguistics*, Vol. 29, No. 1, Copyright © German Society for Computational Linguistics and Language Technology, Berlin, ISSN 2190-6858, pp. 47-61, Aug 2014

*Cited by (excluding self-citations, according to Google Scholar): 4*

Kateřina Veselovská, Jan Hajič jr. and Jana Šindlerová. Tracing Sentiments: Syntactic and Semantic Features in a Subjectivity Lexicon. *Proceedings of the XVI EURALEX International Congress: The User in Focus*, Copyright © EURAC research, Bolzano/Bozen, Italy, ISBN 978-88-88906-97-3, pp. 405-414, 2014

*Cited by (excluding self-citations, according to Google Scholar): 3*

Milan Straka, Jan Hajič, Jana Straková and Jan Hajič jr. Parsing Universal Dependency Treebanks using Neural Networks and Search-Based Oracle. *14th International Workshop on Treebanks and Linguistic Theories (TLT 2015)*, Copyright © IPIPAN, Warszawa, Poland, ISBN 978-83-63159-18-4, pp. 208-220, 2015

*Cited by (excluding self-citations, according to Google Scholar): 21*

Jan Hajič jr. and Pavel Pecina. Matching Illustrative Images to “Soft News” Articles. *UFAL WDS 2015 (Conference of PhD Students in Mathematical Linguistics)*, Copyright © Institute of Formal and Applied Linguistics, Charles University in Prague, Praha, Czechia, pp. 49-56, 2015

Silvie Cinková, Jana Straková, Jakub Hajič, Jan Hajič, Jan Hajič jr., Jolana Janoušková, Milan Straka, Miroslava Urešová: WordSim353-cs: Evaluation Dataset for Lexical Similarity and Relatedness based on WordSim353. *Data/software*, Charles University, Prague, Czech Republic, 2016. <http://hdl.handle.net/11234/1-1713>

Jan Hajič, jr. and Pavel Pecina. How to Exploit Music Notation Syntax for OMR? *14th International Conference on Document Analysis and Recognition / GREC*, pages 55–56, Kyoto, Japan, 2017. ISBN 978-1-5386-3586-5, doi: 10.1109/ICDAR.2017.275.

Matthias Dorfer, Jan Hajič jr. and Gerhard Widmer. Attention as a Perspective for Learning Tempo-invariant Audio Queries. *Proceedings of the 2018 Joint Workshop on Machine Learning for Music*, Stockholm, Sweden, Jul 2018.

Matthias Dorfer, Jan Hajič jr., Andreas Arzt, Harald Frostel and Gerhard Widmer. Learning Audio–Sheet Music Correspondences for Cross-Modal Retrieval and Piece Identification. *Transactions of the International Society for Music Information Retrieval*, Vol. 1, No. 1, pages 22–33, 2018. ISSN 2514-3298.

*Cited by (excluding self-citations, according to Google Scholar): 1*

Jorge Calvo-Zaragoza, Jan Hajič jr., Alexander Pacha. Discussion Group Summary: Optical Music Recognition. *Lecture Notes in Computer Science, Graphics Recognition. Current Trends and Evolutions. 12th IAPR International Workshop, GREC 2017, Kyoto, Japan, November 9-10, 2017, Revised Selected Papers*, Vol. 11009, No. 1, pages. 152-157, Basel, Switzerland, 2018. ISBN 978-3-030-02284-6, ISSN 0302-9743.