

Selbstlernende Optische Notenerkennung

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Alexander Pacha, B.Sc. M.Sc. with honours

Matrikelnummer 00828440

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Ao. Univ.-Prof. Mag. Dr. Horst Eidenberger

Diese Dissertation haben begutachtet:

Ichiro Fujinaga

Oge Marques

Wien, 18. Juni 2019

Alexander Pacha

Self-Learning Optical Music Recognition

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Alexander Pacha, B.Sc. M.Sc. with honours

Registration Number 00828440

to the Faculty of Informatics

at the TU Wien

Advisor: Ao. Univ.-Prof. Mag. Dr. Horst Eidenberger

The dissertation has been reviewed by:

Ichiro Fujinaga

Oge Marques

Vienna, 18th June, 2019

Alexander Pacha

Erklärung zur Verfassung der Arbeit

Alexander Pacha, B.Sc. M.Sc. with honours

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 18. Juni 2019

Alexander Pacha

Danksagung

Ich widme diese Arbeit meinem Vater—dem besten Vater den sich ein Kind wünschen konnte.

Ein riesengroßes Dankeschön an meinen Betreuer Horst Eidenberger, der mich exzellent betreut hat und mir - wann immer es nötig war - einen kleinen Schubs in die richtige Richtung gegeben hat. Weiters möchte ich meiner Mutter, meiner Schwester und meinen Freunden danken, insbesondere Daniela Stoll, Iris Stuhr, Peter Frühwirt, Markus Pöschel und Florian Ganglberger, die mich zu dieser Arbeit ermutigt haben und für mich da waren, wann immer ich sie gebraucht habe.

Ein besonderen Dank ergeht auch an meine Kollegen Jorge Calvo-Zaragoza und Jan Hajič jr. für die großartige Zusammenarbeit, ohne die diese Arbeit nicht möglich gewesen wäre. Zusätzlich danke ich auch Peter Kán, Iana Podkosova, und Khrystyna Vasylevska für die vielen Kleinigkeiten die mein Leben an der Universität bereichert haben.

Zuletzt möchte ich noch meinem Freund Friedrich Plank für das Korrekturlesen danken.

Acknowledgements

I dedicate this work to my father—the best father a child could hope for.

Many thanks to my supervisor Horst Eidenberger who supervised me exquisitely by providing me guidance whenever I needed it. I would also like to thank my mother, my sister and my friends, especially Daniela Stoll, Iris Stuhr, Peter Frühwirt, Markus Pöschel, and Florian Ganglberger who encouraged me to this work and always supported me.

A special thank-you goes to my colleagues Jorge Calvo-Zaragoza and Jan Hajič jr. for the fantastic collaboration. Without you, this work would not have been possible. I would also like to thank Peter Kán, Iana Podkosova, and Khrystyna Vasylevska for all the small things that made my day.

Finally, I would like to thank my dear friend Friedrich Plank for proofreading this thesis.

Kurzfassung

Musik ist ein essenzieller Teil unserer Kultur und unseres Erbes. Durch die Jahrhunderte wurden Millionen an Liedern komponiert und mittels Musiknotation auf Papier festgehalten. Die optische Notenerkennung (engl. Optical Music Recognition, kurz OMR) ist das Forschungsfeld, das untersucht, wie der Computer das Lesen von Musiknoten erlernen kann. Trotz jahrzehntelanger Forschung, gilt die optische Notenerkennung bis heute als alles andere als gelöst. Ein Grund hierfür ist die Tatsache, dass viele traditionelle Ansätze auf Heuristiken beruhen, die sich nur schwer verallgemeinern lassen. Deshalb schlage ich in dieser Arbeit einen anderen Weg vor, nämlich den Computer das Lesen von Musiknoten selbstständig erlernen zu lassen, mittels maschinellem Lernen, insbesondere Deep Learning.

In zahlreichen Experimenten konnte ich demonstrieren, dass der Computer unter Überwachung des Lernprozesses die meisten Herausforderungen der optischen Notenerkennung robust erlernen kann. Zu diesen Herausforderungen zählen die Analyse der Dokumentenstruktur, die Erkennung und Klassifikation von Symbolen, sowie die Konstruktion von einem Musiknotationsgraphen, der als zwischenzeitliche Repräsentation fungiert, die in ein passendes Format zur Weiterverarbeitung exportiert werden kann. Ein trainiertes neuronales Netzwerk kann zuverlässig vorhersagen, ob ein Bild Noten enthält oder nicht, während ein anderes imstande ist, den selben Takt in verschiedenen Ausgaben derselben Musik zu finden und miteinander zu verknüpfen, sodass man bequem zwischen diesen hin und her navigieren kann. Die Erkennung von Symbolen in gesetzten und handgeschriebenen Noten kann ebenfalls erlernt werden, sofern man ausreichend annotierte Daten zur Verfügung hat. Die Klassifikation der erkannten Symbole hat sogar eine niedrigere Fehlerrate als die von Menschen. Für Noten, die in Mensurnotation verfasst wurden, kann man die gesamte Erkennung in drei Schritte vereinfachen, wovon zwei mittels maschinellem Lernen gelöst werden können.

Neben dem Verfassen von wissenschaftlichen Artikeln, habe ich auch die größte Sammlung von Datensätzen für OMR zusammengetragen und dokumentiert, sowie die wahrscheinlich umfangreichste Bibliographie, die derzeit verfügbar ist. Beide Sammlungen sind online verfügbar. Desweiteren war ich an der Organisation des 1st International Workshop on Reading Music Systems beteiligt, habe gemeinsam mit Kollegen ein Tutorial bei der International Society For Music Information Retrieval Conference zum Thema optischer

Notenerkennung gegeben, und ein weiterer Workshop bei der Music Encoding Conference findet im Sommer 2019 statt.

Viele Herausforderungen der optischen Notenerkennung können mit Deep Learning effizient gelöst werden, wie die Analyse des Layouts oder die Erkennung von Musikobjekten. Allerdings ist die Musiknotation ein strukturelles Schreibsystem, bei dem die Beziehungen und das Zusammenspiel zwischen den einzelnen Objekten die Semantik bestimmen. Ein Musiknotationgraph ist eine geeignete Datenstruktur um diese Information abzubilden und erlaubt es klar zwischen zwei Dingen zu unterscheiden: der Rekonstruktion von Informationen aus dem Bild und der Kodierung der rekonstruierten Information in ein bestimmtes Format unter Berücksichtigung der Regeln der Musiknotation. So eine Konstruktion eines Musiknotationsgraphen kann zwar erlernt werden, bleiben einige Forschungsfragen offen. Ich bin zuversichtlich, dass das Trainieren des Computers auf einem hinreichend großen Datensatz unter menschlicher Überwachung einen nachhaltigen Ansatz darstellt, mit dem man in Zukunft viele Anwendungsfälle der optischen Notenerkennung lösen wird können.

Abstract

Music is an essential part of our culture and heritage. Throughout the centuries, millions of songs were composed and written down in documents using music notation. Optical Music Recognition (OMR) is the research field that investigates how the computer can learn to read those documents. Despite decades of research, OMR is still considered far from being solved. One reason is that traditional approaches rely heavily on heuristics and often do not generalize well. In this thesis, I propose a different approach to let the computer learn to read music notation documents mostly by itself using machine learning, especially deep learning.

In several experiments, I have demonstrated that the computer can learn to robustly solve many tasks involved in OMR by using supervised learning. These include the structural analysis of the document, the detection and classification of symbols in the scores as well as the construction of the music notation graph, which is an intermediate representation that can be exported into a format suitable for further processing. A trained deep convolutional neural network can reliably detect whether an image contains music or not, while another one is capable of finding and linking individual measures across multiple sources for easy navigation between them. Detecting symbols in typeset and handwritten scores can be learned, given a sufficient amount of annotated data, and classifying isolated symbols can be performed at even lower error rates than those of humans. For scores written in mensural notation the complete recognition can even be simplified into just three steps, two of which can be solved with machine learning.

Apart from publishing a number of scientific articles, I have gathered and documented the most extensive collection of datasets for OMR as well as the probably most comprehensive bibliography currently available. Both are available online. Moreover I was involved in the organization of the International Workshop on Reading Music Systems, in a joint tutorial at the International Society For Music Information Retrieval Conference on OMR as well as in another workshop at the Music Encoding Conference.

Many challenges of OMR can be solved efficiently with deep learning, such as the layout analysis or music object detection. As music notation is a configurational writing system where the relations and interplay between symbols determine the musical semantic, these relationships have to be recognized as well. A music notation graph is a suitable representation for storing this information. It allows to clearly distinguish between the challenges involved in recovering information from the music score image and the encoding

of the recovered information into a specific output format while complying with the rules of music notation. While the construction of such a graph can be learned as well, there are still many open issues that need future research. But I am confident that training the computer on a sufficiently large dataset under human supervision is a sustainable approach that will help to solve many applications of OMR in the future.

Contents

Kurzfassung	xi
Abstract	xiii
Contents	xv
1 Introduction	1
2 Understanding Optical Music Recognition	9
3 Towards Self-Learning Optical Music Recognition	61
4 Towards A Universal Music Symbol Classifier	69
5 Music Object Detection	73
5.1 Handwritten Music Object Detection	73
5.2 General Music Object Detection	81
6 Measure Detection and Structure Analysis	103
7 Music Notation Graph Construction	113
8 OMR for Mensural Notation	123
9 Other contributions	133
9.1 Optical Music Recognition Datasets project	133
9.2 ISMIR Tutorial “Optical Music Recognition for Dummies”	134
9.3 Workshop on Reading Music Systems (WoRMS)	135
9.4 Workshop at MEC 2019: Let’s Formalize Music Notation	135
9.5 Discussion Group Summary: Optical Music Recognition	135
9.6 Community Engagement and Website for OMR-Research	135
9.7 OMR Bibliography	136
10 Conclusions and Outlook	137

List of Figures	139
Bibliography	141

Introduction

“Music is the one incorporeal entrance into the higher world of knowledge which comprehends mankind but which mankind cannot comprehend.”

— Ludwig van Beethoven [Sul36]

Music, rhythm, and dance amounts to a universal language which is used and understood worldwide. It existed long before spoken languages emerged. It is used to convey information and emotions as well as to entertain us. Music manifests itself as sound pressure waves that travel through the air. They are a temporal phenomenon that only exists between the musician emitting it and the listener perceiving it. To preserve music it either has to be reproduced by a musician or recorded in one way or the other. Long before electricity was invented, people thought it worthwhile to preserve music in order to reproduce it. They invented a language called music notation, which is an abstraction that captures the essential bits of music. As with other languages, music notation evolved over the centuries and emerged in many different forms.

Millions of pieces have been composed and written down through the centuries, and this heritage still lives on and is actively extended by contemporary composers. It represents an essential part of our culture. Unfortunately, we are not born with the ability to read and understand music notation but acquire this skill by practicing it throughout our life. Starting to read music notation is very challenging and presents a large obstacle for beginners. Even experienced musicians are often surprised when they learn about yet another aspect of music notation. The reason why music notation is so hard to learn is its enormous complexity, imposed by the underlying information it tries to abstract and capture music, which is virtually without limits.

The arguably most prominent music notation is called Common Western Music Notation (CWMN) or Modern Staff Notation (see Fig. 1.1). It is a visual representation of the musical parameters: pitch, duration, velocity, and timbre. The sequence of notes and

rests are described by specific glyphs within a reference system of (typically) five parallel lines, called staff. The position on the y-axis represents the relative pitch while the x-axis depicts the temporal sequence. Additional symbols can contain instructions regarding velocity, timbre or the lyrics to be sung.

Figure 1.1: Excerpt from the waltz “An der schönen blauen Donau” by Johann Strauss, Jr.

By following these instructions, musicians can comprehend the original ideas of the composer, which enables them to reproduce the music—similar to books that can capture ideas, facts, moods and the likes for others to learn about. However, due to the complexity of the syntactic and semantic rules of CWMN, which requires years of practicing before it can be mastered, a large portion of the population cannot read it. One possibility of teaching them is by having a computer-assisted conversion of the written music scores into an audible version of the same piece. This process of reading music notation and automatically decoding it into a machine-readable format is the goal of Optical Music Recognition (OMR). More precisely:

“Optical Music Recognition is the research field that investigates how to computationally read music notation in documents.” [CZHjP19]

OMR has plenty of applications, including teaching students how to read music notation. It can also be used to digitize handwritten manuscripts for restoration and publication, support musicological examinations of large bodies of music, or enable practical applications such as providing accompanying voices while practicing a piece of music.

Follow me in this fictional story: imagine Lisa, a sixteen-year-old girl who loves music. Recently she discovered her passion for rock music. She loves it so much that she decided to pick up playing the guitar. She got a guitar from her parents for Christmas and took a few classes but quickly got bored by the music her teacher wanted her to play. She went on the internet and found a website that offers free scores of her favorite band in tabulature notation that she quickly understood. After all, tabulature notation can be much easier to read, since each line corresponds directly to one string on the guitar and the number indicates the fret of that string (see Fig. 1.2).

While playing the music that she enjoys so much, she keeps on practicing, and her skills improve considerably. One day, her favorite band releases a new song. Another

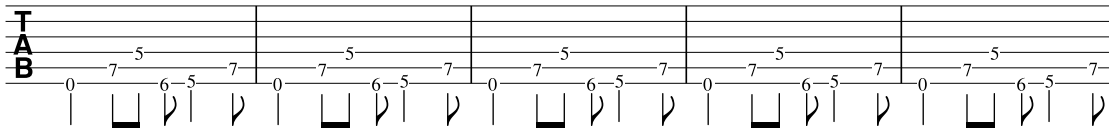


Figure 1.2: First measures from the guitar riff of the song “Enter Sandman” by Metallica.

enthusiastic fan goes through the lengthy process of transcribing the entire song by ear and publishes the scores on the same website shortly afterward. Unfortunately, it is written in CWMN instead of tabulature. She grabs her smartphone and takes a picture of the music score. That picture is processed by an OMR system that produces a digital version of the scores that she can open in a music score editor. The editor supports her in automatically converting the music into tabulature notation which she can comprehend. After playing alone for half a year, she decides to join a band. But given her lack of experience, she struggles to keep up with the other musicians. So she decides to practice every day at home, but without the accompanying voices she does not really get in the right mood for the music. So she grabs her smartphone again and takes a picture of the full score with all voices of the band. The OMR system detects and reads all voices and produces a digital version of the song that she can play along to, at a slower tempo. After a while, she disables the guitar voice and just keeps the other voices to simulate the presence of her bandmates while she keeps practicing.

Eventually, she learns how to read and write CWMN and composes her first song for the band. She writes it down on a piece of paper (see Fig. 1.3).

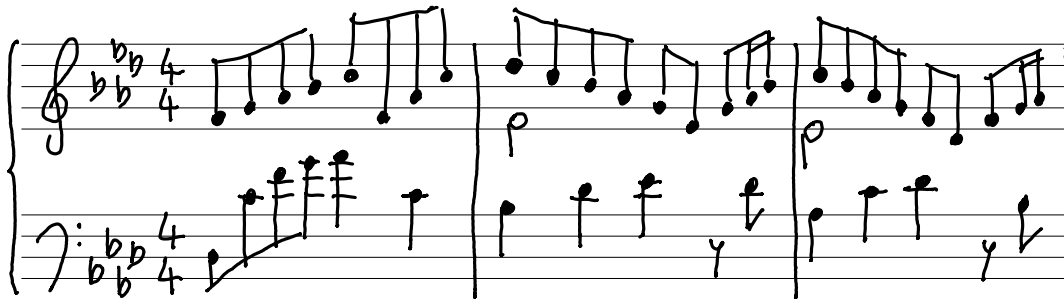


Figure 1.3: The initial three measures of Lisa’s first composition for the piano.

However, Lisa is uncertain if she got everything right and how the song sounds when played on the piano. Again, she picks up her smartphone, takes a photo of her handwritten manuscript and runs the OMR application. While listening to the replay, she notices that the digital version has some errors, so she quickly fixes them in her music notation editor before creating the final version of a nicely rendered score that she hands over to her band.

I hope this short fictional story demonstrates the potential of Optical Music Recognition

in helping musicians learning and practicing their art. It can even be useful in the everyday life of professional musicians and composers. Completely new use-cases have been invented in the last few years, such as a digital music stand that turns pages automatically. The conductor can jump to a specific location in everyone’s music scores at the same time without having to wait for them to turn their pages manually.

Despite some existing applications and a history of over 50 years of research, OMR is still considered a wide-open challenge for everything except very simple music scores. While there are a few commercial applications, they all have significant drawbacks and are far away from products that can be used to digitize music scores robustly on a larger scale. For example, a common wish of many musicians and librarians would be to have a born-digital version of the International Music Score Library Project (IMSLP), which is the largest collection of freely available music scores with over 460.000 scores. But instead of using commercial products, initiatives like OpenScore [GJB⁺18] rather use humans to digitize these scores manually. A similar approach is also used in many libraries, as Laplante and colleagues learned from interviews with librarians [LF16]. While the potential benefit is unquestioned, they still refrain from using OMR system because of the high error rate.

So why is OMR still performing so poorly? There are a couple of reasons. Underestimating the challenges is probably the most common one. Whenever someone joins the field, they see some scores like the example in Fig. 1.4 and classify it a moderately difficult task. It is only until they actually start building the system when they realize the number of problems which the recognition entails.



Figure 1.4: A born-digital version of music scores, typeset by a music score editor and without artifacts or degradations.

A (naive) computer scientist might see the score above and think: “There are always five parallel lines, larger and smaller black dots with vertical lines going up or down and several additional glyphs. This task of recognizing the symbols can be solved by running a line-detector to find the horizontal staff lines which should be removed first. Then a connected-component analysis can be applied to find the individual symbols. Finally, one runs a few scan-lines and template matching algorithms to find the remaining symbols, which should result in the recognition of everything in that image.” However, that is only half of the story. First of all, scores more often look like Fig. 1.5. Or they might even be handwritten, like Fig. 1.6.



Figure 1.5: The same musical snippet as in Fig. 1.4, but degraded, as it can happen in real-world scenarios: The staff is slightly slanted, the image is blurred and noisy due to a poor image capturing process, and some straight lines are bent, which frequently happens when making photos of scores that are bound in a book.

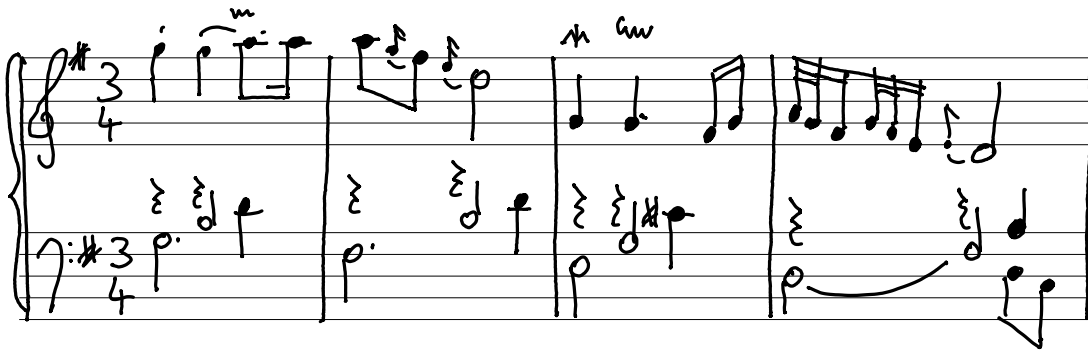


Figure 1.6: The same musical snippet as in Fig. 1.4, but handwritten on a tablet with a stylus.

What can be learned from these examples is that the same scores might look very different, although containing the same information: the staff lines might be skewed, or the image quality so poor that it can be difficult to reliably count the number of flags attached to grace-notes or distinguish an articulation dot from noise. Humans usually fill this gap with their experience and prior knowledge about the rules of music notation. Given two ways how to interpret a particular situation, they chose the one which makes more sense.

But even if we were able to devise a perfect algorithm for detecting everything in that score, i.e. we know exactly which pixel belongs to which object and have the right class information for each object (e.g., quarter rest, g-clef, or notehead), we would still only be half-way through because unlike Optical Character Recognition (OCR) which tries to read texts, OMR attempts to read music notation. And unlike text, music notation is a configurational writing system. This means that the semantics of the primitives, appearing in music scores are determined by their configuration, i.e. the position and positional relationship to other primitives. In other words, the letter 'a' in the Word 'Research' remains an 'a', regardless of whether it is slightly shifted upwards or downwards,

whereas an ‘A’ in music scores becomes a ‘B’ when moving it a little bit up or a ‘G’ when moving a little bit down (see Fig. 1.7).



Figure 1.7: The word ‘Research’ written three times with vertically shifted letters, which always remains the word research, whereas the values of the three notes that are also slightly shifted vertically represent three different notes with the pitches A, B, and G.

Apart from knowing the vertical position of a note within the reference-system of five parallel lines, the pitch can furthermore be altered by the presence of accidentals before that note, the clef at the beginning of the staff, the key signature as well as other symbols that might appear in the music score. To illustrate this effect of how primitives interact with each other, consider the snippet in Fig. 1.8.

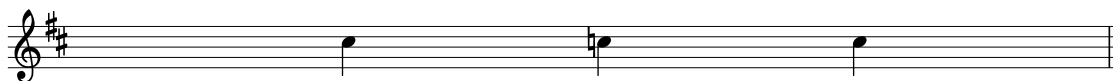


Figure 1.8: Three quarter-notes appear in the second space from the top within the reference system. The reference system’s origin is given by the G-Clef at the beginning, which specifies the G to be on the second line from the bottom. So the first note corresponds to a C, but with the given key-signature at the beginning which depicts two sharps with one of them placed on the second space from the top, it makes the note a C#. The second note has a local modifier that undoes this alteration from the key signature, which makes the note a C. The third note has no local modifier, but the effect of the local modifier from the second note is propagated to consecutive notes within the measure, making it also a C. So even if the first and third note visually look exactly the same, their semantics (pitch) is different.

As demonstrated, OMR requires more than just the recognition of the primitives, i.e. something like the construction of a (notation) graph that holds the configuration of the primitives and their relationships. And finally, the generation of music notation in the desired machine-readable format, typically a standard for music exchange, such as MIDI, MusicXML or MEI. Both tasks can become very complex when a system tries to recognize and process more sophisticated scores.

OMR can also be seen as teaching the machine to read and understand music scores to a certain extent. A task that certainly can be automated, as demonstrated by many applications as early as 1985, with the Wabot-2 robot [MSH⁺85] reading music scores

and playing them on an organ. Unfortunately, the robot was only capable of playing a very limited number of songs. Likewise, many systems that were developed in the last thirty years only worked well on the limited set of scores that were used during their development. The reason is that the machines did not learn to read music scores, but were given a set of rules and processing directives by their developers optimized to a certain body of music. If a music sheet violates the assumptions that were weaved into these rules or contained cases that were forgotten during development, the system breaks down and propagates errors through the process. This state of affairs is not satisfying. It would be preferable if an OMR system was more independent from the developer and datasets. Ideally, the system would learn the rules of music by itself and be more generalizable, extensible and robust. This brings us to the fundamental research question of this thesis:

Can a machine *learn* to read music scores reliably?

Throughout the last few years I investigated this question from several perspectives and tried to find ways of how I can teach the computer to learn reading music scores mostly by itself. The central idea is to devise a data-driven approach that requires as little human intervention as possible. The most suitable technology for this approach available today is Machine Learning, especially Deep Learning, which has proven to provide superior solutions to many image recognition problems among other things.

Given that developing an entire OMR system can be very complex, I decided to adapt existing workflows and reformulate the individual steps to make them machine-learnable. They are:

1. *Detect and analyze the structure of the music score:* This can be a simple decision, whether there are scores in the image at all, or finding the positions of staves and measures, depending on the design of the following steps.
2. *Find all objects in the music score:* Music scores can contain hundreds of (tiny) objects in a single image. This step is responsible for finding them and classifying them accordingly. In computer vision, this task is called object detection, and its goal is to retrieve the bounding boxes and class labels of all objects in an image.
3. *Understanding the relationship between music objects:* Once the individual objects are found, their relationship has to be determined, and a notation graph can be constructed that holds this information.
4. *Exporting the notation graph into music notation:* The complete notation graph is still an abstraction that cannot be read by music notation editors or other programs. It needs to be exported into a portable format to enable compatibility with these editors.

Except for the last step, I investigated how to machine-learn them and published my findings in the following articles.

Understanding Optical Music Recognition

During the last two years, I worked closely together with several other researchers. Most notably was my collaboration with Jorge Calvo-Zaragoza from the University of Alicante and Jan Hajič jr. from the University of Prague. We co-authored several papers, and our biggest venture was the paper “Understanding Optical Music Recognition” [CZHjP19]. It is currently under review as tutorial paper for the ACM Computing Survey series. It discusses fundamental questions, such as: What is OMR? Why is it worth attempting? What are the underlying challenges that make it into such a hard problem? What are the outputs of OMR systems and how to classify existing research with regards to them?

To understand what OMR is, we collected and reviewed more than 200 papers that define or talk about OMR in many different ways. We tried to put an umbrella over them by proposing the following definition, which we hope will be adopted by future researchers:

Optical Music Recognition is the field of research that investigates how to computationally read music notation in documents.

The second major contribution from this paper is an in-depth analysis of how OMR inverts the music encoding process. We begin with the creation of a musical composition, how it is conceptualized, and then materialized. We then show how OMR can be described as the inversion of the encoding process.

Furthermore, we discuss how OMR relates to other fields, such as Text Recognition or other Graphics Recognition challenges and what makes it particularly different from them, including the complex typographical alignment of objects, the interactions between objects and the extremely complex semantics, which can even be hard for humans to interpret correctly.

Finally, we propose a comprehensive taxonomy of OMR inputs and outputs. We realized that the complexity of OMR systems is directly related to the required level of comprehension of the document. We propose four categories, starting with document metadata extraction that requires only limited comprehension up to structured encoding, which not only tries to recover the musical content, but also the information on how it was encoded.

We conclude the paper with a brief discussion of current approaches, but in contrast to most survey papers, we do not discuss technical details. We also provide a list of open issues and perspectives for future research.

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), 50 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; Jan Hajič jr. Charles University, Prague, Czech Republic, 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.

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 difficult 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 contributions of this paper are 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, it is difficult to clearly relate their work to the field, as illustrated in Section 2.

Many authors also think of OMR as 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](#)).¹

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.

¹<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. These solutions 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 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 not. 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](#).

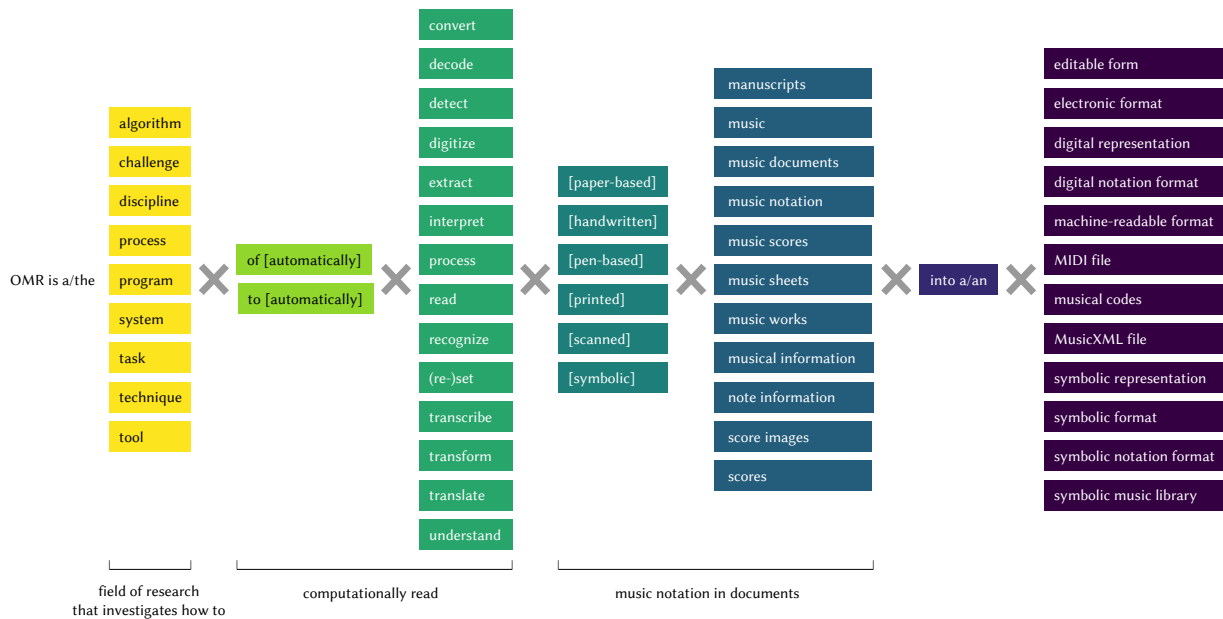


Fig. 1. How OMR tends to be defined or described and how our proposed definition relates to them. For example: “OMR is the challenge of (automatically) converting (handwritten) scores into a digital representation.”

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,² 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.³ 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).⁴ 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

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

³Musical 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.

⁴Feist [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 evolved 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.e. 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 engrave⁵ 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.⁶ 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.

Note that step (3) 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

⁵Normally, 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.

⁶While 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.

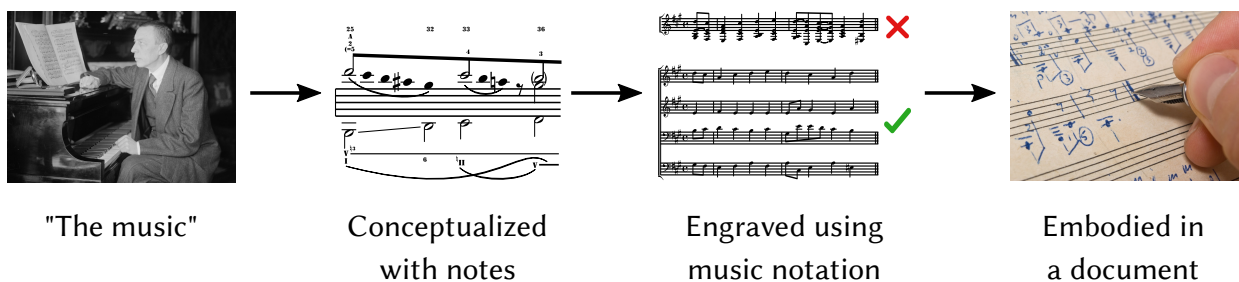


Fig. 3. How music is typically expressed and embodied (written down).

are reading the very same piece of music [98].⁷ 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) in the analysis above 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.
- (B) *Recover musical semantics*, which we define as the notes, represented by their pitches, velocities, onsets, and durations. This corresponds to stage (2)—we use the term “semantics” to refer only to the information that can be unambiguously inferred from the music notation

⁷Analogously, 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].

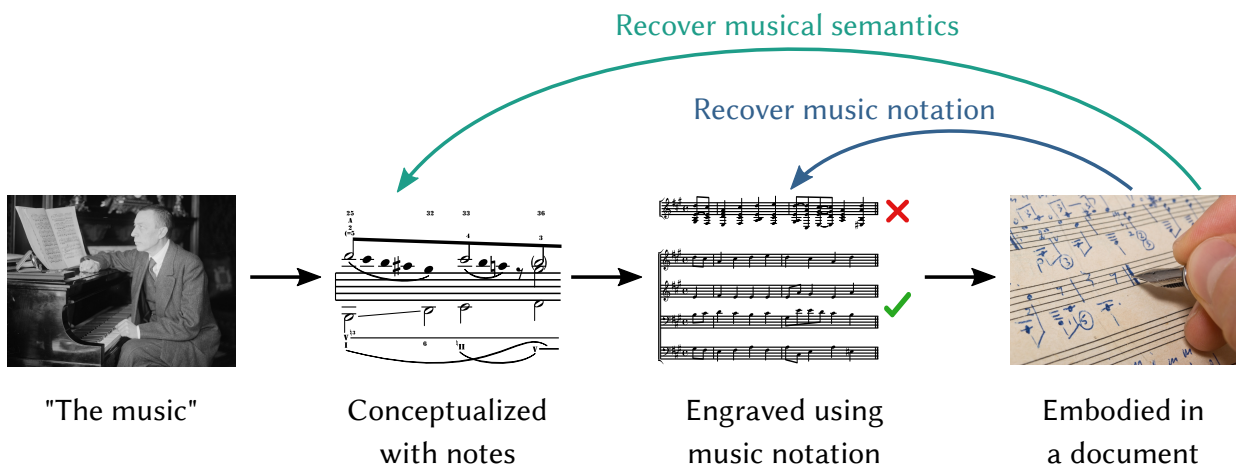


Fig. 4. How “reading” music can be interpreted as the operations of inverting the encoding process.

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 Recognition⁸, 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.

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.

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

⁸which is similar to the German equivalent “Optische Notenerkennung”

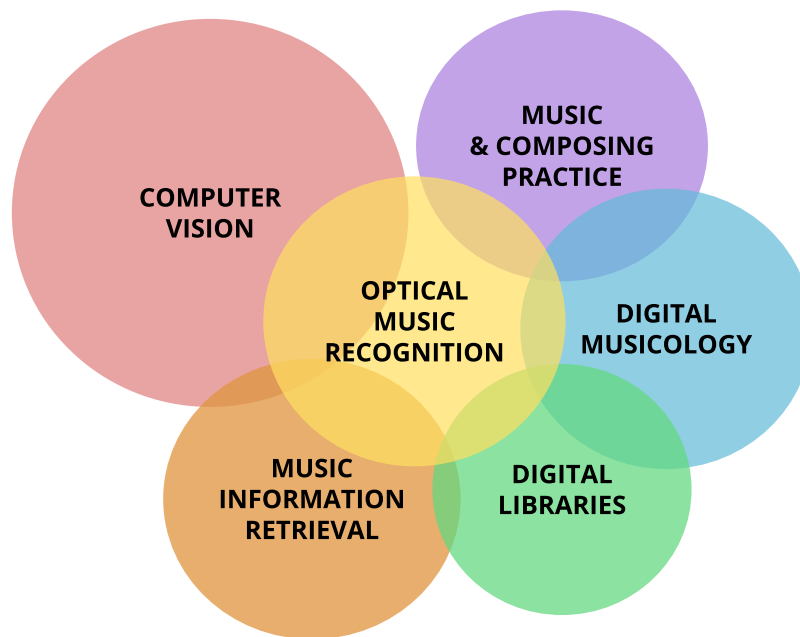


Fig. 5. Optical Music Recognition with its most important related fields, methods, and applications.

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 frequently been called “OCR for music” [25, 26, 68, 80, 93, 94, 109, 128, 129, 147]?⁹ 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 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

⁹Even 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.

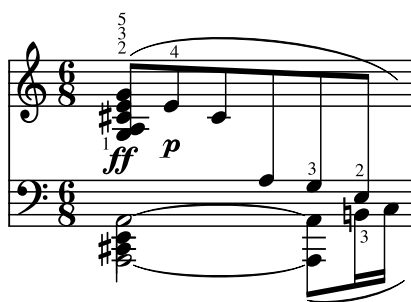


Fig. 7. This excerpt by Ludwig van Beethoven, Piano Sonata op. 2 no. 2, Largo appassionato, m. 31 illustrates some properties of the music notation that distinguish it from other types of writing systems: a wide range of primitive sizes, the same primitives appearing at different scales and rotations,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

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

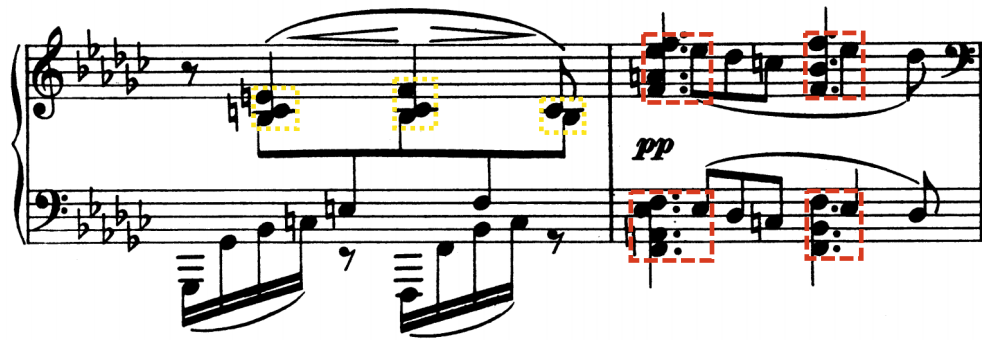


Fig. 8. Brahms Intermezzo, Op. 117 no. 1. Adjacent notes of the chords in the first bar in the top staff are shifted to the right to avoid overlappings (yellow dotted boxes). The moving eighths in the second bar are forced even further to the right, although being played simultaneously with the chord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

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 beat.¹⁰ 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).

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

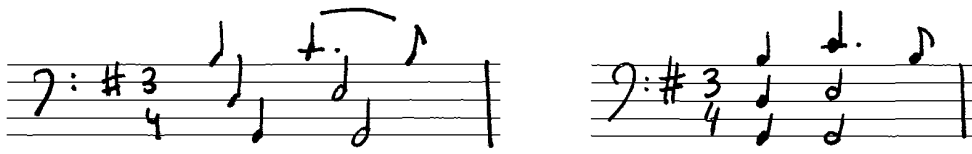


Fig. 9. Sample from the CVC-MUSCIMA dataset [60] with the same bar transcribed by two different writers. The first three notes and the second three notes form a chord and should be played simultaneously (see right figure) but is sometimes horizontally spelled out (see left figure) left is sometimes used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

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:

“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*¹¹ 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*¹², 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 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.

¹¹Although 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.

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

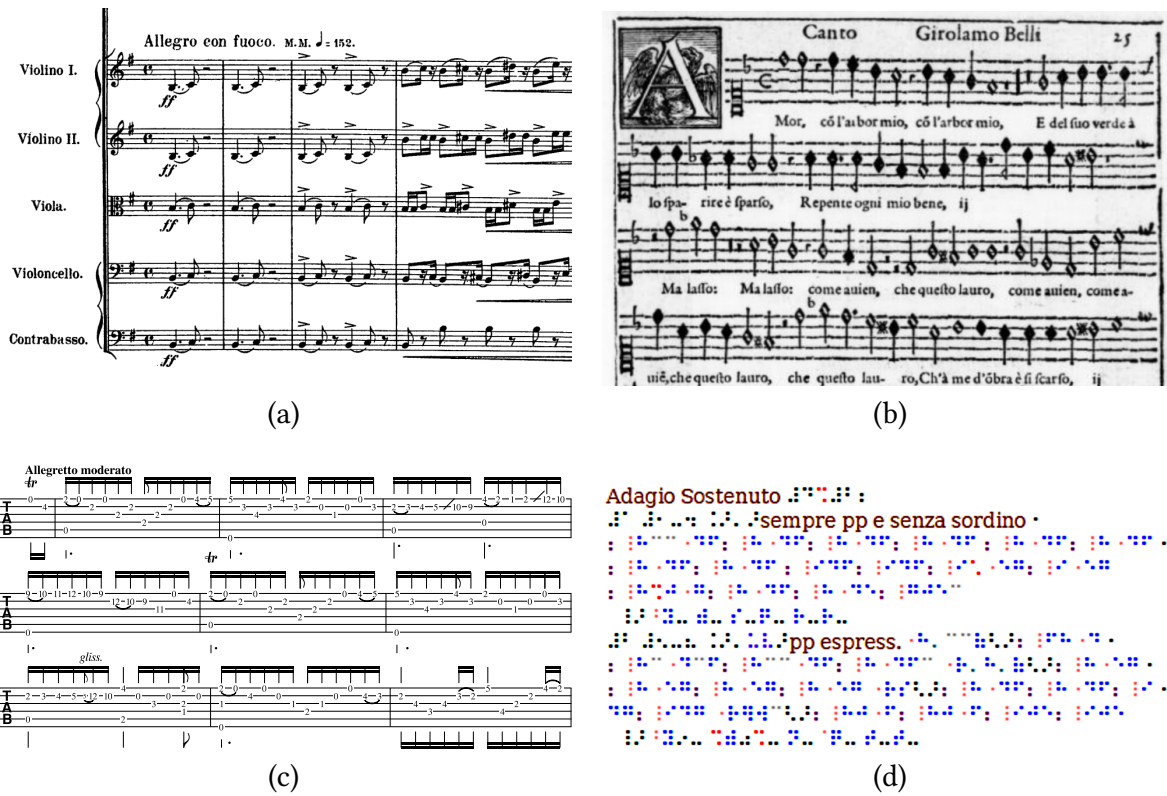


Fig. 11. Examples of scores written in various notations: (a) Common Western Music Notation (Dvorak Symphony No.9, IV), (b) White Mensural Notation (Belli [121]), (c) Tabulature (Regondi, Etude No.10) and (d) Braille (Beethoven, Sonata No.14 Op.27 No.2).

(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 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

(a) Monophonic

(b) Homophonic

(c) Polyphonic

(d) Pianoform

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

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]¹³. 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,¹⁴ while other projects still prefer to

¹³The 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].

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

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.
- *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. For example 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.¹⁵

Multiple similarity metrics for comparing MIDI files have been proposed during the Symbolic Melodic Similarity track of the Music Information Retrieval Evaluation eXchange (MIREX),¹⁶ 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*¹⁷ [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 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.

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,

¹⁵This 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.

¹⁶https://www.music-ir.org/mirex/wiki/MIREX_HOME

¹⁷roughly translated from German as “ideal conductor”

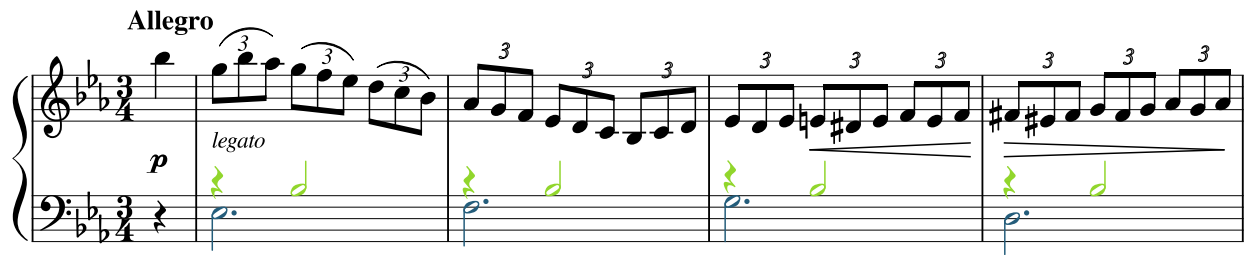


Fig. 14. Beginning of Franz Schubert, Impromptu D.899 No. 2 with omitted thirds starting in the second measure of the top staff (gray) and a color-coding of the two distinct voices in the second staff (green and blue).

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:¹⁸ 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 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,¹⁹ and there is an ongoing effort in the development of the Music Encoding Initiative format [137], best illustrated by the annual Music Encoding Conference.²⁰ 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

¹⁸http://music.informatics.indiana.edu/don_notation.html

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

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

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 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 intrinsic²¹ 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 yet have 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 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

²¹Extrinsic 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.

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].²² 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].

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

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

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.
- *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.

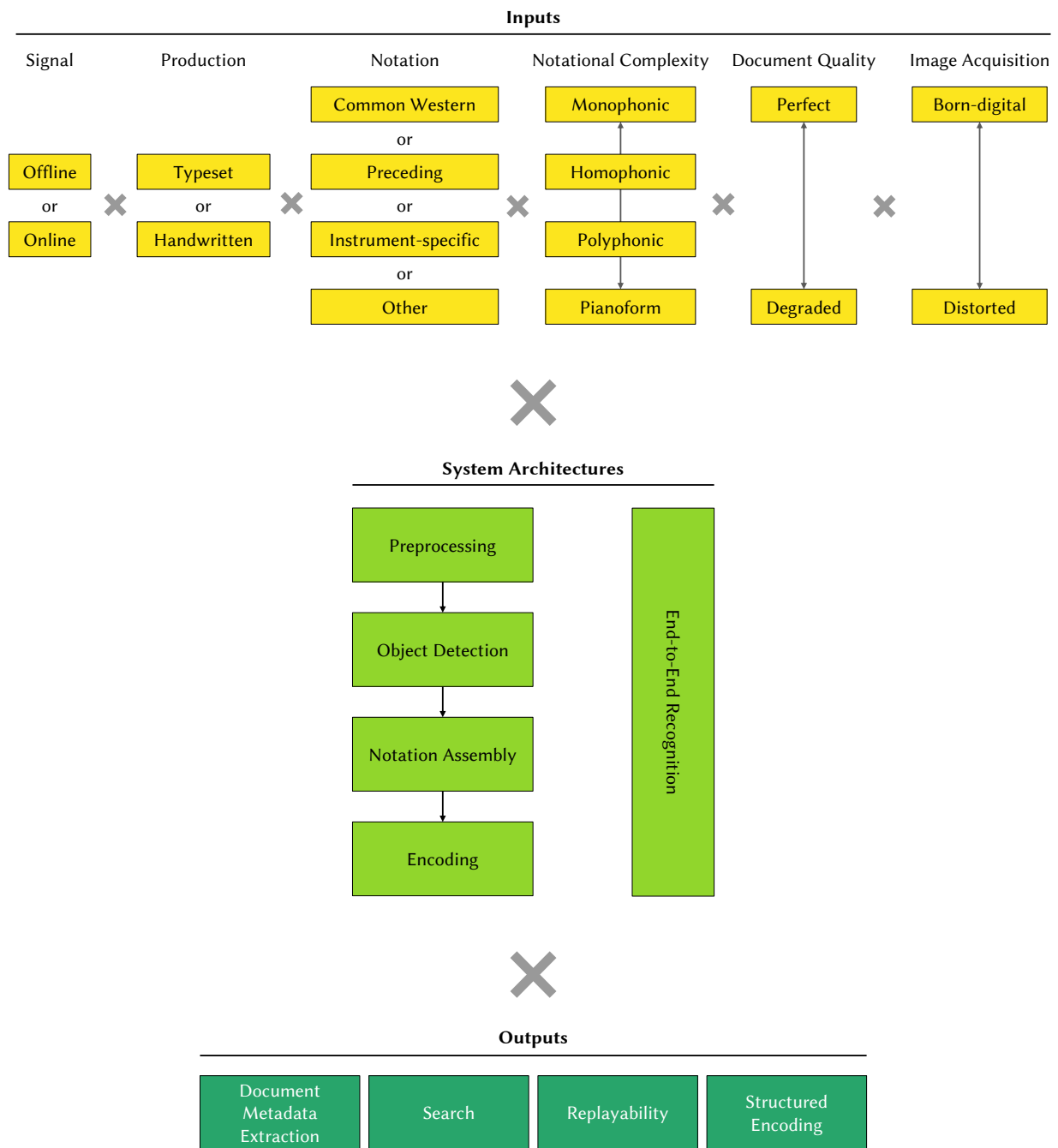


Fig. 15. An overview of the taxonomy of OMR inputs, architectures, and outputs. A fairly simple OMR system could, for example, read high-quality scans (offline) of well-preserved documents that contain typeset, monophonic, mensural notation, process it in a tradition pipeline and output the results in 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.

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, Toshiaki 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*. 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. In *Advances in Structural and Syntactic Pattern Recognition*. World Scientific, 363–372.
- [17] Mert Bay, Andreas F. Ehmman, 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. In *Structured Document Image Analysis*. Springer Berlin Heidelberg, 405–434.
- [23] Dorothea Blostein and Nicholas Paul Carter. 1992. Recognition of Music Notation: SSPR’90 Working Group Report. In *Structured Document Image Analysis*. 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. In *A New Companion to Digital Humanities*. 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 Hajič jr., 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* 8, 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. In *Advances in Structural and Syntactic Pattern 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*. 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 Hajič jr., 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 Giakoupiis, 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*. 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 Hajič jr. 2018. A Case for Intrinsic Evaluation of Optical Music Recognition. In *1st International Workshop on Reading Music Systems*. Paris, France, 15–16.
- [82] 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*. Paris, France, 225–232.
- [83] 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 *5th International Conference on Digital Libraries for Musicology*. Paris, France, 57–61.
- [84] Jan Hajič jr., 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, USA, 157–163.
- [85] Jan Hajič jr. 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 Hajič jr. 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. In *Graphics Recognition Algorithms and Applications*. 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*. 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] Chirstopher 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*. 215–218.
- [110] Tam Nguyen and Guesang 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 Couasnon, 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*. Kyoto, Japan, 35–36.

- [119] Alexander Pacha, Jan Hajič jr., 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 Modern-Early 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*. 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*. 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*. 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*. 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> (Last visited 16.04.2019). <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).

APPENDIX A: OMR BIBLIOGRAPHY

Along with this paper, we are also publishing the most comprehensive and complete bibliography on OMR that we were able to compile at <https://omr-research.github.io/>. It is a curated list of verified publications in an open-source Github repository (<https://github.com/OMR-Research/omr-research.github.io>) that is open for submissions both via pull requests and via templated issues. The website is automatically generated from the underlying BibTeX files using the BibTeX2HTML library, available at <https://www.lri.fr/~filliatr/bibtex2html/>.

The repository contains three distinct bibliographic files that are rendered into separate pages:

- (1) *OMR Research Bibliography*: A collection of scientific and technical publications, that were manually verified for correctness from a trustworthy source (see below). Most of these entries have either a Digital Object Identifier (DOI) or a link to the website, where the publication can be found.
- (2) *OMR Related Bibliography*: A collection of scientific and technical publications, that were manually verified for correctness from a trustworthy source but are not primarily directed towards OMR, such as musicological research or general computer vision papers.
- (3) *Unverified OMR Bibliography*: A collection of scientific and technical publications, that are related to Optical Music Recognition, but they could not be verified from a trustworthy source and might contain incorrect information. Many publications from this collection were authored before 1990 and are often not indexed by the search engines, or the respective proceedings could no longer be accessed and verified by us.

Acquisition and Verification Process

The bibliography was acquired and merged from multiple sources, such as the public and private collections from multiple researchers that have historically grown, including a recent one by Andrew Hankinson, who provided us with an extensive BibTeX library. Additionally, we have a Google Scholar Alert on [174] as it currently represents the latest survey and is cited by almost every publication.

To verify the information of each entry in the bibliography, we proceeded with the following steps:

- (1) Search on Google Scholar for the title of the work, if necessary with the authors last name and the year of publication.
- (2) Find a trustworthy source such as the original publisher, the authors' website, the website of the venue (that lists the article in the program) or indexing services including IEEE Xplore Digital Library, ACM Digital Library, Springer Link, Elsevier ScienceDirect, arXiv.org, dblp.org or ResearchGate. Information from the last three services are used with caution and if possible backed up with information from other sources.
- (3) Manually verify the correctness of the metadata by inspecting and correct it by obtaining the necessary information from another source, e.g., the conference website or the information state in the document. Suspicious information could be if the author's name is missing letters because of special characters or if the year of publication is before that of cited references.

Once we verified the entry, we add it to the respective bibliography with JabRef (<http://www.jabref.org/>) and link the original PDF file or at least the DOI. Articles that were only found as PDF without the associated venue of publication were classified as technical reports. Bachelor theses and online sources such as websites of commercial applications were classified as 'Misc' because of the lack of an appropriate category in BibTeX.

APPENDIX B: LIST OF OMR DEFINITIONS AND DESCRIPTIONS FROM PUBLISHED WORKS

To demonstrate how versatile OMR was referred to in the literature, we collected a list of definitions and descriptions (alphabetically ordered by the first author name). While most of these are direct citations (we omitted quotation marks for better readability), some were shortened or slightly rephrased to unify their structure and make them comparable.

Optical Music Recognition has been defined or described as:

- technology which transforms sheet music or printed scores into a machine readable format [1]
- automatic recognition and classification of symbolic music notation [2]
- system that aims to minimise human involvement in music input. The musical score is scanned to a bitmap image, and the computer attempts to parse the bitmap [3]
- form of structured document analysis where symbols overlaid on the conventional five-line stave are isolated and identified so that the music can be played through a MIDI system, or edited in a music publishing system [5]
- identifying musical symbols on a scanned sheet of music, and interpreting them so that the music can either be played by the computer, or put into a music editor [6]
- system to convert optically scanned pages of music into a versatile machine-readable format [4]
- system that aims at converting optically scanned pages of music into a versatile machine-readable format [9]
- system that aims at converting the vast repositories of sheet music in the world into an on-line digital format [11]
- computer system that can 'read' printed music [7]
- system that can be used to convert music scanned from paper into a format suitable for playing or editing on a computer [8]
- technique that makes it possible to automatically build indexes on the actual content of sheet music [10]
- process to automatically extract symbolic note information from scanned pages [12]
- system to convert sheet music images to symbolic music representations [13]
- the recognition of music scores [15]
- field devoted to transcribe sheet music into some machine-readable format [14]
- the process to convert a music score image into a machine-readable format [16]
- task of transcribing a music score into a machine readable format [17]
- task of recognizing and interpreting printed music and its transformation into MIDI [19]
- research directed towards the recognition of printed scores as well as handwritten music notation [18] (Actually referred to as Optical Music Reading)
- systems for music score recognition [20]
- software that recognises music notation and produces a symbolic representation of music [21]
- key problem for coding western music sheets in the digital world [22]
- system that aims at saving time in converting hardcopy of the music score into an electronic version [23]
- task devoted to convert an image of a music score into a machine-readable format, such as MIDI, MEI or MusicXML [179]
- systems that consist of three main steps, namely image pre-processing, symbol recognition and musical reconstruction [24]

- software unit called Computerized Note Recognition, whose function is to interpret and recognize handwritten musical notes [26]
- musical analog to optical character recognition [27]
- musical analogue to optical character recognition [29]
- converting images of musical scores into faithful symbolic representations of the same score [28]
- electronic conversion of scanned or photographed images of handwritten or printed sheet music into symbolic and therefore editable form [30]
- automatic score transcription tool [36]
- task of automatically extracting the musical information from an image of a score in order to export it to some digital format [32]
- offline music score recognition systems [37]
- field (of computer science) devoted to providing computers (with) the ability to extract the musical content of a score from the optical scanning of its source [40, 42, 44, 46]
- ability of a computer to understand the musical information contained in the image of a music score [35]
- field of computer science devoted to understanding the musical information contained in the image of a music score [38]
- research field that consists in [sic] extracting the musical content of a given score image in a structured, symbolic format [41]
- field devoted to the automatic transcription of sheet music into some machine-readable format [43]
- branch of artificial intelligence, focused on automatically recognizing the content of a musical score from the optical scan of its source [45]
- systems to import a scanned version of the music sheet and try to automatically export the information into some type of structured format such as MusicXML, MIDI or MEI [34]
- systems, whose objective is to automatically extract the information contained in the image of a musical score [48]
- system to automatic transcription of musical documents into a structured digital format [47]
- field of research that investigates how to computationally decode music notation from images [39]
- research field that focuses on the automatic detection and encoding of musical content from scanned images [33]
- research field that investigates how to make computers be capable of reading music [31]
- technology for automatically transcribing musical documents [25]
- digitization of music works [49, 50]
- computational process that reads musical notation from images, with the aim of automatically exporting the content to a structured format [52]
- technique that converts (or interprets) printed musical documents into computer readable/editable formats [60]
- automatic processing and analysis of images of musical notation [53]
- musical cousin of Optical Character Recognition, (which) seeks to convert score images into symbolic (music) representations [54, 59]
- system to transform score images into symbolic music libraries [57]
- key technology in Music Information Retrieval by mining symbolic knowledge directly from images of scores [56]
- seeking to convert music score images into symbolic representations [55, 58]
- software to convert scanned sheets of music into computer readable formats [62]

- software to generate a logical representation of the score [61]
- software to transform an image of a score into symbolic format [63]
- field of document analysis [64–66]
- automatic transcription of scores [67]
- process similar to the well-known optical character recognition to extract score data such as note events, the key and time signatures and other musical symbols [68]
- process of recognising a printed music score and converting it to a format that is understood by computers [69]
- program to automatically recognize music scores (translated from German “Ein Programm zur automatischen Erkennung von Musiknoten”) [70]
- the study of automatic techniques in information engineering, which can be used to determine the musical style of the singer [71]
- field to recognize and play live the notes in images captured from sheet music [72]
- process of automatically (re-)setting the score to create a symbolic, computer-readable representation of sheet music, such as MusicXML or MIDI [73, 74]
- technology that promises to accelerate the process of entering music scores in a machine-readable format by automatically interpreting the musical content from (the digitized image of) the printed score [75, 76]
- transformation of digital music score images to computer readable format symbols [77]
- automatic recognition of a scanned page of printed music [78, 79]
- research area that consists in [sic] the identification of music information from images of scores and their conversion into a machine readable format [80]
- process of identifying music information from images of scores and converting them into machine legible format [84]
- classical area of interest of Document Image Analysis and Recognition that combines textual and graphical information [86]
- classical application area of interest, whose aim is the identification of music information from images of scores and their conversion into a machine readable format [85]
- research field that consists in [sic] the understanding of information from music scores and its conversion into a machine readable format [87], [82]
- recognition of handwritten music scores [81], [83]
- automatic recognition of music notation by the computer [88]
- task of converting scanned sheet music into a computer readable symbolic music format such as MIDI or MusicXML [90]
- process of extracting musical note parameters (onset times, pitches, durations) along with 2D position parameters from the scanned image [89]
- task of converting scores into a machine-readable format [91]
- program for recognition of musical notation [92]
- technology which transforms digital images of music into searchable representations of music notation [93, 94]
- process of automatically transcribing music notation from a digital image [95]
- research field, which focuses on detecting and storing the musical content of a score from a scanned image. The objective is to import a scanned musical score and export its musical content to a machine-readable format, typically MusicXML or MEI [96]
- technique to transform paper musical scores into musical acoustic, and it is a basic way to apply to digital medium music data, large digital music library, robot reading musical score and perform, computer music education, Chinese tradition music digitalization [sic][97]
- technique to convert scanned pages of music into a machine-readable format [98]

- problem of recognising and interpreting the symbols of printed music notation from a scanned image [100]
- systems designed to perform recognition of music notation, chiefly from a scanned image of music notation [99]
- process that aims to “recognize” images of music notation and capture the “meaning” of the music [101]
- system for recognizing music notation [102]
- systems, designed to recognise printed sheet music scores [103]
- branch of OCR oriented to musical documents [104]
- field of document analysis that aims to automatically read musical scores [110]
- process that attempts to extract musical information from its written representation, the musical score [108]
- task of recovering symbolic musical information such as MIDI from the image of the written score [105]
- field of graphics recognition that aims to automatically read music [109]
- field of document analysis that aims to automatically read music [111]
- field of computationally reading music notation in documents [107]
- field of automatically reading music notation from images [106]
- tool for document transcription that tries to extract symbolic music from page images for use in an editor [113]
- technology that can transform large quantities of music document page images into searchable and retrievable document entities [112]
- field of research that attempts to transcribe musical symbols into digital format [114]
- process of structured data processing applied to music notation [115]
- research and technological field aimed on recognizing and representing music notation [117]
- technology to automatically recognize music notation [116]
- technique for processing music notes in old manuscripts and books [118]
- form of optical character recognition that use different method and algorithms to convert printed music into its digital form [sic] [119]
- direct path to create rich and extensive symbolic databases for music in machine-generated common Western notation [121]
- process that automatically converts the image of a music score into symbolic data [120]
- systems that convert music scores into a computer-readable format, similar to Optical Character Recognition (OCR) except that it is used to recognize musical symbols instead of letters [122]
- OCR for music [123]
- system that can play printed or handwritten music score images without any knowledge of music primitives or musical instruments [124]
- system to transform a sheet music into a format readable by a machine [125]
- case of optical character recognition for the automatic recognition and classification of music notation [126]
- system that can convert digital image data into digital semantic data [127]
- system that addresses the problem of musical data acquisition, with the aim of converting optically scanned music scores into a versatile machine-readable format [128, 129]
- subcategory of optical character recognition that recognizes an image of printed sheet music and interprets it to a machine-readable document [130]
- technology that is a rewarding subject for pattern recognition researches [131]
- system for music score recognition [132]

- technology that makes it possible to extract symbolic representations from scores or micro-films of scores [133]
- technique that involves interpreting the symbols in a picture, such as a scanned image of sheet music, and recreating the information in a format that encapsulates the implied audio content [134]
- process of converting a graphical representation of music (such as sheet music) into a symbolic format [135]
- process of automatically extracting musical meaning from a printed music score. It is sometimes also called musical score recognition or simply score recognition [137]
- process of automatically processing and understanding an image of a music score [136]
- process that recognized music from any form of score sheet and makes sheet readable and editable for computer [sic] [138]
- automatic conversion of scanned music scores into computer readable data in variable formats, e.g., MusicXML, or MEI [139]
- technique that achieves the automatic recognition of music notation with high-speed and further plays music automatically, which is an important topics (sic!) in the process [140]
- process to convert handwritten music symbols on sheets of paper into computer readable data [141]
- systems that analyze and convert digitized music scores to machine readable formats [142]
- process of automatically recovering the information present on music scores based on scanned data [144]
- input technique to obtain a machine representation of music [147]
- efficient and automatic method to transform paper-based music scores into a machine representation [148]
- system that can provide an automated and time-saving input method to transform paper-based music scores into a machine readable representation, for a wide range of music software, in the same way as Optical Character Recognition is useful for the processing applications [145]
- system to transform paper-based music scores and manuscripts into a machine-readable symbolic format [146]
- equivalent task for music, that is OCR for digital images of words [149]
- system that can automatically interpret the images and automatically create new scores that can be understood by the computer [150]
- discipline that investigates music score recognition systems [151]
- area of document analysis that aims to automatically understand written music scores. Given an image of musical scores, an OMR system attempts to recognize the content and translate it into a machine-readable format such as MusicXML [155]
- branch of artificial intelligence that aims at automatically recognizing and understanding the content of music scores [156]
- challenge of understanding the content of musical scores [154]
- research field that investigates how to automatically decode written music into a machine-readable format [152]
- field of research that investigates how to build systems that decode music notation from images [153]
- field of research that investigates how to computationally read music notation in documents [157]
- task of recognizing all music symbols in a score sheet [158]

- system to convert music scores into a machine-readable data that could be reproduced in computer and stored as compact digitalised data [sic] [159]
- process of identifying music from an image of a music score [160]
- system to transform paper-based music scores and manuscripts into a machine-readable symbolic format [161, 162]
- tools for the creation of searchable digital music libraries [163]
- systems that create encodings of the musical content in digital images automatically [164]
- musical analogue to optical character recognition (OCR) [165]
- applications that enable document images to be encoded into digital symbolic music representations [167]
- the equivalent of OCR for music [166]
- pathway to a large set of symbolic scores [168]
- analogous to optical character recognition to convert music score images into symbolic form [169]
- form of structured document image analysis where music symbols are isolated and identified so that the music can be conveniently processed [172]
- system to transform paper-based music scores and manuscripts into a machine-readable (symbolic) format [51, 170, 173]
- system with three main objectives: the recognition, the representation and the storage of musical scores in a machine-readable format [177]
- tool for the automatic recognition of digitized music scores [176]
- computer system that can automatically decode and create new scores [174]
- research field, that deals with the recognition, the representation and the storage of musical scores in a machine-readable format [171]
- tool to transform pen-based music scores and manuscripts into a machine-readable symbolic format [175]
- system capable of recognizing printed music of reasonable quality [178]
- task of recognizing images of musical scores [180]
- recognition of images of musical scores [181]
- key tools for publication of music score collections that are currently found only on paper [182]
- system that can automatically recognize the main musical symbols of a scanned paper-based music score [183]
- field of research that aims at reading automatically scanned scores in order to convert them in an electronic format, such as a midi file [184]
- method that aims at automatically reading scanned scores [185]
- method that aims at automatically reading scanned scores in order to convert them into an electronic format, such as MIDI file, or an audio waveform [186]
- automatic recognition of a scanned page of printed music notation by a computer program [187]
- translation of a digitized image of a music score into a representation more amenable to computer manipulation of the musical content [188]
- systems that analyse images of music scores to convert their content to machine readable formats [143]
- problem of obtaining a complete representation of a musical document given only a digital image [189]
- problem of recognizing musical scores in images [190]

- application of optical character recognition to interpret sheet music or printed scores into editable or playable form [191]
- systems that play a very important role in the process of creating the digital libraries of musical documents [192]
- tools for automatic sheet music transcription [193]
- system for extracting musical symbols from images similar to the Optical Character Recognition [194]
- method that involves identifying musical symbols on a scanned sheet of music and transforming them into a computer readable format [195]
- process of converting digitized sheets of music into an electronic form that is suitable for further processing such as editing and performing by computer [196]
- efficient and automatic method for transforming paper-based music scores into a machine representation [197]
- algorithm for processing images of musical scores [198]
- work for automatically recognizing music expressions for printed and handwritten music [199]
- program to convert scanned score into an electronic format and even recognize and understand the contents of the score [200]
- application to automatically transcribe digitized page images of music [202]
- automatic recognition of a scanned music score [203]
- system to input music by detecting musical symbols, based on strokes drawn by the user [201]
- automatic recognition of scanned music scores [203]
- area of document recognition and computer vision that aims at converting scans of written music to machine-readable form, much like optical character recognition [204]
- area within music information retrieval with the goal of transforming images of printed or handwritten music scores into machine readable form, thereby understanding the semantic meaning of music notation [205]
- process of identifying music from an image of a music score [207]
- process of turning musical notation represented in a digital image into a computer-manipulable symbolic notation format [208]
- process of converting a scanned image of pages of music into computer readable and manipulable symbols using a variety of image processing techniques [209]
- process that reads and extracts the content from digitized images of music documents [210]
- a computer system for automatically storing and interpreting musical information (of music scores) [212]
- system that can automatically interpret images of music scores and create new scores that the computer could understand [211]
- particular case of high-level document analysis [214, 215]
- task of interpreting the content of the bitmap image of a musical score and reformulating it with a high-level symbolic structure [213]
- way to convert music notation into a digital representation, and its acoustic rendition [216]
- systems whose main purpose is to convert images of paper-based music scores into digitised formats [217]
- application of recognition algorithms to musical scores, to encode the musical content to some kind of digital format [206]
- tool to recognize a scanned page of music scores automatically [218, 219]
- conversion of scanned pages of music into a musical database [220]

- process of a computer reading sheet music [221]
- process of converting paper sheets of music score into an electronic format which can be “read” by computer [222]
- tool that takes a score that is likely to be correct, scans it and tries to recreate what it scans in a digital notation format [223]

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] Jamie Anstice, Tim Bell, Andy Cockburn, and Martin Setchell. 1996. The design of a pen-based musical input system. In *6th Australian Conference on Computer-Human Interaction*. 260–267.
- [4] David Bainbridge. 1997. *Extensible optical music recognition*. Ph.D. Dissertation. University of Canterbury.
- [5] David Bainbridge and Tim Bell. 1996. An extensible optical music recognition system. *Australian Computer Science Communications* 18 (1996), 308–317.
- [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 Nicholas Paul Carter. 1997. Automatic reading of music notation. In *Handbook of Character Recognition and Document Image Analysis*. World Scientific, Singapore, 583–603.
- [10] David Bainbridge, Xiao Hu, and J. Stephen Downie. 2014. A Musical Progression with Greenstone: How Music Content Analysis and Linked Data is Helping Redefine the Boundaries to a Music Digital Library. In *1st International Workshop on Digital Libraries for Musicology*.
- [11] David Bainbridge and Stuart Inglis. 1998. Musical image compression. In *Data Compression Conference*. 209–218.
- [12] 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.
- [13] Stefan Balke, Christian Dittmar, Jakob Abeßer, Klaus Frieler, Martin Pfeleiderer, and Meinard Müller. 2018. Bridging the Gap: Enriching YouTube Videos with Jazz Music Annotations. *Frontiers in Digital Humanities* 5 (2018), 1–11.
- [14] 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*. Kyoto, Japan, 25–26.
- [15] 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.
- [16] 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.
- [17] Arnau Baró-Mas. 2017. *Optical Music Recognition by Long Short-Term Memory Recurrent Neural Networks*. Master’s thesis. Universitat Autònoma de Barcelona.
- [18] Stephan Baumann. 1995. A Simplified Attributed Graph Grammar for High-Level Music Recognition. In *3rd International Conference on Document Analysis and Recognition*. 1080–1083.
- [19] Stephan Baumann and Andreas Dengel. 1992. Transforming Printed Piano Music into MIDI. In *Advances in Structural and Syntactic Pattern Recognition*. World Scientific, 363–372.
- [20] Pierfrancesco Bellini, Ivan Bruno, and Paolo Nesi. 2001. Optical music sheet segmentation. In *1st International Conference on WEB Delivering of Music*. 183–190.
- [21] Pierfrancesco Bellini, Ivan Bruno, and Paolo Nesi. 2004. An Off-Line Optical Music Sheet Recognition. In *Visual Perception of Music Notation: On-Line and Off-Line Recognition*. IGI Global, 40–77.
- [22] Pierfrancesco Bellini, Ivan Bruno, and Paolo Nesi. 2008. Optical Music Recognition: Architecture and Algorithms. In *Interactive Multimedia Music Technologies*. IGI Global, Hershey, PA, USA, 80–110.
- [23] Tomáš Beran and Tomáš Macek. 1999. Recognition of Printed Music Score. In *Machine Learning and Data Mining in Pattern Recognition*. 174–179.
- [24] Alexandra Bonnici, Julian Abela, Nicholas Zammit, and George Azzopardi. 2018. Automatic Ornament Localisation, Recognition and Expression from Music Sheets. In *ACM Symposium on Document Engineering*. Halifax, NS, Canada, 25:1–25:11.

- [25] Vicente Bosch Campos, Jorge Calvo-Zaragoza, Alejandro H. Toselli, and Enrique Vidal Ruiz. 2016. Sheet Music Statistical Layout Analysis. In *15th International Conference on Frontiers in Handwriting Recognition*. 313–318.
- [26] Alex Bulis, Roy Almog, Moti Gerner, and Uri Shimony. 1992. Computerized recognition of hand-written musical notes. In *International Computer Music Conference*. 110–112.
- [27] 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.
- [28] John Ashley Burgoyne, Ichiro Fujinaga, and J. Stephen Downie. 2015. Music Information Retrieval. In *A New Companion to Digital Humanities*. Wiley Blackwell, 213–228.
- [29] John Ashley Burgoyne, Yue Ouyang, Tristan Himmelman, Johanna Devaney, Laurent Pugin, and Ichiro Fujinaga. 2009. Lyric Extraction and Recognition on Digital Images of Early Music Sources. In *10th International Society for Music Information Retrieval Conference*. Kobe, Japan, 723–727.
- [30] 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.
- [31] Jorge Calvo-Zaragoza. 2018. Why WoRMS?. In *1st International Workshop on Reading Music Systems*. Paris, France, 7–8.
- [32] Jorge Calvo-Zaragoza, Isabel Barbancho, Lorenzo J. Tardón, and Ana M. Barbancho. 2015. Avoiding staff removal stage in optical music recognition: application to scores written in white mensural notation. *Pattern Analysis and Applications* 18, 4 (2015), 933–943.
- [33] Jorge Calvo-Zaragoza, Francisco J. Castellanos, Gabriel Vigliensoni, and Ichiro Fujinaga. 2018. Deep Neural Networks for Document Processing of Music Score Images. *Applied Sciences* 8, 5 (2018).
- [34] Jorge Calvo-Zaragoza, Antonio-Javier Gallego, and Antonio Pertusa. 2017. Recognition of Handwritten Music Symbols with Convolutional Neural Codes. In *14th International Conference on Document Analysis and Recognition*. Kyoto, Japan, 691–696.
- [35] Jorge Calvo-Zaragoza, Luisa Micó, and Jose Oncina. 2016. Music staff removal with supervised pixel classification. *International Journal on Document Analysis and Recognition* 19, 3 (2016), 211–219.
- [36] 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.
- [37] Jorge Calvo-Zaragoza and Jose Oncina. 2015. Clustering of strokes from pen-based music notation: An experimental study. *Lecture Notes in Computer Science* 9117 (2015), 633–640.
- [38] Jorge Calvo-Zaragoza, Antonio Pertusa, and Jose Oncina. 2017. Staff-line detection and removal using a convolutional neural network. *Machine Vision and Applications* (2017), 1–10.
- [39] Jorge Calvo-Zaragoza and David Rizo. 2018. End-to-End Neural Optical Music Recognition of Monophonic Scores. *Applied Sciences* 8, 4 (2018).
- [40] Jorge Calvo-Zaragoza, David Rizo, and José Manuel Iñesta. 2016. Two (note) heads are better than one: pen-based multimodal interaction with music scores. In *17th International Society for Music Information Retrieval Conference*. New York City, 509–514.
- [41] 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.
- [42] Jorge Calvo-Zaragoza, Alejandro H. Toselli, and Enrique Vidal. 2017. Early handwritten music recognition with Hidden Markov Models. In *15th International Conference on Frontiers in Handwriting Recognition*. 319–324.
- [43] Jorge Calvo-Zaragoza, Jose J. Valero-Mas, and Antonio Pertusa. 2017. End-to-end Optical Music Recognition using Neural Networks. In *18th International Society for Music Information Retrieval Conference*. Suzhou, China.
- [44] Jorge Calvo-Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. 2016. Document Analysis for Music Scores via Machine Learning. In *3rd International workshop on Digital Libraries for Musicology*. New York, USA, 37–40.
- [45] Jorge Calvo-Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. 2017. A machine learning framework for the categorization of elements in images of musical documents. In *3rd International Conference on Technologies for Music Notation and Representation*. A Coruña, Spain.
- [46] Jorge Calvo-Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. 2017. 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.
- [47] Jorge Calvo-Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. 2017. Pixel-wise binarization of musical documents with convolutional neural networks. In *15th International Conference on Machine Vision Applications*. 362–365.
- [48] Jorge Calvo-Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. 2017. Pixelwise classification for music document analysis. In *7th International Conference on Image Processing Theory, Tools and Applications*. 1–6.

- [49] Artur Capela, Jamie dos Santos Cardoso, Ana Rebelo, and Carlos Guedes. 2008. Integrated recognition system for music scores. In *International Computer Music Conference*. 3–6.
- [50] Artur Capela, Ana Rebelo, Jamie dos Santos Cardoso, and Carlos Guedes. 2008. Staff Line Detection and Removal with Stable Paths. In *International Conference on Signal Processing and Multimedia Applications*.
- [51] Jamie dos Santos Cardoso, Artur Capela, Ana Rebelo, Carlos Guedes, and Joaquim Pinto da Costa. 2009. Staff Detection with Stable Paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 6 (2009), 1134–1139.
- [52] Fancisco J. Castellanos, Jorge Calvo-Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. 2018. Document Analysis of Music Score Images with Selectional Auto-Encoders. In *19th International Society for Music Information Retrieval Conference*. Paris, France, 256–263.
- [53] 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.
- [54] Liang Chen, Rong Jin, and Christopher Raphael. 2014. Optical Music Recognition with Human Labeled Constraints. In *CHI'14 Workshop on Human-Centred Machine Learning*. Toronto, Canada.
- [55] Liang Chen, Rong Jin, and Christopher Raphael. 2017. Human-Guided Recognition of Music Score Images. In *4th International Workshop on Digital Libraries for Musicology*.
- [56] Liang Chen, Rong Jin, Simo Zhang, Stefan Lee, Zhenhua Chen, and David Crandall. 2016. A Hybrid HMM-RNN Model for Optical Music Recognition. In *Extended abstracts for the Late-Breaking Demo Session of the 17th International Society for Music Information Retrieval Conference*.
- [57] Liang Chen and Christopher Raphael. 2016. Human-Directed Optical Music Recognition. *Electronic Imaging* 2016, 17 (2016), 1–9.
- [58] 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.
- [59] Liang Chen, Erik Stolterman, and Christopher Raphael. 2016. Human-Interactive Optical Music Recognition. In *17th International Society for Music Information Retrieval Conference*. 647–653.
- [60] Yung-Sheng Chen, Feng-Sheng Chen, and Chin-Hung Teng. 2013. An Optical Music Recognition System for Skew or Inverted Musical Scores. *International Journal of Pattern Recognition and Artificial Intelligence* 27, 07 (2013).
- [61] G. Sayeed Choudhury, Tim DiLauro, Michael Droettboom, Ichiro Fujinaga, and Karl MacMillan. 2001. Strike Up the Score: Deriving searchable and playable digital formats from sheet music. *D-Lib Magazine* 7, 2 (2001).
- [62] 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*.
- [63] Maura Church and Michael Scott Cuthbert. 2014. Improving Rhythmic Transcriptions via Probability Models Applied Post-OMR. In *15th International Society for Music Information Retrieval Conference*. 643–648.
- [64] Bertrand Couasnon, Pascal Brisset, and Igor Stéphan. 1995. Using Logic Programming Languages For Optical Music Recognition. In *3rd International Conference on the Practical Application of Prolog*.
- [65] 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.
- [66] Bertrand Couasnon and Jean Camillerapp. 1995. A Way to Separate Knowledge From Program in Structured Document Analysis: Application to Optical Music Recognition. In *3rd International Conference on Document Analysis and Recognition*. 1092–1097.
- [67] 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.
- [68] David Damm, Christian Fremerey, Frank Kurth, Meinard Müller, and Michael Clausen. 2008. Multimodal Presentation and Browsing of Music. In *10th International Conference on Multimodal Interfaces*. Chania, Greece, 205–208.
- [69] Arnaud F. Desaedeleer. 2006. *Reading Sheet Music*. Master's thesis. University of London.
- [70] Jürgen Diet. 2018. Optical Music Recognition in der Bayerischen Staatsbibliothek. *BIBLIOTHEK – Forschung und Praxis* (2018).
- [71] 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.
- [72] Cong Minh Dinh, Hyung-Jeong Yang, Guee-Sang Lee, and Soo-Hyung Kim. 2016. Fast lyric area extraction from images of printed Korean music scores. *IEICE Transactions on Information and Systems* E99D, 6 (2016), 1576–1584.
- [73] Matthias Dorfer, Andreas Arzt, and Gerhard Widmer. 2016. Towards Score Following In Sheet Music Images. In *17th International Society for Music Information Retrieval Conference*. 789–795.

- [74] Matthias Dorfer, Florian Henkel, and Gerhard Widmer. 2018. Learning To Listen, Read And Follow: Score Following As A Reinforcement Learning Game. In *19th International Society for Music Information Retrieval Conference*. Paris, France, 784–791.
- [75] Michael Droettboom and Ichiro Fujinaga. 2001. *Interpreting the semantics of music notation using an extensible and object-oriented system*. Technical Report. John Hopkins University.
- [76] Michael Droettboom, Ichiro Fujinaga, and Karl MacMillan. 2002. Optical Music Interpretation. In *Structural, Syntactic, and Statistical Pattern Recognition*. Berlin, Heidelberg, 378–387.
- [77] Yang Fang and Teng Gui-fa. 2015. Visual music score detection with unsupervised feature learning method based on K-means. *International Journal of Machine Learning and Cybernetics* 6, 2 (2015), 277–287.
- [78] Miguel Ferrand, João Alexandre Leite, and Amílcar Cardoso. 1999. Hypothetical reasoning: An application to Optical Music Recognition. In *Appia-Gulp-Prode'99 joint conference on declarative programming*. 367–381.
- [79] Miguel Ferrand, João Alexandre Leite, and Amílcar Cardoso. 1999. Improving Optical Music Recognition by Means of Abductive Constraint Logic Programming. In *Progress in Artificial Intelligence*. Berlin, Heidelberg, 342–356.
- [80] Alicia Fornés. 2005. *Analysis of Old Handwritten Musical Scores*. Master's thesis. Universitat Autònoma de Barcelona.
- [81] 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.
- [82] 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.
- [83] Alicia Fornés, Anjan Dutta, Albert Gordo, and Josep Lladós. 2013. The 2012 Music Scores Competitions: Staff Removal and Writer Identification. In *Graphics Recognition. New Trends and Challenges*. Berlin, Heidelberg, 173–186.
- [84] 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.
- [85] 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.
- [86] 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.
- [87] 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.
- [88] Stavroula-Evita Fotinea, George Giakoupi, 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 *RIAO '00 Content-Based Multimedia Information Access*. Paris, France, 1006–1014.
- [89] Christian Fremerey, David Damm, Frank Kurth, and Michael Clausen. 2009. Handling Scanned Sheet Music and Audio Recordings in Digital Music Libraries. In *International Conference on Acoustics NAG/DAGA*. 1–2.
- [90] 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.
- [91] Ichiro Fujinaga. 1988. *Optical Music Recognition using Projections*. Master's thesis. McGill University.
- [92] Ichiro Fujinaga. 1996. Exemplar-based learning in adaptive optical music recognition system. In *International Computer Music Conference*. Hong Kong, 55–56.
- [93] 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.
- [94] 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*. 1–3.
- [95] Ichiro Fujinaga, Andrew Hankinson, and Laurent Pugin. 2018. Automatic Score Extraction with Optical Music Recognition (OMR). In *Springer Handbook of Systematic Musicology*. Springer Berlin Heidelberg, Berlin, Heidelberg, 299–311.
- [96] Antonio-Javier Gallego and Jorge Calvo-Zaragoza. 2017. Staff-line removal with selectional auto-encoders. *Expert Systems with Applications* 89 (2017), 138–148.
- [97] Chen Genfang, Zhang Wenjun, and Wang Qiuqiu. 2009. Pick-up the Musical Information from Digital Musical Score Based on Mathematical Morphology and Music Notation. In *1st International Workshop on Education Technology and Computer Science*. 1141–1144.
- [98] Susan E. George. 2003. Online Pen-Based Recognition of Music Notation with Artificial Neural Networks. *Computer Music Journal* 27, 2 (2003), 70–79.
- [99] Susan E. George. 2004. Evaluation in the Visual Perception of Music Notation. In *Visual Perception of Music Notation: On-Line and Off Line Recognition*. IRM Press, Hershey, PA, 304–349.

- [100] Susan E. George. 2004. *Visual Perception of Music Notation On-Line and Off-Line Recognition*. IRM Press.
- [101] 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.
- [102] Velissarios G. Gezerlis and Sergios Theodoridis. 2002. Optical character recognition of the Orthodox Hellenic Byzantine Music notation. *Pattern Recognition* 35, 4 (2002), 895–914.
- [103] Roland Göcke. 2003. Building a system for writer identification on handwritten music scores. In *IASTED International Conference on Signal Processing, Pattern Recognition, and Applications*. 250–255.
- [104] Gianmarco Gozzi. 2010. *OMRjX: A framework for piano scores optical music recognition*. Master’s thesis. Politecnico di Milano.
- [105] Jan Hajič jr. and Matthias Dorfer. 2017. Prototyping Full-Pipeline Optical Music Recognition with MUSCIMARKER. In *Extended abstracts for the Late-Breaking Demo Session of the 18th International Society for Music Information Retrieval Conference*. Suzhou, China.
- [106] 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*. Paris, France, 225–232.
- [107] 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 *5th International Conference on Digital Libraries for Musicology*. Paris, France, 57–61.
- [108] Jan Hajič jr. and Pavel Pecina. 2017. Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression. *Computing Research Repository* abs/1708.01806 (2017).
- [109] Jan Hajič jr. 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.
- [110] Jan Hajič jr. and Pavel Pecina. 2017. In Search of a Dataset for Handwritten Optical Music Recognition: Introducing MUSCIMA++. *Computing Research Repository* abs/1703.04824 (2017), 1–16.
- [111] Jan Hajič jr. 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.
- [112] Andrew Hankinson. 2014. *Optical music recognition infrastructure for large-scale music document analysis*. Ph.D. Dissertation. McGill University.
- [113] 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 *13th International Society for Music Information Retrieval Conference*. 577–582.
- [114] Ali Hemmatifar and Ashish Krishna. 2018. *DeepPiano: A Deep Learning Approach to Translate Music Notation to English Alphabet*. Technical Report. Stanford University.
- [115] Władysław Homenda. 2001. Optical Music Recognition: the Case of Granular Computing. In *Granular Computing: An Emerging Paradigm*. Physica-Verlag HD, Heidelberg, 341–366.
- [116] Władysław Homenda. 2006. Automatic understanding of images: integrated syntactic and semantic analysis of music notation. In *International Joint Conference on Neural Network*. Vancouver, Canada, 3026–3033.
- [117] Władysław Homenda and Marcin Luckner. 2004. Automatic Recognition of Music Notation Using Neural Networks. In *International Conference on AI and Systems*. Divnormorkoye, Russia.
- [118] 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.
- [119] Krzysztof Jastrzębski. 2014. *OMR for sheet music digitization*. Master’s thesis. Politechnika Wroclawska.
- [120] Rong Jin. 2017. *Graph-Based Rhythm Interpretation in Optical Music Recognition*. Ph.D. Dissertation. Indiana University.
- [121] Rong Jin and Christopher Raphael. 2012. Interpreting Rhythm in Optical Music Recognition. In *13th International Society for Music Information Retrieval Conference*. Porto, Portugal, 151–156.
- [122] Linn Saxrud Johansen. 2009. *Optical Music Recognition*. Master’s thesis. University of Oslo.
- [123] 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.
- [124] Elyor Kodirov, Sejin Han, Guee-Sang Lee, and YoungChul Kim. 2014. Music with Harmony: Chord Separation and Recognition in Printed Music Score Images. In *8th International Conference on Ubiquitous Information Management and Communication*. Siem Reap, Cambodia, 1–8.
- [125] Worapan Kusakunniran, Attapol Prempnichnukul, Arthid Maneesutham, Kullachut Chocksawud, Suparus Tongsamui, and Kittikhun Thongkanhorn. 2014. Optical music recognition for traditional Thai sheet music. In *International Computer Science and Engineering Conference*. 157–162.
- [126] Wojciech Lesinski and Agnieszka Jastrzebska. 2015. Optical Music Recognition: Standard and Cost-Sensitive Learning with Imbalanced Data. In *IFIP International Conference on Computer Information Systems and Industrial Management*.

- 601–612.
- [127] Karen Lin and Tim Bell. 2000. Integrating Paper and Digital Music Information Systems. In *International Society for Music Information Retrieval*. 23–25.
 - [128] Xiaoxiang Liu. 2012. Note Symbol Recognition for Music Scores. In *Intelligent Information and Database Systems*. Berlin, Heidelberg, 263–273.
 - [129] Xiaoxiang Liu, Mi Zhou, and Peng Xu. 2015. A Robust Method for Musical Note Recognition. In *14th International Conference on Computer-Aided Design and Computer Graphics*. 212–213.
 - [130] Nawapon Luangnapa, Thongchai Silpavarangkura, Chakarida Nukoolkit, and Pornchai Mongkolnam. 2012. Optical Music Recognition on Android Platform. In *International Conference on Advances in Information Technology*. 106–115.
 - [131] Marcin Luckner. 2006. Recognition of Noised Patterns Using Non-Disruption Learning Set. In *6th International Conference on Intelligent Systems Design and Applications*. 557–562.
 - [132] Simone Marinai and Paolo Nesi. 1999. Projection Based Segmentation of Musical Sheets. In *5th International Conference on Document Analysis and Recognition*. 3–6.
 - [133] Cory McKay and Ichiro Fujinaga. 2007. Style-independent computer-assisted exploratory analysis of large music collections. *Journal of Interdisciplinary Music Studies* 1, 1 (2007), 63–85.
 - [134] John R. McPherson. 1999. *Page Turning — Score Automation for Musicians*. Technical Report. University of Canterbury, New Zealand.
 - [135] John R. McPherson. 2002. Introducing Feedback into an Optical Music Recognition System. In *3rd International Conference on Music Information Retrieval*. Paris, France.
 - [136] John R. McPherson. 2006. *Coordinating Knowledge To Improve Optical Music Recognition*. Ph.D. Dissertation. The University of Waikato.
 - [137] John R. McPherson and David Bainbridge. 2002. *Coordinating Knowledge Within an Optical Music Recognition System*. Technical Report. University of Waikato, Hamilton, New Zealand.
 - [138] 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.
 - [139] Yevgen Mexin, Aristotelis Hadjakos, Axel Berndt, Simon Waloschek, Anastasia Wawilow, and Gerd Szwillus. 2017. Tools for Annotating Musical Measures in Digital Music Editions. In *14th Sound and Music Computing Conference*. Espoo, Finland, 279–286.
 - [140] Du Min. 2011. Research on numbered musical notation recognition and performance in a intelligent system. In *International Conference on Business Management and Electronic Information*. 340–343.
 - [141] Hidetoshi Miyao and Minoru Maruyama. 2004. An online handwritten music score recognition system. In *17th International Conference on Pattern Recognition*.
 - [142] Igor dos Santos Montagner, Roberto Jr. Hirata, and Nina S. T. Hirata. 2014. Learning to remove staff lines from music score images. In *International Conference on Image Processing*. 2614–2618.
 - [143] Igor dos Santos Montagner, Roberto Jr. Hirata, and Nina S. T. Hirata. 2014. A Machine Learning based method for Staff Removal. In *22nd International Conference on Pattern Recognition*. 3162–3167.
 - [144] Diego Nehab. 2003. *Staff Line Detection by Skewed Projection*. Technical Report.
 - [145] Kia Ng. 2002. Music manuscript tracing. *Lecture Notes in Computer Science* 2390 (2002), 322–334.
 - [146] Kia Ng. 2004. Optical Music Analysis for Printed Music Score and Handwritten Music Manuscript. In *Visual Perception of Music Notation: On-Line and Off Line Recognition*. IGI Global, 108–127.
 - [147] Kia Ng and Roger Boyle. 1992. Segmentation of Music Primitives. In *BMVC92*. London, 472–480.
 - [148] Kia Ng, Roger Boyle, and David Cooper. 1995. Low- and high-level approaches to optical music score recognition. In *IEE Colloquium on Document Image Processing and Multimedia Environments*. 31–36.
 - [149] 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*. 215–218.
 - [150] Vo Quang Nhat and GueeSang Lee. 2014. Adaptive Line Fitting for Staff Detection in Handwritten Music Score Images. In *8th International Conference on Ubiquitous Information Management and Communication*. Siem Reap, Cambodia, 991–996.
 - [151] 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.
 - [152] Alexander Pacha. 2018. Self-learning Optical Music Recognition. In *Vienna Young Scientists Symposium*. 34–35. ISBN: 978-3-9504017-8-3.
 - [153] 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.
 - [154] Alexander Pacha, Kwon-Young Choi, Bertrand Couasnon, 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.
- [155] Alexander Pacha and Horst Eidenberger. 2017. Towards a Universal Music Symbol Classifier. In *14th International Conference on Document Analysis and Recognition*. Kyoto, Japan, 35–36.
 - [156] Alexander Pacha and Horst Eidenberger. 2017. Towards Self-Learning Optical Music Recognition. In *16th International Conference on Machine Learning and Applications*. 795–800.
 - [157] Alexander Pacha, Jan Hajič jr., and Jorge Calvo-Zaragoza. 2018. A Baseline for General Music Object Detection with Deep Learning. *Applied Sciences* 8, 9 (2018), 1488–1508.
 - [158] 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*. 415–426.
 - [159] Roberto M. Pinheiro Pereira, Caio E.F. Matos, Geraldo Jr. Braz, João D.S. de Almeida, and Anselmo C. de Paiva. 2016. A Deep Approach for Handwritten Musical Symbols Recognition. In *22nd Brazilian Symposium on Multimedia and the Web*. Teresina, Piauí; Brazil, 191–194.
 - [160] João Caldas Pinto, Pedro Vieira, and João M. Sousa. 2003. A new graph-like classification method applied to ancient handwritten musical symbols. *Document Analysis and Recognition* 6, 1 (2003), 10–22.
 - [161] Telmo Pinto, Ana Rebelo, Gilson Giraldo, and Jamie dos Santos Cardoso. 2010. *Content Aware Music Score Binarization*. Technical Report. Universidade do Porto, Portugal.
 - [162] Telmo Pinto, Ana Rebelo, Gilson Giraldo, and Jamie dos Santos Cardoso. 2011. Music Score Binarization Based on Domain Knowledge. In *Pattern Recognition and Image Analysis*. 700–708.
 - [163] Laurent Pugin, John Ashley Burgoyne, and Ichiro Fujinaga. 2007. Goal-directed Evaluation for the Improvement of Optical Music Recognition on Early Music Prints. In *7th ACM/IEEE-CS Joint Conference on Digital Libraries*. Vancouver, Canada, 303–304.
 - [164] Laurent Pugin, John Ashley Burgoyne, and Ichiro Fujinaga. 2007. MAP Adaptation to Improve Optical Music Recognition of Early Music Documents Using Hidden Markov Models. In *8th International Conference on Music Information Retrieval*. 513–516.
 - [165] 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.
 - [166] 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.
 - [167] Laurent Pugin, Jason Hockman, John Ashley Burgoyne, and Ichiro Fujinaga. 2008. Gamera versus Aruspix – Two Optical Music Recognition Approaches. In *9th International Conference on Music Information Retrieval*.
 - [168] Christopher Raphael. 2011. *Optical Music Recognition on the IMSLP*. Technical Report. Indiana University, Bloomington.
 - [169] Christopher Raphael and Rong Jin. 2013. Optical music recognition on the international music score library project. In *IS&T/SPIE Electronic Imaging*.
 - [170] Ana Rebelo. 2008. *New Methodologies Towards an Automatic Optical Recognition of Handwritten Musical Scores*. Master’s thesis. Universidade do Porto.
 - [171] Ana Rebelo. 2012. *Robust Optical Recognition of Handwritten Musical Scores based on Domain Knowledge*. Ph.D. Dissertation. University of Porto.
 - [172] Ana Rebelo, Artur Capela, Joaquim F. Pinto da Costa, Carlos Guedes, Eurico Carrapatoso, and Jamie dos Santos Cardoso. 2007. A Shortest Path Approach for Staff Line Detection. In *3rd International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution*. 79–85.
 - [173] Ana Rebelo, G. Capela, and Jamie dos Santos Cardoso. 2010. Optical recognition of music symbols. *International Journal on Document Analysis and Recognition* 13, 1 (2010), 19–31.
 - [174] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R.S. Marçal, 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.
 - [175] Ana Rebelo, André Marçal, and Jamie dos Santos Cardoso. 2013. Global constraints for syntactic consistency in OMR: an ongoing approach. In *International Conference on Image Analysis and Recognition*.
 - [176] Ana Rebelo, Filipe Paszkiewicz, Carlos Guedes, Andre R. S. Marçal, and Jamie dos Santos Cardoso. 2011. A Method for Music Symbols Extraction based on Musical Rules. In *Bridges 2011: Mathematics, Music, Art, Architecture, Culture*. 81–88.
 - [177] Ana Rebelo, Jakub Tkaczuk, Sousa Sousa, and Jamie dos Santos Cardoso. 2011. Metric Learning for Music Symbol Recognition. In *10th International Conference on Machine Learning and Applications and Workshops*. 106–111.
 - [178] K. Todd Reed and J. R. Parker. 1996. Automatic Computer Recognition of Printed Music. In *13th International Conference on Pattern Recognition*. 803–807.
 - [179] 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*. Kyoto, Japan, 27–28.

- [180] Dan Ringwalt, Roger Dannenberg, and Andrew Russell. 2015. Optical Music Recognition for Interactive Score Display. In *International Conference on New Interfaces for Musical Expression*. Baton Rouge, Louisiana, USA, 95–98.
- [181] Dan Ringwalt and Roger B. Dannenberg. 2015. Image Quality Estimation for Multi-Score OMR. In *16th International Society for Music Information Retrieval Conference*. 17–23.
- [182] 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.
- [183] Florence Rossant. 2002. A global method for music symbol recognition in typeset music sheets. *Pattern Recognition Letters* 23, 10 (2002), 1129–1141.
- [184] Florence Rossant and Isabelle Bloch. 2004. A fuzzy model for optical recognition of musical scores. *Fuzzy Sets and Systems* 141, 2 (2004), 165–201.
- [185] Florence Rossant and Isabelle Bloch. 2005. Optical music recognition based on a fuzzy modeling of symbol classes and music writing rules. In *IEEE International Conference on Image Processing 2005*. II–538.
- [186] Florence Rossant and Isabelle Bloch. 2006. 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 (2006), 081541.
- [187] Martin Roth. 1994. *An approach to recognition of printed music*. Technical Report. Swiss Federal Institute of Technology.
- [188] Alan Ruttenberg. 1991. *Optical Reading of Typeset Music*. Master’s thesis. Massachusetts Institute of Technology, Boston, MA.
- [189] W. Brent Seales and Arcot Rajasekar. 1995. Interpreting music manuscripts: A logic-based, object-oriented approach. In *Image Analysis Applications and Computer Graphics*. Berlin, Heidelberg, 181–188.
- [190] 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.
- [191] Rui Miguel Filipe da Silva. 2013. *Mobile framework for recognition of musical characters*. Master’s thesis. Universidade do Porto.
- [192] Maciej Smiatacz and Witold Malina. 2008. Matrix-based classifiers applied to recognition of musical notation symbols. In *1st International Conference on Information Technology*. 1–4.
- [193] Javier Sober-Mira, Jorge Calvo-Zaragoza, David Rizo, and José Manuel Iñesta. 2017. Multimodal Recognition for Music Document Transcription. In *10th International Workshop on Machine Learning and Music*. Barcelona, Spain.
- [194] 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).
- [195] Mu-Chun Su, Chee-Yuen Tew, and Hsin-Hua Chen. 2001. Musical symbol recognition using SOM-based fuzzy systems. In *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*. 2150–2153 vol.4.
- [196] Mariusz Szwoch. 2005. A Robust Detector for Distorted Music Staves. In *Computer Analysis of Images and Patterns*. Berlin, Heidelberg, 701–708.
- [197] Mariusz Szwoch. 2007. Guido: A Musical Score Recognition System. In *9th International Conference on Document Analysis and Recognition*. 809–813.
- [198] Mariusz Szwoch. 2008. Using MusicXML to Evaluate Accuracy of OMR Systems. In *International Conference on Theory and Application of Diagrams*. Herrsching, Germany, 419–422.
- [199] Paul Taele, Laura Barreto, and Tracy Hammond. 2015. Maestoso: An Intelligent Educational Sketching Tool for Learning Music Theory. In *27th Conference on Innovative Applications of Artificial Intelligence*. Austin, Texas, 3999–4005.
- [200] Lorenzo J. Tardón, Simone Sammartino, Isabel Barbancho, Verónica Gómez, and Antonio Oliver. 2009. Optical Music Recognition for Scores Written in White Mensural Notation. *EURASIP Journal on Image and Video Processing* 2009, 1 (2009), 843401.
- [201] Gabriel Taubman. 2005. *MusicHand : A Handwritten Music Recognition System*. Technical Report. Brown University.
- [202] 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*.
- [203] Fubito Toyama, Kenji Shoji, and Juichi Miyamichi. 2006. Symbol Recognition of Printed Piano Scores with Touching Symbols. In *18th International Conference on Pattern Recognition*. 480–483.
- [204] 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.
- [205] Lukas Tuggener, Ismail Elezi, Jürgen Schmidhuber, and Thilo Stadelmann. 2018. Deep Watershed Detector for Music Object Recognition. In *19th International Society for Music Information Retrieval Conference*. Paris, France, 271–278.

- [206] 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.
- [207] Pedro Vieira and João Caldas Pinto. 2001. Recognition of musical symbols in ancient manuscripts. In *International Conference on Image Processing*. 38–41 vol.3.
- [208] 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.
- [209] Gabriel Vigliensoni, Gregory Burlet, and Ichiro Fujinaga. 2013. Optical measure recognition in common music notation. In *14th International Society for Music Information Retrieval Conference*. Curitiba, Brazil.
- [210] 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.
- [211] 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).
- [212] Quang Nhat Vo, Tam Nguyen, Soo-Hyung Kim, Hyung-Jeong Yang, and Guee-Sang Lee. 2014. Distorted music score recognition without Staffline removal. In *22nd International Conference on Pattern Recognition*. 2956–2960.
- [213] Marc Vuilleumier Stückelberg and David Doermann. 1999. On musical score recognition using probabilistic reasoning. In *5th International Conference on Document Analysis and Recognition*. 115–118.
- [214] Marc Vuilleumier Stückelberg, Christian Pellegrini, and Mélanie Hilario. 1997. An architecture for musical score recognition using high-level domain knowledge. In *4th International Conference on Document Analysis and Recognition*. 813–818 vol.2.
- [215] Marc Vuilleumier Stückelberg, Christian Pellegrini, and Mélanie Hillario. 1997. *A preview of an architecture for musical score recognition*. Technical Report. University of Geneva.
- [216] Matthias Wallner. 2014. *A System for Optical Music Recognition and Audio Synthesis*. Master’s thesis. TU Wien.
- [217] Lee Ling Wei, Qussay A. Salih, and Ho Sooi Hock. 2008. Optical Tablature Recognition (OTR) system: Using Fourier Descriptors as a recognition tool. In *International Conference on Audio, Language and Image Processing*. 1532–1539.
- [218] Cuihong Wen, Ana Rebelo, Jing Zhang, and Jamie dos Santos Cardoso. 2014. Classification of optical music symbols based on combined neural network. In *International Conference on Mechatronics and Control*. 419–423.
- [219] Cuihong Wen, Ana Rebelo, Jing Zhang, and Jamie dos Santos Cardoso. 2015. A new optical music recognition system based on combined neural network. *Pattern Recognition Letters* 58 (2015), 1–7.
- [220] K. Wijaya and David Bainbridge. 1999. Staff line restoration. In *7th International Conference on Image Processing and its Applications*. 760–764.
- [221] Carl Witt. 2013. Optical Music Recognition Symbol Detection using Contour Traces.
- [222] Yang Yin-xian and Yang Ding-li. 2012. Staff Line Removal Algorithm Based on Trajectory Tracking and Topological Structure of Score. In *4th International Conference on Computer Modeling and Simulation*.
- [223] Emily H. Zhang. 2017. *An Efficient Score Alignment Algorithm and its Applications*. Master’s thesis. Massachusetts Institute of Technology.

Towards Self-Learning Optical Music Recognition

The paper “Towards Self-Learning Optical Music Recognition” [PE17b], published at the International Conference on Machine Learning and Applications 2017 in Cancun, Mexico, was the first publication of my research and contains the overall plan of my research. In particular, it discusses why a new paradigm is needed for OMR and what it could look like. Along with the discussion, two experiments are presented.

In the first one, a convolutional neural network was trained to distinguish images of music scores from images depicting something else, such as natural photographs, documents or tables. The goal was to see if a neural network was capable of learning the concept of “how music scores look like.” A real-world application emerged at the WoRMS 2018 when a librarian expressed her need for such a classifier to assist her in automatically finding music scores in millions of documents in her library. For training the network, a new dataset of 2000 images was collected by taking real photos of scores and other documents under various angles and lighting conditions. The results on that dataset were exceptional with nearly 100% accuracy, which means that the task is easily solvable with deep learning.

The second experiment reproduced a previously conducted study, trying to classify isolated, handwritten music symbols from the HOMUS [CZO14] dataset with a deep convolutional neural network. Previously reported results were already very good with 96% and 97% accuracy, so the state of the art could only be improved slightly to 98% accuracy, which is even better than the performance of humans on the same task (95% accuracy). Another interesting observation was made: the trained network was coping exceptionally well (97% accuracy) with superimposed staves that were artificially introduced into the images of isolated symbols. This indicates that the removal of staves might be superfluous when using convolutional neural networks.

Towards Self-Learning Optical Music Recognition

Alexander Pacha, Horst Eidenberger

Interactive Media Systems, TU Wien, Vienna, Austria

alexander.pacha@tuwien.ac.at, horst.eidenberger@tuwien.ac.at

Abstract—Optical Music Recognition (OMR) is a branch of artificial intelligence that aims at automatically recognizing and understanding the content of music scores in images. Several approaches and systems have been proposed that try to solve this problem by using expert knowledge and specialized algorithms that tend to fail at generalization to a broader set of scores, imperfect image scans or data of different formatting. In this paper we propose a new approach to solve OMR by investigating how humans read music scores and by imitating that behavior with machine learning. To demonstrate the power of this approach, we conduct two experiments that teach a machine to distinguish entire music sheets from arbitrary content through frame-by-frame classification and distinguishing between 32 classes of handwritten music symbols which can be a basis for object detection. Both tasks can be performed at high rates of confidence (>98%) which is comparable to the performance of humans on the same task.

I. INTRODUCTION

Music plays a central role in our cultural heritage with written music scores being an essential way of communicating the composer's intention to musicians that perform a piece of music. The music notation encodes the information into a graphical form that follows certain syntactic and semantic rules to encode pitch, rhythm, tempo, and articulation. Optical Music Recognition (OMR) tries to recognize and understand the notation and the contents of an image for a machine to be able to comprehend the music. Given a system that is able to translate an image into a machine-readable format, the applications are manifold, including preservation and digitization of hand-written manuscripts, supporting music education or accompanying musicians that practice their performance.

Although considerable research has been conducted and many systems have been developed [1] that reportedly perform well on the specific set of music scores for which they have been designed for, the robustness and extensibility of these systems is limited due to the underlying architecture and used algorithms that discard information and propagate errors from one step to the next, e.g. an error in the binarization which is often the first step of an OMR system might cause the symbol detection to detect notes where there are none. Many algorithms have been proposed to improve individual steps of this linear process, but to the best of our knowledge, there exists no system that is capable of automatically recognizing a large set of real-world data with satisfactory precision, good usability, and reasonably low

editing costs [2] of errors that were introduced during the process. Many people could benefit from digitizing a large body of music scores that is accessible and searchable [3]. As a result, there are ongoing projects to do so including SIMSSA¹ and OpenScore². To support such projects, we propose a new approach: rather than designing features and defining rules by hand, the system should learn to extract features and appropriate rules by itself (given a certain amount of supervision). Ideally, such a system is capable of transcribing music scores as accurately as humans.

II. RELATED WORK

OMR has been a subject of interest at least since 1966 [4], and received substantial attention by Bainbridge and Bell [5] who established a general framework for OMR that has been adopted by many researchers [1]. Since then, many researchers suggested entire OMR systems [6], [7] or proposed specialized algorithms for solving or improving sub-tasks such as binarization [8] or staff-line detection and removal [9], [10]. However, most of them use ad-hoc solutions based on expert knowledge that follow widely used practices that work best on datasets fulfilling certain prerequisites, e.g. detecting staff-lines with horizontal projections requires the scores to have straight staff-lines. Unfortunately, these systems tend to experience difficulties when confronted with images that deviate from the expected input format for which they were designed (e.g. if the staff-lines are curved due to the bonding of a textbook). Adding another preprocessing step or improving an algorithm can help to overcome one or the other limitation, but might not help a system to gain robustness beyond a certain level.

In the last few years, machine learning - and especially Deep Learning with Convolutional Neural Networks (CNNs) - received a lot of attention with results that surpass human-level performance on computer vision tasks such as image classification [11]. Wen et al. proposed a machine learning approach for symbol segmentation and symbol classification [12] in combination with a pre-defined ruleset. Calvo-Zaragoza et al. [13] classify music scores at pixel-level with CNNs into foreground, background, and staff-lines. Gallego et al. [14] use auto-encoders to remove staff lines and finally Pinheiro Pereira et al. [15] classify handwritten

¹<http://simssa.ca/>, last visited on Oct. 4, 2017

²<http://openscore.cc>, last visited on Oct. 4, 2017

music symbols from the HOMUS database [16] into 32 different categories with a precision of over 96%. Together, they provide strong evidence, that machine learning can successfully be applied to develop new types of OMR systems that are robust and extensible to a wide range of scores.

III. HOW HUMANS READ SCORES

We believe that an OMR system should be able to read and comprehend music scores with all their facets as well as humans. To the best of our knowledge, there exists no system that would come close to human performance [1]. As far as it is understood today, humans process visual scenes in a hierarchical way at three levels [17, p. 557]:

- 1) Low-level, where contrast, orientation, color, and movement are processed, primarily in the retina and ganglion cells [17, p. 600]
- 2) Intermediate-level, where the layout of the scene is processed by parsing the visual image into contours and surfaces of objects, segregating them from the background, involving the primary visual cortex [17, p. 619].
- 3) High-level, where actual object identification is performed, by matching surfaces and contours to known shapes from our memory (or more precisely to their neuronal representation) which happens primarily in the Inferior Temporal Cortex [17, p. 622]

By processing visual information in this hierarchical way, humans become very good at arriving at scene descriptions, grasping the gist of a scene. But reading music scores includes not only the visual perception of objects, but also relating objects to each other and to the context, a process where, unfortunately, today little is known about how humans perform this task, apart from certain brain regions that have been identified to be involved in this process [18], [17, p. 1353]. Note that for relating elements to each other and interpreting them correctly, it appears that humans use all information available. For music scores, this includes the staff-lines as the reference system, knowledge about the type of music, the notational system and also prior knowledge such as the probabilities of continuations within idioms [18] to resolve ambiguities if the available information is incomplete or doubtful. The expectancy can even replace a stimulus, making up for misprints as shown in the Goldovsky experiment [18] indicating that reading involves both top-down (or conceptually-driven) and bottom-up (or data-driven) processes.

Learning from the way humans read scores, binarizing the image as a first step or removing staff-lines seems to be counterproductive as it discards potentially relevant information. In summary, we conclude that OMR systems could benefit from operating directly on the input image (which is possible with Deep Learning), providing feedback

loops from later steps to refine earlier steps and consider information that might not have been used so far.

IV. HOW MACHINES READ SCORES

David Marr proposed a computational framework of vision that has three levels and to us appears very useful when discussing vision problems [19]:

- Computational theory, which specifies how a vision task can be solved in principle
- Algorithmic level, that gives precise details on how the theory can be implemented. In other words: What is the input and output and how to obtain the output given the input?
- Hardware for realizing the algorithm in a physical system (which is not necessarily computer hardware, but in our case it is)

Given this framework, we think that the computational theory of how humans or machines can read scores is correct and sound: detecting systems, staves and staff-lines and using them as structural guidance is a solid foundation; segmenting elements into smaller parts and constructing a relational mapping leads to a symbolic representation; finally, this symbolic representation can be interpreted in its context, according to syntactic and semantic rules that correspond to a particular notational language.

The algorithmic level, however, seems to be much harder to solve, possibly because the inherent complexity of the problem is often underestimated. Many proposed approaches can be seen as concept-driven because they use prior knowledge of the specific object, in this case, music sheets. We believe that a data-driven, Deep Learning approach is a viable alternative that should be investigated further. Therefore, we propose the following five questions as a model for bottom-up music processing that are specifically formulated to facilitate the development of such an Optical Music Recognition algorithm.

Can a machine mimic human behavior in ...

- Q-I distinguishing between music scores and arbitrary content?
- Q-II understanding the structure of music scores (staves, systems) and distinguish basic music symbols from each other and from the background?
- Q-III detecting and locating music symbols (notes, rests, ornaments, accidentals, bar-lines, articulations, ...) in the scores?
- Q-IV understanding the relation of objects to each other in music scores (the relation between a note and the staff-lines, an accidental to the left of a note which relates to that note, etc.)?
- Q-V fully understanding the syntax and semantics of music scores (inferring the actual note from relative position, shape and preceding symbols such as key signatures or accidentals)?

These five questions define our research program for the data-driven investigation of the OMR problem using deep networks. In our opinion, each question can be solved using an appropriate model and sufficient data. Note that the questions are of increasing complexity with Q-V representing a complete system that is capable of reading scores and fully understanding their content like humans. Q-I and Q-II can be implemented by using CNNs that operate directly on the raw input data. A promising approach for Q-III is to extend a classifier into an object detector by using region proposal networks [20]. As for questions Q-IV and Q-V, Recurrent Neural Networks (RNN) seem to be a good fit [21], as they can learn relationships in sequential data and already achieved remarkable results in Optical Character Recognition [22], a task that is comparable to OMR but in many regards simpler [5].

V. EXPERIMENTATION

To evaluate whether a data-driven approach is suitable for improving the state-of-the-art in OMR, two experiments were conducted that try to answer Q-I and partially Q-II. The first, to recognize music scores in an image and classify that image into one of the two categories: 'scores' or 'other'. The second, to classify isolated handwritten music symbols into 32 different classes, reproducing [15] in greater depth and improving their results significantly. For both experiments, a Convolutional Neural Network was trained using the popular Deep Learning frameworks Keras³ and Tensorflow⁴. The resulting models can then be used for inference on almost any machine including mobile devices (see Figure 1) to classify images from the live camera-feed and display a frame-by-frame classification.

A. Datasets

The dataset used for training, validation, and testing in the first experiment contains over 5500 images of which 2000 images contain scores and 3500 images contain something else (see Table I). The largest portion was obtained by using two publicly available datasets: the MUSCIMA database, which contains 1000 handwritten music scores [23] and the training database of the Pascal VOC Challenge 2006 which contains over 2600 images [24] that were considered part of the ground-truth for the category 'other'. Additionally, we created a new dataset containing 2000 imperfect but realistic images, by taking 1000 images depicting music scores and 1000 images of text documents and other objects with a smartphone camera. Preliminary testing showed that text documents were likely to be confused with scores, especially if they contain tables. Hence, a large portion of the additional images contains such documents in order to enable the network to learn the distinction. The complexity of the scores ranges from simple childrens' tunes to modern

³<http://keras.io/>, last visited on Oct. 4, 2017

⁴<http://www.tensorflow.org/>, last visited on Oct. 4, 2017

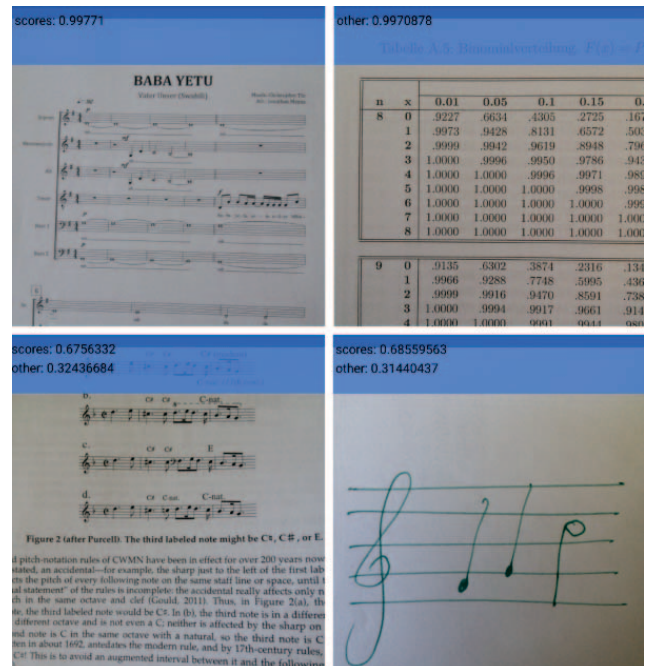


Figure 1: Screenshots of the Android application, classifying a sheet of music scores (left top) and a table with data (right top) with a certainty of 99%. When presented with images that contain scores and text (left bottom) or unusual forms (right bottom), certainty drops to approximately 70% but the system still classifies the image correctly.

orchestral scores, taken in various lighting conditions and from different angles.

The dataset for the second experiment is the Handwritten Online Musical Symbols (HOMUS) dataset [16] that contains 15200 samples of hand-written musical symbols, written by 100 different musicians⁵.

B. Architecture and Training

For both experiments, various network architectures were evaluated, including a VGG-like architecture [25] and residual networks [26].

The first experiment attempts to answer Q-I and uses color-images that are non-uniformly resized to 128x128 pixels for the first trial and 256x256 pixels for the second. For the second experiment that is targeted towards Q-II, black and white images are generated from the textual representation of strokes by connecting the points of each stroke. Since individual symbols vary drastically in size, while CNNs expect a fixed-size image as input, the following two approaches were evaluated:

⁵Note that the original dataset contained a few mistakes and artifacts that were reported to the authors and corrected before the training see <https://github.com/apacha/Homus> for details, last visited on Oct. 4, 2017

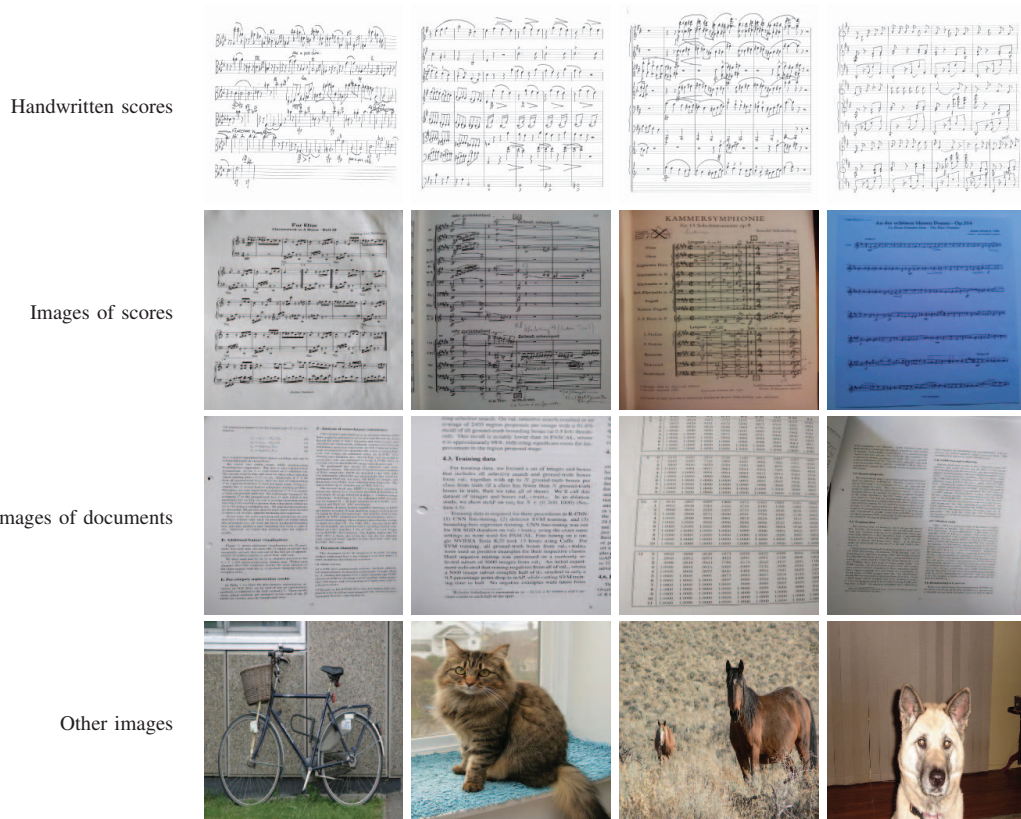


Table I: Sample images of the various categories, as they were shown to the classifier during training (non-uniformly resized). The upper two rows form the class 'scores' and the lower two rows the category 'other'.

- 1) Drawing the symbols in the center of a large enough canvas that fits most of them (e.g. 192x96 pixels, with only 23 out of 15200 symbols exceeding this size)
- 2) Drawing each symbol in a canvas that exactly fits its size and rescaling all symbols non-uniformly to a fixed size, e.g. 96x96 pixels

These particular sizes were empirically selected because they yielded the best results while allowing multiple down-scaling operations by a factor of two without interpolation.

Batch-normalization, early-stopping, weight-decay and dynamic learning-rate-reduction are used as regularization strategies to improve training speed and overall performance. Random-rotation by 10° and random-zoom of 20% are used as data-augmentation strategies to simulate the images being taken from slightly different points-of-view which leads to results that are robust to minor variations.

C. Evaluation

To evaluate each experiment, the respective dataset was split into three parts of which 80% are used as training data, 10% are used for validation during the training and for hyperparameter optimization and the final 10% are used

for evaluating the performance of the trained model on previously unseen data.

To obtain a baseline, a subset of the images was also shown to a number of people that were asked to perform the same classification task in a desktop application on a computer screen. The application did not allow for zooming and the users classified the images using the keyboard but were allowed to go back and revise their decisions without any time constraints.

1) *First Experiment:* Typical training took 30 epochs before early stopping the training to prevent overfitting. The trained model classified 98.5% of the images in the test set correctly on the 128x128 pixels condition and 100% on the 256x256 condition, meaning that this task appears almost trivial to the machine.

The more than 500 images from the test set were also shown to three different users, who were asked to manually classify them either as 'something that displays music scores' or 'something else'. The images were down-scaled to the same 128x128 pixels that correspond to approximately 3.5cm on a desktop screen. In total, they classified over 1500 images with an average precision of 96.49%. The main

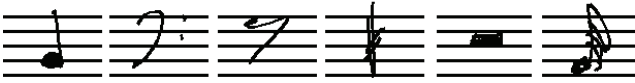


Figure 2: Superimposed staff-lines over isolated symbols to create meaningful context. Five parallel lines are drawn with an equal spacing of 14 pixels between each line [16]. From left to right: Quarter-Note, F-Clef, Eighth-Rest, Sharp, Whole-Half-Rest, Sixty-Four-Note

source of error was due to the very small images. Partially repeating the process with images of size 256x256 pixels, which corresponds to approximately 7cm on a desktop screen, showed that humans can perform this task without exceptional errors.

2) *Second Experiment*: The second experiment contains a wide range of conditions whose effects were investigated: image-size, stroke-thickness, superimposing staff-lines (see Figure 2) and of course the hyperparameters for the training of a deep neural network, including the network architecture, the used optimizer, and minibatch-size. A total of over 150 different hyperparameter-combinations were tested and documented. The following hyperparameters have empirically shown to work very well for this task:

- Monitoring the accuracy on the validation set after each epoch and reducing the learning-rate by a factor of 0.5 if it does not improve for 8 epochs. Similarly, the entire training was stopped if no improvement was observed for 20 epochs.
- Adam, Adadelta and Stochastic gradient descent (SGD) were evaluated as optimizers with Adadelta performing slightly better than Adam and much better than SGD.
- Evaluated minibatch-sizes included 16, 32 and 64 but the impact is rather small and in our opinion can be neglected.

The obtained results reach up to 98.02% accuracy on a test-set of 1520 images which is a significant improvement, compared to previously reported results of 97.26% [27] and 96.01% [15]. For images with undistorted symbols drawn on a fixed canvas (Section V-B, approach 1) a Res-Net architecture with 25 convolutional layers and about five million parameters performed best. Similar results were obtained with a VGG architecture for non-uniformly resized symbols (Section V-B, approach 2) that consists of 13 convolutional layers and about 8 million parameters.

The results of the best run, broken down by symbol class, are given in Table II and show that the network struggled most with notes and rests that are only discriminable by the number of flags, such as Thirty-Two- and Sixty-Four-Notes.

Five users were asked to perform the same task on a random sample of the dataset. In total, they classified 1520 images with an average precision of 95% and experiencing most difficulties in Quarter-Rests and Sixteenth-Rests that

Table II: The recall and precision per class for the best trained residual network in comparison to human performance on the same task.

Class name	Residual Network		Human test subjects	
	Recall	Precision	Recall	Precision
12-8-Time	1.00	1.00	1.00	0.97
2-2-Time	1.00	1.00	0.95	1.00
2-4-Time	0.97	0.95	1.00	0.98
3-4-Time	0.95	1.00	1.00	0.97
3-8-Time	1.00	1.00	1.00	1.00
4-4-Time	1.00	0.98	0.97	1.00
6-8-Time	1.00	1.00	1.00	1.00
9-8-Time	1.00	1.00	1.00	1.00
Barline	1.00	0.98	0.97	0.92
C-Clef	1.00	1.00	1.00	0.91
Common-Time	1.00	1.00	0.97	1.00
Cut-Time	0.95	1.00	0.98	0.98
Dot	0.97	1.00	1.00	1.00
Double-Sharp	1.00	1.00	0.97	1.00
Eighth-Note	0.99	0.95	0.92	0.98
Eighth-Rest	1.00	1.00	0.98	0.86
F-Clef	1.00	1.00	0.97	0.92
Flat	0.97	1.00	0.95	0.95
G-Clef	1.00	0.95	0.98	0.98
Half-Note	1.00	1.00	0.97	0.94
Natural	0.95	1.00	0.74	1.00
Quarter-Note	1.00	1.00	0.93	0.95
Quarter-Rest	0.95	0.95	0.89	0.82
Sharp	1.00	1.00	1.00	0.97
Sixteenth-Note	0.94	0.95	0.90	0.92
Sixteenth-Rest	0.97	0.97	0.76	0.81
Sixty-Four-Note	0.96	0.95	0.94	0.94
Sixty-Four-Rest	0.97	0.97	0.83	0.97
Thirty-Two-Note	0.91	0.95	0.99	0.91
Thirty-Two-Rest	0.97	0.95	0.91	0.89
Whole-Half-Rest	1.00	0.98	1.00	1.00
Whole-Note	1.00	0.98	1.00	0.98

both have manifestations that deviate from their printed counterparts dramatically or are simply ambiguous (see Figure 3).

Another very interesting detail was observed: When superimposing staff-lines as depicted in Figure 2, test-accuracy remains at high rates of up to 97.03%, indicating that the network can learn to ignore them almost entirely, thus providing evidence that staff-line removal might be omitted in future systems, as discussed in Section III.

VI. CONCLUSION

Given the results presented in Section V-C we conclude that Q-I can be answered with yes, showing that humans and machines can achieve similar results on the given dataset. Detecting music scores and distinguishing them from arbitrary content is a relatively easy problem compared to the entire challenge of Optical Music Recognition but what experiment 1 shows, is that machines can learn something as abstract as the concept of 'what music scores look like' by just providing enough data and using a Deep Learning approach. As for Q-II, we showed that a CNN can be trained to distinguish handwritten music symbols from each other at high rates of confidence, even with staff-lines being present.

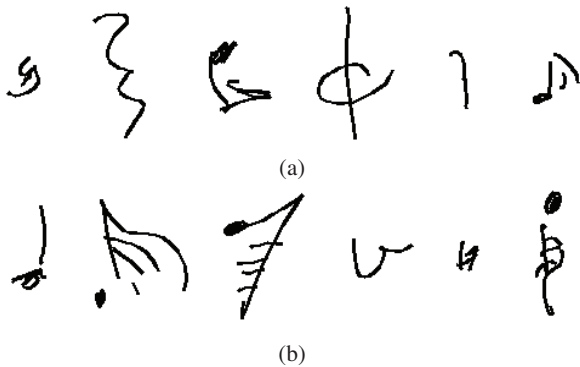


Figure 3: Examples of symbols from the test set that were misclassified by the machine (a) and by humans (b). Their intended classes from left top to right bottom: Sixteenth-Rest, 2-4-Time, Sixteenth-Note, Cut-Time, Quarter-Rest, Sixteenth-Note, Quarter-Note, Sixty-Four-Note, Sixty-Four-Rest, Quarter-Rest, Natural, and Sixty-Four-Note.

When combining these results with the work from [28] and [13] we conclude that Q-II can also be answered with yes.

VII. FUTURE WORK

To promote collaboration and reproducibility, all datasets, the entire source-code and the raw data from both experiments have been released on Github at <https://github.com/apache/MusicScoreClassifier> and <https://github.com/apache/MusicSymbolClassifier> under a liberal MIT-license. We are confident, that by following the described path, an OMR system can be created that is capable of not only classifying entire images but also recognizing the structure of the document, reliably detecting objects in the image and even understanding the relation of elements to each other without formulating explicit rules by only training appropriate models on a comprehensive dataset.

REFERENCES

- [1] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. Marcal, C. Guedes, and J. S. Cardoso, "Optical music recognition: state-of-the-art and open issues," *International Journal of Multimedia Information Retrieval*, vol. 1, no. 3, pp. 173–190, 2012.
- [2] P. Bellini, I. Bruno, and P. Nesi, "Assessing optical music recognition tools," *Computer Music Journal*, vol. 31, no. 1, pp. 68–93, 2007.
- [3] A. Laplante and I. Fujinaga, "Digitizing musical scores: Challenges and opportunities for libraries," in *Proceedings of the 3rd International workshop on Digital Libraries for Musicology*. ACM, 2016.
- [4] A. Rebelo, G. Capela, and J. S. Cardoso, "Optical recognition of music symbols," *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 13, no. 1, pp. 19–31, 2010.
- [5] 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.
- [6] L. Pugin, J. Hockman, J. A. Burgoyne, and I. Fujinaga, "Camera versus Aruspix – two optical music recognition approaches," in *ISMIR 2008–Session 3C–OMR, Alignment and Annotation*, 2008.
- [7] Y.-S. Chen, F.-S. Chen, and C.-H. Teng, "An optical music recognition system for skew or inverted musical scores," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 27, no. 07, 2013.
- [8] Q. N. Vo, S. H. Kim, H. J. Yang, and G. Lee, "An MRF model for binarization of music scores with complex background," *Pattern Recognition Letters*, vol. 69, pp. 88 – 95, 2016.
- [9] C. Dalitz, M. Droettboom, B. Pranzas, and I. Fujinaga, "A comparative study of staff removal algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, May 2008.
- [10] J. dos Santos Cardoso, A. Capela, A. Rebelo, C. Guedes, and J. P. da Costa, "Staff detection with stable paths," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, June 2009.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.
- [12] C. Wen, A. Rebelo, J. Zhang, and J. Cardoso, "A new optical music recognition system based on combined neural network," *Pattern Recognition Letters*, vol. 58, pp. 1 – 7, 2015.
- [13] J. Calvo-Zaragoza, G. Vighienoni, and I. Fujinaga, "Document analysis for music scores via machine learning," in *Proceedings of the 3rd International workshop on Digital Libraries for Musicology*. ACM, 2016, pp. 37–40.
- [14] A.-J. Gallego and J. Calvo-Zaragoza, "Staff-line removal with selectional auto-encoders," *Expert Systems with Applications*, vol. 89, 2017.
- [15] R. M. Pinheiro Pereira, C. E. Matos, G. Braz Junior, J. a. D. de Almeida, and A. C. de Paiva, "A deep approach for handwritten musical symbols recognition," in *Proceedings of the 22nd Brazilian Symposium on Multimedia and the Web*, ser. Webmedia '16. New York, NY, USA: ACM, 2016, pp. 191–194.
- [16] J. Calvo-Zaragoza and J. Oncina, "Recognition of pen-based music notation: The HOMUS dataset," in *2014 22nd International Conference on Pattern Recognition*, Aug 2014, pp. 3038–3043.
- [17] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. A. Siegelbaum, and A. J. Hudspeth, *Principles of neural science*. McGraw-hill New York, 2012, vol. 5.
- [18] J. Sloboda, *Exploring the musical mind*. Oxford University Press, 2005.
- [19] J. P. Frisby and J. V. Stone, *Seeing, Second Edition: The Computational Approach to Biological Vision*, 2nd ed. The MIT Press, 2010.
- [20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015.
- [21] B. Shi, X. Bai, and C. 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*, no. 99, 2016.
- [22] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for OCR in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [23] A. Fornés, A. Dutta, A. Gordo, and J. 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)*, vol. 15, no. 3, pp. 243–251, 2012.
- [24] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool, "The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results," <http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf>, 2006.
- [25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [27] J. Calvo-Zaragoza, A.-J. Gallego, and A. Pertusa, "Recognition of handwritten music symbols with convolutional neural codes," *Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition*, 2017.
- [28] 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.

Towards A Universal Music Symbol Classifier

The classification of symbols can be seen as a substantial part of detecting objects in an image because modern approaches tackle the problem usually in two stages: the first stage proposes regions of interest in the image and the second stage classifies these proposals accordingly. Therefore, it is useful to evaluate how well a classification task can be solved with Deep Learning, especially when using both handwritten and typeset music scores.

The paper “Towards a Universal Music Symbol Classifier” [PE17a], presented at the 12th IAPR International Workshop on Graphics Recognition 2017 in Kyoto, Japan, extended the previously conducted classification experiment [PE17b] to a much larger scale. While the HOMUS dataset already contains 15 000 samples, the dataset collected for this work was significantly larger with over 90 000 isolated musical symbols, collected from seven heterogeneous datasets and categorized into 79 classes (see Fig. 4.1 for a few samples). The resulting dataset contains more than 74 000 handwritten symbols and more than 16 000 symbols that were typeset, unfortunately with a heavy class-imbalance.

A convolutional neural network was trained to classify the symbols, and the results were auspicious with an error rate below 2%. More than 200 different hyperparameter combinations were evaluated, including a range of model architectures (inspired by VGG [SZ14] and ResNet [HZRS16]), image-sizes and class-balancing methods. Some combinations performed slightly better than others. However, it should be noted all tested combinations achieved error-rates between 2%-3%. As with all other experiments, the source code as well as the dataset and the results, were made publicly available online [Pac17a]. The exact details of the best-performing hyperparameter combination can be found there.

Towards a Universal Music Symbol Classifier

Alexander Pacha

Institute of Software Technology and Interactive Systems
 TU Wien
 Vienna, Austria
 alexander.pacha@tuwien.ac.at

Horst Eidenberger

Institute of Software Technology and Interactive Systems
 TU Wien
 Vienna, Austria
 horst.eidenberger@tuwien.ac.at

Abstract—Optical Music Recognition (OMR) aims to recognize and understand written music scores. With the help of Deep Learning, researchers were able to significantly improve the state-of-the-art in this research area. However, Deep Learning requires a substantial amount of annotated data for supervised training. Various datasets have been collected in the past, but without a common standard that defines data formats and terminology, combining them is a challenging task. In this paper we present our approach towards unifying multiple datasets into the largest currently available body of over 90000 musical symbols that belong to 79 classes, containing both handwritten and printed music symbols. A universal music symbol classifier, trained on such a dataset using Deep Learning, can achieve an accuracy that exceeds 98%.

Index Terms—Optical Music Recognition, dataset, classification, deep learning

I. INTRODUCTION

Optical Music Recognition (OMR) is an area of document analysis that aims to automatically understand written music scores [1]. Given an image of musical scores, an OMR system attempts to recognize the content and translate it into a machine-readable format such as MusicXML.

Music symbol classification is the subtask of OMR, where isolated symbols are assigned with class labels. In this work we present the first attempt of building a universal music symbol classifier, that is capable of classifying music symbols regardless of whether they are well printed or just handwritten. To build such a classifier, we propose a data-driven approach. Therefore, we developed tools that can unify multiple datasets into a single large dataset on which the universal music symbol classifier can be trained. In our test setup, we were unifying seven datasets into a collection of over 90000 samples, belonging to 79 classes.

II. DATASETS

For training a universal music symbol classifier, we tried to obtain the largest possible dataset that contains both printed and handwritten symbols. We did so by combining the following publicly available datasets:

- The Handwritten Online Musical Symbols (HOMUS) dataset [2] contains 15200 samples of isolated music symbols of 32 different classes.
- The MUSCIMA++ dataset [3] is the largest available dataset that contains detailed annotations for the underlying CVC-MUSCIMA dataset [4] of handwritten

music scores. More than 55000 complete symbols can be extracted from the music symbol primitives.

- The group of Rebelo et al. collected at least three different datasets [5], containing more than 15000 printed music symbols.
- The group of Fornés et al. collected a dataset of approximately 4100 images of handwritten symbols [6] depicting accidentals and clefs.
- The Audiveris OMR dataset¹ is a small dataset of four images of scores, along with annotations of 400 printed symbols in those images.
- The Printed Music Symbols dataset² is a new dataset created by us, in which we collected more than 200 printed music symbols of 36 different classes.
- The OpenOMR dataset³ is the last included dataset, that contains 500 printed music symbols of seven different classes.

The resulting dataset contains more than 74000 handwritten and more than 16000 printed symbols, with a substantial amount of inter-class variation.

III. UNITING THE DATASETS

A. Selecting classes and resolving ambiguities

Modern musical notation knows over 100 different symbol classes, with some classes being more present, like quarter notes or G clefs, whereas other classes are rarely used or just used for specific instruments like glissando or breath marks. Apart from selecting which classes to include into the dataset (ideally all of them), one has to deal with ambiguous class names. E.g. a quarter note may also be called quaver or a G clef is also referred to as Treble clef. To resolve this issue, a common terminology is selected and all aliases and variations are mapped to those names. The actual names are secondary, as long as the schema is clear. We follow the naming conventions of the HOMUS dataset and map all other names to their respective counterparts or to similar class names if they did not exist in the HOMUS dataset.

Besides class names, symbols themselves can be ambiguous too. Although having the same visual appearance, they might resolve to different semantics depending on the context (e.g.

¹<https://github.com/Audiveris/omr-dataset-tools>

²<https://github.com/apacha/PrintedMusicSymbolsDataset>

³<https://sourceforge.net/projects/openomr/>

tie vs. slur vs. phrase mark or staccato vs. dot of a dotted note). This ambiguity can not be resolved when working with isolated symbols outside of a context which determines the class. Therefore, all ambiguous symbols are placed in a unifying super-class such as *Dot* or *Whole-Half-Rest*.

B. Joining different levels of decomposition

Some creators of OMR systems suggest to decompose music symbols into individual primitives (e.g. note-heads, stems, numbers, letters) and combine them in a later stage, whereas others choose to work with entire sets of symbols that might consist of multiple smaller units (e.g. eighth-note, 2/4-time). This decision can be made for notes, accidentals, numbers, and letters. While some primitives form a class on their own (e.g. flat or sharp), others do not (e.g. stem, flag). Datasets with different conventions are at least partially incompatible. To integrate them nevertheless, a decision has to be made for each type, whether to exclude samples, use primitive symbol classes, preprocess primitives into compound symbols or enumerate all variants of combining primitives (e.g. 2/4-time, 3/4-time, 6/8-time, ...). To lose as little data as possible when joining the mentioned datasets, we propose a mixed approach: notes only appear as compound classes which require preprocessing in some cases, time signatures are enumerated and key signatures consisting of multiple flats or sharps are excluded with only their primitives being considered.

C. Tools for the automatic unification

We have built tools that are capable of automatically downloading all datasets and processing them. As images are the input for music symbol classification in OMR, all other representations have to be processed to obtain images: Our *HOMUS image generator* allows to render textual descriptions into symbol images and the *MUSCIMA++ image generator* creates symbol images from the underlying masks. The *image extractor* for the Audiveris OMR dataset takes annotations and extracts sub-images that contain individual symbols while the *image inverter* converts the white-on-black images from the Fornés dataset to black-on-white images. Finally, the entire dataset can be obtained and split into a training-, validation-, and test-set by calling a single script, the *training dataset provider*.

IV. BUILDING A UNIVERSAL MUSIC CLASSIFIER

A universal music classifier should be able to recognize all sorts of music symbols, regardless of whether they are handwritten or printed. Deep neural networks, especially convolutional neural networks offer a convenient, yet powerful way of solving computer vision tasks like the one at hand [7]. Therefore, we aim to build such a classifier by training a convolutional neural network on the presented dataset. Extending it to other notations is possible by adding a respective dataset. To the best of our knowledge, no such work has been done before.

V. DISCUSSION AND CONCLUSION

By providing tools for easily obtaining and merging multiple datasets, we believe that building a universal music symbol classifier can be reduced to the training of a suitable deep neural network. We evaluated this thesis by training various networks on the presented dataset and our preliminary results are promising with an error rate below 2% and over 98% precision and recall on an unseen test-set containing 10% of the data⁴. Our next step is to analyze the results and build a music symbol object detector on top of the classifier.

The united dataset is not perfect and currently suffers from being somewhat unbalanced with some classes having fewer than 10 instances while others have more than 1000, with the quarter note alone having almost 18000 samples. This poses a problem to any classifier that optimizes for accuracy on this dataset, as it might just learn the underlying distribution and simply ignore the classes with the fewest samples. Therefore, there is a need to gather more samples from classes with insufficient instances. Furthermore, our dataset has the following limitations:

- It currently contains modern notation symbols only.
- Some datasets have one dedicated class for non-recognizable symbols, including text fragments and dynamics. We incorporated that container class and store symbols in there, that currently do not fit our categorization as opposed to discarding them. In the next version, some symbols will be extracted from this container and put into their appropriate classes.
- Despite their prominence, beamed notes are currently underrepresented, because most underlying datasets do not contain any or decompose them into primitives that can not be joined easily.

To have the greatest possible impact, we publish all tools under a liberal MIT license along with a list of other OMR datasets at <https://apacha.github.io/OMR-Datasets/>.

REFERENCES

- [1] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. Marcal, C. Guedes, and J. S. Cardoso, "Optical music recognition: state-of-the-art and open issues," *International Journal of Multimedia Information Retrieval*, vol. 1, no. 3, pp. 173–190, 2012.
- [2] J. Calvo-Zaragoza and J. Oncina, "Recognition of pen-based music notation: The HOMUS dataset," in *2014 22nd International Conference on Pattern Recognition*, Aug 2014, pp. 3038–3043.
- [3] J. j. Hajič and P. Pecina, "In search of a dataset for handwritten optical music recognition: Introducing MUSCIMA++," *arXiv preprint arXiv:1703.04824*, vol. 1, pp. 1–16, 2017.
- [4] A. Fornés, A. Dutta, A. Gordo, and J. 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)*, vol. 15, no. 3, pp. 243–251, 2012.
- [5] A. Rebelo, G. Capela, and J. S. Cardoso, "Optical recognition of music symbols," *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 13, no. 1, pp. 19–31, 2010.
- [6] A. Fornés, J. Lladós, and G. Sánchez, *Old Handwritten Musical Symbol Classification by a Dynamic Time Warping Based Method*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 51–60.
- [7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, insight.

⁴<https://github.com/apacha/MusicSymbolClassifier>

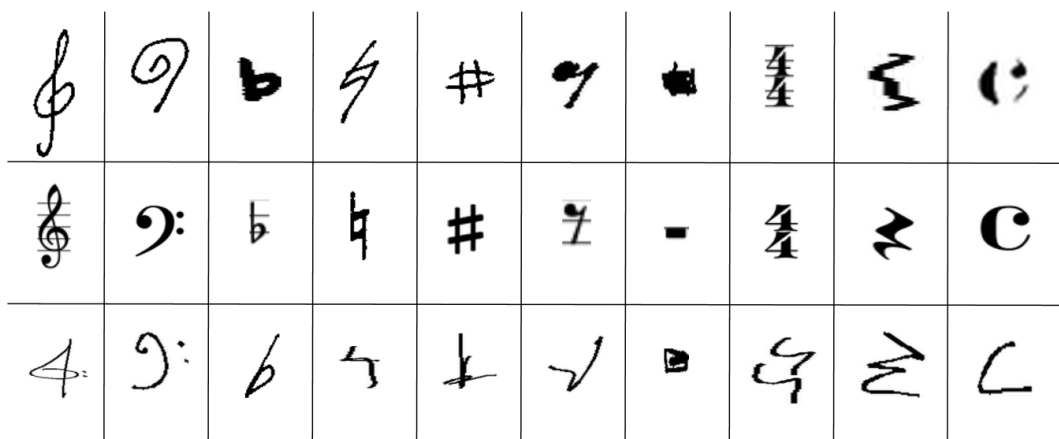
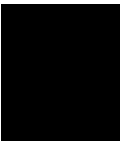


Figure 4.1: A small sample of music symbols that are part of the collected music symbols dataset. It depicts ten different classes of handwritten and typeset symbols in modern notation.



Music Object Detection

Being able to classify music symbols so well laid the foundations for building an object detector with a deep convolutional neural network—which in its simplest form is just a classifier over a sliding window. However, more advanced approaches for solving object detection with deep learning were used in subsequent experiments.

5.1 Handwritten Music Object Detection

The first experiments on trying to solve music object detection with deep learning were conducted for the paper “Handwritten Music Object Detection: Open Issues and Baseline Results,” [PCC⁺18] presented at the 13th IAPR International Workshop on Document Analysis System 2018 in Vienna, Austria. The main idea is to unify all steps of the music object detection into a single, learnable stage that can be solved by a deep convolutional neural network. The MUSCIMA++ dataset [HjP17] served as the data source because it provided a large body of handwritten music scores that were manually annotated. The full score images were preprocessed into smaller chunks to ease the detection, and sequentially fed into the network. The entire image was first cropped in such a way that each sub-image contains only one staff, and then horizontally cropped to maintain an aspect ratio of approximately 1:2 (see Fig. 5.1). It turned out later that while the cropping of images per staff makes sense, the additional horizontal cropping does not because many objects such as beams or slurs often cross boundaries and could, therefore, not be detected reliably.

Various models and hyperparameter-configurations were evaluated with a Faster R-CNN model performing best. The mean average precision (mAP), which is a commonly used metric for object detection tasks, yielded a value of over 80%, which is very high

and comparable to the best results for detecting objects in natural images¹. We also showed that the removal of staves has no significant impact on the detection performance, complying with previously found evidence.

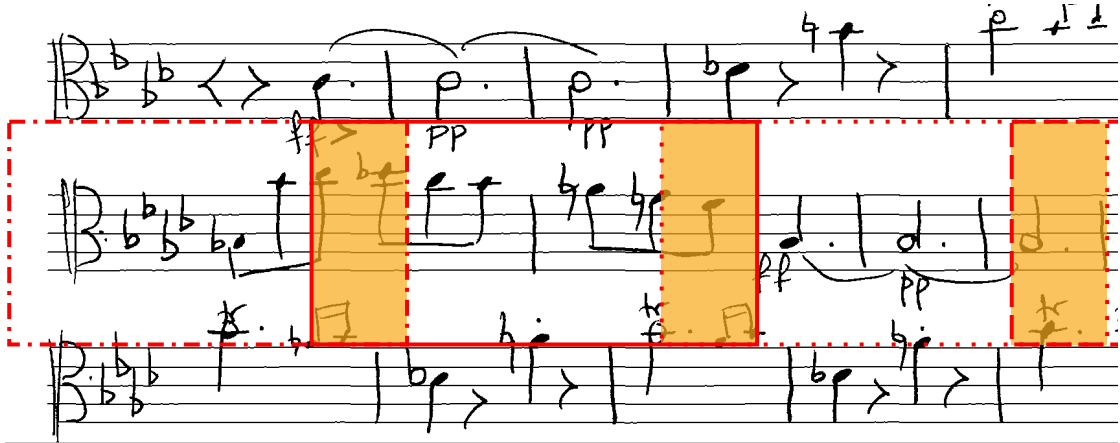


Figure 5.1: Illustration of the sliding window approach, used to crop music scores into sub-images (red boxes). Boxes overlap both vertically with the boxes above and below as well as with adjacent crops (orange).

¹Top entry of COCO Detection leaderboard [LPR⁺] as of 2018 was Megvii (Face++) with Average Precision of 0.53 for IoU=0.5:0.05:0.95 and 0.73 for IoU=0.5, submitted 05.10.2017.

Handwritten Music Object Detection: Open Issues and Baseline Results

Alexander Pacha, Horst Eidenberger
Institute of Visual Computing and Human-Centered
Technology, TU Wien, Vienna, Austria
{first name}. {last name}@tuwien.ac.at

Kwon-Young Choi, Bertrand Couïasnon,
Yann Ricquebourg
Univ Rennes, CNRS, IRISA, F-35000 Rennes, France
{first name}. {last name}@irisa.fr

Richard Zanibbi
Rochester Institute of Technology, Rochester, USA
rlaz@cs.rit.edu

Abstract—Optical Music Recognition (OMR) is the challenge of understanding the content of musical scores. Accurate detection of individual music objects is a critical step in processing musical documents because a failure at this stage corrupts any further processing. So far, all proposed methods were either limited to typeset music scores or were built to detect only a subset of the available classes of music symbols. In this work, we propose an end-to-end trainable object detector for music symbols that is capable of detecting almost the full vocabulary of modern music notation in handwritten music scores. By training deep convolutional neural networks on the recently released MUSCIMA++ dataset which has symbol-level annotations, we show that a machine learning approach can be used to accurately detect music objects with a mean average precision of over 80%.

Keywords—Optical Music Recognition; Object Detection; Handwritten Scores; Deep Learning

I. INTRODUCTION

Optical Music Recognition (OMR) attempts to understand the musical content of documents containing printed or handwritten music scores by recognizing the visual structure and the objects within a music sheet. Once, all objects are recognized, a semantic reconstruction step attempts to understand the relations of objects to each other and recover the musical semantics. With recent advances in computer vision, accelerated by the popularity of deep convolutional neural networks (CNN), OMR received a number of groundbreaking contributions that generate very accurate results for particular sub-problems, such as staff line removal [1] or symbol classification [2]. In this work, we investigate the challenge of music object detection which aims at accurately detecting music objects in music scores. Music objects can be both primitive glyphs (e.g. note-head, stem, beam) or compound symbols (e.g. notes, key-signatures, time-signatures) used in music notation. A music object detector takes an image and outputs the bounding-box and class-label for each found object. Traditionally, this was solved by first removing the staff lines, followed by symbol segmentation and classification [3] (see Figure 1).

In this work, we present the first attempt to establish a baseline for music object detection of handwritten scores with the full vocabulary of modern music notation. By following a machine learning approach and using an end-to-end trainable object detector on the recently published

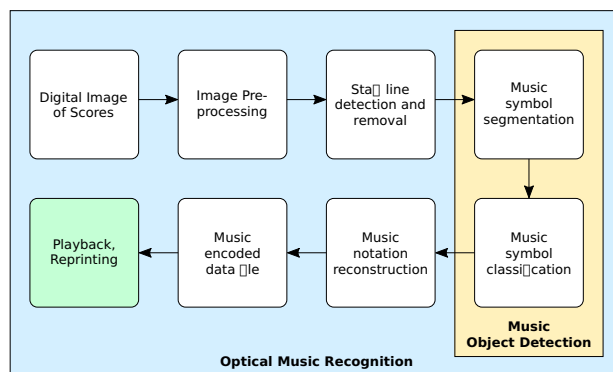


Figure 1. The traditional pipeline for Optical Music Recognition. Music object detection subsumes segmentation and classification of music symbols.

MUSCIMA++ dataset, we demonstrate how to build a generalizable and accurate music object detector and investigate the effects of various technical choices like the use of a particular detector or feature extractor.

II. RELATED WORK

Visual object detection is a very active field of research with remarkable results on detecting objects in natural images with a variety of active competitions. Many competing approaches have been proposed in the last few years such as Faster R-CNN [4], R-FCN [5] and Single shot detectors [6], [7]. While some optimize for accuracy, others strive for high-performance [8]. However, all of them share the fact, that they heavily make use of deep convolutional neural networks.

The traditional pipeline of segmenting and classifying symbols has been shown to work well on simple typeset music scores with a known music font [9]. But when considering low-quality images, complex scores or even handwritten ones [10], these systems tend to fail, mainly because errors propagate from one step to subsequent steps [11], e.g. a segmentation error could cause incorrectly detected objects. Initial attempts to overcome this limitation by directly detecting music objects with CNNs were made by Hajič and colleagues, who suggest an adaptation of Faster R-CNN with a custom region proposal mechanism based on the morphological skeleton to

accurately detect noteheads [12] and Choi and colleagues, who are able to detect accidentals in dense piano scores with high accuracy, given previously detected noteheads, that are being used as input-feature to the network [13]. However, both of them are limited to experimentations on a tiny subset of the full vocabulary used in modern music notation. Although both approaches can be extended to other classes, it remains an open question, whether a general purpose detector that can learn a large vocabulary is superior to multiple class-specific detectors.

A very interesting alternative to the traditional OMR pipeline is the attempt of solving OMR in a holistic fashion. The first notable attempt at doing so was by Pugin [14], who used Hidden Markov Models to read typographic prints of early music. More recently, the combination of using CNNs jointly with Recurrent Neural Networks to build an end-to-end trainable OMR system [15] was adapted and extended in [16] and [17]. Both train very similar models on a very large set of monophonic music scores containing a single staff per image. Although the reported results on the given datasets are very good, the two systems mentioned lastly, currently exhibit the following limitations:

- They operate only on very primitive, printed, monophonic scores. Extending their pipeline to more complex music scores with multiple voices requires a different formulation of the output data to at least include onset and offset of each note and not only the pitch and duration.
- By using pooling operations during the feature extraction, the network gains location invariance that conflicts with the interest of precise location information, which is needed to correctly infer the pitch of a note.
- By omitting the positional information of individual symbols and only considering the audible information of music symbols as output, such systems restrict themselves to replayability, as reprinting of music scores requires precise positional information [18].

While in theory semantic segmentation of the scores would go one step further and extract considerable more information – basically a classification of each pixel – two things should be noted: classifying pixels assumes that the class of each pixel is unique and mutually exclusive [19], an assumption that might not hold for overlapping symbols but can probably be ignored for practical applications; and most traditional systems that attempt to perform semantic reconstruction operate on detected objects, not on individual pixels, thus requiring a clustering step after the semantic segmentation. Therefore we argue, that detecting bounding boxes of musical objects directly is preferable for OMR.

III. THE CHALLENGE OF DETECTING MUSIC SYMBOLS

When comparing music object detection to detection of objects in natural scenes or optical character recognition, two unique challenges are worth noting: firstly, music

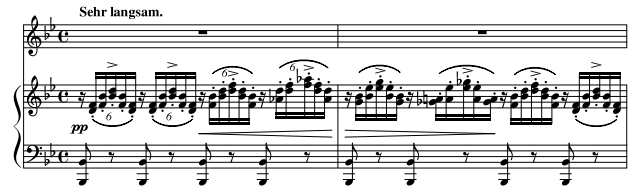


Figure 2. Beginning of Franz Schubert’s Ave Maria D. 839, with simplifications in the second bar that intentionally violate the syntactic rules of common music notation.

scores often have a very high density of objects with more than 1000 objects printed on a single page. Secondly, the relative position between a symbol and its staff lines is crucial. Already a tiny error along the y-axis may have a significant impact on recovering the correct pitch of a note.

The detection of music objects is of paramount importance to the overall OMR process because once all symbols were detected accurately, a set of rules can be applied to infer the semantics of the objects and perform music notation reconstruction as demonstrated by [20]. We also suggest that the point right after individual objects were detected and classified, is probably the best moment for putting the user into the loop, if that is intended. Fixing errors at this stage can be performed locally without dealing with complicated semantic rules or affecting neighboring symbols (changing the duration of a single note in a music notation program often entails side effects on other notes within the same of subsequent bars). Highlighting uncertain detections and suggesting likely alternatives could improve the usability and reduce editing costs even further.

Note that even with all symbols being correctly detected and classified, recovering the musical semantics still remains a very challenging problem, as demonstrated in Figure 2. Here, the second staff in the first bar contains a small 6 for each triplet, indicating that the first rest and the following five chords sum up to a quarter note. This small number is intentionally omitted in the second bar for simplification but would now result in an invalid meter if interpreted in isolation. Only with the preceding information and prior knowledge about common simplifications, a musician can interpret such scores correctly.

To be able to introduce such semantics into an OMR system, it is necessary to formalize and use musical notation knowledge. Rule-based systems can perform such formalization. For example, with the DMOS system [20] it has been possible to formalize the musical notation, graphically and syntactically, for full polyphonic scores, and produce a system which allows to assign notes to multiple voices and use the vertical alignments of synchronized notes in orchestral scores as well as the number of beats in a bar to detect and correct recognition errors. This grammatical formalization is built on terminals which correspond to the musical objects we propose to recognize with deep convolutional neural networks.

IV. BUILDING A MUSIC OBJECT DETECTOR

For building a robust and extensible music object detector, we propose a machine-learning approach with deep convolutional neural networks, which operate directly on the input image. This simplifies the OMR process to the following steps: preprocessing, music object detection, and semantic reconstruction. Steps such as removing the staff lines and segmenting symbols do not need to be addressed explicitly. Existing state-of-the-art object detectors such as Faster R-CNN or R-FCN were designed to detect objects in natural scenes and have been shown to work well on challenging datasets such as COCO [21] or ImageNet [22]. But applying them out-of-the-box on sheets of music can lead to a suboptimal performance, due to the dense nature of music scores with many tiny objects. Therefore, we suggest applying a certain amount of preprocessing to the data and tailor these detectors to perform well on the task at hand.

A. Dataset and Preprocessing Steps

For training a music object detector, we use the MUSCIMA++ dataset [23], as it contains 140 high-quality images with over 90000 symbol-level annotations, made by human annotators across 105 different classes of music symbols for the underlying CVC-MUSCIMA dataset [24]. The images have a high resolution of about 3500x2000 pixel, are binarized and optionally come with staff lines removed. For consistency, all white-on-black images are first inverted and then converted to RGB, as the evaluated implementations take colored images as input¹. To efficiently train an object detector on such images, the image size has to be reduced. We propose to crop the images in a context-sensitive way, by cutting images first vertically and then horizontally, such that each image contains exactly one staff and has a width-to-height-ratio of no more than 2:1, with about 15% horizontal overlap to adjacent slices (see Figure 3). Basically, each horizontal slice extends from the bottom of the staff above to the top of the staff below. This cropping can also be done by automatically detecting staves and then applying the same slicing rules leading to image crops that partially overlap both horizontally and vertically. For splitting the cropped images into a train and test set, we follow the recommendations from [23] to ensure that the test set contains scores of all complexities and that there is no overlap of writers between the training and the test set. We furthermore used 10% of the remaining training set for validation during the training. In total, we obtained 6181 samples, that were divided into a training, validation and test set, containing 4794, 533 and 854 images respectively.

One limitation of this approach is, that all objects significantly exceeding the size of such a cropped region, will not appear in the data, as only annotations that have an intersection-over-area of 0.8 or higher between the object and the cropped region are considered part of the ground truth.

¹The overhead created by this conversion is only minimal, as the duplicated information gets merged again in the first layer of the CNN.

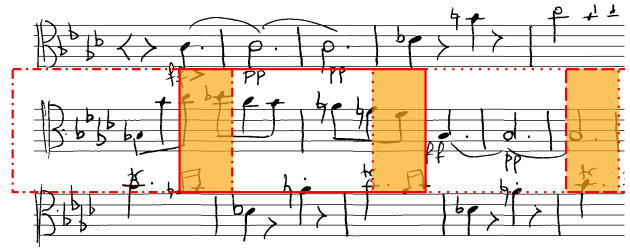


Figure 3. Illustration of the sliding window approach, used to crop music scores into meaningful subimages (red) with horizontally overlapping areas (orange) between adjacent crops.

As music objects, we consider the full vocabulary of all 105 classes contained in the MUSCIMA++ dataset, containing both primitives such as noteheads as well as compound objects such as key-signatures that consist of one or multiple accidentals.

B. Experimental Design

For evaluating our suggested approach, we conducted several experiments to study the performance of various object detectors and feature extractors, as well as the effects of staff line removal, transfer-learning and removing classes with rare symbols. Using the deep learning library TensorFlow², we adapted the work from [8] to detect music objects by training on the data described in Section IV-A. The entire source code, including training protocols and detailed instructions to reproduce our results, can be found at <http://github.com/apacha/MusicObjectDetector-TF>. We considered:

- the three meta-architectures Faster R-CNN, R-FCN, and SSD as object detectors. Faster R-CNN and R-FCN are both two-stage detectors with a region proposal network and a region classifier. The difference is that Faster R-CNN uses a sliding window for classification, whereas R-FCN uses position sensitive score maps and per-RoI pooling, which is more efficient at the cost of a slightly reduced precision. SSD is a generalized region proposal network for one stage detection on multiple feature maps
- ResNet50, Inception-ResNet-v2, MobileNet-v1 and Inception-v2 as feature extractors, explicitly excluding custom-made networks that cannot benefit from transfer-learning
- images with and without staff lines (based on the images provided along the CVC-MUSCIMA dataset)
- the full vocabulary of all 105 classes included in the MUSCIMA++ dataset, as well as a reduced set of only 71 classes, removing 34 classes that appear less than 50 times in the ground truth and are only of minor importance such as uncommon numerals and letters. Exceptions were only made for the classes double sharp and the numerals 5, 6, 7 and 8: although they appear less than 50 times in the dataset, we consider them essential to recover music semantics such as pitch and time signature.

²<https://www.tensorflow.org>, last seen 9th February 2018

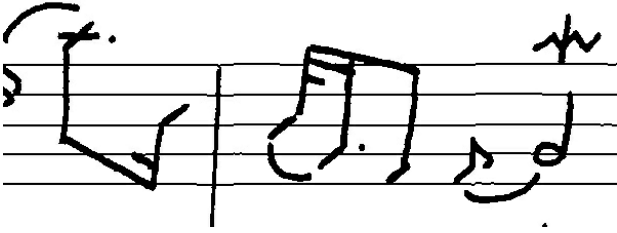


Figure 4. Typical sample of a cropped image that serves as input for the music object detector.

All of the above-mentioned object detectors have a certain set of hyperparameters that need to be fine-tuned for the particular dataset. For example, [7] shows that using statistical analysis to obtain a sensitive number of anchor boxes, anchor box sizes, and anchor box ratios can improve the results significantly compared to handpicked priors. When running similar analysis on the cropped images, we obtain the following characteristics: For a typical input image of 600 pixels width and 300 pixels height (see Figure 4), we found the average square box size is about 37 pixels with a standard deviation of 48 pixels. Note, that the dataset also contains extreme cases of small objects like dots with only a few pixels and large objects that spans hundreds of pixels. The mean ratio from width to height of boxes is 0.7 which means that the majority of boxes are higher than they are wide. Furthermore, cropped images that are to be fed to the detector contain 19 symbols on average, with a standard deviation of 11. Concluding the analysis, we decided to use a grid of 32x32 pixels with a stride of 8 pixels and aspect ratios of 0.06, 0.29, 0.48, and 2.2 with the scales 0.25, 0.5, 0.75, 1.0, 1.75, and 4.0 to reflect the wide range of object shapes in the dataset.

C. Evaluation and Results

Following the evaluation protocols of the Pascal VOC challenge [25], we report the mean average precision (mAP) for each completed training in Table I and the detailed average precision per class for the combination that yielded the best results in Table II. Figure 5 shows a typical detection within a single image.

We find that the best performing detector with regards to precision is the Faster R-CNN using the Inception-Resnet V2 feature extractor, pre-trained on the COCO dataset. This model produces a mAP of over 80%. The training on a GeForce GTX 1080 Ti takes approximately one day per configuration before results become stable. Validating 500 images takes about 2-4 minutes, so inference should take less than half a second per (cropped) image. When comparing the results of training on images with and without staff lines, the impact is no longer significant, supporting the claim of [14], that staff line removal might no longer be necessary. However, readers should also note that the staff lines in the CVC-MUSCIMA dataset are synthetic and do not experience the usual distortions that apply to scans or pictures of real music scores.

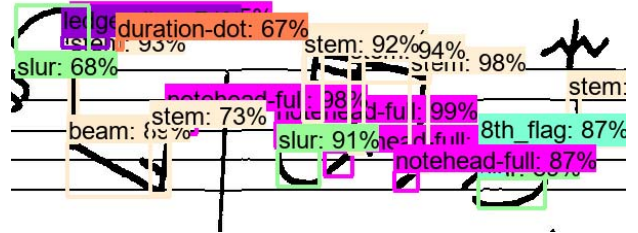


Figure 5. Typical detection results with most symbols recognized correctly.

Other detectors like the R-FCN or SSD produce good results as well, with a mAP of 75% and 71% respectively. Our results, therefore, comply with the findings of [8], where in particular the SSD model trades smaller accuracy for higher processing speed. Using pre-trained weights, instead of random initialization and the RMSprop optimizer as opposed to Stochastic Gradient Descent, improved the results significantly, speeded up convergence and was therefore used throughout the experiments. Modifying the set of classes by removing underrepresented classes as described in Section IV-B, boosted the mAP by up to 6% in some cases. Note, that Table II is missing six classes, that did not have any instances in the test set because they exceeded the size of the image crops and were thus discarded during the preprocessing.

V. DISCUSSION AND CONCLUSION

In this work, we show that state-of-the-art deep learning detectors like Faster R-CNN, R-FCN and SSD can produce accurate detection results on a wide range of music symbols. After optimizing different hyperparameters, we achieve a mAP of over 80%, which is a solid baseline.

However, there are still a couple of open issues, that need to be addressed in future work, like how to process a whole page of a score. In this work, we used a simple overlapping sliding window approach. This method, although simple to use, has many well-known downsides like the poor performance of processing empty images or cutting up large symbols as well as a non-trivial merging step that has to fuse information from multiple overlapping sections.

Another problem, specific to OMR, is the inherent imbalance of symbol classes: some symbols like noteheads are extremely frequent whereas others like double sharps are rare and often tied to a specific type of score. Having experimented with state-of-the-art deep learning object detectors, we found that classes do not interact with each other: simplifying the task by removing line-shaped classes did not improve the overall precision. There also seems to be a minimum threshold of about 20 samples per class, in order to be meaningful during the training. Currently, there is no guarantee, that the model does not overfit, but with recently published work like the RetinaNet and its focus loss [26] the effects of this class-imbalance could be mitigated to improve the training, especially on hard to detect classes.

Table I
DETAILED RESULTS FOR VARIOUS HYPERPARAMETER COMBINATIONS OF THE MUSIC OBJECT DETECTOR.

Meta-Architecture	Feature Extractor	Number of classes	Images have staff lines	Mean Average Precision on Test Set (%)	Weighted Mean Average Precision on Test Set (%)
Faster R-CNN	Inception-ResNet-v2	105	✓	81.56	94.22
Faster R-CNN	Inception-ResNet-v2	105	✗	81.23	94.56
Faster R-CNN	Inception-ResNet-v2	71	✓	85.12 [†]	94.68
Faster R-CNN	Inception-ResNet-v2	71	✗	87.80 [‡]	95.05
Faster R-CNN	ResNet50	105	✓	76.39	93.07
Faster R-CNN	ResNet50	105	✗	78.45	93.10
Faster R-CNN	ResNet50	71	✓	82.30	93.47
Faster R-CNN	ResNet50	71	✗	84.85	93.63
R-FCN	Inception-ResNet-v2	105	✓	69.75	89.12
R-FCN	Inception-ResNet-v2	105	✗	70.88	89.42
R-FCN	ResNet50	105	✓	75.53	92.59
R-FCN	ResNet50	105	✗	74.29	92.33
SSD	Inception-v2	105	✓	71.52	82.44
SSD	Inception-v2	105	✗	70.40	81.75
SSD	MobileNet-v1	105	✓	62.30	74.97
SSD	MobileNet-v1	105	✗	61.56	76.74

Although we used the test set, proposed by the MUS-CIMA++ authors, where writers in the test set do not appear in the training set, we are still not certain whether this system is truly writer independent or not. One way to confirm this would be to perform a cross-validation, where each writer in the dataset is evaluated independently.

Finally, we have shown that removing staff lines can be omitted for music object detection, when using CNNs. Future experiments that apply data-augmentation using noise models and deformed images, as proposed for the staff removal challenge [27], can give even more insights into the robustness of our approach.

ACKNOWLEDGMENT

The authors would like to thank all creators of public OMR datasets for collecting them and making them freely available to other researchers.

REFERENCES

- [1] A.-J. Gallego and J. Calvo-Zaragoza, "Staff-line removal with selectional auto-encoders," *Expert Systems with Applications*, vol. 89, pp. 138 – 148, 2017.
- [2] A. Pacha and H. Eidenberger, "Towards a universal music symbol classifier," in *Proceedings of the 12th IAPR International Workshop on Graphics Recognition*, IAPR TC10 (Technical Committee on Graphics Recognition). New York, USA: IEEE Computer Society, 2017, pp. 35–36.
- [3] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. Marcal, C. Guedes, and J. S. Cardoso, "Optical music recognition: state-of-the-art and open issues," *International Journal of Multimedia Information Retrieval*, vol. 1, no. 3, pp. 173–190, 2012.
- [4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
- [5] Y. Li, K. He, J. Sun *et al.*, "R-FCN: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 379–387.
- [6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.
- [7] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.
- [8] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," *CoRR*, vol. abs/1611.10012, 2016.
- [9] F. Rossant and I. Bloch, "Robust and adaptive OMR system including fuzzy modeling, fusion of musical rules, and possible error detection," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, p. 081541, 2006.
- [10] Arnau Baró, Pau Riba, and Alicia Fornés, "Towards the recognition of compound music notes in handwritten music scores," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. Institute of Electrical and Electronics Engineers Inc., 2016, pp. 465–470.
- [11] A. Pacha and H. Eidenberger, "Towards self-learning optical music recognition," in *Proceedings of the 16th IEEE International Conference On Machine Learning and Applications*, 2017, in print.
- [12] J. j. Hajič and P. Pecina, "Detecting noteheads in handwritten scores with convnets and bounding box regression," *arXiv preprint arXiv:1708.01806*, 2017.
- [13] K.-Y. Choi, B. Couasnon, Y. Ricquebourg, and R. Zanibbi, "Bootstrapping samples of accidentals in dense piano scores for cnn-based detection," in *Proceedings of the 12th IAPR International Workshop on Graphics Recognition*, IAPR TC10 (Technical Committee on Graphics Recognition). New York, USA: IEEE Computer Society, 2017.
- [14] L. Pugin, "Optical music recognition of early typographic prints using hidden markov models." in *ISMIR*, 2006, pp. 53–56.

Table II
DETAILED PRECISION RESULTS PER CLASS FOR THE BEST OBTAINED
MUSIC OBJECT DETECTOR ON THE REDUCED SET OF CLASSES (SEE
TABLE I, LINE 3[†] AND 4[‡]).

Class name	Total number of instances	Average precision on the test set (%)	
		with staff lines [†]	w/o staff lines [‡]
notehead-full	31084	99.85	99.64
stem	27108	98.82	98.71
ledger_line	14500	97.89	97.40
beam	8677	93.86	94.57
slur	3859	90.34	88.54
duration-dot	3195	95.12	94.21
thin_barline	3071	99.49	99.64
8th_flag	2744	93.46	93.37
measure_separator	2649	43.64	52.09
staccato-dot	2507	94.23	94.97
sharp	2420	99.42	99.46
notehead-empty	2385	99.31	99.11
flat	1467	96.97	97.98
natural	1427	96.90	97.61
dynamics_text	1374	85.25	87.12
8th_rest	1339	98.86	99.36
tie	1085	82.39	81.85
quarter_rest	1060	96.05	96.78
letter_p	1038	89.70	89.84
letter_f	1035	93.10	92.77
letter_e	926	82.12	85.29
letter_r	750	51.64	62.25
key_signature	697	79.31	77.80
letter_o	655	94.47	93.82
16th_flag	652	36.62	40.19
letter_s	649	71.89	74.30
grace-notehead-full	576	85.75	85.37
numeral_3	548	98.73	98.04
16th_rest	531	96.17	99.93
letter_t	513	92.10	94.42
other_text	508	83.99	89.30
letter_c	469	89.82	88.57
tuple	459	30.41	77.11
accent	421	99.08	95.75
g-clef	403	100.00	100.00
other-dot	402	94.40	95.19
repeat-dot	359	99.75	100.00
trill	315	100.00	99.74
letter_d	313	93.49	89.36
letter_m	293	74.19	74.43
f-clef	285	100.00	98.21
half_rest	241	95.53	91.16
time_signature	221	96.33	95.02
tenuto	216	88.45	74.79
letter_l	192	78.75	86.00
c-clef	190	97.68	98.68
whole_rest	183	90.73	84.66
letter_P	177	45.83	45.80
tempo_text	174	69.40	78.32
letter_i	171	66.48	81.08
letter_n	164	79.51	80.26
numeral_4	155	99.60	99.47
letter_a	134	90.36	83.81
multiple-note_tremolo	126	81.01	82.42
ornament(s)	123	85.22	83.90
letter_M	115	65.83	71.47
grace_strikethrough	110	98.14	97.96
letter_u	106	65.98	62.69
repeat	73	84.42	88.87
double_sharp	44	100.00	100.00
numeral_2	40	100.00	92.50
numeral_6	36	100.00	100.00
numeral_8	36	100.00	91.67
numeral_7	24	28.32	62.59
numeral_5	11	26.67	100.00

- [15] B. Shi, X. Bai, and C. 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*, vol. PP, no. 99, pp. 1–1, 2016.
- [16] J. Calvo-Zaragoza, J. J. Valero-Mas, and A. Pertusa, "End-to-end optical music recognition using neural networks," in *18th International Society for Music Information Retrieval Conference*, 2017.
- [17] E. van der Wel and K. Ullrich, "Optical music recognition with convolutional sequence-to-sequence models," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China*, 2017.
- [18] 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, pp. 497–506.
- [19] 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 Coruna: University of A Coruna*, 2017.
- [20] B. Coüasnon, "Dmos: a generic document recognition method, application to an automatic generator of musical scores, mathematical formulae and table structures recognition systems," in *Proceedings of Sixth International Conference on Document Analysis and Recognition*, 2001, pp. 215–220.
- [21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft COCO: Common Objects in Context*. Cham: Springer International Publishing, 2014, pp. 740–755.
- [22] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 248–255.
- [23] J. j. Hajič and P. Pecina, "The MUSCIMA++ dataset for handwritten optical music recognition." *Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition*, 2017.
- [24] A. Fornés, A. Dutta, A. Gordo, and J. 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)*, vol. 15, no. 3, pp. 243–251, 2012.
- [25] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.
- [26] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017.
- [27] A. Fornés, A. Dutta, A. Gordo, and J. Lladós, *The 2012 Music Scores Competitions: Staff Removal and Writer Identification*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 173–186.

5.2 General Music Object Detection

Given that multiple competing approaches for solving music object detection were developed simultaneously, Jorge Calvo-Zaragoza, Jan Hajič jr., and I joined forces to evaluate these approaches on a common ground—with the same datasets and the same evaluation protocol. Our efforts resulted in the paper “A Baseline for General Music Object Detection with Deep Learning,” published in a special issue of the applied sciences journal [PHjCZ18]. The three approaches Faster R-CNN, RetinaNet, and U-Net were evaluated on three different datasets: DeepScores [TES⁺18, ETPS18], MUSCIMA++ [HjP17] and Capitan [PCZ18]. The same evaluation protocol was used for all experiments, reporting the mAP and weighted mAP. In contrast to the previous paper [PCC⁺18], the more strict version of the metric was used, as defined by the COCO evaluation protocol [LMB⁺14]. This means that the average precision was not just taken at a single point where the Intersection over Union (IoU) is 50%, but averaged across a range of different values for the IoU, ranging from 50% to 95% in steps of 5%.

The results were mixed but particularly disappointing for the Faster R-CNN method on the MUSCIMA++ dataset, which only produced a mAP of 3.9%, as opposed to over 80% from previous research [PCC⁺18]. There are two reasons for this circumstance. First, the evaluation metric got much stricter. Second, we trained both the Faster R-CNN as well as the RetinaNet network on the entire image, instead of on sub-images. This increased the required memory so much that we were forced to reduce the image sizes, which in turn caused small objects to nearly disappear. On the Capitan dataset, which contained mostly bigger objects, Faster R-CNN and RetinaNet performed much better with 15.2% mAP and 14.5% mAP, respectively. In contrast, U-Nets are more independent from the input image size, since they contain convolutional filters only. Therefore, they could process the entire image in its full resolution. It also avoids some problems of Faster R-CNN which has a limited number of region proposals internally, which can become an issue in densely populated regions.

Unfortunately, the Deep Watershed Detector [TESS18] was not yet available when this paper was written. However, according to Lukas Tuggener, their results on the DeepScores dataset were approximately as good as ours.

Article

A Baseline for General Music Object Detection with Deep Learning

Alexander Pacha ^{1,*} , Jan Hajič, Jr. ² and Jorge Calvo-Zaragoza ³¹ Institute for Visual Computing and Human-Centered Technology, TU Wien, 1040 Wien, Austria² Institute of Formal and Applied Linguistics, Charles University, 116 36 Staré Město, Czech Republic; hajicj@ufal.mff.cuni.cz³ 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].

Hajić 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.

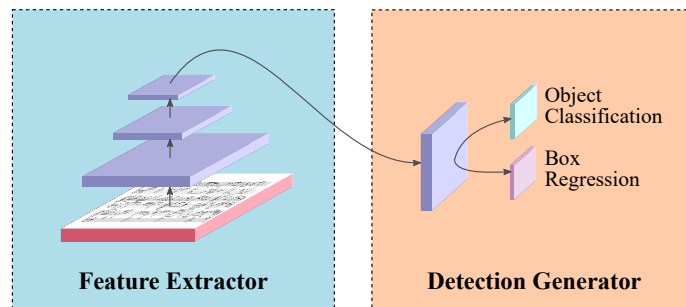
4.1.2. RetinaNet

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.

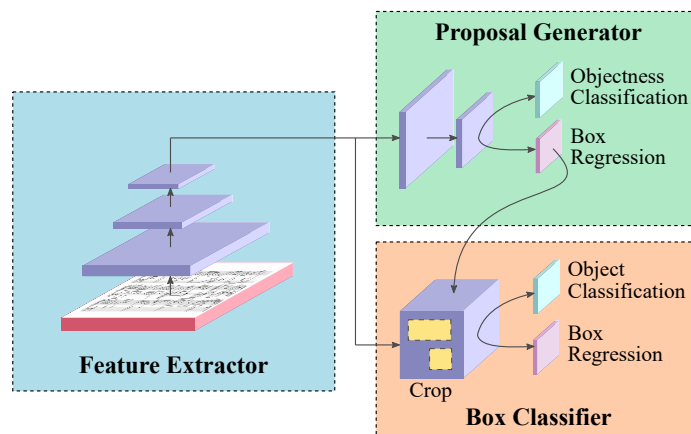
The RetinaNet [39] is an adaptation of a Residual Network [43] with lateral connections to create features on multiple scales [44]. Small convolutional subnetworks perform classification and bounding box regression on each output layer. RetinaNet was proposed along with the focal loss function, which 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],

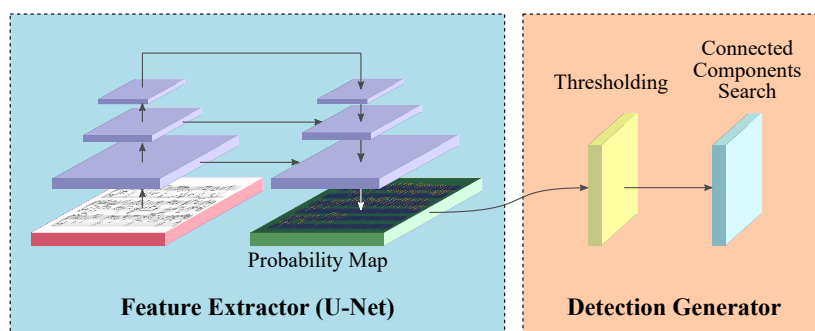
DenseNet121 [46], and a highly simplified version of the DenseNet. Various anchor dimension settings were also examined: the ResNet50 feature extractor performed best in preliminary experiments and was subsequently chosen. The negative overlap threshold was set to 40%, so every box with lower Intersection 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.

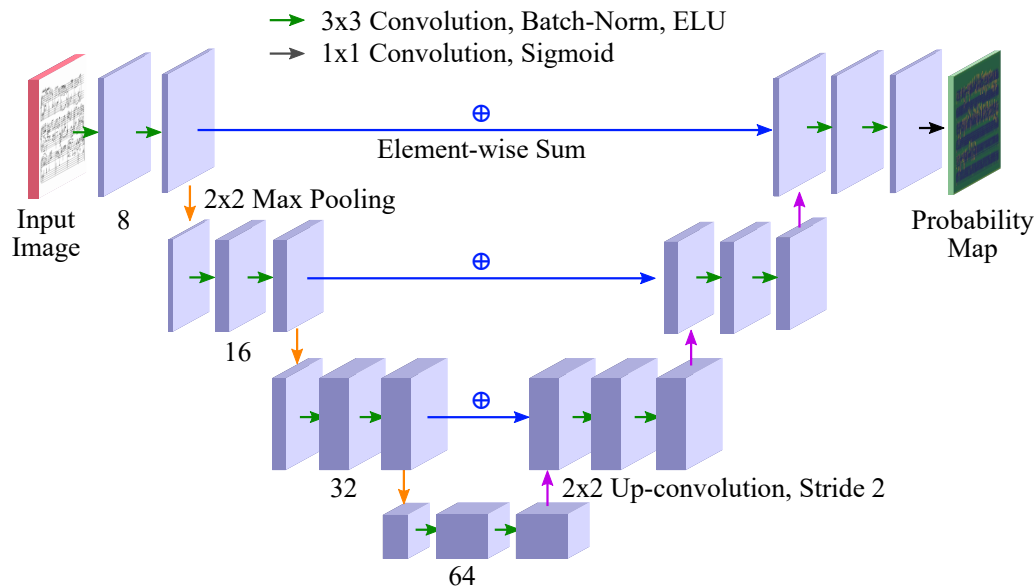


Figure 2. The U-Net architecture, with computation flowing left-to-right; the “hourglass” is unrolled downwards. Green arrows indicate 2D convolution with 3×3 kernels, downward orange arrows indicate 2×2 Max-Pooling, upward purple arrows indicate 2×2 up-convolution, and blue arrows indicate element-wise 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 1×1 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 256×512 window from each training image (corresponding to a 512×1024 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.

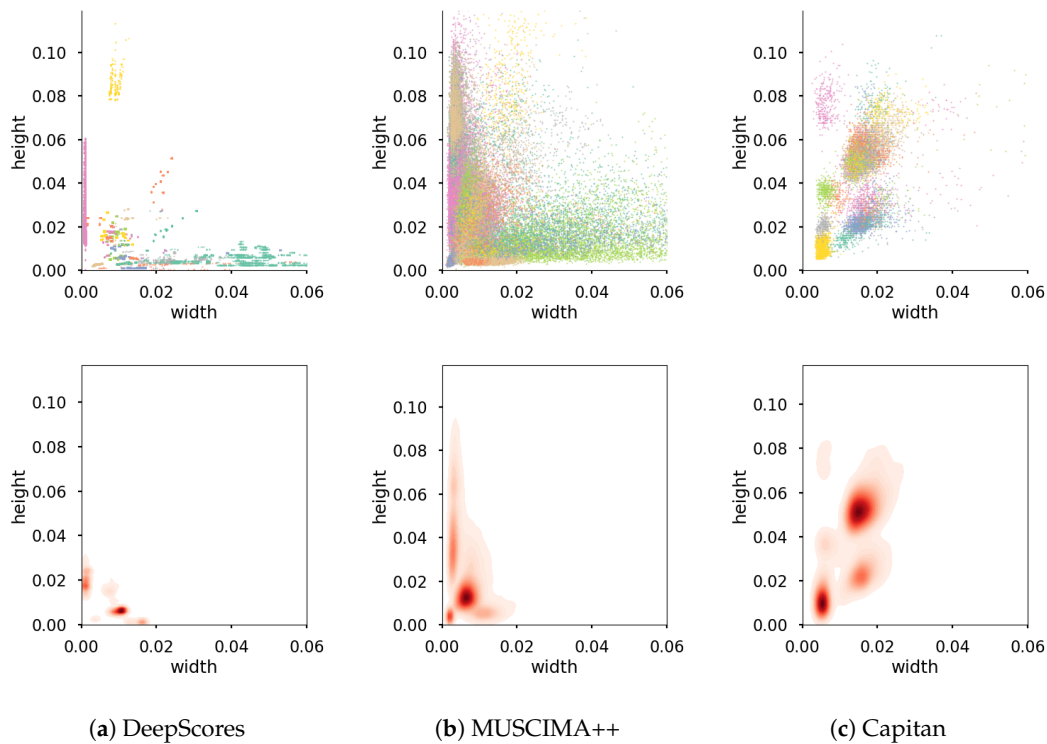


Figure 4. Scatter plot (**top**) row and density plot (**bottom**) row of the normalized object sizes for the considered corpora to illustrate the challenges of each dataset (best viewed in color). Each point in the top row depicts one instance from the dataset with the color encoding the respective 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_{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
Faster R-CNN	19.6	3.9	15.2	14.4	7.9	23.2
RetinaNet	9.8	7.7	14.5	1.9	4.9	34.9
U-Net	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

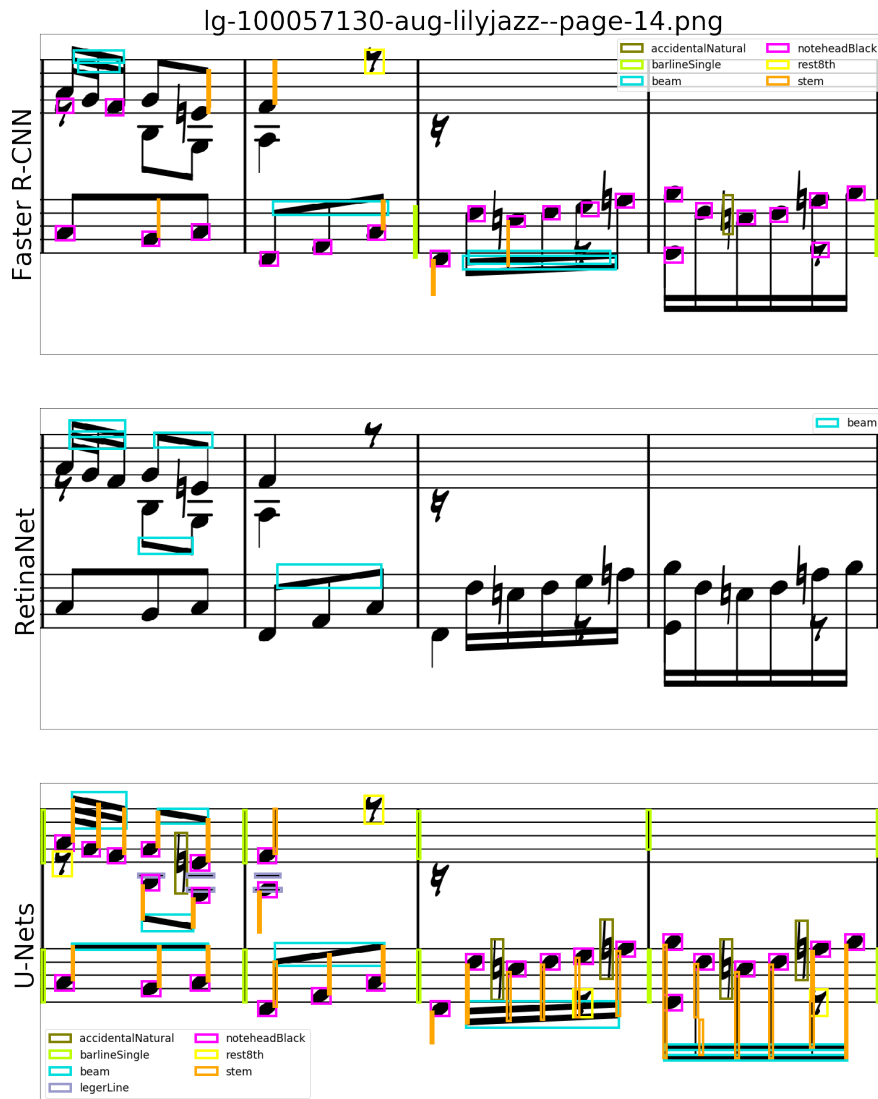


Figure 5. Detection sample on some selected classes from the DeepScores dataset. (top–bottom): Faster R-CNN, RetinaNet, and U-Nets detection results.

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

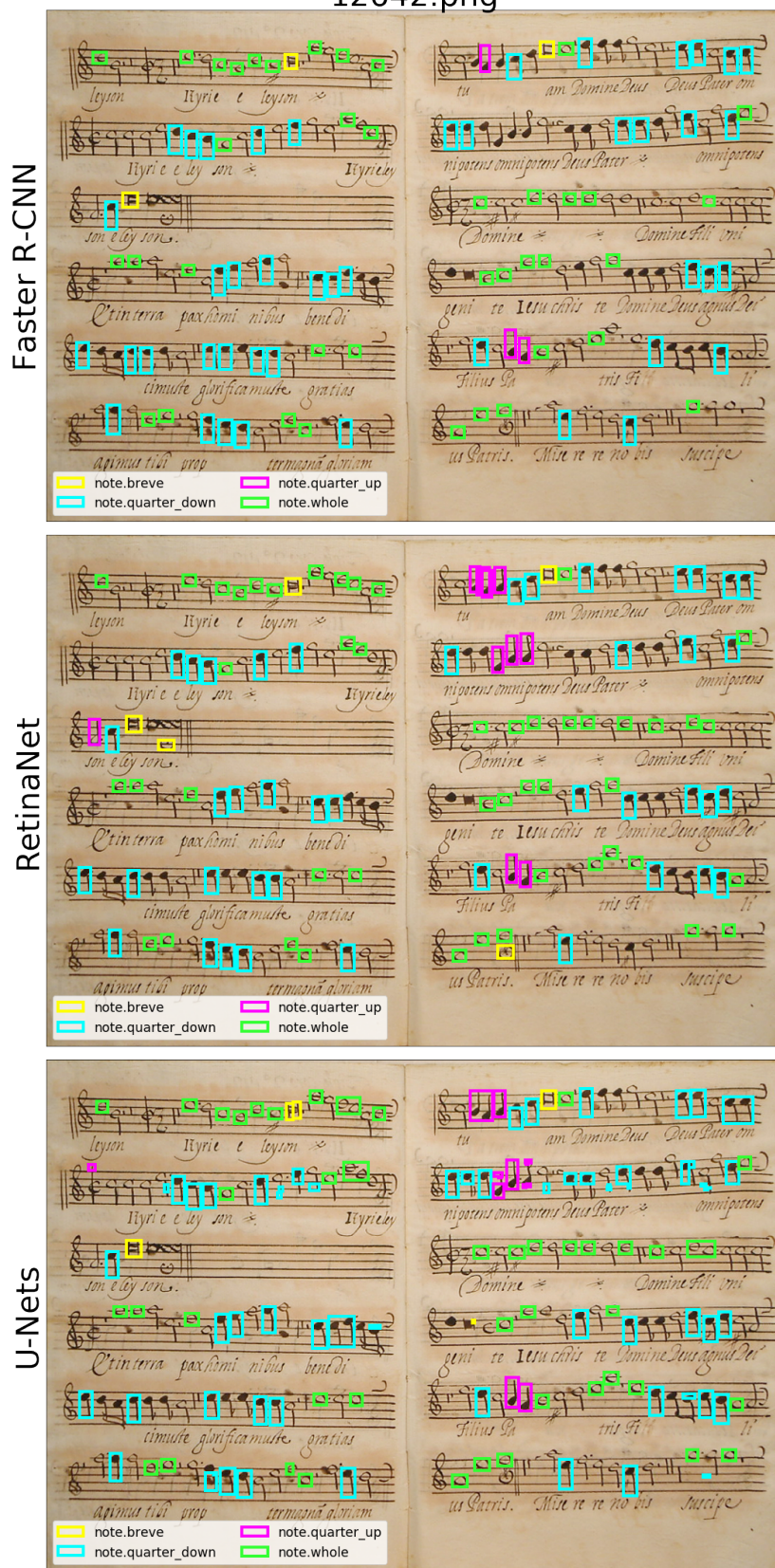


Figure 7. Detection sample on some selected classes from the Capitan dataset. (top–bottom): Faster R-CNN, RetinaNet, and U-Nets detection results.

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.; Coüason, 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/apache/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.; Coüason, 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. Coüasnon, 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.; Vigiensoni, 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/>).

Measure Detection and Structure Analysis

Knowing the structure of music scores can have significant benefits when performing music object detection, as could be seen in the two previously mentioned papers. When the images were cut into smaller images, containing only one staff, the results were extraordinary, whereas they were disappointing, when feeding a shrunken version of the whole image into the Faster R-CNN network.

But knowing the structure as preprocessing step for music object detection is only one reason, why it can make sense to analyze the layout and structure of music scores. In the paper “Identification and Cross-Document Alignment of Measures in Music Score Images,” Simon Waloschek, Aristotelis Hadjakos, and I worked on the structural analysis for a completely different reason [WHP19].

When creating critical editions of musical works, musicologists regularly compare multiple sources of the same musical piece. For allowing them to navigate between them efficiently, cross-source navigation is required which is aware of the musical content. Traditionally, measures were annotated by hand and then related to each other. In this paper, we trained a deep convolutional neural network, similar to the ones used for music object detection, to detect musical measures on a large, diverse body of over 8000 music scores, containing both handwritten and typeset scores. The interesting challenge is that musical measure can span across multiple staves and requires a certain amount of understanding to know how individual measures are joined into a system. Luckily, the trained object detectors were capable of learning these things very well and the results look very promising.

After having detected the individual measures, they need to be aligned across multiple scores for navigating between them. To this end, a second convolutional neural network was trained to compute the similarity between two measures to determine if they contain

the same music and should, therefore, be linked. Sequences are matched using Dynamic Time Warping.

My contribution to this work was limited to the first challenge: detecting measures. Simon Waloschek provided me with the body of manually annotated music scores and I was in charge of training and optimizing a convolutional neural network that is capable of solving this task without human intervention. This part of the work is publicly available on Github [Pac19b].

This paper has been accepted for the 20th International Society for Music Information Retrieval Conference 2019 in Delft, The Netherlands.

IDENTIFICATION AND CROSS-DOCUMENT ALIGNMENT OF MEASURES IN MUSIC SCORE IMAGES

Simon Waloschek, Aristotelis Hadjakos

Center of Music and Film Informatics
Detmold University of Music, Germany
{s.waloschek, a.hadjakos}@cemfi.de

Alexander Pacha

Institute of Information Systems Engineering
TU Wien, Austria
alexander.pacha@tuwien.ac.at

ABSTRACT

In the course of editing musical works, musicologists regularly compare multiple sources of the same musical piece, such as composers' autographs, handwritten copies, and various prints. For efficient comparison, cross-source navigation is essential, enabling to quickly jump back and forth between multiple sources without losing the current musical position. In practice, measures are first annotated by hand in the individual source images and then related to each other. Our approach automates this time-consuming and error-prone process with the help of deep learning. For this purpose, we train a neural network that automatically finds bounding boxes of all measures in images. A second network is trained to compute the similarity between two measures to determine if they have the same musical content and should, therefore, be linked for navigation. Sequences of outputs from the second network are matched using Dynamic Time Warping to provide the final proposal of measure relationships, so-called *concordances*. In addition to cross-source navigation, the results can be used to spot structural differences across the sources which are essential for editorial work, so that musicologists can focus more on analytical tasks.

1. INTRODUCTION

Modern musical editions are the result of a long musicological process. From the composer's manuscript to the printed music book, a musical work usually undergoes a large number of iterations and minor corrections, occasionally even substantial changes, such as striking or reworking complete parts [1]. Many of these changes are either unintentional—e.g., errors in handwritten copies, typographical errors by publishers—or generally not documented in a transparent manner. Musicologists, therefore, work on this genesis when editing a work and try to record the chronological order and causalities in their edition creation process.

The first step in this process is, therefore, the screening of the source material to identify differences between the various sources of a work. To facilitate this process, links are created between the sources so that editors can quickly switch back and forth between them. Adequate granularity of these links are usually musical measures, a feasible compromise between annotation effort and accuracy [29]. Currently, the measures of all sources are manually annotated with bounding boxes and related to each other in a very time-consuming and error-prone way.

We have automated this multi-stage process by first recognizing and sorting measures in score images (both handwritten and typeset) and then linking them according to their musical content. For this purpose, deep learning was used to develop a distance metric in an end-to-end fashion without an intermediate representation. The results can be further processed using classic alignment algorithms from the MIR community such as *Dynamic Time Warping* (DTW). While DTW-based approaches have achieved sufficient quality for practical use, audio-to-score alignment is still an active field of research [31]. Promising approaches for the synchronization of scans and sound recordings [5,6] are currently limited to monophonic and piano music and have not yet achieved sufficient accuracy for most real-world scenarios. With the contribution of this paper, we decrease a potential gap in the "audio – symbolic score – image" triangle and offer a new way for measure-accurate alignment across modal boundaries.

2. RELATED WORK

Detecting measures can be seen as a preprocessing step in Optical Music Recognition (OMR). Therefore, it was rarely singled out as a dedicated task. While Pedersoli and Tzanetakis perform document segmentation, they only distinguish between music scores and text blocks [22]. The only research we know of, that specifically addresses the automatic extraction of measures is by Vigliensoni et al. [30]. In their work, they attempt to extract measures with a traditional computer vision approach by heuristically finding all bar lines and then joining them into measures. Their approach requires human intervention for each page and straight bar lines to work well.

For retrieval of sixteenth-century musical texts, Crawford et al. [4] have recently proposed a two-step procedure. They run an OMR algorithm to obtain an intermedi-



ate format, followed by a second step that uses n-grams and minimal absent words (MAWs) to find duplicates, related texts, or parts that have the same musical material. Neural networks make such intermediate formats partly obsolete and allow for learning bimodal embeddings end-to-end as shown by Dorfer et al. [5, 6], who correlate the scanned music score with a sound recording. For this purpose, synchronization was considered either a reinforcement learning problem [6] or a metric learning problem [5]. In the metric learning approach, Dorfer et al. use the *pairwise ranking loss*—also known as *triplet loss* [26]—that draws triplets from a dataset consisting of an anchor, a positive example (picture fits the audio) and a negative example (picture does not fit the audio). This loss function creates an embedding, where images and audio with the same content appear close together, while non-matching images and audio are placed relatively far apart. Their approach has successfully been used before in other application domains, such as facial recognition [26]. We resort to a similar cost function for metric learning (see section 4.2).

As the basis for our detection, we use a convolutional neural network (CNN). While CNNs are currently an active field of research for OMR, the most influential approaches come from the research area of computer vision. They are used for many tasks, including image recognition, semantic segmentation, object detection, and instance segmentation. R-CNN [9] performs object detection by analyzing a large number of heuristically generated region proposals that are classified into background or one of the classes of interest. Additionally, the bounding box is refined with regression. R-CNN uses a CNN that extracts features for object detection. These features are used in a downstream SVM for classification and regression. Faster R-CNN [23] improves the process by incorporating both the region proposal step as well as the classification and regression into the architecture of the neural network.

CNN-based computer vision approaches are largely transferable to OMR and actively used for Music Information Retrieval: Gallego and Calvo-Zaragoza are using auto-encoders to remove staff lines [8]. Pacha et al. compare various CNN-based approaches for detecting music symbols in scores [21]. CNNs can also be used for semantic segmentation for staff-line removal, music and text separation as well as for layout analysis as shown by Calvo-Zaragoza et al. [3]. Using U-Nets [25], Hajic et al. do semantical segmentation of handwritten music [10]. Pacha and Calvo-Zaragoza recognize musical objects in mensural notation using region-based CNNs [20]. By learning energy levels that are used as inputs to a watershed algorithm, Tuggener et al. recognize music symbols [28]. In addition to the energy levels, the network also predicts class labels and bounding boxes. And finally, Calvo-Zaragoza and Rizo use convolutional recurrent neural networks trained with a Connectionist Temporal Classification (CTC) loss to recognize musical symbols in monophonic music scores [2]. To simulate non-ideal image conditions, they artificially distort the images.

3. DATA & ANNOTATIONS

The success of Deep Learning approaches largely depends on the amount and diversity of data used during training. Since no dataset of sufficient size was available for measure recognition or the concordance task, we created a large dataset ourselves in cooperation with musicologists and professional musicians.

Our dataset contains measure annotations that were created manually by musicologists for digital music editions. In most cases, the image sources are high-resolution scans of facsimiles, occasionally supplemented by early music prints and PDFs exported directly from music engraving software. Due to an imbalance between handwritten and typeset scores, we additionally obtained scores from the *IMSLP/Petrucci Music Library* while paying attention to varying image quality, the used engraving mechanism as well as diverse musical content. We complemented our collection with 140 pages from the MUSCIMA++ dataset¹ [7, 11].

Our data collection has a total of 8 251 pages with 81 124 annotated measures. The distribution according to engraving type and the number of systems per page is given in Table 1. One category is particularly over-represented: handwritten music scores with just one system per page because of a large quantity of full orchestral scores from operas by Carl Maria von Weber. Pages with zero systems include book covers, text pages, and prefaces.

Systems per page	Pages per engraving type	
	Handwritten	Typeset
0	413	113
1	5627	932
2	175	553
3	122	175
4 or more	102	39
Total pages	6439	1812

Table 1. Overall distribution of the dataset used.

The accuracy of the measure annotations varies. Since the exact boundaries are not relevant for musicologists they were recorded only roughly. That is why many bounding boxes contain small overlaps with adjacent measures, as shown in Figure 1.

To annotate the measures in the individual pictures, the Android app *Vertaktoid*² [18] was used. It allows to conveniently draw bounding boxes for all measures with a pen directly on the tablet screen. The results can then be exported to the MEI format [24] and used as ground truth training data.

Data coming from digital music editions are partly provided with concordance annotations between the measures.

¹The measure annotations are published as separate dataset at <https://apacha.github.io/OMR-Datasets/#muscima>

²<https://github.com/cemfi/vertaktoid>



Figure 1. Examples of cropped measures originating from different sources of the same work. All measures represent the same musical position, i.e. the same measure, within the work, but are in part extremely diverse in terms of instrumentation, graphic representation and also image resolution.

4. ALIGNING MEASURE SEQUENCES

Our proposed solution for the given task can be split into three individual parts. First, we have to find the bounding boxes of all measure in the score images. Then we need a metric in order to compute the similarity between two given measure in terms of musical content. And finally, we have to compute actual concordances for multiple sources of the same music.

4.1 Optical Measure Recognition

For automatically detecting measures in complete music scores, we propose a machine-learning approach with deep convolutional neural networks and a Faster R-CNN detector [23]. Faster R-CNN has been shown to work well in a range of situations, including detecting music objects [21]. In this case, there is just one class of objects that needs to be detected, and the objects typically cover large portions of the entire image with little overlap. Our implementation is based on the TensorFlow Object Detection API framework [14] and freely available online³.

We split the dataset randomly into 80% for training, 10% for validation, and 10% for testing. To avoid a bias toward scores with just one system, we categorize the samples into the ten categories depicted in table 1. From the training set we only use about 2000 images and draw them equally distributed from these ten categories, which results in some examples being used more than once. The only exemption are images without systems which are sampled only half as often as the other categories. For the validation and test sets we use all images from that split.

We tested the three different backbones, ResNet50, ResNet101 [13], and Inception-ResNet-V2 [27] and restricted ourselves to these to enable transfer-learning by

initializing the networks with weights trained on ImageNet which generally improves the learning process, especially at the beginning. Input images are resized to be no longer than 1024 pixel on the longest edge. The Intersection over Union (IoU) measures how well two bounding boxes overlap. If two predictions are very close, non-maximum suppression filters the box with the lower score. The IoU threshold is set to 0.6 and a maximum of 600 objects are detected per image. These parameters are derived from statistical analysis of the entire data set and cover > 99.99% of the dataset.

We evaluated the optical measure detection with the commonly used average precision (AP) metric, as defined for the COCO detection challenge [15]. It produces a single number that measures how well objects were detected. A detection is considered a match with the underlying ground truth if the IoU is above a certain threshold. The trained models achieve very good results with 78.7% AP (IoU=0.5:0.95) on the test set for the top-performing model with Inception-ResNet-V2 [27] backbone. A few samples of the detection output are depicted in Figure 2.

Given that the measure recognition step does not necessarily return the measures of a page in the musically correct order, we sort them according to the measure numbering rules outlined by Mexin et al. in [18].

4.2 Metric Learning

Now that the scans of all scores are divided into individual measures, they have to be compared with each other to identify equivalent measures. Again, we decided to take a deep learning approach to learn such a musical similarity metric between two measures directly from the images. The neural network is trained to compute an embedding for measure images so that similar measures are placed in the proximity of one another in the embedding space. This allows for convenient comparison of two measures by com-

³ <https://github.com/OMR-Research/MeasureDetector>

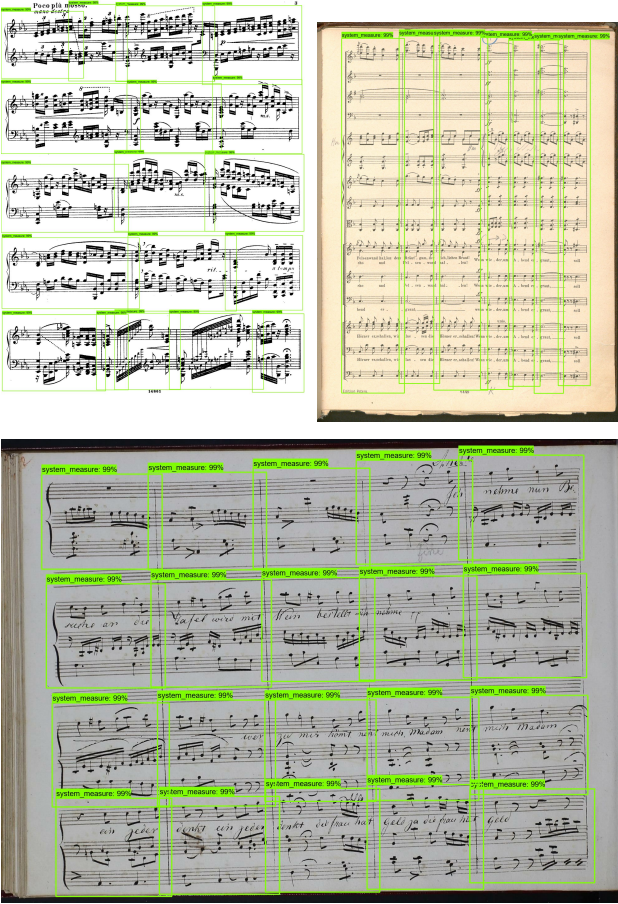


Figure 2. Three samples of the detection results. The neural network is capable of detecting measures robustly in typeset and handwritten scores, regardless of whether they contain piano scores or full orchestral scores. It does make occasional errors, but the majority of measures is being recognized correctly.

putting their distance, e.g., using the L^2 norm.

The idea is based on *triplet loss* [26]: A pair of equivalent measure images from two different sources is drawn from the list of concordances. We will call them the *anchor* image and the *positive* image. Additionally, a *negative* measure image is drawn from the same source as the positive image, serving as a counterexample, i.e. having no musical relation to the anchor or the positive measure image. Each of these three images is fed separately into the same neural network, resulting in three k -dimensional vectors. The loss function is defined as

$$\mathcal{L} = \max(d(f^a, f^p) - d(f^a, f^n) + \alpha, 0) \quad (1)$$

with f^a , f^p , and f^n being the resulting vectors from the network f for the three images and a distance measure d . Training with this loss function minimizes the distance from the anchor to the positive image while maximizing the distance between the anchor and the negative image. The additional margin α defines how far away the least dissimilarity should be. Finally, the surrounding $\max(\dots)$ function ensures that the loss never gets negative.

We chose ResNet50 as the base network and replaced

the usual final average pooling and classification layers by a fully connected layer with k -dimensional output. (Other CNN-based networks used for computer vision would most likely work comparably well.) All measure images are resized to 512×512 pixels but the original width and height information is also passed to the network as additional input.

The success of the used loss function depends heavily on the sampling strategy for the image triplets as discussed by Wojke and Bewley in [32]. In our context, there are three specific problems in the dataset:

1. A randomly sampled negative image might accidentally have the same musical content as the two other images. Those cases are not covered in the concordance dataset since not all measures with equal content have to be linked together.
2. Intuitively, it seems beneficial to take the previous or subsequent measure of the positive sample as the negative measure with the goal of enhancing the contrast between them in terms of increased distance in the embedding space. This would make adjacent measures more distinguishable. But again, the chance of these measures having the same content is higher compared to random sampling.
3. Especially handwritten sources sometimes exhibit excessive use of measure repeats and other abbreviations as can be seen in the left part of Figure 1. Such symbols are meaningless if their immediate context is not given.

The first two problems could be solved by manually adding all measures with the same content to the list of concordances. Given the amount of images, we decided against doing so and rely on rare collisions thanks to the large number of data. We also discarded the (perfectly valid) idea of looking at adjacent measures to form the triplets.

The third problem—presence of measure repeats and abbreviations—has a direct impact on the appropriate choice of the distance metric d in our loss function; When using triplet loss, it is common practice to normalize the embedding vectors. This constraint puts all embeddings on a k -dimensional hypersphere, leading to some advantages for further processing (see [26]). Furthermore, *cosine distance* is often used to calculate the distances. Both decisions make it impossible to get an embedding vector that is equally distant to all other possible vectors. This very property, however, characterizes the meaning of measure repeats if no context is given. We, therefore, opted for no vector normalization and chose the L^2 norm as our distance metric, resulting in

$$\mathcal{L} = \sum_{i=1}^N [\|f_i^a - f_i^p\|_2 - \|f_i^a - f_i^n\|_2 + \alpha]_+ \quad (2)$$

for a training batch with size N . To speed up training and ensure fast convergence we select triplets that violate the following constraint:

$$\|f_i^a - f_i^p\|_2 + \alpha < \|f_i^a - f_i^n\|_2. \quad (3)$$

This filter step is performed for each batch during training

and makes sure that only those triplets are used that significantly contribute to the learning process. It also prevents the network from overfitting.

4.3 Concordance Computation & Manual Adjustments

Given the embedding vectors for all measures of each source of a musical work, we can compare two sources by computing the distances between all measures from one source to the other. The resulting similarity matrices can then be used for *dynamic time warping* (DTW) as described by Müller in [19] to get an alignment path between the sources as shown in Figure 3.

We implemented the canonical DTW algorithm without any noteworthy modifications to the core. Allowed step sizes inside the similarity matrix during path computation are $(0, 1)$, $(1, 0)$, and $(1, 1)$. It rarely happens that a measure gets divided into two parts at system or page breaks, so we penalized steps along a single axis by a factor of 2 to slightly enforce one-to-one mappings of the measures.

The quality of the alignment was evaluated using a dataset with two sources and given ground truth concordances as outlined in Table 2. We have decided in favor of this particular dataset because it offers several challenges that occur only rarely in other works:

Split measures: Some measures are split into two parts at page breaks. Therefore, one measure of source *A* maps to two other measures of source *B*.

Completely different sections: An entire part of the piece was replaced in source *B*. Finding the "correct" concordance is impossible.

Additional parts: Source *B* contains a 16-measure Aria that is not present in the other source.

Missing measure annotations: We also intentionally removed measures from source *A* to simulate annotation errors.

	Pages	Measures
Source <i>A</i> (typeset)	250	3098
Source <i>B</i> (handwritten)	532	3176
Total	782	6274

Table 2. Structure of the evaluation dataset.

In the MIR community, DTW is often used to synchronize audio and/or symbolic score sources with each other [12]. The time resolution of the features in such scenarios is usually in the range of several dozen milliseconds. Deviations in the alignment path are therefore undesirable, but can often be neglected as long as they do not exceed certain limits. In our context, however, any deviation from the ground truth marks a significant error. We took this into account and defined a very simple score for the over-

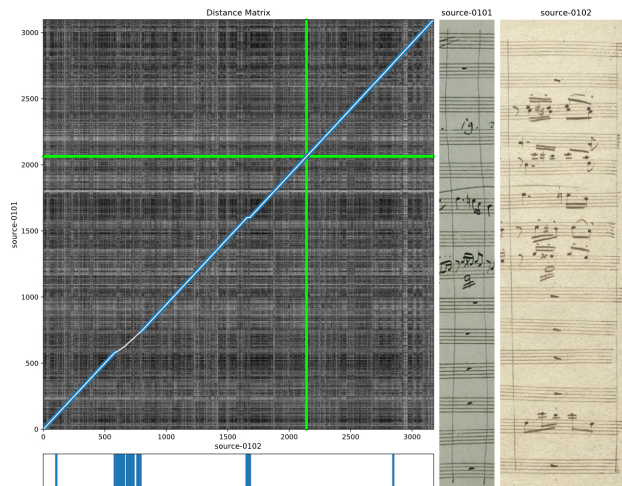


Figure 3. Interface for inspecting the computed measure concordances. The alignment (white) and ground truth (blue, only available in evaluation dataset) are plotted over the currently visible part of the similarity matrix. Measures of both sources (right) can be compared by moving a cursor within the matrix (green crosshair). A plot at the bottom indicates potentially interesting positions.

all performance:

$$score = 1 - \frac{\text{Number of } (x, y) \text{ pairs from alignment not in ground truth}}{\text{Total number of concordances in ground truth}} \quad (4)$$

Our evaluation showed 14 errors in relation to 3079 concordance pairs, resulting in a score of **99.545%**.

As pointed out, the remaining 0.455% error rate still present a non-negligible problem. Therefore, we developed an interface for manual adjustments to the alignment. Apart from being able to quickly compare the measures from two sources as shown in Figure 3, users can define points in the similarity matrix that have to be part of the alignment path. Each of these points splits the matrix into two parts and computes the warping path for each part individually, ensuring that either the beginning or end of the path matches the desired point. An event plot at the bottom of the matrix helps to identify regions with potential errors by showing where the alignment path is not diagonal, i.e. taking a step in $(0, 1)$ or $(1, 0)$ direction.

The mentioned obstacles for correct alignment have been handled successfully by either resulting in a correct alignment or—in case of substantial structural differences—indicating a problem that cannot be solved without human intervention by marking these parts in the plot below the similarity matrix.

This alignment and adjustment step has to be repeated for each source in regard to a *master source* of choice. The corrected alignment data can then finally be imported into the tools used by musicologists for their editorial work.

5. CONCLUSIONS AND FUTURE WORK

In this paper we proposed an approach to automate the tedious task of annotating and linking measures in heterogeneous score images, thereby allowing for cross-source navigation between measures without losing the current musical position. We used deep learning to find bounding boxes of measures in score images, learned a distance metric for measures, and used that to align measures from various sources, effectively linking equivalent musical positions across sources. The evaluation showed that our approach is feasible and solves a real-world problem while still retaining complete flexibility in case editors need to make manual adjustments, thanks to an interactive correction tool.

The presented solution still does not cover all possible situations that might occur in the editorial process. If the measure sequences to be compared have a different order, the alignment fails for these parts if not completely. We will address this specific problem in the future by identifying such passages and proposing reasonable re-ordering.

Having a musically meaningful distance metric for measures also allows closing the gap between score images and symbolic scores. The latter can be rendered with suitable engraving software and divided into individual measures, followed by the steps of our alignment pipeline. Since audio can also be rendered from symbolic scores, alignments between all three modalities are possible.

Another interesting application of our distance metric is the ability to visualize datasets in image fields as shown in Figure 4. Using dimensionality reduction algorithms such as T-SNE [16] or UMAP [17], the measures are positioned such that musically similar measures appear proximate to one another, giving new insight into a musical piece but also into the inner workings of the distance metric. For example, the visualization shows that measure repeats are placed almost in the center, indicating that their learned embedding retains the musical property of being close to basically every other measure in the embedding space.

6. REFERENCES

- [1] Benjamin W. Bohl, Axel Berndt, Simon Waloschek, and Aristotelis Hadjakos. Dem Igel Sitte lehren... Musikedition: von der digitalen Verfügbarkeit zur aktiven Nutzung. In Kristina Richts and Peter Stadler, editors, „*Ei, dem alten Herrn zoll' ich Achtung gern*“ – *Festschrift für Joachim Veit zum 60. Geburtstag*, chapter 12, pages 141–163. Allitera Verlag, Munich, Germany, 2016.
- [2] Jorge Calvo-Zaragoza and David Rizo. Camera-primus: Neural end-to-end optical music recognition on realistic monophonic scores. In *19th International Society for Music Information Retrieval Conference*, pages 248–255, Paris, France, 2018.
- [3] Jorge Calvo-Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. A machine learning framework for the categorization of elements in images of musical documents. In *3rd International Conference on Technologies for Music Notation and Representation*, A Coruña, Spain, 2017. University of A Coruña.
- [4] Tim Crawford, Golnaz Badkobeh, and David Lewis. 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*, pages 233–239, Paris, France, 2018.
- [5] 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, 2018.
- [6] 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, 2018.
- [7] 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.
- [8] Antonio-Javier Gallego and Jorge Calvo-Zaragoza. Staff-line removal with selectional auto-encoders. *Expert Systems with Applications*, 89:138–148, 2017.



Figure 4. 46 344 measure images from 15 different sources of the same piece are projected into a two-dimensional manifold with the UMAP algorithm. The map is interactively zoomable.

- [9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158, 2016.
- [10] 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, 2018.
- [11] 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, 2017.
- [12] Yun Hao. Real-time audio to score alignment (a.k.a score following). [https://www.music-ir.org/mirex/wiki/2019:Real-time_Audio_to_Score_Alignment_\(a.k.a_Score_Following\)](https://www.music-ir.org/mirex/wiki/2019:Real-time_Audio_to_Score_Alignment_(a.k.a_Score_Following)), 2019.
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [14] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014.
- [16] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
- [17] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.
- [18] Yevgen Mexin, Aristotelis Hadjakos, Axel Berndt, Simon Waloschek, Anastasia Wawilow, and Gerd Szwillus. Tools for annotating musical measures in digital music editions. In *14th Sound and Music Computing Conf. (SMC-17)*, Espoo, Finland, 2017. Aalto University.
- [19] Meinard Müller. *Information Retrieval for Music and Motion*. Springer-Verlag, Berlin, Heidelberg, 2007.
- [20] 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.
- [21] 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, 2018.
- [22] Fabrizio Pedersoli and George Tzanetakis. Document segmentation and classification into musical scores and text. *International Journal on Document Analysis and Recognition*, 19(4):289–304, 2016.
- [23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28*, pages 91–99. 2015.
- [24] Perry Roland. The music encoding initiative (MEI). In *1st International Conference on Musical Applications Using XML*, pages 55–59, 2002.
- [25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015.
- [26] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.
- [27] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, 2017.
- [28] Lukas Tuggener, Ismail Elezi, Jürgen Schmidhuber, and Thilo Stadelmann. Deep watershed detector for music object recognition. In *19th International Society for Music Information Retrieval Conference*, pages 271–278, Paris, France, 2018.
- [29] Joachim Veit and Kristina Richts. Current status and perspectives of MEI usage in musicology and in libraries. *Bibliothek Forschung und Praxis*, 42(2):292–301, 2018.
- [30] Gabriel Viglienconi, Gregory Burlet, and Ichiro Fujinaga. Optical measure recognition in common music notation. In *14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil, 2013.
- [31] S. Waloschek and A. Hadjakos. Driftin’ down the scale: Dynamic time warping in the presence of pitch

drift and transpositions. In *19th International Society for Music Information Retrieval Conference*, Paris, France, 2018.

- [32] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 748–756. IEEE, 2018.

Music Notation Graph Construction

To complete the proposed OMR pipeline, two steps are remaining after music objects were successfully detected in the image: constructing the notation graph and exporting it into the desired output format. In the article “Learning Notation Graph Construction for Full-Pipeline Optical Music Recognition,” Jorge Calvo-Zaragoza, Jan Hajič jr., and I investigated how the construction of a notation graph can be formulated as a machine-learning problem and thus be solved robustly and efficiently [PCZHj19].

The foundation for this work is the (music notation) graph representation inside the OMR pipeline, which consists of three things: the vertices, which represent the primitives, appearing in the image, the syntactic edges that relate these primitives with each other, and the precedence edges that specify the order of events, which is crucial when recognizing polyphonic music with simultaneous events. The vertices are created as a result of the music object detection stage. The edges on the other side are still missing. The initial attempt to build a binary classifier that decides whether a pair of nodes have an edge or not, showed room for significant improvement [HjDWP18]. We extended the initial work by adding a grammar which eliminates the proposal of illegal pairs, such as between two rests. The input of the neural network has changed to three channels: one for the image-patch that contains the two objects in question, one for the binary mask of the first object, and one for the binary mask of the second object. The results improved significantly, and the best model achieves an F1-score of over 95%.

This paper has been accepted for the 20th International Society for Music Information Retrieval Conference 2019 in Delft, The Netherlands.

LEARNING NOTATION GRAPH CONSTRUCTION FOR FULL-PIPELINE OPTICAL MUSIC RECOGNITION

Alexander Pacha

Institute of Information Systems
Engineering, TU Wien, Austria
alexander.pacha@tuwien.ac.at

Jorge Calvo-Zaragoza

Pattern Recognition and Artificial
Intelligence Group
University of Alicante, Spain
jcalvo@dlsi.ua.es

Jan Hajič jr.

Institute of Formal and
Applied Linguistics,
Charles University, Prague
hajicj@ufal.mff.cuni.cz

ABSTRACT

Optical Music Recognition (OMR) promises great benefits to Music Information Retrieval by reducing the costs of making sheet music available in a symbolic format. Recent advances in deep learning, have turned typical OMR obstacles into clearly solvable problems, especially the stages that visually process the input image, such as staff line removal or detection of music-notation objects. However, merely detecting objects is not enough for retrieving the actual content, as music notation is a configurational writing system, where the semantic of a primitive is defined by its relationship to other primitives. Thus, OMR systems must employ a notation assembly stage to infer such relationships among the detected objects. So far, this stage has been addressed by devising a set of predefined rules or grammars, which hardly generalize well. In this work, we formulate the notation assembly stage from a set of detected primitives as a machine learning problem. Our notation assembly is modeled as a graph that stores syntactic relationships among primitives, which allows us to capture the configuration of symbols in a music-notation document. Our results over the handwritten sheet music corpus MUSCIMA++ show 95.2% precision, 96.0% recall, and an F-score of 95.6% in establishing the correct syntactic relationships. When inferring relationships on data from a music object detector, the model achieves 93.2% precision, 91.5% recall and an F-score of 92.3%.

1. INTRODUCTION

Optical Music Recognition is the field of research that investigates how to read music notation in documents computationally. This technology enables many computational tasks that, otherwise, could not be performed directly on the music sources themselves [17]. One interesting application of OMR is concerned with reconstructing the notes encoded in the music-notation document, also referred to

as *replayability* [22]. In particular, the objective of the replayability application is to recover the pitches, onsets, durations, and velocities of notes from a document and export them into a symbolic representation. This symbolic representation—e.g., a MIDI file—is already a very useful abstraction of the music itself and allows for plugging in a wide range of music information retrieval tools. However, despite prolonged efforts, the replayability application is still under research [4, 7, 16, 36].

Given the wealth of information that is contained in a music score, the task of decoding its content is usually addressed by dividing the process into smaller stages that represent limited challenges. The general pipeline, proposed first by Bainbridge and Bell [3] and later refined by Rebelo et al. [29], is considered a de-facto standard, which organizes the process into four main blocks: i) preprocessing, which works over the input image to ease further steps and make the system more robust; ii) music object detection, which is in charge of retrieving and classifying all objects and glyphs of the image; iii) notation assembly, which must infer the relationships among the detected objects to reconstruct the music notation itself; and iv) encoding, which exports the symbolic reconstruction into the desired format, typically MIDI for replayability or an XML-based encoding such as MusicXML [15] or MEI [19] for further computational processing.

As our starting point towards completing the OMR pipeline, we assume that the music object detection stage can be solved reliably, which allows us to investigate how to deal with the later stages. In this paper, we want to focus in particular on the third stage, which is responsible for the notation assembly. Although previous work exists, most approaches are based on predefined rules that hardly generalize, and that only work for a limited set of scenarios. In contrast, we propose a well-principled machine learning approach, which addresses the problem in a generalizable way, provided there is convenient training data.

2. RELATED WORK

Most literature on OMR focuses on the first stages of the pipeline. This comes as no surprise because if one struggles with detecting music objects in an image reliably, it is understandable that subsequent stages that build on top of that are often neglected. With the appearance of deep



learning in OMR, however, many steps that traditionally produced suboptimal results, such as the staff-line removal or symbol classification, have seen drastic improvements [14, 26] and are no longer considered obstacles for OMR development.

Deep learning also caused some steps to become obsolete or collapse into a single (bigger) stage. For instance, the music object detection stage, which was traditionally separated into segmentation plus classification stages, is currently addressed in a single step. Convolutional 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 [25]. A compelling advantage is the capability of these models to be trained in a single step by merely providing pairs of images and positions of the music objects to be found, eliminating the preprocessing step altogether [24, 35]. This issue has been the subject of intense recent research. A comparison of existing approaches to holistic music object detection is presented in the work of Pacha et al. [27].

Since the beginning of the OMR research, there have been attempts to complete the full pipeline, including the notation assembly stage. Below, we introduce some particular proposals to perform this stage that can be found in the existing literature. They can be broadly divided into grammar-based approaches, and approaches that rely on heuristics and pre-defined rules.

2.1 Grammar-based approaches

Formal grammars represent the most widely used description of music notation. This feels natural, given that music notation has syntactic rules and hierarchical structures that invite such a formalization. These grammars are manually built to describe the expected relationships among music-notation objects and then used to reconstruct the music notation from the detected primitives [1–3, 5, 6, 30, 33]. The 2D nature of music notation also inspired graph grammars, as in the work of Fahmy and Blostein [12]. A prominent example of this approach is the DMOS system, proposed by Coüasnon et al. [8, 9], which uses a definite clause grammar for describing the relations between graphical objects on two levels: a graphical one that assists the recognition of symbols and a syntactic one, which introduces the musical semantics into the process.

2.2 Heuristical approaches

The other set of approaches relies on *ad hoc* rules for the music notation at hand. This includes assumptions about the configuration and position of the occurring primitives to reconstruct composite symbols and the notation graph [10, 23, 28, 34]. Rossant et al. [31] additionally considered fuzzy modeling, which allows for self-correction during the recognition [32]. Their system evaluated different hypotheses of recognized symbols to verify the compatibility between them.

3. NOTATION ASSEMBLY

The related works clearly show a lack of machine learning approaches. This work aims to fill that gap, by proposing a formulation of the notation assembly stage based on machine learning models. The advantage of such models is that they provide greater flexibility since they can vary their behavior by just changing the provided training set. This is especially interesting for OMR, where there is a great diversity of scenarios depending on the epoch or type of composition of the music scores.

The conventional OMR pipeline foresees that the notation assembly stage infers the relationships among previously detected music objects to retrieve the necessary information to infer the sequence of notes and rests.

Our approach understands that music notation can be modeled as a directed graph $G = (V, T)$, hereafter referred to as Music Notation Graph (MuNG). V represents the set of vertices, where $\zeta(v)$, $v \in V$ is the label associated with a vertex. T represents the set of directed edges, such that $t_i = (v_1, v_2)$, $t_i \in T$, $v_1, v_2 \in V$ denotes an edge from vertex v_1 to vertex v_2 . The primitives that make up the music notation, such as noteheads or stems, are modeled as vertices of this graph, while the relationships between these symbols are modeled by the edges. In our MuNG, the edges are not labeled, but there are two types of relationships:

- Syntactic edges that relate elements syntactically. This includes relationships between primitives that make up a composite symbol, such as an eighth note, which consist of a notehead, a stem, and a flag or beam as well as general relationships, e.g., between an accidental and the notehead that is affected by it.
- Precedence edges that specify the temporal order between notes. In most cases, the position on the horizontal axis is sufficient to infer this kind of relationship; however, for polyphonic music, a more sophisticated mechanism is needed to handle ambiguous situations.

We can, therefore, define the set of edges as $T = S \cup P$, where S is the set of edges that define the syntactic relationships and P is the set of edges that define the precedence relationships. A graphical representation of MuNG is shown in Fig. 1. The primary goal of our work is to train a machine learning model to construct such a MuNG G from a music score image.

4. LEARNING MUSIC NOTATION GRAPH ASSEMBLY

There are existing algorithms that are capable of dealing with the input image and retrieving a set of detected music-notation primitives. In other words, these algorithms process the input and provide the set of vertices V , along with its associated labels and bounding-boxes. In order to complete the OMR pipeline for replayability, we also need to recover the set of edges T .

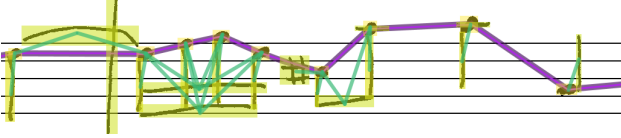


Figure 1: Graphical representation of a Music Notation Graph in a selected excerpt of music notation: vertices are highlighted with transparent yellow bounding boxes around the music-notation primitives, syntactic edges are shown as transparent cyan lines, and precedence edges are shown as transparent purple lines connecting the note-heads.

We propose a principled way of inferring T without resorting to a set of fixed rules but using machine learning. Our system learns to establish these relationships from a conveniently annotated training set so that the rules are implicitly modeled by the machine learning model.

The edges that relate vertices of the set T have an unlabeled binary nature; i.e. for each pair of vertices, a relationship either exists or not. Formally speaking, the inference of these relationships can be formulated as a function $f : V \times V \rightarrow \{0, 1\}$. However, given their different nature, the set of edges S and P are inferred by independent models. To learn the functions f_S and f_P , for the edges of S and P , respectively, we propose to train binary classifiers that receive two vertices and predict whether such relationship must be established or not. To do so, one would have to estimate the potential relationship between each pair of symbols, which entails high computational costs. However, it is obvious that most of these relationships are unfeasible. Since the music object detection stage also retrieves some associated information, such as the label $\zeta(v)$ associated to each vertex and the bounding box of that object in the input score image, we can use this information to filter edges by two criteria:

1. An edge is only feasible if the distance between the bounding boxes of their vertices falls below a certain threshold t . In other words, two vertices that are too far apart cannot be related.
2. An edge is only feasible if the labels of its associated vertices are “compatible”, e.g., a notehead with a stem. This eliminates a large number of incompatible combinations, such as an edge between an accidental and a rest. The compatibility map is a fixed list of vertex pairs that, according to the syntax of modern music notation, can hold a relationship to each other.

Then, given two vertices v_1 and v_2 , for which their edge is declared feasible, we train a deep convolutional neural network to predict whether there must be an edge from v_1 to v_2 or not. We generate a multi-channel image with a fixed size that serves as input features for the neural network, which consists of:

- Channel 1: the patch of the input score image that is centered at the objects represented by v_1 and v_2 .

- Channel 2: the binary mask of the object v_1
- Channel 3: the binary mask of the object v_2

The required information to generate these multi-channel images can be obtained from the bounding boxes of v_1 and v_2 , which are expected to be generated during the preceding music object detection stage. Note, that the masks for channel 2 and 3 are obtained from the bounding boxes and the underlying image, which means that the masks can (partially) include other objects as well unless the exact masks are provided via pixelwise segmentation [16, 35].

The network is then fed with this 3-channel image and trained to predict 1 if there should be a relationship between the vertices, and 0 otherwise. Visualizations of the input images are given in Fig. 2.

4.1 Dataset

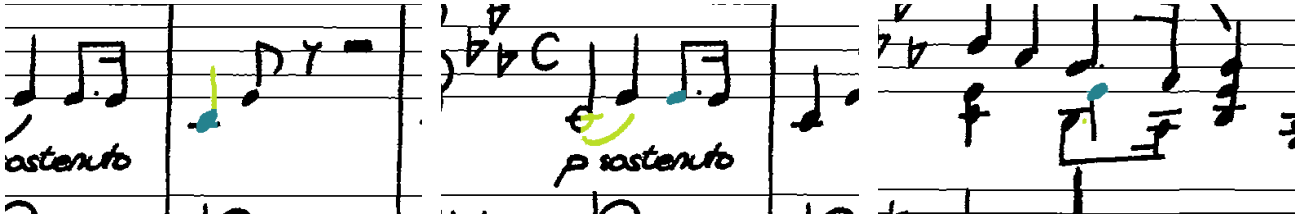
To carry out our experiments we need a corpus, which has annotations for both the individual symbols as well as their relationships. Currently, the only publicly available dataset which fulfills this requirement is the MUSCIMA++ dataset [18] of handwritten music notation. It provides symbol-level annotations as well as relationship annotations for 140 out of 1 000 images from the CVC-MUSCIMA dataset [13]. The MUSCIMA++ dataset contains 91 254 annotated symbols, consisting of both notation primitives and higher-level notation objects, such as key signatures or time signatures as well as 82 247 explicitly marked relationships between symbol pairs.

Unfortunately, the precedence relationships between notes are not included in the MUSCIMA++ dataset, so our experiments consider only the syntactic edges. However, the formulation and the proposed approach are very similar and should work for both kinds of edges.

4.2 Relationship Reconstruction

For learning the relationships, we train a convolutional neural network in PyTorch with five consecutive blocks, each consisting of a convolution, batch normalization, a non-linearity (ReLU), and max-pooling, before going into a fully connected layer with a single output neuron followed by a sigmoid activation function that produces the final estimation. The network has 28 865 parameters in total. We use the Binary Cross-Entropy loss and train with the Adam optimizer [20] until the validation performance has not improved for ten epochs, upon which we stop.

The data-loading routine presents the biggest challenge because it has to construct the multi-channel images as described in Sect. 4. To efficiently generate the set of vertex-pairs, we compute the pairwise distance between all objects in an image but filter them considerably afterward by the distance and compatibility criteria (see Sec. 4). The distance threshold was set to $t = 200$ pixels for including most valid edges from the MUSCIMA++ dataset. Valid relationships between objects that are further apart than 200 pixels are extremely rare and were neglected in favor of



(a) A positive example of two objects that are related. (b) A negative example of two objects that are unrelated. (c) A hard negative example of a dot that could be related to the notehead, but is not.

Figure 2: Three samples of images that are used during training. The mask given in channel 2 is shown as bright green overlay and the mask from channel 3 as cyan overlay.

computational efficiency. Our compatibility map contains 225 valid combinations of primitives. To improve the performance even further and simplify the classification task, the input image for the neural network is cropped to a sub-image of 512×256 pixels (width \times height), containing the two objects of interest at its center. Both the distance threshold and the sub-image dimensions are hyper-parameters that are dataset-dependent but can be obtained by running a statistical analysis on the used dataset.

We split the 140 images of the dataset into 60% training data, 20% validation data, and 20% test data. In each epoch, the network is trained on approximately 250 000 images of candidate pairs. Approximately 25 percent of the candidates contain positive examples. The best results were obtained after just 12 epochs before the network started to overfit and the validation performance declined. The source-code is publicly available on Github.¹

4.3 Music Object Detection

Since the notation assembly stage begins after the music objects have been detected in the score image, we also wanted to evaluate, how well our approach works on actual detection results. For obtaining such results, we resort to a state-of-the-art music object detector as proposed by Pacha et al. [25] with a minor modification: While we do divide the full page into sub-images containing one stave each, we do not see the need for cutting the images any further. The model selection and training procedure remains unchanged. We split the dataset into 100 images for training, 20 images for validation and 20 images for testing, as proposed by the authors of the MUSCIMA++ dataset. The improved implementation is publicly available.²

We evaluate the trained model on the test set for the stave-wise individual images and report the Mean Average Precision (mAP) as defined for the COCO challenge [21] which is a unified metric, commonly used for object detection tasks. The trained model achieves 69.5% mAP. For comparison, we also report a mAP of 93.3% when using the mAP as defined for the PASCAL VOC challenge [11], which was used in the original paper. Finally, the images are merged into the full-page results upon we achieve:

34.5% mAP / 45.2% w-mAP³ (COCO) and 53.8% mAP / 80.9% w-mAP (PASCAL). As our main focus is on learning relationships and not music object detection, we do not go into further details on these numbers. However, we want to point out that the COCO metric is very strict and probably underestimating the performance of the music object detector (see Fig. 3 for an example output).

4.4 Evaluation Protocol

Once the music objects have been detected, and their relationships established, the system can produce a complete MuNG that can be compared with the reference MuNG, provided as ground truth. However, it is necessary to first establish the correspondences between vertices from the prediction and the ground-truth. To do so, we assume that a detected object v_1 corresponds to a ground-truth object v_2 if they depict the same class $\zeta(v_1) = \zeta(v_2)$ and their Intersection over Union exceeds 50%.

Once the vertices of the ground-truth are matched with the detected objects, it is possible to compute the statistics. If an established relationship is correct, it is considered a true positive (TP); if an established relationship is incorrect, it is considered a false positive (FP); and, if an expected relationship is not predicted, it is considered a false negative (FN). Then, we can compute precision (P), recall (R), and F-score (F_1) metrics:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F_1 = 2 \frac{P \times R}{P + R}$$

P measures how reliable the established relationships are, whereas R measures the ability of the model to retrieve as many relationships as possible. F_1 summarizes both metrics with a single value.

Note that, although our evaluation is primarily focused on the relationships between objects, the used metrics are affected by the performance of the music object detector. Errors from earlier stages of the OMR process propagate to later stages. So if musical objects were missed, their relationships are counted as false negatives. To account for this, we evaluate our model in two ways:

¹ <https://github.com/OMR-Research/MungLinker>

² <https://github.com/apacha/MusicObjectDetector-TF>

³ Weighted Mean Average Precision is the Mean Average Precision, weighted by the frequency of the occurring classes, which is higher because frequent classes yielded better results than rare ones.



Figure 3: Sample output of the improved music object detector. Each detected object v has a box around it, with the color representing the class $\zeta(v)$ of the particular object, e.g., light green for full-noteheads.

- over a hypothetical set of perfect detections, which we can extract from the ground-truth of the corpus, and
- over the result of an actual music object detector, specifically using the state-of-the-art model, described in Sect. 4.3.

These settings allow us to answer the two questions: Does the proposed approach for reconstructing the MuNG with a machine learning model work at all? If yes, how well does the system perform in a real-world scenario, when confronted with (imperfect) object detector results instead of the perfect ground-truth bounding boxes?

4.5 Results

The main objective of our work is to demonstrate that the notation assembly stage can be formulated as a machine learning task. The main results of our experiments are given in Table 1. It can be observed that the proposed approach is highly effective: in all cases, values above 90% are reported.

When starting from ground-truth music object detection, our model yields $P = 95.2\%$, $R = 96.0\%$, and $F_1 = 95.2\%$, which indicates a successful approach to completing the OMR pipeline. In case of starting from actual results of a state-of-the-art detector, performance decreases slightly to $P = 93.2\%$, $R = 91.5\%$, and $F_1 = 92.3\%$. We think this is because the location of the

objects is not always exact (leading to a lower P) and missing symbols cause relationships to be irrecoverable (leading to a lower R).

	Graph Edges / Relationships		
	Precision	Recall	F-Score
Perfect Detection	95.2%	96.0%	95.6%
Real Detector	93.2%	91.5%	92.3%

Table 1: Overall performance of the proposed machine learning model to reconstruct syntactic edges of the Music Notation Graph (MuNG), given hypothetically perfect detection results (top row), and given results from a state-of-the-art detector (bottom row).

In order to provide more experimental insights, Table 2 reports 10 out of the 225 compatible combinations of relationships that are most common in the MUSCIMA++ dataset. As might be expected, the *notehead* primitives are involved in all of these frequent combinations. In this regard, our model obtains nearly optimal results for these over-represented cases. Note that these relationships are of particular relevance to be able to decode the notes that appear in the score. When comparing the individual results to the overall results in Table 1, the discrepancy can be explained by looking at the remaining 215 combinations that are not shown. Many of these have a much lower F_1 , probably because they are under-represented in the dataset.

From	To	Number of candidate pairs in the dataset	F-Score on the test set
notehead-full	stem	158064	99.5%
notehead-full	beam	61253	98.7%
notehead-full	leger_line	47503	98.1%
notehead-full	slur	24738	96.4%
notehead-full	8th_flag	12877	97.7%
notehead-full	sharp	12563	97.5%
notehead-full	duration-dot	12305	96.7%
notehead-empty	stem	9488	100.0%
notehead-full	staccato-dot	8628	96.8%
notehead-full	natural	7160	98.7%

Table 2: Overview of the ten most common combinations of object-pairs, along with the number of generated candidate pairs in the dataset, as seen by the network. The last column contains the F-scores that were reported for the individual combinations when evaluating the trained model on the test set, containing the ground truth of music primitives v .

5. CONCLUSION AND OUTLOOK

In this work, we study how to complete the OMR pipeline from the previous efforts to detect the music objects within the input image. Our approach seeks the construction of a music notation graph that stores the information of the notation primitives as well as their syntactic and precedence relationships. We propose a machine learning model that can predict whether two primitives are related to each other or not.

Results over the set of syntactic relationships from the handwritten sheet music dataset MUSCIMA++ show that our approach is very effective. We obtain success rates close to the optimum when establishing the correct relationships from the ground-truth primitives ($F_1 = 95.6\%$). When re-evaluating the results starting from the primitives detected by a state-of-the-art music object detector, a slightly lower performance can be observed ($F_1 = 92.3\%$). These figures indicate that the notation assembly stage of the OMR pipeline can be solved reliably with a machine learning model, given a curated set of annotated scores. Comparing our approach to existing methods is extremely difficult, if not impossible, because:

- most existing solutions are black boxes with closed source-code, or there is no available implementation at all,
- only a few systems are capable of handling handwritten modern notation, and
- it is unclear how to compare the music notation assembly stage between two different systems, especially given that the notation graph is only an intermediate representation.

So, although the results are promising, we still see many interesting avenues for further research. For instance, by adding data augmentation during training to make the notation assembly model more robust against variations in the bounding box retrieval of the first stage. Also, we plan to look into providing other meaningful features to the net-

work, such as the class labels $\zeta(v)$ of the involved primitives $v \in V$. Furthermore, we observed that the fixed-sized input patch given to the network is often covering a much larger area than required to contain the objects of interest, especially when they are very close (see Fig. 2c). This could be handled by using size-independent neural network layers such as *Global Pooling*, instead of flattening the features and feeding them into a fully-connected layer, allowing us to adjust the input patch for each sample.

We also believe that the notation assembly stage could benefit from having a broader set of hypotheses about the objects detected in the previous stage, instead of a fixed set of proposals. State-of-the-art music object detectors are based on statistical neural models that are able to provide a probability distribution over the whole set of possible detection hypotheses. When it comes to recognizing, we are typically interested in the most likely hypothesis—the one that is proposed as an answer—forgetting the other ones. However, it is certainly interesting to exploit this statistical modeling: the notation assembly 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 yet to be explored in the field of OMR.

And finally, for completing the OMR pipeline, the encoding stage is still missing. However, we see two benefits of the notation graph representation: the encoding can be implemented by experts in music encoding that are proficient in a particular format and given a complete graph representation, there is no restriction on the actual output format because the graph contains all the information that is present in the image.

6. REFERENCES

- [1] Alfio Andronico and Alberto Ciampa. On automatic pattern recognition and acquisition of printed music. In *International Computer Music Conference*, Venice, Italy, 1982.

- [2] David Bainbridge. *Extensible optical music recognition*. PhD thesis, University of Canterbury, 1997.
- [3] David Bainbridge and Tim Bell. The challenge of optical music recognition. *Computers and the Humanities*, 35(2):95–121, 2001.
- [4] Arnau Baró, Pau Riba, Jorge Calvo-Zaragoza, and Alicia Fornés. From optical music recognition to handwritten music recognition: A baseline. *Pattern Recognition Letters*, 123:1–8, 2019.
- [5] Pierfrancesco Bellini, Ivan Bruno, and Paolo Nesi. Optical music recognition: Architecture and algorithms. In *Interactive Multimedia Music Technologies*, pages 80–110. IGI Global, Hershey, PA, USA, 2008.
- [6] Dorothea Blostein and Henry S. Baird. A critical survey of music image analysis. In *Structured Document Image Analysis*, pages 405–434. Springer Berlin Heidelberg, 1992.
- [7] Jorge Calvo-Zaragoza and David Rizo. End-to-end neural optical music recognition of monophonic scores. *Applied Sciences*, 8(4), 2018.
- [8] Bertrand Couasnon, Pascal Brisset, and Igor Stéphan. Using logic programming languages for optical music recognition. In *3rd International Conference on the Practical Application of Prolog*, 1995.
- [9] Bertrand Couasnon and Jean Camillerapp. A way to separate knowledge from program in structured document analysis: Application to optical music recognition. In *3rd International Conference on Document Analysis and Recognition*, pages 1092–1097, 1995.
- [10] Michael Droettboom, Ichiro Fujinaga, and Karl MacMillan. Optical music interpretation. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 378–387, Berlin, Heidelberg, 2002.
- [11] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.
- [12] Hoda M. Fahmy and Dorothea Blostein. A graph grammar programming style for recognition of music notation. *Machine Vision and Applications*, 6(2):83–99, 1993.
- [13] 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.
- [14] Antonio-Javier Gallego and Jorge Calvo-Zaragoza. Staff-line removal with selectional auto-encoders. *Expert Systems with Applications*, 89:138–148, 2017.
- [15] Michael Good. MusicXML: An internet-friendly format for sheet music. Technical report, Recordare LLC, 2001.
- [16] 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, 2018.
- [17] 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, 2018.
- [18] 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, 2017.
- [19] Andrew Hankinson, Perry Roland, and Ichiro Fujinaga. The music encoding initiative as a document-encoding framework. In *12th International Society for Music Information Retrieval Conference*, pages 293–298, 2011.
- [20] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *Computing Research Repository*, abs/1412.6980, 2014.
- [21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014.
- [22] 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.
- [23] Kia Ng. Music manuscript tracing. *Lecture Notes in Computer Science*, 2390:322–334, 2002.
- [24] 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.
- [25] Alexander Pacha, Kwon-Young Choi, Bertrand Couasnon, 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, 2018.
- [26] Alexander Pacha and Horst Eidenberger. Towards a universal music symbol classifier. In *14th International Conference on Document Analysis and Recognition*, pages 35–36, Kyoto, Japan, 2017.

- [27] 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, 2018.
- [28] Christopher Raphael and Jingya Wang. New approaches to optical music recognition. In *12th International Society for Music Information Retrieval Conference*, pages 305–310, Miami, Florida, 2011.
- [29] 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.
- [30] K. Todd Reed and J. R. Parker. Automatic computer recognition of printed music. In *13th International Conference on Pattern Recognition*, pages 803–807, 1996.
- [31] Florence Rossant and Isabelle Bloch. A fuzzy model for optical recognition of musical scores. *Fuzzy Sets and Systems*, 141(2):165–201, 2004.
- [32] 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.
- [33] Mariusz Szwoch. Guido: A musical score recognition system. In *9th International Conference on Document Analysis and Recognition*, pages 809–813, 2007.
- [34] Lorenzo J. Tardón, Simone Sammartino, Isabel Barbancho, Verónica Gómez, and Antonio Oliver. Optical music recognition for scores written in white mensural notation. *EURASIP Journal on Image and Video Processing*, 2009(1):843401, 2009.
- [35] Lukas Tuggener, Ismail Elezi, Jürgen Schmidhuber, and Thilo Stadelmann. Deep watershed detector for music object recognition. In *19th International Society for Music Information Retrieval Conference*, pages 271–278, Paris, France, 2018.
- [36] 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.

OMR for Mensural Notation

Before modern staff notation was established, a number of preceding notations have evolved. One of them is mensural notation. It was used, for example, to write down sacred chants that were sung during the mass in the Catholic church (see Fig. 8.1).



Figure 8.1: Chant from the Capitan collection, written in mensural notation during the 17th century.

In contrast to modern notation, this early notation system had a smaller vocabulary and was more limited with regard to what could be expressed with it. This motivated Jorge Calvo-Zaragoza and me to work on a complete OMR system for these scores, which requires fewer building blocks than OMR systems that attempt to recognize modern staff notation.

In our work “Optical Music Recognition in Mensural Notation with Region-Based Convolutional Neural Networks,” published at the 19th International Society for Music

Information Retrieval Conference 2018 [PCZ18], we devised a simplified pipeline that only consists of three stages: music object detection, position classification, and semantics recognition. The first stage is similar to the one described in the papers above. The position classification on the other hand represents a new building block which can be reused in other scenarios to improve the robustness of OMR systems. The idea is to obtain the vertical position, which corresponds to the pitch,¹ by a neural network classifier. The benefit is that no stave recognition and removal stage is needed, while symbols can be classified robustly, relying on local information only. The position classification network worked exceptionally well, making virtually no errors and even spotting errors that were done by human annotators.

The last step dealt with reconstructing the semantics, which can be done with a set of simple heuristics. For example, there are no simultaneous events, so notes can simply be read left to right to determine their order. The encoding step, however, is non-trivial because the interpretation of the recognized symbols requires specialized domain knowledge. This part of the research was conducted by David Rizo and published along with a description of the MuRET project [RCZIn18].

¹Note that the actual pitch still depends on other symbols, such as the clef and accidentals.

OPTICAL MUSIC RECOGNITION IN MENSURAL NOTATION WITH REGION-BASED CONVOLUTIONAL NEURAL NETWORKS

Alexander Pacha

Institute of Visual Computing and Human-Centered Technology, TU Wien, Austria
alexander.pacha@tuwien.ac.at

Jorge Calvo-Zaragoza

PRHLT Research Center
Universitat Politècnica de València, Spain
jcalvo@upv.es

ABSTRACT

In this work, we present an approach for the task of optical music recognition (OMR) using deep neural networks. Our intention is to simultaneously detect and categorize musical symbols in handwritten scores, written in mensural notation. We propose the use of region-based convolutional neural networks, which are trained in an end-to-end fashion for that purpose. Additionally, we make use of a convolutional neural network that predicts the relative position of a detected symbol within the staff, so that we cover the entire image-processing part of the OMR pipeline. This strategy is evaluated over a set of 60 ancient scores in mensural notation, with more than 15000 annotated symbols belonging to 32 different classes. The results reflect the feasibility and capability of this approach, with a weighted mean average precision of around 76% for symbol detection, and over 98% accuracy for predicting the position.

1. INTRODUCTION

The preservation of the musical heritage over the centuries makes it possible to study a certain artistic or cultural paradigm. Most of this heritage exists in written form and is stored in cathedrals or music libraries [10]. In addition to the possible issues related to the ownership of the sources, this storage protects the physical preservation of the sources over time, but also limits their accessibility. That is why efforts are being made to improve this situation through initiatives to digitize musical archives [17,21]. These digital copies can easily be distributed and studied without compromising their integrity.

Nevertheless, this digitalization, which indeed represents a progress with respect to the aforementioned situation, is not enough to exploit the actual potential of this heritage. To make the most out of it, the musical content itself must be transcribed into a structured format that can be processed by a computer [6]. In addition to indexing

the content and thereby enabling tasks such as content-based search, this could also facilitate large-scale data-driven musicological analysis in general [39].

Given that the transcription of sources is extremely time-consuming, it is desirable to resort to automatic systems. Optical music recognition (OMR) is a field of research that investigates how to build systems that decode music notation from images. Regardless of the approach used to achieve such objective, OMR systems vary significantly due to the differences amongst musical notations, document layouts, or printing mechanisms.

The work presented here deals with manuscripts written in mensural notation, specifically with sources from the 17th century, attributed to the Pan-Hispanic framework. Although this type of mensural notation is generally considered as an extension of the European mensural notation, the Pan-Hispanic situation of that time underwent a particular development that fostered the massive use of handwritten copies. Due to this circumstance, the need for developing successful OMR systems for handwritten notation becomes evident.



Figure 1. A sample page of ancient music, written in mensural notation.

We address the optical music recognition of scores written in mensural notation (see Figure 1) as an object detection and classification task. In this notation, the symbols are atomic units,¹ which can be detected and categorized independently. Although there are polyphonic composi-

¹ Except for beamed notes, in which the beam can be considered an atomic symbol itself.



tions from that era, each voice was placed on its own page, so we can consider the notation as monophonic on the graphical level. Assuming the aforementioned simplifications allows us to formulate OMR as an object detection task in music score images, followed by a classification stage that determines the vertical position of each detected object within a staff. If the clef and other alterations are known, the vertical position of a note encodes its pitch.

We propose using region-based convolutional neural networks, which represent the state of the art in computer vision for object detection, and demonstrate their capabilities of detecting and categorizing the musical symbols that appear in the image of a music score with a high precision. We believe that this work provides a solid foundation for the automatic encoding of scores into a machine-readable music format like Music Encoding Initiative (MEI) [38] or MusicXML [15]. At present, there are thousands of manuscripts of this type that remain to be digitized and transcribed. Although each manuscript may have its own particularities (such as the handwriting style or the layout organization), the approach developed in this work presents a common and extensible formulation to all of them.

2. RELATED WORK

Most of the proposed solutions to OMR have focused on a multi-stage approach [34]. This traditional workflow involves steps that have been addressed isolatedly, such as image binarization [4,47], staff and text segmentation [44], staff-line detection and removal [5, 11, 46], and symbol classification [3, 30, 33]. In other works, a full pipeline is proposed for a particular type of music score [31, 32, 43].

Recent works have shown that the image-processing pipeline can largely be replaced with machine-learning approaches, making use of deep learning techniques such as convolutional neural networks (CNNs) [1, 16, 29, 45]. CNNs denote a breakthrough in machine learning, especially when dealing with images. They have been applied with great success to many computer vision tasks, often reaching or even surpassing human performance [18, 22]. These neural networks are composed of a series of filters that operate locally (i.e. convolutions, pooling) and compute various representations of the input image. These filters form a hierarchy of layers, each of which represents a different level of abstraction [20]. The key is that these filters are not fixed but learnt from the raw data through a gradient descent optimization process [23], meaning that the network can learn to extract data-specific, high-level features.

Here, we formulate OMR for mensural notation as an object detection task in music score images. Object detection in images is one of the fundamental problems in computer vision, for which deep learning can provide excellent solutions. Traditionally, the task has been addressed by means of heuristic strategies based on the extraction of low-level, general-purpose features such as SIFT [28] or HOG [7]. Szegedy and colleagues [8, 42] redefined the use of CNNs for object detection for the first time. Instead

of classifying the image, the neural network predicted the bounding box of the object within the image. Around the same time, the ground-breaking work of Girshick et al. [14] definitely changed the traditional paradigm. In their work, a CNN was in charge of predicting whether each object of the vocabulary appeared in selected bottom-up regions of the image. This scheme has been referred to as region-based convolutional neural network (R-CNN). Afterwards, several extensions and variations have been proposed with the aim of improving both the quality of the detection and the efficiency of the process. Well-known examples include Fast R-CNN [13], Faster R-CNN [37], R-FCN [24], SSD [27] or YOLO [35, 36].

In this work, we use these region-based convolutional neural networks for OMR, which are trained for the direct detection and categorization of music symbols in a given music document. Thereby allowing for an elegant formulation of the task, since the training process only needs score images along with their corresponding set of symbols and the regions (bounding boxes) in which they appear.

3. AN OMR-PIPELINE FOR MENSURAL SCORES

Music scores written in mensural notation share many properties with scores written in modern notation: the sequence of tones and pauses is captured as notes and rests within a reference frame of five parallel lines, temporally ordered along the x-axis with the y-axis representing the pitch of notes. But unlike modern notation, mensural scores are notated monophonically with a smaller vocabulary of only around 30 different glyphs, reducing the overall complexity significantly and thus allowing for a simplified pipeline that consists of only three stages. A representative subset of the symbols that appear in the considered notation is depicted in Table 1.





















Group	Symbol			
	Semibrevis	Minima	Col. Minima	Semiminima
Note				
Rest	Longa	Brevis	Semibrevis	Semiminima
				
Clef	C Clef	G Clef	F Clef (I)	F Clef (II)
				
Time	Major	Minor	Common	Cut
				
Others	Flat	Sharp	Dot	Custos
				

Table 1. Subset of classes from mensural notation. The symbols are depicted without considering their pitch or vertical position on the staff.

3.1 Music Object Detection

The first stage takes as input an entire high-quality image that contains music symbols. The entire image is fed into

a deep convolutional neural network for object detection and yields the bounding boxes of all detected objects along with their most likely class (e.g., *g-clef*, *minima*, *flat*).

3.2 Position classification

After detecting the symbols and classifying them, the second stage performs position classification of each detected object to obtain the relative position with respect to the reference frame (staff) which is required to recover a notes pitch. For this process, we extract a local patch from the full image with the object of interest in the center and feed the image into another CNN, which outputs the vertical position, encoded as shown in Figure 2.

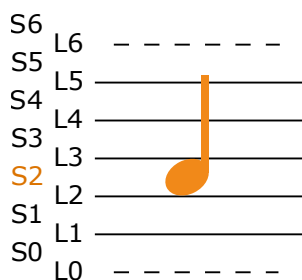


Figure 2. Encoding of the vertical staff line position into discrete categories. The five continuous lines in the middle form the regular staff and the dashed lines represent ledger lines, that are inserted locally as needed. A note between the second and third line from the bottom would be classified as S2 (orange).

3.3 Semantics Reconstruction and Encoding

Given the detected objects and their relative position to the staff line, the final step is to reconstruct the musical semantics and encode the output into the desired format (e.g., into modern notation [48]). This step has to translate the detected objects into an ordered sequence for further processing. Depending on the application and desired output, semantic rules need to be taken care of, such as grouping beams with their associated notes to infer the right duration or altering the pitch of notes when accidentals are encountered.

4. EXPERIMENTS

To evaluate the proposed approach, we conducted experiments² for the first two steps of the pipeline. While a full system would also require the third step, we refrain from implementing it, to not restrict this approach to a particular applications. It is also noteworthy, that translating mensural notation into modern notation can be seen as its own field of research that requires a deep understanding of

²Source code is available at <https://github.com/apacha/Mensural-Detector>

both notational languages, which exceeds the scope of this work.

4.1 Dataset

Our corpus consists of 60 fully-annotated pages in mensural notation from the 16th-18th century. The manuscript represents sacred music, composed for vocal interpretation.³ The compositions were written in music books by copyists of that time. To ensure the integrity of the physical sources, the images were taken with a camera instead of scanning the books in a flatbed scanner, leading to sub-optimal conditions in some cases. An overview of the considered corpus is given in Table 2.

Pages	60
Total number of symbols	15258
Different classes	32
Different positions within a staff	14
Average size of a symbol ($w \times h$)	44×84 pixels
Number of symbols per image	42–447 (\varnothing 250)
Image resolution ($w \times h$)	$\sim 3000 \times 2000$ pixels
Dots per inch (DPI)	300

Table 2. Statistics of the considered corpus.

The ground-truth data is collected using a framework, in which an electronic pen is used to trace the music symbols, similar to that of [2]. The bounding boxes of the symbols are then obtained by computing the rectangular extent of the users' strokes.

4.2 Setup

Our experiments are based on previous research by [29], where a sliding-window-approach is used to detect handwritten music symbols in sub-regions of a music score. In contrast to their work, we are able to detect hundreds of tiny objects in the full page within a single pass. To train a network in a reasonable amount of time within the constraints of modern hardware, it is currently necessary to shrink the input image to be no longer than 1000px on the longest edge, which corresponds to a downscaling operation by a factor of three on our dataset.

For detecting music objects, the Faster R-CNN approach [37] with the Inception-ResNet-v2 [41] feature extractor has been shown to yield very good results for detecting handwritten symbols [29]. It works by having a region-proposal stage for generating suggestions, where an

³The dataset is subject to ongoing musicological research and can not be made public at this point in time, so it is only available upon request.

object might be, followed by a classification stage, which confirms or discards these proposals. Both stages are implemented as CNNs and trained jointly on the provided dataset. The first stage scans the image linearly along a regular grid with user-defined box proposals in each cell of that grid.

To be able to generate meaningful proposals, the shape of these boxes has to be similar to the actual shape of the objects that should be found. Since the image contains a large number of very tiny objects (sometimes only a few pixels), a very fine grid is required. After a statistical analysis of the objects appearing in the given dataset, including dimension clustering [35], several experiments were conducted to study the effects of size, scale, and aspect ratios of the above-mentioned boxes, concluding that sensibly chosen priors for these boxes work similarly good as the boxes obtained from the statistical analysis. For the down-scaled image, boxes of 16x16 pixels, iterating with a stride of 8 pixels and using the scales 0.25, 0.5, 1.0, and 2.0, with aspect ratios of 0.5, 1.0, and 2.0 represent a meaningful default configuration. Accounting for the high density of objects, the maximum number of box proposals is set to 1200 with a maximum of 600 final detections per image.

For the second step of our proposed pipeline, another CNN is trained to infer the relative position of an object to its staff line upon which it is notated (see Figure 2). Different off-the-shelf network architectures are evaluated (VGG [40], ResNet [19], Inception-ResNet-v2 [41]) with the more complex models slightly outperforming the simpler ones. Using pre-trained weights instead of random initialization accelerates the training, improves the overall result, and is therefore used throughout all experiments. The input to the classification network is a 224 x 448 pixels patch of the original image with the target object in the center (see Figure 3). The exact dimensions of the patch are not important, as long as the image contains enough vertical and horizontal context to classify even symbols notated above or below the staff. When objects appear too close to the border, the image is padded with the reflection along the extended edge to simulate the continuation of the page as shown in Figures 3(d) and 3(e).

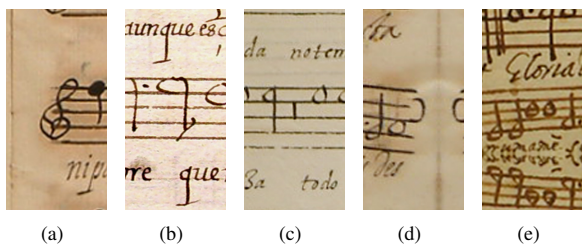


Figure 3. Sample inputs for the position classification network depicting a *g-clef* (a), *semiminima* (b), *brevis rest* (c), *custos* (d) and *semibrevis* (e), with vertical (d) and horizontal (e) reflections of the image to enforce the target object to be in the center, while preserving meaningful context.

It is important to notice that the vertical position defines the semantical meaning only for some symbols (e.g.,

the pitch of a *note* or the upcoming pitch with a *custos*). Classes for which the position is either undefined or not of importance include *barlines*, *fermatas*, different *time-signatures*, *beams* and in particular for mensural notation: the *augmentation dot*. Symbols from these classes can be excluded from the second step.

4.3 Evaluation metrics

Concerning the music object detection stage, the model provides a set of bounding box proposals, as well as the recognized class of the objects therein. The model also yields a *score* of its confidence for each proposal. A bounding box proposal B_p is considered positive if it overlaps with the ground-truth bounding box B_g exceeding 60%, according to the Intersection over Union (IoU) criterion: ⁴

$$\frac{\text{area}(B_p \cap B_g)}{\text{area}(B_p \cup B_g)}$$

If the recognized class matches the actual category of the object, it is considered a true positive, being otherwise a false positive. Additional detections of the same object are computed as false positives as well. Those objects for which the model makes no proposal are considered false negatives. Given that the prediction is associated with a score, different values of *precision* and *recall* can be obtained for each possible threshold. To obtain a single metric, Average Precision (AP) can be computed, which is defined as the area under this precision-recall curve. An AP value can be computed independently for each class, and then we provide the mean AP (mAP) as the mean 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. For the second part of the pipeline (position classification), we evaluate the performance with the accuracy rate (ratio of correctly classified samples).

5. RESULTS

Both experiments yielded very promising results while leaving some room for improvement. The detection of objects in the full image (see Figure 4) was evaluated by training on 48 randomly selected images and testing on the remaining 12 images with a 5-fold cross-validation. This task can be performed very well and yielded 66% mAP and 76% w-mAP. When considering practical applications, the weighted mean average precision indicates the effort needed to correct the detection results, because it reflects the fact that symbols from classes that appear frequently are generally detected better than rare symbols.

When reviewing the error cases, a few things can be observed: Very tiny objects such as the *dot*, *semibrevis rest* and *minima rest* pose a significant challenge to the network, due to their small size and extremely similar appearance (see Figure 5). This problem might be mitigated,

⁴ as defined for the PASCAL VOC challenge [9]



Figure 4. Detected objects in the full image with the detected class being encoded as the color of the box. This example achieves a mAP of approximately 68% and a w-mAP of 85%.

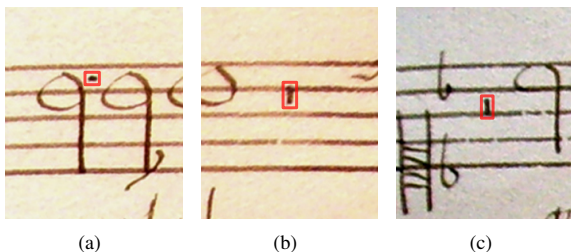


Figure 5. The smallest objects from the dataset that are hard to detect and often confused (from left to right): *dot*, *semibrevis rest*, and *minima rest*.

by allowing the network to access the full resolution image, which potentially has more discriminative information than the downsized image. Unsurprisingly, classes that are underrepresented such as *dots*, *barlines*, or all types of *rests* are also frequently missed or incorrectly classified, leading to average precision rates of only 10–40% for these classes.

Another interesting observation can be made, that in many cases, objects were detected but the IoU with the underlying ground-truth was too low for considering them a true positive detection (see Figure 6 with a red box being very close to a white box).

For the second experiment, a total of 13246 sym-

bols were split randomly into a training (80%), validation (10%) and test set (10%). The pre-trained Inception-ResNet-v2 model is then fine-tuned on this dataset and achieves over 98% accuracy on the test set of 1318 samples. Analyzing the few remaining errors reveals that the model makes virtually no errors and that the misclassified samples are mostly human annotation errors or data inconsistencies.

For inference, both networks can be connected in series. Running both detection and classification takes about 30 seconds per image when running on a GPU (GeForce 1080 Ti) and 210 seconds on a CPU.

6. CONCLUSION

In this work, we have shown that the optical music recognition of handwritten music scores in mensural notation, can be performed accurately and extendible by formulating it as an object detection problem, followed by a classification stage to recover the position of the notes within the staff. By using a machine learning approach with region-based convolutional neural networks, this problem can be solved by simply providing annotated data and training a suitable model on that dataset. However, we are aware that our proposal still has room for improvement. In future work we would like to:



Figure 6. Visualization of the performance of the object detection stage with selected patches of the music documents: green boxes indicate true positive detections; white boxes are false negatives, that the network missed during detection; red boxes are false positive detections, where the model reported an object, although there is no ground-truth; yellow boxes are also false positives, where the bounding-box is valid, but the assigned class was incorrect.

- evaluate the use of different network architectures, such as feature pyramid networks [25,26], that might improve the detection of small objects, which we have identified as the biggest source of error at the moment. These networks allow the use of high-resolution images directly, without the inherent information loss, that is caused by the downscaling operation.
- merge the staff position classification with the object detection network, by adding another output to the neural network, so the model simultaneously predicts the staff position, the bounding box and the class label.
- apply and evaluate the same techniques for other notations, including modern notation
- study models or strategies that reduce (or remove) the need for specific ground-truth data of each type of manuscript. For example, unsupervised training

schemes such as the one proposed in [12], which allows the network to adapt to a new domain by simply providing new, unannotated images.

We believe that this research avenue represents a ground-breaking work in the field of OMR, as the presented approach would potentially deal with any type of music scores by just providing undemanding ground-truth data to train the neural models.

7. ACKNOWLEDGEMENT

Jorge Calvo-Zaragoza thanks the support from the European Union’s H2020 grant READ (Ref. 674943), the Spanish Ministerio de Economía, Industria y Competitividad through Juan de la Cierva - Formación grant (Ref. FJCI-2016-27873), and the Social Sciences and Humanities Research Council of Canada.

8. REFERENCES

- [1] J. Calvo-Zaragoza and D. Rizo. End-to-End Neural Optical Music Recognition of Monophonic Scores. *Applied Sciences*, 8(4):606–629, 2018.
- [2] J. Calvo-Zaragoza, D. Rizo, and J. M. Iñesta. Two (note) heads are better than one: pen-based multimodal interaction with music scores. In *17th International Society for Music Information Retrieval Conference*, pages 509–514, 2016.
- [3] J. Calvo-Zaragoza, A. J. G. Sánchez, and A. Pertusa. Recognition of Handwritten Music Symbols with Convolutional Neural Codes. In *14th IAPR International Conference on Document Analysis and Recognition*, pages 691–696, 2017.
- [4] J. Calvo-Zaragoza, G. Vigliensoni, and I. Fujinaga. Pixel-wise binarization of musical documents with convolutional neural networks. In *15th IAPR International Conference on Machine Vision Applications*, pages 362–365, 2017.
- [5] J. S. Cardoso, A. Capela, A. Rebelo, C. Guedes, and J. P. da Costa. Staff detection with stable paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1134–1139, 2009.
- [6] G. S. Choudhury, M. Droetboom, T. DiLauro, I. Fujinaga, and B. Harrington. Optical music recognition system within a large-scale digitization project. In *1st International Symposium on Music Information Retrieval*, pages 1–6, 2000.
- [7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.
- [8] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2154, 2014.
- [9] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.
- [10] I. Fujinaga, A. Hankinson, and J. 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, 2014.
- [11] A.-J. Gallego and J. Calvo-Zaragoza. Staff-line removal with selectional auto-encoders. *Expert Systems with Applications*, 89:138–148, 2017.
- [12] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.
- [13] R. Girshick. Fast R-CNN. In *IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.
- [14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.
- [15] M. Good and G. Actor. Using MusicXML for file interchange. In *Third International Conference on WEB Delivering of Music*, page 153, 2003.
- [16] J. Hajič Jr. and P. Pecina. Detecting Noteheads in Handwritten Scores with ConvNets and Bounding Box Regression. *Computing Research Repository*, abs/1708.01806, 2017.
- [17] A. Hankinson, J. A. Burgoyne, G. Vigliensoni, A. Porter, J. Thompson, W. Liu, R. Chiu, and I. Fujinaga. Digital Document Image Retrieval Using Optical Music Recognition. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, pages 577–582, 2012.
- [18] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.
- [19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [20] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, pages 1106–1114, 2012.
- [21] A. Laplante and I. Fujinaga. Digitizing musical scores: Challenges and opportunities for libraries. In *3rd International workshop on Digital Libraries for Musicology*, pages 45–48. ACM, 2016.
- [22] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [24] Y. Li, K. He, J. Sun, et al. R-FCN: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*, pages 379–387, 2016.
- [25] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature Pyramid Networks for Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

- [26] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal Loss for Dense Object Detection. *Computing Research Repository*, abs/1708.02002, 2017.
- [27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37, 2016.
- [28] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [29] A. Pacha, K.-Y. Choi, B. Coüasnon, Y. Ricquebourg, R. Zanibbi, and H. Eidenberger. Handwritten music object detection: Open issues and baseline results. In *13th IAPR Workshop on Document Analysis Systems*, pages 163–168, 2018.
- [30] A. Pacha and H. Eidenberger. Towards a Universal Music Symbol Classifier. In *12th IAPR International Workshop on Graphics Recognition*, pages 35–36, 2017.
- [31] L. Pugin. Optical music recognition of early typographic prints using hidden markov models. In *7th International Conference on Music Information Retrieval*, pages 53–56, 2006.
- [32] C. Ramirez and J. Ohya. Automatic recognition of square notation symbols in western plainchant manuscripts. *Journal of New Music Research*, 43(4):390–399, 2014.
- [33] A. Rebelo, A. Capela, and J. S. Cardoso. Optical recognition of music symbols. *International Journal on Document Analysis and Recognition*, 13(1):19–31, 2010.
- [34] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. Marcal, C. Guedes, and J. S. Cardoso. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 1(3):173–190, 2012.
- [35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.
- [36] J. Redmon and A. Farhadi. YOLO9000: Better, Faster, Stronger. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6517–6525, 2017.
- [37] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- [38] P. Roland. The music encoding initiative (MEI). In *Proceedings of the First International Conference on Musical Applications Using XML*, pages 55–59, 2002.
- [39] X. Serra. The computational study of a musical culture through its digital traces. *Acta Musicologica*. 2017; 89 (1): 24-44., 2017.
- [40] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *Computing Research Repository*, abs/1409.1556, 2014.
- [41] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *31st AAAI Conference on Artificial Intelligence*, pages 4278–4284, 2017.
- [42] C. Szegedy, A. Toshev, and D. Erhan. Deep Neural Networks for Object Detection. In *Advances in Neural Information Processing Systems 26*, pages 2553–2561, 2013.
- [43] L. J. Tardón, S. Sammartino, I. Barbancho, V. Gómez, and A. Oliver. Optical Music Recognition for Scores Written in White Mensural Notation. *EURASIP Journal on Image and Video Processing*, 2009(1):1–23, 2009.
- [44] R. Timofte and L. Van Gool. Automatic stave discovery for musical facsimiles. In *Asian Conference on Computer Vision*, pages 510–523, 2012.
- [45] E. van der Wel and K. Ullrich. Optical music recognition with convolutional sequence-to-sequence models. In *18th International Society for Music Information Retrieval Conference*, pages 731–737, 2017.
- [46] M. Visaniy, V. C. Kieu, A. Fornés, and N. Journet. The ICDAR 2013 music scores competition: Staff removal. In *International Conference on Document Analysis and Recognition*, pages 1407–1411, 2013.
- [47] Q. N. Vo, S. H. Kim, H. J. Yang, and G. Lee. An MRF model for binarization of music scores with complex background. *Pattern Recognition Letters*, 69:88–95, 2016.
- [48] Yu-Hui Huang, Xuanli Chen, Serafina Beck, David Burn, and Luc J. Van Gool. Automatic Handwritten Mensural Notation Interpreter: From Manuscript to MIDI Performance. In *16th International Society for Music Information Retrieval Conference*, pages 79–85, 2015.

Other contributions

Apart from the scientific publications mentioned above and two more scientific papers ([Pac18c] and [Pac18a]), some side-projects evolved over the last few years. They gained significant attention both from the community but also from prospective researchers who are making use of these projects and the resources that I have shared publicly.

9.1 Optical Music Recognition Datasets project

One of the most pressing issues among OMR researchers has been the lack of datasets. While music in principle was available on a large scale, annotated datasets were not. So most researchers resorted to creating their own small datasets while researching the subject. This situation changed in recent years. Therefore, I collected the datasets that have been published so far and made that list available online [Pac17b]. It is a curated list with more than 20 datasets that were developed explicitly for OMR. Each entry contains a summary, a link to the official website, optionally the scientific publication where it was published as well as a small example from the dataset, cf. Fig. 9.1:

Apart from the links and the summaries, the OMR datasets project also provides a Python software package `omrdatasettools` [Pac18b] that facilitates working with the datasets, including downloader scripts, converters and image generators for datasets that only have a textual description of the underlying data. The Github repository also mirrors most of the referenced datasets to prevent them from suddenly disappearing in case the original websites are taken down.

This contribution has gained significant attention in the community and is referenced from various scientific articles.

Universal Music Symbol Collection

Official website: <https://github.com/apacha/MusicSymbolClassifier>, [Slides](#)

License MIT

Summary: A collection of various other datasets which combines 7 datasets into a large unified dataset of 90000 tiny music symbol images from 79 classes that can be used to train a universal music symbol classifier. 74000 symbols are handwritten and 16000 are printed symbols.

Scientific Publication: Alexander Pacha, Horst Eidenberger. Towards a Universal Music Symbol Classifier. Proceedings of the 12th IAPR International Workshop on Graphics Recognition, Kyoto, Japan, November 2017. DOI: [10.1109/ICDAR.2017.265](https://doi.org/10.1109/ICDAR.2017.265)

Example:

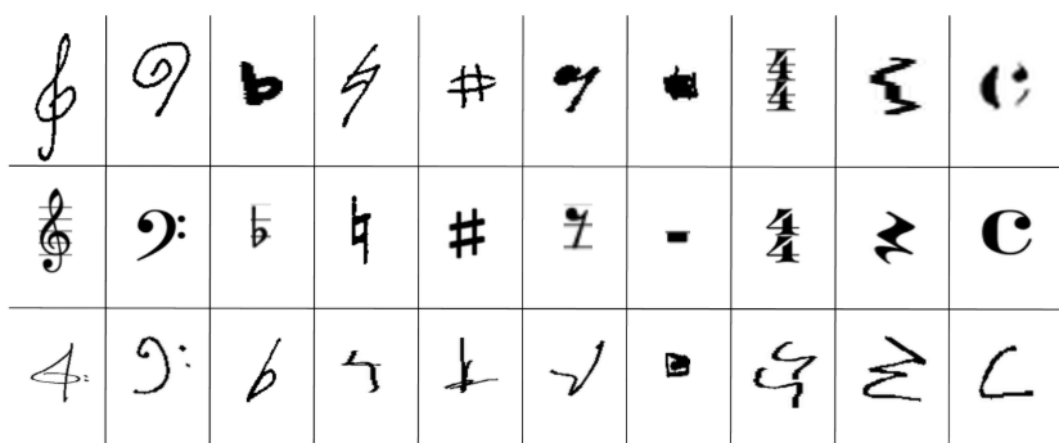


Figure 9.1: Screenshot of the website for the OMR Datasets project.

9.2 ISMIR Tutorial “Optical Music Recognition for Dummies”

At the International Society of Music Information Retrieval (ISMIR) Conference 2018 in Paris, France, Jorge Calvo-Zaragoza, Jan Hajič jr., Ichiro Fujinaga, and I gave a 3-hour tutorial on Optical Music Recognition, called “Optical Music Recognition for Dummies.” It spanned the entire spectrum of OMR: from the history of the field to modern approaches which were presented a few days later at the conference. The entire session was recorded by us and published on YouTube [CZHjPF18]. So far, the videos have been viewed more than 400 times (April 2019).

9.3 Workshop on Reading Music Systems (WoRMS)

In 2018 Jorge Calvo-Zaragoza, Jan Hajič jr., and I organized the first Workshop on Reading Music Systems (WoRMS) [CZHjP18b] which was a satellite event to ISMIR 2018 in Paris, France with about 30 attendees. It was the first time that the majority of active researchers in OMR sat in the same place. WoRMS was organized similar to the GREC workshop [FL17], where the idea for a dedicated OMR workshop was born. The workshop featured 12 talks from researchers who work on OMR as well as users of OMR systems, such as librarians. Each session was followed by an interactive discussion on the presented papers. Another WoRMS is planned for 2019 in Delft, The Netherlands, again as a satellite event to ISMIR [CZPR19].

9.4 Workshop at MEC 2019: Let's Formalize Music Notation

The Music Encoding Conference [DKKG19] is an annual conference on music encodings, digital musicology, digital editions, and symbolic music information retrieval. In 2019, the workshop “Let's Formalize Music Notation for OMR” was held by Jorge Calvo-Zaragoza, Heinz Roggenkemper, and me. The goal of this workshop was to work towards a standard representation for OMR, something that does not yet exist. Given that the Music Encoding Initiative has a large body of knowledge in the field and significant interest in the results of OMR systems, it was an ideal place to jointly work on this subject.

9.5 Discussion Group Summary: Optical Music Recognition

In 2017, I attended the 12th IAPR International Workshop on Graphics Recognition [FL17] in Kyoto, Japan. During the workshop multiple discussion groups were formed, including one on Optical Music Recognition. The discussion was summarized by Jorge Calvo-Zaragoza, Jan Hajič jr., and me, and published as part of Springer Lecture Notes in Computer Science [CZHjP18a].

9.6 Community Engagement and Website for OMR-Research

As a result of the Workshop on Reading Music Systems, the community decided that it wanted a website for future OMR research. A few months later, we launched <https://omr-research.net>, which is an expanding collection of resources on OMR, links to upcoming events as well as blog-entries with ideas that are still in rough shape. We have also established a Slack channel [Pac17c] that is actively being used by researchers as well as a Github Organization [Pac19a] to channel the development of various OMR projects.

9.7 OMR Bibliography

A side-effect of writing a thesis is that one has to study the literature of the subject thoroughly. For writing the paper “Understanding Optical Music Recognition,” we tried to gather all papers that were written on the subject of OMR. We collected the BibTex citations for hundreds of articles and manually verified them. Similar efforts were made before by Ichiro Fujinaga [Fuj00], Kia Ng, and Andrew Hankinson [Han12]. They published extensive bibliographies on the internet as static websites. We wanted to go one step further and have published a curated list of BibTeX entries on OMR research along with a static website that is generated from those entries online. The idea is to make the life of future OMR researchers easier by providing a verified bibliography of nearly all past research. The Github repository is open for submissions from the community, the rendered website can easily be updated, and our ultimate goal is to provide a valuable asset for all OMR researchers to correctly quote previous research.

Conclusions and Outlook

The most important conclusion from this thesis is that large parts of the OMR can be formulated as a machine learning problem, which in turn can be solved efficiently with deep learning. While the overall pipeline has been slightly reformulated, its general structure remains unchanged: preprocessing, symbol detection, semantic reconstruction, encoding. Especially when trying to recover all of the information for structured encoding, I believe that there will not be an alternative to this anytime soon. While there have been other attempts to solve OMR in a complete end-to-end fashion—feeding in an image, and getting encoded music out—they leave much to be desired. Single stave, monophonic music can be processed this way, but as soon as polyphony is involved or interactions between multiple staves appear, these approaches face their limits because they rely on the serializability of the score for encoding music as an ordered sequence. Certainly, it would be desirable if a system could learn everything from reading an image to producing a MIDI file, without any intervention being necessary at all. Unfortunately, I see no evidence that such a system is feasible. An alternative approach could be to learn the construction of the notation graph directly. But even that exceeds the boundaries of what I think is possible today.

Coming back to what is actually possible: Music Object Detection, one of the subjects I spent the most effort on, is now clearly solvable. Object Detectors that use deep convolutional neural network are powerful enough to provide very good results. I have shown that the approach generalizes well across datasets, meaning that it performs well on the dataset it was trained on. However, it is still unclear whether the trained networks generalize well across datasets. Can they transfer easily from one dataset to another?

While the music object detectors operate well, there is also a catch, which is the need for large, annotated datasets required for the training. Building such a dataset is a costly endeavor; so, if anyone decides to put an effort into it, it will be of immense benefit to publicly share it. To this end, I see the OMR datasets project as a milestone that will help

future researchers getting started much faster. I also contributed two smaller datasets, but more importantly the facilities for finding and working with existing datasets.

With the newly introduced definition of OMR and its taxonomy, it became much clearer why there is no answer to the question “Does OMR work?”: Because it is an ill-defined question. Certain applications for OMR already work well, whereas others do not. Recovering the structured encoding remains a big challenge, and I believe it will still take some time before we will see reliable OMR systems in commercial products.

One important distinction can help future systems to become more flexible and robust: decoupling the internal representation used during the recognition from the final representation into which the final results are encoded. The reason is that music encodings such as MusicXML, MEI, MuseScore XML, or MIDI were not designed for the specific needs of OMR. For example, they struggle to represent syntactically incorrect scores, which can quickly happen if the recognition fails. By representing music notation in a graph with vertices and edges, this barrier is removed, which allows the system to store the information as faithfully as possible. The last part of the OMR pipeline, the export, can then be handled by someone who is an expert in music encoding and not necessarily in machine learning or computer vision. While some encodings only require a fraction of the information from the notation graph, it can be useful in other situations to have all the information, e.g., when deciding how to resolve ambiguities, or where to ask for user-intervention if the system fails. Unfortunately, it is unclear, whether the idea of the notation graph will be picked up and developed further by other researchers or not. The past has shown that existing tools will only be adopted and used if they provide useful features, are well-designed, documented, and ready-to-use. Therefore, one key ingredient that could make the notation graph successful would be if the export into at least one widely used format was already available.

A benefit of pushing music object detection into the area of clearly solvable problems was that later process stages now started to receive more attention. For many years, OMR research was struggling with early stages such as the detection and removal of staff lines. This has changed for good. I believe that music object detection still has plenty of room for improvement. For instance, the trained models can probably be much smaller than 200 MB with hundreds of layers, while still producing excellent detection results.

As we started several community activities, we saw more collaboration between the research groups as well as new application scenarios popping up during discussions. I think that the research conducted for this thesis pushed the state of the art forward significantly. Maybe in five years, OMR as a whole will be considered a solved problem, although I doubt it will. But at least we will be a big step closer to solving it.

List of Figures

1.1	Excerpt from the waltz “An der schönen blauen Donau” by Johann Strauss, Jr.	2
1.2	First measures from the guitar riff of the song “Enter Sandman” by Metallica.	3
1.3	The initial three measures of Lisa’s first composition for the piano.	3
1.4	A born-digital version of music scores, typeset by a music score editor and without artifacts or degradations.	4
1.5	The same musical snippet as in Fig. 1.4, but degraded, as it can happen in real-world scenarios: The staff is slightly slanted, the image is blurred and noisy due to a poor image capturing process, and some straight lines are bent, which frequently happens when making photos of scores that are bound in a book.	5
1.6	The same musical snippet as in Fig. 1.4, but handwritten on a tablet with a stylus.	5
1.7	The word ‘Research’ written three times with vertically shifted letters, which always remains the word research, whereas the values of the three notes that are also slightly shifted vertically represent three different notes with the pitches A, B, and G.	6
1.8	Three quarter-notes appear in the second space from the top within the reference system. The reference system’s origin is given by the G-Clef at the beginning, which specifies the G to be on the second line from the bottom. So the first note corresponds to a C, but with the given key-signature at the beginning which depicts two sharps with one of them placed on the second space from the top, it makes the note a C#. The second note has a local modifier that undoes this alteration from the key signature, which makes the note a C. The third note has no local modifier, but the effect of the local modifier from the second note is propagated to consecutive notes within the measure, making it also a C. So even if the first and third note visually look exactly the same, their semantics (pitch) is different.	6
4.1	A small sample of music symbols that are part of the collected music symbols dataset. It depicts ten different classes of handwritten and typeset symbols in modern notation.	72

5.1	Illustration of the sliding window approach, used to crop music scores into sub-images (red boxes). Boxes overlap both vertically with the boxes above and below as well as with adjacent crops (orange).	74
8.1	Chant from the Capitan collection, written in mensural notation during the 17th century.	123
9.1	Screenshot of the website for the OMR Datasets project.	134

Bibliography

- [CZHjP18a] Jorge Calvo-Zaragoza, Jan Hajič jr., and Alexander Pacha. Discussion Group Summary: Optical Music Recognition. In *Graphics Recognition, Current Trends and Evolutions*, Lecture Notes in Computer Science, pages 152–157. Springer International Publishing, 2018.
- [CZHjP18b] Jorge Calvo-Zaragoza, Jan Hajič jr., and Alexander Pacha. Website of the Workshop on Reading Music Systems 2018. <https://sites.google.com/view/worms2018/home> (Last visited 18.06.2019), 2018.
- [CZHjP19] Jorge Calvo-Zaragoza, Jan Hajič jr., and Alexander Pacha. Understanding Optical Music Recognition. *ACM Computing Surveys (under review)*, 2019.
- [CZHjPF18] Jorge Calvo-Zaragoza, Jan Hajič jr., Alexander Pacha, and Ichiro Fujinaga. The recording of the ISMIR Tutorial "OMR for Dummies" on YouTube. <https://www.youtube.com/playlist?list=PL1jvwDVNwQke-04Uxz1zY4FM33bo1CGS0> (Last visited 18.06.2019), 2018.
- [CZO14] 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.
- [CZPR19] Jorge Calvo-Zaragoza, Alexander Pacha, and Heinz Roggenkemper. Website of the Workshop on Reading Music Systems 2019. <https://sites.google.com/view/worms2019/home> (Last visited 18.06.2019), 2019.
- [DKKG19] Norbert Dubowy, Robert Klugseder, Franz Kelnreiter, and Paul Gulewycz. Website of the Music Encoding Conference 2019. <https://music-encoding.org/conference/2019/> (Last visited 18.06.2019), 2019.
- [ETPS18] Ismail Elezi, Lukas Tuggener, Marcello Pelillo, and Thilo Stadelmann. DeepScores and Deep Watershed Detection: Current State and Open Issues. In *1st International Workshop on Reading Music Systems*, pages 13–14, Paris, France, 2018.

- [FL17] Alicia Fornés and Bart Lamiroy. Website of the 12th IAPR International Workshop on Graphics Recognition. <https://grec2017.loria.fr/> (Last visited 18.06.2019), 2017.
- [Fuj00] Ichiro Fujinaga. Optical Music Recognition Bibliography. <http://www.music.mcgill.ca/~ich/research/omr/omrbib.html> (Last visited 18.06.2019), 2000.
- [GJB⁺18] Mark Gotham, Peter Jonas, Bruno Bower, William Bosworth, Daniel Rootham, and Leigh VanHandel. Scores of Scores: An Openscore Project to Encode and Share Sheet Music. In *5th International Conference on Digital Libraries for Musicology*, pages 87–95, Paris, France, 2018. ACM.
- [Han12] Andrew Hankinson. Optical Music Recognition Bibliography. http://ddmal.music.mcgill.ca/research/omr/omr_bibliography (Last visited 18.06.2019), 2012.
- [HjDWP18] 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, 2018.
- [HjP17] 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, 2017.
- [HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [LF16] Audrey Laplante and Ichiro Fujinaga. Digitizing Musical Scores: Challenges and Opportunities for Libraries. In *3rd International Workshop on Digital Libraries for Musicology*, pages 45–48, 2016.
- [LMB⁺14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.
- [LPR⁺] Tsung-Yi Lin, Genevieve Patterson, Matteo R. Ronchi, Yin Cui, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, Larry Zitnick, and Piotr Dollár. COCO Detection Leaderboard. <http://cocodataset.org/#detection-leaderboard> (Last visited 18.06.2019).
- [MSH⁺85] T. Matsushima, I. Sonomoto, T. Harada, K. Kanamori, and S. Ohteru. Automated High Speed Recognition of Printed Music (WABOT-2 Vision

- System). In *International Conference on Advanced Robotics*, pages 477–482, 1985.
- [Pac17a] Alexander Pacha. Github Repository of the Music Score Classifier. <https://github.com/apacha/MusicScoreClassifier> (Last visited 18.06.2019), 2017.
- [Pac17b] Alexander Pacha. The OMR Datasets Project. <https://apacha.github.io/OMR-Datasets> (Last visited 18.06.2019), 2017.
- [Pac17c] Alexander Pacha. Slack Channel for Research on Optical Music Recognition. <http://omr-research.slack.com> (Last visited 18.06.2019), 2017.
- [Pac18a] Alexander Pacha. Advancing OMR as a Community: Best Practices for Reproducible Research. In *1st International Workshop on Reading Music Systems*, pages 19–20, Paris, France, 2018.
- [Pac18b] Alexander Pacha. Documentation of the OMR Dataset Tools Python package. <https://omr-datasets.readthedocs.io/en/latest> (Last visited 18.06.2019), 2018.
- [Pac18c] Alexander Pacha. Self-learning Optical Music Recognition. In *Vienna Young Scientists Symposium*, pages 34–35. Book-of-Abstracts.com, Heinz A. Krebs, 2018. ISBN: 978-3-9504017-8-3.
- [Pac19a] Alexander Pacha. Github Organisation for Research on Optical Music Recognition. <https://github.com/omr-research> (Last visited 18.06.2019), 2019.
- [Pac19b] Alexander Pacha. Github Repository for the Deep Learning Based Detector for Measures in Musical Scores. <https://github.com/OMR-Research/MeasureDetector/> (Last visited 18.06.2019), 2019.
- [PCC⁺18] Alexander Pacha, Kwon-Young Choi, Bertrand Couïasnon, 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, 2018.
- [PCZ18] 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.
- [PCZHj19] Alexander Pacha, Jorge Calvo-Zaragoza, and Jan Hajič jr. Learning Notation Graph Construction for Full-Pipeline Optical Music Recognition. In *20th International Society for Music Information Retrieval Conference (in press)*, 2019.

- [PE17a] Alexander Pacha and Horst Eidenberger. Towards a Universal Music Symbol Classifier. In *14th International Conference on Document Analysis and Recognition*, pages 35–36, Kyoto, Japan, 2017. IEEE Computer Society.
- [PE17b] Alexander Pacha and Horst Eidenberger. Towards Self-Learning Optical Music Recognition. In *16th International Conference on Machine Learning and Applications*, pages 795–800, 2017.
- [PHjCZ18] 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, 2018.
- [RCZIn18] 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.
- [Sul36] John W. N. Sullivan. *Beethoven: His Spiritual Development*. Knopf, Alfred A., 1936.
- [SZ14] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *Computing Research Repository*, abs/1409.1556, 2014.
- [TES⁺18] 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.
- [TESS18] Lukas Tuggener, Ismail Elezi, Jürgen Schmidhuber, and Thilo Stadelmann. Deep Watershed Detector for Music Object Recognition. In *19th International Society for Music Information Retrieval Conference*, pages 271–278, Paris, France, 2018.
- [WHP19] Simon Waloschek, Aristotelis Hadjakos, and Alexander Pacha. Identification and Cross-Document Alignment of Measures in Music Score Images. In *20th International Society for Music Information Retrieval Conference (in press)*, 2019.