Apply Lightweight Recognition Algorithms in Optical Music Recognition

Viet-Khoi Pham, Hai-Dang Nguyen, Tung-Anh Nguyen-Khac, and Minh-Triet Tran
Faculty of Information Technology, University of Science, VNU-HCM
{pvkhoi,nhdang12,nktanh}@apcs.vn, tmtriet@fit.hcmus.edu.vn

ABSTRACT

The problems of digitalization and transformation of musical scores into machine-readable format are necessary to be solved since they help people to enjoy music, to learn music, to conserve music sheets, and even to assist music composers. However, the results of existing methods still require improvements for higher accuracy. Therefore, the authors propose lightweight algorithms for Optical Music Recognition to help people to recognize and automatically play musical scores. In our proposal, after removing staff lines and extracting symbols, each music symbol is represented as a grid of identical $M * N$ cells, and the features are extracted and classified with multiple lightweight SVM classifiers. Through experiments, the authors find that the size of $10 * 12$ cells yields the highest precision value. Experimental results on the dataset consisting of 4929 music symbols taken from 18 modern music sheets in the Synthetic Score Database show that our proposed method is able to classify printed musical scores with accuracy up to 99.56%.

Keywords: Optical Music Recognition, Support Vector Machine, Stable Paths approach, lightweight algorithm

1. INTRODUCTION

Music is shared under the form of printed or handwritten documents that people call musical scores. Although a large number of musical scores are the traditional heritage of many countries, they are currently in danger of being lost through time due to not being published and specifically preserved. Moreover, people find it necessary to have a system that can help them to learn music, help music composers in their composing process, assist in various situations such that machine-readable format of music sheet is needed. As a result, this has given birth to the Optical Music Recognition (OMR) field.

OMR has been under intensive research [1]. Although several software applications have been developed (PIANOSCAN, MIDISCAN, PHOTOSCORE…), none of them has given out a satisfactory result [2]. OMR remains a tough problem, especially for handwritten music sheets, due to the variety of musical notations, the way the notations are written in the composers' writing styles, the quality of papers that the music symbols are printed on, etc.

The OMR system's framework is divided into 4 main stages [2]: image preprocessing, recognition of musical symbols, musical notation reconstruction, and final representation construction. We focus on the first 2 stages: image preprocessing and music symbols recognition. In the preprocessing step, we apply Otsu's [3] method to binarize the image, and then we simplify and use the Stable Paths [4] approach to remove staff lines. We also propose a simpler method to reduce the complexity of the symbol segmentation step based on the idea in [4]. In the music symbol recognition stage, we suggest a new way for representing music symbols by using grids of $M * N$ cells and apply multi-lightweight SVM classifiers for classification of this type of features. The authors conduct experiments on dataset consisting of music symbols extracted from 18 modern music sheets in the Synthetic Score Database [5] with the total number of 4929 symbols and achieve the results of accuracy up to 99.56%.

The structure of the paper is organized as follows. In section 2, we present about the background and related works in the OMR field. In section 3, details on methods of each step are presented. Experimental results are shown in section 4 and conclusions are drawn in section 5.

2. BACKGROUND AND RELATED WORKS

The field of OMR first began with Pruslin [6] in 1966, and Prerau [7] in 1970. Since then OMR has been under intensive research for over 4 decades. One of the first objectives of OMR is to help people to preserve musical scores and to help music composers to digitalize music sheets into machine-readable format. As stated by Distributed Digital Music

Seventh International Conference on Machine Vision (ICMV 2014), edited by Antanas Verikas,
Branislav Vuksanovic, Petia Radeva, Jianhong Zhou, Proc. of SPIE Vol. 9445, 944504
 · doi: 10.1117/12.2180715

Archives Library Lab [1], there have been over 300 published papers and theses related to OMR up to the year 2012. Furthermore, various OMR software applications have been developed and published [8]. Many researches have been done in the field, however, especially for handwritten musical scores, none of the existing algorithms and software has given out satisfying results [2].

There are several challenges in OMR [4]. First is that the music symbols' appearances are irregular and decided by the writing styles of composers, including sizes, shapes, and intensity. The second problem is the complexity of a musical score, since there are music sheets that are complicated and hard to read even for human. Other problems such as tilting or curving of staff lines, discontinuities in symbols, noise and stain on papers, low quality of papers are also challenges people need to overcome. As a result, the field of OMR continues to be investigated since the problem of recognizing music notation effectively still remains unsolved.

Staff lines are objects that need to be removed from the score, since it overlaps with other symbols. In 2012, Bolan Su [9] proposes a solution that models the staff line shape. In 2013, Ana Rebelo proposes algorithms for removing staff lines in the grayscale domain [10]. In 2011, the International Conference of Document Analysis and Recognition (ICDAR) organizes a competition for people to present their staff lines removal algorithms. Until today, staff lines removal problem still attract researchers, since a good staff lines removal algorithm provides better results in later stages.

The symbols segmentation and classification step is to extract the symbols, then use a classification algorithm to classify them. Some methods apply Template Matching or machine learning algorithms such as k-nearest neighbors, Hidden Markov Model, neural networks, support vector machines to classify the symbols [4]. Hence, the authors decide to follow the trend to use machine learning algorithms to achieve high results in the classification problem.

3. PROPOSED METHODS

The authors focus on the first 2 stages of a common OMR system's framework [2]: image preprocessing and music symbol recognition (figure 1). The preprocessing step is presented in section 3.1, and the music symbol recognition stage is explained in section 3.2.

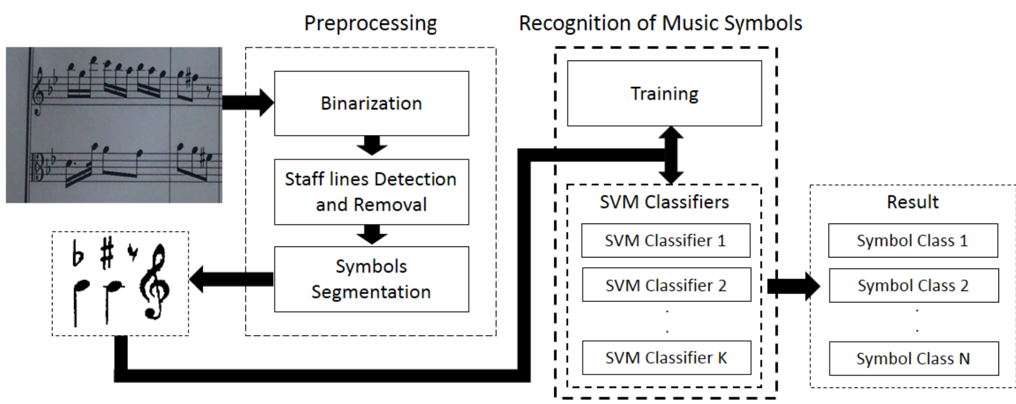

Figure 1. Proposed framework for preprocessing and recognition steps in OMR.

3.1 Preprocessing

3.1.1 Binarization

In this step, the authors use binarization algorithm proposed by Otsu [3] to separate the score into foreground and background. Otsu's method classifies the image into two classes, background and foreground, by iterating a threshold value κ , which is the gray-level of a pixel. Pixel that is lower than the threshold is chosen to be the foreground, otherwise it is the background. The desired value for κ is the value such that it minimizes the weighted sum of variance of the two classes (sum of within-class variance), and divides the image of the score into two classes of pixels.

However, Otsu's algorithm is a global binarization algorithm. If the image contains a part that is darker than the remaining parts in the image, the algorithm will fail. Thus, the authors propose to use other adaptive binarization algorithms to solve this step more effectively.

3.1.2 Staff lines detection and removal

Staff lines are needed to be removed since they overlap with other symbols. However, our main goal is to classify precisely the symbols, so the results of removing staff lines need not be perfect. From this observation, the authors suggest using a simplified method based on the Stable Paths approach [4] because we do not necessarily remove and reconstruct all staff lines with high accuracy. We only use it as the preprocessing step for the main problem, symbols recognition.

The idea of the approach is to build a graph for the musical score. Each pixel in the score is a vertex, and there exists an edge connecting 2 vertices if the 2 vertices are 2 adjacent pixels. The weight function is as follows: each edge is initially given with weight equal to 2, then it is incremented by 1 if the 2 pixels are diagonally adjacent. We then increase the edge's weight by a value equal to the number of white pixels in the 2 pixels. After the graph is built, the staff lines can be detected by looking for the shortest paths that start from the first column and end in the last column of the image.

Let $pixel_{i,j}$ be the pixel that is on column i and row j of the image, $F_{i,j}$ be the cost of the shortest path that starts from an arbitrary pixel in column 1 in the image and stops at $pixel_{i,j}$, and $w_{x,y}$ be the weight of the edge connecting $pixel_x$ and $pixel_y$. $F_{i,j}$ can then be calculated using the following recurrence:

$$F_{i,j} = min\begin{Bmatrix} F_{i-1,j-1} + w_{(i-1,j-1),(i,j)} \\ F_{i-1,j} + w_{(i-1,j),(i,j)} \\ F_{i-1,j+1} + w_{(i-1,j+1),(i,j)} \end{Bmatrix}$$

The time taken for this method is long, since after having found one path, we need to remove it and rebuild the graph. Hence, this leads to the Stable Paths approach. Given $pixel_{1,j}$, let $cols$ be the number of columns, let the end point of the shortest path starting from $pixel_{1,i1}$ be $pixel_{cols,i2}$, then if the start point of the shortest path ending at $pixel_{cols,i2}$ is $pixel_{1,i1}$, we say the path from $\mathrm{pixel}_{1,i1}$ to $pixel_{cols,i2}$ is stable, and that is the staff line. Thus, for a given graph, we can find multiple staff lines at once by finding all stable paths. The algorithm stops when a new path does not contain the number of black pixels larger than a threshold (taken as 70% of the line with the maximum black pixels experimentally).

3.1.3 Symbols segmentation

Normally, finding connected components is good enough to segment out every symbol in a musical score. However, there is a symbol called beam that connects multiple different notes together, which leads to a connected component consisting of several notes. As a result, the beams need to be found in order to split the components into different notes.

The authors inherit the idea of beam detection from [4]. For a region of symbols with beams, it is noted that the stems are the ones connecting the beams with the note heads. Hence, the algorithm firstly removes the stems from the region. The stems are detected by looking for long vertical run-length of black pixels with length larger than a threshold (taken as $2 * staffSpaceHeight$ through experiments). After that, connected components are found again on the region. The beams will be detected by choosing components with height lower than $4 * staffSpaceHeight$, width larger than $2 * staffSpaceHeight + 2.5 * staffLineHeight$, and are connected with stems (the values are taken experimentally).

After having removed the beams, the remaining connected components are extracted out. They are considered as music symbols and will be classified in the classification process.

3.2 Music symbol recognition

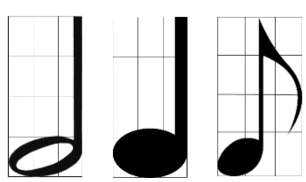

Figure 2. Examples of representing music symbols by grids of $M * N$ cells

For an extracted symbol from the previous step, the author's objective is to classify it into the appropriate class. The authors propose to use support vector machines as the classifier for this step. The classification step is divided into 2 phases, the training phase and the classification phase.

Both the training phase and the classification phase all share the common work of converting images into feature descriptors. This process is described as follows. For a given image of size $W * H$ (W equals to width, and H equals to height of the image), the image is divided into a grid of size $M * N$. Cell (i, j), which is located at column i and row j of the grid, will contain all pixels of the image that are in the rectangle with top-left pixel $(\lfloor i * W/M \rfloor, \lfloor j * H/N \rfloor)$ and right-bottom pixel $(\lfloor (i+1) * W/M \rfloor, \lfloor (j+1) * H/N \rfloor)$. After that, cell (i, j) is assigned the color with the most frequency in its corresponding rectangle in the image. The grid is then converted into a 1-dimensional descriptor with size $1 * M * N$, where value of cell (i, j) in the grid is assigned to cell $(1, i * M + j)$ in the descriptor. The feature descriptors of all sample images are then used as training materials for the SVM classifier.

For an image of an extracted symbol, the SVM classifier is used to calculate the probability that this symbol belong to any of the class. At first, the result for a symbol is chosen to be the class with the highest probability. However, the precision achieved using this method is not as high as expected. As a result, the authors propose to apply a strategy in which multiple SVM classifiers are trained and used in order to improve the accuracy on this classification step. In general, a number of k SVM classifiers are chosen to train on a proportion of samples images, instead of training all of the samples for a single classifier. Then by applying voting method, the class with the highest majority in the prediction of the k SVM classifiers is chosen to be the classification result for the symbol.

4. EXPERIMENTAL RESULTS

There are several available datasets to test on different stages. For staff lines detection and removal, 2 datasets including Synthetic Score Database [5] by Christoph Dalitz, and the CVC-MUSCIMA Database by Alicia FornTs [11] are used. For the classification phase, since the dataset of Desaedeleer [12] contains staff lines remaining in the images of the data, the authors cannot use it for training and testing. Furthermore, datasets from the other authors are not public. As a result, the authors create their own classification dataset by extracting symbols from the Synthetic Score Database.

4.1 Staff lines detection and removal

As the main reason of the paper is to propose the multiple SVM classifiers algorithm for the classification step, the authors test this step only on the Synthetic Score Database of Christoph Dalitz [5]. The precision of the system is calculated by counting the number of black pixels that belong to a symbol, but are mistakenly removed by the staff lines removal algorithm. The system is tested on the 32 printed ideal scores of the Database, and got the precision results averagely 97.74%. The main problem is that it mistakenly removes pixels belong to symbols, hence, it is necessary to figure out a way to determine if a pixel belongs to any symbol or not.

4.2 Music symbols recognition

For the training and classification phase, the authors create a ground truth dataset consists of music symbols extracted mainly from the 18 modern musical scores in the Synthetic Score Database. The dataset consists of a total of 4929 music symbols, and is randomly splitted into training set and testing set.

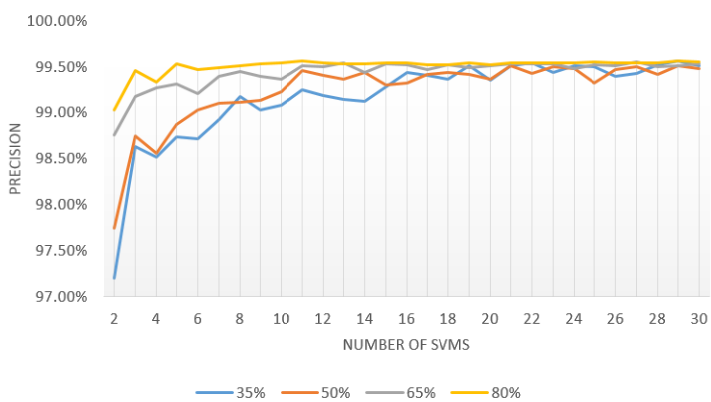

Figure 3. Precision of music symbols recognition in 4 scenarios of training SVM classifiers using 35%, 50%, 65%, 80% of training samples

First we evaluate to find the optimal values of M and N to represent music symbols. By iterating the value of each parameter from 5 to 100, we decide to use M = 10 and N = 12 because these two values yield the highest average precision. Second, we evaluate the efficiency of using only a small proportion of sample images to train multiple lightweight SVM classifiers to achieve high accuracy for music symbols recognition. We consider 4 scenarios, each of which corresponds to the value of X - the proportion of sample images to train for each classifier – of 35%, 50%, 65%, and 80% respectively. For each scenario, we evaluate the accuracy of using k SVM classifiers where k is from 2 to 30.

Figure 3 illustrates the accuracy of the 4 scenarios with different number of SVM classifiers on the same testing dataset. For each scenario, the accuracy first increases rapidly when the number of SVM classifiers increases; then the accuracy tends to converge towards a stable value. It is obvious that the more sample images are used to train classifiers, the higher accuracy we can achieve with the same number of classifiers. We achieve the highest accuracy of 99.56% from the case of using 11 SVM classifiers trained with 80% sample images. It is interesting that even when using only a small proportion of sample images (35%), we can achieve the accuracy of 98.5% using 4 classifiers, and 99% using 10 classifiers. This demonstrates that our proposed method can provide high accuracy for music symbols recognition with multiple SVM classifiers even with small number of training samples.

5. CONCLUSION

The authors propose a simple method to classify printed music symbols in Optical Music Recognition with two main contributions. First, we propose a simplified method to detect and remove staff lines based on the Stable Paths approach. Our goal is not to achieve the perfect accuracy for this step but low computational cost and acceptable results for the main second step: music symbols recognition. Second, we propose to represent music symbols by dividing their images into grids of $M * N$ cells and conduct experiments to find the optimal values of $M = 10$ and $N = 12$. By using the idea of multiple lightweight SVM classifiers, we see that by training only with 35% proportion of samples and using 10 SVM classifiers, the result achieved is as high as 99%. And by training with more samples using more SVM classifiers, the highest precision value is 99.56% on the dataset created by the authors consisting of music symbols extracted from 18 modern music sheets in the Synthetic Score Database. Currently we continue to study different methods for classifications, especially deep neural network approach, for OMR with both printed and handwritten musical scores.

REFERENCES

[1] Optical Music Recognition Bibliography, a list of works done on OMR: http://ddmal.music.mcgill.ca/wiki/Optical_Music_Recognition_Bibliography - accessed on Feb 25th 2014.

[2] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R. S. Marcal, Carlos Guedes, and Jaime S. Cardoso, "Optical music recognition: state-of-the-art and open issues," International Journal of Multimedia Information Retrieval, Vol 1, Issue 3, 173-190 (2012).

[3] Nobuyuki Otsu. "A threshold selection method from graylevel histograms,". IEEE Transaction on Systems, Man, and Cybernetics, Vol 9, No. 1, 62-66 (1979).

[4] Ana Rebelo, "New methodologies towards an automatic optical recognition of handwritten musical scores," Master of Science thesis, University of Porto, Portugal, 2008.

[5] Synthetic Score Database: http://gamera.informatik.hsnr.de/addons/musicstaves/testsetmusicstaves.tar.gz - accessed on Feb 25th 2014.

[6] D. Pruslin, "Automatic recognition of sheet music," PhD thesis, Massachusetts Institute of Technology (1966).

[7] D. Prerau, "Computer pattern recognition of standard engraved music notation," PhD thesis, Massachusetts Institute of Technology (1970).

[8] Donald Byrd, "Optical Music Recognition Systems survey," School of Informatics and School of Music, Indiana University (rev. Jan. 2007).

[9] Bolan Su, Shijian Lu, Umapada Pal, and Chew Lim Tan, "An effective staff detection and removal technique for musical documents,", The 10th IAPR International Workshop on Document Analysis Systems DAS 2012, 160-164 (2012).

[10] Ana Rebelo and Jaime S. Cardoso, "Staff line detection and removal in the grayscale domain," The 12th International Conference on Document Analysis and Recognition ICDAR 2013, 57-61 (2013).

[11] CVC-MUSCIMA Database: http://www.cvc.uab.es/cvcmuscima/ - accessed on Feb 25th 2014.

[12] Desaedeleer Open source OMR project: http://sourceforge.net/p/openomr/wiki/Home/ - accessed on Feb 25th 2014.