ISLAMIC UNIVERSITY OF TECHNOLOGY

Alphabet recognition in unconstrained Air Writing using Depth Information

By

Robiul Islam (134601)

A thesis submitted in partial fulfilment of the requirements
for the degree of Master of Science in Computer Science and Engineering

Academic Year: 2014-2015

Department of Computer Science and Engineering

Islamic University of Technology.
A Subsidiary Organ of the Organization of Islamic Cooperation.
Dhaka, Bangladesh.

August 2018

Declaration of Authorship

I, ROBIUL ISLAM, declare that this thesis titled, 'Alphabet recognition in unconstrained Air Writing using Depth Information' and the work presented in it is my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Any part of this thesis has not been submitted for any other degree or qualification at this University or any other institution.
- Where I have consulted the published work of others, this is always clearly attributed.

Submitted By:

(Signature of the Candidate)
Robiul Islam (134601)
August 2018

Alphabet recognition in unconstrained Air Writing using Depth Information

Approved By:

Dr. Md. Kamrul Hasan
Thesis Supervisor,
Professor,
Department of Computer Science and Engineering,
Islamic University of Technology.

Dr. Muhammad Mahbub Alam
Head of the Department and Professor,
Department of Computer Science and Engineering,
Islamic University of Technology.

Dr. Md. Hasanul Kabir
Professor,
Department of Computer Science and Engineering,
Islamic University of Technology.

Dr. Mohammad Shoyaib
Professor,
Institute of Information Technology, University of Dhaka, Bangladesh

Abstract

In this thesis, we present a machine learning approach to recognize on-air writing of English Capital Alphabets (ECAs) using different feature is introduced include depth information. The hand finger's motion while writing the alphabet in the air was captured as depth images with the help of a depth camera. The depth images were then processed to track finger movements and after that smoothing procedure was applied to generate hand trajectory data. 11 point-wise features including depth value were calculated from the hand trajectory data which are also time series. Each air written alphabet is then compared with 26 alphabet templates using Dynamic Time Warping (DTW). The DTW distance features are normalized between 0 to 1 and used as features. So, a feature vector of 11x26 =286 normalized features and the appropriate class label was fed to Support Vector Machine for training and testing. 15 fold cross verification classification result provided an average accuracy of 55.4% with 15 users.

We also explored feature removal method based on a gain ratio. We removed the features that have the worst gain ratio. Iteratively 60 features were removed and the accuracies were compared. However, the best accuracy of 57.17% was found by removing eight features.

Keyword - Air Writing; Gesture Recognition; Depth Information; Time Series; Dynamic Time Warping; Support Vector Machine;

Acknowledgements

First of all I express my gratitute to the almighty for the uncountable blessings on us. I am thankfull to my supervisor for investing his valuable time and knowlege on me to help me overcome obstacles. Besides, my supervisor, I would like to thank the rest of the thesis committee, and all the members of Computer Science and Engineering department. Especially I must mention the name of Mr. Hasan Mahmud who contributed in ideas and writing and even worked with me like a peer. I again thank my supervisor for making the thesis interesting to me, for his inspiring talks and great assistance. Furthermore, I thank all users who participated voluntarily in data collection phase in the implementation of this work. Last but not the least, I would like to thank my family for raising me, mentoring me spiritually.
This work is partially supported by

ICT Division,
Ministry of Posts, Telecommunications and IT,
Government of the people's republic of Bangladesh

&

Systems and Software Lab (SSL)
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)

Contents

Declaration of Authorship i

Abstract iii

Acknowledgements iv

List of Figures vii

List of Tables viii

1 Introduction 1
- 1.1 Overview . . . 1
- 1.2 Significance of the Problem . . . 2
- 1.3 Research Challenges . . . 2
- 1.4 Thesis Contributions . . . 3
- 1.5 Organization of the Thesis . . . 3

2 Background Study 4
- 2.1 Gesture Recognition . . . 4
- 2.2 Unconstrained Air Writing . . . 5
- 2.3 Handwriting Recognition . . . 7
- 2.4 Depth Information . . . 8
  - 2.4.1 Depth Image . . . 8
  - 2.4.2 Microsoft Kinect . . . 9
- 2.5 Machine Learning Techniques . . . 9
  - 2.5.1 Support Vector Machine . . . 9
  - 2.5.2 Dynamic Time Warping (DTW) . . . 11
- 2.6 Features used in on-air gesture recognition . . . 13
- 2.7 Feature Removal Technique . . . 14

3 Proposed Approach 16
- 3.1 Image Acquisition . . . 16
- 3.2 Segmentation and Preprocessing step . . . 17
- 3.3 Feature Extraction and Classification . . . 20

3.3.1 Point Vector 20
3.3.2 Depth Value 21
3.3.3 Point-wise Distance 21
3.3.4 Theta Value 21
3.3.5 Velocity 22
3.3.6 Log Normal Probability Density function 22
3.3.7 Freeman chain code 4,8,16 23
3.3.8 DTW Distances as Derived Features 23
3.4 Post Classification: Feature Removal 25

4 Experimental Results and Evaluation 30
4.1 Result Analysis 31
4.1.1 True positive rate (TPR) 31
4.1.2 False positive rate (FPR) 31
4.1.3 Matthews Correlation Coefficient (MCC) 31
4.1.4 F-Measure 31
4.1.5 Classification Accuracy 32

5 Conclusion 37

A Appendix 38

Bibliography 40

List of Figures

2.1 Basic strokes for English characters 6
2.2 Sample RGB image 8
2.3 Sample Depth Image 8
2.4 Draw a line that separates black circles and blue squares. 10
2.5 Sample cut to divide into two classes. 11
2.6 A simple DTW figure of two signal 12
2.7 Comparison of two curves using one to one comparison and Dynamic Time Warping. As can be seen, the DTW-comparison is more intuitive than the one to one comparison (images retrieved from http://ciir.cs.umass.edu/ trath/prj/hw retr/wordspot retr.html). . . 12

3.1 Proposed Approach in block diagram 17
3.2 RGB image with depth value 18
3.3 Hand segmentation 18
3.4 Writing of 'A' 18
3.5 'A' after smoothing 19
3.6 DTW signal for character A 19
3.7 point-wise distance 22
3.8 point-wise distance 23
3.9 Theta Value 24
3.10 Velocity point 25
3.11 Log normal Probability density function point 26
3.12 Free man chain code 4 point 27
3.13 Free man chain code 8 point 28
3.14 Free man chain code 16 point 29

List of Tables

3.1 Feature Table 21

4.1 Confusion Matrix 32
4.2 Classification Accuracy by removing features 33
4.3 Result Table 34
4.4 Classification Accuracy by removing features one by one 35
4.5 True positive rate comparisons between different features and our approach 36

A.1 Rank vs Feature 38

Dedicated to my mother ...

Chapter 1

Introduction

In this chapter, we first present an overview of our thesis that includes the significance of the problem and the problem statement in detail. Besides, we also discuss the different research challenges that we faced in the whole scenario. After that, we present our thesis objectives and contributions. The chapter ends with a short description of the organization of this thesis.

1.1 Overview

Different computer interfaces are used to give commands nowadays, for the communication between humans and computers. Most of these are the particular devices which are designed for the human and machine fit. The development of computer vision technologies make it possible to approach towards the interface problem from a human perspective, establishing the communication between the computer and human more natural. The first task is to develop a system which will recognize hands for enabling real-time hand gesture recognition (HGR) via depth image. Depth image contains depth value including RGB values. From the hand movement, we draw a real-time hand shape in the form of a graph. This graph can provide hand movements or change patterns. Recongnizing the hand movement trajectory as an air written English Capital Alphabet (ECA) is the problem to solve.

The idea of recognizing 'air writing' was incubated by computer scientists at the Karlsruhe Institute of Technology[1]. Air writng means recognizing alphabets written on the air. Alphabet recognition is a part of broader gesture recognition

research [2]. Air writing can be used as another input modality to the computer systems.

Air writing might seem to be similar to online hand writing recongition [3]. In online hand writing the user can lift his/her hand from the touch pad. But in air writing the system cannot differentiate which movements are part of writing and which movements are not. Consequently many different extra strokes are mixed with the actual writng which complicates the recognition process.

1.2 Significance of the Problem

Depth video based writing recogntion is natural and unconstrained. The use of depth information makes the hand tracking easier without ambiguity. While writing in the air the hand may be near to face or the body and their similar colour might be confusing. To overcome the problem many researchers have used special markers[1] around the writing finger. A special version of air writing can be to write on a surface (which is not touch pad), beause people feel natural writing on a surface. The use of depth information will help clearly segment the hand where regular cameras will fail.

1.3 Research Challenges

Translating signals into English alphabets is the Challenge in this thesis. When somebody writes an alphabet, he/she writes it as strokes. The best algorithm of air writing should be able to segment the strokes accurately from the air gestures. However, in air writing many extra movements of the user match with perfect strokes[4] and hence become part of the writing. Initially we investigated into the approach and discovered those piculiarities discussed. Then we concentrated on finding the appropriate time series featuers from the whole tragectory of an alphabet writing. Dynamic Time Warping (DTW) is the algorithm to match two time series data. We used DTW to compare an alphabet signal to the alphabet templates. The dicision taken from the DTW distances was not that accurate even with small number of users[5] . When we increased the number of users from 5 to 15, the accuracy droped. Then we looked for other features besides point vector such as point wise distance, theta value of points, velocity, log normal probability density and freeman chain code that are used regularly in online hand

writing recognition. We also included depth information as feature. The details of the features are discussed in later sections. Each of the 11 features were a time series. The DTW distances of the 11 time series features compared with the alphabet templates were directly used for classification. The result was almost half of the result that we reported. After reading the literate for quite a long time, we discovered that the phase shift in signals reduces the accuracy of recognition if the geometric shapes are important features [6, 7] . In such situation, the literature suggested to use all pair comparision of the training data and use the DTW distances for learning in another classifier. The result we reported is the best of all possible experiments we performed. Because of all pair comparsion with templates and using the distances in learning, feature reduction techniques also could not increase the accuracy much.

1.4 Thesis Contributions

We took the initiative to make the air writing unconstrained by using depth camera. Previous work[1] used wearable sensor for capturing air writing motions. we have created a unique dataset that we shared with the research community for further research. We have introduced the machine learning approach of using DTW distances as features in the domain of air writing where retaining shape information in time series distance comparision is necessary. We also explored a feature dimension reduction technique and could further increase the accuracy by about 2 percent.

1.5 Organization of the Thesis

The rest of the thesis is be organized as follows: In chapter 2 we focused on literature review to know the current state of the problem. We also discussed some background literature that may help to understand the later part of the thesis. In chapter 3 we focused on our proposed methods and algorithms. In chapter 4, we discussed the experimental setup and results.

Chapter 2

Background Study

In this chapter, we discuss the related works on on-air gesture recognition. There is huge scope of research in recognition of the gesture produced on-air like writing English alphabets through hand gestures. Then we describe the unconstrained air writing which means writing English characters or numerical digits in a natural environment using bare hand finger movement without prior training or without any guidance. After that, we describe the state-of-the-art gesture recognition techniques, use of depth image, and varieties of related features suitable for on-air gesture recognition and the relevant research works on the techniques we used in our approach.

2.1 Gesture Recognition

Human gesture is an important input modality for communication with computers using gesture-based interfaces. In hand based gesture recognition technology, a camera (typical stereo camera) reads the hand movement data, perform the hand tracking and then recognize a meaningful gesture to control any devices or applications. For example, a person clapping his hands together in front of a camera can produce the sound of cymbals being crashed together when the gesture is fed through a computer. It has long been considered a promising approach to enable a natural and intuitive method for human-computer interactions for various computing domains, tasks, and applications. The first gestures that were applied to computer interactions date back to the PhD work of Ivan Sutherland [8], who demonstrated Sketchpad, an early form of stroke-based gestures using a light pen to manipulate graphical objects on a tablet display. This form of

gesturing has since received widespread acceptance in the human-computer interaction (HCI) community, inspiring the stroke-based gesture interactions commonly used for text input on personal digital assistants (PDAs), mobile computing, and pen-based devices [9, 10]. Since then, the notion of using gestures to facilitate a more expressive and intuitive style of computer interactions has gained popularity among researchers seeking to implement novel interactions with computers. Gloves augmented with electronic motion and position sensors were abstract developed to enhance interactions with virtual reality applications, enabling users to manipulate digital objects using natural hand motions [2, 11, 12] and polhemus motion sensors tracked arm movements for controlling large screen displays from a distance, presented by Bolt [13] in the Put That There system. By the mid-1980s, computer vision technology was gaining popularity within the computing sciences, however it was not until the early 1990s that Freeman and Weissman [14] first demonstrated a vision-based system that enabled gestures to control the volume and channel functions of a television. While this work represented a new direction of perceptual, device-free gestures, computer- vision interactions to date, remain a technique restricted to laboratory studies.

2.2 Unconstrained Air Writing

In hand gesture recognition research, air writing is a prominent and difficult topic to work with. Air writing [4] means gesture based writing on the air through movement of hand fingers by which a computer system can recognize characters and other symbols in natural handwriting. Characters can be viewed as sequence of strokes as mentioned in [4]. The capital alphabet A, for instance is composed of three strokes mainly /, \ and -. If the discrete strokes can be pulled out from the seemingly continuous movement of the hand, it is possible to infer the characters. To this end, they have analyzed the English alphabets and constructed a basic set of strokes, as in Figure 2.1. Here, the main challenge is while writing on air each movement of hand becomes a strokes. So, a lot of noises are accumulated into the writing process.

Amma et. al. [1] showed how a wearable device can recognize hand gesture for air writing. The Airwriting glove fits on the back of the hand. It has motion sensors, accelerometers and angular rate sensors equipped with smart phone and the signals are recorded and transmitted via Bluetooth. Wearable hand motion tracking system captures movement signals using accelerometer and gyroscope.

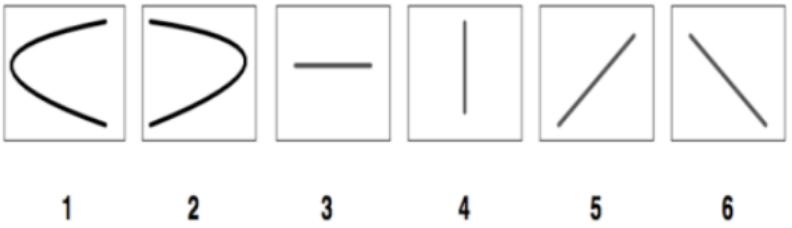


FIGURE 2.1: Basic strokes for English characters

However, converting the acceleration signal into important features to recognize strokes can be erroneous. Moreover, wearing a special device makes the air writing system cumbersome and not natural. Once it has determined that letters are indeed being drawn, the computer then starts identifying the individual letters. The program incorporates statistical models of the unique signal patterns for every letter in the alphabet and can account for differences in individual writing styles. This idea drove the development of air writing developed by computer scientists at the Karlsruhe Institute of Technology in Germany [15]. Sensors attached to a glove record hand movements, a computer system captures relevant signals and translates them into text, which can then create an email, text message, or any other type of mobile app.

Kim et. al. [16] showed a way to recognize different peoples handwriting on continuous images based on similarity of the different shapes of characters or digits based on the strokes and the ligature model. They did not used the concept of bare hand writing without using any special input pen. They tried to virtual 3D characters from 2D shapes using ligature model and then used Baysian model to recognize real on air writing. In our approach, we are using unconstrained environment to write English alphabets, creating training model using real on on-air writing gestures. We are using the character shape information as features in the form of time-series curves.

The system shown in [15] can recognize complete sentences written in capital letters and presently has a vocabulary of 8000 words. Developers claim the system has an error rate of 11% .

2.3 Handwriting Recognition

The technique where a computer system can recognize characters and other symbols written by hand in natural handwriting from sources such as printed physical documents, pictures, or to use handwriting as a direct input to touch-screen and then interpret it as text. This technique is generally known as Handwriting Recognition which has gathered a lot of attention in recent years [17].

In [18], authors used Hidden Markov Model that can be employed to recognize typewritten documents. Three documents (old memo, old war letter and newly typewritten essay) were used to create three datasets of typewritten characters each consisting of 1995, 702 and 2049 characters respectively. The research result showed that, recognition accuracy values are 94.88%, 91.45% and 97.24% for old memo, old war letter and newly typewritten essay datasets respectively.

There has been significant growth in the application of offline handwriting recognition during the past decade. Few of those are mail sorting, bank processing, document reading, postal addresses recognition, handwritten address interpretation and writer identification. Handwritten address interpretation is the task of assigning a mail piece image to a delivery address by determining the country, state, city, post office, street number, the firm or the persons name [3]. Bank processing includes recognition of legal amount, date and signature. A complete bank check recognition system for industrial application is described in [19]. Writer Identification deals with the establishment of authorship of a document for which some prototypes tool sets for document examination. As on line recognition refers to methods dealing with the automatic processing of a message as it is written using a digitizer. Over the years these methods have evolved from academic exercises to developing technology driver applications such as pen based computers, sign verifiers, developmental tools as well as in home safety using handwritten pattern recognition system. The concept of pen based computer was proposed by Kay. Signature verification refers to the comparison of test signature with reference specimens. The most promising application to be emerged will be related to long distance authorization, personalization, tracking of money and document and much more. Developmental tools includes educational software for teaching handwriting to children, LCD with digitizer, digitized tablets [3].

2.4 Depth Information

Depth information is the distance value from the user to the depth camera (e.g. Microsoft Kinect, Intel Realsense, etc.). This information help to generate depth image and used as skeleton features to different gesture recognition systems [20].

2.4.1 Depth Image

The depth image has a standard size, but for every pixel, it is known that how particular distances away the object are from the camera. 3D image is considered as depth image which has depth value. For those reasons, we can quickly calculate the length of an object. 3D reconstruction is the method through which shape and appearances of real objects are captured from a set of 2D images. It is widely used in fields such as computer vision, computer graphics, 3D reconstruction, and robotics. If we consider Figure 2.2 and 2.3, the Figure 2.3 represent depth value of the Figure 2.2. Using those information we can easily calculate of distance value. This technology has a wide variety of application, from augmented reality in computer game and app to robot interaction and self-driving car.

FIGURE 2.2: Sample RGB image

FIGURE 2.3: Sample Depth Image

2.4.2 Microsoft Kinect

In our thesis we have used Microsoft Kinect V1 , also known as Xbox-360 sensor. The device features an RGB camera, depth sensor and multi-array microphone running proprietary software, [21] which provide full-body 3D motion capture, facial

recognition and voice recognition capabilities. The depth sensor consists of an infrared laser projector combined with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions.[22] The sensing range of the depth sensor is adjustable, and Kinect software is capable of automatically calibrating the sensor based on game play and the players physical environment, accommodating for the presence of furniture or other obstacles.

Described by Microsoft personnel as the primary innovation of Kinect, the software technology enables advanced gesture recognition, facial recognition and voice recognition. According to information supplied to retailers, Kinect is capable of simultaneously tracking up to six people, including two active players for motion analysis with a feature extraction of 20 joints per player. However, PrimeSense has stated that the number of people the device can see (but not process as players) is only limited by how many will fit in the field-of-view of the camera [22].

2.5 Machine Learning Techniques

Machine learning technique deals with the classification tasks to classify a new test sample to a labelled one, if it is a supervised learning. There are problems for which the machine learning method need to deal with variable length data objects in case sequential data classification. The variable length sequential data need to be converted in to a suitable way so that the supervised learning algorithm (e. g. Suppoert Vector Machine (SVM) can be applied. There are other learning techniques like semi-supervised learning, statistical-based learning etc [23].

2.5.1 Support Vector Machine

SVM is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional

space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

The statistical learning theory provides a framework for studying the problem of gaining knowledge, making predictions, making decisions from a set of data. In simple terms, it enables the choosing of the hyperplane space such a way that it closely represents the underlying function in the target space [24]. In statistical learning theory the problem of supervised learning is formulated as follows. We are given a set of training data (x1,y1)... (xl,yl) in Rn R sampled according to unknown probability distribution P(x,y), and a loss function V(y,f(x)) that measures the error, for a given x, f(x) is "predicted" instead of the actual value y. The problem consists in finding a function f that minimizes the expectation of the error on new data that is, finding a function f that minimizes the expected error: V(y,f(x)) P(x,y) dx dy [24] In statistical modeling we would choose a model from the hypothesis space, which is closest (with respect to some error measure) to the underlying function in the target space. More on statistical learning theory can be found on introduction to statistical learning theory [25].

Suppose there are given a plot of two label classes on a graph as shown in figure 2.4. It might have come up with something similar to following figure 2.5.It reasonably separates the two classes. Any point that is left of the line falls into black circle class and on the right falls into the blue square category. Separation of classes, Thats what SVM does in simple. It finds out a line/hyperplane (in multidimensional space that separates outs classes).

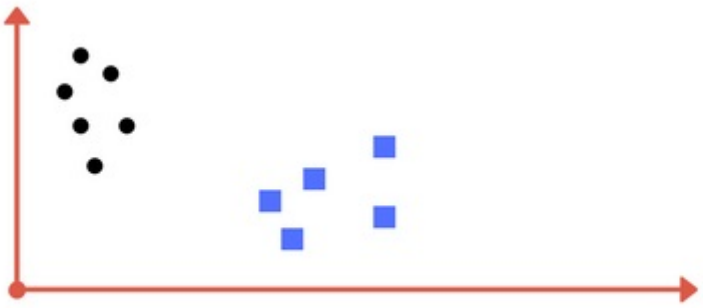


FIGURE 2.4: Draw a line that separates black circles and blue squares.

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new

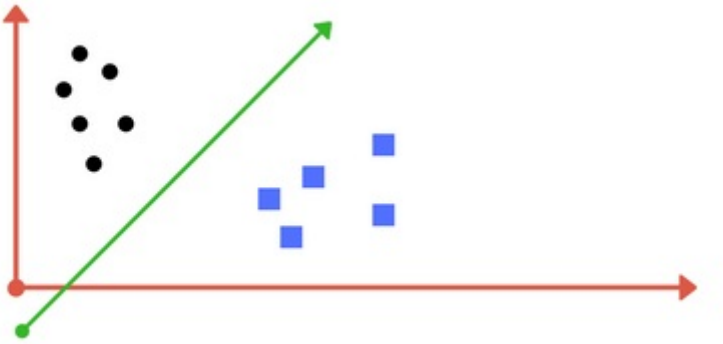


FIGURE 2.5: Sample cut to divide into two classes.

examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

2.5.2 Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) has since long been a popular technique for matching variable length signals (not only speech). Popularity is due to the simplicity and elegance of the technique. DTW is motivated by the fact that manipulating the duration of sounds within bounds is allowed without having an impact on meaning (of most sounds). In practice the matching is not done on the time domain signal but on short-time spectra computed in a sliding window approach. In speech recognition it is common practice to include apart from the spectra also time derivatives of these in the feature vector. This should give sufficient emphasis to the important transients in speech. In figure 2.6 we have shown how DTW distance work for both signal.

This thesis is about handwriting recognition, and a technique called Dynamic Time Warping (DTW) that can be used for handwriting recognition. We believe that the technique can be of importance for the handwriting recognition research: it gives a relatively new view on the data, and thus can be an addition to existing systems (it can be combined with other systems in a so called Multiple Classifier System). It can also be used in a standalone handwriting system that can read human handwriting (these systems are called handwriting classifiers or simply classifiers). As can be seen in Figure 2.7, the DTW-algorithm is able to compare two curves in a way that makes sense to humans (we call this sense intuitive) [26], because, at a very basic level, handwritten characters are nothing more than

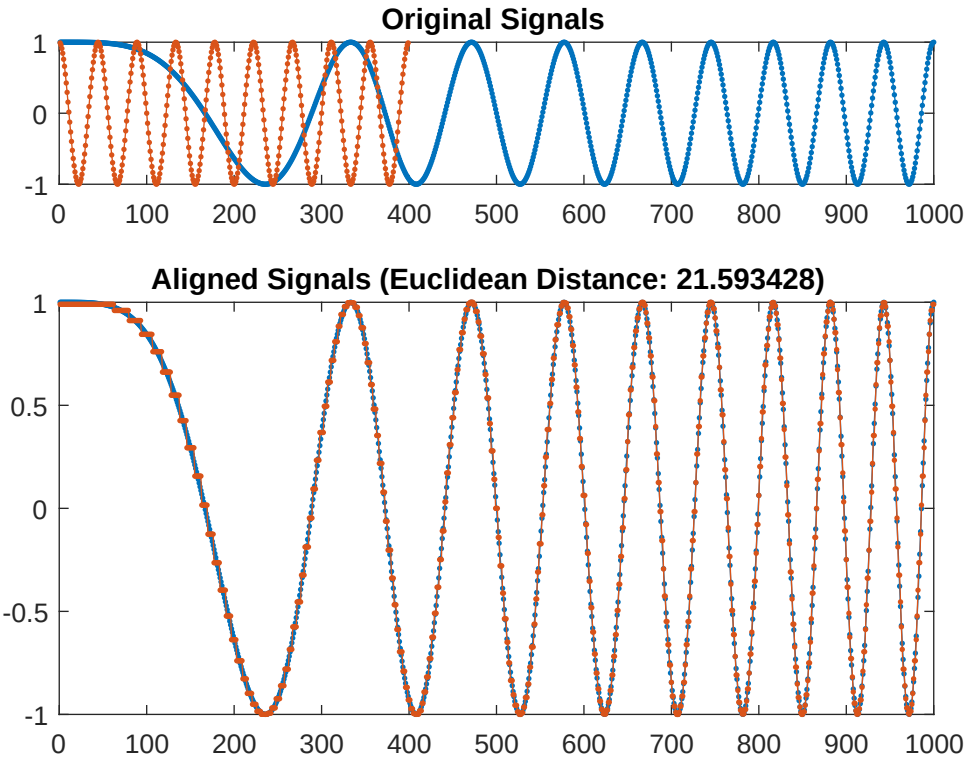
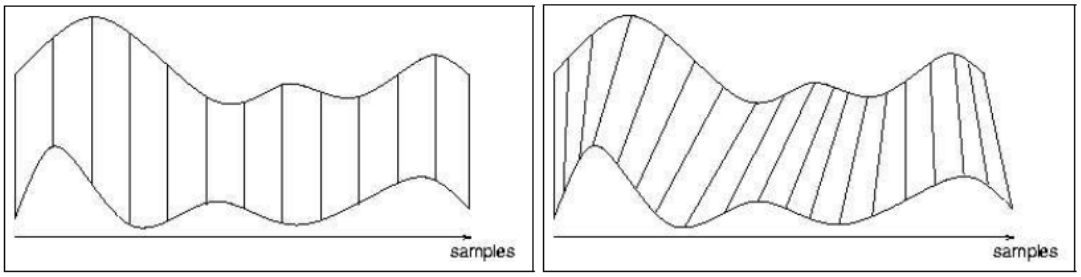


FIGURE 2.6: A simple DTW figure of two signal



(a) One to one comparison (b) DTW-comparison

FIGURE 2.7: Comparison of two curves using one to one comparison and Dynamic Time Warping. As can be seen, the DTW-comparison is more intuitive than the one to one comparison (images retrieved from http://ciir.cs.umass.edu/ trath/prj/hw retr/-wordspot retr.html).

special cases of curves, we believe that DTW can compare characters in a way that is similar to the way humans compare characters, or at least generates the same results.

The main limitations of DTW algorithm is that, it does not consider the phase differences between the reference signal and the test signal [6]. This may lead to the lower accuracy problem for shape-based matching applications. To overcome this problem the researchers in [6] have applied two-step DTW-SVM classification where in the first step the features were presented as DTW distance measure. The have represented each sample as a DTW distances to all other samples. In the second step, they have considered the DTW matrix as the input of a two-class SVM classifier.

As our alphabet recognition technique tries to match the shapes between English characters, so, rather than using DTW-based recognition, we can fed the DTW distances as features in the learning algorithm like SVM.

2.6 Features used in on-air gesture recognition

In this section, we discuss the important related features that are used to recognize on-air gestures. We describe the mathematical definition of the related features and their uses in the related research. In [27] the hand fingertip positions, finger joint point, 3D positions were used for fingertip tracking and from 3D hand written trajectory they have extracted 2D and 3D features. The features includes the fingertip positions and their derivatives, velocity, acceleration, slop angle, path angle, log radius of curvature etc. A time series of 16 dimension features is used to represent 3D handwritten. Then, the DTW distances were calculated between two 3D handwritten features. Those distances were used as the feature vectors for SVM. We have extracted the point vector of the gesturing finger and used it as a feature. We have calculated the point-wise distance, the theta value of the corresponding point vector in polar coordinate, the velocity, the depth coordinate, the log normal probability density function and added to our feature list.

In [7], the author used Freeman chain code methods that quantizes the point wise angles. For example, if I want to use freeman code 4, any point is angled with its previous point and quantized to one of the four angles 45,90, 135, 180 degrees and given the level 1,2,3,4. We use 4/8/16 freeman chain code in this topic. For

example in free man chain code 4 for every quadrant we give a level. Evey data from 0 to 90 is given a single label and this will goes on [7].

In [28], the histogram of oriented gradients (HOG) were used for gesture recognition. It is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

HOG basically evaluates image gradient values in a dense grid form [28]. HOG divides an image into cells of certain pixel size. The gradient magnitude G and gradient 9 are computed for all pixels in cells using equation 2.1.

$$|G| = \sqrt{((I_x)^2 + (I_y)^2)} \tag{2.1}$$

Each pixel within the cell casts a weighted vote for an orientation based histogram bins corresponds to the values found in the gradient computation. The histogram bins are evenly divided over 0 to 180. Block level histograms are normalized later for the same purpose. All normalized block histograms are concatenated to form the entire HOG feature vector.

Another important feature that the researchers apply is the Gain Ratio. It evaluates the valuable attribute by measuring the gain ratio with respect to the class as described in [29]

$GainR(Class, Attribute) = (H(Class) - H(Class|Attribute))/H(Attribute)$
Valid options are: treat missing values as a separate value.

In the next section we describe our proposed approach in detail.

2.7 Feature Removal Technique

Input variable selection is the most important part of the model selection process, because it interprets the the data modeling problem by specifying those explanatory variables most relevant to the target variables. However, exhaustive search for a set of optimal input variables is exponentially complex. Some heuristic search strategies are needed to select a set of suboptimal input variables. Three search

strategies frequently used in selecting regressors for linear models are Forward Selection, Backward Elimination and Stepwise Regression [30]. The same strategies can be applied to select inputs for nonlinear models. To guide the search, we need a saliency criterion to rank the input variables according to their relevance to the target variables. We also need a selection criterion to evaluate the relevance for a set of selected input variables. The saliency and selection criteria are often different.

In post result analysis we used [31] on the model-independent approach for input variable selection based on joint mutual information (JMI). The increment from MI to joint MI is the conditional mutual information.

We also used Chi2 for removing features that select minimum number of feature which may produced best output, [32] a simple and general algorithm that uses the $\tilde{\chi}^2$ statistic to discretized numeric attributes repeatedly until some inconsistencies are found in the data, and achieves feature selection via discretization. The empirical results demonstrate that Chi2 is effective in feature selection and discretization of numeric and ordinal attributes.

Chapter 3

Proposed Approach

Any machine learning research must have data collection i.e. Image Acquisition, preprocessing step which may include segmentation, feature extraction and then classification. Sometimes a post processing step may be used for feature dimension reduction or for doing other analytical tasks. Figure 3.1 shows the simple overview of the whole thesis. We describe the steps sequentially and highlight our contribution in context.

3.1 Image Acquisition

Collection of data is always very important and tough job. First we tried to find any bench mark dataset. Unfortunately no dataset are available. So, we decided to collect dataset.

In air writing users write on a imaginary writing board. To facilitate unconstrained writing, we did not impose any restriction to the user, such as 'write slow' or 'try to write perfectly'. We placed a depth camera (Microsoft Kinect) in front of the users and told them to write a letter on an imaginary blackboard. Each user used his own writing style or font and the size and speed of writing varied. Hence the dataset varied widely. We asked every user to write from A to Z in order. Then we isolate every alphabet, with depth and RGB value, based on image signal. Usually user take a pause between writing two letters. We took data from 22 users from where 15 user data were used for experiment. Apart from this 15 users one best user's data were used as template. The rest of the user data were not usable because of very few number of points in their alphabets. We share the dataset

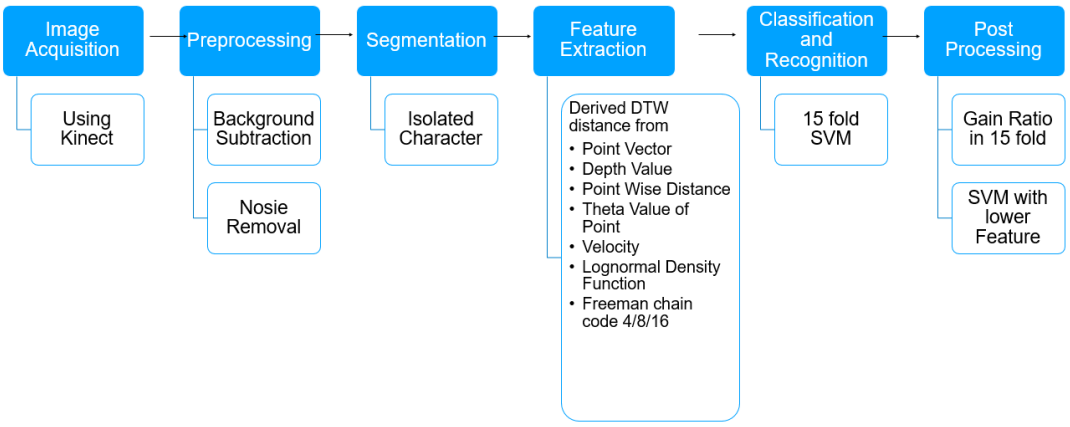


FIGURE 3.1: Proposed Approach in block diagram

with the research community for extending the research. We consider the dataset as one of our contributions.

3.2 Segmentation and Preprocessing step

We assume the hand is the front-most object while the user is writing on the air. In Kinect camera depth value are given in millimeter. For every image pixel we can tell from depth matrix how far the object is from camera. Regularly we write in alphabet on air and hand is always in front of the body. In deferent part of body distance and background noise we got over thousand plus depth values. From those data we take smallest 10 depth values which mainly covers only hand. Using those ten depth values we could separated hand as front part from the body. We have applied this step to every single image.

We have preprocessed the images to track the hand trajectories. It was done by first separating the hand from the background by using depth information. Then the middle pixel of the hand was calculated. The middle point movement was taken as the writing point of the letter for the future recognition. So, for each air writing, we have traced out the written points (x,y) and their corresponding depth values (d), from now on represented as points (x, y, d) . The hand motion is tracked from image to image, which generates a series of points (x, y, d). Those set of points are actually the time series information of the particular alphabet. As the hand movement is noisy, the time series data is smoothed using moving average filter [33]. The use of moving average filter has replaced the zig-zags with

straight lines and reshaped the angles. Moving average is a new sequence defined from a signal ,a by taking the arithmetic mean of subsequences of n terms show as equation 3.1 where S is new signal.

$$S_i = \frac{1}{n} \sum_{j=1}^{i+n-1} a_j \tag{3.1}$$

We did it for all 26 English capital alphabets. The process of writing "A" and its corresponding time series is shown in figure 3.2 , 3.3, 3.4 , 3.5.

FIGURE 3.2: RGB image with depth value

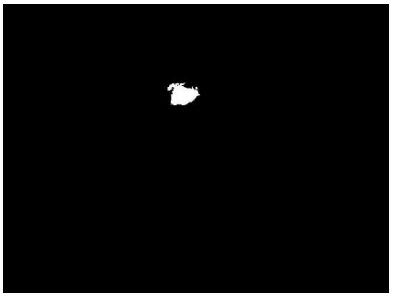


FIGURE 3.3: Hand segmentation

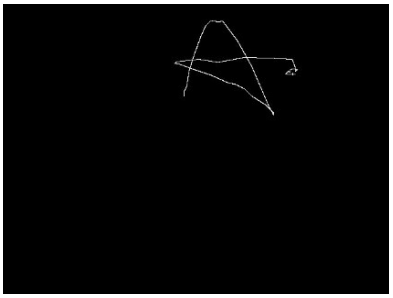


FIGURE 3.4: Writing of 'A'

In our approach, we do not ask for any special wearable device so that the user can write naturally without any obstruction. Algorithmically, previous approaches

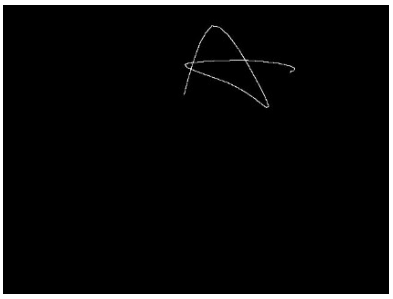


FIGURE 3.5: 'A' after smoothing

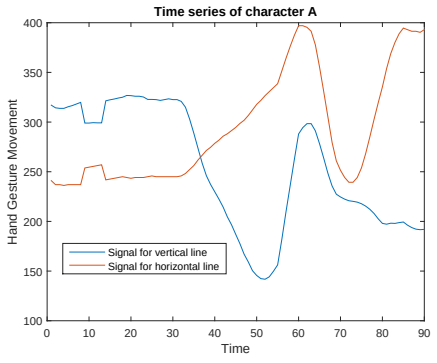


FIGURE 3.6: DTW signal for character A

studied air writing by converting them into strokes. While writing in the air users puts pause unintentionally or bends abnormally and thus creates extra strokes into the air written characters. However, recent data mining algorithms enable us to study a gesture signal such as air written character as a time series shown in Figure 3.6, information which can be matched with standard time series character templates where original alphabets image shown in Figure 3.5.

Algorithm 1 Dataset preprocessing and smoothing

```
1: function DataPreprocessing(AlphabetData)
2:     Create a empty signal S
3:     while still a file in AlphabetData do
4:         background subtraction using depth information
5:         find the middle pixel
6:         add this pixel and corresponding depth value to S
7:     end while
8:     smooth S using moving average filter
9:     return S
10: end function
```

3.3 Feature Extraction and Classification

The features are one of the most important parts of this thesis. Features and classification are inter-related. Hence we put the sections together to ease of description. A short summary is shown in Table 3.1.

After converting the air written alphabet to a time series of of x, y and d; the task is now how to classify them. As finding stroke feature proved to be very difficult, we thought of classifying based on time series data. So, we investigated the the use of DTW as the classifier. Our earlier work [5] was matching 2D trajectory (x,y) of an alphabet with templates and come up with a decision based on distance using the equation 3.2.

$$classified\,class\,label\,(trajectory(x,y)) = argmax(dist(trajectory(x,y), template(x,y))) \tag{3.2}$$

The decision taken from the DTW distances was not that accurate even with small number of users [5]. When we increased the number of users from 5 to 15, the accuracy reduced to half. Then we looked for other features besides point vector such as point wise distance, theta value of points, velocity, log normal probability density and freeman chain code that are used regularly in online hand writing recognition. We also included depth information as feature. Each of the 11 features were a time series. The DTW distances tof the 11 time series features compared with the alphabet templates were directly used for classification. Still The result was almost half of the result that we reported. After reading the literate for quite a long time, we discovered that the phase shift in signals reduces the accuracy of recognition if the geometric shapes are important features [6] . In such situation, the literature suggested to use all pair comparison of the training data and use the DTW distances for learning in another classifier.

3.3.1 Point Vector

We have taken raw point core for every alphabet. If we draw that character to an image matrix, we find a visual alphabet, but that loses the movement and rotation of the user that is why this feature is significant to this method. The hand movement trajectory are smoothed by moving average filter as we discussed

TABLE 3.1: Feature Table

Feature	Description
Feature 1 and 2	Point vector of alphabets
Feature 3	Depth vale of point
Feature 4	Point wise distance of point vector
Feature 5	Theta value of point
Feature 6	Velocity of point
Feature 7 and 8	Log normal probability density function calculation mean and standard deviation of data point
Feature 9	Freeman chain code of 4
Feature 10	Freeman chain code of 8
Feature 11	Freeman chain code of 16

in previous section. The point vector generates 2 time series features: one for x-dimension and one for y-dimension.

3.3.2 Depth Value

Depth value was extracted from the hand trajectory and smoothed. As there as less movements in the depth, other derived features such as velocity were not calculated from the depth information and use as a feature. However, if some user writes in angular plane, those derived features might be useful. In figure 3.7 shown that the difference between point vector and depth value for our dataset.

3.3.3 Point-wise Distance

This is the euclidean distance of consecutive two trajectory points (x,y). In figure 3.8 shown that the difference between point vector and point wise distance.

3.3.4 Theta Value

This feature helps to measure angular coordination and also pixel-wise angular distance which helps to generate data point that helps to measure angular coordinate. In figure 3.9 shown that the difference between point vector and Theta Value.

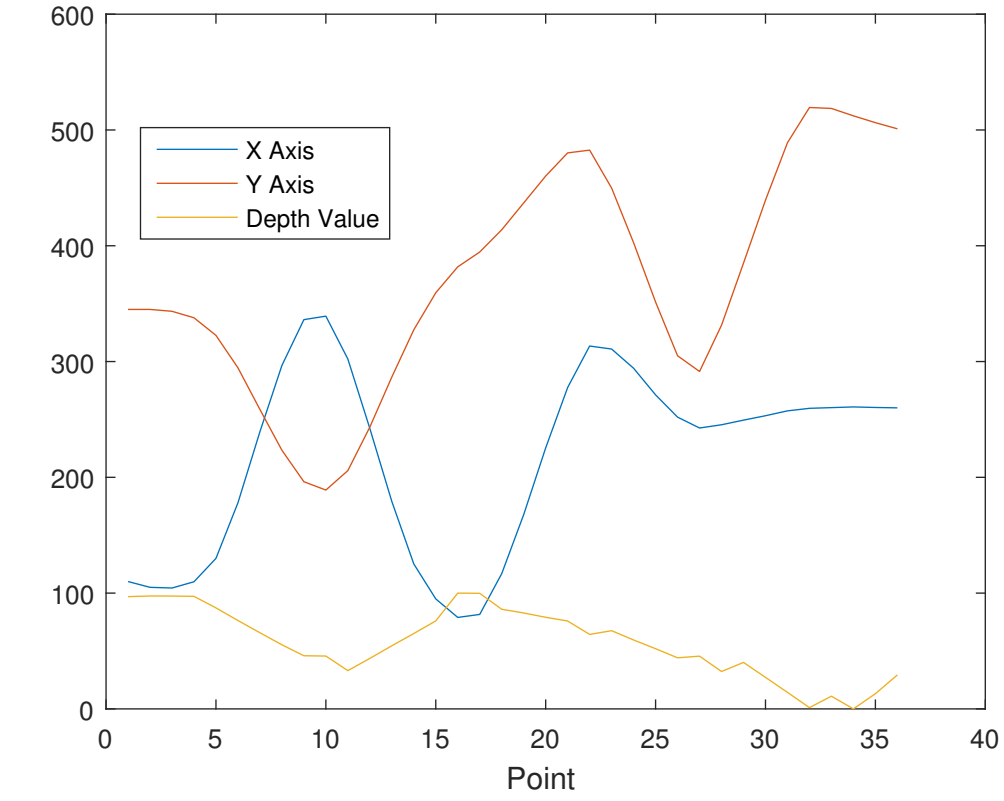


FIGURE 3.7: point-wise distance

3.3.5 Velocity

This feature helps to generate data point from point wise distance which shows the speed of that distance either forward or backward.In figure 3.10 shown that the difference between point vector and velocity.

3.3.6 Log Normal Probability Density function

This function based on the average and standard deviation. This feature shows us the overview of the whole dataset in a single row. In figure 3.11 shown that the difference between point vector and log normal probability density function both vertical and horizontal. We get two time series features from this. The Log normal density function generates 2 time series features: one for x- dimension and one for y-dimension.

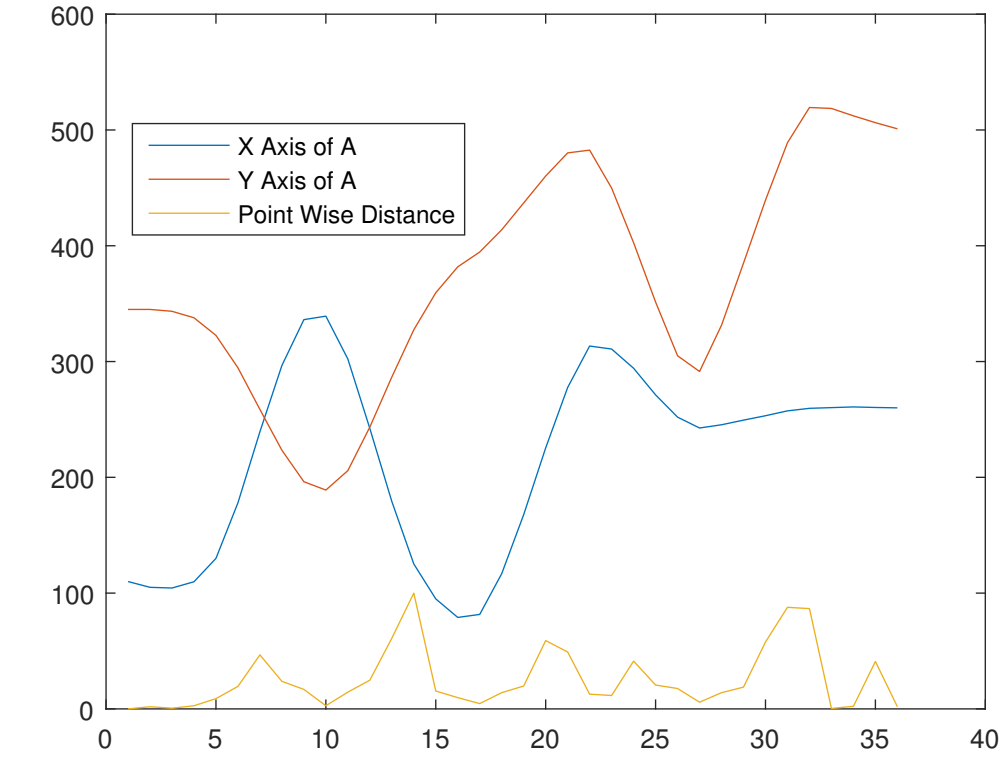


FIGURE 3.8: point-wise distance

3.3.7 Freeman chain code 4,8,16

This function will generated or converted whole dataset into some supertitle points. In this case we user freeman chain methods [7]. We take points consecutively and convert angles in degrees, returned as a scalar, vector, matrix, or N-D array. These angles correspond to the points defined by X and Y, and they lie in the closed interval [180,180]. In figure 3.12 shown that the difference between point vector and Freeman chain code 4. In same way figure 3.13 and 3.14 is discuss for 8 and 16. Difference of those figure is that a dataset given by any number is divided by only this value. Free man chain code generates a string to be matched with the template. 4,8 and 16 freeman chain code generates 3 time series features.

3.3.8 DTW Distances as Derived Features

After separating template and user data from total dataset we transfer every image to a signal that is shown in figure 3.6. We have created every signal to a feature

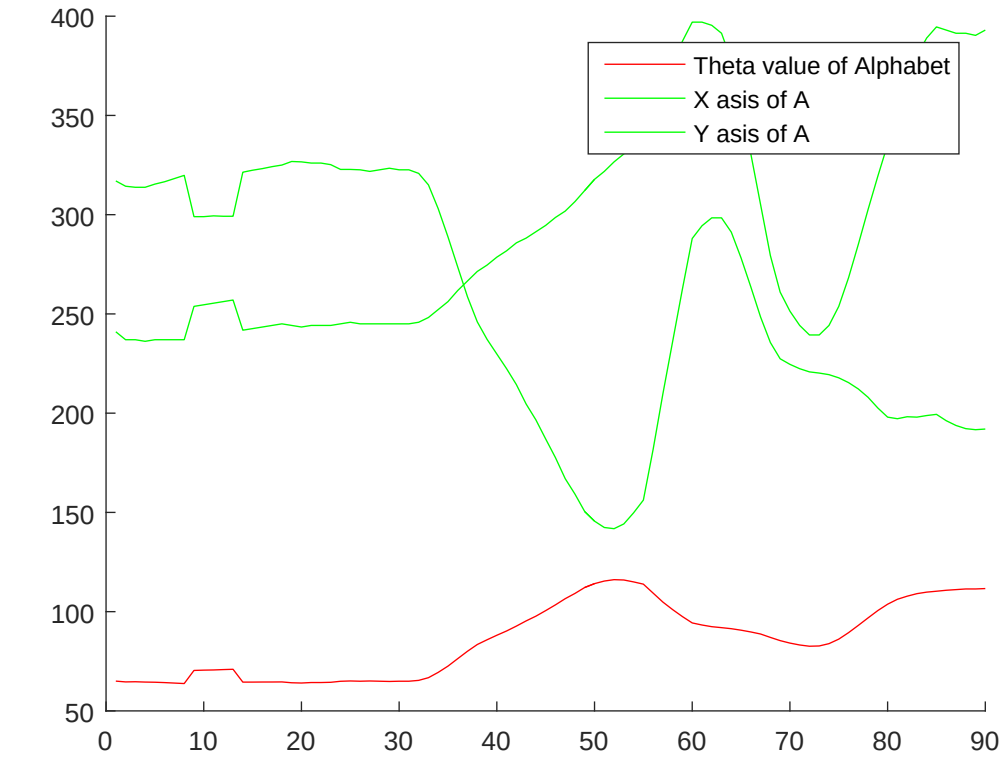


FIGURE 3.9: Theta Value

vector for every one. The list of 11 features are given table 3.1.

DTW gives us a minimum distance between time series. When a user write an alphabet imagine a blackboard, does not round up with specific length data point. So, applying this algorithm is very helpful in this scenario. So we find a minimum distance of two alphabets. This distance shows us how much near or far in every character is. Figure 2.6 shows the concept of DTW signal.

In our proposed approach we compare an alphabet point vector to all alphabets point vector to templates using DTW algorithm. DTW algorithm give us a minimum distance value comparing both data point. Comparing 11 time series features of an alphabet with corresponding features of the template, gives 11 distances. Comparing an alphabet with all 26 templates generate $11 * 26 = 286$ distance features. The class label for these 286 distance features are given as the the alphabet in consideration.

$$F_A = \{F_{A_1}, F_{A_2}, F_{A_3}, ..., F_{11}\} \tag{3.3}$$

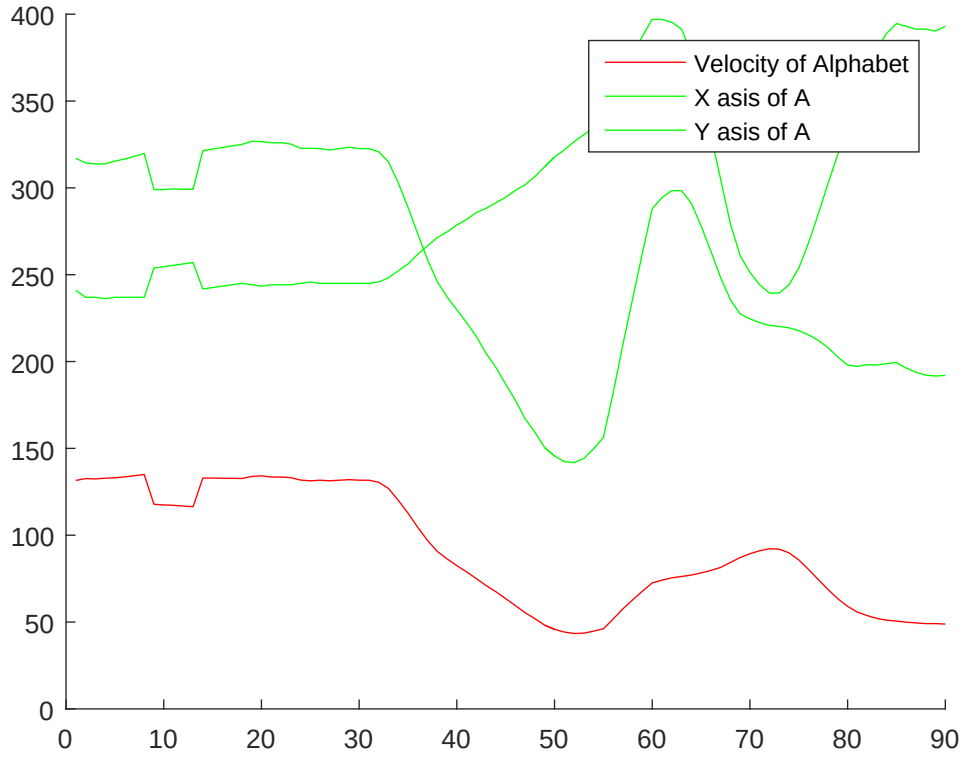


FIGURE 3.10: Velocity point

$$F_T = \{F_{T_1}, F_{T_2}, ..., F_{T_{11}}\} \tag{3.4}$$

$$\sum F_{T_i} = \{m * 1\} \tag{3.5}$$

$$\sum F_{A_i} = \{n * 1\} \tag{3.6}$$

Each of the dtw distance features were normalized [0-1]. The dtw distances had a wide range and introduced many decision points. Normalization quantized the dtw distances. We then apply Support Vector Machine (SVM) classifier with poly kernel.

3.4 Post Classification: Feature Removal

In this part we are calculated rank base on features efficiency i.e gain ratio[34]. In this process we will find less efficient feature to gain maximum accuracy. The first approach we took is to remove one worst feature at a time and recorded the classification accuracy. In this sequential method, we removed up to 60 features.

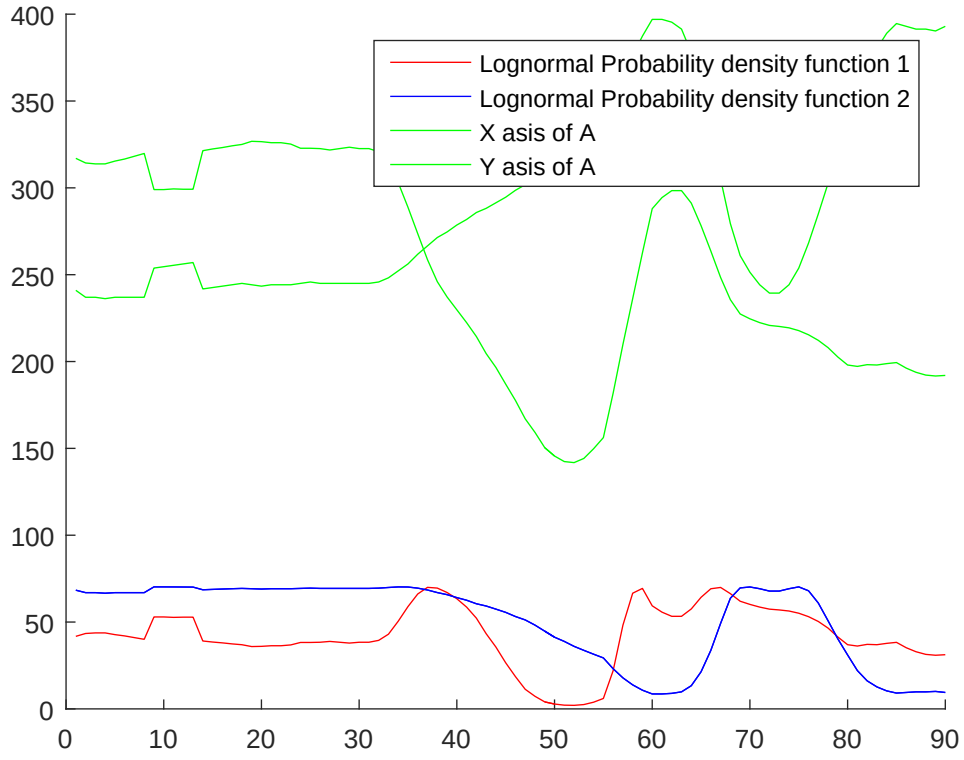


FIGURE 3.11: Log normal Probability density function point

Algorithm 2 Feature Selection

function FEATURE($trajectoryFile$)
  create a empty signal F
  create a empty signal S
  put x values to S
  put y values to S
  put $depth$ values to S
  while still a point in $trajectoryFile$ **do**
    add point wise euclidean distance to S
    add theta value to S
    add Velocity value to S
    add Log normal Probability density value to S with $\sigma = \sum \frac{total\ row\ point}{total\ number\ of\ raw\ point}$
    level freeman code 4 add this value to S
    level freeman code 8 add this value to S
    level freeman code 16 add this value to S
  end while
  normalize [0-1] S
  $F \leftarrow [S]$
  return F
end function

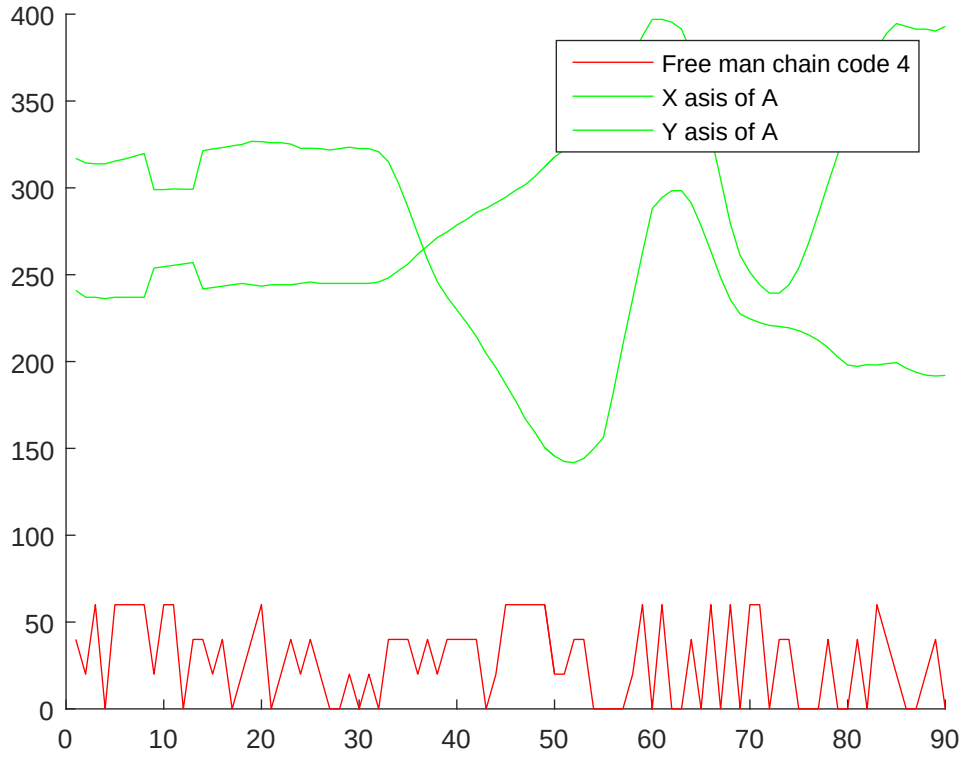


FIGURE 3.12: Free man chain code 4 point

However, the best accuracy came from removing only 8 features. We think that the derived distance features from the comparison of an alphabet with all templates contain useful information for classification. We also used a heuristics based approach [34] to remove some number of features based on chi-square distance. However that technique did not provide comparable accuracy in our dataset.

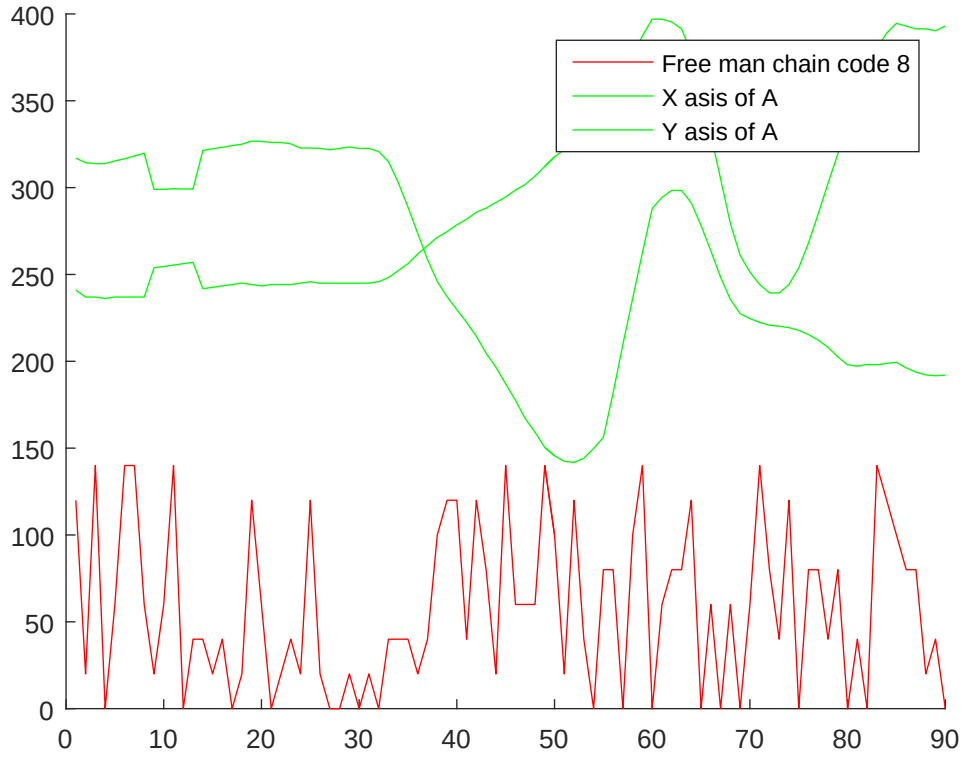


FIGURE 3.13: Free man chain code 8 point

Algorithm 3 proposed algorithm

procedure PROPOSED ALGORITHM()
    Create a empty signal $SVMinput$
    for To all alphabet **do**
        DATASETSIGNALCOLLECTION($variation$) ▷ shown in algorithm 1
        for template A to Z **do**
            FEATURE($variation$) ▷ shown in algorithm 2
            DTW($Feature\,of$, $user\,alphabet$, $Feature\,of\,template\,alphabet$) ▷ shown in algorithm 2
        end for
        $SVMinput \leftarrow [DTW]$ ▷ this use for SVM input
    end for
    Calculate gain ratio based on 15 fold cross verification
    Removing worst feature and apply SVM for recognition with 15 fold cross verification to find maximum accuracy
end procedure

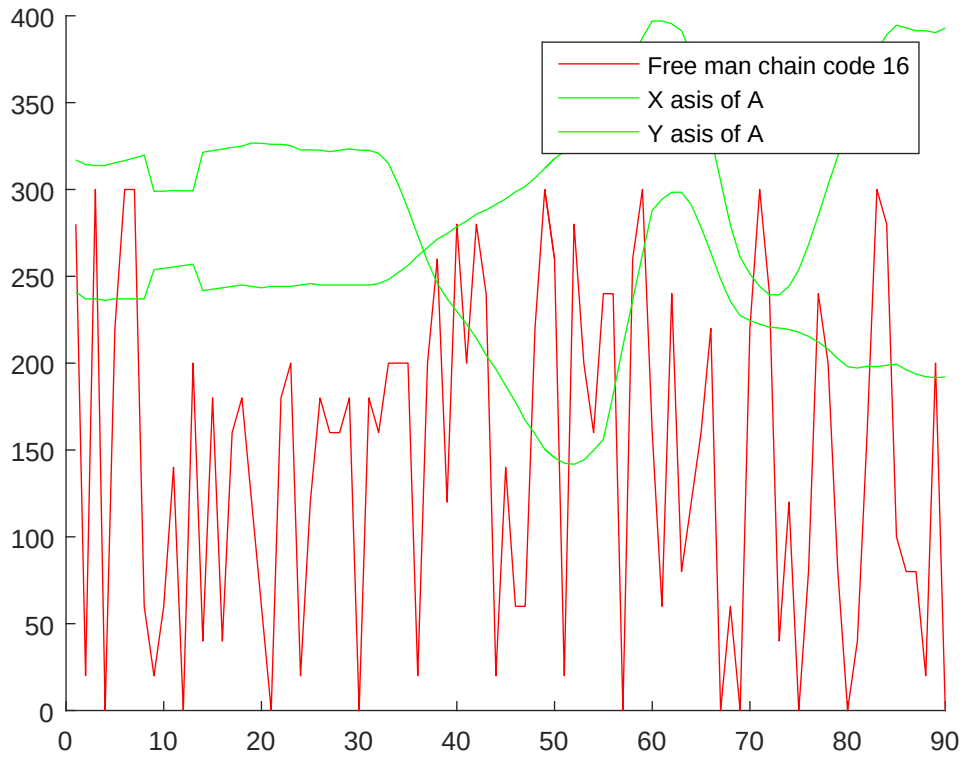


FIGURE 3.14: Free man chain code 16 point

Chapter 4

Experimental Results and Evaluation

We have used DTW distances as feature vector to classify 26 ECAs, its a multi class classification problem for which we have used SVM. DTW needs at least one template for each ECA. Normally any distance should be calculated from an ideal references. Hence reference template has been generated from the best ECAs written by one of the users. Apart from that, 15 other user data were used for training and testing. The templates were used neither in training nor in testing. The DTW distance features had wide value ranges and hence the classification results were influenced by a little change of the distance values. Normalization helps in this regard. We applied normalization between 0 to 1 in each of the 286 feature dimensions.

The classification result has been summarized in confusion matrix in Table 4.1.

The classification accuracy has been calculated by determine the True Positive (TP) rate according to the equation 4.1

$$TPR\,of\ Alphabet = \frac{Correctly\,classified\,instances\,of\,the Alphabet}{Number\,of\,instances\,of\,the\,Alphabet} * 100 \quad (4.1)$$

The result of TP rate and other for the alphabets are shown in Table 4.2. The result is generated considering all the 286 normalized DTW distance features.

4.1 Result Analysis

4.1.1 True positive rate (TPR)

True positive rate determines the actual positive cases out of True positive (TP) and False Negative (FN). It has been calculated using the equation 4.2. We got the the average TPR is 57.2% in our proposed Method.

$$TPR = \frac{TP}{TP + FN} \tag{4.2}$$

4.1.2 False positive rate (FPR)

FPR determines the actual negatives out of total negative cases. It has been calculated using the equation 4.3

$$FPR = \frac{FP}{FP + TN} \tag{4.3}$$

4.1.3 Matthews Correlation Coefficient (MCC)

The Matthews Correlation Coefficient (MCC) has a range of -1 to 1 where -1 indicates a completely wrong binary classifier while 1 indicates a completely correct binary classifier. Using the MCC allows one to predict how well the classification model/function is performing. MCC is calculated using the equation equation 4.4

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{[(TP + FP) * (FN + TN) * (FP + TN) * (TP + FN)]}} \tag{4.4}$$

4.1.4 F-Measure

F-Measure determine the harmonic mean of the precision and sensitivity as shown in equation 4.5

$$F - Measure = \frac{2 * TP}{2 * TP + FP + FN} \tag{4.5}$$

We got the average F-measure value 56.5%.

TABLE 4.1: Confusion Matrix

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	5	1	1	0	1	1	1	0	0	0	2	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0
B	0	11	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
C	1	0	11	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
D	0	3	0	6	1	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	2	0	0
E	0	1	0	0	8	0	1	0	0	0	2	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0
F	1	0	1	0	0	9	2	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	2	0	2	1	7	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0
H	1	1	0	0	1	1	0	7	0	0	1	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0
I	0	0	0	1	1	1	0	0	5	2	0	2	0	1	1	0	0	0	0	0	0	0	0	0	0	1
J	1	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	2	0	0	0	0	0	1	0
K	0	0	0	0	1	0	0	3	0	0	6	0	0	1	0	0	2	1	0	0	0	0	1	0	0	0
L	0	0	1	0	0	0	0	0	1	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	1	0	0	2	3	0	0	0	0	7	0	0	0	0	0	0	0	0	0	1	0	1	0
N	1	0	0	0	0	0	0	0	0	0	1	0	1	8	0	0	0	0	0	0	1	0	3	0	0	0
O	1	1	1	0	0	0	0	0	0	0	0	0	0	0	9	1	0	0	0	1	1	0	0	0	0	0
P	0	2	0	2	0	0	0	0	0	0	0	0	0	0	1	10	0	0	0	0	0	0	0	0	0	0
Q	0	0	1	0	1	2	2	0	0	0	1	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0
R	0	0	1	0	1	1	0	0	0	0	4	0	0	0	0	0	0	6	0	1	0	0	0	0	1	0
S	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	11	0	0	0	0	1	0	0
T	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	13	0	0	0	0	0	0
U	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	8	3	1	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3	11	0	0	0	0
W	1	0	0	0	0	1	0	0	0	0	1	0	0	3	0	0	0	0	0	0	0	1	8	0	0	0
X	0	0	1	2	0	0	2	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	5	2	0
Y	0	0	0	2	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	2	0	0	0	1	7	0
Z	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13

4.1.5 Classification Accuracy

We report the classification accuracies in the individual alphabets in Table 4.3 according to the followings:

- Using best 50 ranked features : Per cross validation iteration step, we have selected best 50 features
- Randomly removed worst features: Per cross validation iteration step, we have removed worst feature and in total iteration on an average 216 feature is selected
- Using all 286 features
- Removing 8 worst ranked features

The accuracy by removing one by one worst ranked feature is shown in Figure 4.4. After removing up to 60 features, we have seen the accuracy does not vary significantly.

TABLE 4.2: Classification Accuracy by removing features

Class	Rank with best 50 Feature based on [31]	Randomly Remove worst feature average 216 [31, 32]	Result on 286 Feature [29]	Result on 276 Feature removing feature based on [34]
A	0.2	0.267	0.333	0.333
B	0.733	0.467	0.667	0.733
C	0.733	0.8	0.733	0.733
D	0.266	0.333	0.333	0.4
E	0.6	0.6	0.467	0.533
F	0.333	0.333	0.6	0.6
G	0.267	0.133	0.467	0.467
H	0.2	0.4	0.467	0.467
I	0.2	0.267	0.333	0.333
J	0.6	0.6	0.733	0.733
K	0.133	0.133	0.4	0.4
L	0.867	0.6	0.867	0.867
M	0.6	0.467	0.467	0.467
N	0.6	0.467	0.533	0.533
O	0.4	0.6	0.6	0.6
P	0.467	0.667	0.6	0.667
Q	0.133	0.133	0.533	0.533
R	0.4	0.333	0.4	0.4
S	0.733	0.6	0.667	0.733
T	0.667	0.667	0.867	0.867
U	0.667	0.667	0.533	0.533
V	0.733	0.667	0.667	0.733
W	0.8	0.667	0.533	0.533
X	0.267	0.267	0.333	0.333
Y	0.733	0.667	0.4	0.467
Z	0.733	0.667	0.867	0.867
AVG	0.5026	0.479	0.554	0.571

We have tested accuracy using single features to validate their suitability shown in Table 4.5.

- Freeman chain code 4
- Freeman chain code 8
- Histogram of Oriented Gradients (HOG)
- Log normal Probability Density Function

The table represent this comparison Result among the features we have used. We have got always better accuracy for our proposed approach that have applied 11 features.

TABLE 4.3: Result Table

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
A	0.333	0.019	0.417	0.333	0.37	0.35	0.872	0.289
B	0.733	0.024	0.55	0.733	0.629	0.618	0.969	0.493
C	0.733	0.024	0.55	0.733	0.629	0.618	0.964	0.537
D	0.4	0.021	0.429	0.4	0.414	0.391	0.927	0.284
E	0.533	0.027	0.444	0.533	0.485	0.468	0.861	0.388
F	0.6	0.027	0.474	0.6	0.529	0.529	0.512	0.867
G	0.467	0.029	0.389	0.467	0.424	0.401	0.942	0.321
H	0.467	0.029	0.467	0.467	0.467	0.454	0.81	0.276
I	0.333	0.021	0.455	0.333	0.385	0.969	0.712	0.257
J	0.733	0.016	0.611	0.733	0.667	0.655	0.978	0.68
K	0.4	0.019	0.3	0.4	0.343	0.316	0.834	0.239
L	0.867	0.037	0.867	0.867	0.867	0.861	0.987	0.79
M	0.467	0.005	0.538	0.467	0.5	0.483	0.957	0.433
N	0.533	0.016	0.533	0.533	0.533	0.515	0.893	0.385
O	0.6	0.019	0.643	0.6	0.621	0.606	0.919	0.488
P	0.667	0.013	0.667	0.667	0.667	0.653	0.958	0.547
Q	0.533	0.013	0.667	0.533	0.593	0.582	0.898	0.373
R	0.4	0.011	0.545	0.4	0.462	0.449	0.836	0.31
S	0.733	0.013	0.688	0.7333	0.71	0.698	0.967	0.614
T	0.867	0.013	0.765	0.867	0.813	0.806	0.969	0.685
U	0.533	0.011	0.571	0.533	0.522	0.535	0.907	0.416
V	0.733	0.016	0.733	0.733	0.733	0.723	0.978	0.667
W	0.533	0.011	0.615	0.533	0.571	0.557	0.941	0.435
X	0.333	0.019	0.417	0.333	0.37	0.35	0.886	0.249
Y	0.467	0.019	0.5	0.467	0.483	0.463	0.928	0.4
Z	0.867	0.0003	0.929	0.867	0.897	0.897	0.893	0.851
Avg.	0.572	0.018	0.568	0.572	0.565	0.575	0.896	0.472

TABLE 4.4: Classification Accuracy by removing features one by one

Feature Removed Number	True Positive Value	Feature Removed Number	True Positive Value	Feature Removed Number	True Positive Value
0	55.3846	22	55.8974	43	55.3846
1	56.1538	23	55.641	44	55.3846
2	55.641	24	55.8974	45	55.3846
3	56.1538	25	55.3846	46	55.3846
4	56.4103	26	55.1282	47	55.3846
5	56.4103	27	55.1282	48	55.3846
6	56.1538	28	54.6154	49	55.3846
7	56.6667	28	55.3845	50	56.1538
8	57.1795	29	54.1056	51	55.8974
9	56.9231	30	54.6154	52	55.8974
10	56.9231	31	54.6154	53	56.6667
11	56.1538	32	55.3845	54	56.1538
12	56.6667	33	55.641	55	56.9231
13	55.8974	34	55.1282	56	56.1538
14	55.641	35	55.1282	57	56.4103
15	55.1282	36	55.1282	58	56.4103
16	54.6154	37	55.8974	59	56.4103
17	54.8718	38	55.641	60	56.1538
18	54.8718	39	56.1538		
19	55.1282	40	55.1282		
20	55.3846	41	55.1282		
21	55.641	42	54.8718		

TABLE 4.5: True positive rate comparisons between different features and our approach

Class	TP Rate for Free man chain code 4	TP Rate for free man code 8	TP Rate for HOG feature	TP Rate for log normal probability density	TP value of Proposed methods
A	0.2	0.333	0.067	0.067	0.333
B	0.2	0.133	0.933	0.2	0.733
C	0.133	0.133	0	0.133	0.733
D	0.067	0.067	0	0.067	0.4
E	0.333	0.467	0	0.267	0.533
F	0.133	0.2	0	0.2	0.6
G	0.133	0.067	0	0.2	0.467
H	0.267	0.133	0	0.133	0.467
I	0.133	0.067	0	0	0.333
J	0.133	0.2	0	0.333	0.733
K	0.2	0.133	0	0.133	0.4
L	0.2	0.2	0	0.333	0.867
M	0.067	0.067	0	0.067	0.467
N	0.067	0	0	0.067	0.533
O	0.067	0	0	0.133	0.6
P	0	0.267	0	0.067	0.667
Q	0.133	0.2	0	0.067	0.533
R	0.133	0.2	0	0	0.467
S	0	0	0	0	0.733
T	0.133	0.133	0	0	0.8
U	0.133	0.067	0	0.067	0.533
V	0.067	0.067	0	0.4	0.733
W	0	0.067	0	0	0.533
X	0.133	0	0	0	0.333
Y	0.133	0.2	0	0.333	0.467
Z	0	0.067	0	0.2	0.867
AVG	0.123	0.133	0.038	0.133	0.571

Chapter 5

Conclusion

In this thesis work, we have dealt with on-air gesture recognition problem. We tried to recognize ECA characters generated through dynamic hand gesture. The hand trajectory vector were used from each of the gesturing image to extract 11 features. we have created a unique dataset in a complex natural environment from 16 users. Each of the ECAs is presented as time-series values. Then, all pair DTW distances were calculated and total 286 distances are used as features for SVM training and testing. We have performed 15-fold cross-validation and found higher accuracy for our selected features. we have verified the accuracy by comparing with other individual feature like Freeman chain code 4, 8, HOG features, Lognormal probability density function and found best result for our selected combination of features.

In future we will continue our work to recognize small letter English alphabets as well as Bangla alphabets.

Appendix A

Appendix

This rank vs feature matrix is calculated via

$$GainR(Class, Attribute) = (H(Class) - H(Class|Attribute))/H(Attribute)$$

this equation where H represent entropy using 15 fold classifier. Helping with this matrix we create 4.4 table that will help to maximize our result accuracy.

TABLE A.1: Rank vs Feature

Avg. Rank	Feat.	Avg. Rank	Feat.	Avg. Rank	Feat.	Avg. Rank	Feat.	Avg. Rank	Feat.	Avg. Rank	Feat.
3.5	101	54.2	200	109.1	28	143.9	73	204.6	273	251.9	133
4.8	100	54.3	259	109.5	7	144.6	48	206	269	253.1	135
4.8	156	54.8	225	109.5	6	145.3	51	206.1	245	254.7	137
5	189	55	232	109.5	110	145.6	47	206.2	274	254.7	136
6.5	222	55.2	243	110	109	145.8	49	207.2	275	255.1	159
6.9	35	56.6	203	110.2	5	146.3	63	208.2	278	256.7	160
10.6	167	58.4	204	110.4	9	147.6	72	209.4	284	261.1	183
12.2	244	59.1	112	110.7	108	150.8	64	210.1	120	261.7	182
12.5	116	59.7	261	110.8	17	154.9	65	210.1	267	262.9	161
13	145	63	226	111.2	84	155.3	71	210.4	148	264	185
13.6	233	63	217	112	39	157.7	66	210.6	282	264.5	181
14.6	79	63.7	195	112.2	3	158.8	67	210.8	279	265.7	192
15.1	115	67.3	236	112.5	4	161.1	69	211.3	268	265.8	186
15.5	228	67.9	248	114.5	10	161.3	70	212.2	184	266.8	187
17.1	90	68.5	1	115.6	83	162.4	118	212.5	197	267.7	180
17.3	111	70.4	96	116.5	16	168.4	286	213.3	246	269.9	190
17.5	13	71.6	94	116.7	29	169.7	119	214.3	196	269.9	191
18.6	221	71.7	46	117.7	11	171.1	224	216.5	251	269.9	163
20.3	178	72.7	89	119.1	15	171.3	223	216.9	249	270.8	179
20.7	134	74.9	237	121.5	12	173.7	220	217	252	274.3	177
20.9	107	80.5	50	121.7	14	173.7	227	218.6	253	276.2	166
21.7	24	81.3	93	122.7	82	175.4	266	219.1	265	276.4	164

Avg. Rank	Feat.	Avg. Rank	Feat.	Avg. Rank	Feat.	Avg. Rank	Feat.	Avg. Rank	Feat.	Avg. Rank	Feat.
24.3	41	85.9	97	128.5	81	175.7	229	219.4	256	276.8	165
24.5	239	86.6	98	128.9	78	176.9	230	221.3	257	277.6	168
25.1	126	87.3	99	129.3	31	177	219	222.7	149	278.1	169
25.7	258	88.9	95	130.1	80	177.7	216	223.3	260	278.7	176
26.9	123	89.1	25	131.2	40	177.9	231	223.6	264	280.8	170
27.7	211	90.6	22	131.9	44	179.5	218	224.8	263	281.6	175
28.6	255	92.7	92	132.6	57	179.9	238	225.3	262	282.6	171
29.9	281	93.4	87	132.7	77	180.1	234	231.3	147	283.6	174
30.2	85	93.5	91	132.9	76	180.6	235	231.5	194	283.9	172
32.5	250	93.5	88	133.3	58	183.7	138	232.7	146	286	143
33.1	122	93.5	26	134.5	75	185.8	215	235.1	150		
33.1	280	93.7	283	134.7	55	185.9	202	235.9	285		
34.5	23	93.7	21	134.9	56	186.9	201	236.3	152		
34.7	151	97.1	20	135.3	60	187.9	205	237.6	153		
34.9	127	98.5	86	136.3	59	190.3	240	237.7	121		
36.5	277	98.6	8	136.5	43	190.5	206	237.7	144		
37.7	173	99.5	102	136.6	38	190.9	214	238.9	142		
38.7	162	99.5	103	137	32	191.2	241	238.9	154		
38.9	140	99.7	113	137.6	117	191.7	207	240.1	155		
40.1	30	99.9	52	139.9	45	192.7	208	241.9	157		
40.7	19	101.6	210	140.2	61	194.5	209	242.5	188		
42.3	129	101.9	74	140.2	34	194.9	213	244.6	158		
44.1	105	102.8	114	140.6	37	195.6	212	244.8	141		
46	276	103	27	141.1	33	197.6	242	246.1	130		
48.9	254	104.2	128	141.8	53	199.1	198	246.1	125		
49.1	104	107.6	18	141.8	62	200.8	271	248.9	131		
51.2	2	108.3	106	142.5	54	201.2	272	250.4	139		
52.9	68	108.7	199	143.3	36	201.7	124	250.4	193		
53	247	108.7	199	143.7	42	202.8	270	250.8	132		

Bibliography

[1] C. Amma, M. Georgi, and T. Schultz, "Airwriting: A wearable handwriting recognition system," *Personal Ubiquitous Comput.*, vol. 18, no. 1, pp. 191–203, Jan. 2014.

[2] A. Wexelblat, "An approach to natural gesture in virtual environments," *ACM Transactions on Computer-Human Interaction*, vol. 2, no. 3, pp. 179–200, sep 1995.

[3] A. Priya, S. Mishra, S. Raj, S. Mandal, and S. Datta, "Online and offline character recognition: A survey," in *2016 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, apr 2016.

[4] S. Agrawal, I. Constandache, S. Gaonkar, R. Roy Choudhury, K. Caves, and F. DeRuyter, "Using mobile phones to write in air," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '11. New York, NY, USA: ACM, 2011, pp. 15–28.

[5] M. K. H. H. A. R. Robiul Islam, Hasan Mahmud, "Alphabet recognition in air writing using depth information," *The Ninth International Conference on Advances in Computer-Human Interactions*, pp. 299–301, April 2016.

[6] A. Jalalian and S. K. Chalup, "Gdtw-p-svms: Variable-length time series analysis using support vector machines," *Neurocomputing*, vol. 99, pp. 270 – 282, 2013.

[7] N. S. Sreekanth and N. K. Narayanan, "Dynamic gesture recognition—a machine vision based approach," pp. 105–115, 2017.

[8] I. E. Sutherland, "Sketchpad," in *Proceedings of the May 21-23, 1963, spring joint computer conference on - AFIPS '63 (Spring)*. ACM Press, 1963.

[9] W. Buxton, "Lexical and pragmatic considerations of input structures," *ACM SIGGRAPH Computer Graphics*, vol. 17, no. 1, pp. 31–37, jan 1983.

[10] P. R. Cohen, M. Johnston, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clow, "QuickSet," in *Proceedings of the fifth conference on Applied natural language processing -*. Association for Computational Linguistics, 1997.

[11] D. J. Sturman, D. Zeltzer, and S. Pieper, "Hands-on interaction with virtual environments," in *Proceedings of the 2nd annual ACM SIGGRAPH symposium on User interface software and technology - UIST '89*. ACM Press, 1989.

[12] F. K. H. Quek, "Toward a vision-based hand gesture interface," in *Proceedings of the Conference on Virtual Reality Software and Technology*, ser. VRST '94. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 1994, pp. 17–31.

[13] R. A. Bolt, ""put-that-there"," *ACM SIGGRAPH Computer Graphics*, vol. 14, no. 3, pp. 262–270, jul 1980.

[14] W. T. Freeman and C. D. Weissman, "Television control by hand gestures," in *International Workshop on Automatic Face and Gesture Recognition*, 1995, pp. 179–183.

[15] euronews Knowledge. Future of texting: writing in the air! Youtube. [Online]. Available: https://www.youtube.com/watch?v=XMU4zh083l4

[16] D. H. Kim, H. I. Choi, and J. H. Kim, "3d space handwriting recognition with ligature model," in *Proceedings of the Third International Conference on Ubiquitous Computing Systems*, ser. UCS'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 41–56.

[17] S. Mori, C. Suen, and K. Yamamoto, "Historical review of OCR research and development," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1029–1058, jul 1992.

[18] I. Adeyanju, O. Ojo, and E. Omidiora, "Recognition of typewritten characters using hidden markov models," vol. 12, pp. 1–9, 01 2016.

[19] N. Greco, D. Impedovo, M. Lucchese, A. Salzo, and L. Sarcinella, "Bank-check processing system: modifications due to the new european currency," pp. 343– 348 vol.1, 09 2003.

[20] F. L. Siena, B. Byrom, P. Watts, and P. Breedon, "Utilising the intel realsense camera for measuring health outcomes in clinical research," *Journal of medical systems*, vol. 42, no. 3, p. 53, 2018.

[21] S. Totilo. Natal recognizes 31 body parts, uses tenth of xbox 360 "computing resources". [Online]. Available: https://kotaku.com/5442775/natal-recognizes-31-body-parts-uses-tenth-of-xbox-360-computing-resources

[22] Kinect - wikipedia. Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Kinect

[23] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, ser. EMNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 79–86.

[24] T. Evgeniou, M. Pontil, and T. Poggio, "Statistical learning theory: A primer," vol. 38, pp. 9–13, 06 2000.

[25] O. Bousquet, S. Boucheron, and G. Lugosi, "Introduction to statistical learning theory," in *Advanced Lectures on Machine Learning*. Springer Berlin Heidelberg, 2004, pp. 169–207.

[26] R. Mullin, "Time warps, string edits, and macromolecules: The theory and practice of sequence comparison. edited by d. sankoff and j. b. kruskal. addison-wesley publishing company, inc., advanced book program, reading, mass., don mills, ontario, 1983. 300 pp. u. s. $31.95. ISBN 0-201-07809-0," *Canadian Journal of Statistics*, vol. 13, no. 2, pp. 167–168, jun 1985.

[27] C. Qu, "Online kinect handwritten digit recognition based on dynamic time warping and support vector machine," vol. 12, pp. 413–422, 01 2015.

[28] D. M. Viswanathan and S. M. Idicula, "Recognition of hand gestures of english alphabets using HOG method," in *2014 International Conference on Data Science & Engineering (ICDSE)*. IEEE, aug 2014.

[29] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[30] N. R. Draper and H. Smith, "Applied regression analysis, john wiley & sons inc," *Applied regression analysis. 2nd ed. John Wiley and Sons, NY.*, 1981.

[31] M. Bennasar, Y. Hicks, and R. Setchi, "Feature selection using joint mutual information maximisation," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8520 – 8532, 2015.

[32] H. Liu and R. Setiono, "Chi2: feature selection and discretization of numeric attributes," in *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, Nov 1995, pp. 388–391.

[33] J. F. Kenney and E. S. Keeping, "Moving averages," pp. 221–223, 1962.

[34] A. G. Karegowda, A. Manjunath, and M. Jayaram, "Comparative study of attribute selection using gain ratio and correlation based feature selection," *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 271–277, 2010.