Camera-based Optical Music Recognition using a Convolutional Neural Network

Adrià Rico, Alicia Fornés
Computer Vision Center - Computer Science Department
Universitat Autònoma de Barcelona
Bellaterra, Catalonia, Spain
adria.rico@e-campus.uab.cat, afornes@cvc.uab.cat

***Abstract*—Optical Music Recognition (OMR) consists in recognizing images of music scores. Contrary to expectation, the current OMR systems usually fail when recognizing images of scores captured by digital cameras and smartphones. In this work, we propose a camera-based OMR system based on Convolutional Neural Networks, showing promising preliminary results.**

***Index Terms*—optical music recognition; document analysis; convolutional neural network; deep learning**

I. Introduction

Optical Music Recognition (OMR) is the task devoted to convert an image of a music score into a machine-readable format, such as MIDI, MEI or MusicXML. After decades of research, the recognition of scanned printed music scores can be considered to be in a mature state [3], [4]. However, the recognition of camera-based music scores still needs more research. The current OMR methods designed for scanned scores usually fail in such scenarios due to perspective distortions, changes in illumination, etc. For this reason, some researchers have proposed specific methods for binarization [5], staff removal [1] or lyrics extraction [2] in camera-based music scores. However, as far as we know, the few complete camera-based OMR systems, such as the one in [6], are based on basic techniques (e.g. run lengths, projections, template matching), showing a moderate performance. Given the fact that deep learning architectures have lately shown impressive performance in many computer vision tasks, we aim to explore their use for camera-based OMR. Concretely, after the preprocessing, staff removal and symbol segmentation steps, we use a Convolutional Neural Network (CNN) to classify each music symbol. For training the network, we have generated synthetic music scores.

The rest of the paper is organized as follows. Section II describes the preprocessing steps, Section III describes the symbol segmentation process and Section IV describes the CNN architecture and the generation of synthetic music scores.

II. Preprocessing

As usual, the first step is to preprocess the image in order to make the subsequent analysis easier. In this case we are dealing with photos taken with a camera, which often present some types of distortions, such as light level variation and perspective deformations. Below we list all the preprocessing procedures that are applied to the input image.

A. Binarization

A simple mean-based adaptive threshold is applied to the image to separate the background from the foreground. Experimentally, we determined that a high number of neighbours produces cleaner results (in fact, clean enough that no noise removal needs to be made afterwards).

B. Perspective correction

A homography is applied to correct the perspective. The challenge is to find a reliable set of points which can be used to compute the transformation matrix. We determined that a good option is finding the corners of the staves, as they usually form a quadrilateral and will always be present on any image. To perform this task, we first apply a Hough Line transform, then filter the lines to select only those which are parallel and equally distant. Each group of five equally distant lines is assumed to be part of a staff. Finally, we select one of the groups given an heuristic function, determine the segments which from part of the upper and lower staff lines, and consider their starting and end points as the corners of the staff.

C. Staff location detection

Determining where each staff is in the image is important, because then we won't have to deal with any drawings, text, etc. on subsequent steps. To accomplish this, we perform connected component analysis and get the bounding box of each component. For a component to be considered a staff candidate, it needs have a width of at least half of the total image width, and needs to have an aspect ratio of 5:1 or more (values chosen experimentally). All the staff candidates are fed into the staff detection algorithm, which will be explained in the next section. Additionally, we look for connected components vertically close to the staff candidates and assume they also form part of the staff's symbols (e.g. tuplet indicators, ties, etc. which are generally disconnected from the staff lines but are very close to musical notes).

III. Staff removal and note segmentation

After the preprocessing procedure, we have all the staves located, binarized and with most of the distortion caused by perspective corrected. The next step is to segment all the notes and symbols to be able to feed them to the neural network.

A. Staff removal

To ease pitch detection and the segmentation of musical symbols, the staff lines are detected and removed using already existing techniques, taking into account that the lines might not be completely horizontal. Also during this phase, the false staves introduced in the last preprocessing step are removed by comparing the estimated staff line height and staff space height of all the staves and looking for outliers.

B. Detection of musical symbols

First, we perform connected component analysis, and find the bounding box of each component. Then, we recursively join the boxes which are very close to or inside another. This is made because some symbols could become disconnected in some of the previous steps.

C. Segmentation of note heads

Our approach is to feed individual notes and symbols to the neural network to recognize them separately, and then reconstruct the music using that information. Because of that, notes connected by a beam need to be separated. To accomplish this, all black note heads are detected and, in case two or more lie in the same box (from the previous step), the box is split horizontally equally dividing all note heads. For black note head detection, we apply the *opening* morphological operation to the image with an elliptic kernel, find all connected components, and determine which of them are note heads given some properties (size, aspect ratio, area and being close to a stem).

IV. Symbol recognition

Once every symbol has been segmented, the recognition stage begins. First of all, the bars are detected given their very distinct properties (very small width and very small aspect ratio), the end bar is also detected separately. As for the rest of the symbols, they are sent to the neural network for recognition. In the following subsections, we'll explain the architecture of the neural network and how we trained it.

A. Neural network architecture

We used a Convolutional Neural Network, given the success they have had in many graphics recognition tasks. We've chosen a simple, generic architecture, because the classification task isn't very demanding. It might change in the future after more testing has been done. Currently, only 10 classes are considered: notes and rests from whole to a sixteenth. In the near future, more symbols such as accidentals, clefs and time signatures will be added. A scheme of the network architecture is shown in Figure 1. The network has been trained with 4487 samples, generated as explained in the next section.

B. Ground truth generation

We have found no datasets of labelled, segmented printed music symbols. To avoid having to label thousands of images by hand, we generated random note sequences and fed them to the open source program *LilyPond* to obtain the corresponding image of the music score. We then applied the segmentation procedure previously described to the image, and labelled each segment according to their position in accordance with the generated sequence. Although *LilyPond* allows for the scores to be generated without a staff, we performed staff removal on the output images instead to simulate the artifacts that the network will find when we feed it the real segments. Different symbol and staff thicknesses have also been used.

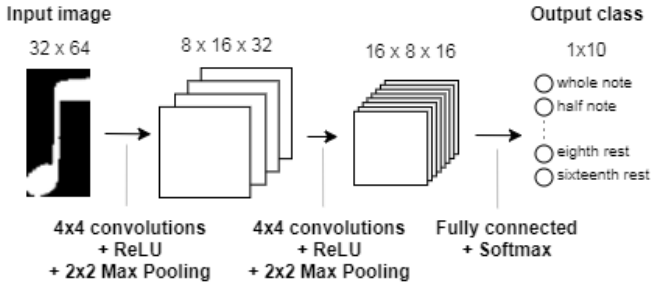

Fig. 1. Scheme of the architecture of the CNN we used. All convolutions are done with a padding of 2 to preserve the image size.

V. Conclusions

In this work, we have explored the use of Deep Learning architectures for camera-based OMR. The preliminary qualitative results are promising. The CNN consistently recognizes eighteenth and sixteenth notes, but confuses quarter and half notes. Further work will be focused on the improvement of the recognition and the addition of the remaining music symbols (such as accidentals, clefs, key signatures, etc.).

Acknowledgment

This work has been partially supported by the Spanish project TIN2015-70924-C2-2-R, the Ramon y Cajal Fellowship RYC-2014-16831 and the CERCA Program/Generalitat de Catalunya.

References

[1] Hoang Nam Bui, In Seop Na, and Soo Hyung Kim. Staff line removal using line adjacency graph and staff line skeleton for camera-based printed music scores. *Proceedings - International Conference on Pattern Recognition*, pages 2787–2789, 2014.

[2] Cong Minh Dinh, Hyung Jeong Yang, Guee Sang Lee, and Soo Hyung Kim. Fast lyric area extraction from images of printed Korean music scores. *IEICE Transactions on Information and Systems*, E99D(6):1576–1584, 2016.

[3] Alicia Fornés and Gemma Sánchez. Analysis and recognition of music scores. In *Handbook of Document Image Processing and Recognition*, pages 749–774. Springer-Verlag London, 2014.

[4] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, André R. S. Marçal, Carlos Guedes, and Jaime S. Cardoso. Optical music recognition: State-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 1(3):173–190, 2012.

[5] Quang Nhat Vo, Soo Hyung Kim, Hyung Jeong Yang, and Gueesang Lee. An MRF model for binarization of music scores with complex background. *Pattern Recognition Letters*, 69:88–95, 2016.

[6] Quang Nhat Vo, Tam Nguyen, Soo Hyung Kim, Hyung Jeong Yang, and Guee Sang Lee. Distorted music score recognition without Staffline removal. *Proceedings - International Conference on Pattern Recognition*, pages 2956–2960, 2014.