# Music Symbols Segmentation through Recognition

Ana Rebelo[1]
arebelo@inescporto.pt
Cuihong Wen[2]
cuihongwen2006@gmail.com
Jaime S. Cardoso[1]
jaime.cardoso@inescporto.pt

[1] INESC TEC and
Faculdade de Engenharia
Universidade do Porto, Portugal
[2] Hunan University
Changsha, Hunan, China

## Abstract

Optical music recognition (OMR) systems have been under intensive development for many years in order to create a robust process for printed and handwritten music scores. In this paper, a method to extract the music symbols from a music sheet without segmentation is presented. The aim is to execute simultaneously the segmentation and recognition of the objects avoiding the issues inherent to the segmentation phase. A Combined Neural Network (CNN) framework is also proposed to classify the music symbols.

## 1 Introduction

The musical symbols detection is a stage on an OMR system where operations to localize and to isolate musical objects are applied. In [4] a process to segment the objects based on a hierarchical decomposition of a music image and in contextual information and music writing rules was proposed. First the image was segmented in order to detect and isolate the primitive elements and then the symbols were classified. In this work, a new method is presented. The idea is to perform segmentation through recognition, that is, the method simultaneously segment and recognize the image. The principal advantage regarding to the previous method is in the elimination of the multiple heuristics used in the first case. As symbols are first detected and extracted from the image and, after that, classified, various parameters related to the size, shape and position of the objects are introduced. These parameters can constitute a severe problem in handwritten music scores, because the variability in writing style of each composer. Segmenting the music sheet using classification simplifies all the process and also overcomes the issues inherent in sequential detection of the objects, leading to less errors.

## 2 Detection Process

The recognition process consists first in the splitting by staffs the music sheet with staff lines previously removed [2], and then analyzing each of these segments. For each staff the connected component technique is applied. This technique has a threshold in order to join neighboring pixels from broken objects. The algorithm proceeds with a scanning of each connected component in order to recognize what is an object and what is not. This analysis is carried out using classifiers and it is an hierarchical process. On the first level of this hierarchy, the detected objects are split into *noise* and *symbol*, and then, on the second level, the *noise* objects are divided into four types (see Figure 1): (1) *connected symbol*, (2) *not symbol*, (3) *split symbol* and (4) *connected and split symbol*. The objects that are classified as *symbol* can be one of the 19 possible symbols presented in Table 1.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Accent | BassClef | Beam | Flat | Natural | Note | NoteFlag | NoteOpen |
| RestI | RestII | Sharp | TrebleClef | AltoClef | TimeN | TimeL | Barlines |
| Dot | Breve | Semibreve | | | | | |

Table 1: The set of the musical symbols considered in the *symbol* class.

On the third level of the hierarchical classification process, the algorithm scans again each connected component that is one of the four types of *noise* objects. These objects, presented in Table 2, can be classified in one of the 14 possible symbols to find. Note that in this step we are trying to split objects, usually notes connected to beams, so the *waste* class is necessary.

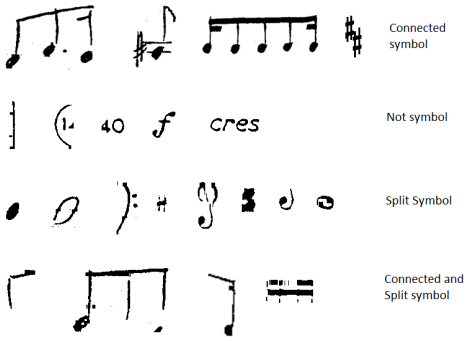


Figure 1: Examples of objects that are considered noise.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Accent | Clef | Beam | Accidentals | Barlines | Note | NoteFlag | NoteOpen |
| RestI | RestII | Semibreve | TimeN | TimeL | Examples of Waste class | | |

Table 2: The set of the musical symbols considered in the *waste&symbol* class.

As mentioned in the beginning of this section, the analysis executed during the procedure to select the musical symbols relies on classifiers. We built four types of classifiers relating them to each possible situation that can appear in the connected components. The CNN_1 classifier[1] is performed in order to divide the objects detected as *noise* or *symbol*. If the detected object is *symbol* then the CNN_3 is used, otherwise the CNN_2 is utilized. For *connected symbol*, *split symbol* and *connected and split symbol* an analyse to each bounding box of the object is carried out using the CNN_4. The construction of these classifiers will be explained later.

**Connected symbol**

*Connected symbol* class encompasses notes connected to beams, notes connected to accidentals and notes connected to accents. In the first situation accidentals and accents can also appear in the bounding box. We proposed a sliding window procedure supported by the CNN_4 to detect and extract the symbols. First, the analysis window with an height equal to the height of the bounding box is moved along the columns. The window width starts equal to `staffspaceheight`[2] and is scaled three times. The choice of using this value was obtained experimentally. Only the notes class is considered on this step . After this the search of the window is changed to go by rows. The aim is to look for accidentals and accents. Again the CNN_4 classifier is used to detect and extract the symbols. The procedure to establish the window size is the same of the previous step. Since an overlap exists between windows, there are repeated objects that need to be removed. Hence, a process to group symbols is executed. The symbols from the same class are compared with each other; if their positions are close enough, they are saved as one symbol. All the symbols detected are removed from the image. Now it is necessary to detect beams

[1] The Combined Neural Network (CNN) will be explained on the next section.
[2] `staffspaceheight` represents the distance between two consecutive stafflines. For more details please see [1].

which are music symbols linking two notes. So, for each image composed by two adjacent notes the algorithm looks for black pixels. It is worth restating that notes were already removed from the image.

**Split symbol**

*Split symbol* class encompasses broken objects. Usually they are notes separated from their stems or fragmented accidentals and clefs. The goal is to join black pixels near to the initial object. For that the window size increases until a certain limit and the CNN_4 recognizer is used to see when we are in presence of a music symbol. The augmentation of the window is first done in height, then in width and then in both. At the end the procedure to look for repeated symbols is again computed.

**Connected and split symbol**

*Connected and split symbol* class encompasses the two previous groups of symbols. For that reason, the techniques already described for each of the classes are applied here.

After the detection of all symbols a process to test some musical rules is executed. In here, the presence of accents only above notes and the position of accidentals before and at same height of notes is verified. If the symbols do not respect these rules then they are eliminated.

## 3 Combined Neural Network

To perform the various necessary classifications during the scanning procedure, we propose a music symbol recognizer based on a majority vote combination of three Multi-Layer Perception (MLP) classifiers named Combined Neural Networks (CNN). Two of the networks have the same architecture, but the initial random weights are different. The third network is fed with a different input and with a different number of neurons in the hidden layer. All these networks have a log-sigmoid activation function. In this way we expect to increase the overall performance of the classifier regarding to the usual way of only one MLP.

For two of the networks each image of a symbol was initially resized to $20 \times 20$ pixels and then converted to a vector of 400 binary values. For the third network each image of a symbol was initially resized to $60 \times 20$ pixels and then converted to a vector of 1200 binary values. Usually the images have an height larger than their width and the idea was thus to favor the height. In this manner, the problem in the classification of barlines, due to its similarity with dots after the resize, is minimized.

A database of training patterns was created according to the possible objects that algorithm could find in the scanning process. Also because of that, for each CNN we have a different database. Each one of these databases was randomly split into training and test sets, with 60% and 40% of the data, respectively. This division was repeated 10 times without restricting the distribution of the categories of symbols over the training and test sets. Only two constraints were imposed: at least one example of each category should be presented in the training set and the size of the noise class should be limited to avoid very unbalanced classes distributions. The best parametrization of each model was found using the training and validation sets, with the expected error estimated on the test set by a 4-cross validation scheme. The results for the different models can be seen in Table 3.

| | Noise/Symbol | Noise | Symbol | Waste&Symbol |
|---|---|---|---|---|
| MLP | [91; 92] | [82; 84] | [88; 89] | [81; 84] |
| CNN | [95; 96] | [90; 91] | [95; 96] | [88; 89] |

Table 3: 99% CI for the expected performance (in percentage) for the classification models.

If we compare with our initial results, with one MLP applied to the same datasets of music scores and also divided in the same way, we obtained an higher accuracy, as expected.

## 4 Experimental Testing and Conclusion

The data set adopted to test the proposed architectures for the music symbols extraction consists of both handwritten and synthetic scores. In total we have 9 scanned printed scores, 26 handwritten scores and 882 images generated from 18 synthetic scores (available from [3]). The metrics accuracy rate, average precision and recall were considered. They are given by

$$accuracy = \frac{\#tp+\#tn}{\#tp+\#fp+\#fn+\#tn}, \quad precision = \frac{\#tp}{\#tp+\#fp}, \quad recall = \frac{\#tp}{\#tp+\#fn}$$

The true positive rate (TPR), false positive rate (FPR), true negative rate (TNR) and false negative rate (FNR) were also considered:

$$TPR = \frac{\#tp}{\#tp+\#fn}, \quad FPR = \frac{\#fp}{\#tn+\#fp}, \quad TNR = \frac{\#tn}{\#tn+\#fp}, \quad FNR = \frac{\#fn}{\#fn+\#tp}$$

where $tp$ are the true positives, $tn$ are the true negatives, $fn$ are the false negatives, $fp$ are the false positives and $tpc$ are the classes of the true positives. A false negative happens when the algorithm identifies a musical symbol as noise; and a false positive is when the algorithm identifies noise as a music symbol. These percentages are computed using the symbols position reference and the symbols position obtained by the segmentation algorithm.

| | Precision | Recall | Accuracy |
|---|---|---|---|
| Handwritten scores | 61.0% | 91.6% | 97.6% |
| Printed scores | 53.9% | 93.0% | 79.4% |
| Digitized scores | 67.9% | 93.8% | 97.8% |

Table 4: Results for music symbols extraction.

| Handwritten Scores | | | Scanned Scores | | | Printed Scores | | |
|---|---|---|---|---|---|---|---|---|
| | **True** | **False** | | **True** | **False** | | **True** | **False** |
| **Positive** | 91.7% | 2.1% | **Positive** | 93.8% | 2.7% | **Positive** | 93.0% | 18.9% |
| **Negative** | 99.9% | 20.5% | **Negative** | 99.9% | 30.5% | **Negative** | 94.1% | 9.0% |

Table 5: Confusion Matrix for the results from the Table 4.

| | Precision | Recall | Accuracy |
|---|---|---|---|
| Handwritten scores | 67.1% | 78.8% | 97.2% |
| Printed scores | 69.1% | 79.9% | 82.2% |
| Digitized scores | 66.1% | 87.3% | 96.7% |

Table 6: Results for music symbols extraction through recognition.

| Handwritten Scores | | | Scanned Scores | | | Printed Scores | | |
|---|---|---|---|---|---|---|---|---|
| | **True** | **False** | | **True** | **False** | | **True** | **False** |
| **Positive** | 78.8% | 2.0% | **Positive** | 87.3% | 2.8% | **Positive** | 79.9% | 7.6% |
| **Negative** | 99.9% | 41.8% | **Negative** | 99.9% | 44.4% | **Negative** | 90.5% | 33.1% |

Table 7: Confusion Matrix for the results from the Table 6.

Looking to the results, we can conclude that the algorithm to extract symbols through recognition detects more false negatives and less true positives than just using the algorithm for symbol extraction. This means that symbol extraction shows a reduction on the correct prediction of the symbols for the three datasets. This rationale is clearly depicted on the results shown in Table 6 where the recall substantially decreased over the recall results shown in Table 4. However, the first process has more missed symbols. Furthermore, comparing the Tables 4 and 6 the symbols extraction through recognition improves the performance in printed scores (82.2%). As future work, other experiments in the neural network could be addressed in order to improve the outcome: change the size of the images, the number of neurons, the number of hidden layers and the activation function. Moreover, a syntactic consistency method should be included as a final stage of the process to overcome possible mistakes in the classification.

## Acknowledgments

This work is financed by the ERDF-European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT-Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within projects FCOMP-01-0124-FEDER-022701, SFRH/BD/60359/2009 and PTDC/SAU-ENB/114951/2009.

## References

[1] J. S. Cardoso and A. Rebelo. Robust staffline thickness and distance estimation in binary and gray-level music scores. In *Proceedings of The Twentieth International Conference on Pattern Recognition (ICPR 2010)*, pages 1856–1859., 2010.

[2] J. S. Cardoso, A. Capela, A. Rebelo, C. Guedes, and J. F. Pinto da Costa. Staff detection with stable paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31 (6):1134–1139, 2009. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/TPAMI.2009.34.

[3] C. Dalitz, M. Droettboom, B. Czerwinski, and I. Fujigana. A comparative study of staff removal algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:753–766, 2008.

[4] A. Rebelo, F. Paszkiewicz, C. Guedes, A. Marcal, and J. S. Cardoso. A method for music symbols extraction based on musical rules. In *Bridges: Mathematical Connections in Art, Music, and Science*, pages 81–88, 2011.