Virtual Music Teacher for New Music Learners with Optical Music Recognition

Viet-Khoi Pham, Hai-Dang Nguyen, Minh-Triet Tran

Faculty of Information Technology, University of Science, VNU-HCM

pvkhoi@apcs.vn, nhdang12@apcs.vn,
tmtriet@fit.hcmus.edu.vn

Abstract. Learn to read and understand a music sheet, then play it on a musical instrument are difficult tasks to most beginner music learners. This motivates the authors to propose Virtual Music Teacher, a system to assist beginner music learners in their learning process. By applying our proposed lightweight Optical Music Recognition algorithm to scan and recognize a music sheet, then combine with sound classifying technique, the proposed system can learn what note to be played next, then help a music learner to play it correctly. The experimental results on the dataset consisting of 15 musical scores for beginners show that the proposed system can classify with precision up to 99.9% using multiple SVM classifiers approach, whereas the sound classifying technique using Fast Fourier Transform can classify note's pitch recorded from a piano with precision up to 95.71%. The system is implemented as an application on mobile devices and can be used to assist a music learner to play not only piano but other musical instruments as well.

Keywords: optical music recognition, note's pitch recognition, virtual music teacher

1 Introduction

It is difficult for a beginner music learner to read a musical score sheet, then to recognize a musical note, i.e. its pitch according to its position on staff lines and its duration established by its note head, stem, and flag. In class, a teacher helps new learners to identify and to play notes in a musical score sheet. However when music learners practice at home, there is no one to warn them when they read or play wrong notes. Therefore, it is necessary to have helpers to notify learners when they make mistakes and assist them in recognizing what musical notes to play next, even when they are practicing by themselves at home, without any teachers.

Although there are different games and utilities in computers or mobile devices to teach or support users to learn playing music, these applications mainly provide lessons, games, or exercises with fixed contents or scenarios. Thus, it would be more efficient for users to have real-time guidance adapting to their current practice on real musical instruments. This motivates us to propose Virtual Music Teacher, a system to assist beginner pianists in their learning process to play piano. Our proposed system can also be adapted to assist music learners to play other instruments.

Our proposed Virtual Music Teacher system has two main useful features for new music learners. First, it can recognize musical notations from regular printed musical score sheets, then speaks out which note to be performed next. Second, it records sound performed by the learner in real-time, recognizes which note is played, checks with the recognized note in the musical score sheet, and give warnings to the learner if he or she plays a wrong note.

The main contribution of this paper is that we propose our idea to apply optical music recognition (OMR [1]) and sound recognition to develop a Virtual Music Teacher system to assist new music learners. We propose an improved version of our light-weight method for optical music recognition [2] to recognize musical notations in musical score sheets with high accuracy and low computational cost. Besides, we also utilize a simple method based on Fast Fourier Transform (FFT) to efficiently recognize musical notes from audio recorded in real-time from a piano at difference distances.

We conduct experiments for our proposed OMR algorithm with 15 musical scores for beginners to play piano and our method achieves precision up to 99.9% using multiple SVM classifiers approach. Then we also conduct experiments for musical note classification from audio recorded in real-time from a piano and the precision is up to 95.71%. With these promising experimental results, our method and its implementation on mobile devices can be used to assist new music learners in their studying and practice to play piano as well as other musical instruments.

The content of this paper is as follows. In Section 2, we briefly review related methods of optical music recognition. Section 3 presents the overview of our proposed Virtual Music Teacher for new music learners. The detailed information of our proposed method to recognize a music sheet is presented in Section 4 whereas our method to recognize musical notes based on audio is discussed in Section 5. Section 6 shows experimental results for optical music recognition and musical note recognition from audio. Conclusions and future work are discussed in Section 7.

2 Background & Related Works

There are various applications on computers or mobile devices to help users to learn how to play musical instruments. These applications usually provide lessons or games to assist users in learning to play music. Several applications also allow a user to practice with audio or visual hints on a virtual musical instrument, such as a virtual keyboard of a piano. However, existing applications do not provide real-time warnings or hints corresponding to the real context when a music learner is practicing on a real musical instrument.

To realize our idea of a virtual music teacher with useful warnings and hints in real-time, the first task is Optical Music Recognition (OMR), i.e. to recognize all music symbols in a score sheet. In late 1960s, Pruslin [3] and Prerau [4] initiated the first steps into the field of OMR. Initially the main objective of OMR is to preserve musical scores and to help music composers to digitalize music sheets into machine-readable format.

However, OMR can also be applied to develop a system that can automatically perform a song directly from a musical score sheet [5].

Although there are different OMR methods, they usually follow a common process. After necessary pre-processing steps, staff lines should be removed before music symbols are extracted and classified with a particular classification method. We inherit this common process in our proposed method [2] and its enhanced version in this paper.

Various classification methods have been proposed to recognize music symbols extracted from a musical score. Template matching is among the first and simple approaches for music symbol recognition [6][7]. Other techniques in machine learning are also used to process music symbols, such as Hidden Markov Model [8], Support Vector Machine [9], or Neural Network [9]. In our proposed method [2] and its enhanced version, we also use Support Vector Machine (SVM) as the main tool to classify a musical notation. However, we do not use a single SVM classifier but we train multiple classifiers and combine the outputs of them to determine which class a musical notation belongs to. By this way, we can boost the overall accuracy of our method.

3 Proposed System

Figure 1 and Figure 2 illustrate the overview of the two main processes in our proposed Virtual Music Teacher, including the musical score recognition process to give hints and the sound recognition process to give warnings to music learners, respectively.

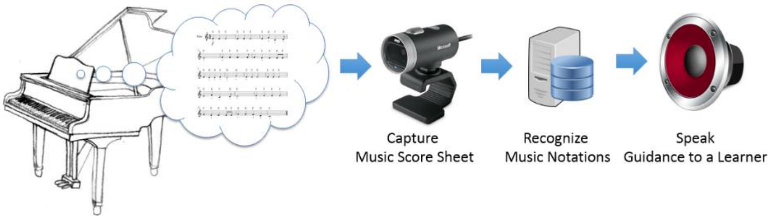


Fig. 1. Overview of musical score recognition process to give hints to music learners

In Figure 1, a regular printed musical score sheet is captured by a regular camera. Musical notes and notations are recognized by our proposed light-weight OMR algorithm. Hints are spoken out by a speaker corresponding to a recognized note so that a learner knows which note to play next.

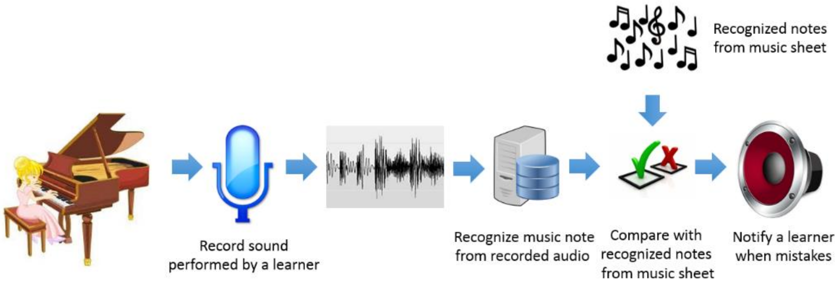


Fig. 2. Overview of sound recognition process to give warnings to music learners

In Figure 2, when a learner is playing a music lesson, his or her performance is recorded and processed in real-time to recognize which note is performed. The note recognized from recorded audio is compared with the expected note recognized from music sheet to verify if the learner plays the correct note. If the learner plays a wrong note, a warning is generated and notified to the learner via a speaker.

4 Light-weight optical music recognition method

We follow the common framework for recognizing musical notes to propose our light-weight OMR method [10]. Figure 3 shows the preprocessing and recognition phases of our OMR method.

In the first phase, staff lines are removed after the binarization step. We simplify and apply the Stable Paths approach for this step [11]. We also suggest a method to simplify the complexity of the symbol segmentation step based on the idea in [11]. 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 system is the enhanced version of our method in [2]. In the enhanced version, staff lines detected are used for filtering out the noise symbols. The removing staff lines process is also improved with a method to avoid breaking the symbols apart. One other improvement is to use the symbol beams detected to deduce the note's type.

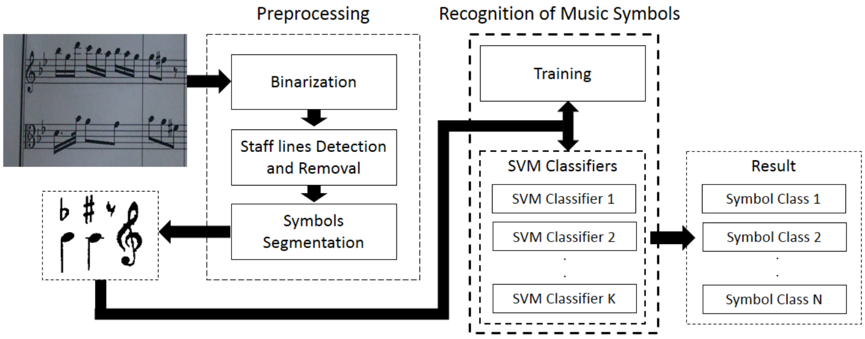


Fig. 3. Proposed framework for preprocessing and recognition phases in OMR.

4.1 Preprocessing

Binarization.

The binarization algorithm proposed by Otsu [12] is suggested to separate the image of the score into foreground and background. Otsu's method is a global binarization algorithm, thus for image with dark and white areas, the algorithm is likely to fail. However, music learners who use the system for their training, is required to take a clear picture of the musical score. Hence the score's image is always expected to be clear without dark areas. As a result, only using Otsu's global binarization algorithm still brings satisfactory results in this step.

Staff lines detection and removal:

Staff lines detection:
It is necessary to detect and remove staff lines from music sheets, since they overlap with music symbols and make the segmentation and classification step become difficult. By detecting staff lines' positions, the results can be used to determine the pitch of any musical notes. Moreover, staff lines' positions can also be used for validating the presence of other music symbols (i.e. any symbols that are far from the staff lines will be ignored as there is a high possibility that they are the lyrics, titles… or any unused data in the score).

One existing algorithm to detect staff lines is using Hough Line transformation [13] to detect all lines in the image. However, the authors suggest using a simplified method based on the Stable Paths approach [11]. The approach's idea is to build a graph for the musical score. Every pixel in the score is represented by a vertex; and if 2 pixels in 2 consecutive columns are adjacent to each other, then there exists an edge connecting the 2 vertices. Every edge in the graph is assigned with weight based on the following rule: initially, each edge is given with weight equal to 2; if the 2 pixels of the edge are diagonally adjacent, the weight is incremented by 1; after that, the edge's weight is incremented by a value equal to the number of white pixels in the 2 of them. The graph is successfully built, and staff lines' position can be detected by finding the shortest paths from the first to the last column of the image.

The problem of finding the shortest paths from the first to the last column can be solved using dynamic programming. Let $pixel_{i,j}$ be the pixel on row i and column j of the image, $F_{i,j}$ be the cost of the shortest path that starts from a pixel in column 1 and stops at $pixel_{i,j}$, and $w_{x,y}$ be the weight of the edge connecting $pixel_x$ and $pixel_y$. $F_{i,j}$ is calculated using the following recurrence:

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

The value of the shortest path is the minimum value in the last column of F, and the shortest path itself is the staff line.

The algorithm's time complexity is high, since after one path is found, that staff line is removed from the score and the graph is rebuilt to find the next staff line. For instance, there is a large computational cost to run the algorithm 20 times to find 20 staff lines in a 1000 by 2000 music sheet. Hence, this problem leads to the idea of the Stable Paths approach [11]. The idea is as follows: let $cols$ be the number of columns in the image; let the end pixel of the shortest path starting from $pixel_{i1,1}$ (an arbitrary pixel in column 1) be $pixel_{i2,cols}$, then if the start pixel of the shortest path ending at $pixel_{i2,cols}$ is

$pixel_{i1,1}$, we say the path from $pixel_{i1,1}$ to $pixel_{i2,cols}$ is a stable path. Thus, it is possible to find all staff lines at once by finding all stable paths in the graph. However, it is necessary to validate the staff lines by calculating the percent of black pixels lie on the paths (the threshold value is taken as 70% by the authors).

Staff lines removal:

After detecting all staff lines in the musical score, it is not obvious to simply remove them by assigning them with white pixels, because this will cut through the music symbols and divide them into smaller different parts. Thus, the authors propose a simple idea: for every black pixel *p* belongs to a staff line, we find the 2 nearest white pixels above and below it on its column; if the distance between the 2 pixels is less than a threshold value (taken as *staffLineHeight + 1* by the authors), then the pixel is assigned with white pixels to remove it from the score. This method is to prevent removing pixels that both lie on staff lines and music symbols.

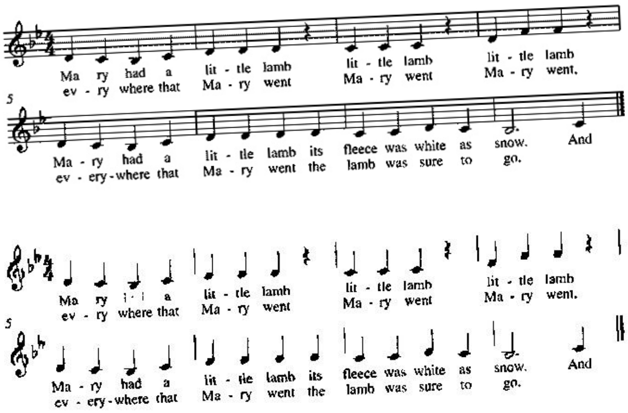

Ma ry had a lit - tle lamb lit - tle lamb lit - tle lamb
ev - ry where that Ma - ry went Ma - ry went Ma - ry went.
5
Ma - ry had a lit - tle lamb its fleece was white as snow. And
ev - ery-where that Ma - ry went the lamb was sure to go.
Ma ry i · i a lit - tle lamb lit - tle lamb lit - tle lamb
ev - ry where that Ma - ry went Ma - ry went Ma - ry went.
5
Ma - ry had a lit - tle lamb its fleece was white as snow. And
ev - ery-where that Ma - ry went the lamb was sure to go.

Fig. 4. Successfully remove staff lines when the musical score is inclined

Symbols segmentation:

It is possible to collect all connected components of black pixels in the image after removing staff lines, then treat them as each individual music symbol for classifying in the next step. However, there is a case when multiple quarter notes are connected by beams, leading to a component consisting of several notes. Thus, beams need to be detected to split the component into different notes, and to convert the notes into appropriate notes (i.e. quarter note connected with 2 beams becomes sixteenth note, etc.). The authors suggest using the idea of beam detection from [11]. Firstly, stems connecting beams and noteheads are removed from the component. The stems are detected by finding long vertical run-length of blacks pixels with length larger than a threshold ($2 * staffSpaceHeight$). After that, connected components are found again on the component. The beams will be the components with height less than $4 * staffSpaceHeight$, width over $2 * staffSpaceHeight + 2.5 * staffLineHeight$, and are connected with stems (the values are taken experimentally).

The remaining connected components are extracted out after the beams are removed. The components are considered as music symbols to go through the next step of classification.

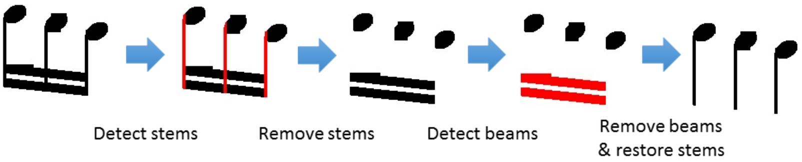


Fig. 5. Process of detecting and removing beams

4.2 Music symbol recognition:

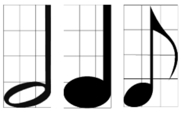


Fig. 6. Examples of representing music symbols by grids of $M * N$ cells

For each music symbol segmented, the objective is to classify it into the correct class of symbol. Support vector machines algorithm is used as the classifier for this step. The approach of using SVM classifiers is based on the method that is applied in [2]. There are 2 processes in the classification step, the training phase and the classification phase.

Before working on the training and the classification phase, it is necessary to convert music symbols into feature vectors. As each symbol has different height and width compared with the others, the next step is to represent music symbols by grids of the same size $M * N$. For each symbol of size $W * H$ (W – width, H – height), the image of the symbol is divided into grid of size $M * N$. For each cell (i, j) in the $M * N$ grid, the cell is assigned with color black or white depending on the higher number of black or white pixels in the corresponding cell (i, j). The grid is then converted into a 1-dimensional feature vector with size $1 * (M * N)$. The feature vectors of all music symbols are then used as training and testing data for the SVM classifier.

Given an image of a music symbol, the SVM classifier will find the class with the highest probability that the music symbol belongs to. However, the precision result achieved using one SVM classifier is not high as expected. Thus, the authors propose to apply a new method of using multiple SVM classifiers to train and classify in order to improve the accuracy on the classification problem. In general, a number of k SVM classifiers are trained with only a proportion of samples data (instead of all of the data for a single classifier). When classifying a music symbol, the class that the majority of SVM classifiers predict is chosen to be the result class.

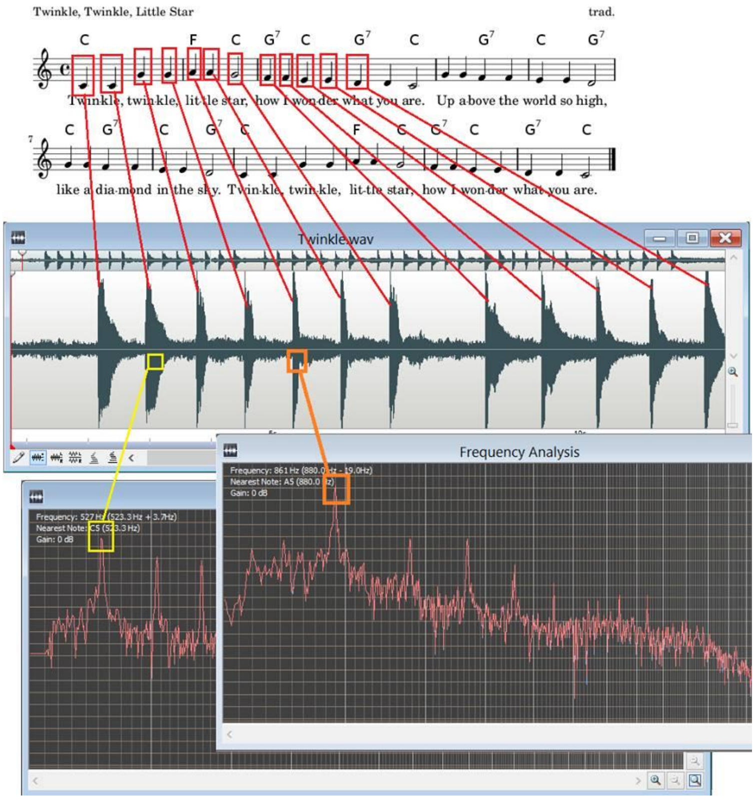
Several connected components extracted from the previous step are not music symbols but noise. The noise can be filtered out by finding the distance between each component to the nearest staff line. If the distance is larger than $2.5 * staffSpaceHeight$, then the component is considered as noise and it is removed.

After recognizing all music symbols in the score, the notes are sorted by their coordinates to figure out the order to play the notes. The coordinates of the symbols combined with the positions of the staff lines help to deduce the pitch of the symbols.

5 Recognize music notes based on audio

The authors aim to serve and help beginner music learners to lean and practice music, thus the system is only developed for using with music sheets consist of the following fundamental music symbols: quarter note, eighth note, half note, whole note, flat, sharp, and natural. As beginner music learners usually practice with one hand, the authors only conduct experiment on notes belonging to 2 octaves C4 and C5.

In order to recognize the musical note's pitch from a given audio file, the authors conduct recording sound of each note's pitch. Each note's pitch is recorded 5 times, as there are 2 octaves with 12 pitches per octave, we have 120 recording files in total. After having recorded, the authors reduce the noise and use Fourier Transform algorithm to convert sound waves into frequency [14]. Then, the authors base on the highest peak of frequency in order to distinguish and identify the note's pitch.

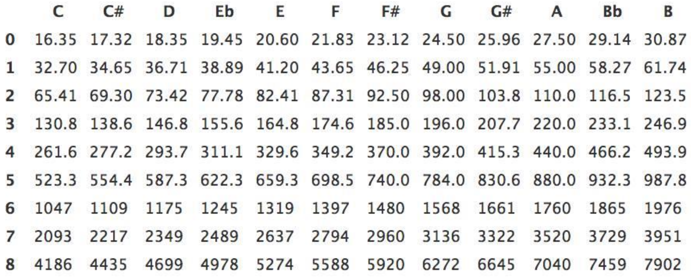

Twinkle, Twinkle, Little Star
trad.
C F C G7 C G7 C G7 C G7
Twinkle, twinkle, little star, how I wonder what you are. Up above the world so high,
7
C G7 C G7 C F C G7 C G7 C
like a diamond in the sky. Twinkle, twinkle, little star, how I wonder what you are.
Twinkle.wav
5s
10s
Frequency Analysis

Frequency: 527 Hz (523.3 Hz + 3.7Hz)
Nearest Note: C5 (523.3 Hz)
Gain: 0 dB

Frequency: 861 Hz (880.0 Hz - 19.0Hz)
Nearest Note: A5 (880.0 Hz)
Gain: 0 dB

Fig. 7. Every note's pitch has a unique highest peak of frequency.

The authors decide to use Fourier Transform because the experimental results show that 100% notes that have the same pitch will produce the same highest peak of frequency, and the highest peak of frequency is distinguishable between different note's pitch. This dataset is used for the system to recognize new input note's audio.



	C	C#	D	Eb	E	F	F#	G	G#	A	Bb	B
0	16.35	17.32	18.35	19.45	20.60	21.83	23.12	24.50	25.96	27.50	29.14	30.87
1	32.70	34.65	36.71	38.89	41.20	43.65	46.25	49.00	51.91	55.00	58.27	61.74
2	65.41	69.30	73.42	77.78	82.41	87.31	92.50	98.00	103.8	110.0	116.5	123.5
3	130.8	138.6	146.8	155.6	164.8	174.6	185.0	196.0	207.7	220.0	233.1	246.9
4	261.6	277.2	293.7	311.1	329.6	349.2	370.0	392.0	415.3	440.0	466.2	493.9
5	523.3	554.4	587.3	622.3	659.3	698.5	740.0	784.0	830.6	880.0	932.3	987.8
6	1047	1109	1175	1245	1319	1397	1480	1568	1661	1760	1865	1976
7	2093	2217	2349	2489	2637	2794	2960	3136	3322	3520	3729	3951
8	4186	4435	4699	4978	5274	5588	5920	6272	6645	7040	7459	7902

Fig. 8. Frequency of pitches [15]

6 Experiments and Implementations

6.1 OMR

Experimental results on the dataset consisting of 4929 music symbols taken from 18 modern music sheets in the Synthetic Score Database [16] show that our proposed method is able to classify printed musical scores with accuracy up to 99.56% using 11 SVM classifiers trained on 80% sample images [2]. However, as the authors only aim to develop the system for beginner music learners, the system is tested on a new dataset consist of 15 simple music sheets for beginners using the same number of SVM classifiers and proportion of samples data (11 SVM classifiers and 80% samples). As the new dataset is more simple compared to the Synthetic Score Database, the accurarcy acquired becomes higher: 99.9% precision and 98.34% recall.

6.2 Recognize note's pitch based on audio

The authors propose experiments to test the accuracy of the recognition step in the case the recording environment has noise and the recorder is placed far from the player.

Our experiment is conducted by recording multiple times the performance of a music learner. The records are taken from different angles, distances, relative to the position of the player (the record even has noise and errors made by the player when he/she mistakenly plays 2 notes at the same time). After that, the authors test the accuracy of the system by playing each record and take note of the result returning from the system everytime it listens to the sound of a note. More specifically, when the system hears the sound of a note, the system will analyze the sound wave and use Fourier Transform algorithm to get the sound frequency and compare it with its dataset. The frequency data in the dataset that looks closest to the new input audio is returned as the result from the system.

There is one problem in the case that the player plays 2 notes at the same time. This leads to the frequency graph having 2 peaks with the same height (each peak for one note's pitch). However, the algorithm only chooses the highest one, because the system is unable to know when the player makes mistakes. Hence, the system may output the wrong note in this case. The problem is illustrated in Figure 9.

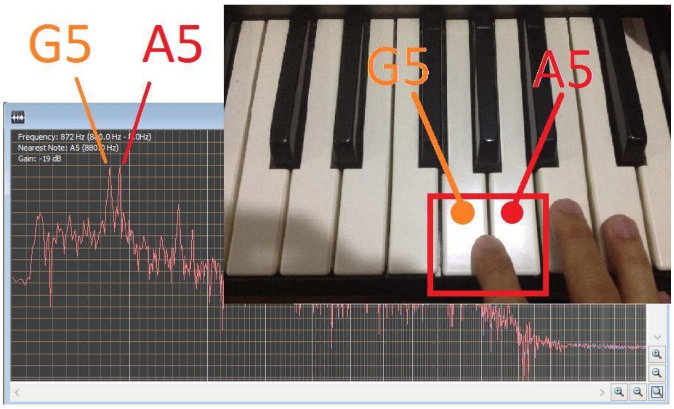

G5
A5
Frequency: 872 Hz (880.0 Hz - 8.0Hz)
Nearest Note: A5 (880.0 Hz)
Gain: -19 dB
G5
A5

Fig. 9. There are 2 highest peaks when the player plays 2 notes at the same time.

The experiment is conducted on a musical score consists of 42 musical notes, including 36 quarter notes and 6 half notes. The performance of the player on this musical score is recorded five times with the distance between the recorder and the player as follows: two times on the left and two times on the right, with the distance of 0.25 meter and 0.5 meter; one time in the middle with the distance 0.25 meter. In 210 notes played by different beginner piano players, only 9 notes are recognized incorrectly, yielding the accuracy of the experiment to be 95.71%. All the incorrect cases are due to a player presses on two or more keys on the keyboard at the same time.

6.3 Implementation

We implement our proposed Virtual Music Teacher in two versions. Beside the initial prototype with a regular laptop with a webcam and a speaker, we also develop our system on mobile devices. Figure 10 shows the main processing components of Virtual Music Teacher implemented on a mobile device. A new music learner can simply use a mobile device to capture a regular printed music lesson via the built-in camera, then our system recognizes music notes, generates hints via the built-in speaker, records sound via the built-in microphone, recognizes notes in recorded sound, and generates warnings via the built-in speaker.

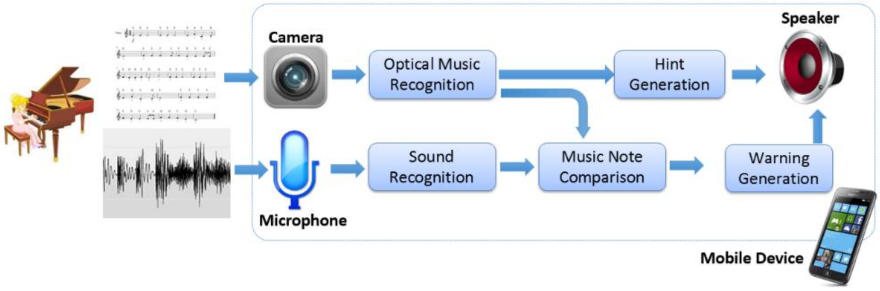


Fig. 10. System implementation on mobile devices

7 Conclusion

Our Virtual Music Teacher system is appropriate to assist new music learners in the very first step to study music and play musical instruments. Lessons for beginners are usually easy with simple notes and notations. This ensures that our proposed method OMR can provide high accuracy to give hints for a music learner to play. Furthermore, such lessons are usually in nearly monotonous rhythms with slow tempo. Thus, the sound recognition process to give warnings to music learners can perform well in audio segmentation and note recognition.

We will continue to integrate the augmented reality feature for smart eyewares to show hints as real-time highlights on the keyboard of a piano to further support to a music learner on playing correct keys.

8 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. Pham, V.K, Nguyen, H.D., Nguyen-Khac, T.A., Tran, M.T.: Apply Lightweight Recognition Algorithms in Optical Music Recognition. Seventh International Conference on Machine Vision (ICMV 2014). Proc. of SPIE Vol. 9445 (2015).
3. Pruslin, D.: Automatic recognition of sheet music. PhD thesis, Massachusetts Institute of Technology (1966).
4. Prerau, D.: Computer pattern recognition of standard engraved music notation. PhD thesis, Massachusetts Institute of Technology (1970).
5. Byrd, D.: Optical Music Recognition Systems survey. School of Informatics and School of Music, Indiana University (rev. Jan. 2007).
6. Rossant, F., Bloch, I.: Robust and adaptive OMR system including fuzzy modeling, fusion of musical rules, and possible error detection. EURASIP Journal on Applied Signal Processing, Vol. 2007, No. 1, pp. 160–160 (2007)
7. Toyama, F. , Shoji, K., Miyamichi, J.: Symbol recognition of printed piano scores with touching symbols. Proceedings of the International Conference on Pattern Recognition,pp. 480–483 (2006).

8. Pugin, L.: Optical music recognition of early typographic prints using Hidden Markov Models. Proceedings of the International Society for Music Information Retrieval, pp. 53–56 (2006).
9. Rebelo, A., Capela, G., Cardoso, J. S.: Optical recognition of music symbols: A comparative study. International Journal on Document Analysis and Recognition, Vol. 13, pp.19–31 (2010).
10. Rebelo, A., Fujinaga, I., Paszkiewicz, F., Marcal, A.R.S, Guedes, C., Cardoso, J.S.: Optical music recognition: state-of the-art and open issues. International Journal of Multimedia Information Retrieval, Vol 1, Issue 3, pp. 173-190 (2012).
11. Rebelo, A. New methodologies towards an automatic optical recognition of handwritten musical scores. Master of Science thesis, University of Porto, Portugal (2008).
12. Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, Issue 1, pp. 62-66 (1979).
13. Duda, R. O., Hart, P. E.: Use of the Hough Transformation to Detect Lines and Curves in Pictures. Communications of the ACM, Vol. 15, Issue 1, pp 11-15 (1972).
14. Marchand, S.: An efficient pitch-tracking algorithm using a combination of Fourier Transforms. Proceedings of the Conference on Digital Audio Effects (DAFX 2001), pp. 170-174 (2001).
15. Frequency of pitches: http://www.seventhstring.com/resources/notefrequencies.html - accessed on Mar 8th, 2015.
16. Synthetic Score Database: http://gamera.informatik.hsnr.de/addons/musicstaves/testset-musicstaves.tar.gz - accessed on February 25th, 2014.