Object Recognition of Handwritten and Printed Musical Notation Using Neuro-Fuzzy

Lintag, Joshua F.
Regodon, Rhyle Abram P.

ATENEO DE DAVAO UNIVERSITY
SCHOOL OF ARTS & SCIENCES
DAVAO CITY

October 2018

Abstract

From the earlier times until today, sheet music has always been an integral part of a musician or a composer's life, whether that individual is a professional, a student, or an enthusiast, they would always write their sheet music onto paper. These papers would either be lost, destroyed, or in the case of non digital copies, would need adjustments on spot. There are also some paper that are not comprehensible to other people because of the way a musician or composer would write their own versions, each of them having their own writing styles for different kinds of musical notations. After long advancements of technology mostly from Optical Characte Recognition, Optical Music Recognition was developed in order to be of great help for these musicians. Although Optical Music Recognition is efficient in recognizing handwritten or printed sheet music, the methods and algorithms used for it barely scratched the surface of OMR. This research and study is aimed to help Optical Music Recognition by classifying both handwritten and printed musical notation using a Concurrent Neuro-Fuzzy System which is a Neural Network that works alongside Fuzzy Logic.

General Terms:

ANN – Artificial Neural Networks
OMR – Optical Music Recognition
FL/FS – Fuzzy Logic / Fuzzy System
CNFS - Concurrent Neuro-Fuzzy System

Additional Key Words and Phrases:
Music Notations

Contents

Chapter 1

Introduction

1.1 Background of the Study

One of the difficult problems faced by computer vision today is the ability to be able to aid musicians when they write their sheets for music. Musicians today write their sheet music onto paper and later use an application like Sibelius, Musecore, or Finale to have a digital copy of their sheet music. The problem is that it is a repetitive process for the musicians and a very tedious one because they would have to do it all over again. Some sheet music are also complex so it would take hours and hours for them to transcribe sheet music from paper.

One of the approaches that would help solve this problem is the use of Optical Music Recognition (OMR). This branch of image processing and computer vision is similar to Optical Character Recognition (OCR), but in a study conducted by Novotny and Pokorny, they stated [10] that OCR proves to be more difficult because unlike the OMR, OCR only focuses on character recognition while OMR focuses on music notation recognition. Therefore, this would prove that Optical Music Recognition can not only be useful, but also more efficient.

For this research, it is necessary to use a Machine Learning algorithm like Neuro-Fuzzy. Using techniques modified from vanilla neural networks, Neuro-Fuzzy is a hybrid neural network tuned with Fuzzy Logic (Fuzzy Sets, Fuzzy Rules) in order to have a human-like reasoning [10].

1.2 Problem Statement

This research and study seeks to identify music notation using optical music recognition with a machine learning algorithm. Primarily, this study would answer the following questions:

1. What are the factors needed to detect a handwritten and printed sheet music?

2. Where does Neuro-Fuzzy algorithm stand when compared to other similar machine learning algorithms in terms of accuracy?

3. How can Neuro-Fuzzy machine learning algorithm be used to identify sheet music?
4. What is the accuracy ratings of the Neuro-Fuzzy machine learning algorithm when applied to reading handwritten and printed sheet music?

1.3 Objectives

The whole objective of this research is to know how a machine learning algorithm works to identify handwritten sheet music. During the research, this study would hope to:

1. To discover the factors need to detect handwritten and printed sheet music.
2. To Find out accuracy ratings of Neuro-Fuzzy and other different machine learning algorithms.
3. To implement the Neuro-Fuzzy machine learning algorithm to read handwritten and printed sheet music.
4. To determine the accuracy of Neuro-Fuzzy machine learning algorithm applied to handwritten and printed sheet music.

1.4 Significance of the Study

Machine learning is one of the most efficient approaches in optical music recognition. This would be of great help to musicians, songwriters, and composers since now they would be able to easily convert their handwritten music sheets into digital format. With the help of Machine Learning, Optical Music Recognition would be faster and more efficient. Furthermore, it will also contribute to researchers of OMR. With all the reasons stated above, this study will be of help to build a better algorithm and to aid musicians and researchers alike.

1.5 Scope and Limitations

This research is limited to handwritten musical notations without the staff line for the training and testing data to be fed on the Concurrent Neuro-Fuzzy system. It can, however, predict musical notations both with and without the staff lines. Furthermore, this research would only use the symbols found on table 1. In the entirety of this research, the Concurrent Neuro-Fuzzy system will be used in the feature extraction and classification of data when training the model. Only the trained Convolutional Neural Network will be used for the testing and prediction of data.

Chapter 2

Review of Related Literature

2.1 Optical Music Recognition

In a study conducted by Novotny and Pokorny, they stated that "Optical music recognition (OMR) refers to a discipline that investigates music score recognition systems. This is similar to well-known optical character recognition systems, except OMR systems try to automatically transform scanned sheet music into a computer-readable format." [10]

Optical Music Recognition (OMR) is a form of Optical Character Recognition (OCR) that is used to understand musical scores and convert it to an audio file or an editable file. OMR has been studied for a decade, and a lot of researchers have addressed these problems in different ways and through different algorithms. These methods and algorithms, however, still only scratch the surface of OMR, and researchers are still bothered with plenty of roadblocks. however readable in human perception, can be easily mistaken by the program [6].

With the power of machine learning, we can use Optical Music Recognition (OMR) to interpret and understand printed or handwritten scores of music. The images taken will be scanned and using OMR, the musical notation would be recognized so the machine will be able to fully understand the image. In this way, Optical Music Recognition (OMR) is almost the same as Optical Character Recognition (OCR) but OCR proves to be more difficult [16].

Table 1 Music notation

Symbols	Description
𝄚	**Staff:** An arrangement of parallel lines, together with the spaces between them.
𝄞 𝄡 𝄢	**Treble, Alto and Bass clef:** The first symbols that appear at the beginning of every music staff and tell us which note is found on each line or space.
♯ ♭ ♮	**Sharp, Flat and Natural:** The signs that are placed before the note to designate changes in sounding pitch.
	Beams: Used to connect notes in note-groups; demonstrate the metrical and the rhythmic divisions.
> ▼	**Accent and Staccatissimo:** Symbols for special or exaggerated stress upon any beat, or portion of a beat.
♩ ♪ 𝅗𝅥	**Crochet, Quaver and Minim:** The Crochet (closed notehead) and Minim (open notehead) symbols indicate a pitch and the relative time duration of the musical sound. Flags (Quaver) are employed to indicate the relative time values of the notes with closed noteheads.
𝄽 𝄾 𝄿 𝅀	**Quarter, Eighth, Sixteenth and thirty-second rests:** Indicate the exact duration of silence in the music; each note value has its corresponding rest sign; the written position of a rest between two barlines is determined by its location in the meter.
⁀	**Ties and Slurs:** Ties are a notational device used to prolong the time value of a written note into the following beat. The tie appears to be identi-

Figure 2.1: Table 1 - Taken from (Rebelo & Capela, 2009)

2.1.1 Contemporary Approaches

In a research conducted by Rebelo, et al. in 2009, they used 4 recognition processes in order to classify both handwritten and printed musical symbols [12].

The 4 recognition algorithms used to approach this problem were:

1. Hidden Markov Models
2. K-nearest neighbor
3. Neural Networks
4. Support Vector Machines

Respectively, Hidden Markov Models, even though it is rarely used for OMR, are hidden symbols generated by a stochastic process. The researches planned to use the Hidden

Markov Models algorithm because it can be used for recognition and segmentation both at the same time.

The next approach the researchers used was the K-Nearest Neighbor algorithm, which is one of the simplest machine learning algorithms. The algorithm uses Instance-Based Learning to find the nearest neighbor which is classified by votes to the class the object belongs to.

In the third approach they used a multi-layer perceptron neural networks, also known as MLP. This type of a feed-forward network is structured with layers of nodes or units (neurons) and connects or links make up the successive layers in between. Network with K values for each class consisting of 1 for correct and 0 otherwise were used.

The final approach Rebelo, et al. used for their research were Support Vector Machines (SVM) [12].

The main objective of an SVM is to look for a decision boundary found on a plane between the points of two classes in the training data.

Accuracy obtained for the handwritten music symbols for the classifiers trained without elastically deformed symbols

	Neural network (%)	Nearest neighbour (%)	Support vector machines (%)	Hidden Markov model (%)
Accent	85	99	99	91
BassClef	13	78	77	56
Beam	85	98	95	90
Flat	84	99	98	87
Natural	93	99	98	91
Note	82	97	96	73
NoteFlag	51	86	89	64
NoteOpen	3	75	40	22
RestI	78	100	97	90
RestII	96	100	100	92
Sharp	85	98	98	84
Staccatissimo	58	100	100	100
TrebleClef	40	92	90	94
Unknown	52	71	89	38
99% CI for the expected performance in percentage: average (standard deviation)	[81 (0.7); 84 (2.6)]	[93 (0.3); 95 (1.2)]	[95 (0.2); 96 (0.6)]	[77 (1.2); 81 (4.3)]

Figure 2.2: Handwritten music symbol accuracy

2.2 Fuzzy Logic

Based on the mathematical theory of fuzzy sets, Fuzzy Logic was created by Zadeh [15] in 1965 as an extension of Boolean logic which is generalized from the classical set theory of mathematics. The verification by notion of degree was introduced to enable a condition to have a state other than true or false providing flexible reasoning, thus giving Fuzzy

Logic a human-like reasoning, e.g. the speed is "too fast or a little slow". Below is an example of fuzzy logic formalizing human-reasoning: (Dernoncourt, 2013)

If the light is red...	if my speed is high...	and if the light is close...	then I brake hard.
If the light is red...	if my speed is low...	and if the light is far...	then I maintain my speed.
If the light is orange...	if my speed is average...	and if the light is far...	then I brake gently.
If the light is green...	if my speed is low...	and if the light is close...	then I accelerate.

Figure 2.3: Human-like reasoning table by using Fuzzy Logic

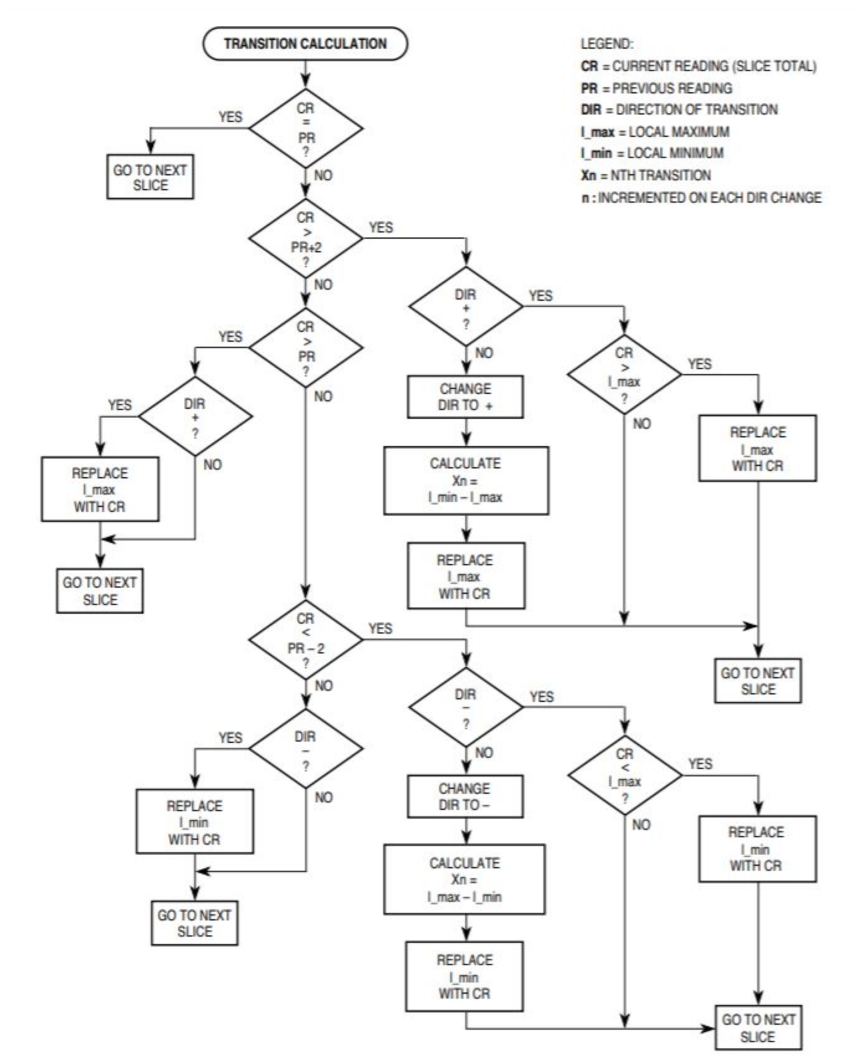


Figure 2.4: Transition Calculation found on Gowan's Optical Character Recognition Using Fuzzy Logic

2.2.1 Fuzzy Set Theory

Fuzzy sets were publicized by Zadeh to create a strict mathematical framework to help with the fuzziness contained in human language. According to Zadeh, "The Fuzzy Set Theory provides a point of departure used in ordinary sets." Later on, this was proved to cover a vaster scope of applicability in both fields of pattern classification and information processing [15].

Definition 1 *If X is a collection of objects denoted generically by x, then a fuzzy set $\tilde{A}$ in X is a set of ordered pairs:*

$$\tilde{A} = \left\{(x, \mu_{\tilde{A}}(x) | x \in X)\right\}$$

Figure 2.5: Fuzz Logic Definition 1

2.3 Digital Image Processing

Digital images contain a limited number of elements called picture elements, commonly referred to as pixels, and these pixels each have their own value and location. Digital Image Processing is the act of using digital computers in handling pixels that are grouped by regions. These regions are divided within the process to complete the image processing much easier and more identifiable to a computer. Machines that handle digital image processing can handle a broad span of the electromagnetic spectrum, unlike human vision which is limited only to the visual spectrum. The images that can be processed by these machines can cover rays that even humans cannot perceive with the naked eye. This is a promising fact for the future of digital image processing going forward for it can be used in a wide range of applications that have not yet been fully explored by man [6].

According to [3], There are 4 phases of Image processing. The first step is the preprocessing of the image. The next step is to segmentation. The third step is feature extraction. And finally, classification and recognition.

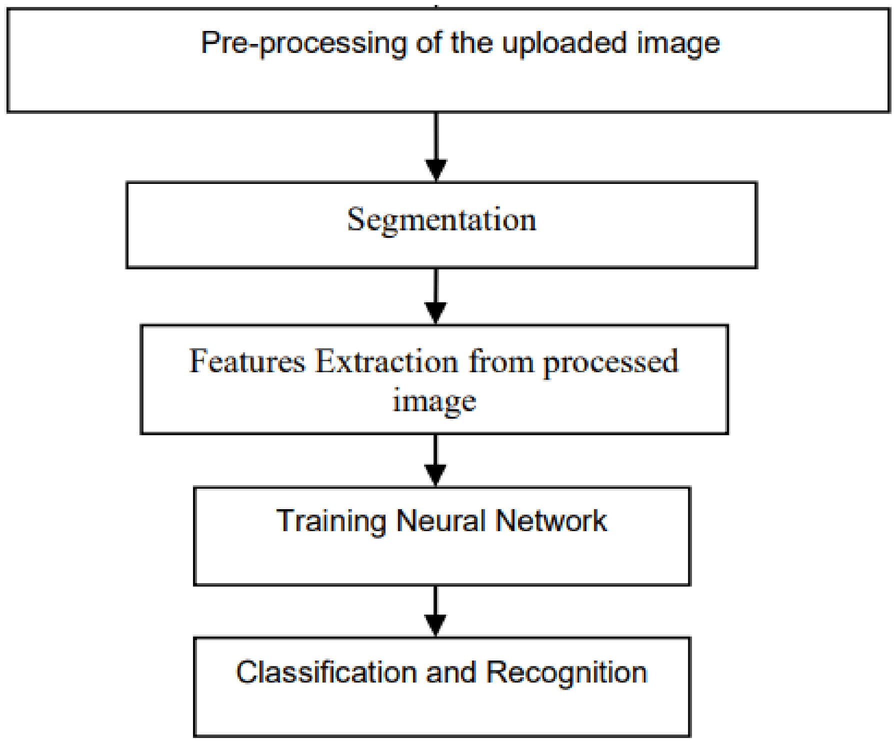


Figure 2.6: The four phases of Image Processing

2.3.1 Pre-Processing

This phase renders the image to prepare it for segmentation. It does that by dissecting the background from the patterns seen in the image. This is done through noise reduction, normalization, and compression. Pre-processing also defines patterns that are closely packed together. It is then turned from a grayscale image to a binary image by means of binarization. Normalization makes the data set uniform. Compression reduces the size of the image [4].

Binarization

The binarization method is the method responsible for turning a grayscale image to a binary image. This enhances the edges of the image and makes the edges of characters distinct from the background by turning the gray levels of an image (from 0 to 256 gray levels) to binary. This method is the one responsible to make details and edges of characters more distinct (Chaudhuri, Mandaviya, Badelia, & K Ghosh, 2017). The researches decided to use Otsu's method to binarize the image. The formula for this method is as follows:

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t)$$

Figure 2.7: Otsu's Formula for Binarization

Noise Reduction

Image noise in Optical Character Recognition adds additional markings on characters such as lines, gaps, marks, etc. Additional alterations to the image include rounding of corners, distortion of the character, dilation, and erosion. All these unwanted markings and distortions of characters would impose problems and it is necessary to correct this flaws before processing the information contained in the image. One of the three major groups of techniques on noise reduction according to Chaudhuri et al. are morphological operations. Morphological operations are designed to correct broken strokes, decompose connected strokes, remove random points, extract boundaries, etc. Its purpose is to remove noise from images that have poor quality of detail due to erroneous hand movements made when writing the input data ([4]. For this method of Pre-Processing, the Gaussian Filter noise reduction was applied:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

Figure 2.8: Gaussian function in one dimension

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

Figure 2.9: The product of two such Gaussian functions

Normalization

Normalization aims to remove different instances of writing that may be caused by erroneous handwriting or image digitalization problems to obtain a uniform data set. Two of the four groups of approaches mentioned by Chaudhuri et al. are skew normalization and baseline extraction, as well as size normalization. Skew normalization and baseline extraction aims to correct the errors caused curving of the image. The size normalization aims to standardize the size of the character to be scanned. Size normalization may be used for both horizontal and vertical standardization. It is used for scaling and is helpful on adjusting the data to be processed so that the whole system would yield less errors in detecting the character [4].

Compression

As the word implies, compression aims to reduce the size of the of the size of the images. Large images are not suitable for Optical Character Recognition due to it needing a longer amount of time in the processing stages. Reduction of the data set's size (while also preserving the shape information) will positively affect the speed of OCR. An approach for compression according to Chaudhuri et al. would be thresholding. Thresholding aims to reduce storage requirements by adding a threshold in binary images [3]. Global thresholding assumes a threshold value basing on the estimated intensity difference of the characters and background of the binarized images. Local thresholding, on the other hand, assumes a threshold value basing on each pixel of a given local area [4].

2.3.2 Segmentation

Segmentation is the process of the isolation of the individual character images from the refined image. This is considered as the area of the study having the most problem [14]. In this phase, the image sequence of characters is fragmented into sub-images of individual character. The pre-processed input is also fragmented into isolated characters, labeling each character with numbers. Numbering of images serves as the indicator of the number of characters in an image. After fragmentation, the characters are fit into a defined number of pixels. The size of the characters is then normalized due to the characters having varied sizes. The purpose of this is to prepare the image for feature extraction [4].

$$\operatorname{argmin}_{u,K} \gamma|K| + \mu \int_{K^C} |\nabla u|^2 dx + \int (u-f)^2 dx.$$

Figure 2.10: Image Segmentation formula

2.3.3 Feature Extraction

Feature extraction aims extract informative and non-redundant features from a data set. The redundant representative of pixels in an image is aimed to be reduced to a set of features, also called a feature vector. Feature selection would then determine a subset of initial features. This stage would then be better processed if the data is being represented by only relevant information as compared to the initial pre-processed data. One of the most difficult pattern recognition problems, the main goal of feature extraction is to obtain the vital features of symbols [5].

Gradient Feature Extraction

Magnitude and direction of the greatest change in intensity in a small neighborhood of each pixel is what is measured by Gradient Feature Extraction. Gradient Feature Vector is formed by combining the strength of a gradient per individual direction. Computing the horizontal and vertical components of the gradient is done by means of the Sobel operator, which is usually used in edge detection [1]. The following figures shown below are image representations for the Sobel technique [3] and neighborhood of pixel.

1	2	1
0	0	0
-1	-2	-1

-1	0	1
-2	0	2
-1	0	1

Figure 2.11: Horizontal Component   Vertical Component

$(i-1, j-1)$	$(i-1, j)$	$(i-1, j+1)$
$(i, j-1)$	(i, j)	$(i, j+1)$
$(i+1, j-1)$	$(i+1, j)$	$(i+1, j+1)$

2.3.4 Classification

Neural Networks

The design of neural networks works like a human brain. According to Vieria,et al., [13] neural networks can be trained with examples using vectors, inputs and outputs of a system.

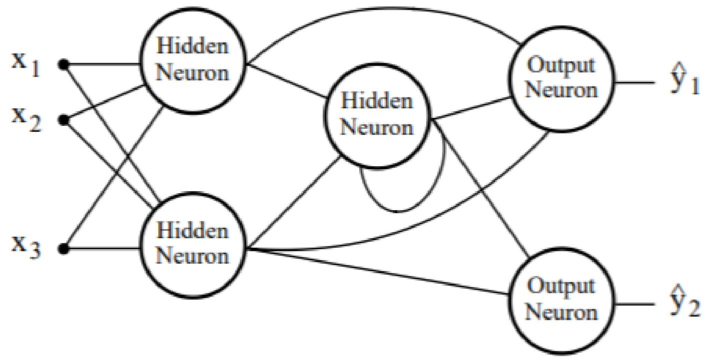


Figure 2.12: The general structure of an artificial neural network with 3 inputs and 2 outputs (Larsen, 1999)

Fuzzy Neural Network

Neuro-Fuzzy networks are a combination of fuzzy logic and neural networks. The hybridization of these two method makes these two different approaches more efficient. Fuzzy logic is designed to handle non-Boolean information, which lie somewhere within 0 and 1. Therefore, fuzzy logic is an efficient way of processing information which data is good at explaining decisions. However, fuzzy logic's weakness is not being able to automatically obtain the rules used to make said decisions. Neural networks, on the other hand, is designed to recognize certain patterns, depending on the rules it was given, however it does not explain very well how it can obtained decisions. The strengths of the two approaches are each the other's weakness, therefore having to combine these two to make a hybrid system would eliminate previous boundaries that the approaches have individually [9].

There are 3 types of Neuro-Fuzzy systems, Cooperative Neuro-Fuzzy Systems, Concurrent Neuro-Fuzzy Systems, and Hybrid Neuro-Fuzzy Systems. Cooperative Neuro-Fuzzy Systems are used primarily on initial phases. Neural networks in this type use training data to divide fuzzy systems by sub-blocks. After this part, the neural network is not anymore used, and the fuzzy system is executed. Concurrent Neuro-Fuzzy Systems are a type of neuro-fuzzy system which is only classified as a neuro-fuzzy system loosely due to it having both approaches work together. Either the input is entered and pre-processed in the fuzzy system, and it is being processed and outputted in the neural network, or vice versa [2]. Hybrid Neuro-fuzzy, according to Kruse [7] "a fuzzy system that uses a learning algorithm based on gradients or inspired by the neural networks theory (heuristical learning strategies) to determine its parameters (fuzzy sets and fuzzy rules) through the patterns processing (input and output)". Hybrid systems are composed of fuzzy logic, neural networks, genetic algorithms, and expert systems, proving their efficiency in facing a number of problems. Specific to every method are unique computational abilities that make them efficient for certain fields. However, these specific abilities also mean that a method is only able to do fairly limited tasks [5].

Although there are a lot of different approaches, neuro-fuzzy system is the term used for approaches which display similar properties. A Neuro-Fuzzy system is a system that primarily uses a combination fuzzy logic and is being trained by a neural network-derived algorithm. A Neuro-Fuzzy system can be viewed as a neural network with three layers. The first layer is the input layer, the second layer represents an area where fuzzy rules are applied, and the third layer which represents the output of the fuzzy rules [11].

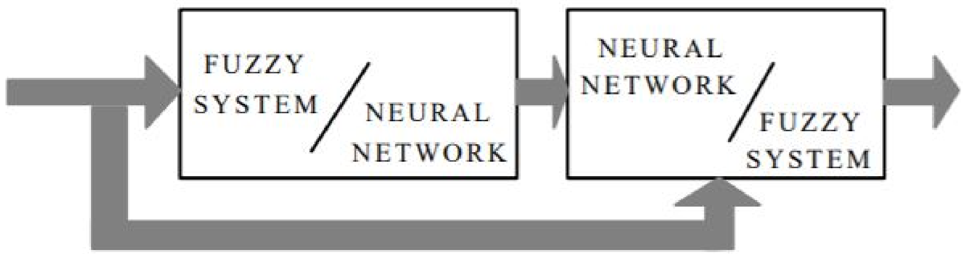


Figure 2.13: Concurrent Neuro-Fuzzy System

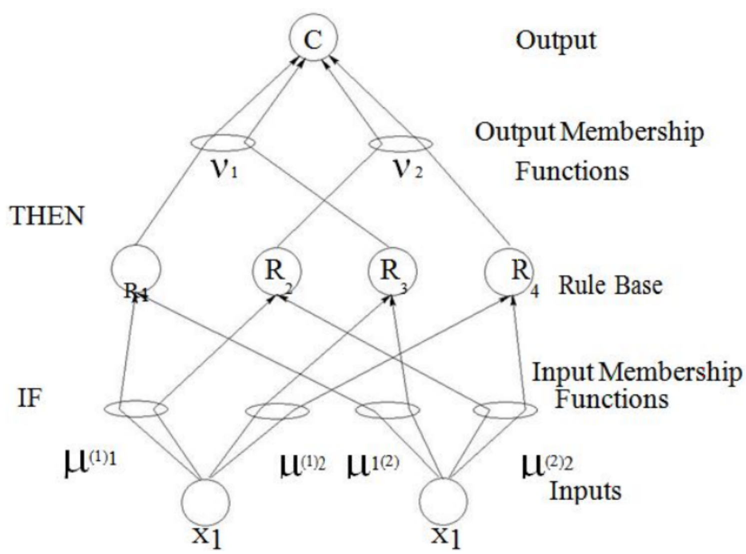


Figure 2.14: Adaptive Neuro-Fuzzy Inference System (ANFIS) Architecture

Neural Networks	Fuzzy Systems
no mathematical model necessary	no mathematical model necessary
learning from scratch	apriori knowledge essential
several learning algorithms	not capable to learn
black-box behavior	simple interpretation and implementation

Figure 2.15: Fuzzy System and Neural Networks Comparison table (L. A., 1965).

Skills		Fuzzy Systems	Neural Nets
Knowledge acquisition	Inputs	Human experts	Sample sets
	Tools	Interaction	Algorithms
Uncertainty	Information	Quantitive and Qualitive	Quantitive
	Cognition	Decision making	Perception
Reasoning	Mechanism	Heuristic search	Parallel computations
	Speed	Low	High
Adaption	Fault-tolerance	Low	Very high
	Learning	Induction	Adjusting weights
Natural language	Implementation	Explicit	Implicit
	Flexibility	High	Low

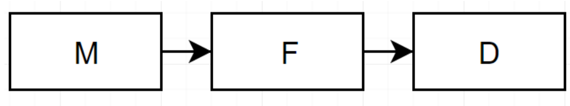
Fuzzy System

Mathematical calculus is used by fuzzy systems to manipulate knowledge with a degree of uncertainty. Using the Fuzzy set theory initiated by Zadeh, Lofti in 1965 [8], a Fuzzy System's behavior could be described as:

IF premise THEN consequent

This system uses linguistic variables combined with symbolic terms each representing a fuzzy set which consists of three stages (Veira, Fernando, & Alexandre, 2004):First, using fuzzification to map the values of the input to a certain level of compatibility to the respective fuzzy sets then processing of the rules so that the Fuzzy System will know the strength of each individual input, and finally, using defuzzification to transform the fuzzy value results into their numerical values.

According to the study of Vieria, et al., [13] fuzzy systems have the following advantages and disadvantages. Fuzzy systems have capacity of representing uncertainties with linguistic variables while having easy interpretation of results and addition of new rules make extension of base knowledge easier. Fuzzy Systems also have disadvantages such as only knowing how to answer what's written on its rule base, changing the system topology would need the rule base altered, an expert is needed to determine the inference logical rules.



The task of pattern recognition involves the computer transforming from the measurement space M to the feature space F, and finally, to the decision space D [11]

The measurement space involves image manipulations like noise reduction, filtering, enhancement, skeleton extraction, and contour extraction in order to extract features. This is commonly known as image processing. If the processed information is recognized and understood, then the complete system would then be called vision. [6]

2.3.5 Theoretical Framework

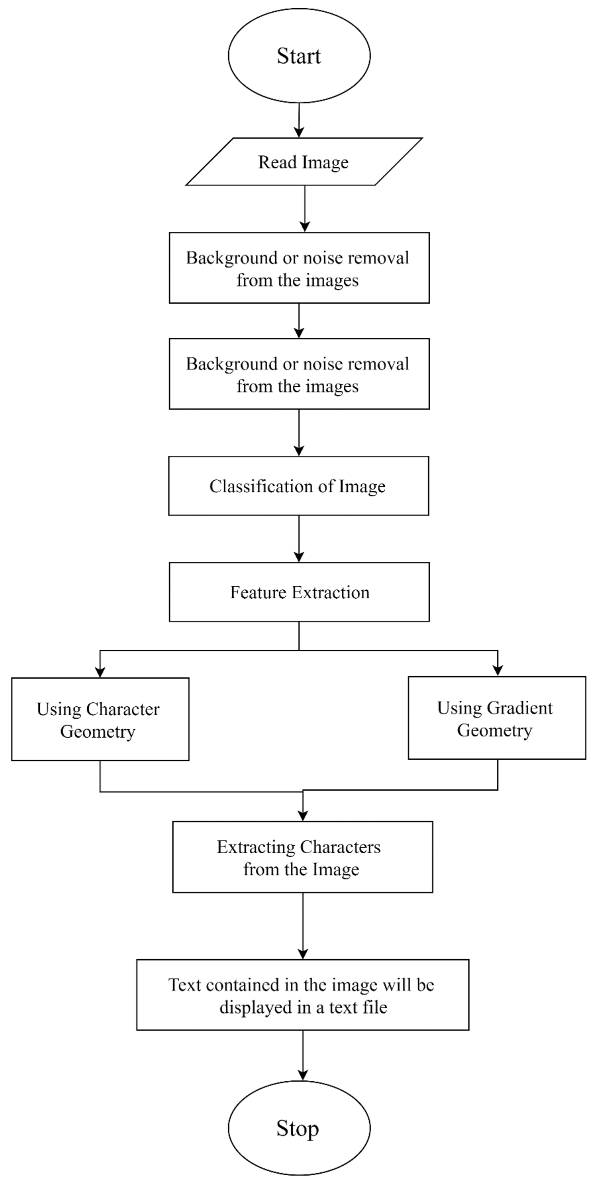
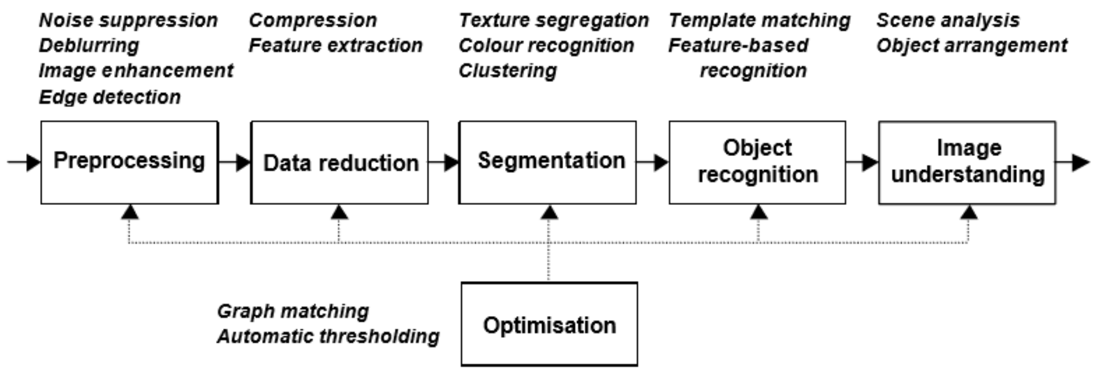


Figure 2.16: Using a neural network in optical image recognition (Adi, 2014)

In this research, a photographed handwritten text is fed to the program. The objective of the program aims to identify all handwritten letters and convert it into its computerized form and put it in a .doc file. The image underwent 4 phases: Pre-processing, segmentation, feature extraction, and classification and recognition.

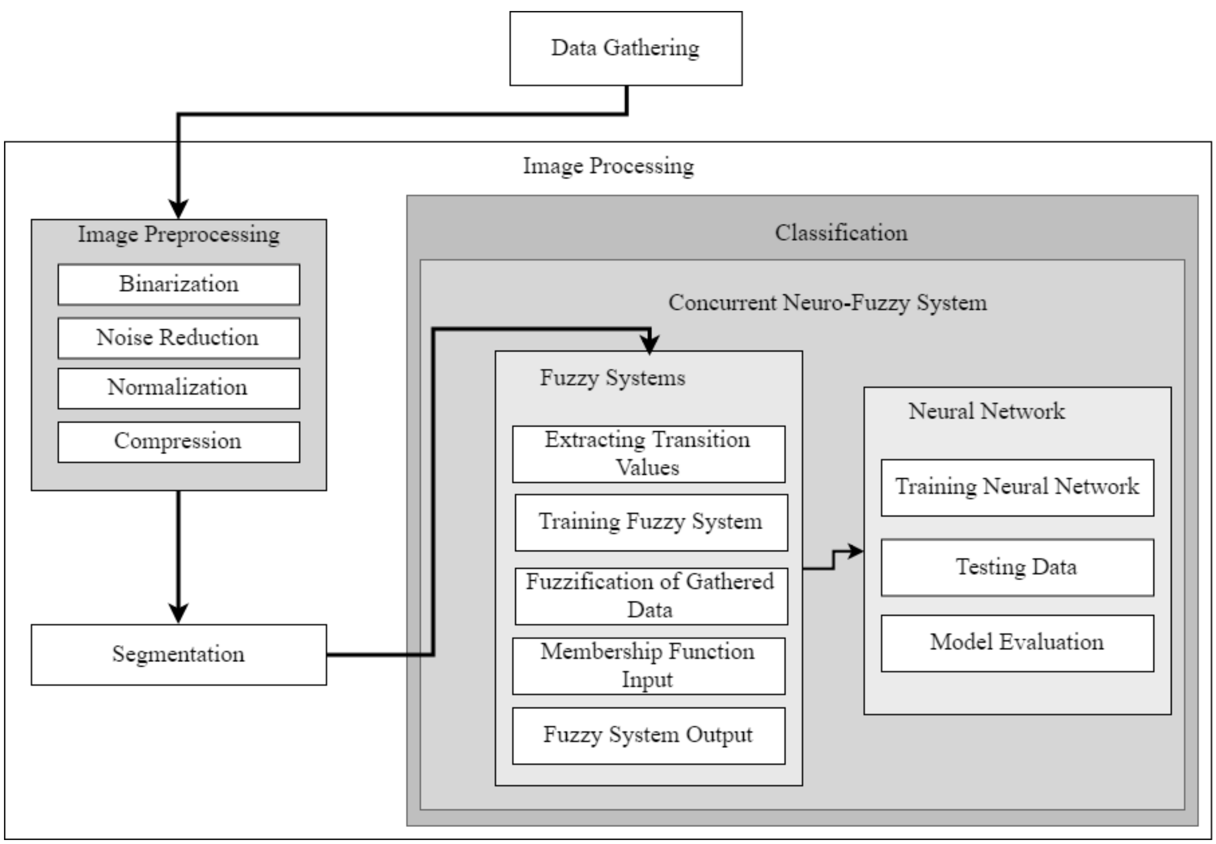


In this research, the researchers created a framework containing five different tasks: preprocessing for image enhancement and edge detection, data reduction for compression and feature extraction, segmentation for texture segregation and clustering, object recognition for feature-based recognition, and finally, image understanding for scene analysis and object arrangement. The researchers used this framework for image recognition using neural networks. (Egmont-Petersen, de Ridder, & Handels, 2001)

Chapter 3

RESEARCH DESIGN AND METHODOLOGY

3.1 Conceptual Framework



3.2 Methodology

In this study, the phases are as follows:

1. Data Gathering
2. Image Processing
    2.1 Pre-Processing
        2.1.1 Binarization
        2.1.2 Noise Reduction
        2.1.3 Normalization
        2.1.4 Compression
    2.2 Segmentation
    2.3 Feature Extraction & Classification
        2.3.1 Fuzzy Systems
            2.3.1.1 Extracting Transition Values
            2.3.1.2 Training of Fuzzy Systems
            2.3.1.3 Membership Function Input
            2.3.1.4 Fuzzy System Output
        2.3.2 Neural Network
            2.3.2.1 Training of Neural Network
            2.3.2.2 Testing of Data
            2.3.2.3 Model Evaluation

3.2.1 Data Gathering

The data gathering for this study is done by having the researchers give participants papers containing musical notations, and papers for them to copy the notations/symbols on. The papers given to the survey participants will have a writing space and a notation so each participant will be guided on what to draw. The aim of this is to have a wide range of different handwritten symbols to make the system learn and determine what the handwritten symbols' digital counterparts are.

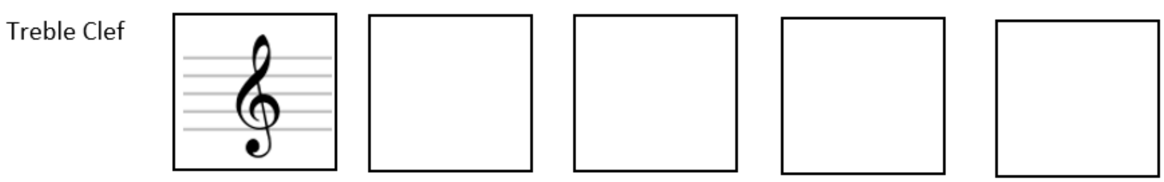


Figure 3.1: An example of a blank survey form

From the survey, the researches were able to acquire 100 handwritten musical notations. These musical notations were added to the main dataset.After the dataset underwent image processing, they are then passed through a data augmentor which produced 200 more images per notation resulting in a dataset of 9000 musical notations. For training, 6000 musical notations were used and for testing the convolutional neural network, 3000 musical notations were fed.

3.2.2 Image Processing

In this step, the data set will undergo many operations relating to correction for it to be refined and ready for classification.

3.2.2.1 Pre-processing

This first step of image processing would oversee the removal of additional unwanted symbols and disfigurements of the image to prepare it for segmentation.

3.2.2.1.1 Binarization

Binarization is the first step the researchers will take in pre-processing the data set. What this does is alter the image, having it convert a grayscale image to a binary image, enhancing its details and leaving symbols on paper distinct.

3.2.2.1.2 Noise Reduction

The next step in pre-processing is noise reduction. In this step, removal of unwanted markings occurs. This step specifically aims to separate the character and the staff through staff line removal. This makes the symbol much more readable than it was before.

3.2.2.1.3 Normalization

The third step in pre-processing is normalization. In here, the data set will be normalized in terms of distortion due to curvature of the image by skew normalization and baseline extraction. The data set will also undergo size normalization to standardize the dimensions of each detected symbol.

3.2.2.1.4 Compression

The fourth and last step in pre-processing is compression. In this step, images will be compressed in terms of size of the image (in bytes) for it to be significantly efficient. This is to be done by thresholding.

3.2.3 Segmentation

After pre-processing the data set, it would then undergo segmentation. Here, both the reference images and the data set will now be divided in sub-images. These se gments are then fit into a defined number of pixels. This would cause the symbols to normalize due to it having varied sizes even after pre-processing.

3.2.4 Feature Extraction

Next phase is feature extraction. This phase separates the background areas of the data set and the areas of the data set where there are significant features. The feature selection would now determine a subset of those significant features, obtaining only the vital features of the symbols. This is done by using the Canny Edge Detector where the edges of a specific symbol are highlighted to make features more visible to the system.

3.2.5 Concurrent Neuro-Fuzzy System

3.2.5.1 Fuzzy System

In the Fuzzy System, some of the data gathered will have their transition values extracted using Gowan's method. Since the Fuzzy System will be used to determine the inputs of the CNN, the researches made a universe of discourse for each of the classification to avoid memberships overlaps and rule conflicts. After this, the entire dataset will be fed in to the system and it will separate what can and cannot be used as the CNN input.

3.2.5.2 Convolutional Neural Network

In order for the Concurrent Neuro-Fuzzy System to work, a neural network must be connected with fuzzy logic or vice-versa. In this case, the researches chose to use Convolutional Neural Networks with TensorFlow. The inputs of this neural network will be decided by the Fuzzy System and with deep learning, the CNN will learn the 18 classifications.

Chapter 4

Results and Discussions

4.1 Data Gathering

The researchers were able to gather approximately 500 musical notations from the survey given to the participants. A data augmentor is then used to create a larger dataset from those 500 musical notations.
For each classification, there was an average of 300 symbols for training and 150 for testing with an overall count of approximately 10,216 symbols.

4.2 Concurrent Neuro-Fuzzy System

4.2.1 Fuzzy System

Extracting Transition Values

Linguistic terms of the fuzzy rules will be defined to express membership functions. For this part, the researchers used the Canny Edge Detector to derive transition values from segmented images. Using a special and manually selected data set and following the steps done by W.Gowan [19], the transition values were extracted as:

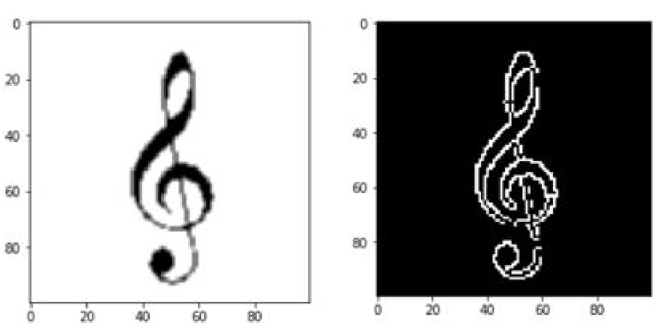
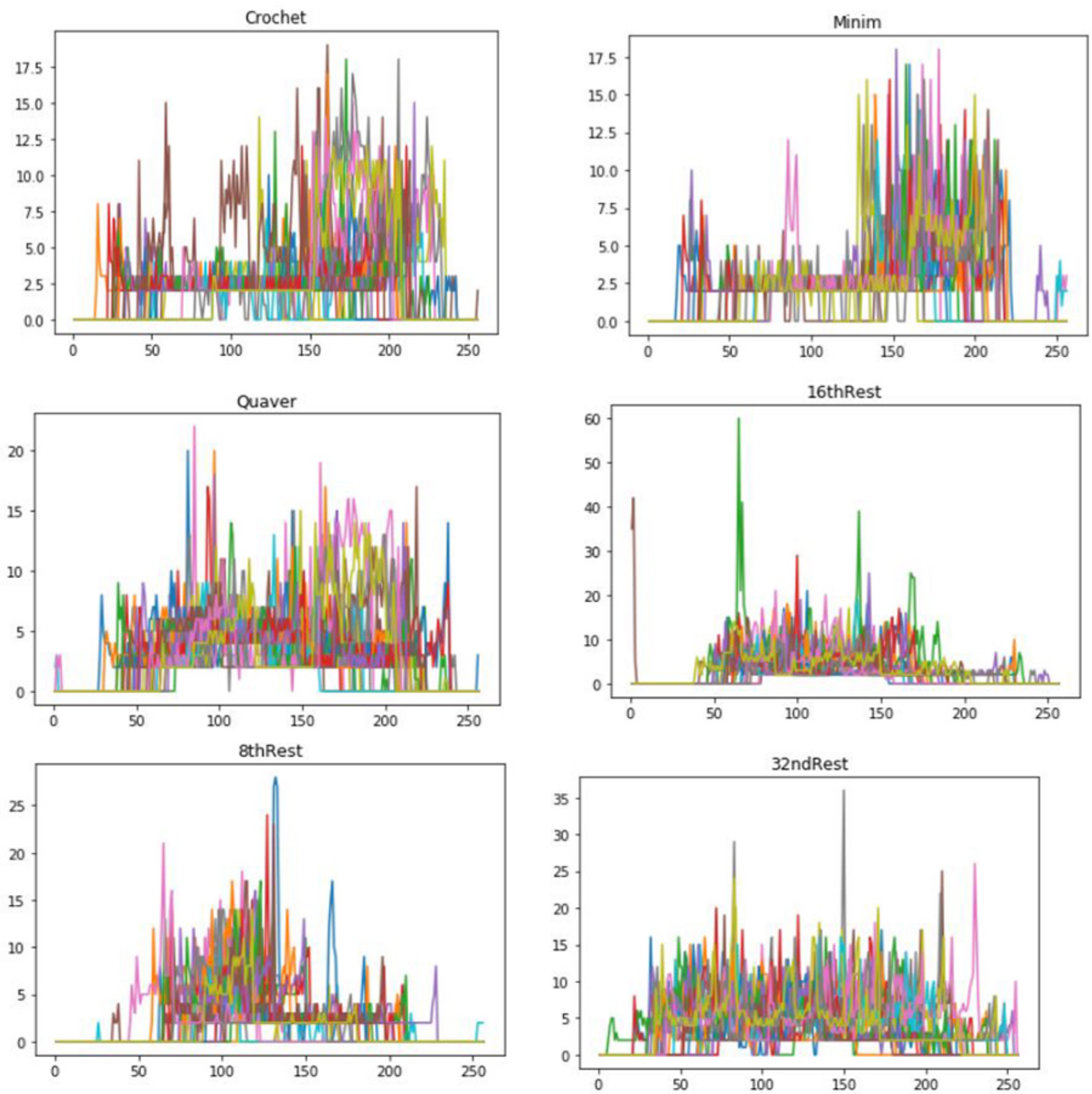


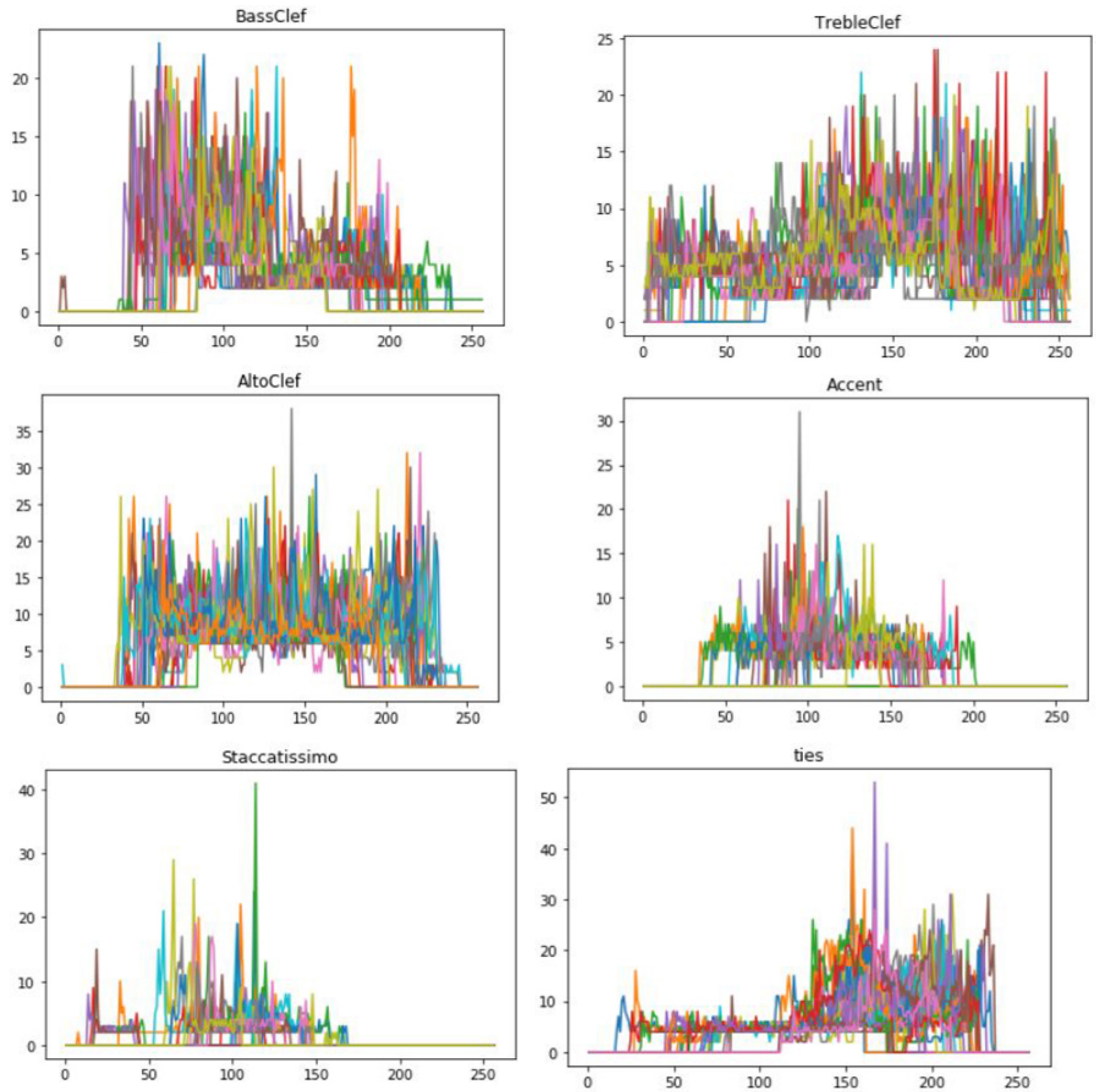
Figure 4.1: Canny Edge Detector Before — After

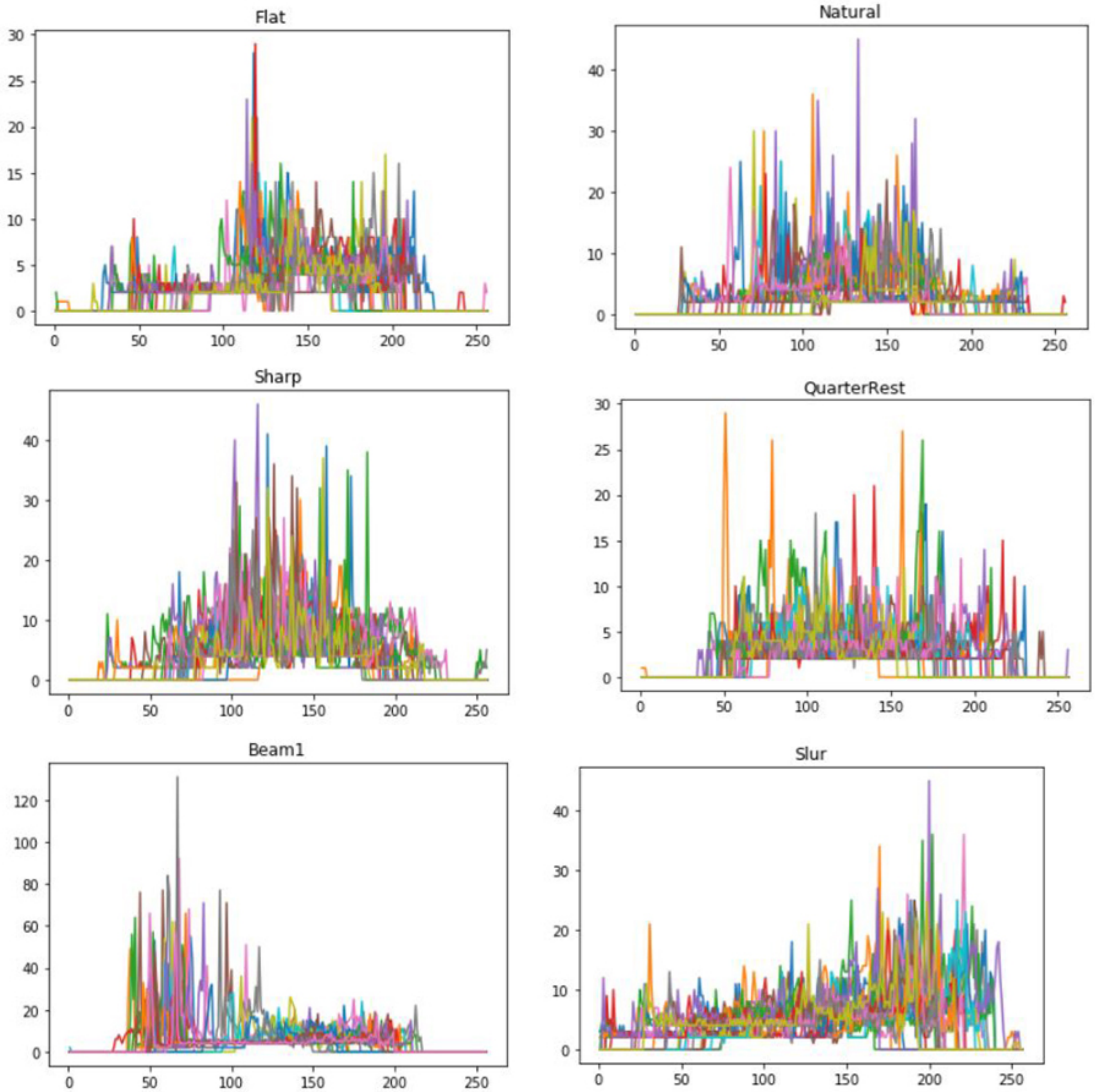
Symbol	X1		X2		X3	
	Low	High	Low	High	Low	High
Treble Clef	-12	-5	0	9	-15	-1
Bass Clef	-21	-3	0	26	-18	-3
Alto Clef	-21	-3	0	20	-22	-4
Sharp	-30	-3	1	22	-18	-3
Natural	-30	-3	0	15	-8	-2
Flat	-29	-1	2	11	-17	-2
Accent	-18	-5	0	23	-27	-3
Crochet	-16	-3	1	22	-16	-2
8th Rest	-16	-3	1	22	-16	-2
16th Rest	-30	-5	0	21	-13	-3
32nd Rest	-16	-3	0	21	-13	-3
Quarter Rest	-29	-3	0	31	-18	-2
Quaver	-12	-3	0	18	-18	-1
Minim	-17	-3	0	21	-15	-3
Slur	-21	-5	1	25	15	-3
Staccatissimo	-10	-1	0	10	-9	-1
Ties	-12	-5	0	12	-8	-1
Beam	-12	-5	0	11	-7	-3

Training Fuzzy Rules

From the gathered data and transition values, the researchers used a Fuzzy Inference program called “FisPro” to feed the derived transition calculation values and make their respective membership functions. To prevent rules and membership functions from overlapping and conflicting with each other, a separate universe of discourse is created for each of the classification so that data that pass through the fuzzy system are strictly controlled.

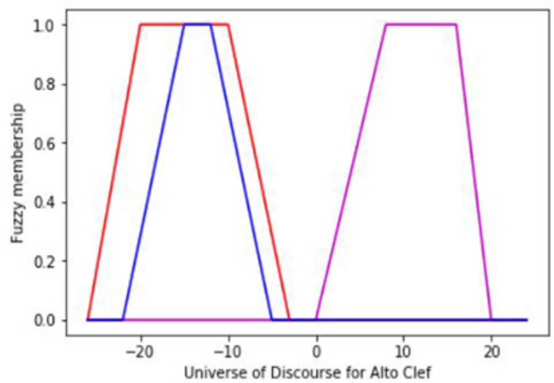

Crochet
Minim
Quaver
16thRest
8thRest
32ndRest

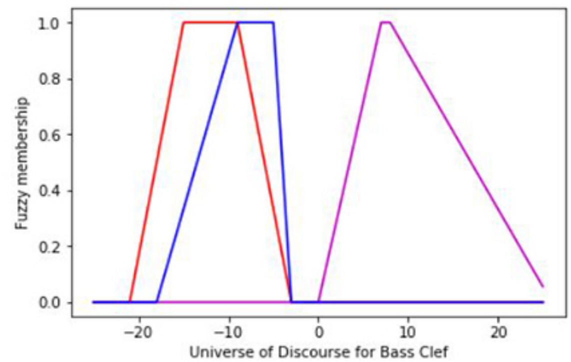

BassClef
TrebleClef
AltoClef
Accent
Staccatissimo
ties

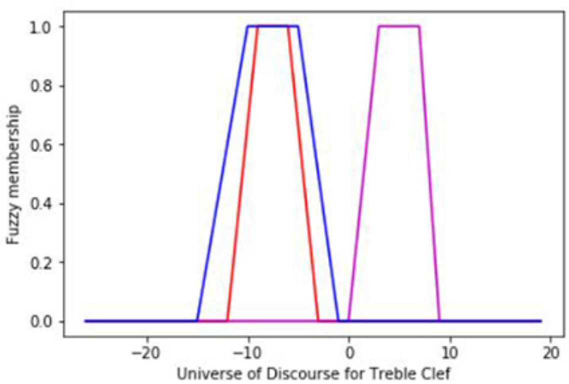


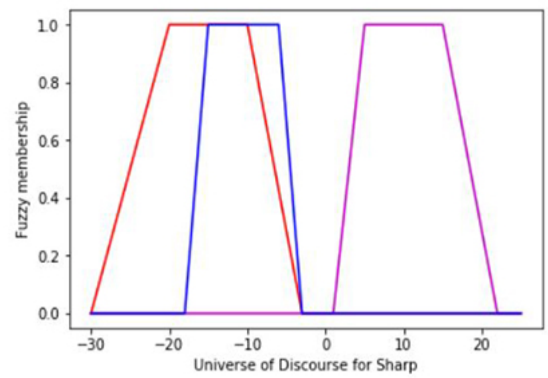
Membership Functions

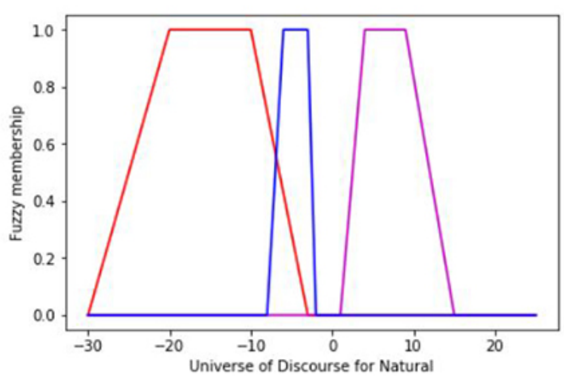
After training the Fuzzy System, membership functions for each classification were created.

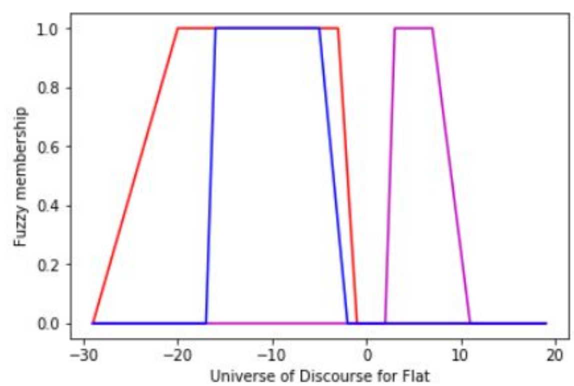

Fuzzy membership
Universe of Discourse for Alto Clef

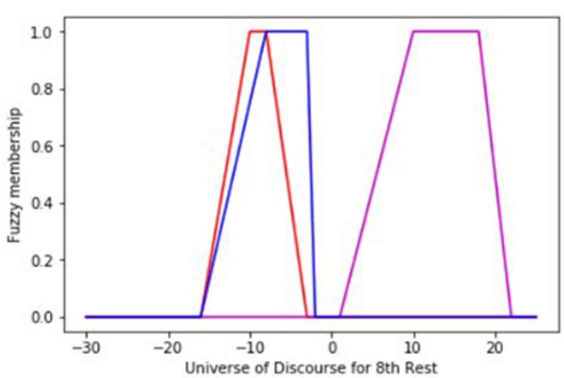

Fuzzy membership
Universe of Discourse for Bass Clef

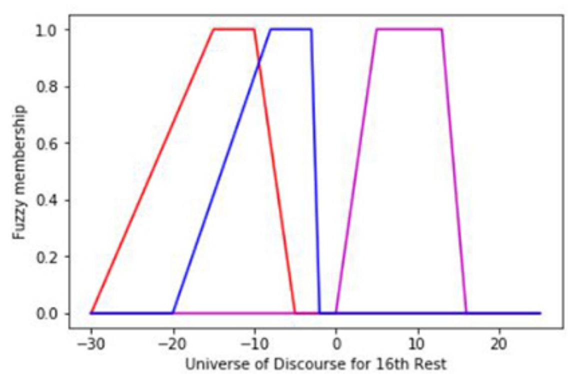

Fuzzy membership
Universe of Discourse for Treble Clef

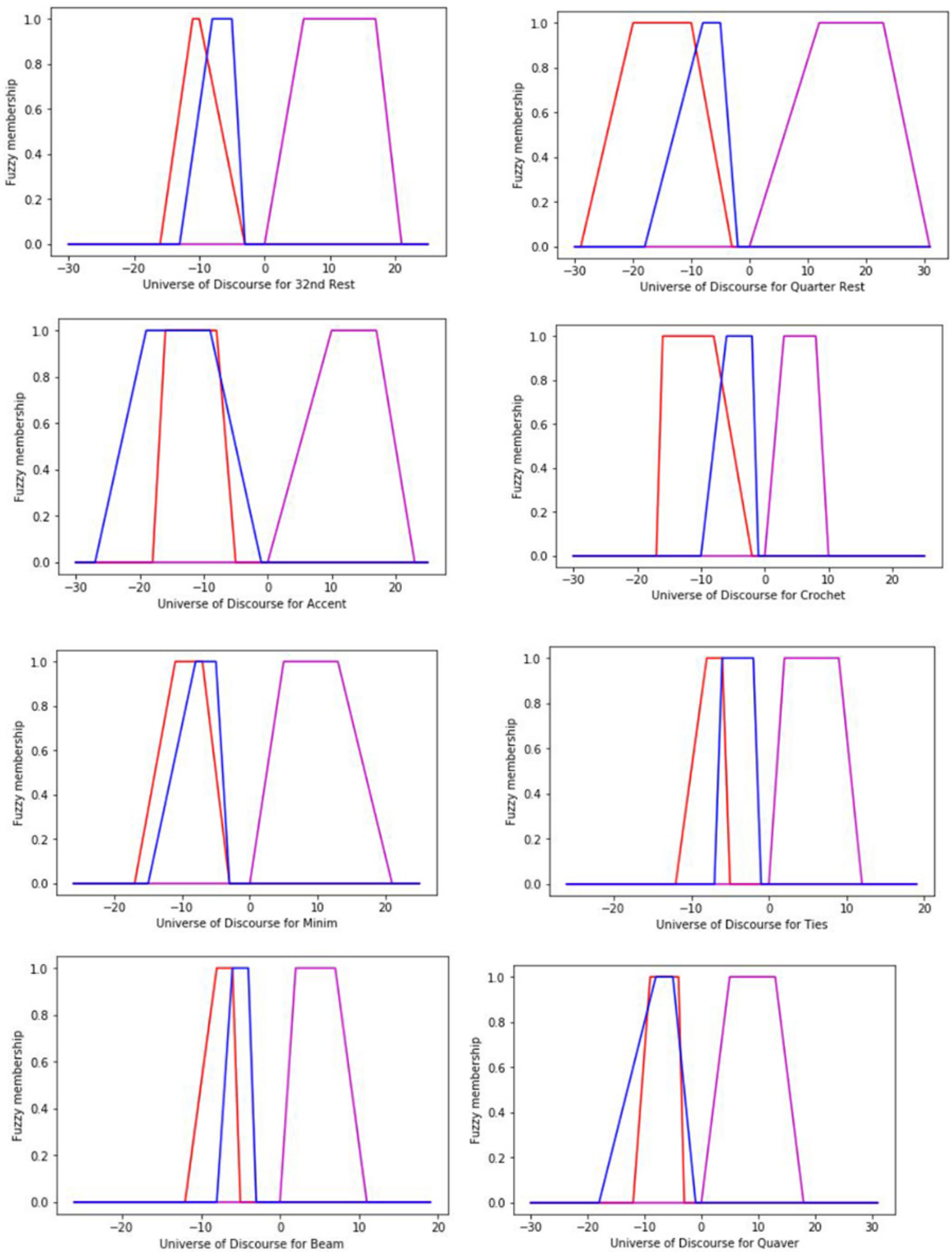

Fuzzy membership
Universe of Discourse for Sharp


Fuzzy membership
Universe of Discourse for Natural


Fuzzy membership
Universe of Discourse for Flat

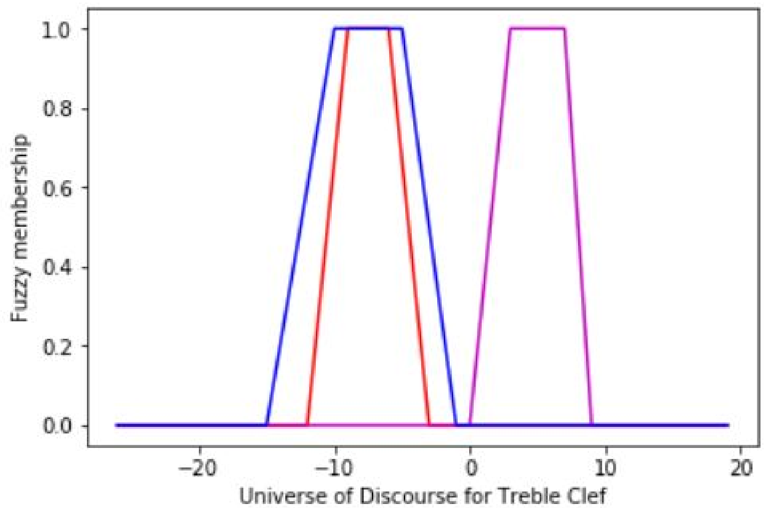

Fuzzy membership
Universe of Discourse for 8th Rest

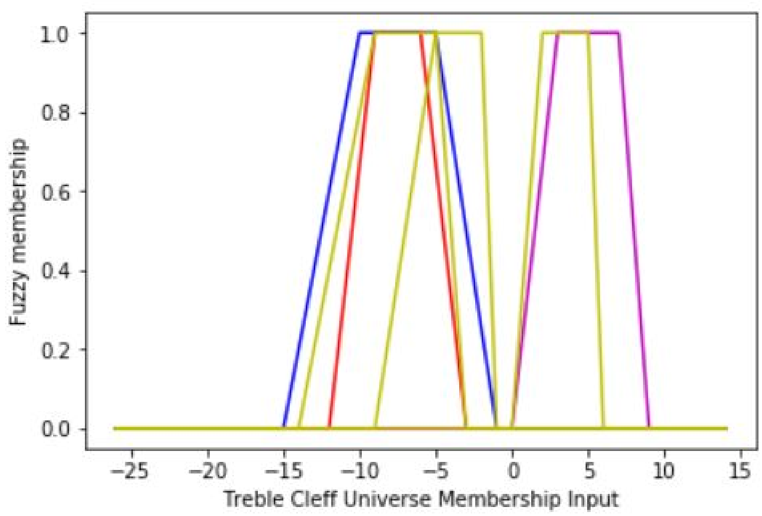

Fuzzy membership
Universe of Discourse for 16th Rest

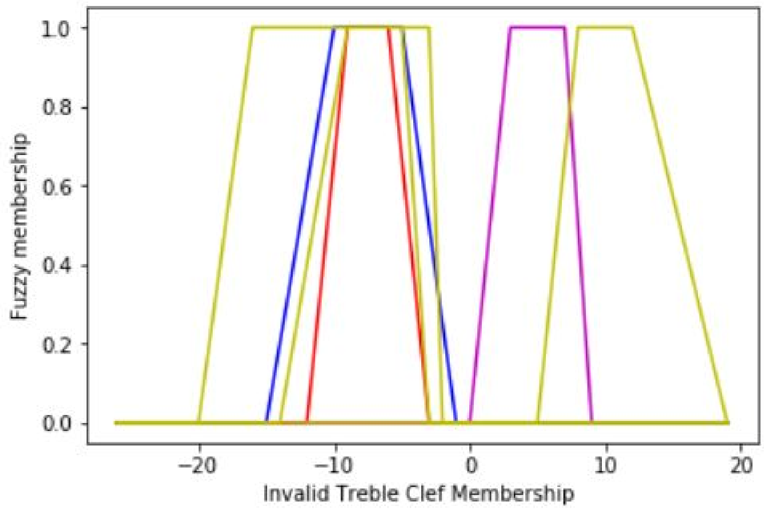

Fuzzy membership
Universe of Discourse for 32nd Rest
Fuzzy membership
Universe of Discourse for Quarter Rest
Fuzzy membership
Universe of Discourse for Accent
Fuzzy membership
Universe of Discourse for Crochet
Fuzzy membership
Universe of Discourse for Minim
Fuzzy membership
Universe of Discourse for Ties
Fuzzy membership
Universe of Discourse for Beam
Fuzzy membership
Universe of Discourse for Quaver

Fuzzy Output

After the membership functions were created, each classification of the dataset was fed to the Fuzzy System. The Fuzzy System then determined the data that were allowed to be fed to the neural network. Since the Fuzzy System only accepted data that had at least a 50% relation to the graph, some of the input data were not accepted.







4.2.2 Convolutional Neural Network

The researches decided to use a Convolutional Neural Network from TensorFlow.The CNN's model summary are as follows:

Layer(type)	Output Shape	Param #
conv2d_(Conv2D)	(None, 148, 148, 256)	2560
activation_3(Activation)	(None, 148, 148, 256)	0
max_pooling2d_2(Max_Pooling2)	(None, 148, 148, 256)	0
conv2d_3(Conv2d)	(None, 148, 148, 256)	590080
activation_4(Activation)	(None, 148, 148, 256)	0
max_pooling2d_3(Max_Pooling2)	(None, 148, 148, 256)	0
flatten_1(Flatten)	(None, 148, 148, 256)	0
dense_1(Dense)	(None, 148, 148, 256)	5971986
activation_5(Activation)	(None, 148, 148, 256)	0

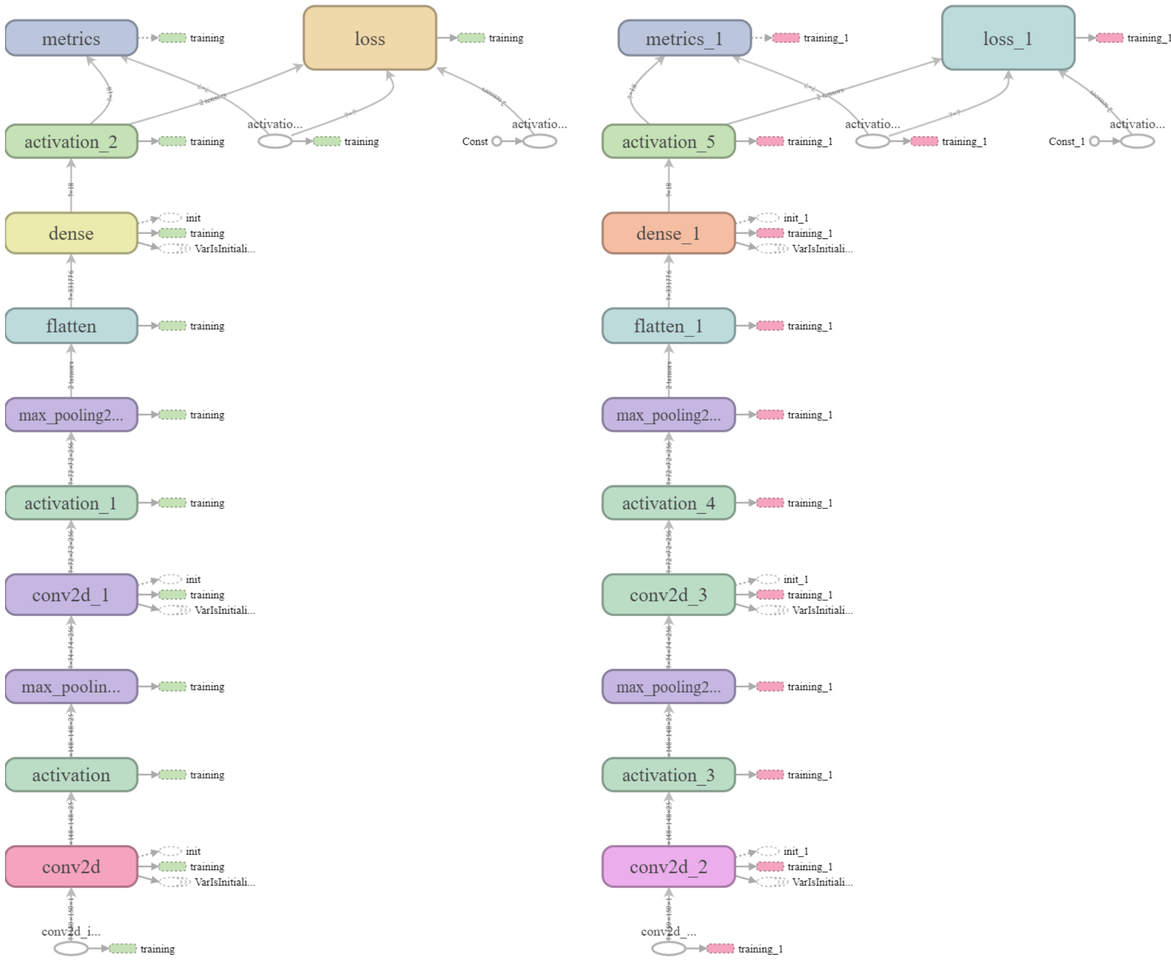


Figure 4.2: Convolutional Neural Network Architecture used alongside Fuzzy System

The inputs that went in for the CNN training were the inputs that the Fuzzy System allowed.
The testing data for the Neural Network were taken from the main dataset without the Fuzzy System. Data that was fed as test data included symbols that are slightly deformed. This was done to see if the CNN would be able to classify them. For training the CNN was able to achieve 92% while testing achieved 82%.

4.2.3 Accuracy Ratings

Training the Convolutional Neural Networks did not achieve a significant increase from 10 to 100 epochs, therefore the researchers decided to use 10. The training accuracy for the CNN achieved 92.86% and the testing accuracy achieved 82.14%.

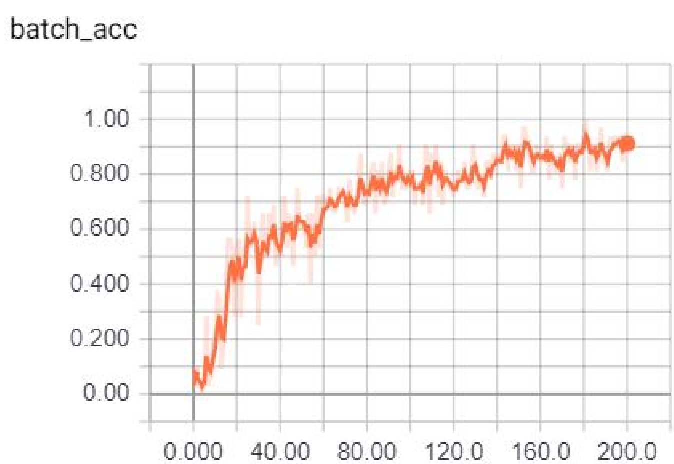


Figure 4.3: History for 10 Epoch Training

The closest research this study can be compared to is the study done by Rebelo et al. entitled "Optical recognition of music symbols - A comparative study", the researchers used this as a basis to compare the average accuracy ratings for musical notations with the Concurrent Neuro-Fuzzy Systems [12].

Symbol	Concurrent Neuro-Fuzzy	Multilayer Perceptron
Sharp	0.9309	0.85
Flat	0.8614	0.84
Natural	8954	93
Minim	0.8825	82
8th Rest	0.8454	96
16th Rest	0.6689	n/a
32nd Rest	0.7591	n/a
Quarter Rest	0.7503	78
Accent	0.9223	85
Alto Clef	0.7322	n/a
Bass Clef	0.7843	n/a
Treble Clef	0.7659	40
Beam1	0.8146	85
Quaver	0.9161	51
Crochet	0.7853	3
Slur	0.7926	n/a
Ties	0.7588	n/a
Staccatissimo	0.9115	58
Average Accuracy	**82.1**	**81**

Nearest Neighbor	Support Vector Machines	Hidden Markov Models
0.98	0.98	84
0.99	0.98	0.97
99	98	91
97	96	73
100	100	92
n/a	n/a	n/a
n/a	n/a	n/a
100	97	90
99	99	91
n/a	n/a	n/a
n/a	n/a	n/a
92	90	94
98	95	90
86	89	64
75	40	22
n/a	n/a	n/a
n/a	n/a	n/a
100	100	100
93	**95**	**77**

Symbol	**Concurrent Neuro-Fuzzy**	**Fuzzy Logic**
Sharp	0.9309	0.5
Flat	0.8614	0.15
Natural	8954	0.15
Minim	0.8825	0.3
8th Rest	0.8454	0.3
16th Rest	0.6689	0.05
32nd Rest	0.7591	0.2
Quarter Rest	0.7503	0.05
Accent	0.9223	0.58
Alto Clef	0.7322	0.35
Bass Clef	0.7843	0.25
Treble Clef	0.7659	0.35
Beam1	0.8146	0.4
Quaver	0.9161	0.3
Crochet	0.7853	0.1
Slur	0.7926	0.6
Ties	0.7588	0.2
Staccatissimo	0.9115	0.35
Average Accuracy	**82.1**	**28.78**

Chapter 5

Conclusion and Recommendations

5.1 Conclusion

At the end of the study, the researchers can answer their objectives.

The entire Concurrent Neuro-Fuzzy system learned mostly from musical notation. The factors that was needed to identify a musical notation are the transition values, and these values are based mostly on how an individual draws their own musical notation. Some drawings are deformed or skewed and as a result, the Fuzzy System will not accept it as an input because it will not pass the membership function for a given classification.

The accuracy of the Concurrent Neuro-Fuzzy System averaged at 82%. It relied heavily on the data that was passed to it from the underlying Fuzzy Membership Functions. From the other Machine Learning algorithms, it was able to achieve a higher accuracy compared to Hidden Markov Models which had an average accuracy of 77%, however, it only a slight difference compared to the Multilayer Perceptron Neural Network, Support Vector Machines, and the Nearest Neighbor which got an average accuracy of 81%, 95%, and 93% respectively.

5.2 Recommendations

The speed of training the Concurrernt Neuro-Fuzzy System would take time since it requires training the Fuzzy system including Fuzzy Membership Rules, Membership Functions, getting the transition values, and training the Convolutional Neural Network Model. The researchers suggest to:

1. Try other types of Neuro-Fuzzy as there are more than 5 types of implementing this kind Neural Network. There is the Adaptive Network based Fuzzy Inference System (ANFIS), which teaches the neural network how to create the membership function, and there is also the Evolving Neural Fuzzy Network (EFuNN), which allows the membership functions to continue evolving and learning.

2. Have a bigger amount of dataset from different survey participants without having the need to use a data augmentor to create a larger data set with genuine data.

3. Implement a membership function also for musical notations that are skewed, distorted, or deformed.

4. Use more slices for transition values instead of just 3.

5. Include Sum of Pixels and Termination Widths for the universe of discourse for each classification.

Bibliography

[1] Adi, C. Handwritten character recognition using neural networks.

[2] Buckley, J. J., and Yoichi, H. Neural nets for fuzzy systems. *Fuzzy Sets and Systems 71*, 3 (1995), 265–276.

[3] Chaudhuri, A., Mandaviya, K., Badelia, P., Ghosh, S. K., et al. *Optical character recognition systems for different languages with soft computing.* Springer, 2017.

[4] Dalitz, C., Droettboom, M., Pranzas, B., and Fujinaga, I. A comparative study of staff removal algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 5 (2008), 753–766.

[5] Fuller, R. Neuro-fuzzy methods for modeling and fault diagnosis. *E" otv" os Lor'and University, Budapest VacationSchool* (2001), 1–22.

[6] Gonzalez, R. C., and Woods, R. E. Image processing. *Digital image processing 2* (2007).

[7] Kruse, R. Fuzzy neural network. *Scholarpedia 3*, 11 (2008), 6043.

[8] Larsen, J. Introduction to artifical neural network. *Section for Digital Signal Processing Department of Mathematical Modeling Technical University of Denmark, 1st Edition November* (1999).

[9] Nauck, D., Klawonn, F., and Kruse, R. *Foundations of neuro-fuzzy systems.* John Wiley & Sons, Inc., 1997.

[10] Novotnỳ, J., and Pokornỳ, J. Introduction to optical music recognition: Overview and practical challenges. In *DATESO* (2015), pp. 65–76.

[11] Pal, S. K., and Ghosh, A. Review neuro-fuzzy computing for image processing and pattern recognition. *International Journal of Systems Science 27*, 12 (1996), 1179–1193.

[12] Rebelo, A., Capela, G., and Cardoso, J. S. Optical recognition of music symbols. *International Journal on Document Analysis and Recognition (IJDAR) 13*, 1 (2010), 19–31.

[13] Vieira, J., Dias, F. M., and Mota, A. Neuro-fuzzy systems: a survey. In *5th WSEAS NNA international conference on neural networks and applications, Udine, Italia* (2004).

[14] Win, H. P. P., Khine, P. T. T., and Tun, K. N. N. Bilingual ocr system for myanmar and english scripts with simultaneous recognition. *International Journal of Scientific & Engineering Research 2*, 10 (2011).

[15] Zadeh, L. A., et al. Fuzzy sets. *Information and control 8*, 3 (1965), 338–353.

[16] Zeng, A. Optical music recognition cs 194-26 final project report.