Bootstrapping Samples of Accidentals in Dense Piano Scores for CNN-Based Detection

Kwon-Young Choi
IRISA - INSA
Rennes, France
kwon-young.choi@irisa.fr

Bertrand Coüasnon
IRISA - INSA
Rennes, France
bertrand.couasnon@irisa.fr

Yann Ricquebourg
IRISA - INSA
Rennes, France
yann.ricquebourg@irisa.fr

Richard Zanibbi
RIT
Rochester, USA
rlaz@cs.rit.edu

***Abstract*—State-of-the-art Optical Music Recognition system often fails to process dense and damaged music scores, where many symbols can present complex segmentation problems. We propose to resolve these segmentation problems by using a CNN-based detector trained with few manually annotated data. A data augmentation bootstrapping method is used to accurately train a deep learning model to do the localization and classification of an accidental symbol associated with a note head, or the note head if there is no accidental. Using 5-fold cross-validation, we obtain an average of 98.5% localization with an IoU score over 0.5 and a classification accuracy of 99.2%.**

a) Introduction: Optical Music Recognition (OMR) systems produce remarkable results on relatively simple and clean images of printed scores. However, when trying to recognize very dense and noisy music scores, these systems fail because the segmentation task is difficult due to touching and broken symbols by printing techniques (see Figure 1). Supervised learning system need annotated training data which doesn't exist at the moment in the OMR domain. We propose to train a Convolutional Neural Network (CNN) based detector to localize and classify three accidental symbols associated with a note head, or the note head if there is no accidental, in complex and damaged piano scores. We manually labeled 2955 examples with 1987 symbols, 968 rejects and developed a randomized bootstrapping technique to artificially augment our training data by 10 to 100 times. We take advantage of the fact that position of the note head is known, and use the centroid position of the note head as a feature.

b) Accidental Localization: OMR systems break down the process of recognizing a music scores into multiple steps [1]. First, preprocessing techniques, like binarization, are used to prepare the image. Then one important step is to accurately detect stafflines, which is a core component of the score. Most OMR systems remove the stafflines from the image in order to do a first segmentation of connected components. In this work, we chose to use the technique described by [2] to detect and remove stafflines. The next steps are to segment and recognize all music symbols or primitive such as blobs, flags or segments in the image. Various techniques, including projections, run-length analysis, contour-line tracking, graphs or template matching, have been used for segmentation and K-NN, SVM or neural networks can be used for classifiers[1]. However, all of these techniques require complex logic to correctly segment music symbols or primitives[3]. We propose to delegate the segmentation and classification of music symbols to a CNN based detector.

Convolutional neural network extract visual cues in early layers from a raw image and then gradually construct a more complex representation of the image. These information are then used to perform a wide range of tasks, but the most common ones are classification or object detection. The authors of [4] proposed a CNN architecture capable of visual attention. A first convolutional localization network can frame a region of interest that is then cropped by a special Spatial Transformer Layer (STL). The cropped region is then classified using a second convolutional network. We saw in this architecture the opportunity to build a simple symbol detector.

c) Architecture: Our architecture is similar to the one described by [4]. However, we explicitly extracted localization information from the geometrical transformation matrix produced by the localization network. In order to do detection of accidental symbols, we actually use two networks in parallel. One network for localization and another network for localization and classification. We modified the input of the localization network by adding the known position of a note head. Indeed, we know that the position of an accidental is strongly correlated to the position of the associated note head. That is why we chose to use two additional input neuron to encode the centroid position of the note head and we connected these neurons directly to the first dense layer of the localization network. We measure localization performance by using the Intersection over Union (IoU) score between the predicted bounding box of the localization network and the ground-truth bounding box. A true positive detection is when we obtain a matching accidental class information and an IoU score over 0.5. Every localization with an IoU under 0.5 is considered as a false positive as commonly done in object detection literature. In the special case where there is no accidental, the localization information is not important and a true positive definition is based only on the class information.

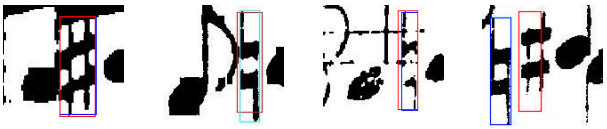
d) Bootstrapping Small Dataset: The background motivation for this work is to build a detector without pre-existing dataset, allowing the re-utilization of the method for many types of documents. Therefore, we constructed our dataset by using 5 music scores from imslp.org providing us with 70 pages of music scores. From these 70 pages, we extracted 2955 examples using a CNN music symbol classifier and

a posterior manual verification. The dataset contains 1987 accidental symbols unevenly distributed between three classes: flat, sharp and natural, and the rest is considered as rejection. For each of the 2955 examples, we extracted a squared image patch normalized from the height of the interline, which is the vertical space between two consecutive staff lines. Finally, we resized all images to the same size of 80x80 pixels (examples are given in Figure 1).

Because we know the position of the note head associated to the accidental, we anchor the position of our cropping box to the position of the note head, where the middle right border is positioned on the center of the note head. Using this placement, we generated a test set of one image for each element of the dataset and we will refer to this method as the *original* method. Because the localization of an accidental is defined by the position of the note head, we noticed that the position of accidentals are heavily concentrated on the right part of the image. This introduces a localization bias that we will try to remove using bootstrapping (oversampling) techniques. For the localization task, what we want to do when augmenting our dataset is to introduce more variability and balance in the training data in order to push our neural network to learn better features.

We chose to experiment four kinds of random sampling methods. Our first random sampling method technique is to take a bounding box with a size of four time the height of an interline and to randomly move it around the accidental (or note head in case of rejection). We names this technique *unconstrained*. We also noticed that the vertical position of the accidental is very stable in relation to the vertical position of the note head. By simplifying the localization task, we hypothesize that the localization performance will improve. That is why in our second method *vertical*, we tried to reduce the possibility of displacement by specifying a maximal vertical distance of 10 pixels between the center of the cropping box and the center of the note head. Because the note head is a strong visual clue in order to detect an accidental, we decided to use two additional techniques *note head* and *vertical note head* that will always keep at least half of the note head inside the image patch. We tested different amounts of data generated: 25,000, 50,000, 100,000, 200,000, 300,000 and 400,000 images, in order to observe the behaviour of our network on different quantity of data.

e) Experimental Results: The training was done using the Adam backpropagation algorithm with a learning rate of 0.0001. The loss function of the localization network we chose was the mean squared error function. The classification network loss function was the mean categorical cross-entropy function. Because the dataset used is small (2995 elements) and in order to observe the variance in performance, we chose to do five fold (599 elements each) cross-validation. The dataset was split before applying bootstrapping techniques. During the training phase, we used our bootstrapping techniques to augment the size of each fold independently. Bootstrapped samples for a fold were ignored when that fold becomes the test fold. This produced results measured on the original 2995 samples after running all five folds. Therefore, samples generated for training folds remain with the fold, and are never a part of the test folds. Early stopping was used to avoid overfitting and ended the training as soon as the loss stopped decreasing for at least 10 epochs.



(a) Correct (Touching) (b) Correct (Touching) (c) Correct (Broken) (d) Incorrect (Confusion)

Fig. 1. Correct and incorrect detection of accidental using in parallel the best classification and localization models. Red boxes are predictions and blue boxes are ground-truth.

We explored variations of three variables: the presence of the note head position as input feature, the bootstrap method and the quantity of data generated by the bootstrap method. We found that the best configuration was by using the note head as input feature, the *vertical* bootstrap method by generating 200k images and obtained an average localization precision of 98.5% with a standard deviation of 0.8% and an average classification accuracy of 99.2% with a standard deviation of 0.5%. Adding the position of the note head improved the mean localization recall by 2.1% for accidental classes and 25.9% for the rejection class. Augmenting the quantity of data gradually improves the results, however, when over 200k data are generated, we observe overfitting for less represented classes. For the classification network, we found that fine-tuning the already learned weights for the localization network improved the results by 1.6%. As shown in Figure 1, we can see our model succeed on detecting symbols presenting segmentation problems with some errors when faced with multiple symbols.

f) Conclusion: We designed a method that produces an accurate symbol detector of accidentals, without a priori rules concerning segmentation problems. We used the note head position as input feature and produced enough data using bootstrapping methods to do an accurate training of the model with a minimum of manual annotation. Although we only experimented on accidental symbols, we are planning to apply this same method to different kind of symbols like note head, attack signs... Future work will be focused on state-of-the-art object detection methods and evaluate their performances, accuracy and their data quantity requirements for different symbol detection tasks.

REFERENCES

[1] A. Fornés and G. Sánchez, "Analysis and Recognition of Music Scores," in *Handbook of Document Image Processing and Recognition*, D. Doermann and K. Tombre, Eds. Springer London, 2014, pp. 749–774.

[2] V. P. d'Andecy, J. Camillerapp, and I. Leplumey, "Kalman filtering for segment detection: application to music scores analysis," in *Proceedings of 12th International Conference on Pattern Recognition*, vol. 1, Oct. 1994, pp. 301–305 vol.1.

[3] C. Wen, A. Rebelo, J. Zhang, and J. Cardoso, "Classification of optical music symbols based on combined neural network," in *2014 International Conference on Mechatronics and Control (ICMC)*, Jul. 2014, pp. 419–423.

[4] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial Transformer Networks," Jun. 2015.