Glyph and Position Classification of Music Symbols in Early Music Manuscripts

Alicia Nuñez-Alcover(✉), Pedro J. Ponce de León, and Jorge Calvo-Zaragoza

Department of Software and Computing Systems, University of Alicante,
Carretera de San Vicente s/n, 03690 Alicante, Spain
ana27@alu.ua.es, {pierre,jcalvo}@dlsi.ua.es

Abstract. Optical Music Recognition is a field of research that automates the reading of musical scores so as to transcribe their content into a structured digital format. When dealing with music manuscripts, the traditional workflow establishes separate stages of detection and classification of musical symbols. In the latter, most of the research has focused on detecting musical glyphs, ignoring that the meaning of a musical symbol is defined by two components: its glyph and its position within the staff. In this paper we study how to perform both glyph and position classification of handwritten musical symbols in early music manuscripts written in white Mensural notation, a common notation system used for the most part of the XVI and XVII centuries. We make use of Convolutional Neural Networks as the classification method, and we tested several alternatives such as using independent models for each component, combining label spaces, or using both multi-input and multi-output models. Our results on early music manuscripts provide insights about the effectiveness and efficiency of each approach.

Keywords: Optical Music Recognition · Handwritten symbol classification · Convolutional Neural Networks · Digital preservation

1 Introduction

Music constitutes one of the main vehicles of cultural heritage. A large number of musical manuscripts are preserved in historical archives. Occasionally, these documents are transcribed to a digital format for its easier access and distribution, without compromising their integrity. However, in order to make these heritage really useful, it is necessary to transcribe the sources to a structured format such as MusicXML [3], MEI [11], or MIDI. So far, this has been done manually by experts in early music notation, making the process very slow and costly. Conveniently, Optical Music Recognition (OMR) techniques can help automating the process of reading music notation from scanned music scores [1].

A. Morales et al. (Eds.): IbPRIA 2019, LNCS 11868, pp. 159–168, 2019.
https://doi.org/10.1007/978-3-030-31321-0_14

Typically, the transcription of historical music documents is treated differently with respect to conventional OMR methods due to their particular features; for instance, the use of certain notation systems, or the state of preservation of the original document. Although there exist several works focused on early music documents transcription [4,8], the specificity of each type of manuscript, or its overall writing style makes it difficult to generalize these developments.

In this context, a music transcription system is one that performs the task of obtaining a digital structured representation of the musical content in a scanned music manuscript. The workflow to accomplish such a task could be summarized as follows: First, a document layout analysis step isolates document parts containing music, mostly music staffs. Then, a OMR system detects music symbols contained in these parts, typically producing a sequence or graph of music symbols and their positions with respect to the staff. From this representation, a semantic music analysis step assigns musical meaning to each symbol, as this often depends on the specific location of the symbol in the sequence. Finally, this intermediate music representation is translated by a coding stage into a structured representation in the desired output format.

Unlike other domains, in the particular case of music notation the symbols to be classified have two components: glyph and position in the staff. Traditionally, OMR systems use supervised learning to predict the glyph [2,6,9], whereas the position is determined by heuristic strategies [13]. Since these OMR systems usually perform a pre-process that normalizes the input images (binarization, deskewing, and so on), these heuristic strategies tend to be quite reliable. However, other approaches might use different pre-processing steps, and so traditional heuristics might not work correctly. This is why we propose to deal with the identification of the glyph position by means of supervised learning as well. To this end, we study the best approach to perform classification of a pre-segmented symbol image into its pair of glyph and position. Specifically, we propose different deep learning architectures that make use of Convolutional Neural Networks (CNN) in order to fully classify a given music symbol. We aim to analyze which architecture gives us the best performance in terms of accuracy and efficiency.

The rest of this paper is organized as follows: Sect. 2 presents the methodology considered for the aforementioned task; Sect. 3 describes our experimental setup; Sect. 4 presents and analyzes the results: Sect. 5 concludes the present work and introduces some ideas for future research.

2 Methodology

Our work assumes a segmentation-based approach, in which the locations of symbols that appear in the input music score have already been detected in a previous stage. This can be achieved under an interactive environment where the user manually locates the symbols [10], but can also be automated with object detection techniques [7]. Either way, the next task is to fully classify the symbols in order to retrieve their musical meaning.

We consider that all musical symbols are defined by two components: glyph and position with respect to the staff lines. This is obvious in the case of notes,

as these components indicate the duration and the pitch, respectively. We can generalize this to any type, as all symbols are located in a specific position with respect to the lines of the staff. Let $\mathcal{G}$ be the label space for the different glyphs and $\mathcal{P}$ the label space for the different positions. A music symbol is therefore fully defined by a pair $(g, p), s \in \mathcal{G}, p \in \mathcal{P}$. A graphical example is given in Fig. 1.

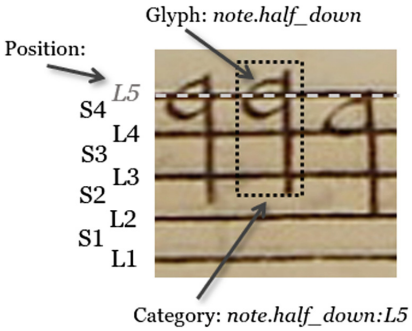

Fig. 1. Example of handwritten music symbols in white Mensural notation, showing its glyph, position, and combined label. Note that position labels refer to the vertical placement of a glyph: *Ln* and *Sn* denote symbol positions over or between staff lines, respectively.

Considering the above, the complete classification of a music-notation symbol consists in predicting both its glyph and its position. This opens up several possibilities as regards this dual process, given that the two components are not completely independent. As a base classification algorithm, we resort to Convolutional Neural Networks (CNN), since they represent the state of the art for image classification. These neural networks consist of one or more convolutional layers, which benefits from local connections, tied weights and pooling operations in order to learn a suitable data representation for the task at hand [14]. The convolutional layers are typically followed by one or more fully-connected layers that perform the final prediction stage.

The classical use of CNN is to consider a single image as input, that must be associated with a single class label. However, since we want to know the glyph of a symbol as well as its position within the staff simultaneously, we shall consider different architectures with shared layers, and multi-output and multi-input models, in order to determine which is the best way to obtain the corresponding full symbol classification.

Below we first introduce how to represent the input for classification, after which we propose the neural architectures that allow us to perform the combined glyph and position classification of music symbols.

2.1 Input Scheme and Preprocessing

Two region-based image inputs are used in this approach. Figure 2 shows a part of a music staff from a Mensural notation score. These images are pre-segmented, either manually or automatically, by defining a bounding box around each music symbol in the staff, as shown in Fig. 3 (left). Thus, each bounding box defines an image instance containing a music symbol. These appropriately annotated images can be used to train and evaluate a music glyph classification model. However, in general these instances do not span vertically as to contain all staff lines. Therefore, they do not convey information about the music symbol position. For example, the symbol labeled as 'A' in Fig. 3 (left) is indistinguishable from symbol 'B' in the same image in spite of appearing at different staff positions. This type of images will be referred as *glyph inputs*.

In order to produce models able to correctly classify music symbol positions, a second image set is constructed by enlarging the bounding box frame vertically to a fixed height large enough to contain all staff lines, as shown in Fig. 3 (right). This type of images will be referred as *enlarged inputs* in the following sections. It has been shown that these enlarged images contain enough information for estimating the vertical position of a music symbol within the staff [5].

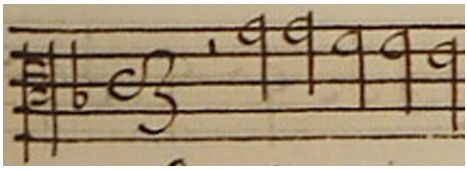

Fig. 2. A sample of a Mensural notation staff.

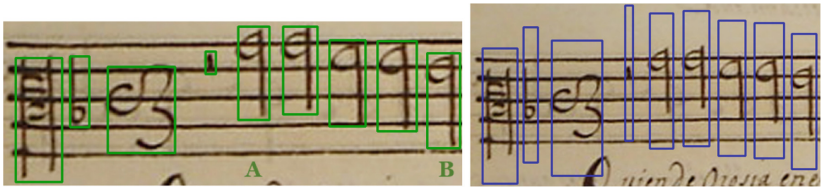

Fig. 3. Left: glyph bounding boxes. Right: enlarged bounding boxes.

2.2 CNN Architectures

Given the aforementioned inputs and targets, we intend to classify every region as one of the available symbols. To accomplish this, we try different approaches that perform the different classifications either simultaneously or independently.

Independent Glyph and Position Models. Our first approach would be to create two different CNN models: one processes a glyph input x_g, and tags it with a glyph label $g \in \mathcal{G}$. The other one processes a enlarged inputs x_p, and tags it with a position label $p \in \mathcal{P}$. These two models are depicted in Fig. 4.

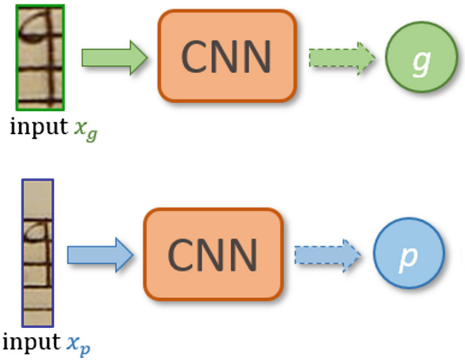

Fig. 4. Top: independent glyph classification model. Bottom: independent position classification model.

Category Output Model. Another approach uses a single enlarged input x_p, and tags it by considering as the label set the Cartesian product of $\mathcal{G}$ and $\mathcal{P}$. We shall refer to the combined label of a symbol as its *category*, denoted by $\mathcal{C}$. Therefore the model tags each input x_p as a pair $c = (g, p)$, such that $g \in \mathcal{G}$ and $p \in \mathcal{P}$. This approach is depicted in Fig. 5.

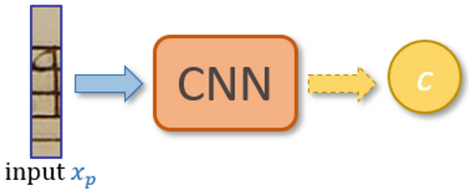

Fig. 5. Category output model: enlarged inputs are provided as input and the model must predict a label from the Cartesian product of glyphs and positions.

Category Output, Multiple Inputs Model. This model uses both glyph and enlarged input images, yet predicting directly the combined category c (Fig. 6).

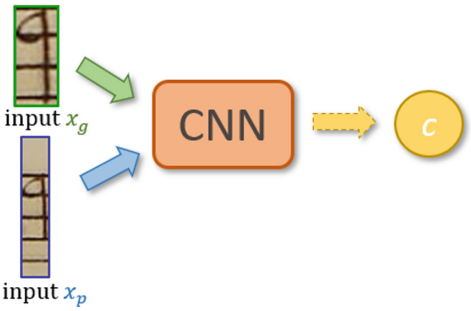

Fig. 6. Category output with multiple inputs: both glyph bounding box and enlarged images are provided as input and the model must predict a label from the Cartesian product of glyphs and positions.

Multiple Outputs Model. This model takes enlarged inputs and predicts the glyph g and position p labels separately, as shown in Fig. 7. The model shares the intermediate data representation layers as input to both final fully-connected classification layers.

Fig. 7. Multiple outputs model: enlarged images are provided as input and the model must predict both the glyph and the position separately.

Multiple Inputs and Outputs Model. Our last model takes both glyph and enlarged inputs x_g and x_p, and predicts glyph g and position p labels as two different outputs, as depicted in Fig. 8.

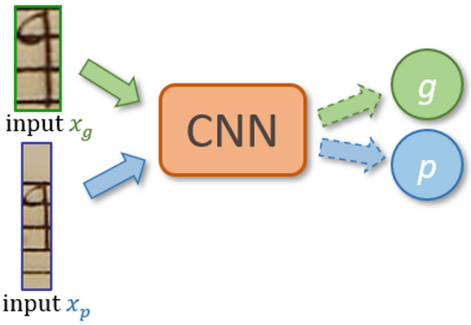

Fig. 8. Multiple inputs and outputs model: both glyph bounding boxes and enlarged images are provided as input and the model must predict both the glyph and the position separately.

3 Experimental Setup

3.1 Network Configuration

Given the proposed general topologies, this section describes the CNN architectures for each model. We have selected a base architecture by means of informal testing. Since this is designed to be used in an interactive scenario, the network must be light to allow for real-time processing.

The input to a convolutional layer is an $m \times n \times r$ image, where m is the height, n is the width of the image and r is the number of channels, being in this case $r = 3$ since we are working with RGB images. Models with one input share the same feature extractor and classification stage. However, they have different inputs and outputs. Their architecture consists of the repeated application of two 3×3 convolutions with 32 filters, each one followed by a rectified linear unit and a 2×2 max-pooling operation with stride 2 for downsampling, then followed by a dropout of 0.25. At the next stack of convolutional layers and max-pooling, we double the number of filters. Finally, after flattening, a fully-connected layer with 256 units followed by a dropout of 0.25 and a *softmax* activation function layer is used for the classification stage.

Models with two inputs have parallel feature extractor for each input, and they are concatenated in their corresponding last stack, after the flattening operation. When using models with two outputs in their classification stages, the network is forked into two different outputs after the last fully-connected layer.

3.2 Dataset

For our experiments, three different manuscripts of handwritten music scores in Mensural notation were available with symbol-level annotation. A summary of the number of samples, and glyph and position classes is shown in Table 1. Combined classes are the result of the Cartesian product of glyph and position classes, as stated above. It is worth noting that most combinations of glyph and position do not appear in our ground-truth, so they have been removed from the set of combined classes.

A major drawback of this ground-truth dataset is the high ratio of label imbalance. Therefore, we took this into account while training our models by weighting the loss for each label according to its relative frequency in the dataset.

Furthermore, since the symbol images can vary drastically in size, and for the sake of creating a dataset suitable for training CNN models, the input images are resized to a fixed size. We resized images to 40×40 pixels for glyph inputs, and to 40×112 pixels for enlarged inputs.

Table 1. Distribution of classes and samples in the Mensural notation symbols dataset.

	Quantity
Glyph classes	37
Position classes	14
Category (combined) classes	157
Total symbols	14373

3.3 Evaluation

In order to evaluate our proposed models, we conducted a 5-fold cross-validation scheme for the six models that were considered. In each fold, 80% of the available data is used for training and 20% for validation. Also, each fold was created by the class imbalance of the original dataset.

Moreover, a grid search was carried out in order to tune hyperparameters. Consequently, we trained every architecture for 15 epochs and a batch size of 32. Additionally, a *RMSprop* optimizer was used [12].

Given that our purpose is to evaluate the correct recognition of music categories, we should consider our models' accuracy taking into account that the glyph and position of a given input are being labeled correctly at once. More precisely, given an input series $x = \{x_1, x_2, \ldots, x_n\}$ we need to calculate the accuracy of the three different outputs of a symbol: position as $p = \{p_1, p_2, \ldots, p_n\}$, glyph as $g = \{g_1, g_2, \ldots, g_n\}$ and category as $c = \{c_1, c_2, \ldots, c_n\}$, for every model.

Another important factor to consider is the complexity of each model since we aforementioned the necessity of good effectiveness and efficiency in the application of these models in a real scenario. In order to provide a value of efficiency that does not depend on the underlying hardware used in the experiments, we consider the number of (trainable) parameters of the neural model as a measure of its complexity.

4 Results

Table 2 presents average and standard deviations achieved by the different classification schemes for the five folds, as well as the complexity (number of trainable parameters) of each model.

The best results in terms of accuracy are obtained by the model with multiple inputs and outputs, then without almost no significant difference, by the model with multiple outputs, followed by the model with independent glyph and position. Moreover, the worst results are obtained by those that consider the full category as output, especially when using multiple inputs.

Another factor for evaluation is the correlation between complexity and accuracy. The models with multiple inputs share the same complexity as well as models with only one input. The best model, with multiple inputs and outputs,

Table 2. Error rate (average ± std. deviation) and complexity with respect to the neural architecture considered for music symbol classification. The complexity of each model is measured as millions of trainable parameters.

Model	Error rate (%)			Complexity (10^6)
	Glyph	Position	Category	
Independent glyph and position	3.0 ± 0.4	5.1 ± 0.6	7.6 ± 0.6	$1.71 + 4.66$
Category output	4.1 ± 0.1	5.9 ± 0.4	8.2 ± 0.3	4.66
Category output, multiple inputs	6.3 ± 0.6	7.5 ± 0.7	11.8 ± 0.9	6.37
Multiple outputs	3.1 ± 0.1	5.2 ± 0.3	7.6 ± 0.3	4.66
Multiple inputs and outputs	2.9 ± 0.1	4.7 ± 0.4	7.1 ± 0.4	6.37

has high complexity compared to the second best one, which is relevant, as both perform comparably well.

Contrary to expectation, it is interesting to point out that models with multiple outputs are performing better. Since glyph and position are dependent features for most instances, we could expect that models predicting combined classes would produce better results. However, we are aware that this outcome might be produced by the possibility of not having enough data to train.

5 Conclusions

In this work, we proposed different segmentation-based approaches to recognize glyph and position of handwritten symbols, since traditional OMR systems mostly focuses on recognizing the glyph of music symbols, but not their position with respect to a staff. Therefore, we propose to classify glyph and position in the same step. Our approach is based on using different CNN architectures where we predict glyph, position, or their combination by training independent models, multi-input and multi-output models.

Experimentation was presented by using a dataset of handwritten music scores in Mensural notation where we evaluated each model by their accuracy on labeling glyph, position, and their combination, as well as their complexity in order to estimate the best model for our purpose.

The results suggest that in order to obtain the best accuracy, models should predict glyph and position labels separately, instead of predicting the combination of both as a single class. We can conclude that interesting insight has been gained with regard to achieving a complete system for extracting the musical content from an image of a score.

As a future work, we plan to consider the use of data augmentation for boosting the performance. However, this has to be designed carefully as traditional data augmentation procedures might not work correctly as regards the positions of the symbols. We also plan to reuse the outcomes of this paper for segmentation-free recognition, which does not need a previous localization of the symbols in the image [10].

As an another step towards the goal of this research, these models must be integrated on a fully-automated transcription system that uses semantic analysis tools to assign actual musical meaning to the output of the considered models.

Acknowledgments. This work is supported by the Spanish Ministry HISPAMUS project TIN2017-86576-R, partially funded by the EU.

References

1. Bainbridge, D., Bell, T.: The challenge of optical music recognition. Comput. Humanit. **35**(2), 95–121 (2001)
2. Calvo-Zaragoza, J., Gallego, A.J., Pertusa, A.: Recognition of handwritten music symbols with convolutional neural codes. In: 14th International Conference on Document Analysis and Recognition, Kyoto, Japan, pp. 691–696 (2017)
3. Good, M., et al.: MusicXML: an internet-friendly format for sheet music. In: XML Conference and Expo, pp. 03–04 (2001)
4. Huang, Y.H., Chen, X., Beck, S., Burn, D., Van Gool, L.: Automatic handwritten mensural notation interpreter: from manuscript to MIDI performance. In: Müller, M., Wiering, F. (eds.) 16th International Society for Music Information Retrieval Conference, Málaga, Spain, pp. 79–85 (2015)
5. Pacha, A., Calvo-Zaragoza, J.: Optical music recognition in mensural notation with region-based convolutional neural networks. In: 19th International Society for Music Information Retrieval Conference, Paris, France, pp. 240–247 (2018)
6. Pacha, A., Eidenberger, H.: Towards a universal music symbol classifier. In: 14th International Conference on Document Analysis and Recognition, IAPR TC10 (Technical Committee on Graphics Recognition), pp. 35–36. IEEE Computer Society, Kyoto (2017)
7. Pacha, A., Hajič Jr., J., Calvo-Zaragoza, J.: A baseline for general music object detection with deep learning. Appl. Sci. **8**(9), 1488–1508 (2018)
8. Ramirez, C., Ohya, J.: Automatic recognition of square notation symbols in western plainchant manuscripts. J. New Music Res. **43**(4), 390–399 (2014)
9. Rebelo, A., Capela, G., Cardoso, J.D.S.: Optical recognition of music symbols. Int. J. Doc. Anal. Recogn. **13**(1), 19–31 (2010)
10. Rizo, D., Calvo-Zaragoza, J., Iñesta, J.M.: MuRET: a music recognition, encoding, and transcription tool. In: 5th International Conference on Digital Libraries for Musicology, pp. 52–56. ACM, Paris (2018)
11. Roland, P.: The music encoding initiative (MEI). In: 1st International Conference on Musical Applications Using XML, pp. 55–59 (2002)
12. Ruder, S.: An overview of gradient descent optimization algorithms. Computer Research Repository abs/1609.04747 (2016)
13. Vigliensoni, G., Burgoyne, J.A., Hankinson, A., Fujinaga, I.: Automatic pitch detection in printed square notation. In: Klapuri, A., Leider, C. (eds.) 12th International Society for Music Information Retrieval Conference, pp. 423–428. University of Miami, Miami (2011)
14. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53