From image to encoding: Full optical music recognition of Medieval and Renaissance music

GABRIEL VIGLIENSONI, ALEX DAIGLE, ERIC LIU, JORGE CALVO-ZARAGOZA, JULIETTE REGIMBAL, MINH ANH NGUYEN, NOAH BAXTER, ZOÉ MCLENNAN, and ICHIRO FUJINAGA

Several centuries of manuscript music sit on the shelves of libraries, churches, and museums around the globe. On-line digitization programs are opening these collections to a global audience, but digital images are only the beginning of true accessibility since the musical content of these images cannot be searched by computers. In the SIMSSA (Single Interface for Music Score Searching and Analysis) project [4] we aim at teaching computers to read music and assemble the data on a single website, allowing users to have access to millions of musical works. In this paper, we describe our full workflow to perform optical music recognition (OMR) of Medieval and Renaissance music sources.

CCS Concepts: • **Applied computing → Sound and music computing**; **Document analysis**; **Graphics recognition and interpretation**;

Additional Key Words and Phrases: Optical music recognition, document analysis, music encoding

1 INTRODUCTION

Our aim is to read, extract, and encode the content from digitized images of Medieval and Renaissance music documents. For high scalability, we are taking a machine learning-based approach to OMR. Instead of using heuristics and features that take advantage of document-specific characteristics, we train the computer with a large number of examples for each category of musical element to be classified and create a model. Once a model is created, it is used to classify new examples that the computer has not yet seen. We have implemented this approach in several steps of a workflow to perform OMR in Medieval and Renaissance music scores images.

2 OMR WORKFLOW FOR MEDIEVAL AND RENAISSANCE MUSIC

Our current OMR workflow is divided into four stages: *document analysis*, *symbol classification*, *music reconstruction and encoding*, and *symbolic score generation and correction*. The entire workflow is depicted in Figure 1.

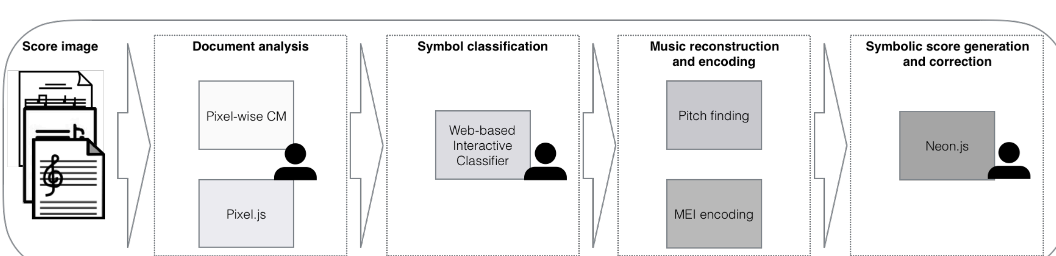

Fig. 1. Full optical music recognition workflow for Medieval and Renaissance music. Boxes indicate the software applications on each step. Human symbols indicate interactive, adaptive stages.

2.1 Document analysis

The first stage in our workflow is *document analysis*. Digitized music scores are the input to the system and document analysis is applied to segment the music document into layers. We use *Pixel.js* [7], an open-source, web-based, pixel-level classification application, to label pixels into their corresponding musical category or to correct the output of other image segmentation processes. We use this tool interactively with a convolutional neural network-based classifier [2] to segment the document into a number of user-defined layers. After a few iterations of training and classification for optimizing the classifier, we obtain a number of image files corresponding to the segmented layers of the original score. For example, these layers may contain notes, staff lines, lyrics, annotations, or ornamental letters. The recognition of the music symbols and the analysis of their relationship is achieved once the symbols are isolated and classified in the found layers.

2.2 Symbol classification

The application we use for the *symbol classification* stage is called *Interactive Classifier* (IC). IC is a web-based version of the Gamera classifier [3]. We use it to automatically group the connected components of a specific layer into *glyphs*. Then, we manually label a series of these musical glyphs into classes. For Medieval and Renaissance music we implement neume-based and neume component-based classification. In either case, IC will extract a set of features for describing each of the neume or neume component classes and will model a classifier. With this model, new, unseen glyphs will be classified based on k-nearest neighbors. Once the symbols of the score are classified, we proceed to add their musical context and encode them into a symbolic music format.

2.3 Music reconstruction and encoding

We obtain the pitches of neumes or neume components by finding their absolute position in the corresponding staves and use the recognized clef of each system to assign a relative pitch [8]. The output of IC conveys the position and size of each musical element in the score image, and so we add this information to the estimated pitch as well as the staff number to which each neume or neume component belongs. Finally, the retrieved musical information is encoded into the machine-readable, symbolic music format MEI 4.0 using the new `MEI.neumes` module.

2.4 Symbolic score generation and correction

The last two stages of our OMR workflow, *music reconstruction and encoding* and *symbolic score generation* have a common interactive checkpoint for visualizing and correcting the output of the automatized OMR process. This human-driven checkpoint is implemented using a web-based application called *Neon* (Neume Editor Online) [1]. Neon allows a user to inspect differences between the original music score image and the rendered version of the output of the OMR process. By visual inspection of the two overlaid scores, the user can observe their difference and manually add, edit, or delete music symbols in the browser. So far, however, corrections entered by the user are not fed back into the learning system, but they change the encoded music file output.

2.5 Workflow management system

All the constituent parts of our OMR workflow are handled by Rodan [5], a distributed, collaborative, and networked adaptive workflow management system that allows to specify *interactive* and *non-interactive* tasks.

3 FUTURE WORK

The automatic retrieval and encoding of music from score images has many complexities and there is much space for improvement. In future iterations of the project we will focus on: (i) implementing a non-heuristic, machine learning-based approach for pitch finding (similar to the approach proposed by Pacha and Calvo-Zaragoza [6]); (ii) appending neumes to syllables (since most neume notation is used to set music to an existing text and the MEI 4.0 Neume module was designed for this); and (iii) devising a way of feeding back into the workflow the corrected output in Neon. We hope that this infrastructure, in combination with the proper teaching strategies and tactics developed by human teachers in the interfaces for training OMR systems [9], will enable the full recognition and encoding of music from music score images.

REFERENCES

[1] Gregory Burlet, Alastair Porter, Andrew Hankinson, and Ichiro Fujinaga. 2012. Neon.js: Neume Editor Online. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*. Porto, Portugal, 121–126.

[2] Jorge Calvo-Zaragoza, Francisco J. Castellanos, Gabriel Vigliensoni, and Ichiro Fujinaga. 2018. Deep neural networks for document processing of music score images. *Applied Sciences* 8, 5 (2018), 654–674.

[3] Michael Droettboom, Karl MacMillan, and Ichiro Fujinaga. 2003. The Gamera framework for building custom recognition systems. In *Proceedings of the 2003 Symposium on Document Image Understanding Technologies*. Greenbelt, MD, 275–286.

[4] Ichiro Fujinaga, Andrew Hankinson, and Julie E Cumming. 2014. Introduction to SIMSSA (Single Interface for Music Score Searching and Analysis). In *Proceedings of the 1st International Workshop on Digital Libraries for Musicology*. 1–3.

[5] Andrew Hankinson. 2015. *Optical Music Recognition Infrastructure for Large-scale Music Document Analysis*. Ph.D. Dissertation. McGill University, Montréal, QC.

[6] Alexander Pacha and Jorge Calvo-Zaragoza. 2018. Optical music recognition in mensural notation with region-based convolutional neural networks. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*. Paris, France, 23–27.

[7] Zeyad Saleh, Ké Zhang, Jorge Calvo-Zaragoza, Gabriel Vigliensoni, and Ichiro 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.

[8] Gabriel Vigliensoni, John Ashley Burgoyne, Andrew Hankinson, and Ichiro Fujinaga. 2011. Automatic pitch recognition in printed square-note notation. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*. Miami, FL, 423–428.

[9] Gabriel Vigliensoni, Jorge Calvo-Zaragoza, and Ichiro Fujinaga. 2018. An environment for machine pedagogy: Learning how to teach computers to read music. In *Proceedings of IUI Workshop on Music Interfaces for Listening and Creation*. Tokyo, Japan.