



**Proceedings of the
5th International Workshop on
Reading Music Systems**

4th November, 2023
Milan, Italy

Organization

General Chairs

Jorge Calvo-Zaragoza
Alexander Pacha
Elona Shatri

University of Alicante, Spain
TU Wien, Austria
Queen Mary University of London, United Kingdom

Support

This workshop is partially supported by project I+D+i PID2020-118447RA-I00 (MultiScore), funded by MCIN/AEI/10.13039/501100011033.



Proceedings of the 5th International Workshop on Reading Music Systems, 2023

Edited by Jorge Calvo-Zaragoza, Alexander Pacha, and Elona Shatri



© The respective authors.
Licensed under a Creative Commons Attribution 4.0 International License (CC-BY-4.0).

Logo made by Freepik from www.flaticon.com. Adapted by Alexander Pacha.

Preface

Dear colleagues!

We are very pleased to present to you the proceedings of the 5th International Workshop on Reading Music Systems (WoRMS).

When we started the workshop in 2018 we did not know how it would be perceived by the community. Therefore, we are very proud that WoRMS has established a fixed place in the community and is seeing great interest from people all around the world that share a common interest in music reading systems.

After hosting several editions in online and hybrid formats, we are excited to be back to the face-to-face setting, once again co-located with the ISMIR conference. One of the main goals for WoRMS is its chance for researchers to come together, share experiences, and kickstart discussions. In doing so, it allows us to fostering valuable connections and create a sense of community. We hope that the face-to-face setting will facilitate these goals even further.

We would also like to use the opportunity here to promote the Github organization <https://github.com/omr-research> which welcomes contributions from everyone and can serve as a central hub for publishing and finding research-related repositories. Finally, we would also like to mention our public YouTube channel <https://www.youtube.com/OpticalMusicRecognition>, which has recordings for previous years' sessions and we plan on adding this year's presentations as well. If you have interesting content that you want to share through this channel, please get in touch with us.

This year's edition features 10 accepted contributions (from 11 total submissions), presenting interesting projects, advancements in the collection of datasets, as well as several really exciting new technical advancements. We are looking forward to very interesting discussions.

Jorge Calvo-Zaragoza, Alexander Pacha, and Elona Shatri

Contents

<i>Ichiro Fujinaga and Gabriel Vigliensoni</i> Optical Music Recognition Workflow for Medieval Music Manuscripts . . .	4
<i>Tristan Repolusk and Eduardo Veas</i> The Suzipu Musical Annotation Tool for the Creation of Machine-Readable Datasets of Ancient Chinese Music	7
<i>Jan Hajič, jr., Petr Žabička, Jan Rychtář, Jiří Mayer, Martina Dvořáková, Filip Jebavý, Markéta Vlková, and Pavel Pecina</i> The OmniOMR Project	12
<i>Juan Carlos Martinez-Sevilla and Francisco J. Castellanos</i> Towards Music Notation and Lyrics Alignment: Gregorian Chants as Case Study	15
<i>Jonáš Havelka, Jiří Mayer, and Pavel Pecina</i> Symbol Generation via Autoencoders for Handwritten Music Synthesis . .	20
<i>Pranjali Hande, Elona Shatri, Benjamin Timms, and György Fazekas</i> Towards Artificially Generated Handwritten Sheet Music Datasets	25
<i>Zihui Zhang, Elona Shatri, and György Fazekas</i> Improving Sheet Music Recognition using Data Augmentation and Image Enhancement	31
<i>Antonio Ríos-Vila</i> Rotations Are All You Need: A Generic Method For End-To-End Optical Music Recognition	34
<i>María Alfaro-Contreras</i> Few-Shot Music Symbol Classification via Self-Supervised Learning and Nearest Neighbor	39
<i>Francisco J. Castellanos, Antonio Javier Gallego, and Ichiro Fujinaga</i> A Preliminary Study of Few-shot Learning for Layout Analysis of Music Scores	44

Optical Music Recognition Workflow for Medieval Music Manuscripts

Ichiro Fujinaga
Schulich School of Music
McGill University
 Montréal, Canada
 ichiro.fujinaga@mcgill.ca

Gabriel Vigliensoni
Faculty of Fine Arts
Concordia University
 Montréal, Canada
 gabriel.vigliensoni@concordia.ca

Abstract—The overall goal of the **Single Interface for Music Score Searching and Analysis (SIMSSA) project is to develop and deploy infrastructure for large-scale musical data search and discovery. In this technical report, we outline the key component of our workflow for processing and encoding Medieval music manuscripts.**

Index Terms—optical music recognition, medieval music, document analysis, machine learning, music encoding

I. INTRODUCTION

The SIMSSA project [5] is a large government-funded, multi-year research project, featuring participation from over 30 researchers, collaboration with 22 institutional partners (primarily music libraries), and engaging with more than 20 students at any given time.

Our main goal is to make musical scores searchable online the way we can now search text (e.g., Google Books). To achieve this, we have developed an Optical Music Recognition (OMR) system that can be used by anyone with some knowledge of music notation. Because of the large volume of musical scores to be processed, we rely on the interested people in the music community, such as musicians, music scholars, and students, to assemble this online music library.

Our system uses and relies on two standards that are becoming widely adopted: the Music Encoding Initiative (MEI)¹ and the International Image Interoperability Framework (IIIF).² Mirroring the Text Encoding Initiative (TEI)³—the scholarly standard for digital text editions—MEI is a symbolic music file format that enables the rich and extensible encoding of various types of musical notations. Every music document processed by our OMR workflow is encoded in the MEI format. Moreover, the SIMSSA project has extended MEI with a Neume Module XML schema⁴ that can be applied to a wide variety of neume music notation. The IIIF is designed to remotely access high-resolution images (our images typically range between 20–30 million pixels in full colour). Leveraging this protocol, we can efficiently process and give access to large collections of musical scores without the need to store these large images locally.

¹<http://music-encoding.org>

²<https://iiif.io>

³<https://tei-c.org>

⁴<https://music-encoding.org/guidelines/dev/content/neumes.html>

Furthermore, our system integrates the Cantus Manuscript Database⁵—a large, expertly curated repository with over 200 manuscripts and early music prints—for text alignment and metadata-driven search.

II. WORKFLOW OUTLINE

As shown in Figure 1, our OMR workflow consists of several stages and components, starting with digitized music score images as the input to the system.

In the Layout Analysis stage, Pixel.js [6]—a web-based tool to separate different components in the image, such as staff lines, text (lyrics), music symbols, and background—is used to create training data for convolutional neural networks (CNN). Once trained, the application automatically segments the different components on unseen musical scores.

For the second stage, Symbol Classification & Correction, we created InteractiveClassifier.js, a k-nearest neighbour classifier designed to automatically classify musical symbols. This is a web-based version of the Gamera classifier [4]. Its interface allows users to interactively train the classifier on musical and non-musical symbols.

The third stage involves the processing and encoding of the neume types, their pitches, and the text associated with them. Automatic Pitch Finder uses CNN to determine the vertical positions of clefs and note heads, which will allow the system to determine the pitch of the note or pitches contained in a neume. OCR & Text Aligner is an automatic process to align a known set of text of the music with what is on the image of the music that contains text. It uses Calamari OCR,⁶ to find the text. Then, dynamic programming (Needleman-Wunsch algorithm [9]) is used to align the corresponding texts [7]. The Neume Mapping Tool is a web application where mappings between musical symbols and MEI encodings are defined. This is particularly useful for early music notation, such as neume notation, where different encodings are required for the many combinations of musical symbols. MEI Generator is used to combine the previous steps to output the MEI file of the musical score.

⁵<https://cantusdatabase.org>

⁶<https://github.com/Calamari-OCR/calamari>

SIMMSA Workflow for Neume Notation

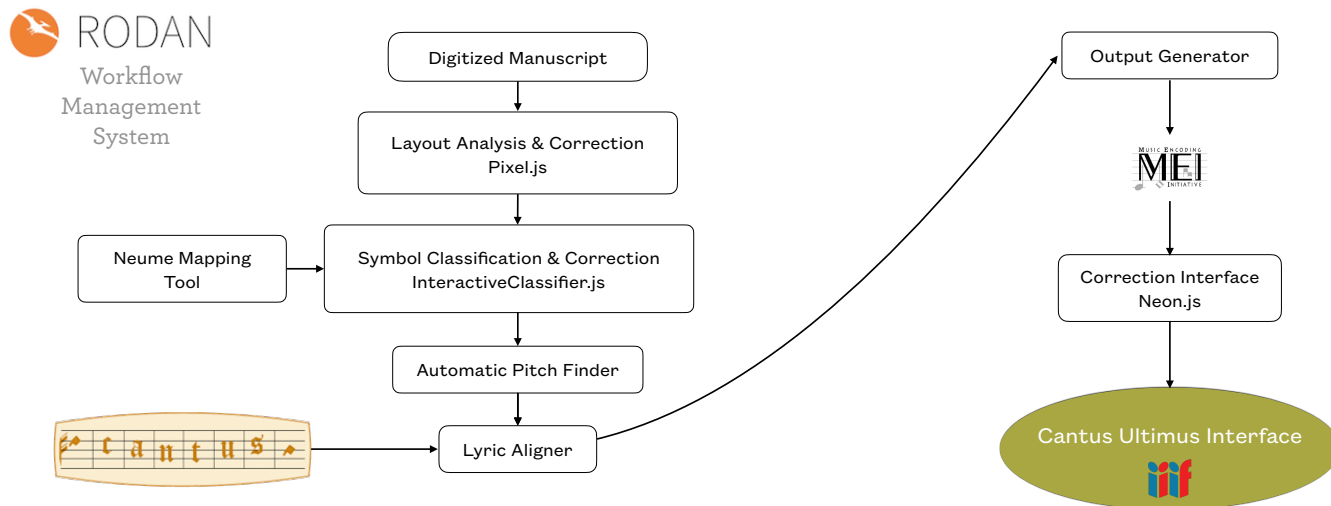


Fig. 1. Workflow diagram for SIMMSA OMR process for Medieval neume notation manuscripts.

The resulting encoding can be corrected using Neon.js (Neume Editor Online) [1], an interactive web application for editing square neume notation based on Verovio.⁷

Finally, results of the OMR process, the MEI files, are merged with the necessary metadata from the Cantus Manuscript Database and presented to the user using the Cantus Ultimus website,⁸ which uses the Diva.js⁹ [7] document image viewer that takes advantage of IIIF. The website allows not only efficient browsing but with full-text searching capabilities. The entire workflow is managed by a software we developed called Rodan¹⁰ allowing multiple users to run multiple processes on the cloud [3].

III. USE CASES AND OUTCOMES

We have completed the encoding of 8,192 chants from two large Medieval music manuscripts (Salzinnes, CDN-Hsmu M2149.L4¹¹ and Einsiedeln, CH-E 611¹²), and we are now preparing CDN-Mlr MS Medieval 0073.¹³ The Alamire Foundation has began using our infrastructure for the encoding of two other large manuscripts: Ts grooten Antiphoner¹⁴ and Aaavverbode: B-Avna, IV.413.¹⁵

The most important outcome of this project is allowing people—from scholars to performers to the general public—to search and discover music held in libraries and archives around the world. We believe this will fundamentally change how we

study music and will allow a global audience of musicians and artists to discover previously unknown or overlooked pieces for performance, making undiscovered repertoires that extend beyond the canon available to the general public. We also expect the public availability of large amounts of musical data to lead to significant advances in the field of music theory and promote the long-awaited blossoming of computational musicology. Lastly, we expect that the free and open-source tools we have developed will help lead to significant advances in various other fields.

ACKNOWLEDGMENT

We would like to thank over 100 developers who worked on this project over the last 20 years. This research has been generously funded by both the Canadian and Québec governments; most recently funded by the Social Sciences and Humanities Research Council (895-2013-1012), the Canada Research Chair Program (2020-00140), and the Fonds de recherche du Québec - Société et Culture (2022-SE3-303927).

REFERENCES

- [1] G. Burret, A. Porter, A. Hankinson, and I. Fujinaga, “Neon.js: Neume Editor Online,” in *Proceedings of the 13th International Society for Music Information Retrieval Conference*. Porto, Portugal, 2012, pp. 121–126.
- [2] F. J. Castellanos, J. Calvo-Zaragoza, G. Vigiensoni, and I. Fujinaga, “Document analysis of music score images with selectional auto-encoders,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference*. Paris, France, 2018, pp. 256–263.
- [3] A. Hankinson. “Optical Music Recognition Infrastructure for Large-scale Music Document Analysis,” *Ph.D. Dissertation*, 2015. McGill University, Montréal, QC.
- [4] M. Droettboom, K. MacMillan, and I. Fujinaga, “The Gamera framework for building custom recognition systems,” in *Proceedings of the Symposium on Document Image Understanding Technologies*, Greenbelt, MD, 2003, pp. 275–286.

⁷<https://www.verovio.org>

⁸<https://cantus.simssa.ca/manuscripts/>

⁹<https://diva.simssa.ca>

¹⁰<https://github.com/ddmal/rodan/>

¹¹<https://cantus.simssa.ca/manuscript/123723>

¹²<https://cantus.simssa.ca/manuscript/123606>

¹³<https://cantus.simssa.ca/manuscript/680970/>

¹⁴<https://cantusdatabase.org/source/702611>

¹⁵https://repository.teneo.libis.be/delivery/DeliveryManagerServlet?dps_pid=IE16026901&fl_pid=FL16027794

- [5] I. Fujinaga, A. Hankinson, and J. E. Cumming, "Introduction to SIMSSA (Single Interface for Music Score Searching and Analysis)," in *Proceedings of the 1st International Workshop on Digital Libraries for Musicology*, 2014, pp. 1–3.
- [6] Z. Saleh, K. Zhang, J. Calvo-Zaragoza, G. Vigiensoni, and I. Fujinaga. 2017. "Pixel.js: Web-based pixel classification correction platform from ground truth creation," in *Proceedings of the 12th IAPR International Workshop on Graphics Recognition*, Kyoto, Japan, 2017, pp. 39–40.
- [7] T. De Reuse, and I. Fujinaga, "Robust transcript alignment on Medieval chant manuscripts," in *Proceedings of the 2nd International Workshop on Reading Music Systems*. Delft, Netherlands, 2019.
- [8] A. Hankinson, W. Liu, L. Pugin, and I. Fujinaga, "Diva.js: A continuous document viewing interface." *Code4Lib Journal*, vol. 14, 2011.
- [9] S. B. Needleman, and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol.48, no. 3, pp.443–453, 1970.

The Suzipu Musical Annotation Tool for the Creation of Machine-Readable Datasets of Ancient Chinese Music

Tristan Repolusk
 Know-Center GmbH
 Graz, Austria
 trepolusk@know-center.at

Eduardo Veas
 Know-Center GmbH
 Graz, Austria
 eveas@know-center.at

Abstract—*Suzipu* notation, also called *banzipu* notation, is a notation which was predominantly used in Song dynasty in China, and is still actively performed in the *Xi'an Guyue* music tradition. In this paper, the first tool for creating a machine-readable digital representation of *suzipu* notation with focus on optical music recognition (OMR) is proposed. This contribution serves two purposes: i) creating the basis for the future development of OMR methods with respect to *suzipu* notation; and ii) the facilitated digitization of musical sources written in *suzipu* notation. In summary, these purposes promote the preservation and understanding of cultural heritage through digitization.

Index Terms—*suzipu*, *banzipu*, Chinese music, optical music recognition, annotation tool

I. INTRODUCTION

The *suzipu* 俗字谱 (literal meaning: common character notation) or *banzipu* 半字谱 (literal meaning: half character notation) is a notation especially common in China in Song dynasty (960–1279). Unfortunately, only a small number of historical pieces using *suzipu* have been preserved until today.

Jiang Kui (1155-1221) is considered one of the most influential poets of the Song dynasty and his 17 *ci* 词 poems with *suzipu* notation are the largest historical source of this notation. Other examples using it include a Tangut manuscript from 1173 [9], or the *Shilin guangji* 事林广记 encyclopedia [11]. Nowadays, the tradition of *Xi'an Guyue* 西安鼓乐 music, which is still practiced today, also uses *suzipu* notation, having over 1200 pieces [3].

However, no digital representations of *suzipu* notation or OMR methods acting on this notation have been attempted yet. The exploration of these techniques is the necessary foundation for facilitated digitization of ancient Chinese music sources, for the creation of digital corpora of these pieces, and for future musicological research involving computer-aided methods. This is highly beneficial for the preservation and understanding of cultural heritage through digitization.

II. RELATED WORKS

The work related to the research conducted in this paper can roughly be divided in two aspects: since this paper

focuses on the creation of a software annotation tool for the creation of corpora, the first aspect is research centered around the creation of score-based musical corpora; the second aspect deals with digitization methods for Chinese traditional music.

For the first aspect, as representatives for culturally diverse musical corpora, two examples are chosen:

- 1) The datasets from the CompMusic project [6], which have the extraction of features from audio recordings as primary goal. They are also endowed with musical score information in MusicXML format. This format is suitable for music with relatively exact note durations (e.g., Beijing opera and Turkish *Makam* music).
- 2) The *New Bach Chorales Figured Bass Dataset* [5], as a representative of a corpus of music from the common-practice period, consisting of 139 annotated chorales by Johann Sebastian Bach, with digital representations in MusicXML, **kern and MEI formats.

The second aspect, directly targeting digitization techniques for Chinese traditional music, is the smallest out of the two aspects. As representatives, two works are chosen, both focusing on *gongche* notation in *Kunqu* opera: Reference [2] is a comparison of different algorithms for recognition of *gongche* pitches, while [1] focuses on extracting semantic information taking into account the spatial structures of *Kunqu* opera pieces.

III. PRELIMINARIES

A. Pitch System

The foundation for the musical system used in Tang and Song dynasties are the absolute pitch system of the 12 *lǚ* 律吕, acting as the basis for the building of scales; and the relative pitch system consisting of seven scale degrees, which in conjunction with the 12 *lǚ* allow for the creation of scales.

1) *Absolute Pitch System - The 12 Lǚ*: The 12 *lǚ* are an ordered set of 12 pitches (corresponding to the 12 semitones) inside an octave. Following the convention in both [8] and [12], the lowest pitch *Huangzhong* 黄钟 is identified with C₄. In table I, the 12 *lǚ* are introduced, including the correspondence with Western pitch names.

Name	Romanization	Western Equivalent
黄钟	<i>Huangzhong</i>	C ₄
大吕	<i>Dalü</i>	D ₄ ^b
太簇	<i>Taicu</i>	D ₄
夹钟	<i>Jiazhong</i>	E ₄ ^b
姑洗	<i>Guxian</i>	E ₄
仲吕	<i>Zhonglü</i>	F ₄
蕤宾	<i>Ruibin</i>	F ₄ [#]
林钟	<i>Linzhong</i>	G ₄
夷则	<i>Yize</i>	A ₄ ^b
南吕	<i>Nanlü</i>	A ₄
无射	<i>Wuyi</i>	B ₄ ^b
应钟	<i>Yingzhong</i>	B ₄
黄钟清	<i>Huangzhong Qing</i>	C ₅
大吕清	<i>Dalü Qing</i>	D ₅ ^b
太簇清	<i>Taicu Qing</i>	D ₅
夹钟清	<i>Jiazhong Qing</i>	E ₅ ^b

 TABLE I: Listing of the 12 *lülü* with extension to the upper register and pitch equivalents.

Scale Degree Name	Romanization	Interval
宫	<i>Gong</i>	perfect unison
商	<i>Shang</i>	major second
角	<i>Jue</i>	major third
变	<i>Bian</i>	augmented fourth
徵	<i>Zhi</i>	perfect fifth
羽	<i>Yu</i>	major sixth
闰	<i>Run</i>	major seventh

TABLE II: A table of the seven relative scale degrees with the intervals. The pentatonic degrees are printed in boldface.

2) *Relative Pitch System - The Scale Degrees*: In addition to the 12 *lülü*, a relative pitch system consisting of five scale degrees (pentatonic) and two additional scale degrees, thus forming a heptatonic scale (similar to Lydian), is introduced. The names of these degrees are listed in table II according to the convention in [8]. The scale degrees describe fixed intervals, where *Gong* 宫 can be placed on any *lülü*, thus defining a tone inventory, which is called *scale* in this paper.

Therefore, each of these scales is uniquely determined by the *lülü* corresponding to the degree *Gong* 宫, which is also reflected in the naming of the scale, where the name is formed by the *lülü* and an affixed “-*jun*” (“-均”), e.g., *Huangzhongjun* 黄钟均, corresponding to C Lydian.

B. The 28 Modes of Yanyue

While naming of modes are not standardized, each mode is uniquely defined by two properties [12]:

- 1) The scale the mode is based on, as described in III-A2, which is uniquely determined itself by the *lülü* corresponding to its *Gong* 宫.
- 2) The final note of the mode, which is the scale degree (according to table II) a piece in this mode is ending on [12].

For example, the mode *Dao Gong* 道宫 has its *Gong* 宫 on 仲吕 (therefore having the scale 仲吕均) with the

<i>Suzipu</i>	<i>Gongchepu</i>	Romanization	<i>Lülüpu</i>	ASCII
ム	合	<i>He</i>	黄钟	'0'
	下四	<i>Xia Si</i>	大吕	'1'
マ	四	<i>Si</i>	太簇	'2'
	下一	<i>Xia Yi</i>	夹钟	'3'
一	一	<i>Yi</i>	姑洗	'4'
么	上	<i>Shang</i>	仲吕	'5'
ㄥ	勾	<i>Gou</i>	蕤宾	'6'
人	尺	<i>Che</i>	林钟	'7'
	下工	<i>Xia Gong</i>	夷则	'8'
フ	工	<i>Gong</i>	南吕	'9'
	下凡	<i>Xia Fan</i>	无射	'A'
リ	凡	<i>Fan</i>	应钟	'B'
久	六	<i>Liu</i>	黄钟清	'C'
	下五	<i>Xia Wu</i>	大吕清	'D'
ウ	五	<i>Wu</i>	太簇清	'E'
あ	高五	<i>Gao Wu</i>	夹钟清	'F'
<i>Suzipu</i>	Name	Romanization	ASCII	
ウ	大顿	<i>Dadun</i>	','	
リ	小住	<i>Xiaozhu</i>	','	
フ	丁住	<i>Dingzhu</i>	'#'	
カ	大住	<i>Dazhu</i>	','	
ク	折	<i>Zhe</i>	'z'	
ㄩ	拽	<i>Ye</i>	'y'	

 TABLE III: The correspondence between *suzipu*, *gongchepu*, *lülüpu*, and the ASCII character representations. Note that for *suzipu* notation, the values '1', '3', '8', 'A', 'D' are not used.

final note also being *Gong* 宫, therefore conceptually being similar to the F Hypolydian mode.

According to this scheme, 84 modes can be constructed. Usually, pieces of Tang and Song dynasty belong to one of the 28 modes of *Yanyue*, which is a special subset of these 84 modes [12], but already in *Baishidaoren Gequ* there are pieces using other modes, e.g., *Jueshao* 角招 [8].

C. Suzipu Notation

The *suzipu* notation has developed from the cursive writing forms of the *gongchepu* notation [8], but in addition, there is a set of secondary *suzipu* symbols indicating rhythmic or melodic alterations.

For the current work, the *suzipu* notation of *Baishidaoren Gequ* is used as a basis. Table III provides a list of both pitch and secondary *suzipu* symbols.

1) *Pitch Symbols*: As opposed to *gongchepu*, the 16 pitches are represented not by 16 individual symbols, but only by 11, since these five pairs share one *suzipu* symbol each: *Xia Si* 下四 and *Si* 四, *Xia Yi* 下一 and *Yi* 一, *Xia Gong* 下工 and *Gong* 工, *Xia Fan* 下凡 and *Fan* 凡, and finally *Xia Wu* 下五 and *Wu* 五 [8].

This ambiguity of notational symbols is resolved by taking into account the mode the piece is written in [11]. The absolute pitch of any *suzipu* notation can be uniquely inferred from the tone inventory as indicated by the mode. An exception to this is *Linzhongjun* 林钟均, where both *Xia Si* 下四 and *Si* 四 correspond to each *Bian* 变 and *Zhi* 徵, but through prioritizing the pentatonic pitch, the correct correspondence of *Si* ㄩ and *Wu* ㄣ with *Zhi* 徵 is determined.

2) *Secondary Symbols*: Secondary symbols indicate melodic or rhythmic alterations. They cannot occur on their own, only together with a pitch *suzipu* symbol, forming a so-called *pair-character notation*, as opposed to the simple *suzipu* notation only consisting of a single pitch symbol. The number and meaning of these secondary symbols is subject to academic debates due to the scarcity of musical sources from Tang and Song [12].

In this paper, the system of secondary symbols established in [10] is used. It is to be noted that the duration of a simple pitch-only *suzipu* notation is defined to be the basic unit of measurement:

- 1) *Dadun* ㄣ: The length is tripled and a long pause follows.
- 2) *Xiaozhu* ㄩ: The length is doubled and a slight pause follows.
- 3) *Dingzhu* ㄩ (sometimes ㄩ): The note is realized twice, each with simple duration, and a slight pause follows.
- 4) *Dazhu* ㄣ: The length is tripled, and a slight pause follows.
- 5) *Zhe* ㄣ: The note is realized twice, the second time one *lǚ* step higher than the first.
- 6) *Ye* ㄣ: The note is realized twice, the second time two *lǚ* step higher than the first.

The transnotation into modern five-line notation used in the annotation tool is also based on [10]: Each simple duration corresponds to a quarter note, the concept of a slight pause is expressed by a bar line, and the long pause by a quarter rest with a bar line after. The two pitches of *Zhe* ㄣ and *Ye* ㄣ are connected using a slur.

IV. METHODS

A. Digital Representation

MusicXML and similar formats are not well-suited for the flexible rhythms and disputed system of secondary symbols in *suzipu* notations. In order to create a digital representation as close as possible to the *suzipu* notation, a novel system based on JSON is created.

In the root folder of the project's repository, the file `json_schema.json` is a JSON schema giving a technical documentation of the digital representation JSON format used in this paper, and it can also be used to verify that a file is of valid structure.

1) *Metadata Properties*: For the metadata, in the root level, the following piece metadata properties are stored:

- 1) **version**: In this paper, version 1.0 is introduced.
- 2) **notation_type**: The value can be either "Suzipu", "Gongchepu", or "Lvlpvu".
- 3) **composer**: Here, the composer's name is stored.
- 4) **mode_properties**: Containing the two subfields uniquely defining the mode, i.e., the *lǚ* which is corresponding to *Gong* 宫 (stored in `gong_lvlv`), and the mode's final note (stored in `final_note`).
- 5) **images** (optional): This field contains an ordered list of relative paths to the image files belonging to the piece.
- 6) **content**: An ordered list of all the individual boxes in the piece, i.e., title characters, mode information characters, preface characters, and musical boxes consisting of notational units with lyrics. The order inside this list corresponds to the order of the boxes in the piece.

Concerning the **content**, each box may contain textual or notational information, including the possibility to store segmentation box coordinates. The fields are as follows:

- 1) **box_type**: This contains the type label of the box, being either "Title", "Mode", "Preface", "Music", or "Unmarked". E.g., a segmentation box containing a musical character has the type "Music".
- 2) **is_excluded_from_dataset** (optional): Since one of the purposes of this format is to provide the basis for OMR algorithms, the optical information must be free of instances detrimental to model performance, e.g., due to ink splatters or misprints. This field allows such instances to be excluded.
- 3) **is_line_break** (optional): This field is **true** if directly after the box a line break occurs.
- 4) **text_coordinates** (optional): This contains the coordinates of the upper left and lower right corners, uniquely describing the boundary of the segmentation box, and only for the textual information of the box.
- 5) **text_content** (optional): This string is used to store the box's current textual annotation. In case of boxes of type **Music**, the lyrics character is stored here.
- 6) **notation_coordinates** (optional): Similar to **text_coordinates**, but for notational information.
- 7) **notation_content** (optional): This field should only be filled for boxes of type "Music". If a box contains *suzipu* notation, the string may contain up to two characters, the first one being a pitch character and the second one, if present, a secondary character. The correspondences between the *suzipu* notations and the ASCII character representations are in table III.

B. The *Suzipu* Musical Notation Annotation Tool

The *Suzipu* Musical Notation Annotation Tool is introduced to make annotation based on image scores of *suzipu* music as convenient as possible. It is based on Python and tkinter.

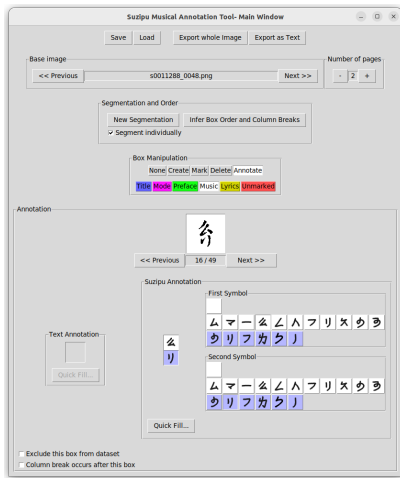


Fig. 1: The Main Window.

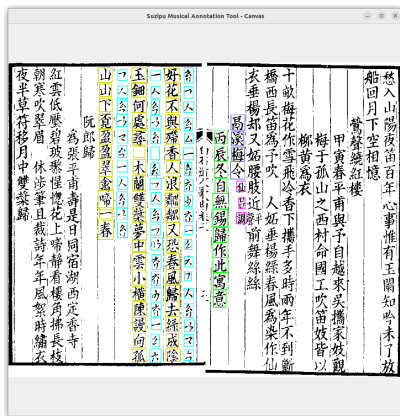


Fig. 2: The Canvas Window.

It consists of three principal windows, the Main Window (figure 1), Canvas Window (figure 2), and Additional Information Window (figure 3).

The Main Window provides features such as the saving and loading of JSON files in the specified format, exporting the piece's contents to text files, or exporting the canvas image. To facilitate the creation of segmentation boxes, the HRCenterNet algorithm [7] can be used. The quick fill functions allow for filling all boxes of a given type simultaneously, instead of filling each box at a time.

The Canvas Window is implemented using OpenCV, and individual segmentation boxes can be selected and manipulated using the right mouse click. Using the mouse wheel and the left mouse button, the scaling and the displayed area can be changed.

The Additional Information Window provides mode information, basic statistics, and provides a real-time view where the pitches are expressed using five line notation or *jianpu* 简谱 notations respectively. Transnotation and export into MusicXML are also supported, which is made possible by the use of the *music21* Python library [4]. Such a transnotation can be seen in figure 4.

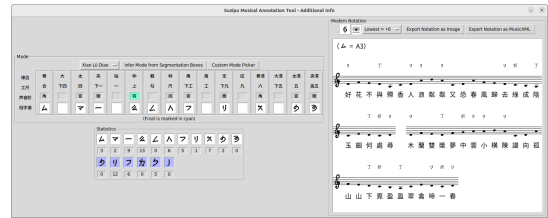


Fig. 3: The Additional Information Window.



Fig. 4: The exported MusicXML transnotation, opened in MuseScore.

V. CONCLUSION AND FUTURE WORK

In this paper, the basics of *suzipu* notation are introduced, including the pitch disambiguation in *suzipu* notation using the piece's mode, and transnotation. With this, the necessity for a new format instead of using established formats (e.g., MusicXML) is clarified.

For the purpose of digital representation of *suzipu* notation, a machine-readable format focused on the development of future OMR techniques is introduced. In addition, the *Suzipu Musical Notation Annotation Tool* for the manual annotation of *suzipu* notation is presented.

The source code of the *Suzipu Musical Notation Annotation Tool* and the JSON schema describing the format is publicly made available¹.

In the future, we propose new research based on the *suzipu* digital representation format and the *Suzipu Musical Notation Annotation Tool* as follows:

- The digitization of corpora of *suzipu*-based music, most notably Jiang Kui's *Baishidaoren Gegu*, and the development of a deep-learning based OMR system for *suzipu* notation, with the annotated *suzipu* characters serving as a dataset.
- The generalization of the annotation tool to support *lülüpu* or *gongchepu* notations.
- The development of a notation editor featuring *suzipu*, *lülüpu* and *gongchepu* notations, which focuses on the input of ancient Chinese notations in a direct way, rather than the score-based annotation in the tool described here.

¹<https://github.com/SuziAI/gui-tools> (Tag: v1.0)

REFERENCES

- [1] Gen-Fang Chen. “Music sheet score recognition of Chinese Gong-che notation based on Deep Learning”. In: *2021 International Conference on Big Data Analysis and Computer Science (BDACS)*. 2021, pp. 183–190. DOI: 10.1109 / BDACS53596 . 2021 . 00048.
- [2] Gen-Fang Chen and Jia-Shing Sheu. “An optical music recognition system for traditional Chinese Kunqu Opera scores written in Gong-Che Notation”. In: *EURASIP Journal on Audio, Speech, and Music Processing* (2014), pp. 7–17. DOI: 10.1186 / 1687-4722-2014-7.
- [3] Yu Cheng. “Xi’an Guyue –Xi’an Old Music in New China. ‘Living fossil’ or ‘flowing river’?” Dissertation. School of Oriental and African Studies, University of London, 2005. URL: <https://eprints.soas.ac.uk/29336/1/10731431.pdf> (visited on 08/03/2023).
- [4] Michael Cuthbert and Christopher Ariza. “Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data.” In: *Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010* (Jan. 2010), pp. 637–642.
- [5] Yaolong Ju et al. “Automatic Figured Bass Annotation Using the New Bach Chorales Figured Bass Dataset”. In: *International Society for Music Information Retrieval Conference*. 2020. URL: <https://api.semanticscholar.org/CorpusID:231627352>.
- [6] Xavier Serra. “Creating research corpora for the computational study of music: The case of the CompMusic project”. In: *Proceedings of the AES International Conference* (Jan. 2014), pp. 1–9.
- [7] Chia-Wei Tang, Chao-Lin Liu, and Po-Sen Chiu. “HRCenterNet: An Anchorless Approach to Chinese Character Segmentation in Historical Documents”. In: *2020 IEEE International Conference on Big Data (Big Data)* (2020), pp. 1924–1930.
- [8] Yaohua Wang 王耀华. *Zhongguo Chuantong Yinyue Yuepuxue 中国传统音乐乐谱学*. Beijing: Renmin Yinyue Chubanshe, [2006] 2020.
- [9] Andrew Christopher West. “Musical Notation for Flute in Tangut Manuscripts”. In: *Tanguty v Central’noj Azii*. Ed. by Irina Popova. Moskva: Vostočnaja literatura, 2012, pp. 443–454.
- [10] Santu Wu 伍三土. “Songci Yinyue Zhuanti Yanjiu 宋词音乐专题研究”. Dissertation. Yangzhou University, 2013.
- [11] Yuanzheng Yang. *Plum Blossom on the Far Side of the Stream. The Renaissance of Jiang Kui’s Lyric Oeuvre with Facsimiles and a New Critical Edition of The Songs of the Whitestone Daoist*. Hong Kong: Hong Kong University Press, 2019.
- [12] Zhu Zhu. “Convergence: Unveiling the Yanyue Modal System of the Tang Dynasty”. Dissertation. Rice University, 2021. URL: <https://scholarship.rice.edu/handle/1911/110421> (visited on 07/20/2023).

The OmniOMR Project

Jan Hajič, jr.

*Institute of Formal and Applied Linguistics (ÚFAL)
Faculty of Mathematics and Physics, Charles University
Prague, Czech Republic
hajicj@ufal.mff.cuni.cz*

Petr Žabička

*Moravian Library
Brno, Czech Republic
petr.zabicka@mzk.cz*

Jan Rychtář

*Trinera, s.r.o.
Brno, Czech Republic
jan.rychtar@trinera.cz*

Jiří Mayer

*ÚFAL
Charles University
Prague, Czech Republic
mayer@ufal.mff.cuni.cz*

Martina Dvořáková

*Moravian Library
Brno, Czech Republic
martina.dvorakova@mzk.cz*

Filip Jebavý

*Moravian Library
Brno, Czech Republic
filip.jebavy@mzk.cz*

Markéta Vlková

*Moravian Library
Brno, Czech Republic
marketa.vlkova@mzk.cz*

Pavel Pecina

*ÚFAL
Charles University
Prague, Czech Republic
pecina@ufal.mff.cuni.cz*

Abstract—In March 2023, the OmniOMR project started with the aim of applying OMR technologies in a digital library environment. The project aims to provide full-text search functionality in musical documents in diverse library collections, covering primarily Common Western Music Notation, but (white) mensural and diastematic chant notations as well, both printed and handwritten. To provide this capability, we aim to build an "OmniOMR Service" – not just for recognition, but also the preceding step of music notation detection and classification, as not all documents that contain music notation are marked in library collections with the appropriate metadata. A significant condition for the project's success is building a dataset that accurately reflects the challenges present in a broad library collection. This report lays out in more detail the goals of the project, its current status, and most importantly, we outline the opportunities for collaboration across the OMR community.

Index Terms—Optical music recognition, digital library, music information retrieval

I. PROJECT OVERVIEW

The Moravian Library, like many others, maintains an extensive digitized collection that contains thousands of known music documents, ranging from medieval chant books over baroque manuscript scores and parts to prints of songs with piano accompaniments for early 20th century households and jazz lead sheets. On top of that, it contains an unknown number of music documents that have so far not been marked as such in the library's metadata scheme, especially in periodicals from the 19th or 20th centuries. In order to provide musical full-text search and music discovery functionality to its users, the library has sought out the OMR team at the Faculty of Mathematics and Physics of Charles University in Prague, resulting in the OmniOMR project.¹

The OmniOMR applied research project aims to integrate OMR technology into the digital library environment to support these goals of providing access to and stewardship of this segment of cultural heritage in the digital domain, with an outlook of generalizing the developed models to other institutions as well. The project runs from March 2023 until

the end of 2027, at a scale of roughly 5 full-time equivalents spread across two institutions: the Faculty of Mathematics and Physics (FMP) of the Charles University, in Prague, and the Moravian Library (MorL), based in Brno. The FMP is responsible for developing a functioning OMR service and overall project leadership, the MorL is responsible for developing search functionality and user interfaces, and for developing the legal framework under which images from as many institutions as possible can be processed and made searchable by the OMR service.

We aim to tackle a wide variety of input conditions (see Fig. 1), but only target the first two levels of comprehension: metadata extraction and search [1].

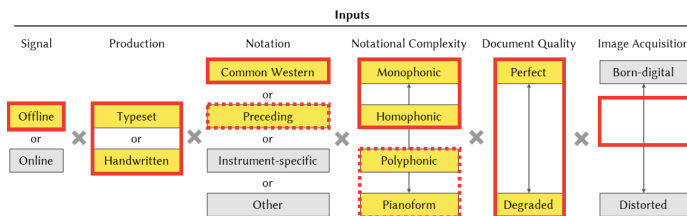


Fig. 1. Targeted music notation input condition.

The project is in its aims comparable to HISPAMUS [2],² although possibly with more weight on polyphonic manuscripts and search, and not as much on MEI/MusicXML output. We aim to utilize the available results of the SIMSSA [3] and OMMR4All projects [4],³ for processing chant notation.

II. OMR COMPONENT

The OMR service will perform detection of music notation and its recognition as two separate steps, because each of these steps is subject to a different set of constraints. The detection step is significantly easier to train, but must be

¹<https://ufal.mff.cuni.cz/grants/omniomr>

²<https://grfia.dlsi.ua.es/hispamus/index.html>

³<https://simssa.ca/>, <https://ommr4all.informatik.uni-wuerzburg.de/en/>

very fast. The MorL collection in which we aim to discover unmarked music notation has approximately 10 million pages. Processing such a collection for one entire calendar year would require a speed of 3 images per second. (Parallelization helps, but computational resources are scarce and the process likely will not run smoothly for an entire year.) Furthermore, due to expected class imbalance (less than 1 in 1000 pages is expected to contain music notation), especially precision on the "contains notation" class must be extremely high – each percentage point of false positives translates to 100 000 pages with music notation falsely detected, likely more than there are actual pages with music notation that do not have the appropriate metadata. Fortunately, music notation has a clear visual identity, so library-quality scans could be compressed without sacrificing accuracy.

The recognition step is subject to less severe computational efficiency and accuracy challenges. However, the task itself is more complex than binary classification and detecting relatively large rectangles. Computational resource constraints again apply, even though only tens of thousands of pages are expected to be processed in the recognition step, as it requires more complex models and may not admit reducing image quality as readily.

III. SEARCH

The second technical aspect of the project is its musical search interface. Designing and selecting distance metrics in the presence of OMR mistakes and polyphonic music is in fact the most open research sub-topic in the project, and at the same time a key element of the user experience. To avoid assumptions of the project team [5], we are conducting a multi-stage user study with a focus group of digital music library users from various stakeholder groups (primarily musicians amateur and professional, teachers, and musicologists) to establish the UX priorities. So far, in the first stage, most significant is the interest in incipit- and keyboard-based search (already explored in the Musiconn project [6] or RISM [7]),⁴ each soliciting enthusiasm from (different) 5 out of the 11 users.⁵ The user study has implications beyond just the search interface: the focus on incipits, for instance, warrants OMR focusing on correctly identify where compositions start on a page (and which pages contain no start).

IV. DATA

No public datasets sufficiently representative of the problem space are available for evaluating, much less training OMR systems. Therefore, while the MorL provides a representative set of digitized images of sheet music, FMP is responsible for annotating at least 1000 of these images with OMR ground truth. We are using the Labelbox web application.⁶ The technical debt of MUSCIMarker [8], is too large, and outsourcing annotation software development and maintenance lightens the

workload on the project development team. Labelbox was selected because it supports relations between objects in images, which is essential for the Music Notation Graph (MuNG) ground truth format [9] (at least for evaluation purposes, as notehead detection accuracy is one of the technical KPIs). Furthermore, Labelbox offers a free license for academic use.

V. OPPORTUNITIES FOR COLLABORATIONS

We see multiple topics on which it seems prudent to consult and collaborate with the OMR community, and some of our outputs can be broadly useful for planning follow-up projects.

Output encodings. It would be a missed opportunity for the OMR community to make the dataset useful only for our limited levels of comprehension. A discussion is urgent on the ground truth formats (particularly improving MuNG, also in light of the new DoReMi dataset [10]), especially other projects that are in the process of data acquisition.

Data. We anticipate publishing the manually annotated data in yearly updates. Furthermore, as we aim for a broad applicability of our models, we are open to directing some of our annotation resources towards images provided by other OMR stakeholders.

Data synthesis. One prerequisite for the success of OmniOMR is building a synthetic data generator that can simulate full pages under the various input conditions and enabling the user to specify a sample document that should be imitated. We are extending the already published Mashcima system [11].

OMR service and indexing for search. We aim to make the recognition service publicly available (up to some manageable scale), with the option to index a user-provided collection alongside all other indexed collections of music documents. This is to enable tasks such as identifying anonymous compositions by comparing them across libraries or mapping transmission of repertoire, such as demonstrated by the F-TEMPO project for menusal part books.⁷ Coordination on persistent music notation image identifiers is thus welcome, especially in light of existing projects' work on a broader systematization of musical culture in the digital domain (i.a. Trompa [12], LinkedMusic [13]),⁸ as well as on legal frameworks for delivering music notation images to users.

User study. Through participation in the user study, we can take into account the manifold priorities of the OMR and the broader music library communities.

It is our hope that the OmniOMR project significantly advances the state of OMR in the following years. In order to best utilize whatever resources we have available to achieve that, we believe that coordination with the broader OMR community is needed – and very much welcome.

ACKNOWLEDGMENT

This paper was supported by the applied research grant of the Czech Ministry of Culture programme NAKI III no. DH23P03OVV008, OmniOMR - optical music recognition using machine learning for digital libraries.

⁴<https://rism.online/?mode=incipits>

⁵Our target focus group size is, however, 20.

⁶<https://labelbox.com/>

⁷<https://f-tempo.org/>

⁸<https://trompamusic.eu/>, <https://linkedmusic.ca/>

REFERENCES

- [1] J. Calvo-Zaragoza, J. Hajič Jr., and A. Pacha, “Understanding optical music recognition,” *ACM Comput. Surv.*, vol. 53, no. 4, 2020.
- [2] J. M. Iñesta, P. J. P. de León, D. Rizo, J. Oncina, L. Micó, J. R. Rico-Juan, C. Pérez-Sancho, and A. Pertusa, “Hispanus: Handwritten spanish music heritage preservation by automatic transcription,” in *1st International workshop on reading music systems*, pp. 17–18, 2018.
- [3] I. Fujinaga, A. Hankinson, and J. E. Cumming, “Introduction to simssa (single interface for music score searching and analysis),” in *Proceedings of the 1st international workshop on digital libraries for musicology*, pp. 1–3, 2014.
- [4] C. Wick and F. Puppe, “Ommr4all-a semiautomatic online editor for medieval music notations,” in *2nd International workshop on reading music systems*, pp. 31–34, 2019.
- [5] A. Green and F. Lawrence, “The shock of the new: Testing the pan-archival linked data catalogue with users.,” in *Proceedings of the Workshops and Doctoral Consortium of the 26th International Conference on Theory and Practice of Digital Libraries*, pp. 108–115, 2022.
- [6] J. Diet, “Innovative mir applications at the bayerische staatsbibliothek,” in *Proceedings of the 5th International Conference on Digital Libraries for Musicology*, 2018.
- [7] K. Keil and L. Pugin, “Das internationale quellenlexikon der musik, rism,” *Bibliothek Forschung und Praxis*, vol. 42, no. 2, pp. 309–318, 2018.
- [8] J. Hajič jr. and P. Pecina, “Groundtruthing (not only) music notation with MUSICMarker: A practical overview,” in *14th International Conference on Document Analysis and Recognition*, (Kyoto, Japan), pp. 47–48, 2017.
- [9] J. Hajič jr. and P. Pecina, “The MUSCIMA++ dataset for handwritten optical music recognition,” in *14th International Conference on Document Analysis and Recognition*, (Kyoto, Japan), pp. 39–46, 2017.
- [10] E. Shatri and G. Fazekas, “Doremi: First glance at a universal omr dataset,” in *Proceedings of the 3rd International Workshop on Reading Music Systems* (J. Calvo-Zaragoza and A. Pacha, eds.), (Alicante, Spain), pp. 43–49, 2021.
- [11] J. Mayer and P. Pecina, “Synthesizing training data for handwritten music recognition,” in *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part III 16*, pp. 626–641, Springer, 2021.
- [12] D. M. Weigl, T. Crawford, A. Gkiokas, W. Goebel, G. Emilia, N. Guti, C. C. Liem, and P. Santos, “Fair interconnection and enrichment of public-domain music resources on the web,” *Empirical Musicology Review*, vol. 16, no. 1, pp. 16–33, 2021.
- [13] L. Pugin and C. Bacciagaluppi, “An analysis of musical work datasets and their current level of linkage,” in *7th International Conference on Digital Libraries for Musicology*, pp. 32–39, 2020.

Towards Music Notation and Lyrics Alignment: Gregorian Chants as Case Study

Juan C. Martinez-Sevilla, Francisco J. Castellanos
 Pattern Recognition and Artificial Intelligence Group
 University of Alicante, Spain
 Email: {jcmartinez.sevilla, francisco.castellanos}@ua.es

Abstract—The Aligned Music Notation and Lyrics Transcription (AMNLT) challenge aims to address the task of extracting content from document images containing vocal music. Current methods can handle both musical notation and text separately, often failing to provide the alignment information. This information is needed to effectively retrieve the content of vocal music pieces. To tackle this challenge we incorporate two key components: an encoding mechanism with explicit alignment information of both information sources and holistic neural models capable of simultaneously extracting this encoding from the image. We focus our case study on Gregorian scores, with the established GABC encoding serving as our target output. Through our experiments, we demonstrate the potential of our approach in addressing the AMNLT task, yielding practical and valuable results.

Index Terms—Optical Music Recognition, Gregorian Scores, Aligned Music Notation and Lyrics Transcription

I. INTRODUCTION

In recent decades, substantial efforts have been dedicated to digitizing musical documents for the sake of preservation, storage, and distribution. In order to make this content accessible, it is necessary to convert the images of these documents into structured digital formats like MusicXML or MEI [1], [2], which encode musical information such as notes, tonality, or dynamics. This not only aids in preserving musical heritage but also opens up possibilities for extensive computational music analysis, content retrieval, and the conversion between various musical notations. Nevertheless, manual encoding of these documents is a time-consuming process that lacks scalability. For these reasons, Optical Music Recognition (OMR), the research field that focuses on automating the extraction of musical content from images, aims to reduce this human effort [3].

In this study, our primary focus lies in automatically transcribing vocal music documents, a domain of particular interest given the wide tradition associated with vocal music. These documents contain both music and text, representing distinct yet complementary sources of information. While music notation and text can be individually transcribed by the respective algorithms—OMR for music and Optical Character Recognition (OCR) for text—a fundamental challenge remains unaddressed: the need to include alignment information between the musical notes and the accompanying text. Retrieving the sequences of notes and syllables independently does not resolve the issue of determining which note or notes correspond to each syllable during vocal performance.

Establishing a meaningful alignment between the notes and syllables is essential for in-depth musicological analysis and efficient content retrieval from vocal music heritage, and it is precisely the Aligned Music Notation and Lyrics Transcription (AMNLT) challenge that addresses this need.

We present a case study using Gregorian music scores in order to address the AMNLT issue. We considered a holistic approach combined with an encoding format, GABC, which is a standard notation used for transcribing this type of manuscript.¹

II. THE AMNLT CHALLENGE

The acronym AMNLT represents the goal of simultaneously extracting both musical and textual content in a single step while acquiring the necessary alignment information between syllables and musical symbols. This novel challenge, introduced by Martinez-Sevilla et al. [4], seeks to contribute to a significant gap in the literature concerning the digitization of vocal music scores where both lyrics and music coexist.

The method presented takes the staff image—including the lyrics—as input and generates an encoding format that explicitly captures the alignment information between both information sources. In this case, we considered the use of GABC for three main reasons: first, because it is a specific format used by experts to transcribe Gregorian music; second, it contains transcription for music and lyrics; and third, it contains inherent alignment information, which is essential for our purpose. Despite the apparent suitability of this encoding, there are several challenges, as depicted in Fig. 1, inherent in AMNLT. For instance, from a pixel-level perspective, syllables may not occupy the same amount of space as their corresponding musical symbols, but this situation is not included in the encoding. Moreover, several syllables may appear within frames associated with other notes, adding complexity even for music experts.

While this paper primarily focuses on the case study of Gregorian scores, the challenge presented is equally applicable to other music notations.

III. METHODOLOGY

One common approach to address AMNLT is to employ a two-step method, where separate systems for music and text

¹<https://gregorio-project.github.io/gabc/> (accessed September 21st, 2023)

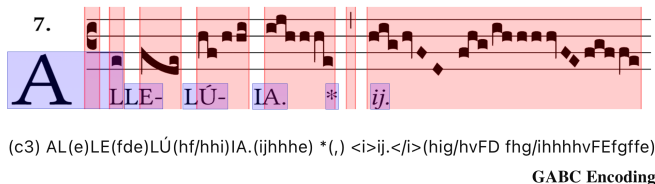


Fig. 1: Example of the alignment challenge between a music-symbol sequence and a text sequence in a Gregorian melody fragment. Red boxes refer to the size inside the image of a music symbol, whereas blue boxes to the syllables. The GABC encoding of the excerpt is also presented under the staff.

are used to transcribe each type of content independently. Subsequently, post-processing operations can be applied to align the output sequences from these systems. However, aligning these sequences poses a considerable challenge that also depends on the individual transcription results. Consequently, devising heuristic algorithms and techniques to combine music and text transcriptions becomes a complex task.

In contrast, in this paper, we adopt a different approach to tackle AMNLT by treating it as an end-to-end sequence retrieval problem. This “holistic” or “end-to-end” formulation involves addressing the recognition problem in a single step, allowing us to learn contextual relationships that are able to enhance the quality of the results. This approach has gained prominence in the current state of the art, particularly in the field of Optical Music Recognition (OMR), where recent literature showcases numerous successful efforts using this paradigm [5]–[7]. In this paper, we extend this end-to-end paradigm in order to address the specific challenges of AMNLT.

A. The challenge of AMNLT with sequence retrieval

In the domain of music and text transcription, a variety of sequence retrieval approaches have been established within the fields of OMR [5]–[7] and Handwritten Text Recognition (HTR) [8]–[10]. These methods typically employ neural architectures designed to directly produce content representations from input images. These architectures consist of two primary components: an encoder and a decoder. The encoder is responsible for extracting pertinent features from the input image, while the decoder models the temporal dependencies among these features to enhance transcription accuracy. A common implementation of such systems involves using a Convolutional Neural Network (CNN) for the encoder and a Recurrent Neural Network (RNN) for the decoder, often referred to as the Convolutional Recurrent Neural Network (CRNN). These networks are trained using the Connectionist Temporal Classification (CTC) loss method [11], which maximizes the likelihood of obtaining a token sequence within a predefined vocabulary, given the information available in the image. This approach is feasible because it assumes that each

image frame contains information from a single symbol.²

However, when dealing with the challenge of AMNLT, the simultaneous transcription of both music and lyrics presents a unique hurdle for these methods. This is because multiple elements, as illustrated in Fig. 1, can be found within the same image frame. This complexity makes it challenging to predict each element in very closely spaced “timesteps”, a critical factor when attempting aligned transcriptions. Consequently, state-of-the-art models may face difficulties in performing these transcriptions due to technical constraints.

B. Reformulating the process of reading scores

For a given staff, a textual line can be generated as a sequence of these music-syllable groups, as illustrated in Fig. 2. Following this approach, graphic alignment with the input source can also be achieved by rotating the score image 90° clockwise. This alignment information essentially allows music scores to be read like text paragraphs within their original document notation, paving the way for a solution through multi-line transcription.

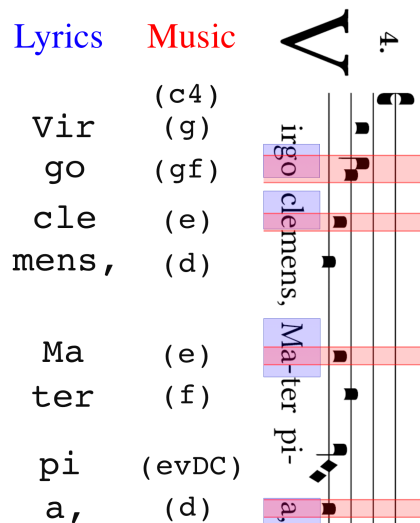


Fig. 2: Gregorian chanted melody fragment with its alignment information in GABC format. The colored highlighted boxes show some examples of alignment.

C. Image unfolding

Our approach is based on the reinterpretation discussed in Section III-B. We adapt and extend recent document unfolding techniques from HTR [12], [13] to accommodate the simultaneous processing of both music notation and lyrics rows.

In contrast to the typical vertical collapse of the feature map, we reshape it by concatenating all its rows, as in the approach in [12]. An illustration of this method is presented in Fig. 3. The resulting sequence has dimensions $(c, h \times w)$, where c represents the number of filters in the last convolutional layer

²Here, a “frame” refers to the column-wise elements of the image.

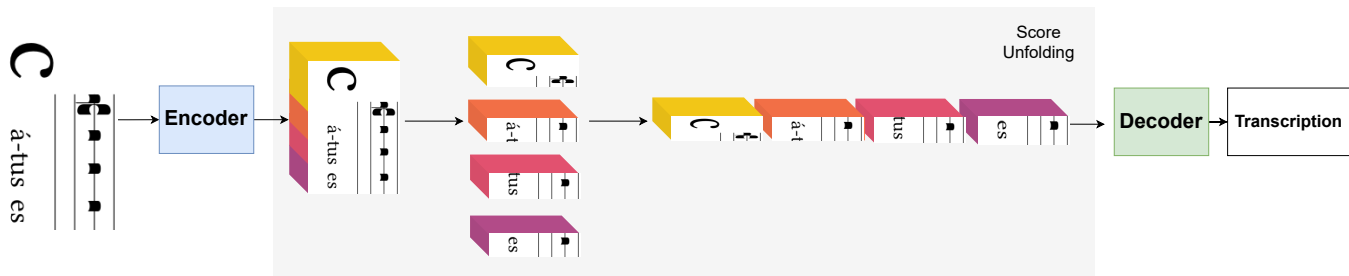


Fig. 3: Graphical scheme of the implemented neural method to address AMNLT.

of the model, h is the height, and w is the width of the feature map. The network takes, as its input, a music staff image rotated by 90° and produces the required AMNLT vocabulary Σ , along with a blank CTC token ϵ used to denote time step separations, forming $\Sigma' = \Sigma \cup \epsilon$.

This method effectively learns to reconstruct the score through the unfolding process without the need for custom layout tokens. During training, the network is forced to align different syllables and music groups, facilitating the transcription of both music and lyrics into a single unified sequence.

IV. EXPERIMENTAL SETUP

A. Dataset

This study is focused on transcribing Gregorian Scores, which are musical documents well-suited for the AMNLT challenge since they contain square notation and lyrics information. In this section, we provide an overview of the notation encoding used for transcribing these scores and describe the corpus utilized in our experiments.

We employ the GABC music encoding format to label the ground truth transcriptions within our corpus. It represents both musical notation and textual information within a single file. To denote the alignment of music and lyrics, the GABC format encapsulates all musical elements to be sung within parentheses following the respective syllable, as depicted in Fig. 1. This choice of the encoding format allows us to seamlessly integrate music and lyrics, thereby ensuring that the alignment information is accurately represented, a fundamental requirement for the successful transcription in AMNLT.

We considered in this paper the Gregorian Chant Database—hereafter referred to as GREGOBASE³—which is a well-known large database of Gregorian music scores. The GREGOBASE corpus contains 17884 chants. As this paper focuses on individual staff images, the data was preprocessed in order to conduct the experiments by extracting fragments of the information GABC and rendering their corresponding individual staff images. GregorioTex⁴ digital engraver was used for obtaining the images from the encoding.

In addition, we created the *distorted* version of the corpus, which aims to simulate adverse conditions associated

with the passage of time for physical documents, where models can struggle to perform transcription. This degradation has been obtained through the application of a random sequence of minor graphical procedures such as blurring, erosion, or rotation as depicted in Fig. 4.

(f3) cá(h) tus(h) es(h) ve(h/l/)<i>he</i>(g) mén(e fr) ter.(f.) (::)
 (<i>Flexa :</i> dux(h) est(h) e(h) ó(h fr) ru:(f.)+ () (h) (h) (::)

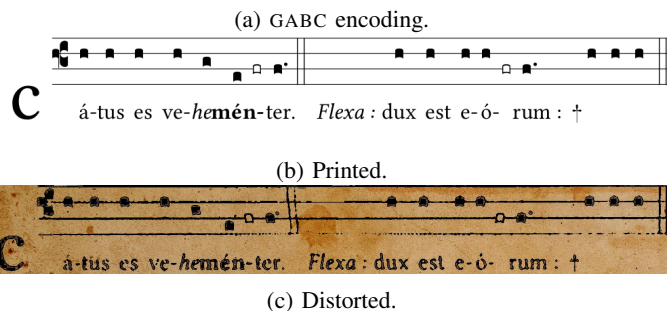


Fig. 4: Example of a GABC sequence with its corresponding staff image from the GREGOBASE corpus and its distorted version.

1) *GABC Tokenization*: We explore the use of GABC music encoding, which has not been previously used in the field of AMNLT. To evaluate the impact of this encoding on the performance of the models that have been analyzed, we encoded the ground-truth sequences into the variants: (i) Character Encoding, as a raw character sequence without differentiating between music and lyrics; (ii) Music-Aware Encoding, indicating music tokens with additional starting-ending tokens to separate both sources; and (iii) Learned Encoding, which uses the Sentence Piece strategy [14], commonly used in the Machine Translation field, where the text is tokenized into sub-word units.

B. Metrics

Since the result of our approach is a sequence that represents the music and lyrics alignment in text format, we considered using a typical and standard metric widely applied in Optical Character Recognition (OCR): Character Error Rate (CER). This metric is computed through the edit distance between a hypothesis sequence—prediction—referred to as \hat{s} and its corresponding reference sequence—ground truth—denoted as s .

³<https://gregobase.selapa.net> (accessed September 21st, 2023)

⁴<https://gregorio-project.github.io/gregoriotex/> (accessed September 21st, 2023)

C. Implementation

We conducted the experiments in order to explore the feasibility and adaptability of existing state-of-the-art transcription methods, successfully applied in OMR but also in other related contexts like OCR. Therefore, we adapted several state-of-the-art models to perform AMNLT.

Recent research in the literature tends to use a combination of convolutional and sub-sampling layers as feature extractors, while different strategies can be used to process these features. In our experiments, we considered fixing this feature extraction and applying one type of decoder commonly referred to in the literature: the Transformer-based approach, so-called Convolutional Neural Network Transformer (CNNT). The Transformer [15] is achieving significant relevance in recent research of handwritten text transcription tasks [10], a very related task to the focus of our contribution. We implemented an encoder block with an embedding size of 512, with 8 attention heads and 1024 of feed-forward dimension.

In addition, another existing strategy in multi-line HTR transcription is that proposed by Coquenot et al. [12], in which a method based on convolutional architectures without sequence processing was applied. The method predicts a 2D representation through back-propagation directly applied to the feature map without any sequence processing block. Since this method could be also useful for the AMNLT context, we implemented it with fully convolutional layers. Henceforth, we denote this strategy as FCN.

V. RESULTS

Table I provides an overview of our results on the test partition, following the experimental framework, and presenting them in terms of CER (%). We explored the application of various models within two scenarios outlined in Section IV-A: *printed* and *distorted*. Furthermore, as detailed in Section IV-A1, we considered three tokenization approaches for encoding GABC sequences: character encoding, music-aware encoding, and learned encoding.

Regarding the outcomes for the *printed* scenario, the optimal performance was achieved using music-aware encoding, with a CER of 16.4% by the FCN model, indicating this tokenization encoding as suitable for the AMNLT task. Conversely, character-based tokenization yielded the least favorable results for both models, with FCN notably performing poorly at 62.3% CER. We can observe that the performance of the FCN model ranges significantly depending on the chosen encoding. Similarly, the CNNT model with the character-based encoding achieved a 57.9% CER, which significantly improved to 18.8% when music-aware encoding was employed. Notably, learned encoding also enhanced the character-based scenario, reducing FCN’s CER from 62.3% to 39.7% and CNNT’s CER from 57.9% to 28.5%. The results with learned encoding matched the performance of music-aware encoding, suggesting that, according to our experiments, the optimal approach for addressing this AMNLT case is through the utilization of music-aware encoding.

Scenario	CER (%)		
	Tokenization	FCN	CNNT
<i>Printed</i>			
	Character	62.3	57.9
	Music-aware	16.4	18.8
	Learned	39.7	28.5
<i>Distorted</i>			
	Character	60.5	71.3
	Music-aware	48.6	34.1
	Learned	55.3	46

TABLE I: Results in terms of CER (%) of the transcription of GABC encoding.

In the more challenging *distorted* scenario, transcription becomes notably harder, with the CNNT model achieving the best-reported performance at 34.1% CER. This scenario represents extremely difficult cases involving heavily degraded music scores. Nevertheless, our approach still yields results that can be considered useful. It is worth highlighting that, despite the significant drop in performance, the conclusions drawn align with those in the *printed* scenario, emphasizing the importance of employing an encoding method that effectively distinguishes between music and lyrics elements.

In summary, our results underscore the considerable impact of tokenization methodology on transcription performance. Music-aware encoding emerges as the most advantageous approach in these experiments, with FCN demonstrating the best results. However, CNNT also delivers competitive results, making it a model of consideration for this application.

VI. CONCLUSION

This work has established that AMNLT can be effectively tackled via end-to-end sequence retrieval techniques. Nevertheless, we have also identified areas for future research. One such refers to enhancing the encoding of music scores, a crucial factor influencing transcription accuracy. Additionally, we advocate for exploring alternative pipelines that combine two-step strategies with automated alignment methods, promising further advancements in transcription precision. In summary, this work aims to contribute to a very recent research field, AMNLT, and provide a reference for potential future work related to this field.

ACKNOWLEDGMENT

This paper is part of REPERTORIUM project, funded by the *European Union’s Horizon Europe* program under grant agreement No 101095065.

REFERENCES

[1] M. Good et al., “Musicxml: An internet-friendly format for sheet music,” in *Xml conference and expo*. Citeseer, 2001, pp. 03–04.

- [2] A. Hankinson, P. Roland, and I. Fujinaga, “The music encoding initiative as a document-encoding framework,” in *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011*, A. Klapuri and C. Leider, Eds. University of Miami, 2011, pp. 293–298.
- [3] J. Calvo-Zaragoza, J. H. Jr, and A. Pacha, “Understanding optical music recognition,” *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–35, 2020.
- [4] J. C. Martinez-Sevilla, A. Rios-Vila, F. J. Castellanos, and J. Calvo-Zaragoza, “A holistic approach for aligned music and lyrics transcription,” in *Document Analysis and Recognition - ICDAR 2023*, G. A. Fink, R. Jain, K. Kise, and R. Zanibbi, Eds. Cham: Springer Nature Switzerland, 2023, pp. 185–201.
- [5] J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, “Handwritten music recognition for mensural notation with convolutional recurrent neural networks,” *Pattern Recognition Letters*, vol. 128, pp. 115–121, 2019.
- [6] P. Torras, A. Baró, L. Kang, and A. Fornés, “On the integration of language models into sequence to sequence architectures for handwritten music recognition,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, Online, November 7-12, 2021*.
- [7] M. Alfaro-Contreras, A. Ríos-Vila, J. J. Valero-Mas, J. M. Iñesta, and J. Calvo-Zaragoza, “Decoupling music notation to improve end-to-end optical music recognition,” *Pattern Recognition Letters*, vol. 158, pp. 157–163, 2022.
- [8] J. Puigcerver and C. Mocholí, “Pylaia,” <https://github.com/jpuigcerver/PyLaia>, 2018.
- [9] M. Yousef, K. F. Hussain, and U. S. Mohammed, “Accurate, data-efficient, unconstrained text recognition with convolutional neural networks,” *Pattern Recognition*, vol. 108, p. 107482, 2020.
- [10] L. Kang, P. Riba, M. Rusiñol, A. Fornés, and M. Villegas, “Pay attention to what you read: Non-recurrent handwritten text-line recognition,” *Pattern Recognition*, vol. 129, p. 108766, 2022.
- [11] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the Twenty-Third International Conference on Machine Learning, (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, 2006, pp. 369–376.
- [12] D. Coquenat, C. Chatelain, and T. Paquet, “Span: A simple predict & align network for handwritten paragraph recognition,” in *16th International Conference on Document Analysis and Recognition, ICDAR*, ser. Lecture Notes in Computer Science, vol. 12823, 2021, pp. 70–84.
- [13] M. Yousef and T. E. Bishop, “Origaminet: Weakly-supervised, segmentation-free, one-step, full page textrecognition by learning to unfold,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [14] T. Kudo and J. Richardson, “SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71.
- [15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

Symbol Generation via Autoencoders for Handwritten Music Synthesis

Jonáš Havelka, Jiří Mayer (✉), Pavel Pecina

Institute of Formal and Applied Linguistics

Charles University, Prague, Czech Republic

Email: jonas.havelka@volny.cz, mayer@ufal.mff.cuni.cz, pecina@ufal.mff.cuni.cz

ORCID: 0000-0002-4718-3372, 0000-0001-6503-3442, 0000-0002-1855-5931

Abstract—Optical Music Recognition is one of the fields where synthetic data is effectively utilized for training deep learning recognition models. Due to the lack of manually annotated data, the training data is generated by an automatic procedure which produces real-looking images of music scores in large quantities. Mashcima, a system for synthesizing training data for handwritten music recognition, generates complete music scores but the individual symbols are not synthetic, they are sampled from real symbol datasets. In this paper, we explore the impact of utilizing an adversarial autoencoder within the symbol synthesis pipeline. We show that in some cases the use of an autoencoder may not only be motivated by the creation of latent-space symbol embeddings but also by improved recognition accuracy.

Index Terms—Optical Music Recognition, Synthetic Training Data, Data Augmentation, Deep Learning

I. INTRODUCTION

Synthetic training data is used in many areas of computer vision to train deep neural models, especially for tasks where obtaining sufficient amounts of manually annotated training data is prohibitively costly. This includes, e.g., handwritten text recognition [1]–[4], natural scene recognition [5], optical flow estimation [6], [7], and Optical Music Recognition (OMR) where several synthetic datasets of typeset (i.e. not handwritten) music were published recently, e.g., PrIMuS, DeepScores, DoReMi, or the Baró’s dataset of historical documents [8]–[12].

For handwritten music, a synthesizer called Mashcima [13] was proposed. The current version is capable of synthesizing binarized images of singular staves of monophonic music. It exploits existing images of individual symbols and places them onto an empty staff according to a given annotation. The symbol images are taken as-is from the MUSCIMA++ dataset [15], [16]. The synthesizing process is performed in three stages as described by Mayer et al. [14]: 1) Symbol synthesis, 2) Layout synthesis, 3) Postprocessing.

In this paper, we tackle the first stage of the Mashcima’s pipeline and try to improve the synthesis by generating symbol images using adversarial autoencoders (AAE) [18]. For com-

This work described in this paper has been supported by the Czech Ministry of Culture (project NAKI no. DH23P03OVV008), Charles University (project GA UK no. 289623), and has been using data provided by the LINDAT/CLARIAH-CZ Research Infrastructure (<https://lindat.cz>), supported by the Ministry of Education, Youth and Sports of the Czech Republic (project no. LM2018101).

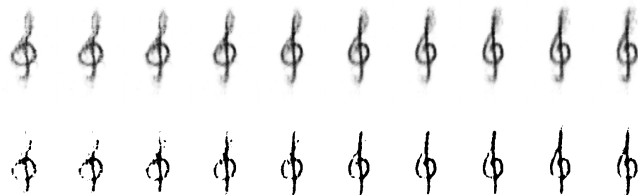


Fig. 1. Symbol interpolation using an autoencoder. Raw autoencoder output (top) and binarized using 0.5 threshold (bottom).

parison, we evaluate this approach using the same recognition model and datasets as in the original Mashcima paper [13].

II. METHODOLOGY

An autoencoder is an unsupervised model that learns to compress given data to a lower-dimensional (latent) space while preserving the underlying structure of the data [21]. The model extracts useful characteristics of the data along the latent space dimensions (e.g., symbol slant, size, thickness) and then can generate symbols it has not seen in the training data, by doing interpolation in the latent space (see Figure 1).

There are many ways to build an autoencoder, utilizing convolutional networks, adversarial loss, or variational loss [18]–[21]. To limit the scope of this paper, we decided to explore adversarial autoencoders only (AAE) [18], since our early experiments showed a marginally better performance over variational autoencoders (VAE) [19].

In our experiments, we employ an AAE model with the encoder producing an L -dimensional continuous embedding vector and a C -dimensional categorical vector containing the symbol class (see Table II for the complete list). The decoder takes these two vectors and reconstructs the input image. The discriminator works only with the continuous embedding vector and forces it to have the standard normal distribution. The categorical vector is trained in a supervised fashion. Both the encoder and the decoder are convolutional (with the exception of the inner-most layer), and the discriminator is a single-hidden-layer fully-connected network. The model is described in more detail in Figure 2 and Table I.

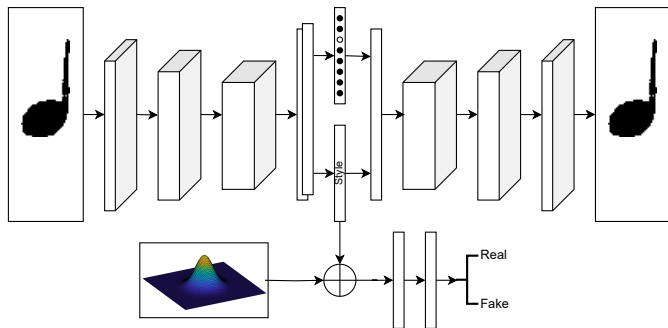


Fig. 2. Architecture of our AAE model. The input symbol (e.g., a quarter note) has its origin (notehead center) aligned with the center of the canvas. The canvas resolution is 126x459 pixels. The encoder models a Gaussian posterior, so the output is a Gaussian mean and a standard deviation vectors that are then sampled. The embedding vector has a categorical and a continuous part and the continuous part is shaped by a fully-connected discriminator network.

A. Symbol dataset mixing

To allow comparison with the Mashcima baseline [13], we use its subset of the MUSCIMA++ dataset as the training dataset for our autoencoder. MUSCIMA++ contains handwritten symbol images from the CVC-MUSCIMA dataset [16]. We also use the Rebelo symbol dataset [17] to expand the amount of symbols for training. The Rebelo dataset contains typeset symbols, which should help with the performance on the PrIMuS typeset evaluation dataset, as defined in the original Mashcima paper [13]. To explore all the effects, we mix these datasets in various different ways (see Figure 5) but we always generate the same amount of symbols as there are in the default Mashcima symbol repository (for each symbol category separately). These counts can be seen in the first column of Table II. These Mashcima symbols are all the available MUSCIMA++ symbols, except for those that come from writers used in the CVC-MUSCIMA evaluation dataset, as defined in the original Mashcima paper (writers 13, 17, 20, 34, 41, 49) [13].

Whenever we mix two symbol sources, we do this in a one-to-one ratio (represented by the \oplus symbol in Figure 5). If one of the two sources does not have enough symbols, we repeatedly loop over the source. This means that symbols from the Rebelo dataset may be used multiple times. The training dataset for an autoencoder is constructed by these same rules as the final symbol repository used for the music score synthesis, unless the autoencoder is trained from a single dataset, in which case we just use the dataset as is, with no repetition nor mixing. Whenever a symbol needs to be obtained from an autoencoder, its latent space is sampled via the standard normal distribution and then decoded. It is also binarized using a 0.5 threshold, so that the synthetic image has the same appearance as the original MUSCIMA++ and Rebelo images.

B. Symbol selection and image preparation

In music notation, there are many dot-like and line-like symbols that are unsuitable for generation by convolutional

TABLE I
LAYERS OF THE ADVERSARIAL AUTOENCODER.

Layer	Shape	Note
Input	$H \times W \times 1$	
Convolution	$H/3 \times W/3 \times 4$	Kernel 5×5 , Stride 3
Convolution	$H/9 \times W/9 \times 16$	Kernel 5×5 , Stride 3
Convolution	$H/27 \times W/27 \times 64$	Kernel 5×5 , Stride 3
Flatten	$H/27 \times W/27 \times 64$	
(2x) Fully connected	$L + C$	(one for σ , one for μ)
Normal distribution	$L + C$	
Latent space		
Fully connected	$H/27 \times W/27 \times 64$	
Reshape	$H/27 \times W/27 \times 64$	
Transposed conv	$H/9 \times W/9 \times 16$	Kernel 5×5 , Stride 3
Transposed conv	$H/3 \times W/3 \times 4$	Kernel 5×5 , Stride 3
Transposed conv	$H \times W \times 1$	Kernel 5×5 , Stride 3

The discriminator takes in only the continuous embedding vector part (of size L) and applies two fully connected layers with dimensions 128 and 1 (the second one being the output layer). H and W stand for canvas width and height respectively and L and C stand for the sizes of the continuous and categorical embedding vector parts. C is equal to the number of symbol classes trained.

The last layer of both the decoder and the discriminator use the sigmoid activation function. The layer generating standard deviation (σ) has the exponential activation function, and the one generating mean (μ) has no activation function (linear). All other layers use the ReLU activation function. Each inner convolution layer (including transposed convolutions) is also followed by a batch normalization layer.

neural networks. We do not synthesize these. From the rest of the music symbols, we choose to generate only those easily obtainable from the Rebelo dataset [17]. The resulting 12 symbol classes are listed in Table II.

The Mashcima synthesizer requires each symbol to have its origin point located. This is often the point that is aligned with the staff lines. MUSCIMA++ already has origins computed, but we need to do so for Rebelo as well. For quarter rests, whole notes¹, C-clefs, we choose the center of the image as the origin point. In accidentals and half notes, we find the "hole" by traditional computer vision methods and pick its center. In the quarter and eighth notes, we find the point that's most distant from the symbol edge. Finally, for F-clef, we choose the rightmost point of the non-dotted part situated vertically between the dots, and for G-clef, we choose the center of the most "circular" part of the image found by the Hough circles filter [22], [23].

We align all symbols such that their origin is in the center of the image from the perspective of the autoencoder. This way, we do not need to predict the origin via regression. Notes with stems are also normalized such that the stem always points up (eighth notes are separated into two symbol classes because the flag either points left or right). We also find the tip of the

¹Whole notes are not present in the Rebelo dataset, but we created them from half notes by removing the stem.

TABLE II
SYMBOL COUNTS IN UTILIZED SYMBOL DATASETS

Sybmol	MUSCIMA++	REBELO proc.	(REBELO)
Quarter note	15 424	634	(873)
Eighth note (up)	1 697	173	(395)
Eighth note (down)		105	
Sharp	1 689	706	(761)
Whole note	1 183	537	(—)
Flat	1 064	544	(559)
Natural	1 021	589	(639)
Half note	845	541	(625)
Quarter rest	553	389	(389)
G-clef	341	407	(414)
F-clef	250	38	(38)
C-clef	155	130	(130)

quarter note’s stem (for the synthesis of beams) as the topmost pixel of the note’s image.

We do not adjust for DPI since both datasets have comparable symbol sizes (in pixels). There are some symbols from Rebelo that appear small, possibly due to a lower DPI, but it only makes the dataset more diverse and the resulting recognition model more robust. The autoencoder produces images of size 162x459 pixels, which is the smallest rectangle that fits all centered training images.

C. Autoencoder training

With the symbols prepared, the autoencoder can be trained. The training process is described in the AAE paper in the supervised section [18]. We use the Adam optimizer [24] with the learning rate 0.001 and the decays of momentum 0.9 and 0.999. We train it for 150 epochs and use batches of 50 images. Occasionally, the autoencoder has problems converging, which manifests by producing black images. We sidestepped this problem by running the experiments with multiple seeds and taking only those that converged.

D. Synthetic symbol evaluation

Once the autoencoder is trained, we use it to synthesize the final set of symbols used for music score synthesis. We modified the Mashcima code to load our own symbols instead of the MUSCIMA++ symbols. The rest of the synthesizer remained unmodified.

Mashcima uses the PrIMuS dataset for obtaining the training melodies (annotations). The second half of the annotations is randomly generated. These annotations are used with the synthetic symbols to produce the synthetic images. All of this is identical to the original version of Mashcima [13]. We use only half the available PrIMuS annotations, analogous to the experiment 3 of the paper. Our experiment A is the exact replication of the experiment 3 from the Mashcima paper.

The recognition model is a CRNN neural network, that takes in the entire staves of music and returns the corresponding annotation (a sequence of tokens in the Mashcima encoding). The evaluation is performed using Symbol Error Rate (SER), which is the number of mistakes divided by the length of the gold annotation. The number of mistakes is computed as the Levenshtein distance [25].



Fig. 3. Symbols sampled from the AAE with latent dimensions from left to right: 2, 2, 4, 4, 6, 6, 8, 8, 10, 10. As the dimension increases so does variability but also noisiness.

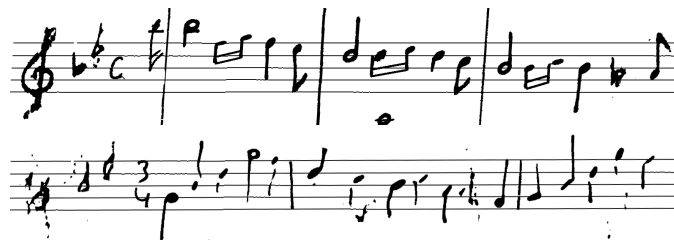


Fig. 4. Staves with symbols generated by AAE with dimension 2 (top) and 10 (bottom). The smaller dimension has prettier symbols with minimal noise. The larger dimension has symbols more noisy and some are unrecognizable (e.g. the clef). Quarter notes are also noisy, which is not the case above.

The evaluation datasets are identical to those from the Mashcima paper, namely a subset of the CVC-MUSCIMA dataset (has the same appearance as the MUSCIMA++ symbols), a subset of the typeset images from the PrIMuS dataset (has typeset appearance), and three music sheets of handwritten music titled Cavatina (referred to as *out-of-domain* test set by the original paper). This makes our experiments directly comparable with that paper.

III. EXPERIMENTS

There are many questions we want to explore. We want to know what impact has the addition of an autoencoder to the symbol synthesis. We utilize two datasets, one handwritten, one typeset. We want to explore how their combination affects the performance. We also want to find the optimal dimension of the autoencoder’s latent space.

The following experiments differ in the way in which the symbol repository is prepared, but then the music image synthesis, training of the recognition model, and its evaluation are identical for all of them. Each experiment is executed 4 times, and the mean and standard deviation are computed.

A. Latent space dimension

We initially wanted to explore dimensions 2, 5, 10, 20, 50, 100, 200, 500, 1000, but experiments with dimension 20 and above never converged. The autoencoder always produced black images. Therefore, we only explore dimensions 2, 4, 6, 8, 10. We evaluate using the same setup as in experiment E in Figure 5.

By looking at the synthetic images for various latent space dimensions, one can see that smaller dimensions make the symbols better looking, but more uniform. Larger dimensions make them more diverse, but also more faulty, noisy, and less comprehensible.

TABLE III
ERROR RATE (%) FOR GIVEN AUTOENCODER LATENT SPACE DIMENSION

Dimension	CVC-MUSCIMA	PrIMuS	Cavatina
2	39.33 ± 0.47	75.10 ± 2.51	65.81 ± 1.77
4	32.05 ± 1.28	64.15 ± 2.66	64.50 ± 1.59
6	32.66 ± 0.66	58.53 ± 7.01	59.76 ± 1.60
8	30.54 ± 1.16	59.67 ± 7.20	57.22 ± 1.94
10	30.05 ± 0.70	57.48 ± 1.59	55.96 ± 2.67

One would expect that the presence of incomprehensible symbols would worsen the recognition model performance, but the evaluation results (Table III) show the opposite effect across all three evaluation datasets. This means that synthetic symbols need not be realistic or good-looking in order to be useful.

Based on the results, we fix the latent space dimension at 10 for all of the following experiments.

B. Topology experiments

To explore the effect of the autoencoder and each dataset, we designed a set of eight experiments. The way in which the datasets are combined, autoencoder trained, and then sampled is captured in Figure 5. Results of these experiments are present in Table IV.

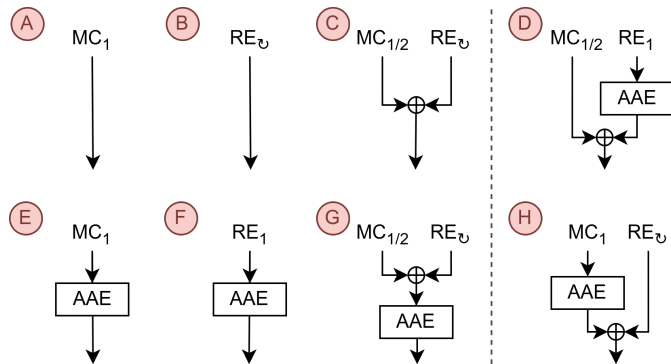


Fig. 5. Topologies of performed experiments. MC and RE are MUSCIMA++ and Rebelo symbol datasets. The subscript is the portion of the dataset used. Circular arrow means we sample the dataset more than once. Oplus combines symbols one to one. AAE is trained by incoming arrows and sampled by outgoing arrows. The final symbols used for synthesis exit the diagram at the very bottom. Looking at the experiments: going from top to bottom, we add the autoencoder; going from left to right we combine datasets in various ways. Experiments D and H are something in between so they are placed aside.

First, we consider the effect of adding an autoencoder, which corresponds to comparing experiments A-C to E-G. You can see the error rate increase on the CVC-MUSCIMA evaluation dataset, when the symbols from MUSCIMA++ are used (A,E and C,G). The same effect happens on the Cavatina test set. However, on the PrIMuS test set, the error rate decreases (for A,E and C,G). It seems that when the training and evaluation symbols come from the same domain, the autoencoder does not help, but it helps with generalization to other domains.

If we look at the experiments that use only Rebelo symbols (B,F), we see an error rate decrease for the PrIMuS test set

TABLE IV
ERROR RATES (%) FOR PROPOSED SYMBOL GENERATION TOPOLOGIES

	A	B	C	D
M:	25.02 ± 1.19	39.47 ± 0.83	26.98 ± 1.49	24.67 ± 0.89
	30.81 ± 0.86	38.19 ± 0.83	33.29 ± 2.19	30.50 ± 0.56
	E	F	G	H
P:	61.17 ± 6.23	77.60 ± 7.84	62.63 ± 6.78	53.85 ± 5.21
	58.62 ± 8.10	67.03 ± 2.98	52.61 ± 1.24	58.29 ± 6.85
	E	F	G	H
C:	49.65 ± 1.61	57.17 ± 2.56	50.17 ± 2.42	49.14 ± 1.29
	56.23 ± 1.49	57.30 ± 1.51	57.02 ± 2.60	53.43 ± 2.61
	E	F	G	H

M = CVC-MUSCIMA P = PrIMuS C = Cavatina

(and a stagnation for CVC-MUSCIMA and Cavatina). This is a decrease for the same domain evaluation. But this might be caused by the differences in the size of the MUSCIMA++ and Rebelo datasets (see Table II). For a smaller dataset, like Rebelo, the autoencoder seems to help, but for a larger dataset, like MUSCIMA++, it seems to hurt.

This hypothesis also holds when we look at the best topologies for a given test set (table numbers in bold). For CVC-MUSCIMA and Cavatina, the best result is obtained by experiment D. For PrIMuS by experiment G. The D experiment uses MUSCIMA++ without an autoencoder, but Rebelo with an autoencoder, thus avoiding the MUSCIMA++ autoencoder harm, and utilizing the Rebelo autoencoder generalization help. The G experiment utilizes one autoencoder that creates the generalization effect on Rebelo and also generalizes using MUSCIMA++ without overfitting on it (like the experiment D does).

The last thing to note is that the experiment A is the baseline from the original Mashcima paper [13], and we managed to surpass it on all three evaluation datasets (exp. D or G). Moreover, the autoencoder played an important role in this, because the simple dataset combination (exp. C) is always worse than the baseline. This indicates that the best way to incorporate smaller symbol datasets (such as Rebelo) into the synthesis pipeline is via autoencoder training.

IV. CONCLUSION

While utilizing autoencoders for music symbol synthesis is not a silver bullet, we showed that they are helpful in situations where the symbol dataset is small, or the evaluation dataset comes from a different domain than the training dataset. Another important observation is that although the synthetic symbols may look noisy they probably make the recognition model more robust. The use of autoencoders may also be motivated by other reasons, such as the use of the latent space for controlling the handwriting style. This possibility may be explored in some future work.

REFERENCES

- [1] Adrià Rico Blanes, “Synthetic handwritten text generation” 2018
- [2] Eloi Alonso, Bastien Moysset, and Ronaldo Messina, “Adversarial Generation of Handwritten Text Images Conditioned on Sequences” 15th IAPR International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 2019, pp. 481-486
- [3] C. V. Jawahar, and Shankar Balasubramanian, “Synthesis of Online Handwriting in Indian Languages” 10th International Workshop on Frontiers in Handwriting Recognition, La Baule, France, 2006
- [4] Alex Graves (2014) “Generating Sequences With Recurrent Neural Networks” arXiv preprint arXiv:1308.0850
- [5] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman (2014) “Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition” arXiv preprint arXiv:1406.2227
- [6] Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox, “What Makes Good Synthetic Training Data for Learning Disparity and Optical Flow Estimation?” *International Journal of Computer Vision*, vol. 126, pp. 942-960, 2018
- [7] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox, “FlowNet: Learning Optical Flow with Convolutional Networks” *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 2758-2766
- [8] Jorge Calvo-Zaragoza, and David Rizo, “End-to-End Neural Optical Music Recognition of Monophonic Scores” *Applied Sciences*, vol. 8, no. 4, pp. 606, 2018
- [9] Jorge Calvo-Zaragoza, and David Rizo, “Camera-PrIMuS: Neural End-to-End Optical Music Recognition on Realistic Monophonic Scores” 19th International Society for Music Information Retrieval Conference (ISMIR), Paris, France, 2018, pp. 248-255
- [10] Lukas Tuggener, Yvan Putra Satyawan, Alexander Pacha, Jürgen Schmidhuber, and Thilo Stadelmann, “The DeepScoresV2 Dataset and Benchmark for Music Object Detection” 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 2021, pp. 9188-9195
- [11] Elona Shatri, and György Fazekas, “DoReMi: First glance at a universal OMR dataset” 3rd International Workshop on Reading Music Systems (WoRMS), Alicante, Spain, 2021, pp. 43-49
- [12] Arnau Baró, Carles Badal, and Alicia Fornés, “Handwritten Historical Music Recognition by Sequence-to-Sequence with Attention Mechanism” 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), Dortmund, Germany, 2020, pp. 205-210
- [13] Jiří Mayer, and Pavel Pecina, “Synthesizing Training Data for Handwritten Music Recognition” 16th IAPR International Conference on Document Analysis and Recognition (ICDAR), Lausanne, Switzerland, 2021, pp. 626-641
- [14] Jiří Mayer, and Pavel Pecina, “Obstacles with Synthesizing Training Data for OMR” 4th International Workshop on Reading Music Systems (WoRMS), Online, 2022, pp. 15-19
- [15] Jan Hajič jr., and Pavel Pecina, “The MUSCIMA++ Dataset for Handwritten Optical Music Recognition” 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 2017, pp. 39-46
- [16] 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*, vol. 15, pp. 243-251, 2011
- [17] Ana Rebelo, Guilherme Artur Capela, and Jaime Cardoso, “Optical recognition of music symbols”, *International Journal on Document Analysis and Recognition (IJDA)*, vol. 13, pp. 19-31, 2010
- [18] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow (2015) “Adversarial autoencoders” arXiv preprint arXiv:1511.05644
- [19] Diederik Kingma, and Max Welling, “Auto-Encoding Variational Bayes”, 2nd International Conference on Learning Representations (ICLR), Banff, Canada, 2014.
- [20] Anders Boesen, Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther (2016) “Autoencoding beyond pixels using a learned similarity metric” arXiv preprint arXiv:1512.09300
- [21] Dor Bank, Noam Koenigstein, and Raja Giryes (2020) “Autoencoders” arXiv preprint arXiv:2003.05991
- [22] , Paul Hough (1962) “Method and means for recognizing complex patterns” patent no. US3069654A
- [23] H.K. Yuen, John Princen, John Illingworth, and Josef Kittler, “Comparative study of hough transform methods for circle finding”, *Image and Vision Computing*, vol. 8, pp. 71–77, 1990
- [24] Diederik Kingma, and Jimmy Ba, “Adam: A method for stochastic optimization”, 3rd International Conference on Learning Representations (ICLR), San Diego, USA, 2015
- [25] Vladimir Levenshtein, “Binary codes capable of correcting spurious insertions and deletions of ones”, *Problems of Information Transmission*, 1965

Towards Artificially Generated Handwritten Sheet Music Datasets

Pranjali Hande^{*}, Elona Shatri^{*}, Benjamin Timms[†] and György Fazekas^{*}

^{*} School of Electronic Engineering and Computer Science, Queen Mary University of London, [†] Steinberg London, UK

^{*}p.s.hande@se22.qmul.ac.uk, ^{*}e.shatri@qmul.ac.uk, [†]b.timms@steinberg.de ^{*}george.fazekas@qmul.ac.uk

Abstract—In optical music recognition (OMR), the scarcity of handwritten music sheet datasets limits technological advancement. Addressing this, our study introduces a method to synthetically generate handwritten music sheets using CycleGAN, a state-of-the-art Generative Adversarial Network (GAN). The synthesised dataset closely mimics real-world handwritten music, opening avenues for further research. Our findings underscore the efficacy of CycleGAN in producing high-fidelity datasets, thus enriching research resources and tackling the challenges of recognising handwritten musical notations.

I. INTRODUCTION

Written music has served as a tool for preserving musical compositions, ensuring their legacy throughout history. In the digital era there is an increasing urgency to transcribe and convert such pieces for continued accessibility and further conservation. However, manual transcription is expensive and time-consuming, leading to the need for research on automatic score transcribing systems. Optical music recognition (OMR) [2], [17] focuses on music score recognition systems, transforming scanned images into structured formats like MusicXML [20], MIDI, and MEI [21]. However, issues such as pre-processing degraded images, handling complex scores and dealing with the variety of handwritten scores [1] make OMR systems challenging.

Handwritten music scores face unique recognition challenges compared to typeset/printed scores. The complexity and variety of handwriting styles require further investigation [8], [15]. Deep learning has amplified research in music recognition, but it requires significant annotated data for effective training [19]. This study explores style transfer from handwritten to printed sheet music. The goal is to create handwritten music sheet compromises using neural networks that mimic human handwriting for a given printed sheet music input.

II. RELATED WORK

OMR has seen the development and use of various datasets, each tailored to different research needs. Table I summarises notable datasets, highlighting their type of engraving, size, format, and typical applications.

These datasets encompass both handwritten and printed musical symbols, serving varied research objectives [3], [16]. Datasets like PrIMuS [25], DeepScores [26], and DoReMi

[23] offer synthetic training data for OMR. There is a noted scarcity of large, well-annotated handwritten music datasets. However, augmentation techniques have been proposed to bridge this gap. Baró et al. [27] introduced the shuffling of handwritten music measures, and the Mashcima [28] proposed shuffling individual symbols, synthesising handwritten scores. Our research focuses on using the CVC-MUSCIMA and DoReMi datasets, with more details provided in Section II-A.

Adjacent domains, particularly handwritten text generation, have employed various deep learning models such as diffusion models, RNNs, and GANs [4]. RNNs are prevalent for online handwriting generation [10]. Conversely, while GANs excel at image generation, they sometimes produce repetitive outputs and can be challenging to train.

Our methodology relies on the image-to-image transfer technique [14] to synthesise handwritten music sheets from printed counterparts. Further insights into our approach are detailed in Section III.

A. Datasets

The two datasets we use in this study are the CVC-MUSCIMA dataset [5] and the DoReMi dataset [23]. CVC-MUSCIMA comprises 1000 handwritten sheet music images by 50 composers, with each transcribing 20 pages for various musical arrangements. The DoReMi dataset has 6432 printed sheet music images with nearly a million annotated elements. For our model training, we utilised 1000 images from both datasets. All images were converted to grayscale or a single input channel for optimised image processing efficiency, conserving memory and reducing processing time. Given the absence of paired datasets between printed and handwritten scores, we propose a method to translate printed sheets into their handwritten counterparts, aiming to encapsulate the unique traits of handwritten music scores.

III. METHODOLOGY

Given the lack of paired data detailed in Section II-A, there is a need to investigate into image-to-image translation techniques that accommodate for unpaired data. Our primary aim is to distinguish the nuances of handwritten music scores and understand the potential translation from printed music scores. We employ CycleGANs [24], a deep learning approach for image-to-image translations. Unlike conventional GANs, CycleGANs have the unique capability of producing images

TABLE I: Music Symbol Datasets

Name	Engraving	Size	Format	Usage
Handwritten Online Musical Symbols (HOMUS)	Handwritten	15,200 symbols	Text-File	Symbol Classification (online + offline)
Universal Music Symbol Collection	Typeset + Handwritten	~ 90,000 symbols	Images	Symbol Classification (offline)
CVC-MUSCIMA	Handwritten	1,000	Images	Staff line removal, writer ID
DoReMi	Typeset	6,432	Images, XML, musicXML, MEI, MIDI	Symbol Classification, Object Detection

in a target domain using source domain images, irrespective of the absence of paired images from both domains. Though their training poses greater challenges than standard GANs, due to the lack of paired datasets, they stand out in their ability to yield genuine-looking images in scenarios where paired training data is unavailable.

A. CycleGANs

GANs consist of two core components: the generator and the discriminator. These components work in tandem, adversarially, to refine training. One of the notable variations of GANs is the pix2pix model [14], a specific form of a conditional generative adversarial network (cGAN). Pix2pix has shown promise in various image-to-image translation tasks, often yielding detailed and realistic outputs.

The generator accepts input from a latent space and produces novel images that seem plausible within the predefined domain. Conversely, the discriminator determines the image’s authenticity, distinguishing real dataset images from generated [22] ones. We deploy the pix2pix model, which integrates both a Generator and a Discriminator. Additionally, this model incorporates a specific loss function [11] and can be used on unpaired training data. The architecture is rooted in U-Net, containing both a Downsampler, comprised of a Convolution layer, instance normalisation, and Leaky ReLU, and an Upsampler, which integrates Transposed Convolution, instance normalisation, Dropout, and Leaky ReLU. Our model employs two generators (G and F) and two adversarial discriminators (X and Y) to autonomously normalise each image without considering batch information.

- Generator G transforms X to Y .

$$(G : X \rightarrow Y) \quad (1)$$

- Generator F transforms Y to X .

$$(F : Y \rightarrow X) \quad (2)$$

- Discriminator D_x differentiates between the original image X and the generated image $X(F(Y))$.
- Discriminator D_y differentiates between the original image Y and the generated image $Y(G(X))$.

B. Loss functions

GANs employ specific loss functions for both the generator and the discriminator. The generator loss measures the generator’s capability to mislead the discriminator, whereas

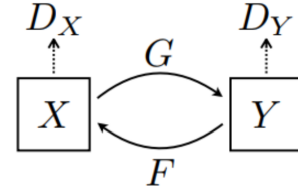


Fig. 1: Generator and Discriminator for Cycle GANs

the discriminator loss quantifies its aptitude in differentiating between authentic and generated samples. CycleGAN uses cycle consistency and adversarial loss to ensure accurate image translation coupled with discriminator accuracy [24].

- **Adversarial Loss** Adversarial loss ensures alignment in distribution between generated images and the target domain data, given in Eq. 3.

$$\begin{aligned} \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = & E_{y \sim P_{\text{data}}(y)} [\log D_Y(y)] \\ & + E_{x \sim P_{\text{data}}(x)} [\log(1 - D_Y(G(x)))] \end{aligned} \quad (3)$$

- **Cycle Consistency Loss**

Given the lack of paired data during training, determining the alignment between input x and target y can be challenging. Cycle consistency underscores that the generated output should resemble the initial input. Generator G takes input Image X and produces Image \hat{Y} . The generated image \hat{Y} is given as input to generator F which generates the cycled image \hat{X} . The Mean Absolute Error (MAE) is computed by taking the difference between X and \hat{X} . The forward cycle consistency loss is defined in Eq. 4.

$$X \rightarrow G(X) \rightarrow F(G(X)) \sim \hat{X} \quad (4)$$

The Backward cycle consistency loss is defined in Eq. 5.

$$Y \rightarrow F(Y) \rightarrow G(F(Y)) \sim \hat{Y} \quad (5)$$

- **Identity Loss** Identity loss described in Eq. 6, ensures that the generator’s conversion of images results in outputs closely resembling, if not indistinguishable from, the original images.

$$\text{IdentityLoss} = |G(Y) - Y| + |F(X) - X| \quad (6)$$

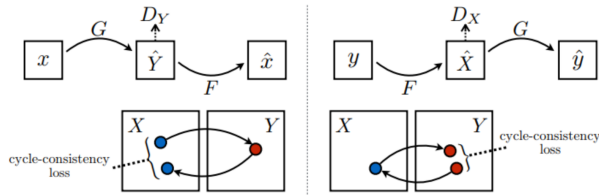


Fig. 2: Cycle Consistency Loss

IV. EVALUATION SETUP

We utilised the CVC MUSCIMA and the DoReMi datasets for our experiments. The datasets were further processed to obtain images suitable for our models, as illustrated in Fig. 3.

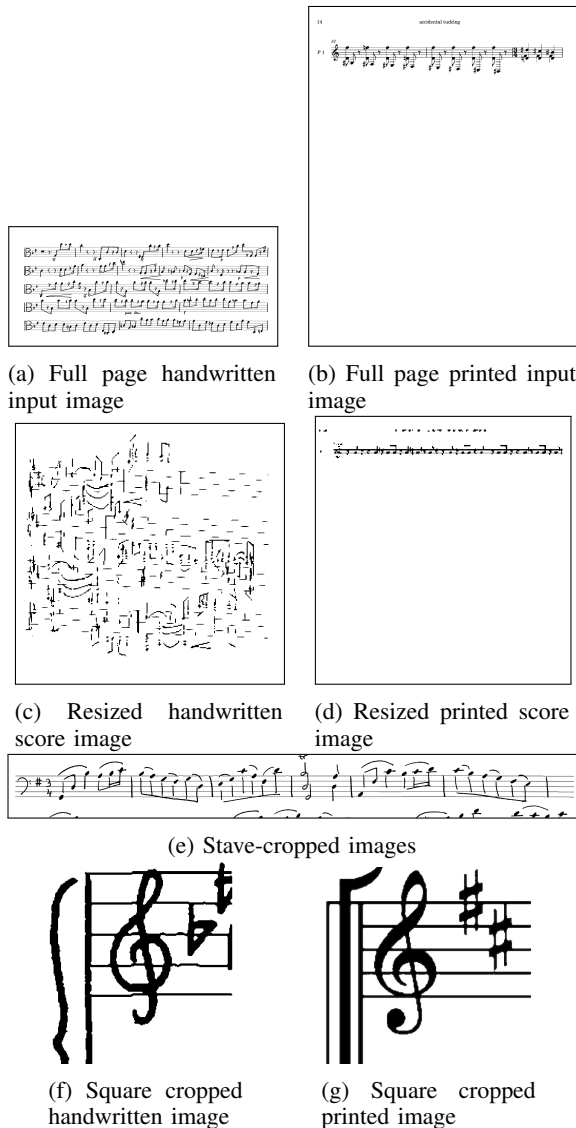


Fig. 3: Collection of images showcasing various input images used.

From the outset, we planned to work with various granularities of musical scores. This was done with the intention of

determining which granularity would yield the most accurate and visually pleasing results. The following enumeration lists the types of images we processed:

- 1) Full page handwritten and printed sheets of dimensions 1938x3450 pixels and 3504x2475 pixels respectively as shown in Fig. 3(a) and (b).
- 2) Resized images to 256x256 pixels to fit the requirements of our models, particularly the pix2pix U-Net generator as seen in Fig. 3(c) and (d).
- 3) Stave-level cropped images yielding single-stave rectangular images from handwritten and typeset images Fig. 3(e).
- 4) Square cropped images to meet the requirements of the CycleGAN model, resulting in images with fewer symbols and larger objects per image Fig. 3(g) and (f).

In the initial phase of our study, we worked with two sets of musical scores: handwritten CVC-MUSCIMA images of dimensions 1938x3450 pixels and printed DoReMi images of dimension 3504x2475 pixels as seen in Fig. 3(a) and (b). To fit the requirements of our models, particularly the pix2pix U-Net generator, the musical scores were resized to 256x256 pixels. The resizing leads to a significant loss of clarity, making the images appear blurred, as evident in Fig. 3(c) and (d).

Given the challenges posed by the full-sheet images, we adopted a revised strategy. We implemented image cropping to focus on smaller regions of the images. This entailed generating stave-level crops, yielding single-stave rectangular images from handwritten and typeset images, as seen in Fig. 3(e). Furthermore, to meet the requirements of the CycleGAN model, which expects square-shaped input images, the stave-cropped images were further processed to create square crops. This resulted in images with fewer symbols and larger objects per image, as depicted in Fig. 3(g) and (f).

Our evaluation setup employs the pix2pix model, which integrates both a Generator and a Discriminator. The architecture of this model is rooted in U-Net, consisting of both a Downsampler and an Upsampler. The Downsampler comprises a Convolution layer, instance normalisation, and Leaky ReLU, while the Upsampler integrates Transposed Convolution, instance normalisation, Dropout, and Leaky ReLU. Our model setup involves two generators (G and F) and two adversarial discriminators (X and Y) to autonomously normalise each image without considering batch information.

V. RESULTS

A. Visual Inspection

The model effectively generated handwritten musical scores mirroring the input using cropped square images. The results, though initially blurry, mimic real-world scanning imperfections. The produced images from UNet and ResNet are shown in Fig. 4 and more examples are provided in Appendix A.

We also conducted inference using pre-trained object detectors on printed scores within the generated images, as illustrated in Fig. 5. Although the object detector frequently misidentifies numerous objects, we recognize that this stems

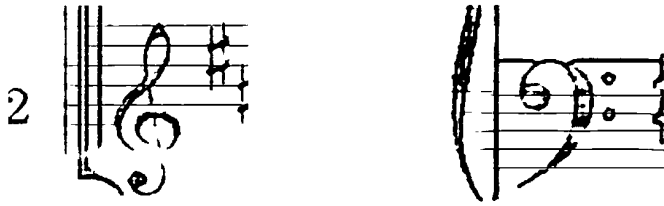


Fig. 4: Sample of Generated images from both UNet and ResNet models

from the absence of regions within individual objects. Employing image inpainting methods can mitigate such challenges. It is important to note that training on the generated samples would be ineffective here, given the unpredictable shifts in the annotations within these samples. However, for end-to-end transcription, annotations largely remain consistent, with only occasional random shifts in pitch.

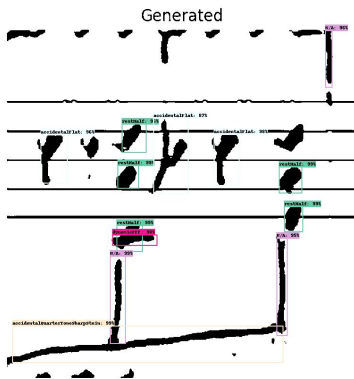


Fig. 5: Object detection results in a generated sample

B. Loss Metrics and Model Comparison

Our study emphasises the generator’s combined adversarial and cycle loss to evaluate model efficacy. The ideal trajectory indicates consistent translation improvements to handwritten styles. We compared the UNet and ResNet architectures to assess effectiveness. The UNet displays rising generator losses and decreasing discriminator losses. This trend suggests rapid development in the discriminator, leading to an increasing generator loss as the discriminator’s precision improves Fig. 6. On the other hand, ResNet shows promising translation results between Generator G and Discriminator X, with opposing trends for Generator F and Discriminator Y, indicating potential gradient issues Fig. 7.

C. Evaluation via Inception Scores

GANs and particularly CycleGAN, are challenging to evaluate due to the lack of ground truth [18] in many applications. Inception Score (IS) and Fréchet Inception Distance (FID) are commonly used metrics for quantitative evaluation, measuring

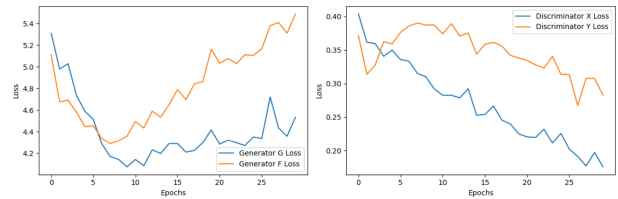


Fig. 6: Generator and Discriminator Losses for UNet model

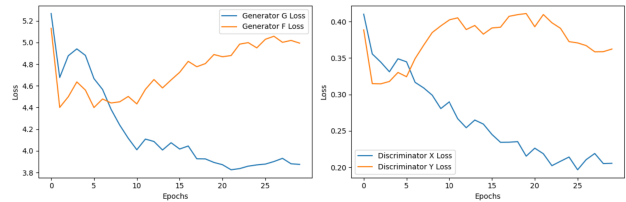


Fig. 7: Generator and Discriminator Losses for ResNet Model

sample diversity and resemblance of target distribution [6], [7], [9], [13], [22].

TABLE II: IS and FID Score

	FID Score	IS Mean	IS Std
UNet	69.83	1.84	0.14
ResNet	60.52	1.65	0.15

Table II presents the FID scores for Unet and ResNet, comparing printed to handwritten images. While there’s a one-to-one mapping between the original and generated images, they aren’t identical, resulting in higher FID values. The lack of a definitive ground truth hinders a full comparison of FID scores, limiting the comprehensive comparison of FID scores, thereby diminishing their significance.

Both models show comparable mean IS values, indicating similar image quality and variety. ResNet’s higher standard deviation suggests its generated handwritten sheets appear more realistic than those from UNet.

VI. CONCLUSION AND FUTURE WORK

Inspired by the need for a comprehensive handwritten music sheet dataset, our research used CycleGAN and pix2pix architectures to translate printed music sheets into handwritten styles. The CVC-MUSCIMA and DoReMi datasets were used to train a model that learned and translated the stylistic elements of handwritten music sheets into printed versions. The model’s efficacy and accuracy were evaluated using quantitative metrics like FID and IS. The model demonstrated a commendable capacity to generate handwritten images that closely resembled the original’s authenticity. However, further refinement and improvement are needed, including fine-tuning hyperparameters, incorporating more complex deep learning architectures, reducing generator loss and stabilising discriminator loss, and using modern GAN designs like BigGAN.

REFERENCES

- [1] Arnau Baró, Pau Riba, Jorge Calvo-Zaragoza, Alicia Fornés, From Optical Music Recognition to Handwritten Music Recognition: A baseline, Pattern Recognition Letters, Volume 123, 2019, Pages 1-8, ISSN 0167-8655. <https://doi.org/10.1016/j.patrec.2019.02.029>.
- [2] Shatri, Elona, and György Fazekas. "Optical music recognition: State of the art and major challenges." TENOR conference 2020, Hamburg, Germany.
- [3] 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/IC-DAR.2017.265
- [4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets" 27th International Conference on Neural Information Processing Systems (NIPS), Montreal, Canada, 2014, pp. 2642-2680
- [5] Anjan Dutta, Albert Gordo, 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, Volume 15, Issue 3, pp 243-251, 2012. DOI: 10.1007/s10032-011-0168-2
- [6] Barratt, S., & Sharma, R. (2018, January 6). A Note on the Inception Score. arXiv.org. <https://arxiv.org/abs/1801.01973v2>
- [7] Borji, Ali. "Pros and Cons of GAN Evaluation Measures: New Developments." Computer Vision and Image Understanding 215 (January 1, 2022): 103329. <https://doi.org/10.1016/j.cviu.2021.103329>.
- [8] Calvo-Zaragoza J, Rizo D. End-to-End Neural Optical Music Recognition of Monophonic Scores. Applied Sciences. 2018; 8(4):606. <https://doi.org/10.3390/app8040606>
- [9] Chong, M.J., Forsyth, D., 2020. Effectively unbiased FID and inception score and where to find them. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6070–6079.
- [10] Chung, K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio. A recurrent latent variable model for sequential data, 2015.
- [11] CycleGAN — TensorFlow Core. (n.d.). TensorFlow. <https://www.tensorflow.org/tutorials/generative/cyclegan>
- [12] E. Alonso, B. Moysset and R. Messina, "Adversarial Generation of Handwritten Text Images Conditioned on Sequences," 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, NSW, Australia, 2019, pp. 481-486, doi: 10.1109/ICDAR.2019.00083.
- [13] Heusel, Martin, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium." In Advances in Neural Information Processing Systems, edited by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Vol. 30. Curran Associates, Inc., 2017.
- [14] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2016, November 21). Image-to-Image Translation with Conditional Adversarial Networks. arXiv.org. <https://arxiv.org/abs/1611.07004v3>
- [15] J. C. Pinto, P. Vieira, and J. M. Sousa, "A new graph-like classification method applied to ancient handwritten musical symbols," IJDAR, vol. 6, no. 1, pp. 10–22, 2003.
- [16] J. Calvo-Zaragoza and J. Oncina, "Recognition of Pen-Based Music Notation: The HOMUS Dataset," 2014 22nd International Conference on Pattern Recognition, Stockholm, 2014, pp. 3038-3043. DOI: 10.1109/ICPR.2014.524.
- [17] J. Calvo-Zaragoza, J. Hajić jr., and A. Pacha, "Understanding optical music recognition," Computing Research Repository, 2019.
- [18] J. Gui, Z. Sun, Y. Wen, D. Tao and J. Ye, "A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications," in IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 4, pp. 3313-3332, 1 April 2023, doi: 10.1109/TKDE.2021.3130191.
- [19] Jan Hajić jr., Pavel Pecina. The MUSCIMA++ Dataset for Handwritten Optical Music Recognition. 14th International Conference on Document Analysis and Recognition, ICDAR 2017. Kyoto, Japan, November 13-15, pp. 39-46, 2017. DOI: 10.1109/ICDAR.2017.16
- [20] M. Good and L. Recordare, "Lessons from the adoption of musicxml as an interchange standard," in Proceedings of XML, 2006, pp. 5–7.
- [21] P. Roland, "The music encoding initiative (MEI)," in 1st International Conference on Musical Applications Using XML, 2002, pp. 55–59.
- [22] Salimans, Tim, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. "Improved Techniques for Training GANs." In Advances in Neural Information Processing Systems, edited by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Vol. 29. Curran Associates, Inc., 2016.
- [23] Shatri, Elona, and György Fazekas. DoReMi: First glance at a universal OMR dataset. WoRMS 2021, 2021. <https://arxiv.org/abs/2107.07786>
- [24] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017, March 30). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. arXiv.org. <https://arxiv.org/abs/1703.10593v7>
- [25] Jorge Calvo-Zaragoza, and David Rizo, "End-to-End Neural Optical Music Recognition of Monophonic Scores" Applied Sciences, vol. 8, no. 4, pp. 606, 2018
- [26] Lukas Tuggener, Yvan Putra Satyawan, Alexander Pacha, Jürgen Schmidhuber, and Thilo Stadelmann, "The DeepScoresV2 Dataset and Benchmark for Music Object Detection" 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 2021, pp. 9188-9195
- [27] Arnau Baró, Carles Badal, and Alicia Fornés, "Handwritten Historical Music Recognition by Sequence-to-Sequence with Attention Mechanism" 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), Dortmund, Germany, 2020, pp. 205-210
- [28] Mayer, Jiří, and Pavel Pecina. "Synthesizing Training Data for Handwritten Music Recognition." In Document Analysis and Recognition – ICDAR 2021, edited by Josep Lladós, Daniel Lopresti, and Seiichi Uchida, 626–41. Cham: Springer International Publishing, 2021.

APPENDIX

Below you can find a list of input images and the handwritten generated samples from both UNet and ResNet models:

A. Handwritten images generated using UNet Model

This first set of images represents the generated handwritten music sheet for a given input printed music sheet using UNet Model.

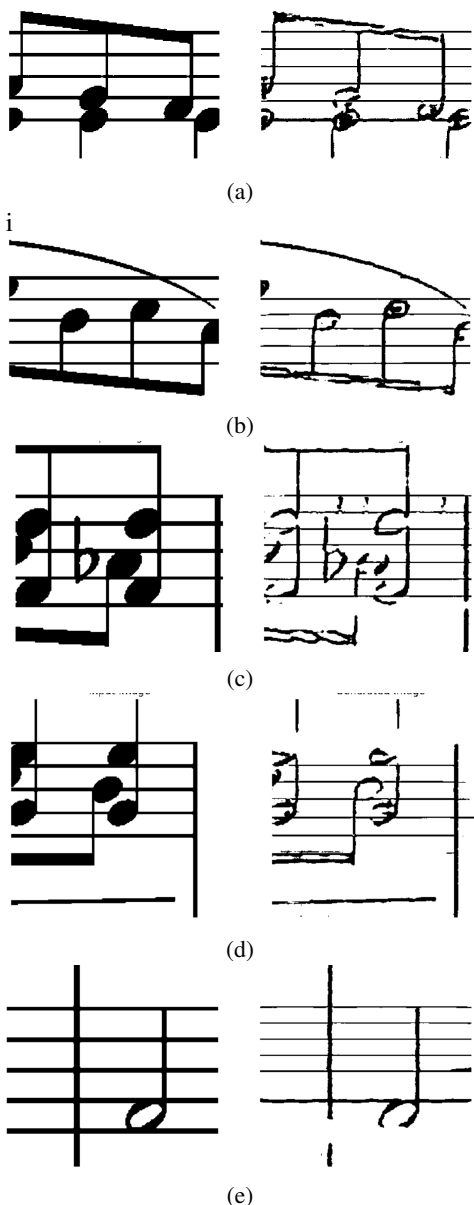


Fig. 8: Samples of handwritten sheet music images generated using UNet on the right and the input samples on the left

B. Handwritten images generated using ResNet Model

This first set of images represents the generated handwritten music sheet for a given input printed music sheet using ResNet Model.

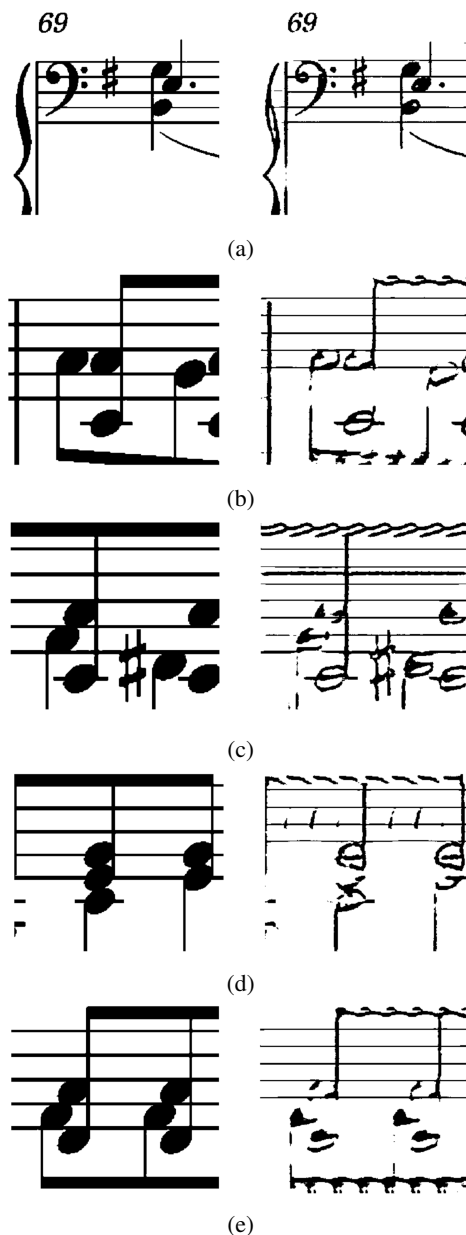


Fig. 9: Samples of handwritten sheet music images generated using ResNet on the right and the input samples on the left

Improving Sheet Music Recognition using Data Augmentation and Image Enhancement

Zihui Zhang*, Elona Shatri*, and György Fazekas*

* School of Electronic Engineering and Computer Science, Queen Mary University of London
London, UK

*zihui.zhang@hss21.qmul.ac.uk, *e.shatri@qmul.ac.uk, *george.fazekas@qmul.ac.uk

Abstract—Data preprocessing techniques are frequently employed to address a model’s generalisation capability and performance constraints due to the quality and diversity of its training data. Effective data techniques such as data augmentation are crucial for data that is difficult to acquire, such as annotated scores of lower image quality. This study mainly explores sheet music image preprocessing methods and delves into the impact of these strategies on model performance. By comparing existing methods, this research aims to identify more optimised strategies, enabling the model to handle noise and interference in real-world scenarios better and improving its generalisation capability and performance.

I. INTRODUCTION

Optical Music Recognition (OMR) is concerned with translating written music notation into machine-readable data [1]. As deep learning revolutionises various Computer Vision (CV) tasks, exploring optimal model efficiency and generalisation in image recognition, including OMR, remains crucial [2]. Model performance often hinges on the diversity and quality of training samples. Given the challenges in obtaining diverse samples, research has pivoted towards data augmentation methods. Our study explored two such techniques: introducing noise and modifying specific image features. We found that these methods affect model efficiency. Compared to the baseline models we employed by Rios-Vila et al. [4], we observed that models using these augmentation techniques can efficiently combat real-world noise and other disturbances, demonstrating better recognition.

II. RELATED WORK

One of the ongoing challenges in OMR is the recognition of degraded or low-quality score images despite the advancements in deep learning. While data augmentation and image preprocessing have been employed [6] to reduce reliance on purely synthetic data, the application of staff-line removal, a known preprocessing step in OMR [2], in end-to-end models remains unexplored. This report investigates into this technique and other preprocessing methods that have been previously applied in non-deep learning contexts. Our primary aim is to discern the impact of such preprocessing on the training of modern models.

III. METHODOLOGY

In this study, we use a hybrid model combining the strengths of Convolutional Neural Networks (CNNs) and Transformers, based on work by Rios-Vila et al. [4]. This model takes a monophonic music notation image, ‘x’, and its note symbol sequence, ‘y’, as inputs. For a given image ‘x’, the model predicts the most probable symbol sequence, \hat{y} , using the equation:

$$\hat{y} = \arg \max P(y|x)$$

The CNN extracts image features through convolutional layers, with batch normalisation and the LeakyReLU activation function applied post-convolution. The feature map size is then reduced using max-pooling. These feature maps are reshaped and fed into the Transformer, which employs a “self-attention mechanism” [5]. This mechanism enables the Transformer to process sequences in parallel, contrasting with traditional RNNs. Each Transformer layer also integrates a feed-forward neural network for additional processing.

The model begins with an input size of $64 \times W \times 1$ proceeding with a convolutional block and batch normalisation, LeakyReLU activation function, and some layers also include a 2×2 max-pooling operation. The model then undergoes a transformation block where the data is permuted and reshaped. This is followed by a transformer block with both encoder and decoder components, and finally the model concludes with a dense layer as shown in Fig. 1.

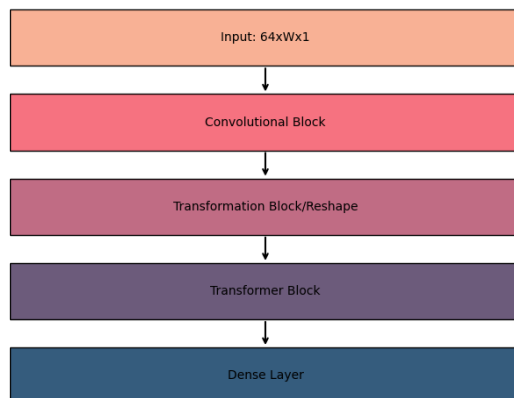


Fig. 1: The basic network architecture used in the transformer model

IV. DATASET

We employed the Camera-PrIMuS dataset [6], enhancing it with additional data augmentation techniques, such as noise introduction and stave removal as seen in Fig. 2.

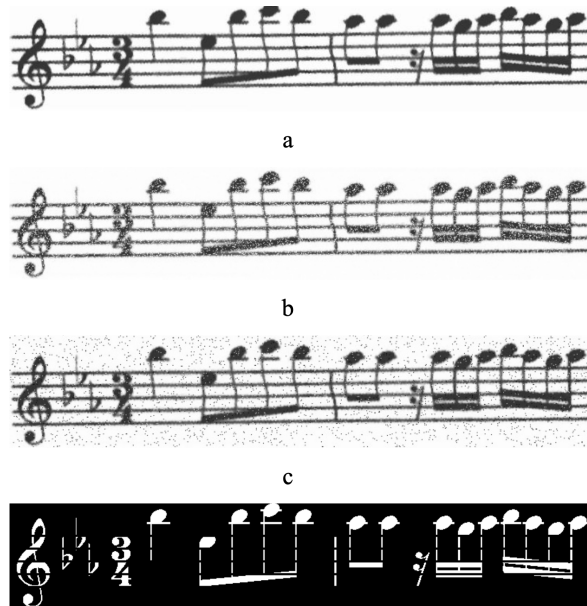


Fig. 2: PrIMuS sample image (a) original image; (b) image with Gaussian noise; (c) image with salt and pepper noise; (d) keep only the pictures of the notes.

V. EVALUATION AND EXPERIMENTS

We trained a baseline model using the original PrIMuS score. This served as a foundation for subsequent data augmentation and image preprocessing. The Levenshtein distance was our chosen evaluation metric in the form of symbol error rate (SER) often used in OMR [1]. We compared various models' performances based on the variations of the dataset.

VI. RESULTS

Our research into sheet music image preprocessing techniques underlines the significant impact of data augmentation on an OMR model performance. Two main preprocessing strategies were investigated: introducing noise and removing stave lines. Both techniques demonstrated improved model performance as evidenced by the Symbol Error Rate (SER) results. Notably, the model fine-tuned with noise (Mo_2) achieved the best performance, reducing the SER to 26.4%, compared to the baseline model's Mo_1 35.7% as shown in Fig. 3. This indicates that noise augmentation aids the model in focusing on key features by disregarding irrelevant information, ultimately strengthening its generalisation capabilities.

Similarly, the model trained on images with stave lines removed (Mo_3) also outperformed the baseline, albeit not as substantially as the noise-introduced model. This suggests that while stave line removal can enhance the model's versatility;

it may not provide as diverse a set of data variations as noise augmentation.

Introducing noise allows the model to prioritise key features by disregarding extraneous information. This approach forces the model to discern inherent data patterns over memorisation and is also practical for real-world scenarios with noise and disturbances. Moreover, it is easy to implement. Adding noise appears to be slightly more effective than removing stave lines. Introducing noise offers the model a wider range of data variations, enhancing its generalisability. In contrast, while stave line removal introduces its own set of challenges, it may not offer as diverse a set of data variations as noise addition. Notably, stave lines in music scores convey vital information, such as note pitch. While their removal doesn't alter the final note sequence, the lack of this context could limit the model's applicability in certain situations.

Model	Mo_1	Mo_2	Mo_3	Mo_4	Mo_5	Mo_6
SER(%)	35.7	26.4	28.1	32.2	31.8	33.5

TABLE I: Mo_1 is the base model trained using only 12,800 sketches. Mo_5 is the model fine-tuned with noisy images. Mo_6 is trained on images with removed staff lines. Mo_4 integrates both the fine-tuning methods. Mo_2 combines training from 12,800 printed scores and an equal number of noisy scores. Mo_3 is trained using 12,800 printed scores complemented by 6,400 images where staves have been omitted.

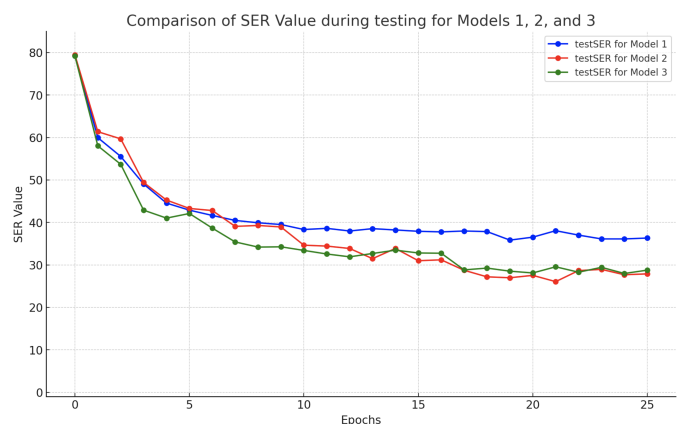


Fig. 3: Comparisons between SER for model 1, 2 and 3

VII. CONCLUSIONS

In OMR, addressing low-quality scores is crucial. Our research emphasises the value of data preprocessing, with methods like noise addition improving model accuracy, while stave removal highlights challenges in deciphering handwritten scores. Concurrently, synthetic data generation offers a pathway for dataset expansion, but its value in OMR is contingent upon the genuineness and quality of the created samples. In contrast, data augmentation refines extant samples, enhancing model adaptability. As OMR evolves, strategically combining these approaches will be vital for optimised real-world performance.

REFERENCES

- [1] J. Calvo-Zaragoza, J. Hajič jr., and A. Pacha, "Understanding optical music recognition," Computing Research Repository, 2019.
- [2] Shatri, Elona, and György Fazekas. "Optical music recognition: State of the art and major challenges." TENOR conference 2020, Hamburg, Germany.
- [3] Calvo-Zaragoza, Jorge, and David Rizo. "End-to-end neural optical music recognition of monophonic scores." Applied Sciences 8.4 (2018): 606.
- [4] Ríos-Vila, Antonio, José M. Iñesta, and Jorge Calvo-Zaragoza. "On the use of transformers for end-to-end optical music recognition." Iberian Conference on Pattern Recognition and Image Analysis. Cham: Springer International Publishing, 2022.
- [5] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
- [6] Calvo-Zaragoza, Jorge, and David Rizo. "Camera-PrIMuS: Neural End-to-End Optical Music Recognition on Realistic Monophonic Scores." ISMIR. 2018.
- [7] López-Gutiérrez, Juan C., et al. "Data augmentation for end-to-end optical music recognition." International Conference on Document Analysis and Recognition. Cham: Springer International Publishing, 2021.

Rotations Are All You Need: A Generic Method For End-To-End Optical Music Recognition

Antonio Ríos-Vila

U.I. for Computing Research, University of Alicante, Spain

arios@dlsi.ua.es

Abstract—End-to-end Optical Music Recognition traditionally involves multi-step processes to address complex documents, where single staff isolation is performed. These state-of-the-art methods often fall short in handling diverse music textures, such as polyphony or Aligned Music Notation and Lyrics Transcription (AMNLT). We introduce a generic end-to-end OMR approach compatible with monophonic, polyphonic, and AMNLT systems. By leveraging score rotations and multi-line transcription unfolding, this model only requires input system annotations during training. Experimental evidence suggests this approach offers a competitive solution with encouraging outcomes, paving the way for future end-to-end music transcription research.

Index Terms—Optical Music Recognition, Monophonic Scores, Polyphonic Scores, Aligned Music Notation & Lyrics Transcription, Score Unfolding, Rotations

I. INTRODUCTION

Optical Music Recognition (OMR) is the research field that focuses on the computational interpretation of music scores [1]. While it remains an active research domain, there have been substantial developments thanks to machine learning, especially neural networks. These advancements have shifted OMR from complex multi-stage methods [2]–[4] to more streamlined techniques. One of them is holistic music transcription. This approach takes a single-staff score excerpt as input and produces a symbolic textual output, either as graphics-based text [5] or a digital music encoding sequence [6].

Despite these advances, the application of state-of-the-art systems remains limited to single-staff monophonic or homophonic music scores. For more complex documents, such as full-page scores, polyphonic systems, or the simultaneous transcription of music and lyrics—referred to as Aligned Music Notation and Lyrics Transcription (AMNLT)—researchers often resort to intricate methods. Current OMR literature provides multi-stage pipelines as a *divide and conquer* approach [7]–[9]. These avenues break the challenge down to single-staff transcription and apply postprocessing to retrieve the representation of the complex score. Yet, this is not ideal for some music textures, especially where polyphony and AMNLT are involved. The key challenge remains in the alignment between score elements, shown in Figure 1, which becomes in a non-trivial issue if addressed with post-processing algorithms.

This paper is part of the project MultiScore (PID2020-118447RA-I00), funded by MCIN/AEI/10.13039/501100011033. The author is supported by grant ACIF/2021/356 from “Programa I+D+i de la Generalitat Valenciana”.



Fig. 1: Illustration of the alignment challenge in a chanted melody fragment between a sequence of music symbols and text. Red boxes denote the pixel dimensions of music symbols within the image, while blue boxes represent those of syllables.

In this paper, we introduce a generic approach based on score rotation and a segmentation-free multiline transcription. This method seamlessly transcribes monophonic, polyphonic, and AMNLT music systems in a single step, producing results in a standard digital music encoding format and implicitly addressing effectively both transcription and alignment challenges.

II. A GENERIC END-TO-END METHOD FOR OMR

A. End-to-end staff-level OMR

State-of-the-art OMR seeks the most probable symbolic representation for each staff-section image.

This process addressed with neural networks. These networks are based on two main blocks: first, there is an encoder that filters an input image to learn and establish its relevant features, which is typically established with Convolutional Neural Networks (CNN). Then, the output of this module is processed by a decoder block, which learns temporal dependencies in the data to enhance results. This is commonly implemented with Recurrent Neural Networks (RNN). The whole model, referred to as a Convolutional Recurrent Neural Network (CRNN), is trained using the Connectionist Temporal Classification (CTC) [10] loss strategy, which forces the network to align the output sequence to the available information extracted from the image.

The method assumes transcription as a sequence retrieval task, making the output of the network always be a sequence of tokens, music-notation symbols and characters in this case. Such sequences are derived from images by transforming the image domain into a sequence domain. These approaches often implement a reshape function that collapses the output feature map of the encoder vertically. This enables left-to-right symbol reading, ensuring each frame column represents only one symbol.

B. The limits of current state of the art

State-of-the-art OMR methods excel at single-staff transcription where each frame contains one and only one symbol. However, this approach falters with more demanding scores.

Complex musical scores, such as polyphonic or AMNLT ones, do not present a strict top-to-bottom, left-to-right reading order. They instead follow an interpretation where certain elements are read concurrently, such as simultaneous notes or a syllable that has to be sung with a specific tone. The paradigm shift from sequential to simultaneous reading presents a significant challenge, since the premise of one frame for one musical symbol collapses.

For scores with moderate complexity, like homophonic ones, where vertically aligned symbols start and end together, vocabulary-based methods may suffice [11]. Such methods can be stretched to accommodate polyphonic transcription, though this significantly lengthens the ground truth output sequence. But, as the complexity escalates, like pianoform or string quartet scores, this method becomes ineffective. The vertical collapsing technique cannot generate enough frames in those cases. An additional challenge is found in AMNLT, where the musical and textual domains merge.

Therefore, to address these transcriptions, we must search beyond current single-staff music transcription techniques for more versatile and scalable solutions.

C. Rotations are all you need

In this paper, we introduce an end-to-end method capable of directly retrieving the content representation of a music score in a single step. This method is compatible with monophonic, polyphonic, and AMNLT systems. Our approach leverages the Humdrum `**kern` [12] notation, subsequently referred to as KERN format, a digital music encoding scheme.

Examining the KERN format closely reveals that each text line in the ground truth corresponds to a distinct *timestep* in the music score. In essence, all elements within a KERN line occur simultaneously across various spines. The reading order of these documents is from top to bottom and then left to right. This pattern aligns with the left-to-right reading direction of the music score, departing from the lowest staff in each system. A graphical alignment between the two can be achieved by rotating the source image 90°. For lyrics, a specific text spine is dedicated to represent this content. Since the KERN syntax establishes the same alignment rules as in the rest of documents, this method can be also applied. However, there is the need of including an additional flip to ensure this alignment. As demonstrated in Fig. 2, this transformation perfectly aligns the reading order for both score image and ground truth.

By following this interpretation, we obtain both a document and a ground-truth text representation that are read like a text paragraph. This consequently makes it possible to reformulate the solution through segmentation-free multiline transcription, which is a text methodology whose objective is to transcribe document images that contain more than one line without the

need to perform any previous line detection process or content retrieval postprocessing.

D. Score unfolding approach

Among all the existent formulations to solve multiline transcription, we have taken inspiration from the document unfolding methods of this field, where the model learns to unfold text lines in order for them to be read sequentially [13], [14].

A graphic visualization of our methodology is depicted in Fig. 3. Here, rather than concatenating frame-wise elements along the height axis (h) during the vertical collapse, we reshape the feature map by concatenating all of its rows (w)—in the same way as [14]—to subsequently obtain a $(c, h \times w)$ sequence, in which c is the number of filters used by the convolutional layers of the model. The input image of the system—which is a music staff (with lyrics) rotated 90° and flipped horizontally—is therefore reshaped as a sequence where music notes can be read. Depending on the challenge we are dealing with, the output vocabulary of the network is the KERN notation $\Sigma_k \cup \{\epsilon\}$ or, in the case of AMNLT, its union with the character set given by the text vocabulary for lyrics transcription $\Sigma' = \Sigma_k \cup \Sigma_t \cup \{\epsilon\}$.

III. EXPERIMENTAL SETUP

A. Considered implementations

The method presented in this paper is implemented through the state-of-the-art OMR networks. All the presented solutions contain a fully convolutional block, which acts as an encoder of the input image features. This network is composed of stacked convolutional layers, which end up producing a feature map of size $(h/16, w/4, c, b)$, h and w being the height and the width of the input image, c the filters in the last convolutional layer, and b the batch size. Then, the following decoding architectures are proposed:

1) *Recurrent Neural Network*: We follow the implementation of the original CRNN-CTC staff transcription model from [15], where the reshaped feature map is fed into a Bidirectional LSTM (BLSTM) and linearly projected onto the music notation dictionary. Specifically, we implemented a BLSTM with 512 units.

2) *The Transformer*: In both the polyphonic and AMNLT scenarios, the model will have to process long sequences in one step, something that can have a negative impact on the performance of RNNs. For this reason, we propose an alternative encoder based on the Transformer [16] (referred to as CNNT). In particular, we implemented one encoder layer with an embedding size of 512, a feed-forward dimension of 1024, and 8 attention heads.

3) *Sequence-processing-free module*: We implemented an analogous text model [14] that does not use sequence processing in the decoding stage, leaving all the data processing to the CNN. With this model, we intend to analyse the relevance of using sequence processing modules in the transcription task.

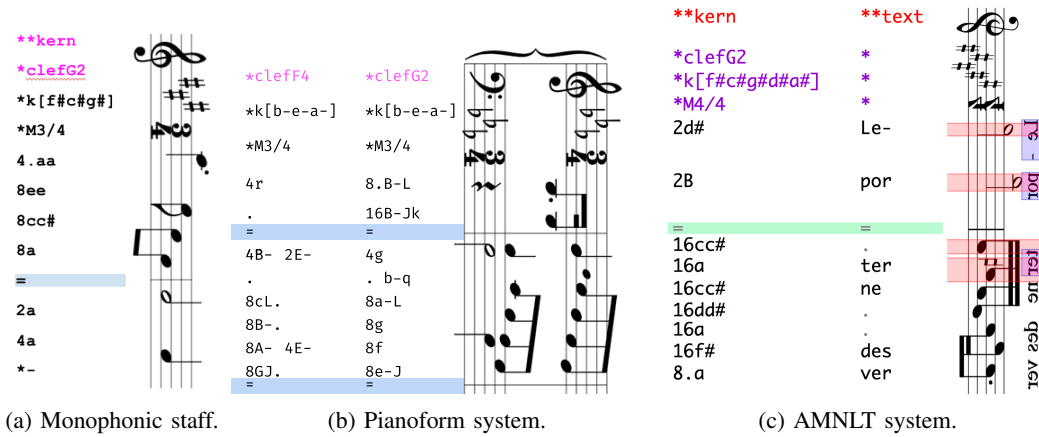


Fig. 2: Examples of KERN scores (left) aligned with its rendered music document (right) for both the polyphonic and AMNLT music scores.

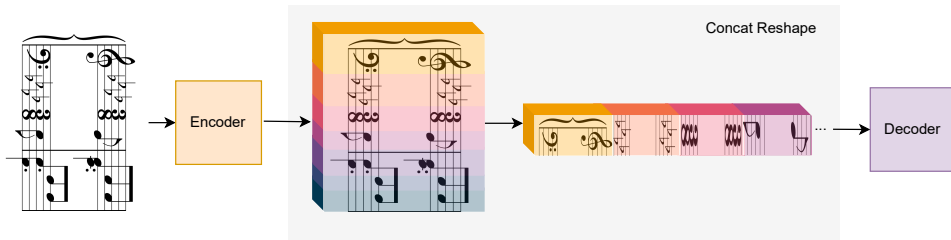


Fig. 3: Graphical scheme of the score unfolding method based on our formulation.

B. Evaluation scenarios

In this paper, we provide three evaluation scenarios based on the music scores that we aim to transcribe.

1) *Monophonic scores*: We test our methodology against the state-of-the-art staff-level transcription system defined in [5]. Since the network of this paper is more heavily parametrized than our reference baseline, we also test this system with the standard vertical collapse method to analyze the goodness of our method. We conducted experiments in the Camera PrIMuS dataset for this

2) *Polyphonic scores*: In the case of polyphonic music scores, we conducted experiments on the GrandStaff dataset [17], which is a collection of 53882 single-system pianoform scores from the KernScores¹ repository designed for end-to-end OMR experiments.

3) *AMNLT*: Since there is no currently available dataset to address the AMNLT challenge, we created a synthetic corpus with the data generator presented in [18] with 35,000 samples. This generator supports among five languages for lyrics generation, which are English, French, Spanish, German and Latin. In this scenario, we compare our approach to a *divide and conquer* method, where music and lyrics are transcribed separately.

C. Metrics

To evaluate model performance on monophonic and polyphonic score transcription, we computed the Symbol Error Rate metric [19]. This metric effectively represents the model accuracy in recognition tasks and reflects the manual correction effort required by users. To include AMNLT and avoid metric ambiguity, we defined a unified evaluation framework. We refer to the Symbol Error Rate metric for music notation errors as Music Error Rate (MER). For lyrics transcription quality, we compute standard Character Error Rate (CER) and a syllable-level Syllable Error Rate (SER). Considering alignment precision is crucial, we introduce the Kern Line Error Rate (KLER), which measures errors at both note pairings and note-syllable levels, which are represented through single KERN lines. As shown in Fig. 2, KERN files separate music and text into columns using indents, with lines representing sequential time steps. Given the format `<spine><tab><spine>`, KLER identifies structural mismatches between predicted and actual lines, ensuring the KERN structure accuracy and the integrity of all pairings.

IV. RESULTS

A. Monophonic scores

Table I showcases the experimental results on monophonic scores, comparing the performance from the models of this study against the state-of-the-art reshape approach (vertical collapse) and our proposed unfolding method. The unfolding

¹<http://kern.ccarh.org/>

TABLE I: Average MER (%) obtained by the studied architectures and reshape methods on the test set for the monophonic Camera PriMuS dataset. Bold values show the best retrieved performance.

Method	MER	
	Vertical Collapse	Unfolding
Baseline	4.7	-
FCN	6	7.8
CRNN	3.3	4.8
CNNT	9.8	10.4

technique demonstrates capability for end-to-end monophonic transcription but mirrors the results of state-of-the-art approaches. We observe a relative performance improvement, but this is primarily attributed to the implemented convolutional architecture, which enhances the state-of-the-art results by 1% SER. Thus, our methodology proves competent in monophonic transcription but does not notably surpass existing state-of-the-art techniques.

B. Polyphonic scores

Table II presents the performance of our methodology on both the perfect and distorted versions of the GrandStaff dataset, with an absence of baseline results due to state-of-the-art inability to converge for this dataset.

The unfolding technique yields promising results, most notably a MER of 7.3 % and 9.9 % with the CRNN implementation. This emphasizes the importance of sequence processing modules for polyphonic transcription, especially given the noticeable performance drop of the FCN in comparison. With the KERN encoding in mind, Recurrent Neural Networks appears as the prime candidate for this task, continuing the insights of recent research [16].

Diving into the KLER, we observe a performance of 23.2% and 29.5%. This highlights the capability of the model to produce accurate KERN score documents, with most errors being intra-line and stemming from the MER. These outcomes suggest our approach is apt for practical applications handling KERN files, and corrections can be confined to individual line contents rather than broader document structure.

TABLE II: Average MER and KLER (%) obtained by the studied models on the test set for both the perfectly printed and the distorted versions of the GrandStaff dataset.

Model	GrandStaff		Camera GrandStaff	
	MER	KLER	MER	KLER
FCN	23.9	67.9	30.2	69.0
CRNN	7.3	23.2	9.9	29.5
CNNT	11.1	32.4	12.3	33.3

C. AMNLT

Table III presents the results over the test set for the AMNLT scenario. The baseline method reports the best results. In both corpus versions, recognition rates from the baseline values indicate similar outcomes. Specifically, the MER values are 0.4% for ideal images and 1.9% for camera-based ones. Both the CER and SER metrics show identical results of 1.2% and 5.2% in each case. This suggests that a single-task end-to-end transcription method works well for independent modalities. Yet, the KLER metric shows it is not suitable for alignment, presenting 42.0% for ideal images and 42.9% for the camera-based setup.

When considering our holistic approach, we observe that it is also competent in the AMNLT task. Despite that, the MER, CER, and SER metrics perform worse than the baseline. However, their KLER values see significant improvements, dropping to 6.7% from 42% for non-distorted images and to 8.6% from 42.9% for distorted ones. This indicates the efficacy of these holistic methods in addressing AMNLT. Even though there is a decrease in precision for individual sequence transcriptions, the alignment is better, which is essential to the information retrieval of the score. Further, all the approaches, except for the CNN, are effective for the task at hand, being the CNNT the best option for perfect images and the CRNN for the distorted ones.

TABLE III: Average values of MER (%), CER (%), SER (%), and KLER (%) for the AMNLT scenario. Bold values highlight the best results per corpus, and underlined ones indicate the top results from the proposed method.

Method	Perfect images				Camera-based images			
	MER	CER	SER	KLER	MER	CER	SER	KLER
Baseline	0.4	1.2	5.2	42.0	1.9	1.2	5.2	42.9
FCN	15.6	19.5	53.6	21.4	13.0	31.7	90.5	25.5
CRNN	5.2	10.8	30.9	9.7	4.8	8.1	25.1	8.6
CNNT	<u>3.1</u>	<u>7.8</u>	<u>22.4</u>	<u>6.7</u>	<u>5.5</u>	<u>13.6</u>	<u>24.6</u>	<u>16.2</u>

V. CONCLUSION

In this work, we introduce a generic method for end-to-end OMR. This strategy relies on score rotation, implicit encoding alignment—aided by the KERN format—and a segmentation-free multiline transcription. Our objective goes beyond transcribing monophonic scores; we target polyphony and the simultaneous transcription of music and lyrics, areas that have not been thoroughly approached before. Our findings indicate that this method not only competes effectively within its domains but also breaks state-of-the-art techniques limitations, such as score elements alignment.

This research opens up future research avenues. Currently, our method is tailored for music systems, still falling short for full-page transcription. Attention-based methods [20], [21] may offer beneficial insights in this aspect. Furthermore, this research suggests the potential for a universal OMR model, challenging the convention of separate models for each musical texture, where a single model is able to perform end-to-end transcription for all available music scores.

REFERENCES

- [1] Jorge Calvo-Zaragoza, Jan Hajič Jr., and Alexander Pacha. Understanding optical music recognition. *ACM Computing Surveys*, 53(4), July 2020.
- [2] John Ashley Burgoyne, Laurent Pugin, Greg Eustace, and Ichiro Fujinaga. A comparative survey of image binarisation algorithms for optical recognition on degraded musical sources. In *Proc. of the 8th Int. Conf. on Music Information Retrieval*, pages 509–512.
- [3] J. dos Santos Cardoso, Artur Capela, Ana Rebelo, Carlos Guedes, and J. Pinto da Costa. Staff detection with stable paths. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(6):1134–1139, 2009.
- [4] Alicia Fornés, Josep Lladós, and Gemma Sánchez. Old handwritten musical symbol classification by a dynamic time warping based method. In Wenyin Liu, Josep Lladós, and Jean-Marc Ogier, editors, *Graphics Recognition. Recent Advances and New Opportunities*, pages 51–60, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [5] Jorge Calvo-Zaragoza and David Rizo. End-to-end neural optical music recognition of monophonic scores. *Applied Sciences*, 8(4), 2018.
- [6] Antonio Ríos-Vila, David Rizo, and Jorge Calvo-Zaragoza. Complete optical music recognition via agnostic transcription and machine translation. In Josep Lladós, Daniel Lopresti, and Seiichi Uchida, editors, *Proc. of the 16th Int. Conf. on Document Analysis and Recognition*, pages 661–675, 2021.
- [7] Francisco J. Castellanos, Carlos Garrido-Munoz, Antonio Ríos-Vila, and Jorge Calvo-Zaragoza. Region-based layout analysis of music score images. *Expert Systems with Applications*, 209:118211, 2022.
- [8] Sachinda Edirisooriya, Hao-Wen Dong, Julian Mcauley, and Taylor Berg-Kirkpatrick. An Empirical Evaluation of End-to-End Polyphonic Optical Music Recognition. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, pages 167–173, Online, November 2021. ISMIR.
- [9] John Ashley Burgoyne, Yue Ouyang, Tristan Himmelman, Johanna Devaney, Laurent Pugin, and Ichiro Fujinaga. Lyric extraction and recognition on digital images of early music sources. In *Proceedings of the 10th Int. Society for Music Information Retrieval Conference*, volume 10, pages 723–727, 2009.
- [10] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the Twenty-Third International Conference on Machine Learning, (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, pages 369–376, 2006.
- [11] María Alfaro-Contreras, Jorge Calvo-Zaragoza, and José M. Iñesta. Approaching end-to-end optical music recognition for homophonic scores. In *9th Iberian Conference Pattern Recognition and Image Analysis*, volume 11868 of *Lecture Notes in Computer Science*, pages 147–158, Madrid, Spain, 2019. Springer.
- [12] David Huron. Humdrum and Kern: Selective Feature Encoding BT - Beyond MIDI: The handbook of musical codes. In *Beyond MIDI: The handbook of musical codes*, pages 375–401. MIT Press, Cambridge, MA, USA, jan 1997.
- [13] Mohamed Yousef and Tom E. Bishop. Origaminet: Weakly-supervised, segmentation-free, one-step, full page textrecognition by learning to unfold. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [14] Denis Coquenot, Clément Chatelain, and Thierry Paquet. Span: A simple predict & align network for handwritten paragraph recognition. In *16th International Conference on Document Analysis and Recognition, ICDAR*, volume 12823 of *Lecture Notes in Computer Science*, pages 70–84, 2021.
- [15] Jorge Calvo-Zaragoza, Alejandro H Toselli, and Enrique Vidal. Handwritten music recognition for mensural notation with convolutional recurrent neural networks. *Pattern Recognition Letters*, 128:115–121, 2019.
- [16] Antonio Ríos-Vila, José M. Iñesta, and Jorge Calvo-Zaragoza. On the use of transformers for end-to-end optical music recognition. In *Pattern Recognition and Image Analysis*, pages 470–481, Cham, 2022. Springer International Publishing.
- [17] Antonio Ríos-Vila, David Rizo, José M. Iñesta, and Jorge Calvo-Zaragoza. End-to-end optical music recognition for pianoform sheet music. *Int. J. Document Anal. Recognit.*, 26(3):347–362, 2023.
- [18] Juan C. Martinez-Sevilla, Antonio Ríos-Vila, Francisco J. Castellanos, and Jorge Calvo-Zaragoza. A holistic approach for aligned music and lyrics transcription. In *Document Analysis and Recognition - ICDAR 2023 - 17th International Conference, San José, CA, USA, August 21-26, 2023, Proceedings, Part I*, volume 14187 of *Lecture Notes in Computer Science*, pages 185–201. Springer, 2023.
- [19] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Citeseer, 2006.
- [20] Denis Coquenot, Clément Chatelain, and Thierry Paquet. Dan: a segmentation-free document attention network for handwritten document recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8227–8243, 2023.
- [21] Sumeet S. Singh and Sergey Karayev. Full page handwriting recognition via image to sequence extraction. In *Proc. of the 16th Int. Conf. on Document Analysis and Recognition*, pages 55–69, 2021.

Few-Shot Music Symbol Classification via Self-Supervised Learning and Nearest Neighbor

María Alfaro-Contreras

Pattern Recognition and Artificial Intelligence Group, University of Alicante, Spain

malfaro@dlsi.ua.es

Abstract—The recognition of music symbols within score images represents one of the main stages in Optical Music Recognition systems. While current state-of-the-art methods based on Deep Learning are capable of adequately performing this task, they generally require a vast amount of data that has to be manually labeled. Such a particularity generally limits their applicability when addressing historical manuscripts with early music notation, for which annotated data is considerably scarce. In this paper, we propose a self-supervised learning-based method that addresses this task by training a neural-based feature extractor with a set of unlabeled documents and performs the recognition task considering just a few reference samples. Experiments on a reference early music corpus report that the proposal outperforms the contemplated baseline strategies even with a remarkably reduced number of labeled examples for the classification task.

Index Terms—Music Symbol Classification, Optical Music Recognition, Self-Supervised Learning, Few-shot Learning

I. INTRODUCTION

Music is a key element of cultural heritage. Throughout history, the primary means of transmitting and preserving this art form has been through the engraving of musical scores. Consequently, millions of musical documents exist only in physical form [19]. In recent years, many institutions have undertaken the digitization of numerous musical scores into images, making them accessible online. However, for these musical documents to be truly accessible, they must be transcribed into structured digital formats that allow for indexing, editing, or critical publication [13].

Optical Music Recognition (OMR) is the field of research that investigates how to computationally read music notation from documents and store them in a digital structured format [4]. OMR systems typically consist of a multi-stage pipeline [1], [14]: (i) *image pre-processing*, where issues related to the scanning process and paper quality are addressed, (ii) *symbol segmentation and classification*, where different image elements are detected and labelled, (iii) *reconstruction of the music notation*, a post-processing phase of the recognition process, and (iv) *output encoding*, where recognized elements are stored in a suitable symbolic format.

Traditional OMR systems achieve competitive recognition rates at the expense of relying on specific heuristics, tailored to the cases they were designed for [14]. Consequently, scalability becomes a significant limitation, as a new set of heuristics

must be designed for each collection. The incorporation of Deep Learning (DL) strategies into OMR produced a shift towards the use of *holistic* or *end-to-end* neural systems for the symbol segmentation and classification stage. These systems treat the recognition process as a single step rather than breaking it down into separate subtasks [4]. By simultaneously learning feature extraction and classification processes, these solutions eliminate the need for designing specific pipelines for each unique case, as the necessary features for classification are directly inferred from the data.

Although the use of DL systems offers a remarkable advantage over manual approaches, these models require to be specifically trained on the graphical domain to which the system is intended to be applied. This means that to retrieve information from a given collection, it is mandatory to manually annotate a representative portion of data that serves as a training corpus. Indeed, the amount of training data required to obtain acceptable results tends to be vast. This is currently a bottleneck when dealing with historical documents, as there is typically few labeled data available. New avenues of research are, therefore, exploring alternative methodologies that overcome this limitation.

Self-Supervised Learning (SSL) represents one of the most recent, yet competitive, paradigms within the DL field, aimed at reducing the large amount of labeled data required by DL models to learn [10]. Note that, while traditional supervised learning relies on human-annotated corpora, SSL is meant to learn through pseudolabeled data—where no human annotation is involved—to then converge in one or more downstream tasks (e.g., classification) [12]. Currently, this paradigm represents an effective solution to data-lacking scenarios, providing multipurpose state-of-the-art models [2], [7]. In this context, while we believe that music symbol classification can benefit from this approach, there exists a remarkable challenge to tackle. Symbols are typically labeled by determining not only their class but also their location in the document—i.e., marking the region they belong to with a bounding box. If SSL aims to be a possible solution, we need the methodology to work with completely unlabeled data, where not even the position of the symbols is known.

This work proposes a music symbol classification workflow that solves this scenario by sticking to the aforementioned premise. Our methodology comprises three stages: (i) data selection from completely unlabeled documents; (ii) pre-training of a DL model via SSL for its use as a feature extractor; and

This paper is part of the project I+D+i PID2020-118447RA-I00, funded by MCIN/AEI/10.13039/501100011033. The author is supported by grant FPU19/04957 from the Spanish Ministerio de Universidades.

(iii) a k -Nearest Neighbor (k NN) classification [8] for labeling query samples using a remarkably reduced reference set of labeled data—namely, few-shot classification—annotated by the user.

II. METHODOLOGY

This section presents the few-shot music symbol classification proposal of the work, which is graphically shown in Fig. 1. We train a neural network to generate an adequate feature representation space that can be then used by a classifier. This process comprises three stages: (i) the automatic extraction of isolated music symbols from unlabeled documents; (ii) training a neural feature extractor via SSL; and (iii) a classification phase that considers the nearest neighbor rule to label symbol queries. The rest of the section thoroughly describes these stages.

A. Stage I: Element extraction

The first stage of the proposal aims to extract all the possible existing categories from a collection of unlabeled documents, i.e., neither class nor location annotations of the symbols within are provided. In this regard, we propose an algorithm to automatically extract these pieces of information by subdividing each document into a set of image patches using a sliding-window approach and select those that may contain a symbol, referred to as *crops*, based on certain criteria.

Formally, let $\mathcal{D} = \{D_1, D_2, \dots, D_{|\mathcal{D}|}\}$ represent a set of document images. Additionally, let ω denote an image mask of size $s_h \times s_w$ pixels, which respectively correspond to its height and width dimensions. Window ω is moved from left to right and from top to bottom—starting from the top-left corner of the image—along all the documents in set \mathcal{D} with a striding factor of $\delta_h \times \delta_w$ pixels for the height and width dimensions, respectively, retrieving a set of patches $\mathcal{P} = \{p_1, p_2, \dots, p_{|\mathcal{P}|}\}$, being $p_i \in \mathbb{R}^{s_h \times s_w \times c_i}$ where c_i stands for the number of channels of the image with $1 \leq i \leq |\mathcal{P}|$.

After that, a filtering process is done on set \mathcal{P} to eliminate spurious patches for the subsequent stages by selecting a set $\mathcal{C} \subseteq \mathcal{P}$ of image portions, namely *crops*, with the following process: each patch $p \in \mathcal{P}$ is converted to gray scale—retrieving sample $p_{bn} \in \mathbb{R}^{s_h \times s_w}$ —and then binarized resorting to the Sauvola binarization method [15], hence obtaining sample $p_{bin} \in \{0, 1\}^{s_h \times s_w}$; the entropy value \bar{e} [16] is computed out of the p_{bin} patch and compared against a threshold e_{min} —user parameter—that, if exceeded, element p is included in set \mathcal{C} .

B. Stage II: Self-Supervised neural feature extraction

The second stage of the proposal aims to obtain a neural-based feature extractor—specifically, a Convolutional Neural Network (CNN)—in a self-supervised manner using the set of crops \mathcal{C} retrieved in the former stage. For that, we resort to the Variance-Invariance-Covariance Regularization (VICReg) method [3] due to its reported competitive overall performance in the related literature. In a broad sense, this strategy allows training a neural model in a self-supervised fashion based on

the so-called concepts of *variance*, *invariance*, and *covariance*, and whose embedded representation space meets certain conditions suitable for classification tasks.

To achieve this goal, the VICReg method initially draws an N -size batch of unlabeled image crops $X \subseteq \mathcal{C}$ that undergoes two independent image distortion processes, hence retrieving collections X_A and X_B . Note that, since these processes perform some controlled distortions in each of the images in the batch, the number of images for each collection remains the same, i.e., $|X_A| = |X_B| = N$. After that, sets X_A and X_B are mapped to an m -dimensional space by considering a function $f(\cdot)$ given by a CNN scheme, thus obtaining collections X_A^f and X_B^f , respectively. Note that this neural model represents the actual feature extractor to be retrieved as a result of this second stage of the proposal. Following this, an additional neural model—namely, *expander*—applies a transformation $h : \mathbb{R}^m \rightarrow \mathbb{R}^{m'}$ to sets X_A^f and X_B^f , hence producing X_A^h and X_B^h . It must be pointed out that, while not strictly necessary, this step is considered in the literature to improve the convergence of the scheme. Finally, the expanded sets X_A^h and X_B^h are used for computing the following loss function:

$$\begin{aligned} \ell(X_A^h, X_B^h) = & \lambda s(X_A^h, X_B^h) \\ & + \mu [v(X_A^h) + v(X_B^h)] \\ & + \phi [c(X_A^h) + c(X_B^h)] \end{aligned} \quad (1)$$

where $s(\cdot, \cdot)$ is the invariance term, which pulls these representations together in space and is computed as mean squared error, $v(\cdot)$ denotes the variance component, where a hinge loss is computed to ensure that elements within a batch are encoded differently, and $c(\cdot)$ is the covariance contribution, which encourages the generation of information-rich vectors and, therefore, prevents an informational collapse due to a high correlation between the variables. The λ , μ , and ϕ terms of the equation represent the respective regularization multipliers for the aforementioned loss components that are experimentally tuned.

C. Stage III: Classification

The final stage of the proposal performs the actual symbol classification considering the CNN-based feature extractor previously obtained. For that, we resort to the k NN classifier [8], which hypothesizes about the class of a given query attending to the labels of its closest k neighbors, based on a certain dissimilarity measure.

Formally, the initial query q and the labeled set of documents \mathcal{T} are mapped to the target m -dimensional representation space as q^f and \mathcal{T}^f using the CNN model obtained in the second stage of the proposal. After that, the k NN rule estimates the class y_q of query q as:

$$y_q = \text{mode} \left(\zeta \left(\underset{t \in \mathcal{T}^f}{\text{argmin}}_k \{d(q^f, t)\} \right) \right) \quad (2)$$

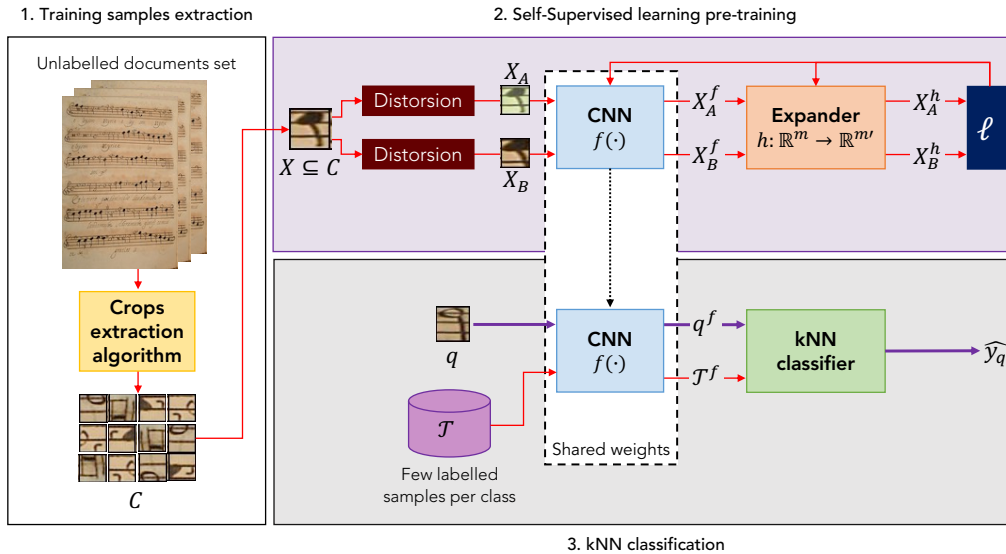


Fig. 1: General scheme of the proposed self-supervised workflow for this paper. Red arrows represent steps that are used during the workflow training—being the crop extraction and training of the neural network performed first and the k NN adjusting with the produced representations next—and the purple ones the inference process, where the representation of the given query is retrieved and then used in the k NN classifier to obtain the symbol class.

where $d : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}_0^+$ represents a dissimilarity measure, $\zeta(\cdot)$ stands for the function that outputs the label of the element in the argument, and $\text{mode}(\cdot)$ denotes the mode operator.

III. EXPERIMENTAL SETUP

A. Corpus

We consider the *Capitan* corpus [5], which comprises 74 handwritten scores from the 17th century of a *missa* (sacred music) in mensural notation. Each page of the manuscript is provided with annotations of the individual symbols in the different music staves. The corpus contains a total number of 17,112 running symbols of 53 different classes. We eventually derive two non-overlapping partitions—train and test—following a 10-iteration bootstrapping scheme for providing more robust performance figures.

B. Pipeline configuration

1) *Stage I: Element extraction*: The crop extraction proposal presented in Section II-A requires the specification of certain parameters to adequately address the process: the height s_h and width s_w dimensions of the image mask ω , the δ_h and δ_w displacement parameters of the sliding-window policy, and the e_{min} entropy threshold.

Based on preliminary experimentation, we eventually considered squared ω windows ($s_h = s_w = s$) of size $s = 64$ pixels with an striding factor equal to half of the window size, i.e., $\delta_h = \delta_w = s/2$. Regarding the entropy threshold, we set a rather restrictive value of $e_{min} = 0.8$ to avoid the selection of background or non-symbol-related data—such as staves or lyrics—, which would be detrimental to the self-supervised feature extractor.

2) *Stage II: Self-Supervised neural feature extractor*: During the training stage, and as aforementioned, the proposed feature extraction method requires two neural models: (i) a CNN architecture that processes the input images to obtain an adequate embedded representation—function $f(\cdot)$ —; and (ii) the Expander block that maps those features into a higher-dimensional space to ease convergence—function $h(\cdot)$.

We consider different CNN architectures for the encoder network: a standard ResNet-34 configuration [9] and a more lightweight architecture proposed in [11]. Note that, independently of the chosen configuration, the top layer is a fully connected network that maps the representation onto the target feature space with $m = 1,600$. In the case of the Expander block, the implementation from [3] is considered. We fix $m' = 1,024$ for the Expander target dimensionality as it yielded the most promising results in preliminary experiments.

Concerning the regularization multipliers for the VICReg loss, we use $\lambda = 10$, $\mu = 10$, and $\phi = 1$. Those values yielded the best convergence results in preliminary experimentation.

Finally, with regard to the image distortion processes followed to obtain sets X_A and X_B , we have resorted to a subset of those suggested in the work by [3] as they are proved to provide an adequate convergence of the neural model.

3) *Stage III: Classification*: Few-shot classification scenarios are defined by the N -way- L -shot approach: the training set contains N classes each with L examples [20], typically being $L < 10$ examples [17], [18]. When N equals the total number of symbols of the dataset, the method is referred to as L -shot classification, being this particular scenario the one addressed in this work. We, therefore, sub-sample the reference \mathcal{T} labeled set of images by randomly selecting L samples per class. For our experiments, we assess the influence of this particular pa-

parameter by considering $L \in \{1, 5, 10, 15, 20, 25, 30\}$ samples per class. It must be noted that results for values $L > 10$ are given for reference purposes as they do not constitute representative values for few-shot learning scenarios.

Regarding the k NN classifier, we set $k = 1$ to ensure $k \leq L$, i.e., the number of examples of the same class is always higher than—or, at least, equal to—the number of requested neighbors. Higher values may result in $k > L$, which would remarkably hinder the performance of the scheme.

C. Baseline approaches

To comparatively assess the performance of the presented proposal, several non-SSL methods have been considered to establish reference results.

The first one, namely *Flatten*, considers the individual pixels of the image as features, flattened as a one-dimensional vector. This approach serves as a reference for how the *classification* stage considered performs without a feature extraction process, i.e., raw unprocessed images.

The second approach is the use of the ResNet-34 network pre-trained with the ImageNet dataset [6], which currently represents one of the state-of-the-art models for image classification tasks. This baseline, denoted as *Pre-trained ResNet34* throughout the rest of the paper, establishes a reference on how transfer learning—a common approach for few-shot learning—works for this scenario.

To provide insights on the effectiveness of traditional supervised learning, the different CNN architectures described in Section III-B2 with a classifier layer are used. Note that, for a fair comparison with the rest of the strategies, these methods are trained with the same few-shot set \mathcal{T} .

Finally, to assess the impact of using algorithmically extracted symbols from unlabeled documents as training data, we also train the self-supervised feature extractor by extracting the symbols of the given corpora by their labeled bounding boxes. This approach is referred to as *Labeled crops* in later reports.

IV. RESULTS

This section discusses the results obtained with the considered experimental set-up. In this regard, Table I reports the average performance of both the proposed and baseline methods in terms of classification accuracy.

The first idea that can be observed is that the self-supervised methods—our approach and the *Labeled crops* one—report the most competitive results when performing few-shot symbol classification, i.e., $L < 10$. Specifically, it can be noticed that these methods achieve accuracy values above 60% with only one sample per class whereas the other alternatives depict performance rates below 40%.

When more examples per class are given, $L > 5$, the supervised learning approach outperforms the self-supervised when the chosen configuration is the ResNet-34 encoder network. The obtained results suggest that the use of lightweight neural architectures might favor the convergence of the SSL model. Moreover, as mentioned in Section III-B3, this work considers an N -way- L -shot approach in which N equals the

TABLE I: Mean accuracy (%) values obtained on the classification stage in the test set of the Capitan corpus for each train set \mathcal{T} size. Bold figures denote the best results for each \mathcal{T} labeled reference set size. The dashed line separates supervised (above) from self-supervised (below) strategies.

	1	5	10	15	20	25	30
Flatten	32.6	50.4	59.9	64.4	68.0	69.7	71.9
Pre-trained ResNet34	37.0	55.7	62.6	66.3	67.8	70.3	71.0
Supervised							
Nuñez-Alcover	41.3	65.6	78.1	83.1	86.7	88.7	90.0
ResNet34	31.0	78.4	90.1	93.3	96.1	96.7	97.1
Labeled crops							
Nuñez-Alcover	65.7	82.0	86.0	88.3	89.2	90.0	90.1
ResNet34	49.2	71.0	75.4	76.7	77.9	79.3	79.2
Our approach							
Nuñez-Alcover	67.2	82.0	86.9	88.9	89.9	90.1	90.5
ResNet34	34.3	52.0	58.6	62.4	64.8	66.0	67.2

total number of classes. The training set size starts from 530 samples when $L > 5$, which could be enough to favor the convergence of the supervised models. Nevertheless, the best results yielded by the two frameworks are slightly similar. Finally, note that pure transfer learning—represented by the *Pre-trained ResNet34* case—does not stand as a competitive solution in this scenario as the reported results are consistently worse than those achieved by the presented proposal.

Focusing on the SSL cases, the unlabeled-data approach obtains the best results, closely followed by the *Labeled crops* method. This suggests that when there are enough training samples, having unlabeled document images from the same corpus seems to be enough to achieve adequate performance, as our crop extraction algorithm produces a set that allows the VICReg method to converge into an easy-to-classify representation space.

V. CONCLUSIONS

This work presents a self-supervised learning method suited for few-shot music symbol classification scenarios. This proposal comprises three different stages: (i) a first stage where symbols are automatically retrieved from unlabeled corpus; (ii) a neural-based feature extractor trained in a self-supervised manner considering the so-called Variance-Invariance-Covariance Regularization loss to generate an adequate representation space; and (iii) a k -Nearest Neighbor classifier for performing the recognition task with a considerably limited set of reference data.

The reported results over a reference early music corpus show that the proposal outperforms the contemplated baseline strategies in terms of classification accuracy when less than 10 labelled samples per category are provided. More precisely, this strategy achieves more than 60% of accuracy considering one single example per class in the reference set—reference strategies achieve, at most, a 40% rate. When the labelled training set is increased in size, the overall best results are attained by the supervised learning framework, although closely followed by the self-supervised ones.

REFERENCES

- [1] Bainbridge, D., Bell, T.: The Challenge of Optical Music Recognition. *Computers and the Humanities* **35**(2), 95–121 (2001)
- [2] Bao, H., Dong, L., Wei, F.: BEiT: BERT Pre-Training of Image Transformers. In: *Proceedings of the 9th International Conference on Learning Representations*. pp. 1–18. OpenReview, Vienna, Austria (May 2022)
- [3] Bardes, A., Ponce, J., LeCun, Y.: VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. In: *Proceedings of the 10th International Conference on Learning Representations*. pp. 1–23. OpenReview, Online (Apr 2022)
- [4] Calvo-Zaragoza, J., Jr, J.H., Pacha, A.: Understanding Optical Music Recognition. *ACM Computing Surveys* **53**(4), 1–35 (2020)
- [5] Calvo-Zaragoza, J., Toselli, A.H., Vidal, E.: Handwritten Music Recognition for Mensural notation with convolutional recurrent neural networks. *Pattern Recognition Letters* **128**, 115–121 (2019)
- [6] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 248–255. IEEE, Miami, Florida, United States (Jun 2009)
- [7] Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota, USA (Jun 2019)
- [8] Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*. Wiley, 2nd edn. (2001)
- [9] 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*. pp. 770–778. IEEE, Las Vegas, Nevada, United States (Jun 2016)
- [10] Jing, L., Tian, Y.: Self-Supervised Visual Feature Learning with Deep Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**(11), 4037–4058 (2020)
- [11] Nuñez-Alcover, A., Ponce de León, P.J., Calvo-Zaragoza, J.: Glyph and Position Classification of Music Symbols in Early Music Manuscripts. In: *Proceedings of the 9th Iberian Conference on Pattern Recognition and Image Analysis*. pp. 159–168. Springer, Madrid, Spain (Jul 2019)
- [12] Ohri, K., Kumar, M.: Review on self-supervised image recognition using deep neural networks. *Knowledge-Based Systems* **224**, 107090–107111 (2021)
- [13] Pugin, L.: The challenge of data in digital musicology. *Frontiers in Digital Humanities* **2**, 4 (2015)
- [14] Rebelo, A., Capela, G., Cardoso, J.S.: Optical recognition of music symbols. *International Journal on Document Analysis and Recognition* **13**(1), 19–31 (2010). <https://doi.org/10.1007/s10032-009-0100-1>
- [15] Sauvola, J.J., Seppänen, T., Haapakoski, S., Pietikäinen, M.: Adaptive Document Binarization. In: *Proceedings of the 4th International Conference on Document Analysis and Recognition*. pp. 147–152. IEEE, Ulm, Germany (Aug 1997)
- [16] Shannon, C.E.: A Mathematical Theory of Communication. *Bell System Technical Journal* **27**(3), 379–423 (1948)
- [17] Snell, J., Swersky, K., Zemel, R.: Prototypical Networks for Few-shot Learning. In: *Advances in Neural Information Processing Systems*. vol. 30, pp. 1–11. Curran Associates, Inc., Long Beach, California, United States (Dec 2017)
- [18] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to Compare: Relation Network for Few-Shot Learning. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1199–1208. IEEE, Salt Lake City, Utah, United States (Jun 2018)
- [19] Treitler, L.: The Early History of Music Writing in the West. *Journal of the American Musicological Society* **35**(2), 237–279 (1982)
- [20] Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M.: Generalizing from a Few Examples: A Survey on Few-Shot Learning. *ACM Computing Surveys* **53**(3), 1–34 (2020)

A Preliminary Study of Few-shot Learning for Layout Analysis of Music Scores

Francisco J. Castellanos

University Institute for Computing Research
University of Alicante
Alicante, Spain
fcastellanos@dlsi.ua.es

Antonio Javier Gallego

University Institute for Computing Research
University of Alicante
Alicante, Spain
jgallego@dlsi.ua.es

Ichiro Fujinaga

Schulich School of Music
McGill University
Montreal, Canada
ichiro.fujinaga@mcgill.ca

Abstract—Few-shot techniques offer a promising avenue to reduce the high demand for annotated data required by current machine learning-based applications, such as Optical Music Recognition (OMR). This is a field dedicated to the automatic transcription of music notation from sheet music images. Traditional OMR systems strongly depend on layout analysis, a crucial step involving the identification and segmentation of several components within a music score, such as staff lines, text, or notes. The standard approach requires extensive fully annotated training data, which are resource-intensive and time-consuming to label and curate by domain experts. We present a preliminary study on the use of few-shot learning to alleviate the disadvantages associated with manual annotations. The proposal minimizes the human effort required by employing only partial annotations. For this, we introduce an oversampling technique to train models using a limited set of annotated patches extracted from the score images. Our experimental findings, conducted on four benchmark datasets, underscore the efficacy of the proposed patch extraction. Despite operating with a reduced amount of annotated data, our method achieves performance levels competitive with models trained on the complete dataset. This work points out the potential of few-shot learning in the context of layout analysis for music scores, offering the promise of more efficient and accessible OMR systems.

I. INTRODUCTION

Optical Music Recognition (OMR) is the field devoted to the transcription of music notation from document images into digital formats, a task of great importance for the preservation and analysis of musical heritage [1]. Among the tasks performed by OMR, one of the most critical is layout analysis, especially on older music manuscripts, which focuses on segmenting the image into essential parts, like staves, lyrics, and notes [2]. Traditional heuristic methods have historically been effective in layout analysis, but struggled with generalization, as they require an expert to adapt them to each application domain (or score type) [3].

Recent advances in layout analysis are predominantly based on deep learning methods, which have emerged as the prevailing trend because of their good performance. Particularly, we can find Convolutional Neural Networks (CNNs) for direct pixel classification and U-net-like architectures for patch-wise classification [4], [5], [6]. These techniques are more generalizable and obtain better results than the traditional approaches. However, they also have two important drawbacks: the requirement of a large amount of annotated data

for training and their high dependency on the domain of the training set used. This means that for each new application domain, it is necessary to have a labeled dataset. In the context of music scores analysis, these requirements pose a significant obstacle to scalability due to their inherent variability of scores in style and appearance.

To improve generalization and reduce the requirement for annotated data, several strategies, such as regularization techniques and data augmentation, have been explored in the literature [7]. In scenarios with limited annotated data, often called few-shot learning, specific neural architectures are employed to estimate data similarity [8]. Examples of such techniques include Siamese Neural Networks [9], Matching Networks [10], Prototypical Networks [11], and Relation Networks [12]. Our work aligns with the principles of few-shot learning but with the advantage of not requiring the use of a special architecture. Instead, it adapts a state-of-the-art layout analysis model: the Selectional Auto-Encoder (SAE) network [6]. For this, we introduce a new oversampling technique that allows training with a minimal collection of annotated data. While oversampling has found application in various contexts, its adaptation to binarization and few-shot learning scenarios by omitting unlabeled regions from the model input represents a novel contribution.

Experimentation carried out on four different corpora shows that the proposal obtains results on par with models trained with fully annotated data, but, in our case, only requiring the annotation of between 16 and 32 patches of a single page instead of having to label the entire dataset. Therefore, this approach substantially reduces manual annotation effort while maintaining high-performance standards.

II. FEW-SHOT LEARNING FOR LAYOUT ANALYSIS

Our approach addresses the challenge of few-shot learning for music score layout analysis based on an existing architecture for this task. In particular, we will rely on a previous work [6] that proposes the combined use of a series of SAE models, one for each layer of information to be predicted (implementation details are described in Section III-C). In this case, we focus on four categories: *staff*, *notes*, *text*, and *background*. Therefore, the objective is to classify each of the input image pixels into one of these four layers.

In this context, few-shot learning involves annotating a small subset \mathcal{S} of patches from a dataset of music score images with which to train the proposed network architecture. Thus, we create the annotated dataset $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i) : \mathbf{x}_i \in \mathcal{X}, \mathbf{y}_i \in \mathcal{Y}\}_{i=1}^N$, which matches each patch \mathbf{x} from the set of all the possible patches \mathcal{X} to be annotated sequentially from the input images and without overlap, with their corresponding pixel-wise annotations \mathbf{y} . We ensure a minimum coverage of $\lambda\%$ of annotated pixels, discarding those patches that do not contain annotated information of the layer to be detected. The size of this subset, denoted by N , is a parameter that will be studied during experimentation in order to analyze its effect as it is reduced. The key advantage of few-shot learning is the reduction of the effort and cost of manual annotation, so to this end, small values for the parameter N will be considered.

However, annotating only a portion of an image can lead to overfitting. Traditional data augmentation methods are less effective in few-shot learning scenarios due to the limited amount of data to manipulate. Our proposal introduces an original oversampling approach to generate more varied samples by drawing on the few annotated data available. Specifically, we extract random patches around annotated regions, ensuring that they contain annotated pixels to be used for the model in the training process. These patches introduce variability in positions and labels of elements within the window, acting as a data augmentation mechanism. This method allows us to control the number of samples extracted from images and thus increase the variability of the data. Note that parts of the extracted patches may fall outside annotated areas, although experimentation has shown this to be advantageous as it appears to act as a regularizer with which to avoid overfitting.

III. EXPERIMENTS

A. Corpora

In our experiments, we employed four datasets with different characteristics, each comprising manual pixel-wise annotations for the four layers considered: staff lines, notes, text, and background. These datasets are as follows:

- Einsiedeln (EIN): It consists of 9-page images of neumatic notation from the Einsiedeln, Stiftsbibliothek, Codex 611(89), dating back to 1314.¹ They have an average resolution of $6,496 \times 4,872$ pixels.
- Salzinnes (SAL): It comprises 10-page images from the Salzinnes Antiphonal manuscript (CDM-Hsmu M2149.14), containing neumatic notation. The average resolution of the images is around $5,847 \times 3,818$ pixels.²
- MS Medieval 0073 (MS73): It includes 10 pages of square notation from the ‘*Dominican, CDN-Mlr MS Medieval 0073*’ choir book, originating in Northern Italy between the 13th and 15th centuries.³ The images have an average resolution of $6,990 \times 4,797$ pixels.

¹<http://www.e-codices.unifr.ch/en/sbe/0611/>

²<https://cantus.simssa.ca/manuscript/123723/>

³<https://cantus.simssa.ca/manuscript/680970/>

- Capitan (CAP): A collection of mensural notation manuscripts from the 17-18th centuries, related to the ‘*Cathedral of Our Lady of the Pillar*’ in Zaragoza (Spain) [13]. We employed a subset of 10 manually pixel-wise annotated pages. The average resolution of these images is $2,126 \times 3,065$ pixels.

Each dataset was divided into train, validation, and test sets, with four images for training, two for validation, and the remainder for testing. For the experiments, annotated patches of size 256×256 pixels were extracted from these images. Models trained with from 1 to 32 patches were compared with the state-of-the-art solution trained using the complete annotation of the corpus. It is important to note that the number of samples in the validation set was adjusted to the size of the training data, using approximately 30% of the training patches. In this way, a more realistic scenario was simulated where the available manual annotation is limited.

B. Metrics

For the experimentation, we considered the F-score (F_1) to perform the quantitative comparison of the results. In a binary classification setting, this metric is formally defined as:

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}, \quad (1)$$

where TP, FP, and FN respectively denote the True Positives, False Positives, and False Negatives.

Given the multi-class nature of our task, we resort to the macro-average F-score (F_1^m), which computes the average F_1 across all layers of information. This is defined as:

$$F_1^m = \frac{\sum_{l=1}^{|\mathcal{L}|} F_1^l}{|\mathcal{L}|}, \quad (2)$$

where F_1^l represents the F_1 calculated for layer l under a one-versus-all evaluation framework, and $|\mathcal{L}|$ denotes the total number of layers of information, which is four in our case.

C. Implementation details

As stated before, our proposal builds upon a prior work [6] and employs a framework featuring a series of SAE models, one for each layer of information to predict. Each SAE architecture adopts a U-Net-like structure, where a patch of 256×256 pixels is used as input. The output of the model is a matrix with the same size as the input, whose values represent the confidence for each pixel belonging to a specific layer.

As in the original work, the first part of the architecture is an encoder with four blocks composed of 32-filter 3×3 convolutions, 2×2 sub-sampling, batch normalization, Rectified Linear Unit (ReLU) activation, and 0.4 of dropout. The decoder part mimics the encoder but replaces sub-sampling with an oversampling operation. The final layer is a convolution with a single 3×3 filter with sigmoid activation to yield predictions within the range $[0, 1]$.

The binary cross-entropy loss was used to train each SAE with a batch size of 16 and during a maximum of 200 epochs.

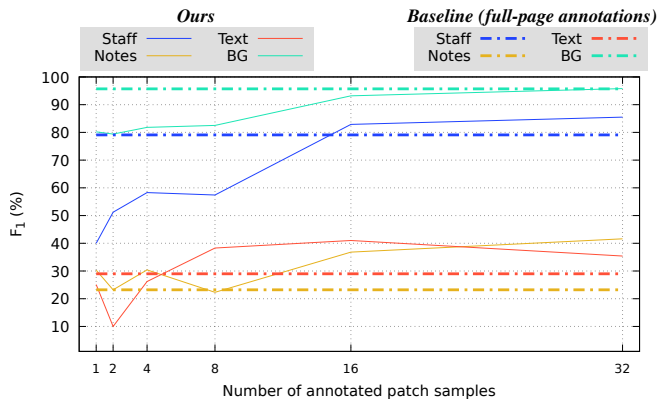


Fig. 1: F_1 (%) results for one page varying the number of annotated samples. The baseline represents the case in which the full page was annotated.

Early stopping was applied when there was no improvement on the validation set for 20 consecutive epochs. Adam [14] was used as the optimization algorithm with a learning rate of 0.001.

To facilitate model convergence, input images were normalized to the $[0, 1]$ range. A λ value of 2.5% was employed for patch extraction to ensure the acquisition of informative chunks and also to simulate a simple manual selection. Standard data augmentation techniques were also considered, introducing random rotations within the range of -45° to 45° , zoom variations between 0.8x and 1.2x, as well as horizontal and vertical flips to enhance data variability. However, in preliminary experiments, we observed that this traditional data augmentation did not contribute to improving the results when combined with our approach in few-shot scenarios, and employing our oversampling approach was sufficient. Therefore, we opted not to employ it in our final experiments. In this preliminary experimentation, we also analyze the influence of the number of patches extracted using the oversampling technique from each of the labeled windows. Based on these results in the validation partition, we selected a sampling size of 512 for the final experiments.

IV. RESULTS

This section presents the results obtained from our proposed methodology. In order to assess the capability of the proposed oversampling method, we conducted an analysis of two key factors: the number of annotated patches per page (ranging from 1 to 32) and the impact of extracting patches from the same page or from multiple pages (in this case, up to 4 pages).

Concerning the influence of the number of annotated patches, Fig. 1 shows a summary of the results obtained for the four layers when the model is trained with only one page, representing the most challenging case due to limited available variability for annotation. In this experiment, the oversampling approach is compared with the model trained with full-page annotations, considered as the baseline [6]. Our approach’s performance improves as the number of annotated samples

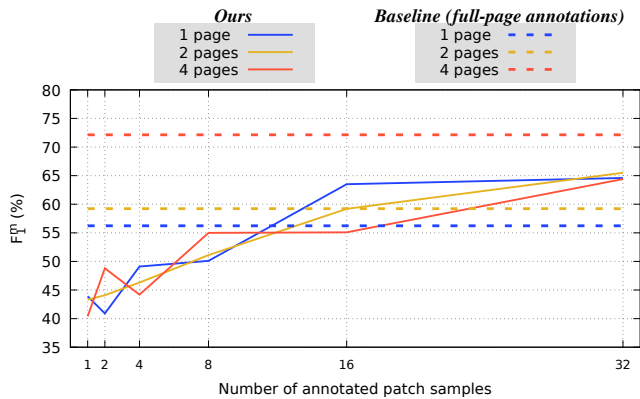


Fig. 2: F_1^M (%) results for scenarios with different numbers of pages and annotated samples per page. The baseline represents the case with full-page annotations.

increases. We can observe that our method exhibits significant improvement over the baseline across all layers when using sufficient annotated data. However, the number of required annotated patches varies depending on the layer. In general, our approach improves the baseline case when annotating 16 or more patches. For simpler and more homogeneous layers like *staff* and *BG*, the baseline performs well, making it harder for our proposal to surpass it. Nevertheless, we observe a considerable improvement for the *staff* layer when 16 or more samples are annotated, from 79.1% to 85.5% of F_1 in the best of the cases, but a modest improvement from 95.7% to 95.8% for the *BG* layer (in this second case with 32 patches). Note that the high performance of the baseline in the *BG* layer makes it challenging to achieve substantial improvements. Conversely, for complex layers with greater variability, such as *notes* or *text*, our approach exhibits a wide margin of improvement with respect to the baseline. In the case of *text*, we observed that 8 samples are sufficient to improve the performance from 29% to 38.3%, while the *notes* layer experiences an enhancement from 23.2% to 36.8% when 16 samples are annotated, and 41.6% in the best case by annotating 32 samples. These results show that the oversampling method is beneficial when scarce data is available—only one page in this case.

Concerning the second factor to evaluate, which is the number of pages used in the training process, Fig. 2 shows the average results of our method compared to those of the baseline. For clarity, given the amount of results to be displayed (different number of pages, patches, and each one for 4 layers), the average (F_1^m metric) is calculated by grouping the results of the different layers. As can be seen, all three cases (with 1, 2, or 4 pages) exhibit similar trends for our oversampling method. As expected, the results show an increasing trend with the number of annotated samples, ranging from an average F_1^m of 40%, when trained with only one annotated sample per page, to approximately 62% with 32 annotated patches. Beyond 16 annotated patches, the improvement stabilizes or decreases slightly.

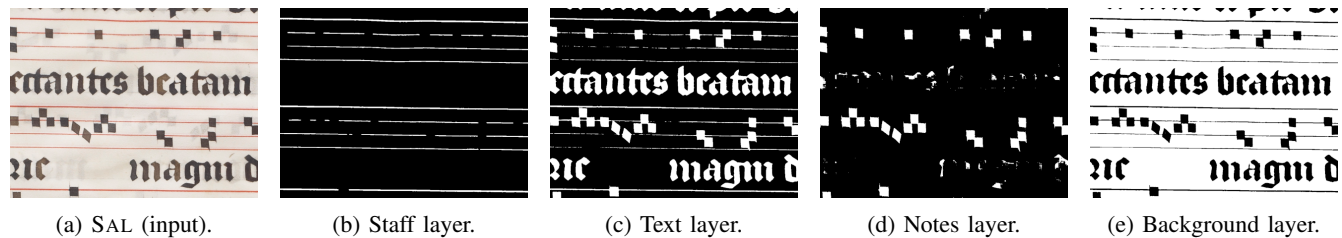


Fig. 3: Example prediction for the SAL manuscript with the four layers considered. In this example, 32 annotated samples from a single page were used for training the models. The detected information is represented in white.

In comparison with the baselines, our approach outperforms the baseline in cases with 1 and 2 pages when using 16 annotated samples. Even in the 4-page scenario, the proposal is only 7% worse than the state-of-the-art result, a remarkable achievement given its substantially reduced requirement for annotated data (32 patch samples). The most interesting point is that, in these experiments, the number of pages seems to be not relevant. This means that by using only 32 patches from a single page, we can obtain competitive results compared with those obtained by using more than one page. These results also reveal that there is still room for improvement compared to the baseline. However, considering that we only need to label 32 samples from one page to almost reach the baseline performance, this represents a notable practical improvement. Observe that in the results, the baseline only achieves competitive results when using 4 pages, which would require labeling around 1 450 samples in total (363 per page on average), compared to the 32 needed for our method.

As a reference, Table I provides an estimate of the time it would take an expert to manually label the 4 pages required to train the baseline model. This estimation is based on the average time it takes to annotate a high-resolution page. Please note that the actual time may depend on several factors, such as the skills of the annotator, the use of tools that facilitate the process, and the density of information and level of detail of each manuscript. Taking all this into account, the baseline scenario, which involves labeling 4 pages, would require an average of 11.2 hours. As can be seen, the few-shot proposal would reduce this human effort to almost fifteen minutes to label the 32 patches of 256×256 pixels. This represents a substantial reduction, indicating that, in practice, our approach is a potential solution for large-scale labeling in layout analysis.

To complement the quantitative results, Fig. 3 shows an example of the prediction obtained using the proposal for the SAL manuscript. The background and staff layers are almost perfectly retrieved, with minor false positives in the notes layer. The text layer has more errors, as it is more difficult to differentiate text from other elements. However, text is recovered and false positives can potentially be eliminated by combining predictions from the other layers.

TABLE I: Comparison of the time (estimate in hours) to perform the manual annotation required to train the baseline model and the proposal. These estimates have been calculated considering that an expert takes an average of 4 hours to annotate a page of the data set MS73. The annotation was supported by a graphical tool and SAE models trained with partial annotations that make preliminary predictions to be corrected manually.⁴

	EIN	SAL	MS73	CAP	Avg.
Baseline (4 full-page annotations)	14.8	10.4	16.0	2.8	11.2
Ours (32 patch samples)	0.24	0.24	0.24	0.24	0.24

V. CONCLUSION

In this study, we have introduced a few-shot neural approach for pixel-wise layout analysis of music score images. The proposal is based on an oversampling technique to maximize the utility of the limited annotations available. For this, the method randomly extracts patches at different random positions during each training iteration around the limited annotated data, resulting in a set of samples with variations in the position of the components of the image. This may be considered a data augmentation technique since a single annotated patch could be used to generate any desired number of samples. This approach is therefore suitable for minimizing the necessary manual annotation for training models.

Experiments were conducted on four benchmark datasets in order to analyze the impact of the amount of annotated data on the layout analysis task. Our findings show that by labeling a relatively small number of samples—between 16 and 32 samples—the proposal achieved an impressive average F_1^m of 65.5% in the best case. This is remarkably close to the state-of-the-art result (72%) obtained when employing the fully annotated training set. It is noteworthy that the proposal obtains similar results when annotating multiple pages, indicating that the method only needs to be trained with partial single-page annotations. These results are very promising for layout analysis and OMR, as they represent a significant reduction in annotated data requirements and manual costs.

Future research will focus on exploring domain adaptation and transfer learning techniques to further reduce human effort. The feasibility of incremental and active learning approaches to address this challenge will also be considered.

⁴<https://rodan2.simssa.ca/>

ACKNOWLEDGMENT

This research was supported by the I+D+i project TED2021-132103A-I00 (DOREMI), funded by MCIN/AEI/10.13039/501100011033, the Social Sciences and Humanities Research Council (895-2013-1012) and the Fonds de recherche du Québec-Société et Culture (2022-SE3-303927).

REFERENCES

- [1] D. Bainbridge and T. Bell, “The challenge of optical music recognition,” *Computers and the Humanities*, vol. 35, no. 2, pp. 95–121, 2001.
- [2] J. Calvo-Zaragoza, J. H. Jr., and A. Pacha, “Understanding optical music recognition,” *ACM Computing Survey*, vol. 53, no. 4, Jul. 2020.
- [3] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Marçal, 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] A. Gallego and J. Calvo-Zaragoza, “Staff-line removal with selectional auto-encoders,” *Expert Systems with Applications*, vol. 89, pp. 138–48, 2017.
- [5] J. Calvo-Zaragoza, G. Vigiensoni, and I. Fujinaga, “One-step detection of background, staff lines, and symbols in medieval music manuscripts with convolutional neural networks,” in *Proceedings of the 18th International Society for Music Information Retrieval Conference, Suzhou, China, 2017*, pp. 724–730.
- [6] F. J. Castellanos, J. Calvo-Zaragoza, G. Vigiensoni, and I. Fujinaga, “Document analysis of music score images with selectional auto-encoders,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23–27, 2018*, E. Gómez, X. Hu, E. Humphrey, and E. Benetos, Eds., 2018, pp. 256–263. [Online]. Available: http://ismir2018.ircam.fr/doc/pdfs/93_Paper.pdf
- [7] X. Li, L. Yu, C.-W. Fu, M. Fang, and P.-A. Heng, “Revisiting metric learning for few-shot image classification,” *Neurocomputing*, vol. 406, pp. 49–58, 2020.
- [8] C. Simon, P. Koniusz, R. Nock, and M. Harandi, “Adaptive subspaces for few-shot learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4136–4145.
- [9] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in *International Conference on Machine Learning (ICML) - Deep Learning Workshop*, vol. 2, 2015, pp. 1126–1135.
- [10] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, “Matching networks for one shot learning,” *Advances in Neural Information Processing Systems*, vol. 29, pp. 3630–3638, 2016.
- [11] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in *Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.
- [12] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, “Learning to compare: Relation network for few-shot learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.
- [13] J. Calvo-Zaragoza, D. Rizo, and J. M. I. Quereda, “Two (note) heads are better than one: Pen-based multimodal interaction with music scores,” in *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States, August 7–11, 2016*, 2016, pp. 509–514.
- [14] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: <http://arxiv.org/abs/1412.6980>