Optical Music Recognition Using Multiple Neural Networks on Paper-based Handwritten Musical Score

A Thesis Proposal

Presented to the

Faculty of the College of Computer Studies of

Silliman University

XAVIER JOHANIS ELNAS

ANDREW VINCOY

YOHAN HWANG

18 July 2017

Table of Contents

[1.](#_1fob9te) Introduction 1

[1.1](#_2et92p0) Project Context 1

[1.2](#_3dy6vkm) Purpose & Description 1

[1.3](#_4d34og8) Objectives 2

[1.4](#_3rdcrjn) Scope & Limitation 2

[1.4.1 Scope 2](#_lnxbz9)

[1.4.2 Limitation 2](#_44sinio)

[1.5](#_z337ya) Significance of the Study 3

[1.6](#_1y810tw) Definition of Terms 3

[2.](#_2xcytpi) Review of Related Literature 5

[3.](#_3whwml4) Technical Background 9

[3.1 Hough Line Transform 9](#_2bn6wsx)

[3.2](#_3as4poj) Global Thresholding 10

[3.3](#_49x2ik5) Bounding Box Algorithm 11

[3.4 Multi-Layer Perception 11](#_147n2zr)

[4.](#_23ckvvd) Methodology 13

[4.1 General Framework 13](#_32hioqz)

[4.2 Image Preprocessing 14](#_2grqrue)

[4.2.1 Binarization 14](#_3fwokq0)

[4.2.2 Deskewing 14](#_4f1mdlm)

[4.2.3 Morphology 15](#_19c6y18)

[4.3 Segmentation 15](#_nmf14n)

[4.3.1 Staff Line Detection & Removal 16](#_1mrcu09)

[4.3.4 Musical Symbol Segmentation 16](#_3l18frh)

[4.5 Musical Note Classification 17](#_4k668n3)

[4.6 Training 18](#_2zbgiuw)

[5. Results and Evaluation 19](#_3ygebqi)

[6. Conclusion 25](#_sqyw64)

[6.1 Recommendation 26](#_3cqmetx)

[7.](#_4bvk7pj) References 27

Table of Figures

[Figure 1: Template with cleff 2](#_35nkun2)

[Figure 2:Basic Principle of Hough Transform 9](#_qsh70q)

[Figure 3: Otsu’s Method 10](#_1pxezwc)

[Figure 6: Bounding box for a circle](about:blank) 11

[Figure 7: The MLP Structure in each neural network](about:blank) 12

[Figure 8: Framework Model 13](#_1hmsyys)

[Figure 9: An image and a binarized image 14](#_1v1yuxt)

[Figure 10: Deskewed Image Using Hough Transform](about:blank) 15

[Figure 11: Skewed Image](about:blank) 15

[Figure 12: staffline removal 16](#_2lwamvv)

[Figure 13: Musical notes in the bounding box](about:blank) 17

[Figure 14: pixel counting](about:blank) 18

[Figure 15: CNN structure 18](#_1egqt2p)

[Figure 16: Bounding box drawback 20](#_2dlolyb)

**Abstract**

A significant amount of musical works produced in the past are still available only as original manuscripts or as photocopies on date. The Optical Music Recognition (OMR) is needed for the preservation of these works which requires digitization and should be transformed into a machine readable format. Such a method is one of the most promising tools to preserve the music scores. In addition, it makes the search, retrieval and analysis of the music sheet easier. An OMR program should thus be able to recognize the musical content and make semantic analysis of each musical symbol of a musical work. Generally, such a task is challenging because it requires the integration of techniques from some quite different areas, i.e., computer vision, artificial intelligence, machine learning, and music theory.

Technically, the OMR is an extension of the Optical Character Recognition (OCR). However, it is not a straightforward extension from the OCR since the problems to be faced are substantially different but the main goal of this study is to automatically decode and interpret the symbols of the music notation from scanned images.

The approach of this research is as follows, pre-processing, segmentation and musical note classification. For the preprocessing stage we want to straighten the image or basically deskew it and in order to do that we convert the scanned image into a binary image that has only two possible values which is black and white or 0’s and 1’s respectively to detect the lines in the images and calculate its appropriate angle that is 180 degrees which is a straight line. Afterwards is the segmentation stage where we remove staff lines as much as possible while maintaining to preserve the shapes of the musical symbols, then we extract the remaining symbols for our last stage which is the training of our classifier.

In this study, we concluded that using multiple neural networks as a classifier in Optical Music Recognition (OMR) handwritten musical symbols as its parameters has a 85.12% high detection rate, also in this study we used a new OMR analysis method of combining multiple neural networks that accelerates the recognition process.

**Aknowledgement**

We would first like to thank our thesis adviser Dr. Chuchi S. Montenegro. The door to Dr. Montenegro’s office was always open whenever we ran into a trouble spot or had a question about our research. She consistently allowed this paper to be our own work, but steered us in the right the direction whenever she thought we needed it.

We would also like to thank the participants in the survey involved for this research project: COPVA Students in general. Without their passionate participation and input, the survey could not have been successfully conducted.

We would also like to acknowledge our panel members: Prof. Marianne Yao, Prof. Albert Rivera and Prof. Allan Credo for reviewing this thesis, and we am gratefully indebted to their very valuable comments on this thesis.

Finally, we would like to express our very profound gratitude to our parents, friends, and to our classmates for providing us with unfailing support and continuous encouragement throughout the entire study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

# Introduction

## Project Context

Music has always been an integral part of human culture. It is capable of breaking boundaries to unite people from different background & cultural heritage. In fact, music can best be described as wonderful force that is capable of bonding people together and instituting international brotherhood, love as well as peace [1]. For centuries, music has been shared and remembered by two traditions: aural transmission and in the form of written documents normally called musical scores. Many of these scores exist in the form of unpublished manuscripts and hence they are in danger of being lost through the normal ravages of time. To preserve the music requires some form of typesetting, ideally, a computer system that can automatically decode the symbolic images and create new scores. The OMR is needed for the preservation of these works which requires digitization and should be transformed into a machine readable format. Such a method is one of the most promising tools to preserve the music scores. In addition, it makes the search, retrieval and analysis of the music sheet easier. Generally, such a task is challenging because it requires the integration of techniques from some quite different areas, i.e., computer vision, artificial intelligence, machine learning, and music theory.

## Purpose & Description

The purpose of this study is to use multiple neural networks that are most suitable for Optical Music Recognition to classify handwritten musical notations. More importantly, our study can be used by other researchers who want to extend our research or improve our methods used in this research process.

What makes this study unique is that there are only a few researchers studying in this area. Additionally, this paper is unique from other researches since they only tackle handwritten musical notations that are written directly on electronic devices (iPad, tablet, etc.) and not on physical paper.

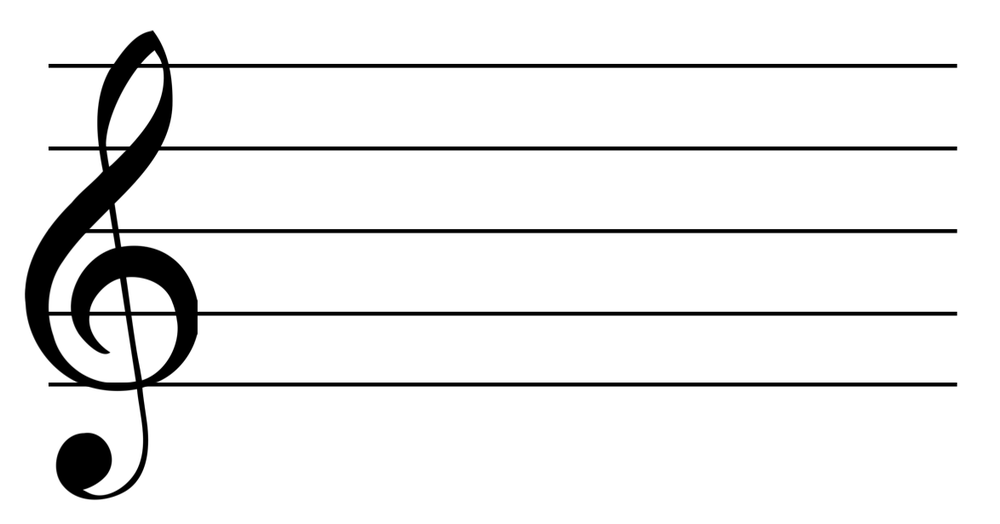
## Objectives

This study aims to determine the accuracy of multiple neural networks as classifier in Optical Music Recognition (OMR) handwritten musical symbols as its parameters.

## Scope & Limitation

### 1.4.1 Scope

1. Staff lines, Time Signature, Key Signature & Treble Clef (also known as G-clef), bar lines on a paper must be provided to serve as a template to write on.
2. This study will focus on converting handwritten musical score for piano.
3. Listed below is the given templates shown in *Figure 1*:



*Figure 1: Template with cleff*

1. The following musical notations that are used in the study:
   * common accidentals
     + flat (♭)
     + sharp (♯)
   * musical symbols:
     + notes
       -  (whole note)
       -  (half note)
       - ♩ (quarter note)
       - ♪ (eighth note)

### 

### 1.4.2 Limitation

The following are not included in the study:

* + 1. Strings:
  1. Title
  2. Date
  3. Lyrics
  4. Name of the author/s
     1. False beats in a measure.

## Significance of the Study

Although there have been many researches on OMR, not much has focused on handwritten musical scores. This study can be a great contribution to the research field of both machine learning and image processing. This study can serve as an additional stepping stone or advancement to achieving a more accurate and efficient OMR on handwritten musical score.

## Definition of Terms

The following are the musical terms that has been mentioned in this paper:

OMR – *Optical music recognition*, a research field specifically aimed to automatically recognize music symbols.

MLP – *Multilayer Perceptron*, feed forward neural network with one or more layers between input and output layer

CNN – *Combined Neural Network,* a technique in artificial intelligence to combine one or more neural networks together.

Music

**Bar Line** - this separates measures.

**Key Signature** - is a set of sharp or flat symbols placed together on the staff.

**Musical Score** - a written form of a musical composition.

**Note/s** a) A sign used in musical notation to represent the relative duration and pitch of a sound (♪, ♫).

b) A pitched sound itself

**Staff** - A staff is made up of five horizontal lines and four spaces. The height of a note on the page determines its pitch. It is standard to draw 5 lines and they are called *staff* or *stave*.

# Review of Related Literature

In almost all previous researches in the field of both handwritten and printed OMR, four major steps were commonly discussed: image preprocessing, segmentation, object classification, reconstruction. For each of the steps mentioned, various algorithms are discussed to perform the respective task. In the image preprocessing stage, various sub-steps are taken: binarization, noise reduction, orientation correction or deskewing. The new image produced by the preprocessing stage will be used for the next stage: segmentation. Prior to the classification stage, several segmentations are done in preparation. First, the staff lines are detected and removed/isolated from the musical score. Second, the symbols are segmented from each other. Last, although this can be intrinsically intertwined with the previous, each symbol is segmented by its primitive features (head, stem, tail, etc). Following the segmentations is the classification. Each of the segmented parts will be classified accordingly. Finally, after all the necessary processes, all the data produced will be reconstructed and represented. [2]

In a comparative study by Burgoyne [3], et al, they stated that nearly all OMR systems require converting the image into binary form (i.e. black and white). In this step, the image is analyzed to determine which is useful (symbols, staff lines, etc.) and which is the background. For the binarization to be executed automatically, several algorithms has been proposed. In a separate study [4] by Pavlos Stathi, et al, they categorized these algorithms as global (Otsu, Histogram, K-means, etc.) and local (SOM neural network, etc.). Global algorithms apply one particular threshold to the entire image. Whereas the local algorithm, on the other hand, selects a specific threshold to every pixel relative to its surrounding pixels. They also stated that the listed global algorithms are faster compared to local.

In paper written by Dutta [5], et al, they argued that the input image will always be skewed. The nature of scanned and photographed input image is that it is never perfectly level. To address this problem, they applied the RLE or run-length encoding method which well-known for its simplicity in which runs on data represented as a single data value and count. The RLE uses the one and zero values of the binary image as input for the recognition process. Thus, the run-length coding is even more compact it requires only the lengths on the run.

In a paper written by Rebelo and his co-authors [6], is a comparative study of OMR. In the OMRs that they have studied, the most usual approach to symbol segmentation is a hierarchal decomposition of the music image wherein the pre-analyzed musical score is split by staffs, followed by extraction of the symbol features: noteheads, rests, dots, stems, flags, etc. However, in some research, the segmentation process is approached differently. After isolating the symbols from the staff lines, instead of segmenting the symbols by their elementary features, the features (noteheads, dots, stems, flags, etc.) are left unsegmented and are will be classified as a whole.

Prior to removal of staff line, detection is required. In a project of Sheridan and George [7], they used horizontal projection to detect the stave lines. In their paper, they discussed that the staff lines are not exactly parallel, horizontal, equidistant, of constant thickness, or even straight. To cope with this problem, they divided the scores into smaller parts and the stave lines from each part’s beginning and end. And these parts are later assembled back together after their identification.

In a paper written by Kwon and Kim [8], they proposed a way to use mobile device to recognize music sheet image and extracting necessary data for a playable sheet of music. Kwon and Kim [8] said, up to present day the methods to recognize the music sheet are line tracking, Hough transform, DP matching, and template matching but, problem caused by processing time and memory limitation, it is not appropriate enough to run on mobile products. There is advantage of responding fast in value of the staff line that taken from horizontal histogram is relatively high. So, based on that advantage found by Kwon and Kim [8] started their research. In pre-processing, this stage is important for computer to recognize the image that taken from camera. So first image will be processed to change into black and white image by applying this formula (*Value = 0.2126* × *R* + *0.7151* × *G* + *0.04721* × *B)* to R, G, B these three sections to change in one gray value. Next step is extraction of staff lines, to recognize the note head of the musical note and its location, removing of staff lines must be performed. For removing of staff lines, they used horizontal histogram to find out the staff lines. In recognition of the musical symbols, they used pixel searching and vertical histogram. And to determine the beat of the notes, the tail of the note must be recognized for example, a note with a tail is eighth note and a note with two tails is the sixteenth note. So, it will be determined through numbers encountered of the black pixel group as stem searching is progressed. The experiment was performed in Visual C++ and experiment image was Twinkle Twinkle Little Star, both left hand and right hand notes were recognized in 100%.

In a comparative study conducted by C. Dalitz, M. Droettboom, B. Pranzas, & I. Fujinaga [9], et al, they mentioned four techniques of staff line removal: line tracking, vector field, runlength, skeletonization. Although their study was not conclusive in finding the best of the list, they found that skeletonization was robust with respect to some defects. However, they also stated that skeletonization is the most susceptible to the typeset emulation of historic prints and does not perform significantly better on the undeformed test set.

In a paper authored by Koh and Lee [10], they developed a system for printed musical scores. In their system, unlike in some studies, removal of stafflines is executed. To remove the staffline, detection is needed. In their system, they detected the stafflines by using the amount of gap space and the thickness of the staff lines. After the stafflines are detected, they remove the staff lines to expand or reduce the pattern images and labelling them to find the musical symbols and its stem.

Contrary to the trend of removing staff line, Scott Sheridan and Susan E. George [11] argued that the method of removing the stave lines has major drawbacks. They stated that although historically and presently the first step in locating musical objects has been to remove the stave lines from the image isolating the remaining shapes for identification, the stave line removal still results in a fragmented musical objects even with the best methods developed so far. They have stated that one of the reasons that the staff lines were removed in the dawn of OMR was to help the primitive software and hardware locate the musical

In a paper authored by David Bainbridge and Tim Bell [12], they have shown that one method of dealing with fragmented objects is to devise a heuristic that merges Bounding Boxes together by comparing adjacent bounding boxes in the vicinity of stave lines. With bounding box analysis, the finding the right threshold is essential. With proper calculation of the threshold, the segmented parts of a symbols caused by the removal of the stafflines can be grouped together correctly as one symbol.

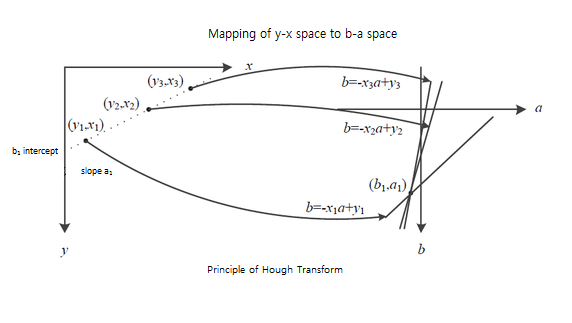
In a dissertation of Dar-Shyang Lee[13], he proposed a theory of classifier combination using the neural network approach. The classifier that he developed is basically a group of classifiers that perform independently whose results are then used to make a collective decision done by a combinator. Since the combinator makes the final decision, he argued that a robust combinator must have the ability to combine and extract useful information from any types of classifiers and any mixture of type. In developing the combinator, three important properties were considered: the ability to eliminate redundant classifiers, the ability to provide non-classifier-specific constraints to model complexity, and lastly, the ability to recognize classifier dependencies on image and pattern characteristics and that information to achieve a better performance. And the basic structure of the combinator that he proposed is a multilayer perceptron that takes all classifier output vectors as input. By nature of the classifier output transformation and the network learning rule, this simple architecture combines any type of classifier and any mixture of types. To prevent performance degradation from correlated classifiers, least-square along with backpropagation is used.

The final and fundamental task of OMR is to convert the semantically recognized scores to coding format. This coding format must have the intended ability to model and store music information. According to Novotny and Pokorny [2], in their paper regarding the overview and practical challenges of OMS, they argued that the best known formats are: MIDI (Musical Instrument Digital Interface), NIFF (Notation Information File Format), SMDL (Standard Music Description Language) and MusicXML MIDI is mainly used as an interchange format between digital instruments and computers. Although its capability of modeling music scores is very limited (e.g. the relationships among symbols cannot be represented), most of the music editors can operate with MIDI files.

# Technical Background

### 3.1 Hough Line Transform

Hough transform is to detect the line in an image and which method is commonly used in image processing and this method can be implemented in OpenCV.



*Figure 2:Basic Principle of Hough Transform*

*Figure 2* above shows us the basic principle of Hough transform. The left side x-y coordinate plane is commonly used in our mathematics class. In x-y plane has three points (x1, y1), (x2, y2), (x3, y3) given and the goal is to draw a line by the computer if these three points are forming a straight line. For example in x-y plane in *Figure 2*, we usually use y=ax + b formula to get a line and now, we are saying about the point (xi, yi) and the formula will be yi = axi + b then let’s think about this formula b = -xi a+ yi, if this will be the formula there will be no longer in x-y plane. So, in a line there are a lot of values after passing on a point to another but in a-b plane, it was expressed in one line. Now we can say about the conclusion that if the three points in x-y plane (x1, y1), (x2, y2), (x3, y3) will be drawn on a-b plane as right side of *Figure 2*, three lines intersects on a point and this proves that three points are lying on a same line.

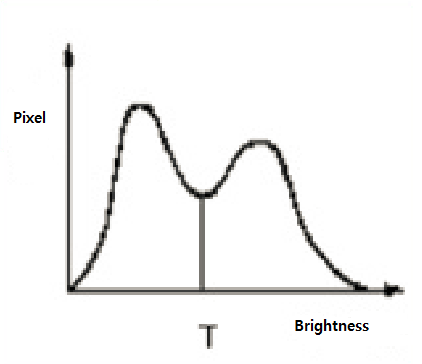
There are many advantages of the Hough transform. It is robust to partial or slightly deformed shaped, it is tolerant to noise, and it can find multiple occurrences of a shape during the same processing pass. While the disadvantage of this method is that it has substantial computational and storage requirement which become acute when object orientation and scale have to be considered.

### Global Thresholding

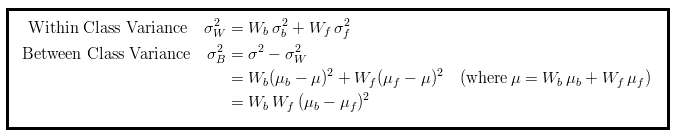
The algorithm that variant data threshold by using math model. The algorithm is used when the intensity of objects and background pixels are sufficiently distinct. In application the algorithm will estimate threshold value automatically. Each image has its own threshold value.

* Otsu’s Method

Otsu's thresholding method involves iterating through all the possible threshold values and calculating a measure of spread for the pixel levels each side of the threshold, i.e. the pixels that either fall in foreground or background and the fall called valley which is the optimal threshold value. The aim is to find the threshold value where the sum of foreground and background spreads is at its minimum.



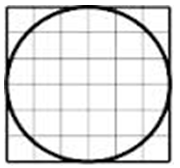
*Figure 3: Otsu’s Method*

In *Figure 3*, T is the threshold. So, to find threshold Otsu recommended a statistical method, within-class variance and between-class variance in Equation 1. When the threshold with the maximum between-class variance also has the minimum within-class variance, that’s the time the best threshold can be found. 

*Equation 1: Otsu's Formula*

### Bounding Box Algorithm

The bounding box also called the minimum or smallest enclosing box is a term used in geometry, for a point set ‘S’ and ‘N’ dimensions which refers to the box with the smallest measure like area, volume, or hypervolume in higher dimensions within which all the points lie. *Figure 6* is the example of the bounding box for a circle.

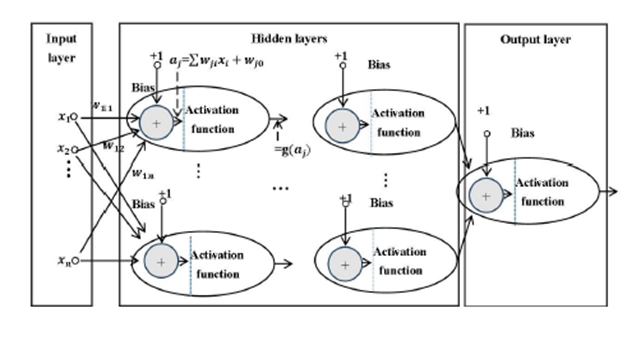


*Figure 4: Bounding box for a circle*

Having formally defined the bounding box, extent can be simply defined as the ratio of the area enclosed by the shape under test and the area covered by the bounding box. So, the formula looks like this: *E=AreaOfCircle/AreaOfBoundingBox.* Solving further, E= (πr2/2r) ×2r then, E =π/4=0.7854.

### 3.4 Multi-Layer Perception

The MLP inside each of the three Neural Networks in *Figure 22(see page 28)* is introduced in the figure below. It is a type of feed forward neural network that have been used in pattern recognition problems. The network is composed of layers consisting of various number of units. Units in adjacent layers are connected through links whose associated weights determine the contribution of units on one end to the overall activation of units on the other end. There are generally three types of layers. Units in the input layer bear much resemblance to the sensory units in a classical perceptron. Each of them is connected to a component in the input vector. The output layer represents different classes of patterns. Arbitrarily many hidden layers may be used depending on the desired complexity. Shown by *Figure 7* is the MLP structure in each neural network.



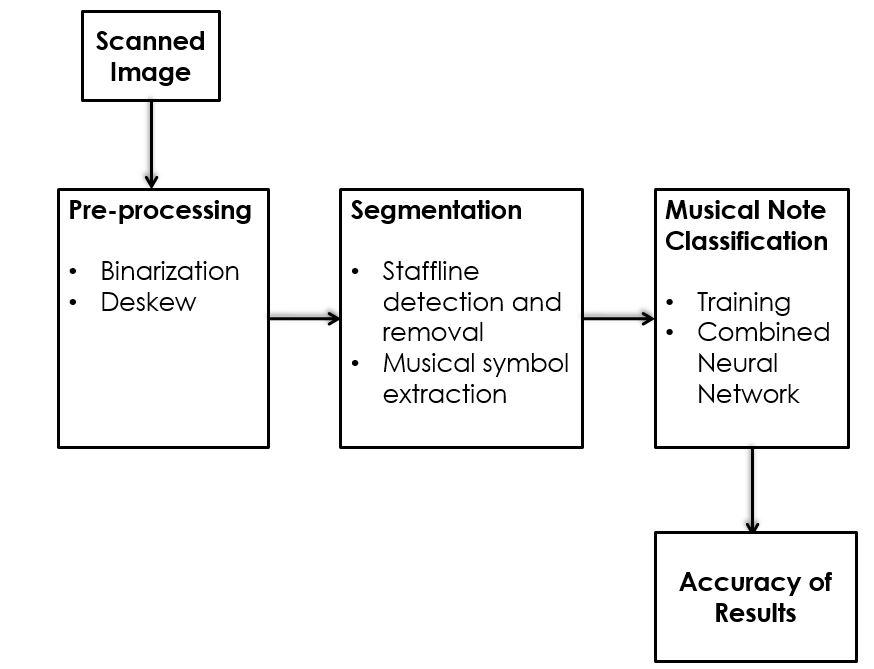
*Figure SEQ Figure \\* ARABIC 7: The MLP Structure in each neural network*

# Methodology

## 4.1 General Framework

Automatic recognition of music scores is a complex task affecting many areas of computer science (e.g. Image Processing, Machine Learning, Neural Network, and etc.) , though different OMR systems use different strategies and algorithms put together to solve certain problems concerning about decomposing the problem. Our framework for the automatic recognition of musical notation encompasses four main stages. *See Figure 8: Framework Model below.*

1. Image preprocessing
2. Segmentation
3. Musical Note Classification



*Figure 8: Framework Model*

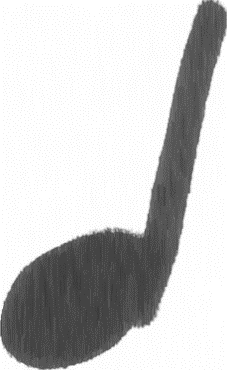
## 4.2 Image Preprocessing

The goal in preprocessing stage is to make the scanned image clearer and clean, this phase is to prepare the image in order to make the recognition process more robust, efficient, and reduce errors that might occur in the next stages of the process.

### 4.2.1 Binarization

The Binarization process will convert image into binary form. Meaning, there are two possible values for each pixel which is black & white.

First, the images are binarized with the Otsu threshold method algorithm (seen in section 3.2) and after that the image will turn from greyscale to binarized image (*Figure 9*). From here it is much easier to design an algorithm for staff detection, segmentation and recognition in binary images than in grayscale images or colored ones [14].

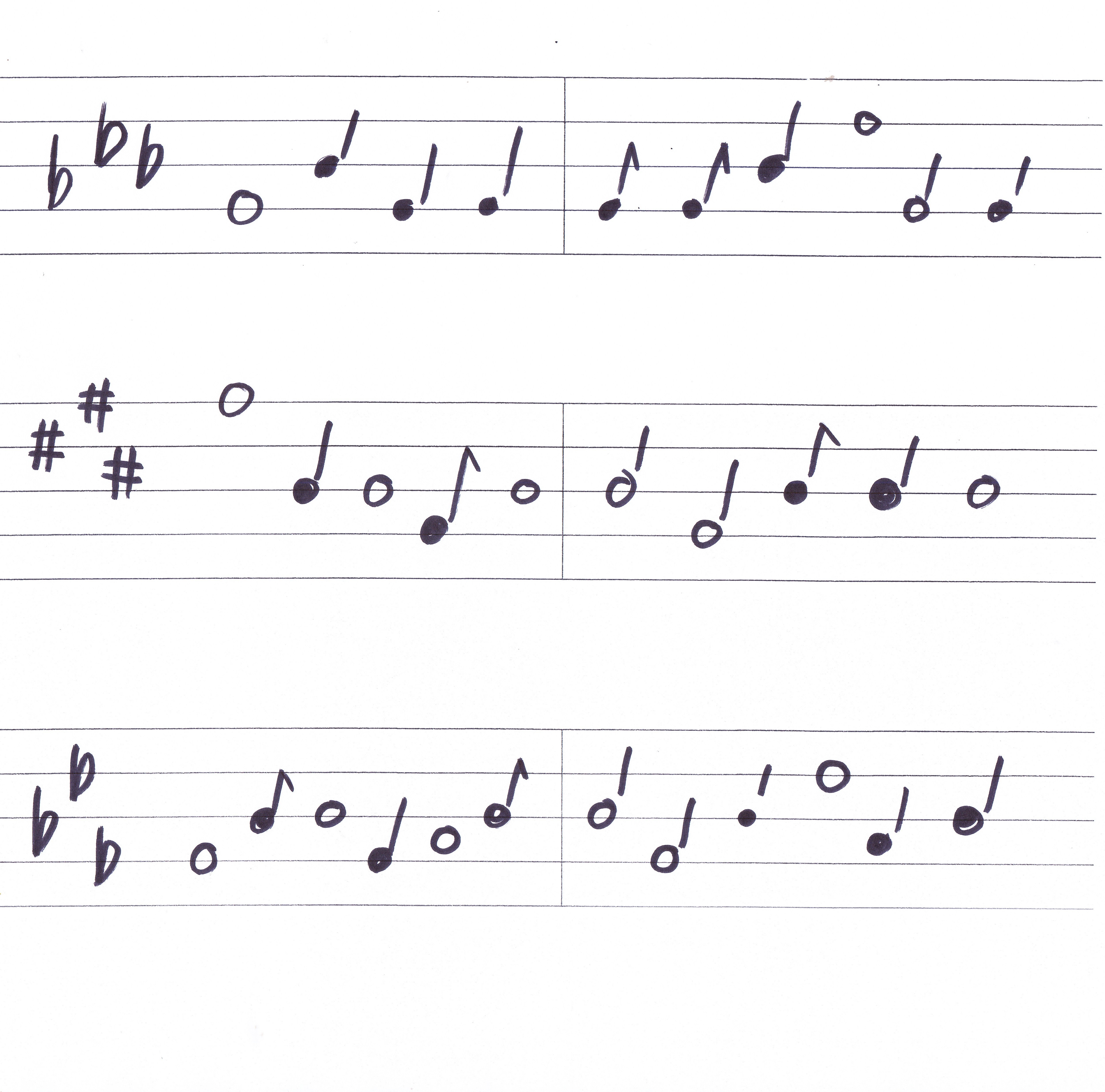


### 

Figure 9: scanned image to binary image

### 4.2.2 Deskewing

Since there is no guarantee that the scanned image will be perfectly horizontal, we will be applying a deskewing technique. *Figure 10* and *Figure 11* shows deskewing an image.



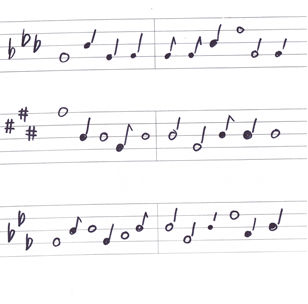


Figure 10: original scanned image Figure 11: deskewed image

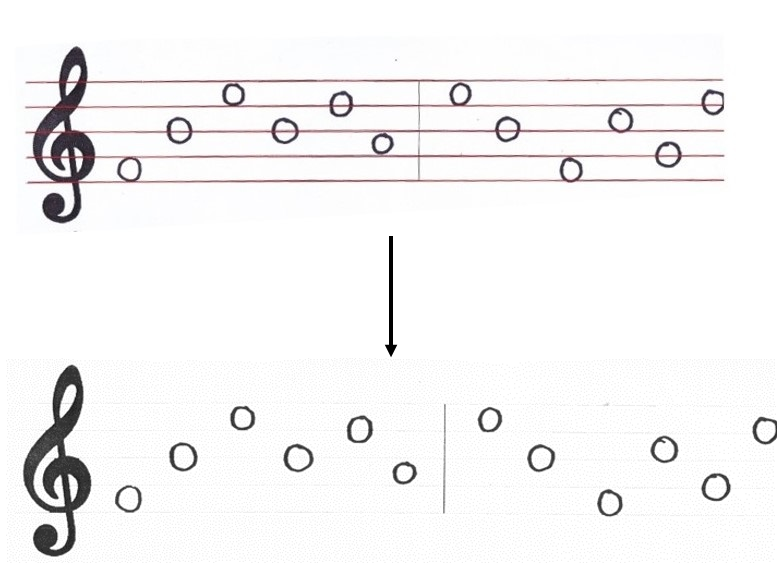
In this process we’ll be using *Hough Transform* algorithm for calculating the skew angle of the image and find the nearest angle that is closest to the original scanned image. In *Figure 10,* clearly the lines are not straight, so we identify first the angle of the scanned image by using *Hough Transform* which is 188°, so to fix this we want to determine the unknown variable *x* or the distance angle between the original angle and the straight line which is 180° then subtract both, that is 8° difference. At this point, we already found *x* or technically the *arctan*, so we just have to rotate the image to 8° to have a straight horizontal line in the image.

## 4.3 Segmentation

Segmentation is an important step before any shape recognition, the segmentation process consists of locating and isolating the musical objects in order to identify them. First, we find the staff line in the input image. Second, we remove the staff line resulting into an image containing only musical symbols.

### 4.3.1 Staff Line Detection & Removal

This part describes the problems involved in the detection and removal of staff lines of musical scores. The removal process is an important step for many Optical Music Recognition systems because it facilitates the segmentation and recognition of musical symbols. The process is complicated by the fact that most music symbols are written on top of the staff lines and these lines are neither straight nor parallel to one another. The challenge here is to remove staff lines as much as possible while maintaining to preserve the shapes of the musical symbols [15].

Again we apply Hough line transform to determine horizontal lines in the image or staff lines and once those lines are determined we select those lines & apply erosion to remove the staff lines. For visual purposes we highlighted the lines with red to show that Hough line transform identified horizontal lines in the image before applying erosion. The process is shown in the *Figure 12* below.

*Figure 12: staffline removal*

Before proceeding to the next step we noticed that there is a side effect which leaves a straight line on the symbol caused the erosion process, so we have to apply dilation to fix those symbols as shown in *Figure 13* below.

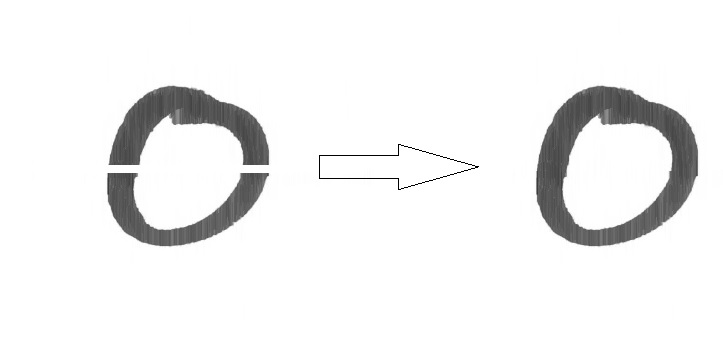
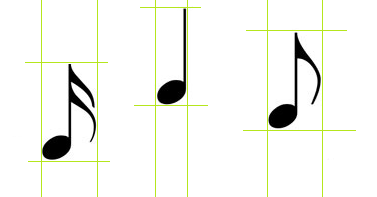


Figure 3: applying dilation

### 4.3.4 Musical Symbol Segmentation

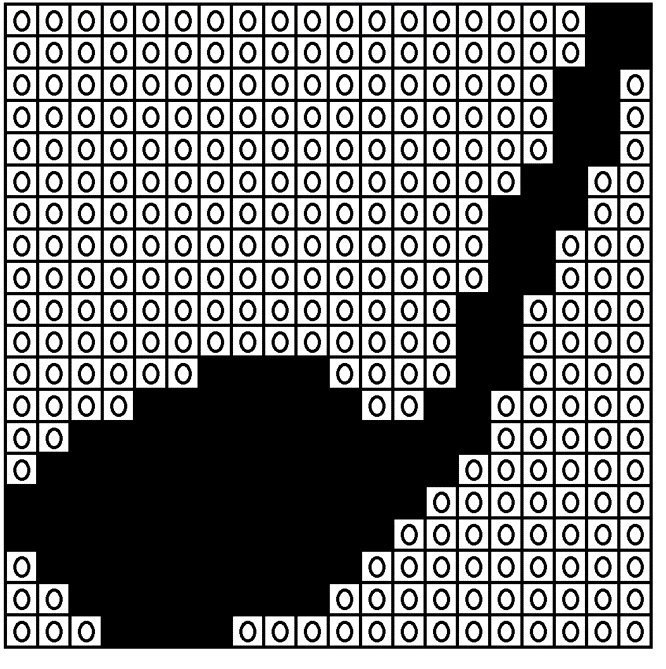
The extraction of musical symbols is the operation following the staff line detection and removal. The extraction process is basically isolating the objects of interest from each other in preparation for classification. To extract the objects of interest, we will be using bounding box analysis. With bounding box, vertical and horizontal projection will be used to identify the location of the object’s first and the last pixel. A simple visualization of this process is in *Figure 14.*

*Figure 14: Musical notes in the bounding box*

In *Figure 14,* the histogram determines the boundaries of each symbol by their first and last pixels vertically and horizontally. Since the boundaries of each symbol has been established, the musical symbols is now isolated from each other and each isolated image will be saved in a 20x20 image.

## 4.5 Musical Note Classification

Since the musical notes has been isolated from each other, individual classification of the musical notes can now applied. For the classification of individual notes, we will be counting individual pixels by iterating through a 20x20 input image of the note and counting each black pixels from top to bottom. For each iteration, if a white pixel is detected our algorithm will output a binary value 0 and 1 for each black encountered. After all notes has been processed, each note or each data is created in form of csv for our training. An abstract illustration shown in *Figure 15.*

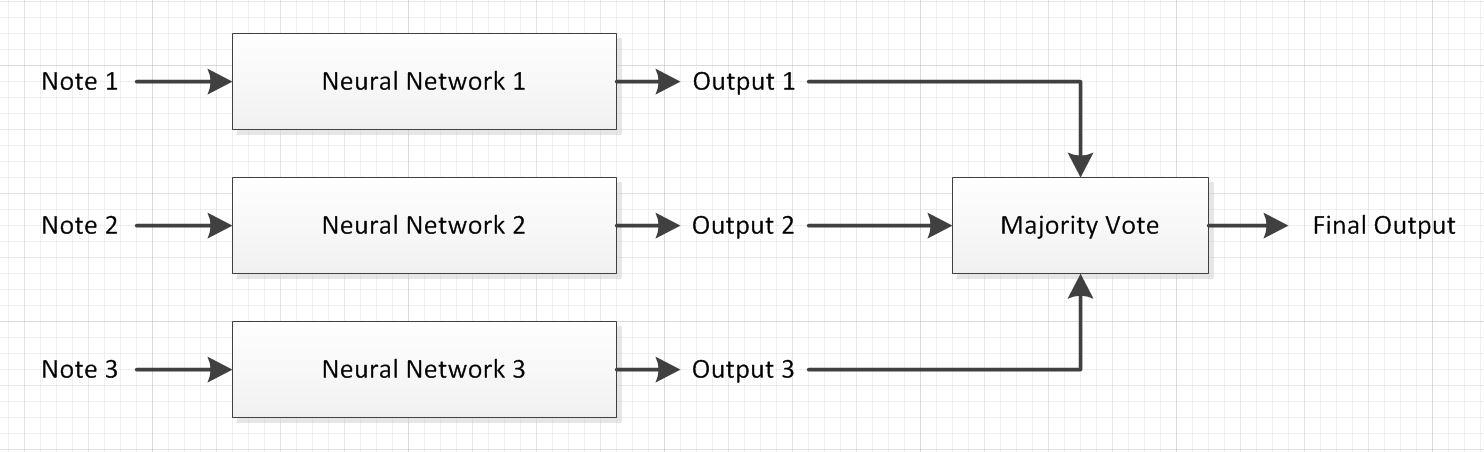


*Figure 15: pixel counting*

After the process, we now have the representation of each note in a form of csv file where all of the coordinates from each notes are located.

## 4.6 Training

The theory of classifier combination of Neural Network as discussed in [13]. Our CNN (*combined neural network*) is based on the theory of D. Lee [13], which is to combine decision of individual classifiers to obtain a better output results, though this technique is also known as MLP *(Multilayer Perceptron)*. For the training each iteration of one note will serve as our input and afterwards the output will be forwarded to the output layer for majority vote, afterwards the training will stop until the error rate is acceptable or above 60%. To make this task more clearly defined and subsequent discussions easier, here we describe the architecture of CNN in *Figure 16.*



*Figure 16: CNN structure*

# 5. Results and Evaluation

Two types of dataset were used to measure the accuracy of the classifier. The first one is a collection of scanned images of musical scores written by persons with none to minimal experience in music writing and the other is a collection of scanned images of musical scores written by persons with more experience in music writing (particularly music major students and composers). From this point forward, we will be calling these datasets as dataSet\_Inexperienced and dataset\_Experienced, respectively.

The dataSet\_Inexperienced consists of 20 music sheets written by 20 persons. While in dataset\_Experienced 10 music sheets were written by 10 persons. Each of the music sheet has 3 staves.

Before the test was performed, the total numbers of notes of each music sheet from both datasets were manually counted: dataSet\_Inexperienced contains a total of 831 notes/symbols while dataset\_Experienced contains a total of 339 notes/symbols. Further details of the tally are shown in Table 1.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Data Set | MANUALLY COUNTED NOTES | | | | | | TOTAL |
| Whole | Half | Quarter | Eighth | Sharp | Flat |
| Inexperienced | 140 | 146 | 132 | 133 | 154 | 126 | 831 |
| Experienced | 30 | 74 | 101 | 72 | 41 | 21 | 339 |
| TOTAL | 170 | 220 | 233 | 205 | 195 | 147 | 1170 |

***Table 1: Number of notes/symbols in the input images***

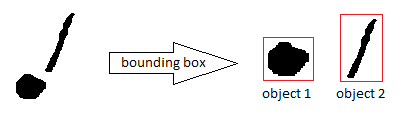
After we have determined the total number of expected notes/symbol, we fed the images into our preprocessing algorithms. Afterwards, we proceeded to note/symbol extraction using bounding box.

With the bounding box method, a total of 840 objects (notes/symbols and fragments/noise) were extracted from all music sheets in dataSet\_Inexperienced. While a total of 440 objects were extracted from all music sheets of dataSet\_Experienced. More details of the extraction are shown in Table 2 below.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DATA SET | OBJECTS EXTRACTED USING BOUNDING BOX | | | | | | | TOTAL |
| Whole | Half | Quarter | Eighth | Sharp | Flat | Noise/  fragments |
| Inexperienced | 140 | 129 | 132 | 131 | 153 | 138 | 17 | 840 |
| Experienced | 30 | 33 | 47 | 43 | 41 | 21 | 228 | 443 |
| TOTAL | 170 | 162 | 179 | 174 | 194 | 159 | 245 | 1283 |

***Table 2: Objects extracted using bounding box***

It is clear that there is a significant number of noise/fragments (objects that are considered unacceptable as notes when classified manually) especially in the dataSet\_Experienced while the number of notes has decreased. This is because most experienced writers tend to write symbols/notes that are fragmented which the bounding box treats as separate objects. Some of the notes, especially the half notes, quarter notes, eighth notes are fragmented into unclassifiable objects, thus, are considered as noise. An example of the drawback of our bounding box is illustrated in *Figure 17*.



*Figure 17: Bounding box drawback*

Due to this drawback, in the bounding box process, only 33 out of 74 half notes, 47 out of 101 quarter notes, and 43 out of 72 eighth notes were selected and classified manually as acceptable. Even before the classifying process, the accuracy of the system when applied on dataSet\_Experienced is expected to be less than 50% due to this drawback alone.

After all the necessary preprocess and note extraction processes has been applied, we fed each note to the classifiers.

Table 3 shows the confusion matrix of the classifier done on dataSet\_Inexperienced. It is clear in the table that 134 of the whole notes were correctly classified while 2 were incorrectly classified as sharp and 4 as flat. For the Half note, 118 were correctly classified while 3 were incorrectly classified as whole, 2 as quarter, 3 as eighth, 1 as sharp, and 2 were not classified due to a tie in the majority voting. It also shows that 112 quarter notes were classified correctly while 6 were classified incorrectly as half, 13 as eighth and 1 as no classification. And out of 131 eighth notes, 124 were correctly classified, while 6 weren’t. The classifier also performed well in classifying sharp symbols where it successfully classified 149 out of 153. And lastly, for the flat symbol, it was able to successfully classify 126 while 12 were incorrectly classified.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NOTES | DATASET\_INEXPERIENCED CLASSIFICATION | | | | | | | |
| Whole | Half | Quarter | Eighth | Sharp | Flat | NoClass | Percentage |
| Whole | 134 | 0 | 0 | 0 | 2 | 4 | 0 | 95.71% |
| Half | 3 | 118 | 2 | 3 | 1 | 0 | 2 | 91.47% |
| Quarter | 0 | 6 | 112 | 13 | 0 | 0 | 1 | 84.85% |
| Eighth | 0 | 0 | 6 | 124 | 0 | 0 | 1 | 94.66% |
| Sharp | 0 | 0 | 0 | 0 | 149 | 4 | 0 | 97.39% |
| Flat | 3 | 7 | 0 | 0 | 2 | 126 | 0 | 91.30% |

***Table 3: Confusion matrix on dataSet\_Inexperienced***

The classifier was able achieve an accuracy of 95.71% on whole notes, 91.47% on half notes, 84.85% on quarter notes, 94.66% on eighth notes, 97.39% on sharp symbols, and 91.30% on flat symbols.

Among the notes/symbols that were tested, quarter notes were the least correctly classified with an accuracy of 84.85%. Most of the half notes were incorrectly classified as quarter notes and eight notes. While majority of the quarter notes were incorrectly classified as half notes and eight notes. Whereas most of the eighth notes were incorrectly classified as half notes and quarter notes. The classifier committed these errors in distinguishing them from each other properly because of their similarities in shape and features (head, stem, and tail).

To calculate for the accuracy with respect to the correctly boxed notes, this formula was followed:

And the accuracy of the classifier with respect to correctly boxed notes on dataSet\_Inexperienced is 92.71%.

Additionally, getting the accuracy of the classifier with respect to the manually counted notes, the formula below was followed.

Where:

*Correct w = number of correctly identified whole note*

*Correct h = number of correctly identified half note*

*Correct q = number of correctly identified quarter note*

*Correct e = number of correctly identified eighth note*

*Correct s = number of correctly identified sharp note*

*Correct f = number of correctly identified flat note*

In Table 4, the confusion matrix of the classifier on dataSet\_Experienced is shown. It shows that out of 30 whole notes extracted, 26 were correctly classified, while 3 were incorrectly classified as sharp, and 1 as flat. For the half note, it classified 25 correctly, while classifying 7 as quarter and 1 as flat. Thirty-three (33) quarter notes were also classified correctly. And for the sharp and flat symbols, it has performed perfectly. All 41 sharp and 21 flat symbols were classified correctly.

And the calculated accuracy of the classifier with respect to the manually counted notes on dataSet\_Inexperienced is 91.82%.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NOTES | DATASET\_EXPERIENCE CLASSIFICATION | | | | | | | |
| Whole | Half | Quarter | Eighth | Sharp | Flat | NoClass | Percentage |
| Whole | 26 | 0 | 0 | 0 | 3 | 1 | 0 | 86.67% |
| Half | 0 | 25 | 7 | 0 | 0 | 1 | 0 | 75.76% |
| Quarter | 0 | 1 | 33 | 4 | 2 | 5 | 2 | 70.21% |
| Eighth | 2 | 2 | 0 | 37 | 0 | 1 | 1 | 86.05% |
| Sharp | 0 | 0 | 0 | 0 | 41 | 0 | 0 | 100% |
| Flat | 0 | 0 | 0 | 0 | 0 | 21 | 0 | 100% |

***Table 4: Confusion matrix on dataSet\_Experience***

And the accuracy of the classifier with respect to correctly boxed notes on dataSet\_Experienced is 85.12%. While the accuracy of the classifier with respect to the manually counted notes on dataSet\_Exerprienced is 53.98%.

To calculate the overall accuracy of the classifier with respect to the correctly boxed notes, the formula is followed.

Where:

*Correct i = number of correctly identified notes/symbols in dataSet\_Inexperienced*

*Correct e = number of correctly identified notes/symbols in dataSet\_Experienced*

*Notes i = number of correctly boxed notes in dataSet\_Inexperienced*

*Notes e = number of correctly boxed notes in dataSet\_Inexperienced*

And the overall accuracy of the classifier with respect to the correctly boxed notes is 91.14%.

And to calculate the overall accuracy of the classifier with respect to the manually counted notes, the formula is followed.

Where:

*Correcti = number of correctly identified notes/symbols in dataSet\_Inexperienced*

*Correct e = number of correctly identified notes/symbols in dataSet\_Experienced*

*Notes i = number of manually counted notes in dataSet\_Inexperienced*

*Notes e = number of manually counted notes in dataSet\_Inexperienced*

Thus, the overall accuracy of the combined neural network with respect to the manually counted notes on both dataSet\_Inexperienced and dataSet\_Experienced is 80.85%.

# 6. Conclusion

In our study, we demonstrated an approach to classify musical notes/symbols (whole note, half note, quarter note, eighth note, sharp, and flat) using multiple neural network combined by majority voting. We applied erosion and dilation to remove the staff lines, and used bounding box method to extract each of the notes/symbols. In our multiple neural network, we used three (3) neural networks to collectively classify each note/symbol. Each note was resized to 20x20, 20x50, and 50x20. The first neural network was trained with the 20x20 notes, the second was fed with 20x50, and the last neural network was fed with 50x20. The first neural network had 9 hidden layers, while the second had 11 hidden layers, and the last had 9 hidden layers. The three neural networks, which are Multi-Layer Perceptrons, were trained with a learning rate of 0.3 and with 500 epochs.

In testing the accuracy of our combined neural network, we asked 30 participants to write the six (6) types of notes/symbols mentioned in our scope. Twenty (20) of the participants had no experience in music writing whereas the other ten (10) have more experienced (particularly music major students and composers). The twenty (20) inexperienced participants wrote a total of 140 whole notes, 146 half notes, 132 quarter notes, 133 eighth notes, 154 sharp symbols, and 126 flat symbols. Whereas the ten experienced participants wrote a total of 30 whole notes, 74 half notes, 101 quarter notes, 72 eighth notes, 41 sharp symbols, and 21 flat symbols. In the inexperienced data set with total notes of 831, our combined neural network was able to correctly classify 763 notes with an accuracy of 91.82% with respect to the manually counted notes, and an accuracy of 92.71 with respect to the correctly boxed notes. While in the data set of the experienced writers, with a total of 339 notes, our classifier was able to correctly identify 183 notes with an accuracy of 53.98% with respect to the manually counted notes, and an accuracy of 85.12% with respect to the properly boxed notes.

Therefore, we conclude that using multiple neural networks is highly favorable in classifying handwritten musical notes. It was able to classify with a high success rate on notes written by both experienced and inexperienced writers. However, the only major drawback was in our image pre-processing methods which affected our classifier’s accuracy since some of the notes (especially notes that fragmented) were not extracted properly.

## 6.1 Recommendation

One of the challenges that the researchers faced was in the staff line removal stage and in the extraction of notes.

The researchers would recommend to find another algorithm for removing the staff lines rather than erosion and dilation because it does not only erode the lines but it also erodes parts of the notes. Additionally, erosion and dilation requires the lines to be significantly thinner than the notes, otherwise the notes will be eroded together with the staff line.

The researchers would also recommend adding another method for extracting the notes. The bounding box method alone is not ideal in extracting the notes especially when the notes are fragmented. When the head and stem of the note are not properly connected, the bounding box method extracts the head and stem as separate objects. Algorithms such as nearest neighbor would be a great addition to the bounding box method since it can help group the note fragments accordingly.

Lastly, the researchers would recommend adding more features to feed into neural network classifier to obtain higher rate of accuracy. Algorithm that can detect the head, stem, tail, and other notes features would be a great addition.

# References

[1] G. Rachiotis, “The Importance of Music in Our Daily Lives.” *Truth inside of You*, 2014

[2] J. Novotny, & J. Pokorny, “Introduction to Optical Music Recognition: Overview and Practical Challenges.” 2015.

[3] A. Burgoyne, L. Pugin, G. Eustace, & I. Fujinaga, “A comparative survey of image binarisation algorithms for optical recognition on degraded musical sources.” *In Proceedings of the 8th International Society for Music Information Retrieval*, 2007.

[4] W. J. Kim, M. J. Chung, & Z. Bien, “Recognition system for a printed music score.” 1978.

[5] A. Dutta, U. Pal, A. Fornes, & J. Llados, “An efficient staff removal approach fromprinted musical documents.” *In Proceedings of the 20th International Conference on Pattern recognition,* pages 1965–1968. IEEE Computer Society, 2010.

[6] A. Rebelo, G. Capela, & J. S. Cardoso, “Optical recognition of music symbols: A Comparative study.” *International Journal on Document Analysis and Recognition*, 2010.

[7] H. Miyao, “Stave Extraction for Printed Music Scores.” 1992.

[8] T. Kwon and H. Kim, "Music Score recognition using pixel searching and histogram.” 2008

[9] C. Dalitz, M. Droettboom, B. Pranzas, & I. Fujinaga, “A Comparative Study of Staff Removal Algorithms.” *IEEE TPAMI*, 2008.

[10] M. Szwoch, “Using MusicXML to Evaluate Accuracy of OMR Systems,” In Stapleton, G., Howse, J., Lee, J. (eds.) Diagrammatic Representation and Inference, Lecture Notes in Computer Science, vol. 5223, pp. 419–422. Springer Berlin Heidelberg, 2008.

[11] S. Sheridan, & S. E. George, “Defacing Music Scores for Improved Recognition.” *School of Computer and Information Science University of South Australia*, 2004.

[12] D. Bainbridge & T. Bell, “The Challenge of Optical Music Recognition.” 2001.

[13] D. S. Lee, “A Theory of Classifier Combination: The Neural Network Approach.” *State University of New York*, 1995.

[14] P. Stathis, E. Kavallieratou, & N. Papamarkos, “An evaluation technique for binarization algorithms.” *Journal of Universal Computer Science*, vol. 14, no. 18, 2008.

[15] G. Hamarneh, K. Althoff, & R. Abu-Gharbieh, “Automatic Line Detection.” September 1999.

[16] U. Pal, A. Fornes, & J. Llados, “An efficient staff removal approach from printed musical documents.” *In Proceedings of the 20th International Conference on Pattern recognition*, pages 1965–1968. IEEE Computer Society, 2010.

[17] A. Aoyama, & H. Tojo, “Automatic recognition of music score”, Proceedings of 6th International Conference on Pattern Recognition, 1982.

[18] A. Capela, A. Rebelo, J. Cardoso, & C. Guedes, “Staff Line Detection and Removal With Stable Paths.” International Conference on Signal Processing and Multimedia Applications, 2008.

|  |  |
| --- | --- |
|  | [19] R. Göcke, “Building a System for Writer Identification on Handwritten Music Scores.” 2003  [20] A. Rebelo, & G. Capela, J. Cardoso, “Optical recognition of music symbols: A comparative study.” *International Journal on Document Analysis and Recognition,* 2010.  [21] K.T. Reed and J.R.Parker, “Automatic Computer Recognition of Printed Music,” in *Proceedings of the ICPR*, 1996.  [22] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. Marcal, C. Guedes, & J. Cardoso, “Optical Music Recognition State of the Art and Open Issues.” 2012  [23] A. Rebelo, “Robust Optical Recognition of Handwritten Musical Scores based on Domain Knowledge.” 2012.  [24] J. Cardoso, & A. Rebelo, “Robust Staffline Thickness and Distance Estimation in Binary and Gray-Level Music Scores.” 2010  [25] A. Zeng, “Optical Music Recognition.” 2010.  [26] T. Pinto, A. Rebelo, G. Giraldi, & J. S. Cardoso, “Music score binarization based on domain knowledge.” 2011.  [27] P. Bellini, I. Bruno, & P. Nesi, “Optical music recognition: Architecture and algorithms.” *In Interactive Multimedia Music Technologies*, 2008.  [28] J. Novotny, & J. Pokorny, “Introduction to Optical Music Recognition: Overview and Practical Challenges.” *Charles University*, 2015.  [29] D. Bainbridge, “Extensible Optical Music Recognition.” January 1997.  [30] S. Sheridan, & S.E. George, “Defacing Music Scores For Improved Recognition.” 2004.  [31] D. Bainbridge, & T. Bell, “An Extensible Optical Music Recognition System.” 1997. |