

**CONVERTING AMERICAN SIGN LANGUAGE TO VOICE USING  
RBFNN**

---

A Thesis  
Presented to the  
Faculty of  
San Diego State University

---

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science  
in  
Computer Science

---

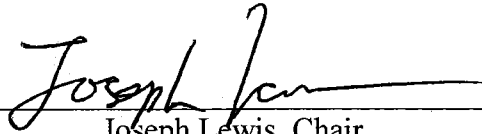
by  
Anirudh Garg  
Summer 2012

**SAN DIEGO STATE UNIVERSITY**

The Undersigned Faculty Committee Approves the

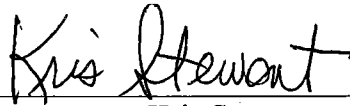
Thesis of Anirudh Garg:

Converting American Sign Language to Voice Using RBFNN



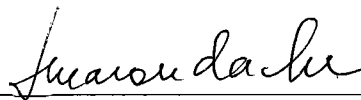
---

Joseph Lewis, Chair  
Department of Computer Science



---

Kris Stewart  
Department of Computer Science



---

Roxana Smarandache  
Department of Mathematics and Statistics

5/23/2012

---

Approval Date

Copyright © 2012  
by  
Anirudh Garg  
All Rights Reserved

## **DEDICATION**

I dedicate this thesis work to my dear parents, for their constant inspiration, priceless sacrifices and the numerous times they have empowered to achieve what I have done in my life.

## **ABSTRACT OF THE THESIS**

Converting American Sign Language to Voice Using RBFNN

by

Anirudh Garg

Master of Science in Computer Science

San Diego State University, 2012

Communication is the most important part of life. Around 1% of the total population of the world is suffering from hearing impairment, and their life is not as easy as it is for human without limitations. In this thesis we propose a model that recognize ASL and convert signs to voice using radial basic function neural network. This model will surely be implemented in real life to make the life of deaf people easier. In this thesis, we are training radial based function for the recognition of ASL. This model starts with image pre-processing of skin detection using grayworld illumination, color space conversion from RGB to YCbCr, and skin detection via threshold. The detected skin regions are represented with centroids and tracked using Euclidean distance measurement. To transform essential data into a more intelligent form, dimension reduction and feature extraction algorithm of principal component analysis (PCA) and linear discriminant analysis (LDA) are used. Finally, a radial basis function neural network (RBFNN) is used for classification to recognize different hand gestures.

## TABLE OF CONTENTS

	PAGE
ABSTRACT.....	v
LIST OF TABLES.....	viii
LIST OF FIGURES .....	ix
ACKNOWLEDGEMENTS.....	x
CHAPTER	
INTRODUCTION .....	1
1.1 Characteristics of Sign Language .....	1
1.2 Related Difficulties and Recognition Approach.....	2
1.3 Framework for Dynamic Gesture Recognition of ASL.....	3
1.3.1 Video Procurement .....	4
1.3.2 Image Processing .....	4
1.3.3 Feature Extraction.....	5
1.3.4 Gestures Allocation.....	5
1.4 Outline.....	5
BACKGROUND .....	6
2.1 Hand Tracking/ Hand Gestures.....	6
2.2 What is ASL (American Sign Language)? .....	8
2.3 What is HMM? .....	10
2.3.1 Architecture of HMM .....	17
2.3.2 Previous Use of HMM's for Recognizing Sign Language .....	17
2.4 What is Artificial Neural Networks? .....	19
2.4.1 Supervised Learning .....	22
2.4.2 Unsupervised Learning.....	23
2.5 What is Radial Basis Function Neural Network? .....	23
2.6 Limitations in Neural Computing.....	24
2.7 Advantage of Neural Network over HMM.....	25
APPROACH .....	27

3.1 Proposed American Sign Language Hand Gesture to Voice Architecture.....	27
3.2 ASL Image Feature Extraction .....	27
3.2.1 Skin Detection.....	27
3.2.2 Blob Detection .....	30
3.2.3 Principal Component Analysis .....	30
3.2.4 Linear Discriminant Analysis .....	31
3.3 Radial Basis Function Neural Network .....	32
3.4 Gesture to Voice .....	35
3.5 What is MATLAB?.....	36
3.6 Image Database.....	37
RESULTS .....	38
4.1 Skin Detection.....	38
4.2 PCA Feature Extraction .....	39
4.3 LDA Feature Extraction.....	41
4.4 RBFNN Recognition.....	42
CONCLUSIONS AND FUTURE ENHANCEMENTS.....	50
5.1 Conclusions.....	50
5.2 Future Enhancements.....	51
REFERENCES .....	52
APPENDIX	
GLOSSARY .....	55

**LIST OF TABLES**

	PAGE
Table 4.1. Alphabets with Classes .....	41



## LIST OF FIGURES

	PAGE
Figure 2.1. ASL manual alphabets.....	9
Figure 2.2. American Sign Language family representation. ....	11
Figure 2.3. American Sign Language places representation.....	12
Figure 2.4. American Sign Language feelings representation.....	13
Figure 2.5. American Sign Language gesture representation. ....	14
Figure 2.6. American Sign Language gesture representation. ....	15
Figure 2.7. Probabilistic parameters of a hidden Markov model.....	16
Figure 2.8. An example artificial neural network with a hidden layer. ....	20
Figure 2.9. ANN Dependency graph. ....	21
Figure 2.10. Neural Net Block diagram.....	22
Figure 3.1. Proposed American Sign Language hand gesture to voice architecture. ....	28
Figure 3.2. Block diagram of a RBF network.....	33
Figure 3.3. Network architecture of the RBF. ....	35
Figure 4.1. Simulated outcome of skin detection. ....	39
Figure 4.2. Simulated outcome of skin detection. ....	40
Figure 4.3. PCA plot for the first two Principal Component.....	41
Figure 4.4. PCA plot for the second vs third Principal Component. ....	42
Figure 4.5. First two feature plot for LDA.....	43
Figure 4.6. Second vs. third feature plot for LDA. ....	43
Figure 4.7. 64x64 in dimension hand images for alphabet “A.”.....	44
Figure 4.8. 64x64 in dimension hand images for alphabet “B.”.....	45
Figure 4.9. The plot of RBFNN training with error rate less than 0.001%. ....	46
Figure 4.10: Montage of frames recognizing alphabets. (A) Recognizing alphabet “A”. (B) Recognizing alphabet “B”. (C) Recognizing alphabet “C”. (D) Recognizing alphabet “D”. (E) Recognizing alphabet “E”. ....	47

## **ACKNOWLEDGEMENTS**

I take this opportunity to sincerely thank my thesis advisor, Dr. Joseph Lewis, for the supervision, inescapable support, and assisting me during every step of my research work. Without his help it would not have been fortuitous for me to achieve this accomplishment. I would also like to thank Professor Kris Stewart and Professor Roxana for being the readers of this thesis and for giving their beneficial suggestions.

## **CHAPTER 1**

### **INTRODUCTION**

In the 21<sup>st</sup> century field of science and technology has reached such a level that people are expecting more comfortable and useful things, which can make their lives easier. Now days, homes with voice recognition built in with the sense of gestures have already been conceived. There are video games are in the market, which can be played with real time gestures, and all this has been possible with the advent of the new technology. Even our mobiles are loaded with all similar technologies. Nevertheless, there are people who are less fortunate than us and are physically challenged, may it be deafness or being aphonic. Such people lag behind their non-handicapped peers in using these technologies. These people have some expectations from the researchers and mostly from a computer scientist that we, computer scientists can provide some machine/model which help them to communicate and express their feelings with others. Very few researchers' have them in mind and provide their continuous works for such people.

One might expect digital technologies will play a huge role in human's daily routines and whole world will be interacting via machines either with the means of gestures or speech recognition within a few decades. If we are in a position to predict such a future, we ought to think about the physically challenged and do something for them. Sign language is the natural language of the deaf and aphonic people. It is the basic method for the communication of deaf person. American Sign Language (ASL) is the language chosen by almost all the deaf communities of United States of America. Different Sign languages are evolved depending on the regions such as GSL (German Sign Language), CSL (Chinese Sign Language), Auslan (Australian Sign Language), ArSL (Arabic Sign Language), and many more [1].

#### **1.1 CHARACTERISTICS OF SIGN LANGUAGE**

Characterization of sign language is between two parameter one being manual and other non-manual. The manual parameter consists of motion, location, hand shape, and hand

orientation. The non-manual parameter includes facial expression, mouth movements, and motion of the head [2]. Sign language does not include the environment which kinesics does. Few terms are use in the sign language like signing space, which refers to signing taking place in 3D space and close to truck and head. Signs are either one-handed or two-handed. When only the dominant hand is in use to perform the signs they are denoted as one-hand signs else when the non-dominant hand also comes in the phase it is termed as two-handed signs [3].

Sign language when evolved is different from spoken language so the grammar of the sign language is primarily different from spoken language. In spoken language, the structure of the sentence is one-dimensional; one word followed by another, while in sign language, a simultaneous structure exists with a parallel temporal and spatial configuration. As based on these characteristics, the syntax of sign language sentence is not as strict as in spoken language. Formation of a sign language sentence includes or refers to: time, location, person, base. In spoken languages, a letter represents a sound. For deaf nothing comparable exists. Hence the people, who are deaf by birth or became deaf early in their lives, have very limited vocabulary of spoken language and faces great difficulties in reading and writing.

## **1.2 RELATED DIFFICULTIES AND RECOGNITION APPROACH**

Many basic problems can occur during recognition of continuous sign language; few of them are listed below [4]:

- While signing, some fingers or even a whole hand can be occluded.
- Boundaries of a sign have to be detected automatically.
- The position of the signer in front of camera may vary.
- A sign is affected by the antecede and consecutive sign.
- Signer's movements, like rotating around the body axis or shifting in one direction, must be considered.
- Each sign is different with respect to time and space. Speed also matters while signing. If the same signer performs the same, sign twice there must be a little difference with respect to speed and position of hand.
- The projection of the 3D scene on a 2D plane results in the loose of depth information. The reconstruction of the 3D trajectory of the hand in space is not always possible.

- The processing of a large amount of image data is time consuming, so real-time recognition is difficult.

Most of these above described problems can be solved using radial basic function neural network (RBFNN), which is based on the neural network. As RBFNN will be able to solve most of these problems, so RBFNN seems to be the best approach for the recognition of ASL. The main purpose of this thesis research is to develop a unique protocol model, which can be further implemented in the future for the help of the deaf person, with the technique of continuous real-time recognition of ASL and convert ASL to the voice of the signer. However, there are few more problems, which can occur during the recognition of continuous sign language. First, same sign performed by the same signer twice might vary in the position and in the speed. Second, sometimes it is hard to recognize the boundaries of the gesture performed as it might overlap. Last but not the least is the co-articulation problem. Co-articulation refers to the extra movements between two consecutive gestures [5][6].

Keeping all those in mind, our research objective here is to present a working model, which overcomes most of these above described problems and converts ASL to voice. This model is the gesture recognition system-using image processing [6] along with the recognition of signs using RBFNN to provide the actual output, which is the voice, related to gestures. Both parts of our research model is very crucial as all the work is proposed to be done in real time and also the percentage error should be as low as possible.

### **1.3 FRAMEWORK FOR DYNAMIC GESTURE RECOGNITION OF ASL**

The two pillars of our model are image processing and recognition of signs using RBFNN. So the framework for the first pillar is recognition of gesture using image pre-processing algorithm. Some work is already done in recognition of sign language using the same technique, but in the earlier researches researchers used either expensive wired gloves [7] or other kind of markers which the signer have to wear while performing gesture. In one of the past year's research, researcher replace wired gloves with the cotton gloves [8] but then again the signer has to wear something. We are proposing the model in which the signers do not have to wear anything at all in their hands. Recognition of ASL from the video stream consists of four main parts.

1. Video procurement
2. Image processing
3. Feature extraction
4. Gesture allocation

### **1.3.1 Video Procurement**

Few of the past researches used camera as an input method for their model. One of them used the camera in the baseball cap [8] that captures the input from the hand only. But experienced signers in ASL sketch a person, place, or thing using gesture of hand and then point to a place in space to store that object temporarily for later reference [9]. For the purpose of this model, this aspect of ASL will be ignored. Furthermore, in ASL facial expressions are used as eyebrows are raised for a question, relaxed for a statement, and folded for a command. To keep this aspect in mind, this model will capture the facial expression using four algorithms for face and hand detection. There has been a system already build to track facial expression [10]. For the development of this model, we are using a database of videos, which are ASL signs that are prepared to use in research works. These videos have some constraints like same background, speed of the signer to perform sign, etc. However, this database is outstanding and perfect fit for our model. The videos are isolated into frames with 5fps [6]. This 5fps rate is suitable rate for the gradual and brisk motion. If we choose any higher rate than it might produce an extra work of figuring signs and generates unnecessary frames. In addition, if we choose lower FPS, gestures might get lost due to the speed by which the gestures are being formed in space. Therefore, choosing 5fps provides good results.

### **1.3.2 Image Processing**

This process in our model is one of the main components on which the further functionality and error percentage depends. As in this process continuous signs, which are stream of 5FPS, are being digitalized and processed through image processing and only the face and hands region are taken out thus removing the background from the frame that may create any extra redundancy. We are trying to complete this process through the help of matlab in which we can detect facial expression and hand movement and convert them as a stream of feature vector or we can say our next stage input. Through image processing, we

have to calculate few properties like shape and position of fingers, hands, and body of the signer. From this information a vector is built that reflect the manual sign parameters of sign language. This process has been covered in more detail in chapter 3 of this book.

### **1.3.3 Feature Extraction**

The output of feature extraction is used as the input for the RBFNN. Feature extraction is the method of changing the input data to the specific features; like in our case we are extracting the gestures of sign language produced by the hands and facial expressions. Feature extraction involves simplifying the amount of resources required to describe a large set of data accurately. Detailed discussion about the way this feature is used to extract the input for RBFNN will be in Chapter3.

### **1.3.4 Gestures Allocation**

This is the final step of the model. This stage can be obtained when the feature vectors can be provided as input and RBFNN model, which is already, trained with most of the ASL (American Sign Language) gestures. Radial Basis Function Neural Network (RBFNN) is used for classification to recognize difference hand gestures and provides output in text form. A text-to-speech (TTS) system converts normal language text into speech. Again, more about this process will be covered in Chapter 3 of this book.

## **1.4 OUTLINE**

Chapter 2 discusses the tracking systems, previous work on sign language, HMM technology, ANN technology, and difference between the two, benefits of applying ANN over HMM to recognition of ASL. Chapter 3 will cover the details of machine vision algorithm, the RBFNN training and recognition methods, proposed architecture of recognition of ASL and converting signs to voice. Chapter 4 describes the experiments performed and lists the results. Summary and discussion of future work is included in Chapter 5 and ending the work with the references and glossary (see Appendix).

## **CHAPTER 2**

### **BACKGROUND**

Recognition of sign language requires two main components: hand tracking and pattern recognition. Machine vision and virtual environment research have provided several tools for hand tracking and continuous speech recognition provides an excellent base for the pattern recognition. In this chapter of the thesis, I throw some light on the tracking systems, previous work on sign language, HMM technology, ANN technology, difference between the two and benefits of applying ANN over HMM to recognition of ASL are discussed.

#### **2.1 HAND TRACKING/ HAND GESTURES**

Term “multimedia” in computers in itself opens wide variety of things, which can be computed. In addition, the multimedia computers are packed with video cameras, and evolve wide scope for the common gestures in everyday use. Wide varieties of interfaces have been proposed, using video driven gestures for mouse control, full body interactions [11], expression tracking, electronic presentation and many more.

As the expressiveness of hand, it has been a point of focus for most of the gesture recognition systems. It is difficult to track the natural hand in real time using camera, but most of the researchers were able to demonstrate their successful systems in controlled settings. A hand tracking navigating 3D worlds has been shown by Freeman [12]. The hand in a small area on a table has been tracked by greyscale camera and it uses hand and finger position to control the direction of a virtual graphics camera.

Researchers Rehg and Kanade [13] have shown a model that consumes two-camera system that can recover the 27 degrees (freedom in hand) of motion in hand. They were able to successfully demonstrate the tracking but with the limited motions, and simple background was required which blocks the observation of natural hand gestures.

Hand gesture research can be classified in three categorized. Glove based analysis is the first, vision based analysis will be second and the third will be analysis of drawing gestures.



The glove based analysis employs sensors either mechanical or optical attached to a glove that converts fingers flexion's into electrical signals for determining the hand posture. An additional sensor verifies the corresponding position of the hand. This sensor is generally a magnetic or acoustic sensor attached to gloves. Tool kits like look-up table software are provided along with gloves for few dataglove applications for, the recognition of hand posture.

Vision based analysis, which falls in the second category of hand gesture research, is based on the way human beings distinguish information about their neighborhood. The implementation of this category is the most difficult in an adequate way. Many different procedures have been tested so far. A three-dimensional model for the human hand is being proposed. The model is to match to images of the hand by one or more cameras and specifications identical to palm orientation and joint angles are determined. These specifications are then used to perform gesture classification. Lee and Kunii [14] developed three-dimensional hand skeleton model that is a hand gesture analysis system with 27 degrees of freedom for hand movement. They joined five major restrictions based on human hand kinematics to reduce the model parameter space search. Marked gloves were used to make the model simple.

The third category, analysis of drawing gestures, usually includes the use of stylus as an input device. Investigation of drawing gestures can also suggest to recognition of written text.

The broad majority of hand gesture recognition work has used mechanical sensing, mostly for direct manipulation of a virtual environment and at times for symbolic communication. To sense the hand posture mechanically there are wide range of problems, anyhow, including reliability, accuracy, and electromagnetic noise.

Complete ASL recognition system includes datagloves. Takashi and Kishino [15] discuss a Dataglove-based system that could recognize 34 of the 46 Japanese gestures using hand orientation and joint angle technique. From their research, it seems that these systems are technically interesting but they suffer from a lack of training as the test user signs each of 46 gestures 10 times to provide data for the main component and group analysis. The user also performed another test, which contains five iterations of the alphabet, with each gesture well separated in time.

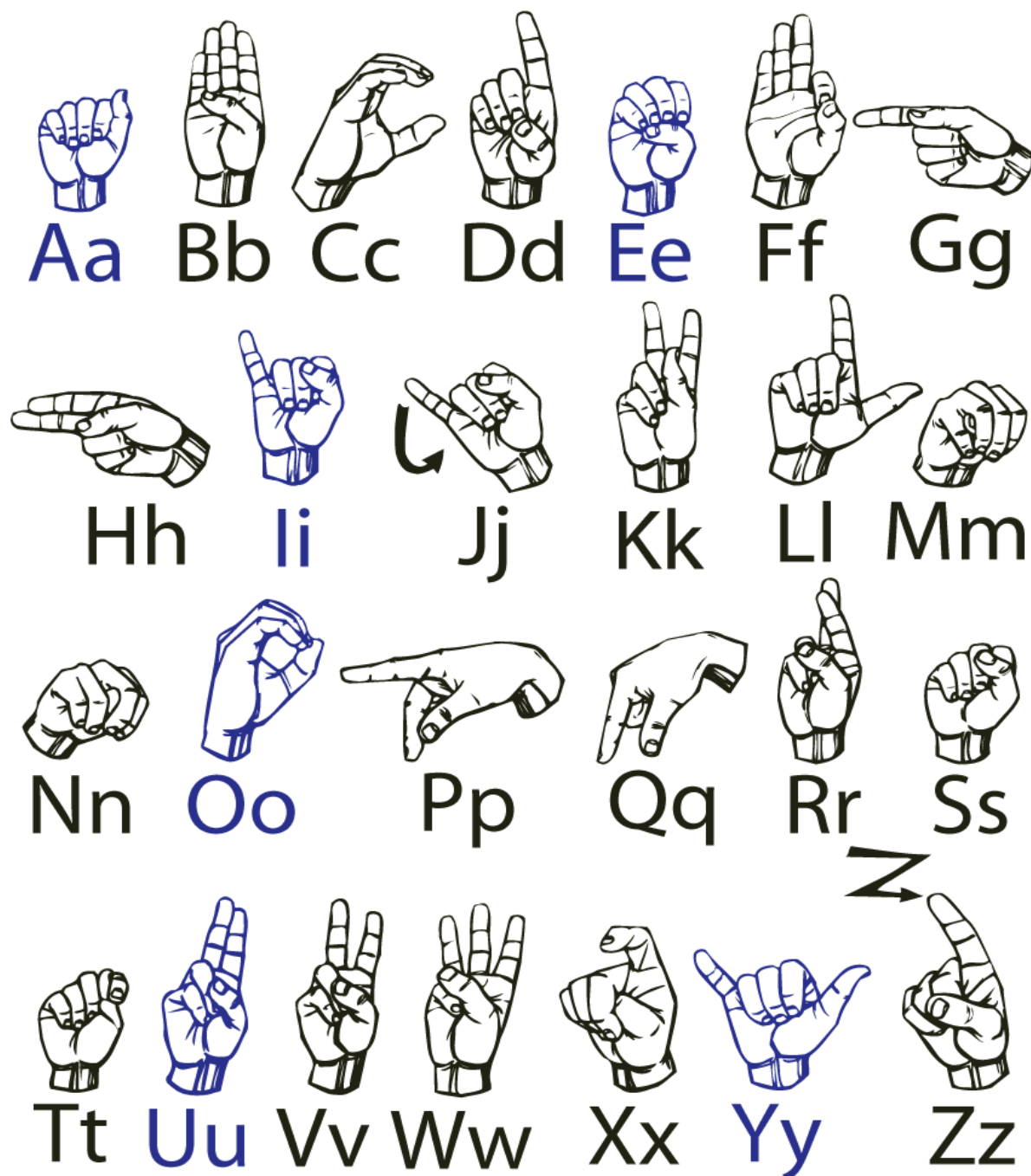
Parish, Sperling, and Landy [16] perform most efficient work in support of machine sign language recognition, they have performed accurate studies on the bandwidth required for sign conversation using temporally sub-sampled images. Most systems to date study isolate/static gestures. In most of the cases, those are fingerspelling signs.

## **2.2 WHAT IS ASL (AMERICAN SIGN LANGUAGE)?**

ASL is the fourth most commonly used language in the USA. American Sign Language is the language, which is extensively used by deaf people, and this language is officially acquired by the deaf society of United States. ASL is a complete, complex language that employs signs made by moving the hands combined with facial expressions and postures of the body. ASL is not defined as the world language but it has its roots in English speaking parts of Canada, few regions of Mexico, and all over United States of America. A signer of ASL would have trouble in understanding any other Sign language of any region or country, as they are very different according to grammar and signs. Even ASL has its own grammar. ASL did not take the English grammar. Grammar of ASL provides more elasticity in arranging words. ASL consists of almost 6000 gestures of common words with finger spelling used to communicate proper nouns. Finger enchanting uses single hand and 26 gestures to communicate the 26 letters of the alphabets (see Figure 2.1 [17]) whereas the user of sign language prefers complete word signs almost everywhere this provides them accession and overrides the pace of conversational English. [7] [18].

Sign language is not universal--each community had developed their own sign language. For example In Britain and in America people speak English but both ASL and British Sign language are very different and both language signers have the difficulty in understanding the signs. But, it is interesting to note that ASL shares lots of vocabulary terms with Old French Sign Language (LSF) it is because the first teacher of the deaf in United States was Laurent Clerc [17], in nineteenth century. Most sign language develops independently and each country (and in some cases, each city) has its own sign language.

Most people define ASL and other sign language as “Gestural” language but this is not completely true as hand gesture is just one component of ASL. Sign language communicates most of their prosody through non-manual signs. Signs includes facial features



**Figure 2.1. ASL manual alphabets.** Source: Wikipedia. File:Aslfingerspellalpha, 2011. <http://en.wikipedia.org/wiki/File:Aslfingerspellalpha.png>, accessed Apr. 2012.

such as eyebrow, eyes, cheeks motion and lip-mouth movements as well as other factors such as body orientation are also significant in ASL. As they all fall under the important part of grammatical system and are used in different combinations to describe several categories of information including lexical distinction, adjectival or adverbial content, and discourse

functions [17]. In ASL, some signs have needed facial components that distinguish them from other signs. This type of lexical distinction can be described as the sign translated ‘not yet’, which requires that the tongue touch the lower lip and that the head rotate from side to side, in addition to the manual part of the sign. Without these features, it would be interpreted as ‘late’.

Grammatical structure that is shown through non-manual signs includes question, negation, relative clauses, boundaries between sentences, and the argument structure of some verbs. ASL and BSL (British Sign language) use similar non-manual marking for yes/no questions. Some adjective and adverbial information is communicated through non-manual signs, but what these signs are varies from language to language. For instance, in ASL, a slightly open mouth with the tongue relaxed and visible in the corner of the mouth means ‘carelessly,’ but a similar sign in BSL means ‘boring’ or ‘unpleasant’ [17].

Discourse functions such as turn taking are largely regulated through head movement and eye gaze. Since the addressee in a signed conversation must be watching the signer, a signer can avoid letting the other person have a turn by not looking at them, or can avoid letting the other person have a turn by not looking at them, or can indicate that the other person may have a turn by making eye contact. [17].

Professionals and experienced signers of American Sign Language use the surrounding space to describe a person, place, or thing. He /She points to a place in space to temporarily load the object for the later reference. However, not all these aspects are under consideration in this thesis.

The goal of this thesis to present the model specifically focuses on working on the real-time processing gestures of ASL. This model supports an implementation that will convert ASL gesture or signs to voice using artificial neural network “radial basic function”, helping in recognition of sign language.

Figure 2.2 to 2.6 [19] illustrates common signs used in daily life of a deaf person.

## **2.3 WHAT IS HMM?**

A hidden Markov model is a doubly stochastic process in which an underlying stochastic process that is not observable (i.e., it is hidden) can only be observed through

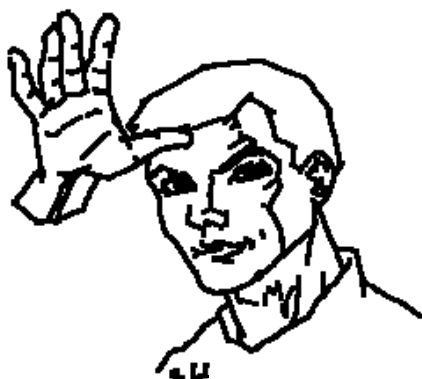
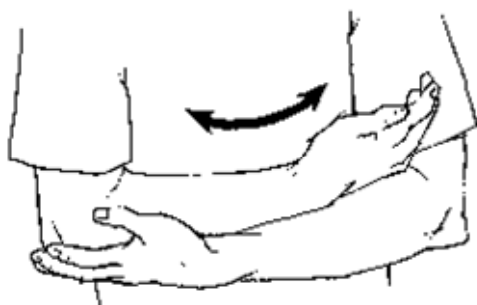
**DAD****MOM****GRANDPA****AUNT****BABY****MARRIAGE**

Figure 2.2. American Sign Language family representation. Source: B. Vicars. Basic Signs Pictures, n.d. <http://www.lifeprint.com/asl101/pages-layout/concepts.htm>, accessed Dec. 2011.

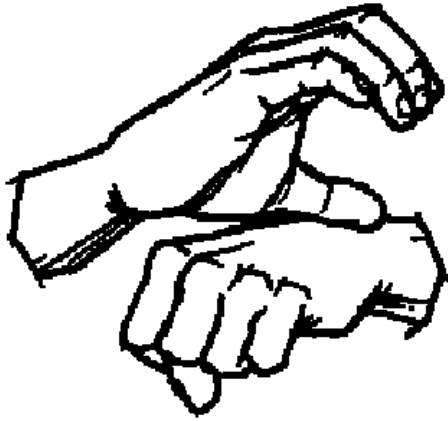
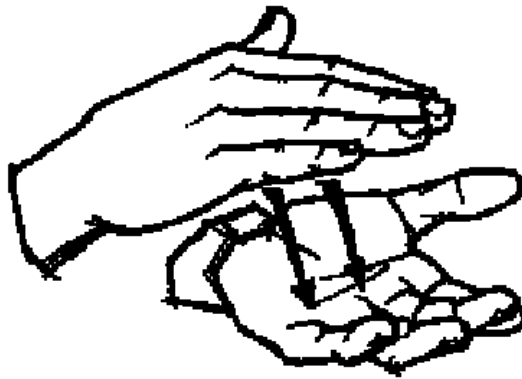
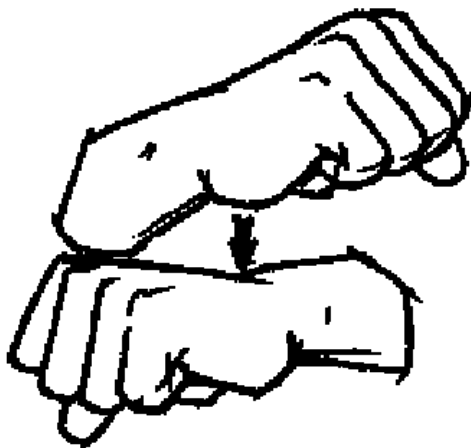
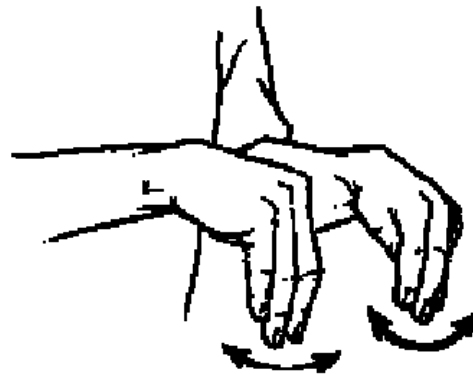
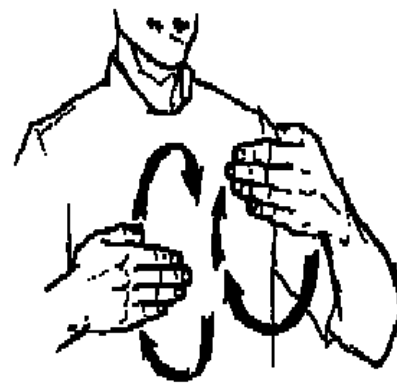
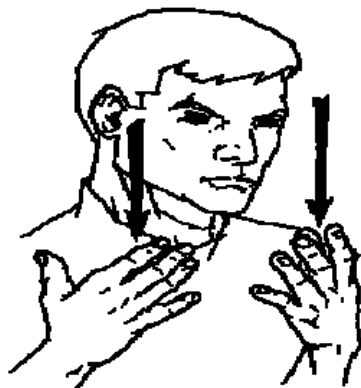
**CHURCH****SCHOOL****WORK****STORE****CAR DRIVING**

Figure 2.3. American Sign Language places representation. Source: B. Vicars. Basic Signs Pictures, n.d. <http://www.lifeprint.com/asl101/pages-layout/concepts.htm>, accessed Dec. 2011.

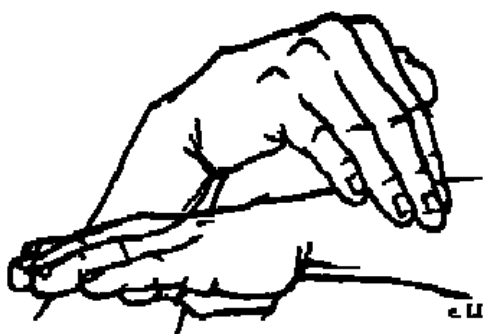
**ANGRY****CRY****EXCUSE****HAPPY****SAD****SORRY**

**Figure 2.4. American Sign Language feelings representation. Source: B. Vicars. Basic Signs Pictures, n.d. <http://www.lifeprint.com/asl101/pages-layout/concepts.htm>, accessed Dec. 2011.**

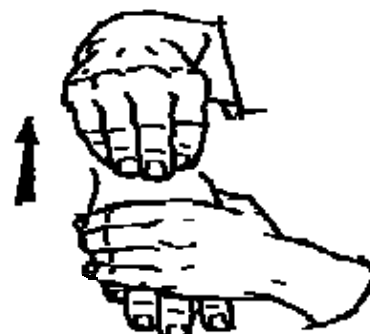
**BATHROOM****CAT****FUTURE****IN****MORE****LOVE**

Figure 2.5. American Sign Language gesture representation. Source: B. Vicars. Basic Signs Pictures, n.d. <http://www.lifeprint.com/asl101/pages-layout/concepts.htm>, accessed Dec. 2011.





NIGHT



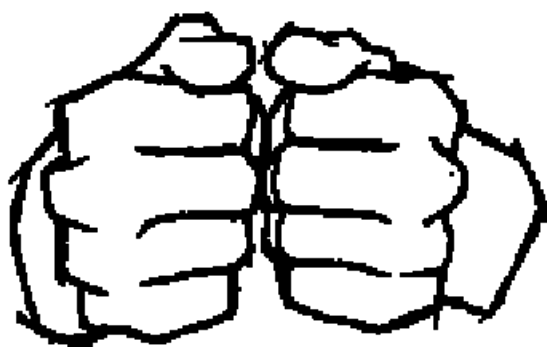
OUT



SINGLE



STOP



WITH

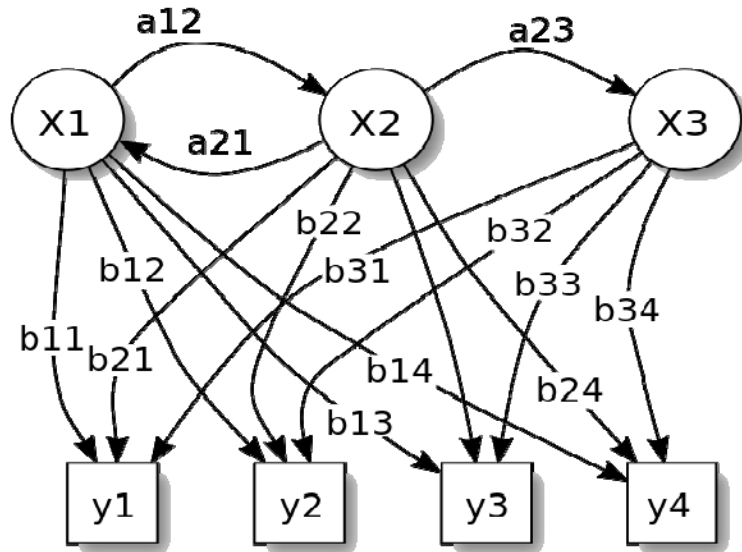


WHERE

Figure 2.6. American Sign Language gesture representation. Source: B. Vicars. Basic Signs Pictures, n.d. <http://www.lifeprint.com/asl101/pages-layout/concepts.htm>, accessed Dec. 2011.

another stochastic process that produces a sequence of observations (see Figure 2.7 [20]).

Thus, if  $S = \{S_n, n = 1, 2, \dots\}$  is a Markov process and  $\Omega = \{\Omega_k, k=1, 2, \dots\}$  is a function of  $S$ , then  $S$  is a hidden Markov process (or hidden Markov model) that is observed through  $\Omega$ , and we can write  $\Omega_k = f(S_k)$  for some function  $f$ . In this way we can regard  $S$  as the state process that is hidden and  $\Omega$  as the observation process that can be observed.



**Figure 2.7. Probabilistic parameters of a hidden Markov model. Source: Wikipedia. File:HiddenMarkovModel.svg, 2012. <http://en.wikipedia.org/wiki/File:HiddenMarkovModel.svg>, accessed Apr. 2012.**

A hidden Markov model is usually defined as a 5-tuple  $(S, \Omega, P, \phi, \pi)$ , where

- $S = \{s_1, s_2, \dots, s_n\}$  is a finite set of  $N$  states.
- $\Omega = \{o_1, o_2, \dots, o_m\}$  is a finite set of  $M$  possible symbols.
- $P = \{p_{ij}\}$  is the set of state-transition probabilities, where  $p_{ij}$  is the probability that the system goes from state  $s_i$  to state  $s_j$ .
- $\Phi = \{\phi_i(o_k)\}$  are the observation probabilities, where  $\phi_i(o_k)$  is the probability that the symbol  $o_k$  is emitted when the system is in state  $s_i$ .
- $\Pi = \{\pi_i\}$  are the initial state probabilities; that is  $\pi_i$  is the probability that the system starts in state  $s_i$ .

As the states and output sequence are understood, it is customary to denote the parameters of an HMM by  $\lambda = (P, \Phi, \pi)$ . [21]

### 2.3.1 Architecture of HMM

The above figure represents the general architecture of an instantiated HMM, where  $x$  represents states,  $y$  possible observations,  $a$  state transition probabilities and  $b$  output probabilities. Each circle represents a random variable that can adopt any of a number of values. The random variable  $x(t)$  is the hidden state at time  $t$ . The random variable  $y(t)$  is the observation at time  $t$  (with  $y(t) \in \{y_1, y_2, y_3, y_4\}$ ). The arrows in figure denote conditional dependencies. This figure is often called as trellis diagram.

As the figure clearly states that the conditional probability distribution of the hidden variable  $x(t)$  at time  $t$ , given the values of the hidden variable  $x$  at all times, depends only on the value of the hidden variable  $x(t-1)$ : the value at time  $t-2$  and before have no influence. This is called Markov property. Similarly, the value of the observed variable  $y(t)$  only depends on the value of the hidden variable  $x(t)$  (both at time  $t$ ). The Markov process itself cannot be observed, and only the sequence of labeled circles can be observed thus this arrangement is called a “hidden Markov process.” [21]

### 2.3.2 Previous Use of HMM's for Recognizing Sign Language

In the last decade or so HMM was widely used for the recognition of ASL. People ask questions like what evidence is there that HMM can be eventually used to address the full ASL recognition problem. The answer to that question confined in the correlation of the ASL recognition domain with the continuous speech recognition domain where HMM has become the technology of choice. As both continuous speech and sign language share common characteristics. Sign language can be seen as indicator (position, shape, orientation of the hands, etc.) over time, just like speech. Detection of the silence is considerably easy to detect in both speech and ASL. Also hesitating between signs will be considered equivalent to wobble or “ums” in speech and the complete information required to specify a fundamental unit in both domains is given regularly in a constraint time frame. The start and end mark of a sign depend on the temporarily neighboring signs. Synonymously, spoken phonemes change due to co-articulation in speech. In both territories, the basic units combine to form more complex wholes (words to phrase in signs and phonemes to words in speech).

In spite of the above similarities, sign language recognition has some basic differences from speech recognition. Unlike speech where phonemes combine to create

words, the basic unit in most of ASL is complete word itself. Therefore, there is not as much support as for individual word recognition in sign as there is in speech. In addition, the basic unit in sign can switch abruptly (like, changing into finger spelling for proper nouns). Moreover, the grammar in ASL is significantly different from that of English speech. Even given these difficulties, there seems the strong likelihood that HMM can apply to sign language recognition.

Hidden Markov models have inherent properties that constitute them very enthralling for ASL recognition. All that is required for training, except when using an optional bootstrapping process is a data and text matching the signs. The process of training aligns the components of the transcription to the data automatically by itself. So, no special effort is required to label training data [21].

Recognition is also performed on a continuous data current. As we know no explicit, segmentation is necessary. The division of sentences into words occurs naturally by introducing the use of lexicon and a language model into the recognition process. The output is continuous text that can be compared to a reference text for error calculation. Consequently, sign language recognition seems an ideal machine vision application of HMM technology.

Some previous work that is based on HMM and contribution in recognition of sign language includes Chen et al. [22] enroot a vision-based hand gesture recognition for Taiwanese Sign Language, which recognize continuous gesture using a real-time tracking method. The system traces the moving hands and extracts the hand area using a real-time hand tracking and extraction algorithm. Hienz, Bauer, and Kraiss [23] developed a video-based signer dependent continuous system for German Sign Language Recognition using HMM and stochastic grammars. In this research, the researcher uses colored cotton gloves. The features were extracted from the hand parameters including hand shape, hand orientation, and hand position. Many more researches were performed based on HMM model, but these researches are not enough to implement to the real time world due to their limitations.

HMM methodology is related to techniques that have been used previously in vision with success. Dynamic time warping, expectation maximization, Q-learning, and several other pattern recognition technique resembles portion of modeling and recognition process.

HMM has advantage over above mentioned techniques but now the Neural network is creating an edge over HMM and is highly in use because neural networks technique have advantages over HMM such as the ability of selectivity, knowledgeably, and scalable tailor the model to the task at hand. These abilities of NN provide an extra advantage over HMM. Even HMM, model is still in the use in wider range. However, new emerging techniques are using NN in recognition of sign language.

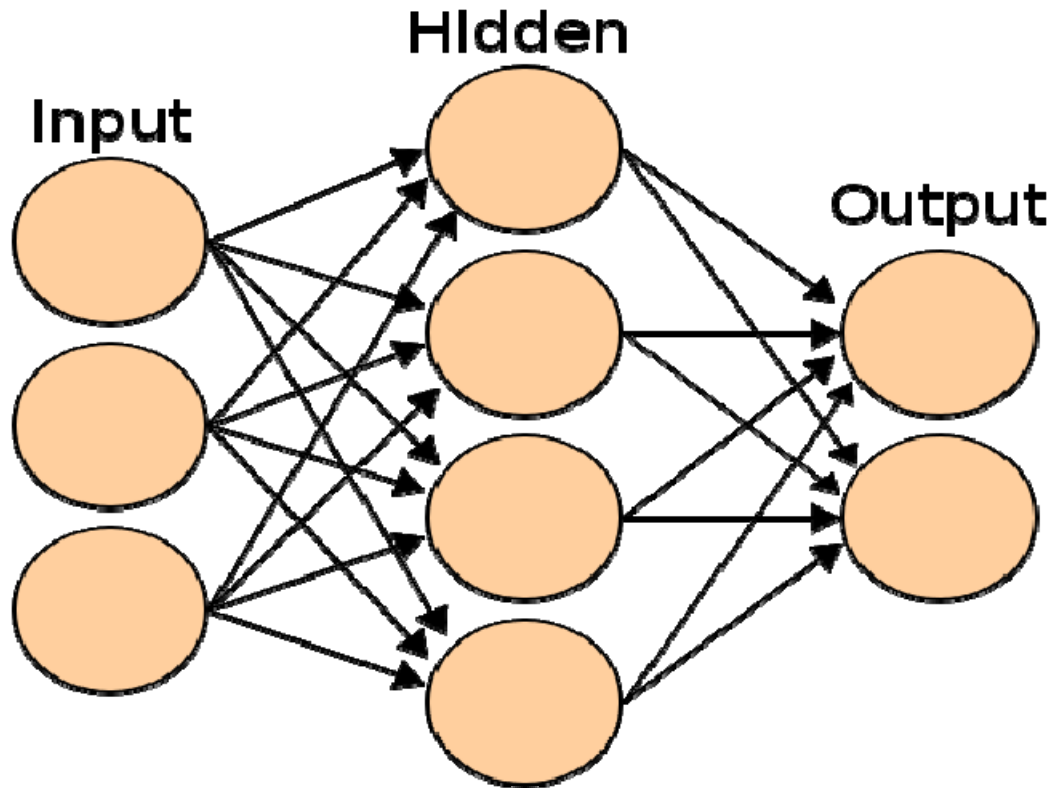
## 2.4 WHAT IS ARTIFICIAL NEURAL NETWORKS?

Neural network (NN) is another term to refer artificial neural network (ANN). A mathematical model or computational model is possessed by the configuration and/or practical aspects of biological neural networks. A neural network subsists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. An ANN is a flexible system that reforms its model according to external or internal information that flows through the network during learning phase. Modern neural networks are non-linear statistical data modeling tools that are used to find the complex relationships between inputs and outputs or to find pattern in data. Models of neural network in artificial intelligence are usually treated as artificial neural networks (ANNs); these models refer the simple mathematical models that define a function  $f: X \rightarrow Y$  or a distribution over  $X$  or both  $X$  and  $Y$ , but in few instances models are also involved with appropriate learning algorithm or any other learning protocol. General use of the terminology ANN model really means the definition of a *class* of such functions where the associate of the class are obtained by irregular parameters, connection weights, or specifics of the architecture like the count of neurons or their bond [24].

The text network in ‘artificial neural network’ refers to the correlation between the neurons in the different layers of each system. Let us take an example of a system having three layers. The very first layer contains the neurons that are marked as input that transfers data via synapses to the next layer that is the second layer. On layer, two this data meets with more neurons and again with more number of synapses it transfers to third layer, which is output neurons. The number of layers depends on the complexity of system. The more complex system has the more number of layers.

Three parameters that define ANN are:

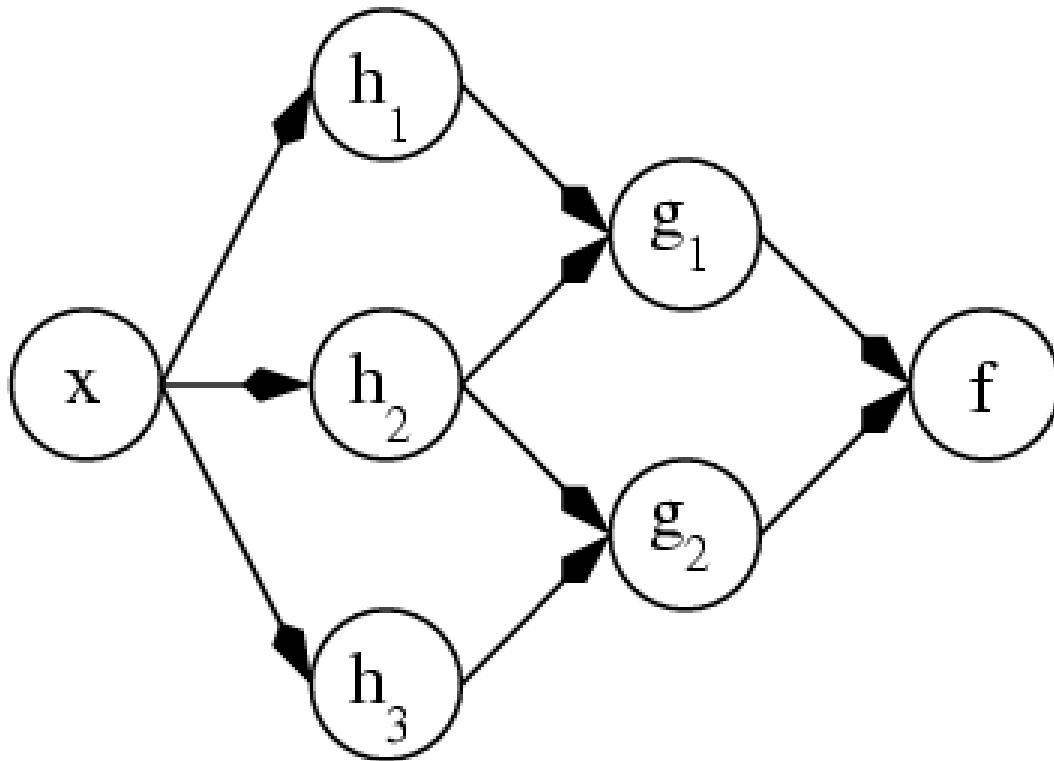
1. Interconnection pattern between different layers of neurons (Figure 2.8 [24]).
2. Learning process for updating weights of the interconnections.
3. Stimulating function that transforms a neuron's weighted input to its output activation. [24]



**Figure 2.8. An example artificial neural network with a hidden layer. Source: Wikipedia. Artificial Neural Network, 2012. [http://en.wikipedia.org/wiki/Artificial\\_neural\\_network](http://en.wikipedia.org/wiki/Artificial_neural_network), accessed Feb. 2012.**

A dependency graph for an artificial neural network (Figure 2.9 [24]). The variable  $h$ , which is a 3-dimensional vector, depends on  $x$ , the input variable. Variable  $g$ , which is a 2-dimensional vector variable depends on  $h$ , and finally, variable  $f$ , the output representative depends on  $g$ . In this network, the vector variables can be further decomposed in parallel units. This means that  $h_1$ , for example, doesn't depend on  $h_2$  given  $x$ .

Mathematical Function  $f(x)$  of neuron network is defined as combination of other functions  $g_i(x)$  that can also be defined as the composition of other functions. This hierarchy of functions can be easily represented as network structure, with the dependencies between variables. [25]

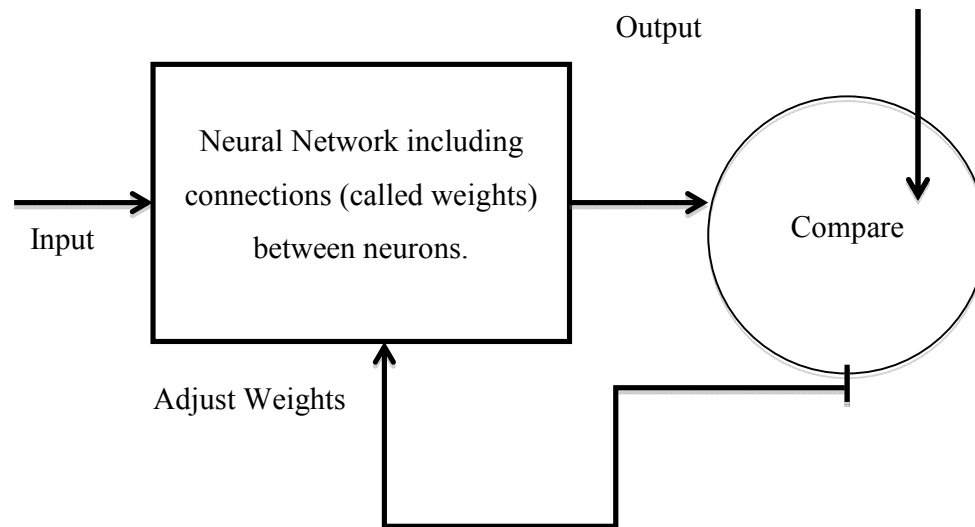


**Figure 2.9. ANN Dependency graph. Source: Wikipedia. Artificial Neural Network, 2012. [http://en.wikipedia.org/wiki/Artificial\\_neural\\_network](http://en.wikipedia.org/wiki/Artificial_neural_network), accessed Feb. 2012.**

Neural networks are accommodate, or trained, to get the desired output with the particular input. Figure 2.10 [26] demonstrate how the network is arranged; depending on a comparison of the output and the target, before the network output matches the target. Commonly countless of input/target pairs are used, in supervised learning, this is a training method, to train a network.

Neural networks have been trained to achieve complex functions in diverse fields of application including identification, classification, speech, vision, and pattern recognition and control systems.

Trained neural networks can solve these today's difficult problems, which are not easy to solve by conventional computers or human beings. Supervised training approaches are commonly used, but some networks can be trained from unsupervised training techniques or from direct design method. Unsupervised networks can be used, for instance, to identify groups of data. Some networks like linear and Hopfield are designed directly. One,



**Figure 2.10. Neural Net Block diagram. Source: K. Symeonidis. Hand gesture recognition using neural networks. Master’s thesis, University of Surrey, Guildford, Surrey, UK, 2000.**

conclusion from this is that there are various types of design and learning techniques that improve the choices that a user can make.

History of neural networks has almost five decades long but found solid application only in the past ten years, and the field is still growing rapidly. Therefore, it is apparently different from the fields of control systems or optimization where the terminology, basic mathematics, and design procedures have been firmly established and applied for many years. [26]

### 2.4.1 Supervised Learning

This learning depends on the system, which attempts to predict outcomes for known examples and is commonly used training method. It matches its predictions to the target answer and “learns” from its mistakes. For the input layer of neuron data will feed as input. These input layer neurons transfer the inputs to the next nodes. While the inputs are transferring along, the weighting, or connection, is enforced and when the inputs reach the next node, the weightings are summed and either intensified or weakened. These steps will continue until the data reaches the output layer where the model predicts the outcome. In this learning the output, which is predicted, is distinguished to the actual output for that case. Now if both the output matches there will be no change in the weights in the system. However, if the predicted output is higher or lower than the actual outcome in the data then



the error flag is raised and generated error will be pushed back in the system and then weights are adjusted accordingly. This feeding of error backward is named as “back-propagation.”

### **2.4.2 Unsupervised Learning**

The most effective for describing data rather than predicting it is applied to neural networks when they are trained with unsupervised learning. The neural network is not shown any outputs or answers as part of the training process. Means, in this type of system there is no concept of output fields. A Kohonen network is the primary unsupervised technique. Generally cluster analysis use Kohonen and other unsupervised neural systems. The main advantage of the neural network for this type of analysis is that it requires no initial assumptions about what are elements of group or about the number of groups. The system begins with the clean slate and is not biased about which factors should be most important.

## **2.5 WHAT IS RADIAL BASIS FUNCTION NEURAL NETWORK?**

Radial basic function is a neural network tool that is generally used for classification. This neural network function emerged as a variant of artificial neural network in late 80's. RBF's are embedded in a two layer neural network, where each hidden unit implements a radial activated function. The output units implement a weighted sum of hidden unit outputs. The input into an RBF network is nonlinear while the output is linear. Due to their nonlinear approximation properties, RBF networks are able to model complex mappings, which perceptron neural networks can only model by means of multiple intermediary layers.

In order to use radial basic function Network we need to specify the hidden unit activation function, the number of processing units, a criterion for modeling a given task and a training algorithm for finding the parameters of the network. Finding the RBF weights is called network training. If we have at hand a set of input-output pairs, called training set, we optimize the network parameters in order to fit the network outputs to the given inputs. After training, the RBF network can be used with data whose underlying statistics is similar to that of the training set. RBF networks have been successfully applied to a large diversity of applications including interpolation, chaotic time-series modeling, system identification, control engineering, electronic device parameter modeling, channel equalization, speech

recognition, image restoration, shape from-shading, motion estimation and moving object segmentation, data fusion, etc.

Radial basic functions are embedded into a two-layer feed forward neural network. Such a network is characterized by a set of inputs and a set of outputs. In between the inputs and outputs there is a layer of processing units called hidden units. Each of them implements a radial basic function. The way in which the network is used for data modeling is different when approximating time-series and in pattern-classification.

A radial basic function (RBF),  $\phi$ , is one whose output is symmetric around an associated center,  $\mu_c$ . That is,

$$\phi_c(\mathbf{x}) = \phi(\|\mathbf{x} - \mu_c\|)$$

where  $\|\cdot\|$  is a vector norm. A set of RBFs can serve as a basis for representing a wide class of functions that are expressible as linear combinations of the chosen RBFs:

$$y(\mathbf{x}) = \sum_{j=1}^M w_j \phi(\|\mathbf{x} - \mu_j\|)$$

A radial basis function network (RBFN), is nothing but an embodiment of above equation as a feed forward network with three layers: the inputs, the hidden/kernel layer and the output node(s). Each hidden unit represents a single radial basis function, with associated center position and width. Such hidden units are sometimes referred to as centroids or kernels. Each output units performs a weighted summation of the hidden units, using the  $w_j$ s as weights. [27]

## 2.6 LIMITATIONS IN NEURAL COMPUTING

Every computing has some limitations. Neural network computing also have some. Main limitation of neural network is its inability to explain the model that has built in a useful way. Researchers wants to know why the model is behaving as it is. Good answers are provided by neural network but they are hard to understand and also neural network faces hard time to explain how they reach at outcome.

Other limitations include extraction of protocols is difficult from neural networks. Extraction of protocol is important for peoples who what to explain their answers to others and to researchers who are involved with artificial intelligence, specifically expert systems which are rule-based.

For getting a good answer from the analytical method, you cannot just throw the data at a neural net but you have to spend some time in understanding the problem for the output that you are trying to predict. In addition, the data used to train the model should be appropriate. If the data are not expressive of the problem, neural computing will not produce good results. This will become the classic situation where “garbage in” will definitely produce “garbage out.”

So definitely, it takes time to train a model with a very complex data set. Neural techniques requires high end computer as this technique is computer intensive and will run slow on low end PC's or machines with no math coprocessors. Even it consumes lot of time to train the model but the overall time will be much faster than the other data analysis approaches. Unlike other analytical approaches neural network is independent of time require for programming, debugging or testing assumptions also processing speed. [26]

## **2.7 ADVANTAGE OF NEURAL NETWORK OVER HMM**

There are few advantages of neural network over HMM but an important one that an analyst realizes after using neural network.

- Neural network structure will attempt to find the connection i.e. a function between the inputs, and the provided outputs, so that whenever the net will be provided with invisible inputs, and with the appropriate weights, it will try to search a correct answer for the new inputs. Whereas hidden Markov models, are used to find the states for which a given stochastic process went through.
- Pattern recognition is an impressive technique for channelizing the information in the data and generalizing about it. Also neural nets master to distinguish the patterns, which exit in the data set.
- One of the most important advantages of neural network is after the training process; neural networks are capable of predicting next state of the system based only on the last state. In addition, neural networks are capable of measuring the prediction error and adapting themselves and they take online changes in the underline process to improve the model of prediction and decrease the estimation error for the next state.
- The system is established through learning process rather than programming. Programming is much more time consuming and complex and need the analyst to define the exact behavior of the model.
- Neural networks are adjustable in the changing environment. Rule based system like HMM or programmed systems are limited to the situation for which they were designed—whenever the environment changes, they are no longer capable to work. Even neural networks may take time to adjust and learn an abrupt change, but they are most efficient at adapting to constantly changing information.

- When the talk comes about performance neural network has an upper hand over HMM. It is at least as good as classical statistical modeling, and better on most problems. The model that uses the approach of neural network significantly takes less time.
- While neural networks are computationally intensive, they can now operate on modest hardware due to the routines that are optimized to a point that they can now run in acceptable time on a personal computer. They no longer require supercomputers as they did in the early days of neural network research. [26]

## **CHAPTER 3**

### **APPROACH**

#### **3.1 PROPOSED AMERICAN SIGN LANGUAGE HAND GESTURE TO VOICE ARCHITECTURE**

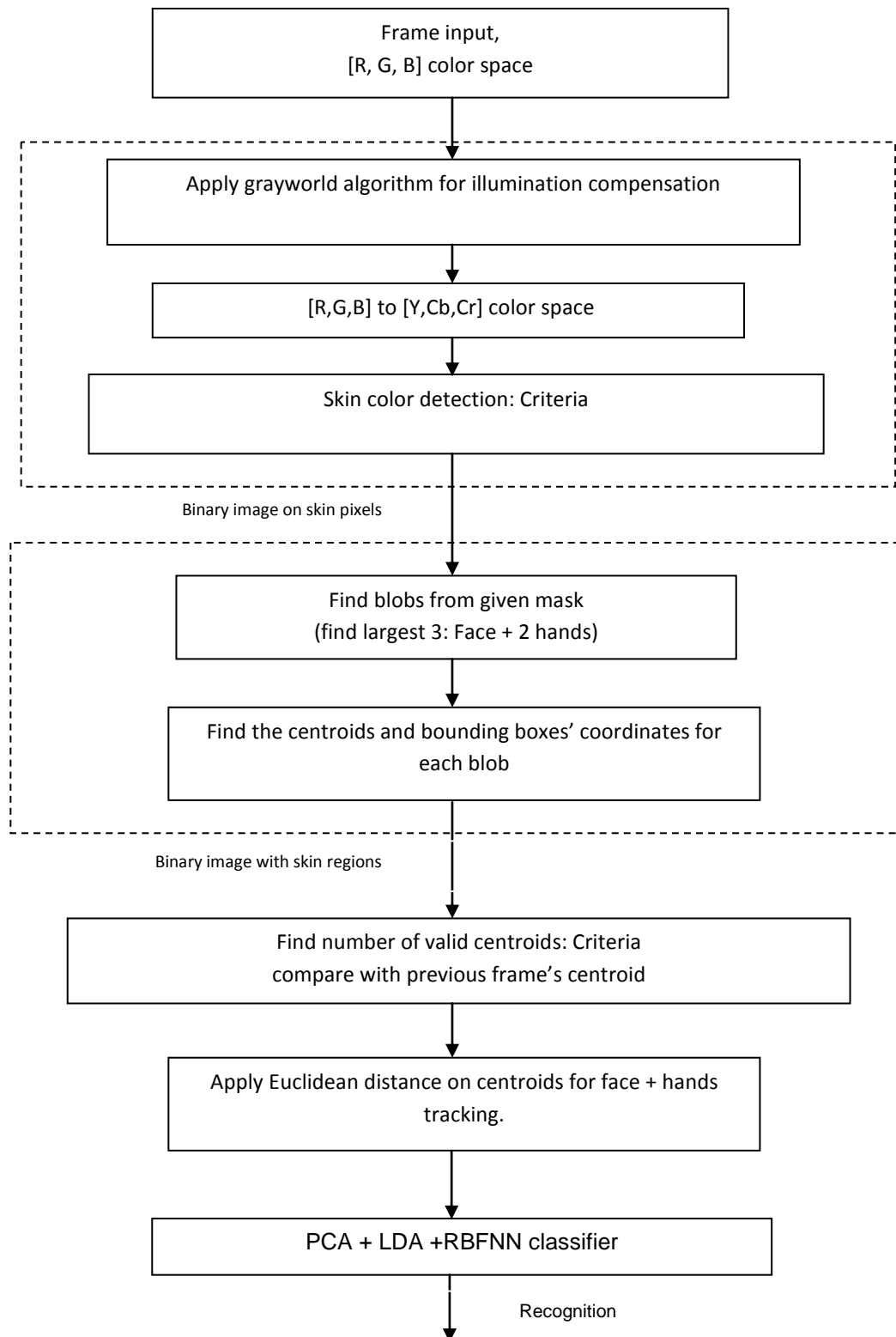
The proposed architecture of Sign language gesture recognition is depicted in Figure 3.1. The system kick starts with the image pre-processing of skin detection. Within the stage of skin detection, there are grayworld illumination compensation, color space conversion from RGB to YCbCr, and skin detection via threshold. Then, skin segmentation via blob is performed. The detected skin regions are represented with centroids and tracked via Euclidean distance measurement. Lastly, dimension reduction and feature extraction algorithm of Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). These algorithms are essential for data representation in transforming in to a more intelligent form. Finally, Radial Basis Function Neural Network (RBFNN) is used for classification to recognize difference hand gestures. Subsequent chapters will cover the theoretical and result outcomes.

#### **3.2 ASL IMAGE FEATURE EXTRACTION**

This topic covers all the necessary detailed explanations, from skin detection, PCA feature extraction algorithm to dimension reduction approach of LDA. Among the image filtering techniques, the use of skin detection algorithm stands out from the rest (frame differencing, background subtraction) for its simplicity yet good performance yield. As the system is image-based gesture recognition, features from the filtered hand need to be extracted intelligently. To accomplish this, the use of Principal component Analysis (PCA) and Linear Discriminant Analysis (LDA) is employed. This topic serves as the in depth knowledge provider and the actual results are included in following chapter.

##### **3.2.1 Skin Detection**

In this project, skin detection is achieved through the transformation from RGB (Red, Green, Blue) to YCbCr color space. Before any of this, image frames are passed to



**Figure 3.1. Proposed American Sign Language hand gesture to voice architecture.**

illumination compensation algorithm of Grayworld. The Grayworld algorithm is simple in terms of its operation and complexity. Based on this assumption, the Grayworld algorithm produces an estimate of the scene illuminant by computing the mean of each RGB element of an image

$$[\mathbf{R}_E, \mathbf{G}_E, \mathbf{B}_E]^T = [\mathbf{mean}(\mathbf{R}), \mathbf{mean}(\mathbf{G}), \mathbf{mean}(\mathbf{B})]^T$$

Where  $[\mathbf{R}_E, \mathbf{G}_E, \mathbf{B}_E]^T$  is the estimate RGB vector of the illuminant. This algorithm computes the mean of every RGB component value in an image. Despite its simplicity, this algorithm produces good results when large numbers of different colors are present in the image, and the image is viewed under a single uniform illuminant.

The unfair advantage of using the YCbCr color space against the rest lies in the benefit of it yields outstanding modeling for human skin color. Other than that, factors that need to be taken into consideration include applicability and effectiveness towards the customized system. In brief, two of the obvious advantage of skin detection performed in YCbCr color space includes:

- Effective use chrominance information for human skin modeling.
- Adoptability in video coding, which assist in reduced computation needed. [28]

YCbCr is a family of color where Y is the luma component and Cb and Cr are the blue-difference and red-difference Chroma components.

Although threshold has its fixed parameters for skin color detection that is a limitation, however this is the best technique devised till today [29]. The code below shows the fixed parameters I am referring to:

```
if ( (Cb[x][y] < 173) &&
    (Cb[x][y] > 133) &&
    (Cr[x][y] < 127) &&
    (Cr[x][y] > 77)
    )
    setPixel(x, y, 255);
else
    setPixel(x, y, 0);
}
```

These parametric limitations are covered extensively in the limitation section of this book.

By converting RGB to YCbCr color space, the luma component, Y, which does not contribute to skin detection, is omitted while Cb, Cr component are used. The equation, which governs, is expressed matrix form of:

$$\begin{aligned} Y &= 0.257 * R + 0.504 * G + 0.098 * B + 16 \\ Cb &= -0.148 * R - 0.291 * G + 0.439 * B + 128 \\ Cr &= 0.439 * R - 0.368 * G - 0.071 * B + 128 \end{aligned}$$

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

It is worth to emphasize that skin detection serves as image pre-filtering. Detected skin regions are Box-bounded. Further computation/executions performed on the segmented skin are done in grayscale color space.

### 3.2.2 Blob Detection

Upon skin segmentation, hand and face regions will both be identified as different blobs. Such blobs labeling is necessary to enable tracking on both hand and face (via Euclidean distance). For blob detection algorithm to execute, detected skin regions are first converted in to binary form. The algorithm starts by iteration on each pixel in the binary format. For each detected pixels, the neighboring pixel is checked whether it belongs to the same region recursively. The algorithm continues until all the pixels iterated. In another way, it is finding the connection pixels within the binary form.

### 3.2.3 Principal Component Analysis

This subsection will explain in details on PCA. Principal component analysis (PCA) is a classical linear feature extraction method. It is based on the analysis of the second order statistics of data, in particular, the eigenvalue analysis of the covariance matrix. The flow of PCA starts by storing the sorted training hand gesture image for all classes in vector of size N. (Assuming there's P images)

$$x^i = [x_1^i, x_2^i, x_3^i, \dots, x_N^i]^T$$

Then, the images are mean centred. This is done by subtracting the mean from each image vectors.



$$\overline{x^i} = x^i - m, \text{ where } m = \frac{1}{p} \sum_{i=1}^p x^i$$

$$\overline{X} = \begin{bmatrix} \overline{x^1} & \overline{x^2} & \overline{x^3} & \dots & \overline{x^p} \end{bmatrix}$$

The resulting vectors are combined (horizontally concatenated) forming a matrix of size NxP. The covariance is then calculated by multiplying by its transpose. It's notable that the covariance could result in too large of value. An alternative way of creating the covariance is by:

$$\Omega = \overline{XX^T}$$

The eigenvectors (characteristic vector), 'e' and its associated eigenvalues (characteristic value), 'R' are found by solving the covariance. The eigenvectors are then sorted in descending order, according to their associated eigenvalues. The eigenvectors corresponding to the largest eigenvalue is the eigenvector that finds the greatest variance in the gesture images. This applies to the second eigenvectors and so on. The eigenvectors corresponding to the smallest eigenvalues is hence the least discrepancy in the images.

### 3.2.4 Linear Discriminant Analysis

Linear discriminant analysis further performs the dimension reduction and features extractions. Here, LDA tends to group the images of the hand gestures of the same class and separate them from other classes. This process can indirectly improve the classification stage later. To begin with, two measures are defined, (a) within class scatter matrix and (b) between class scatter matrix. Within class scatter matrix finds the amount spread out between images of the same class (there are 6 classes). For the  $i$ th class, the scatter matrix is calculated as the sum of covariance matrixes (of the centered images in that class). These scatter matrixes of every class are added together forming the within class scatter matrix,  $S_w$ .

$$S_w = \sum_{j=1}^c \sum_{i=1}^{N_i} (x_i^j - \mu_j)(x_i^j - \mu_j)^T$$

where:

$x_i^j$  is the  $i^{\text{th}}$  sample of class  $j$ ,

$\mu_j$  is the mean of class  $j$ ,

$c$  is the number of classes,

$N_i$  is the number of samples in class  $j$ .

The second measure, between class scatter matrix, measures the amount of scatter between classes. It is calculated as the sum of covariance matrices of the difference between the total mean and the mean of each class.

$$\mathbf{S}_B = \sum_{i=1}^c (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$$

where:

$\boldsymbol{\mu}_i$  is the mean of  $i^{\text{th}}$  class,

$\boldsymbol{\mu}$  is the total mean.

To operate the LDA optimally, the projection of image for this project is done in five-dimensional space (6 classes -1). Feature extraction and dimension reduction are essential tools in image processing. Data transformations of such kind are widely applied for their benefit in:

- Reduced data size for computation.
- Better presentation of data for classification purposes.

Hand image gestures are dimensionally reduced before fed to classifier of RBFNN.

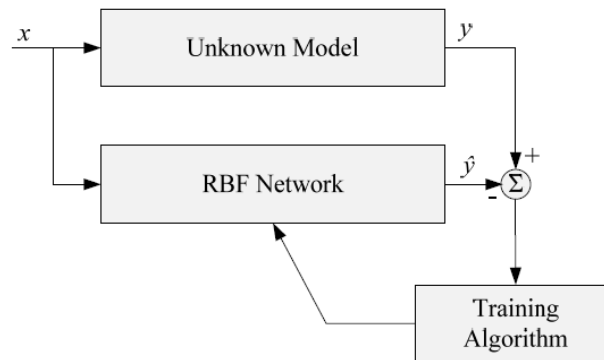
### 3.3 RADIAL BASIS FUNCTION NEURAL NETWORK

As I have talked about this topic in the last chapter and this is the core ingredient of our thesis so I really like to talk it further and describe the fundamental theory of Radial Basis Function and how it is applied to suit the project's need. A radial basis function network (RBFN) is a special type of neural network that uses a radial basis function as its activation function. RBF networks are very popular for function approximation, curve fitting, time series prediction, control, and *classification* problems. The radial basis function network is different from other neural networks, possessing several distinctive features. Because of their universal, approximation, more compact topology, and faster learning speed, RBF networks have attracted considerable attention and they have been widely applied in many science and engineering fields.

In RBF networks, determination of the number of neurons in the hidden layer is very important because it affects the network complexity and the generalizing capability of the network. If the number of the neurons in the hidden layer is insufficient, the RBF network

cannot learn the data adequately; on the other hand, if the neuron number is too high, poor generalization or an overlearning situation may occur. The position of the centers in the hidden layer also affects the network performance considerably, so determination of the optimal locations of centers is an important task. In the hidden layer, each neuron has an activation function. The Gaussian function, which has a spread parameter that controls the behavior of the function, is the most preferred activation function. The training procedure of RBF networks also includes the optimization of spread parameters of each neuron. Afterwards, the weights between the hidden layer and the output layer must be selected appropriately. Finally, the bias values, which are added with each output, are determined in the RBF network training procedure.

Neural networks are non-linear statistical data modeling tools and can be used to model complex relationships between inputs and outputs or to find patterns in a dataset. RBF network is a type of feed forward neural network composed of three layers, namely the input layer, the hidden layer, and the output layer. Each of these layers has different tasks. A general block diagram of an RBF network is illustrated in Figure 3.2.



**Figure 3.2. Block diagram of a RBF network.**

In RBF networks, the outputs of the input layer are determined by calculating the distance between the network inputs and hidden layer centers. The second layer is the linear hidden layer and outputs of this layer are weighted forms of the input layer outputs. Each neuron of the hidden layer has a parameter vector called center. Therefore, a general expression of the network can be given as:

$$\hat{y}_j = \sum_{i=1}^I w_{ij} \phi(\| \mathbf{x} - \mathbf{c}_i \|) + \beta_j$$

The norm is usually taken to be the Euclidean distance and the radial basis function is taken to be Gaussian function and defined as follows:

$$\phi(\mathbf{r}) = \exp(-\alpha_i \cdot \| \mathbf{x} - \mathbf{c}_i \|^2)$$

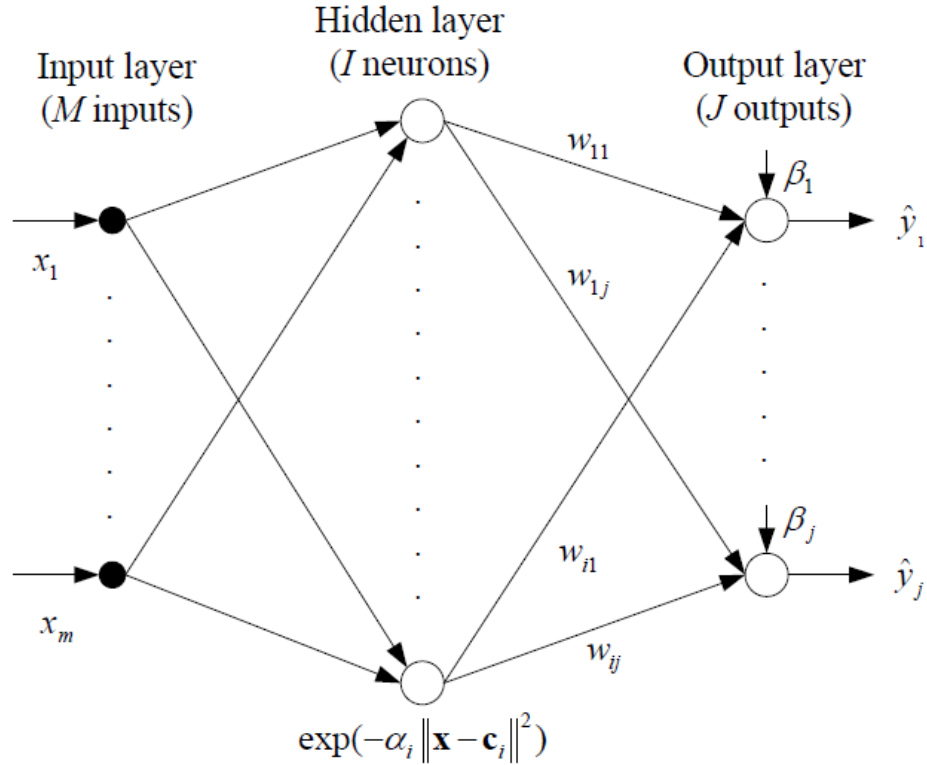
where,

$I$	Number of neurons in the hidden layer	$i \in \{1, 2, \dots, I\}$
$J$	Number of neurons in the output layer	$j \in \{1, 2, \dots, J\}$
$w_{ij}$	Weight of the $i^{\text{th}}$ neuron and $j^{\text{th}}$ output	
$\phi$	Radial basis function	
$\alpha_i$	Spread parameter of the $i^{\text{th}}$ neuron	
$\mathbf{X}$	Input data vector	
$\mathbf{c}_i$	Center vector of the $i^{\text{th}}$ neuron	
$\beta_j$	Bias value of the output $j^{\text{th}}$ neuron	
$\hat{y}_j$	Network output of the $j^{\text{th}}$ neuron	

Figure 3.3 [30] shows the detailed architecture of an RBF network.  $M$  dimensional inputs ( $x_1, \dots, x_m$ ) are located in the input layer, which broadcast the inputs to the hidden layer.

The hidden layer includes  $I$  neurons and each neuron in this layer calculates the Euclidean distance between the centers and the inputs. A neuron in the hidden layer has an activation function called the basis function. In the literature, the Gaussian function is frequently chosen as the radial basis function and it has a spread parameter to shape the curve ( $\alpha_1, \dots, \alpha_i$ ). The weighted ( $w_{11}, \dots, w_{ij}$ ) outputs of the hidden layer are transmitted to the output layer. Here,  $I$  ( $i = \{1, 2, \dots, I\}$ ) denotes the number of neurons in the hidden layer and  $J$  ( $j = \{1, 2, \dots, J\}$ ) denotes the dimension of the output. The output layer calculates the linear combination of hidden layer outputs and bias parameters ( $\beta_1, \dots, \beta_j$ ). Finally, the outputs of the RBF network are obtained ( $\hat{y}_1, \dots, \hat{y}_j$ ) [30].

The design procedure of the RBF neural network includes determining the number of neurons in the hidden layer. Then, in order to obtain the desired output of the RBF neural network  $w$ ,  $\alpha$ ,  $c$  and  $\beta$  parameters might be adjusted properly. Reference based error metrics such as mean square error (MSE) or sum square error (SSE) can be used to evaluate the



**Figure 3.3. Network architecture of the RBF. Source: Wikipedia. Radial Basis Function Network, 2012.**  
[http://en.wikipedia.org/wiki/Radial\\_basis\\_function\\_network](http://en.wikipedia.org/wiki/Radial_basis_function_network),  
 accessed Mar. 2012.

performance of the network. Error expression for the RBF network can be defined as follows:

$$E^{SSE}(w, a, c, \beta) = \sum_{j=1}^J (y_j - \hat{y}_j)^2$$

Here  $y_j$  indicates the desired output and  $\hat{y}_j$  indicates the RBF neural network output. The training procedure of the RBF neural network involves minimizing the error function.

### 3.4 GESTURE TO VOICE

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech. In this project, the speech synthesis employed is Window's text-to-speech – Microsoft Sam. The use of the existing speech synthesis is due to

the nature and objective of this project, as emphasis is weighed on gesture recognition rather than text-to-speech.

To use Microsoft SAM, the API is called from MATLAB environment in a unique way. First, the communication setup in Microsoft is needed. This is done with MATLAB inbuilt function known as “actxserver” and starts communication with Microsoft SAPI (Speech API). Then, MATLAB can now pass in text to the API and the synthesized speech is generated.

### **3.5 WHAT IS MATLAB?**

We have developed our thesis using MATLAB. The name MATLAB stands for matrix laboratory. Matlab is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation.
- Application development, including Graphical User Interface building
- Algorithm development
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Modeling, simulation, and prototyping

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of time it would take to write a program in a scalar non-interactive language such as C or Fortran. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

The reason that I have decided to use MATLAB for the development of this model is because of its toolboxes. Toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-Files) that extend the

MATLAB environment to solve particular classes of problems. It includes among others image processing and neural networks toolboxes. [26]

Toolbox and other built-in functions used in this project are:

- a) Radial Basis Function (newrb)
- b) Excel reading: “xlsread”
- c) RGB to grayscale color space conversion: “rgb2gray”
- d) Euclidian distance determination: “dist”
- e) Image resize: “imresize”

### 3.6 IMAGE DATABASE

The starting task of the project was the creation of a database with all the videos/images that would be used for training and testing. Nevertheless, to decide what kind of database should be used is itself a bigger task to achieve. As the database can have different formats. Images can be either hand drawn, digitized photographers or a 3D dimensional hand. In addition, for videos what kind of quality, resolution should we use. How many frames per second videos are best suited for our purpose? So keeping all these points in mind for skin detection to take effect, benchmark database from “*Purdue RVL-SLLL American Sign Language Database*” was used. [31] The reasons for the use this database is:

- It is well recorded with conditions under
  - Controlled
  - Less-controlled lighting environment.
- Variety in signs available
  - Hand shapes
  - Fingerspelling alphabets
  - Numbers of movement in single signs
  - Challenging example of short discourse narratives.
- Benchmark database use for possible recognition comparison in future research works.

## CHAPTER 4

### RESULTS

This chapter will cover all the simulated results ranging from image pre-filtering (skin detection), dimension reduction and feature extraction of PCA and LDA to Radial Basis Function Neural Network training and recognition. Along with these, explanation in conversion from text to speech is also presented.

#### 4.1 SKIN DETECTION

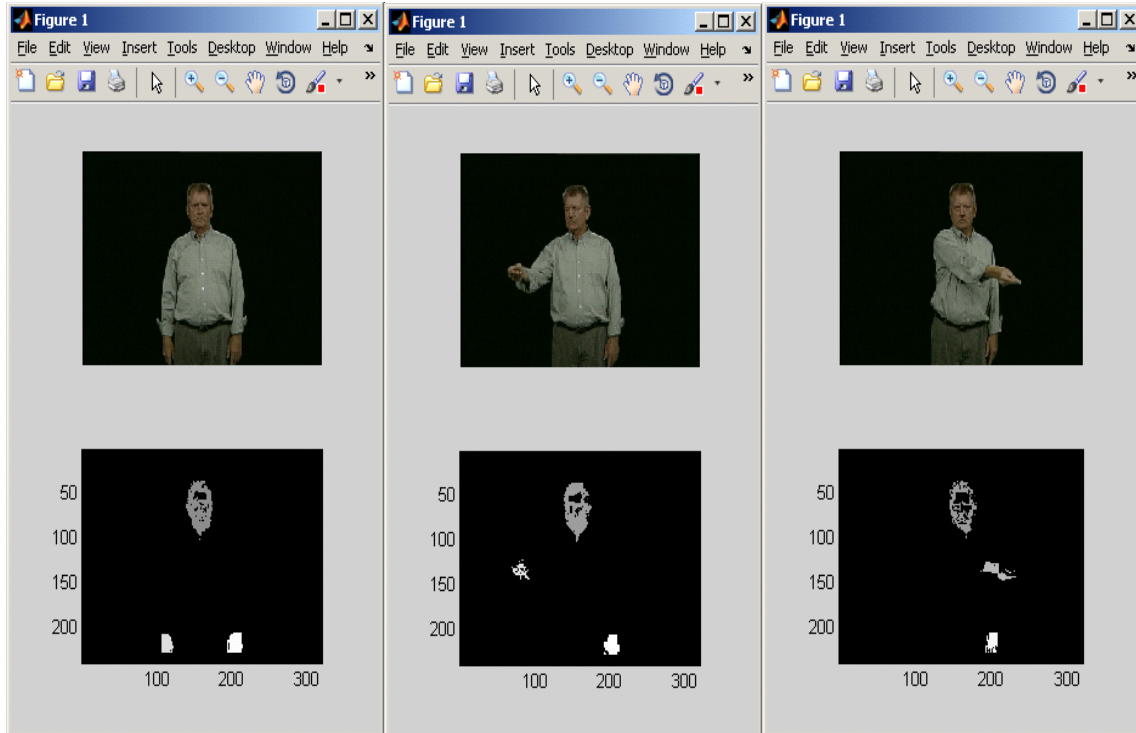
The theoretical explanation on skin detection was presented in the previous chapter. From the equation conversion from RGB to YCbCr, several experiments were conducted and adaptation towards this project's need was significantly noticed. For skin detection to take effect, benchmark database from "*Purdue RVL-SLLL American Sign Language Database*" was used.

For skin detection demonstration, videos from the database are passed into our system for initial image pre-filtering. In this project, assumption is made that only the three (3) largest skin regions will be segmented for further processing. This reduces the consideration and execution wastage for small and negligible regions.

Figure 4.1 is the actual skin detection outcome. The subject in Figure 4.1 is expressing signs of "left-to-right" and detected skin regions are showed by the bottom-window (right below the subject). It is noticeable for each detected skin the intensity varies. This is because the binary mask is labeled from the largest until the smallest detected skin regions. This will aid in further recognition processes, where recognition should be done in hand images rather than face.

Figure 4.2 is another skin detection result. The subject in Figure 4.2 is expressing signs of "in-to-out" and detected skin regions are displayed by the bottom window (right below the subject). From the extracted hand images, recognition will be executed. However, as explained in previous chapters, feature extraction and dimension reduction are crucial steps in recognition. Following this skin detection sub-section, segmented hand images are passed to PCA and LDA.





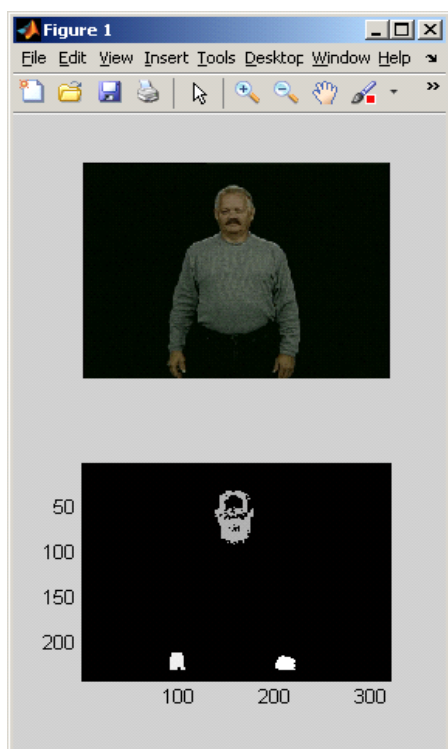
**Figure 4.1. Simulated outcome of skin detection.**

Principal Component Analysis breaks down hand gesture images into mathematical domain where interpretation can be realized. Upon feature extraction performed, the distinct features are weighted in descending order in the Principal Components. This means the most discriminant features are the first and second vectors, followed by third and fourth the list goes on.

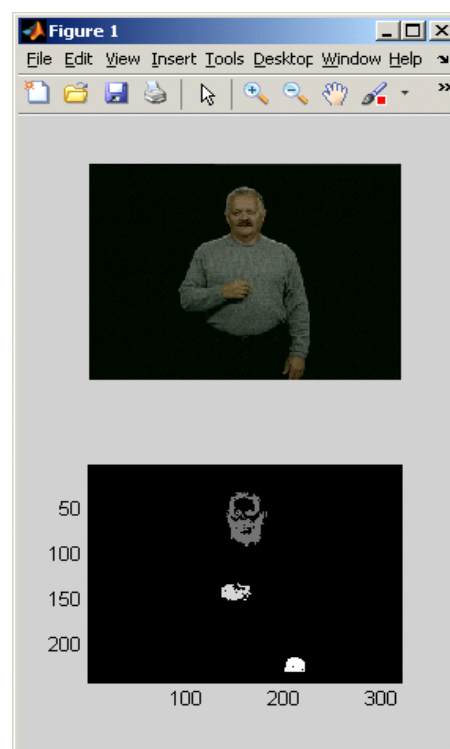
## 4.2 PCA FEATURE EXTRACTION

As the name indicates, class “A” represents images for the hand gesture, which expresses alphabets “A”. The same applies to class “B” to “E”. Class “NA” contains subject’s face images. This will be useful in later stage where recognition will seize upon detection of faces.

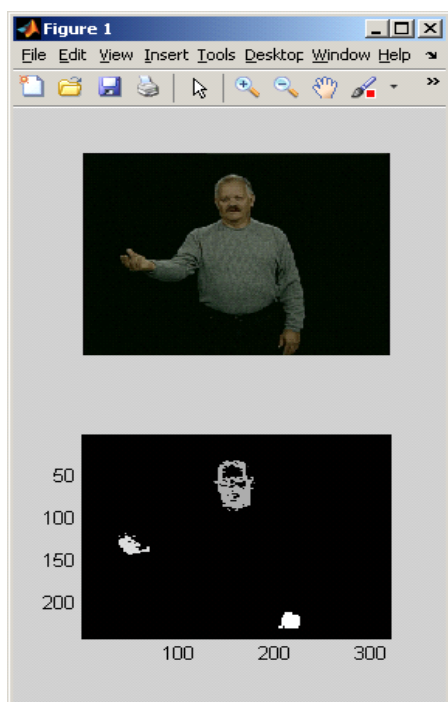
Figure 4.3 is the 2 dimensional plots of the first two vectors for 6 classes of alphabets (“A” to “E” and “NA”) (Table 4.1). Reflecting on PCA plot Figure 4.3, feature of each classes are observed to be well separated. Each cluster is a feature representation of its hand gesture images for their corresponding classes. It can be depicted that the features are well separated. Bare in mind, Figure 4.3 is the plot for the first two Principal Components.



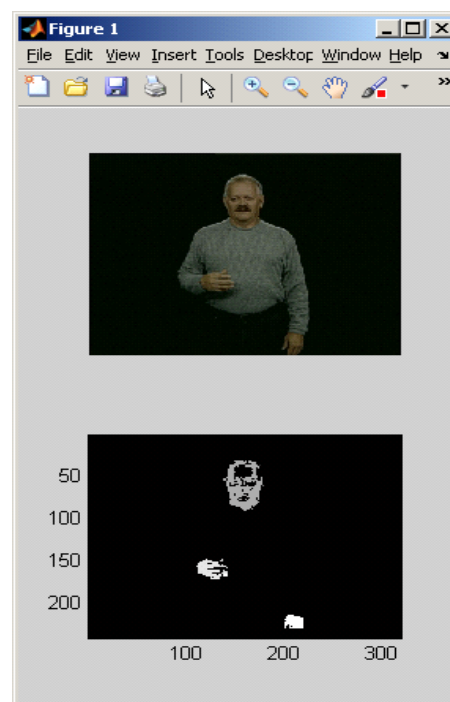
(a)



(b)



(c)

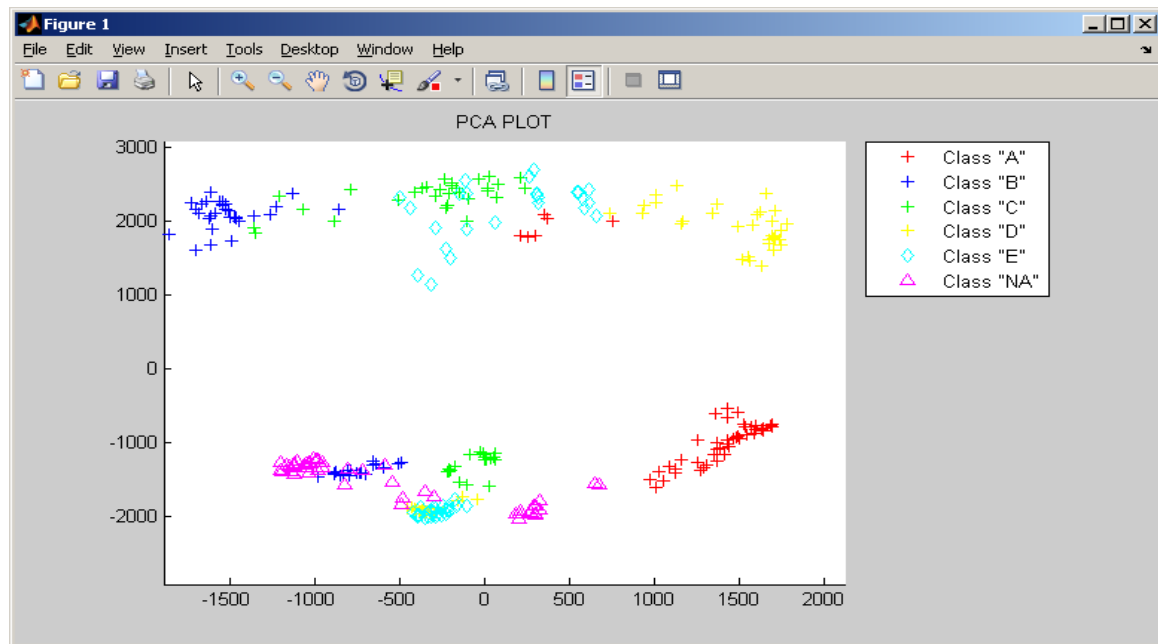


(d)

**Figure 4.2. Simulated outcome of skin detection.**

**Table 4.1. Alphabets with Classes**

Class	A	B	C	D	E	NA
Descriptions	Alphabet “A”	Alphabet “B”	Alphabet “C”	Alphabet “D”	Alphabet “E”	Contains other Alphabets and face images.

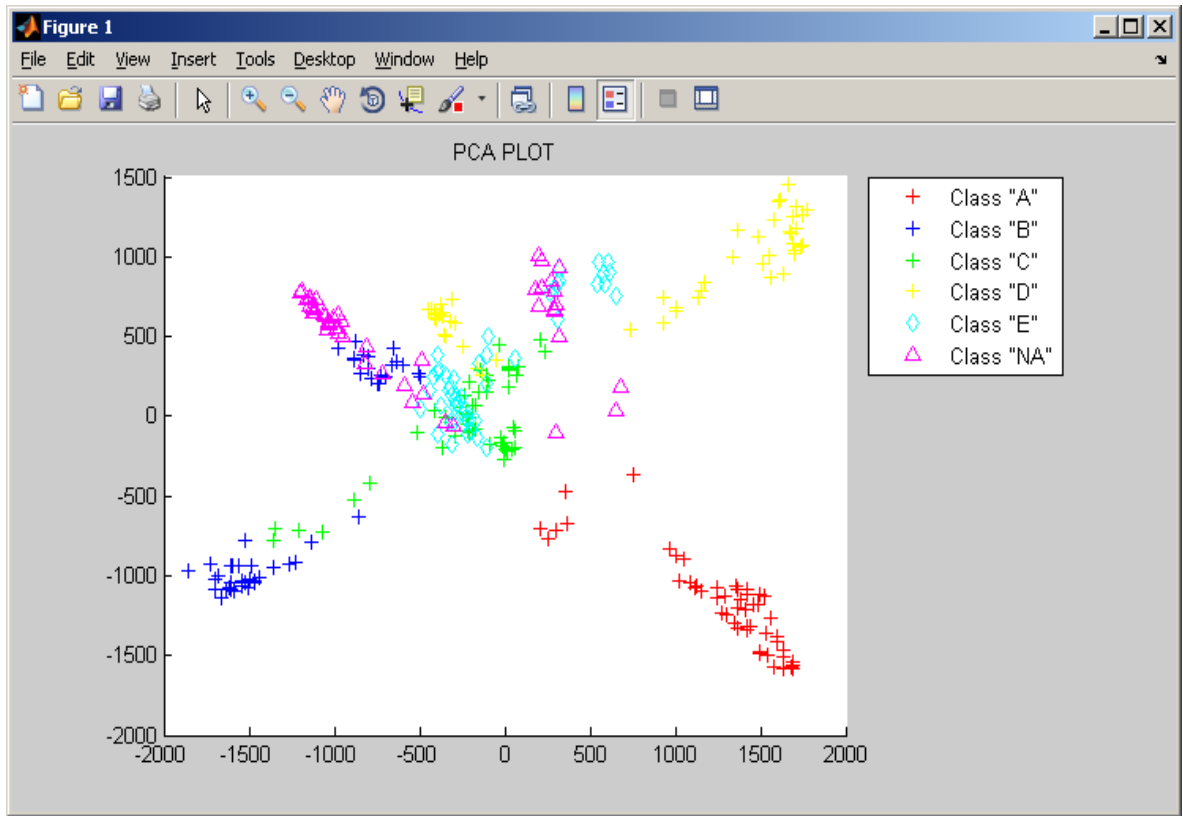
**Figure 4.3. PCA plot for the first two Principal Component.**

In Figure 4.4, the second and third Principal Component is plotted. This depicts the decrease in discriminant features as the. The cluster for each class is still significantly well separated.

However, as PCA algorithm is undeniably an unsupervised feature extraction tool, classification on these features can still be optimized. Thus, LDA is used.

### 4.3 LDA FEATURE EXTRACTION

As PCA breaks down hand gesture images of certain class into feature representation, LDA serves similarly purposes with an additional attribute of supervised feature extraction



**Figure 4.4. PCA plot for the second vs third Principal Component.**

made possible. This means the LDA feature can further separate classes with obvious efficiency.

Figure 4.5 depicts the first two vectors of LDA features. Here, it is observed that the features are further clustered within their classes while distance between each class is widened. This is highly desirable as recognition execution can be reduced as to finding global minima takes intense computation.

Figure 4.6 is the plot for the second versus third LDA feature. It can be noticed that the classes are still well distanced while features of the same class closely clustered.

As the desirable class representation is attained, recognition stage can then be implemented.

#### 4.4 RBFNN RECOGNITION

Before training takes place, PCA and LDA were performed on the hand gesture images. These gestures were segmented from “*Purdue RVL-SLLL American Sign Language*

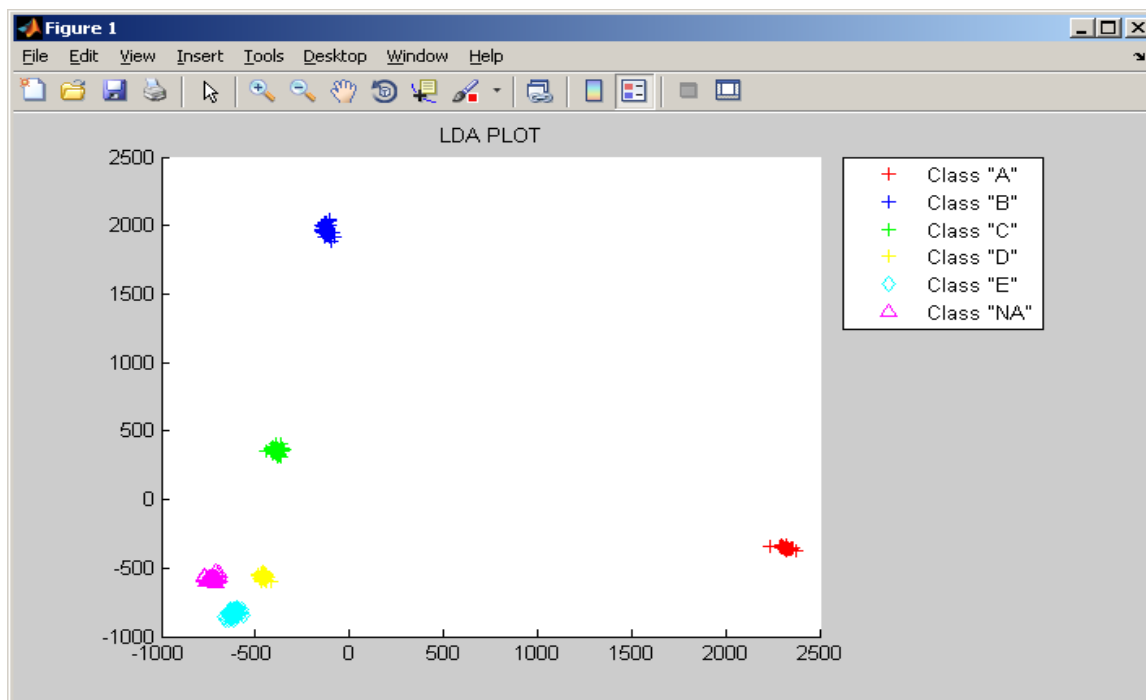


Figure 4.5. First two feature plot for LDA.

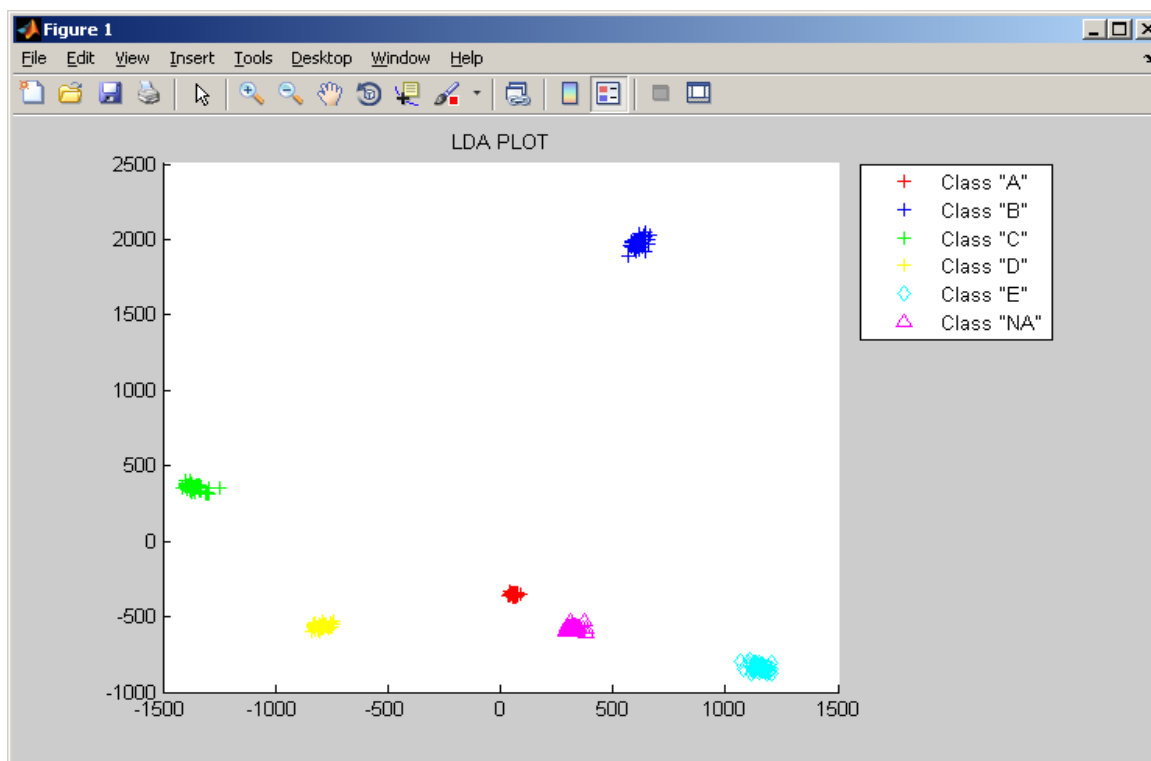


Figure 4.6. Second vs. third feature plot for LDA.

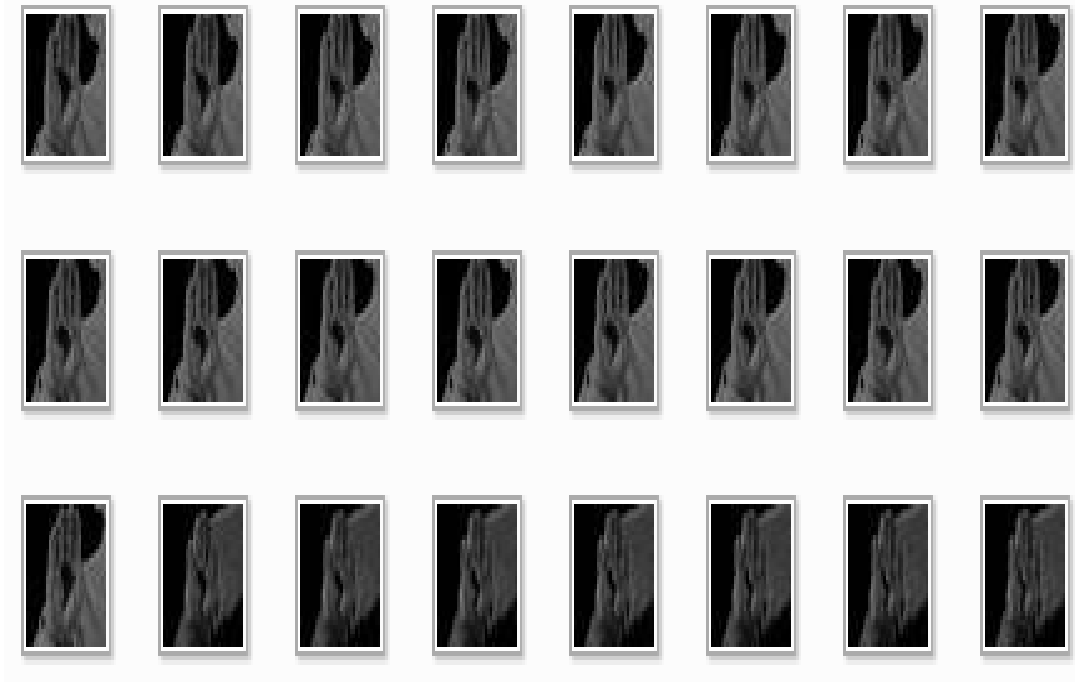
*Database*” [31] via skin detection algorithm. Figure 4.7 and Figure 4.8 below are examples on hand gesture images after the segmentation stage from skin detection. They are each 64x64 in dimension, which is in 4096-dimensional. PCA takes 70% of the first non-zero coefficient and LDA takes 5 features (6 classes -1). Thus, the features for hand gesture were reduced from 4096 to 5-dimensional spaces. This has drastically reduced the input data for RBFNN recognition.



**Figure 4.7. 64x64 in dimension hand images for alphabet “A.”**

In total, there are 50 training samples for each class (alphabet and face). For 6 classes, this would lead to 300 input data for RBFNN. As explain in RBFNN section, the activation function should be generalized to cater all classes. The spread of each activation function is half the minimum distance between any two closest clusters. In RBFNN training, the error goal should decrement for every cycle of iteration.

In this project, we are also trying to achieve the word prediction of the said theory, because theoretically, the nature of training will find the global minima where zero percentage rates are attained. However, as we all know that it is almost impossible to achieve

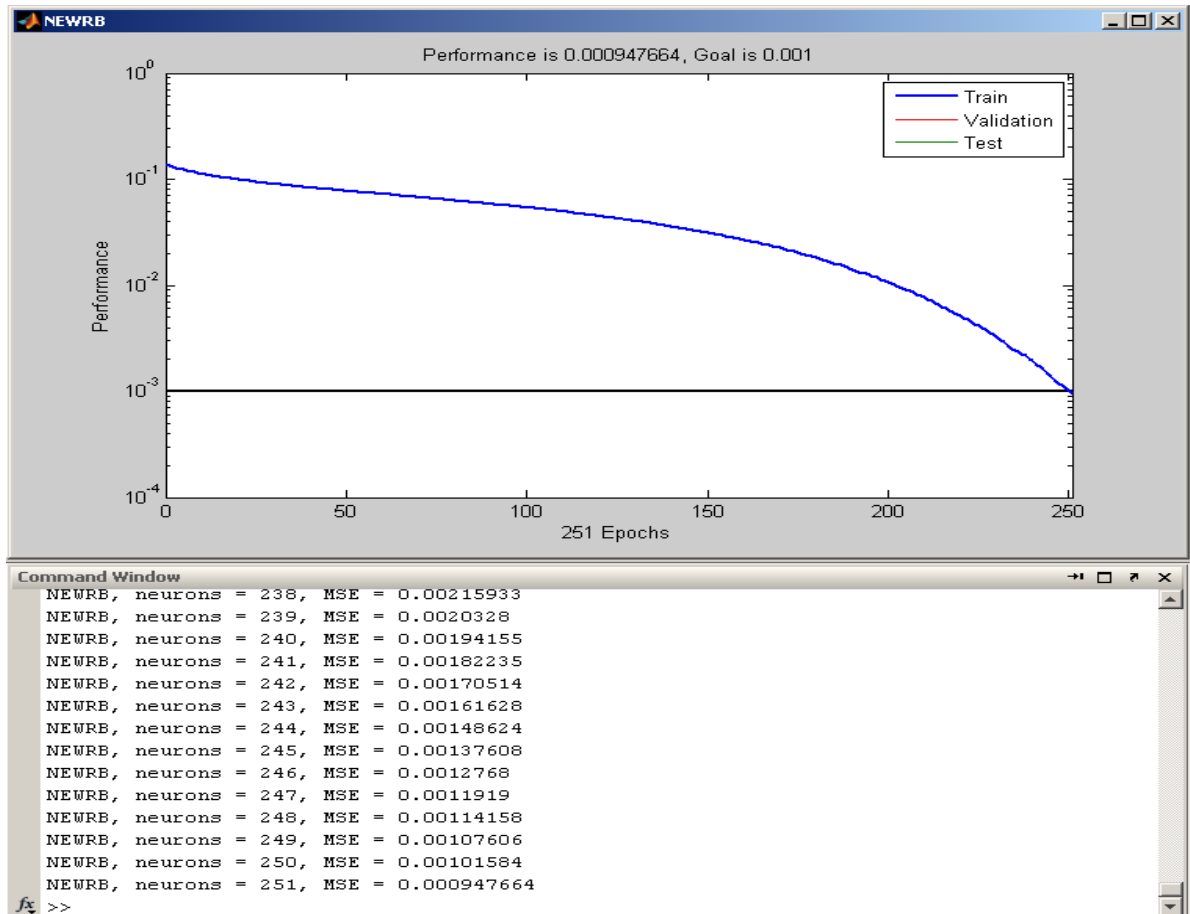


**Figure 4.8. 64x64 in dimension hand images for alphabet “B.”**

perfect results in this case, so thinking practically we have to our goal to 0.001%, which is approximately near zero error percentage recognition accuracy. Next Figure 4.9 is the plot of RBFNN training. It is observed that error percentage less than 0.001% is achieved.

Subsequently, the integration from skin detection, feature extraction, and dimension reduction to recognition is put to test. Figure 4.10 is the montage of the recognition steps, captured in frames elapse just several frames from each other.

An important feature of the approach presented here warrants some additional discussion. Each frame from the video source (5 per second) is actually matched against the six classes, among which is included the "Not Applicable" class. While some techniques for finding certain image events use marker events to determine when to start and stop matching frames for the known classes, the approach here does not do that. Here every single frame can be matched by the algorithm, but of course the frames between "stopping points", where the person is not moving and demonstrating a particular sign, contain image data derived from the actual movements. As such that image data is not useful. So first we note that this algorithm is effective enough to be used in situations where all frames matter; and second we note that in order to eliminate those which, due to the transitional movements cannot

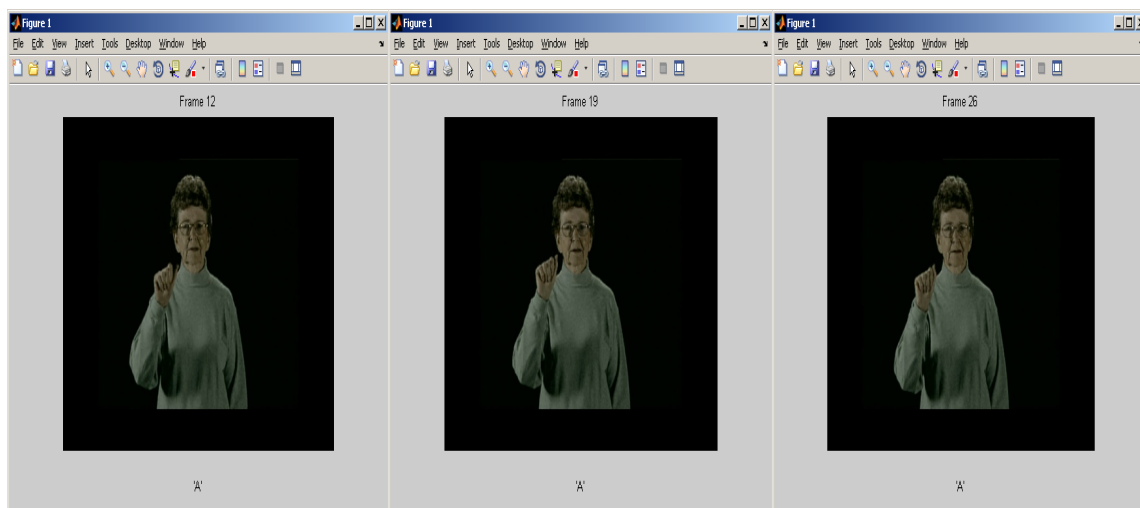


**Figure 4.9. The plot of RBFNN training with error rate less than 0.001%.**

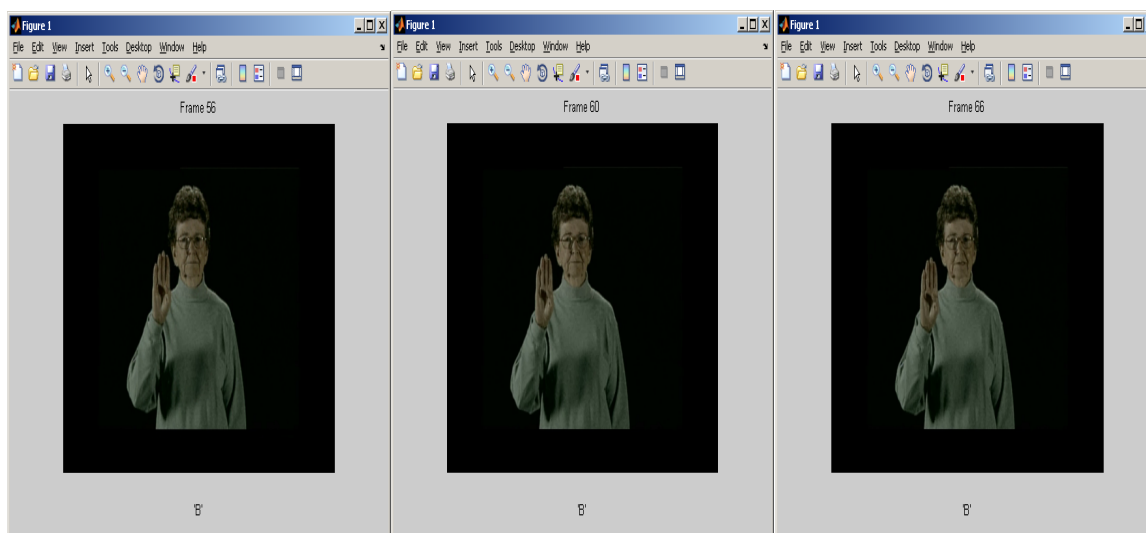
contribute to useful sign recognition, we actually "train" the algorithm to classify those in-between images as "Not Applicable."



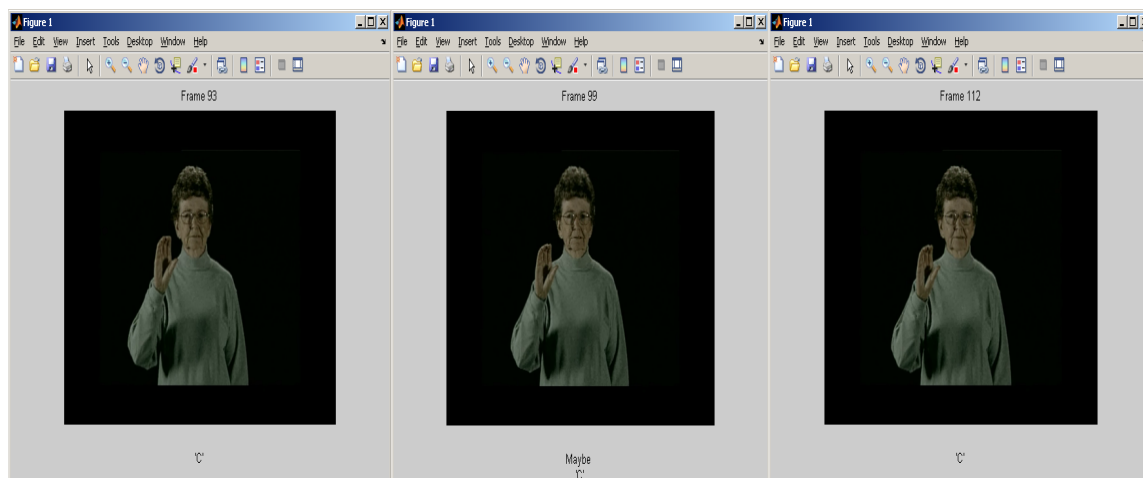
**Figure 4.10: Montage of frames recognizing alphabets. (A) Recognizing alphabet “A”. (B) Recognizing alphabet “B”. (C) Recognizing alphabet “C”. (D) Recognizing alphabet “D”. (E) Recognizing alphabet “E”.**



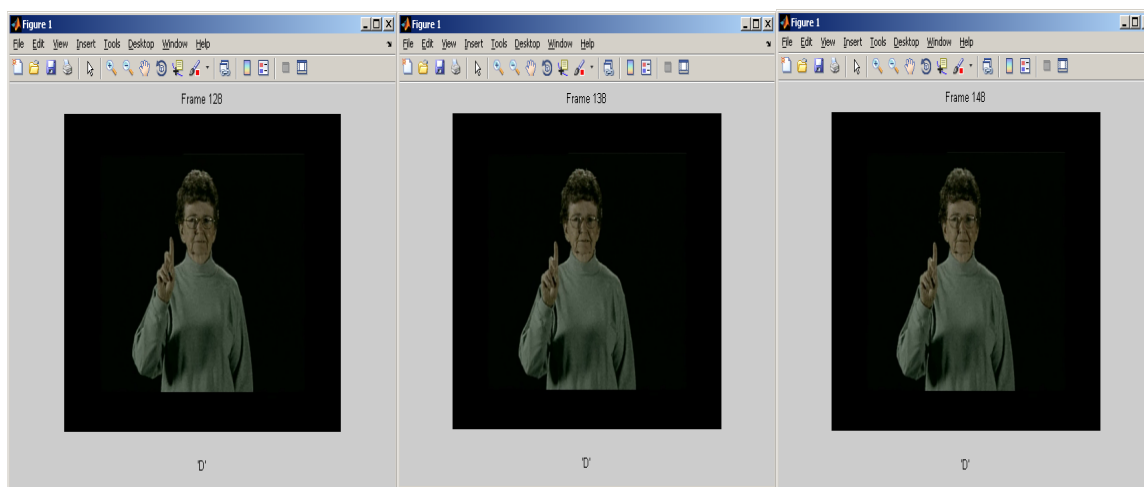
(A)



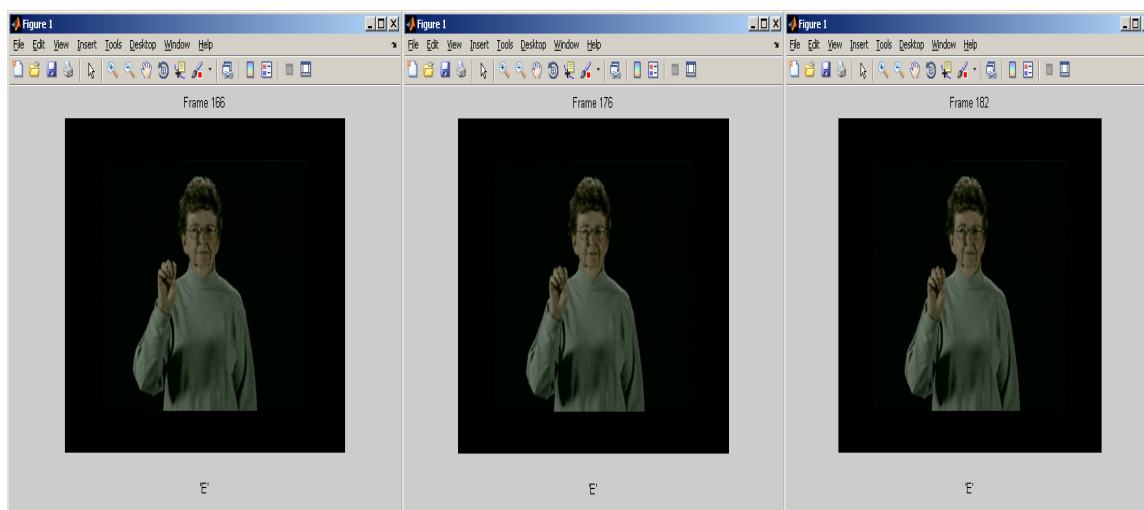
(B)



(C)



(D)



(E)

## **CHAPTER 5**

### **CONCLUSIONS AND FUTURE ENHANCEMENTS**

#### **5.1 CONCLUSIONS**

A system for American Sign Language gesture to voice recognition was presented. From the raw digital inputs of images or frames captured to image pre-filtering, skin detection algorithm is used. Pre-filtering term used here refers to the isolation of unnecessary background, which does not contribute to gesture recognitions. Skin detection algorithm was used in converting RGB to YCbCr color space with threshold set for skin pixel detection. Even though thresholding has the disadvantage of having fixed values for different skin colors, we are still going to use because it has proven successful for real world images. Further details in skin detection algorithm employed can be found in [32] [29].

Moving on, extracted hand gestures are represented by features. For this job, PCA is used. The unsupervised mean of feature extraction finds the most discriminant feature between images and uses these for data clustering. Images that exhibit the similarity in feature will group closer to each other. Following that, LDA further separates the cluster of one class from another, simultaneously reducing the data distance of the same group. These were all visually presented in Chapter 4. Feature extraction and dimension reduction are crucial in classification process as they are pre-filtering for data.

RBFNN is the classification tool used in this context. Given the extracted features of hand gestures, RBFNN serves to classify these data representations with internal logic of activation function. Data, which was closely clustered, has the activation function at highest point in the center while reducing activation function as the radius widens. Recognition in this system was done on five alphabets and another rejects class. In brief, the five alphabets refer to the alphabets “A,” “B,” to “E” whereas the reject class is subjects’ face, transition of hand gestures, which contributes nothing to recognition. In the chapter 4, the reorganization rate has been visually stated. Then, text-to-speech used is Microsoft Window’s built-in API with Microsoft Sam’s voice.

Recognition in this context was based on alphabets. The limitation is strongly related to the drawback in 2-D image-based recognition. As the real-world gesture recognition often takes in signs that relates to depth cue, the 2-D image is simply not suffice for recognition. Phrases that are related to the hand motion towards the image-capturing device cannot be translated in 2-D image, as the depth information is lost. This means the recognition performed on images is simply not capable in handling most of the hand gestures, which are motion-based. Solution to this limitation is the used of stereo-based vision system. This enables the motion vectors to be correctly calculated which could be then used for recognition. As the project was operated in 2-D image processing, migration to stereo-based recognition will be out of the context now.

## **5.2 FUTURE ENHANCEMENTS**

Further study and implementation can be done on our limitation. In future, the same approach can be taken, however implementation with stereo-based recognition will provide an extra ease in the implementation of this model in real world.

Issues such as finger spelling and spatial positioning aspects of ASL were ignored. In future this issue can be implemented which helps in integrating real world 3D Signs which broaden the area of research.

Currently the system ignores semantic context given by facial expressions while signing. By implementing expression-tracking techniques, this information might be recovered in the current system.

Having identified the downsides of thresholding, I believe this study provides a base for people to research on this in future.

In summary, the American Sign Language gesture to voice system is developed with all the supported results demonstrated. This project serves as a good foundation for future research work to be implemented. RBFNN training error ratio is only 0.0009 which proves that the implementation of the training method can be used to implement future enhancements. Despite the achieved recognition operation, enhancement is definitely possible.

## REFERENCES

- [1] Wikipedia. Sign Language, 2012.  
[http://en.wikipedia.org/wiki/Sign\\_language](http://en.wikipedia.org/wiki/Sign_language), accessed Feb. 2012.
- [2] P. Boyes Braem. *Einführung in die Gebärdensprache und ihre Erforschung (Introduction to sign language and its study)*. Signum, Hamburg, Ger., 1995.
- [3] S. Prillwitz, R. Leven, H. Zienert, T. Hanke, and J. Henning. *Hamburg Notation System for Sign Languages – An Introductory Guide*. Institute of German Sign Language and Communication of the Deaf, University of Hamburg, Hamburg, Ger., 1989.
- [4] B. Bauer and H. Hienz. Relevant features for video based continuous sign language recognition. In *FG '00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, page 440. IEEE Computer Society, Washington, D.C., 2000.
- [5] M. Assan and K. Grobel. Video-based sign language recognition using hidden markov models. In *Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, pages 97-109. Springer-Verlag, London, UK, 1998.
- [6] M. Al-Rousan, O. Al-Jarrah, and M. Al- Hammouri. Recognition of dynamic gestures in arabic sign language using two stages hierarchical scheme. *Intl. J. of Know. Int. Eng. Syst.*, 14:139-152, 2010.
- [7] T. E. Starner and A. Pentland. *Visual recognition of American Sign Language using Hidden Markov Models*. Perpetual Computing Section, Media Laboratory, MIT, Cambridge, Mass., 1995.
- [8] T. Starner , J. Weaver , and A. Pentland. A wearable computer Based American Sign Language recognizer. In V. O. Mittal, H. A. Yanco, J. Aronis, and R. Simpson, editors, *Lecture notes in artificial intelligence 1458. Assistive technology and artificial intelligence. Applications in robotics, user interfaces and natural language processing*. Springer-Verlag, Berlin, Ger., 1997.
- [9] G. Sperling, M. landy, Y. Cohen and M. Pavel. Intelligible encoding of ASL image sequences at extremely low information rates. *Comp. Vision, Graphics, and Image Proc.*, 31:335-391, 1985.
- [10] I. Essa, T. Darrell, and A. Pentland. Tracking facial motion. In *Proceedings of the 1994 Workshop on Motion of Non-rigid and Articulated Objects*, pages 36-42. Media Lab, Massachusetts Institute of Technology, Cambridge, Mass., 1994.
- [11] P. Maes, T. Darrell, B. Blumberg, and A. Pentland. The ALIVE system: Full-body interaction with animated autonomous agents. MIT Media Lab Perceptual Computing Group Technical Report 257, Massachusetts Institute of Technology, Cambridge, MA, 1994.

- [12] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. Technical Report 94-03, Mitsubishi Electric Research Labs, Cambridge, MA, 1994.
- [13] J. M. Rehg and T. Kanade. DigitEyes: Vision-based human hand tracking. School of Computer Science Technical Report CMU-CS-93-220, Carnegie Mellon University, Pittsburg, PA, 1993.
- [14] J. Lee and T. Kunii. Model-based analysis of hand posture. *IEEE Comput. Graph.*, 15(5):77-86, 1995.
- [15] T. Takashi and F. Kishino. Hand gesture coding based on experiments using a hand gesture interface device. *SIGCHI Bulletin*, 23(2):67-73, 1991.
- [16] D. H. Parish, G. Sperling, and M. S Landy. Intelligent temporal subsampling of American Sign Language using even boundaries. *J. of Exp. Psychol. Human*, 16(2): 282-294, 1990.
- [17] Wikipedia. File:Aslfingerspellalpha, 2011.  
<http://en.wikipedia.org/wiki/File:Aslfingerspellalpha.png>, accessed Apr. 2012.
- [18] Wikipedia. American Sign Langage, 2012.  
[http://en.wikipedia.org/wiki/American\\_Sign\\_Language](http://en.wikipedia.org/wiki/American_Sign_Language), accessed Jan. 2012.
- [19] B. Vicars. Basic Signs Pictures, n.d.  
<http://www.lifeprint.com/asl101/pages-layout/concepts.htm>, accessed Dec. 2011.
- [20] Wikipedia. File:HiddenMarkovModel.svg, 2012.  
<http://en.wikipedia.org/wiki/File:HiddenMarkovModel.svg>, accessed Apr. 2012.
- [21] O. C. Ibe. *Markov processes for stochastic modeling*. Academic Press, Amsterdam, Boston, MA, 2009.
- [22] F. Chen, C. Fu, and C. Huang. Hand gesture recognition using a real-time tracking method and hidden Markov models. *Image Vision Comput.*, 21:745-758, 2003.
- [23] H. Hienz, B. Bauer, and K. Kraiss. HMM-Based continuous sign language recognition using stochastic grammars. In A. Braffort, R. Gherbi, S. Gibet, J. Richardson, and D. Tiel, editors, *Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, pages 185-196. Springer, Gif-sur-Yvette, France, 2000.
- [24] Wikipedia. Artificial Neural Network, 2012.  
[http://en.wikipedia.org/wiki/Artificial\\_neural\\_network](http://en.wikipedia.org/wiki/Artificial_neural_network), accessed Feb. 2012.
- [25] K. Nakamura. Deaf Resource Library, 2008.  
<http://www.deaflibrary.org/asl.html>, accessed Jan. 2012.
- [26] K. Symeonidis. Hand Gesture recognition using neural networks. Master's thesis, University of Surrey, Guildford, Surrey, UK, 2000.
- [27] R. J. Howlett and L. C. Jain. *Radial basis function networks 2: New advances in design*. Physica-Verlag, Heidelberg, Ger., 2001.
- [28] M. S. Iraj, and A. Yavari. Skin color segmentation in fuzzy YCBCR color space with the Mamdani Inference. *Am. J. Sci. Res.*, 21:131-137, 2011.

- [29] J. A. M Basilio, G. A. Torres, G. S. Perez, and L. Karina. Explicit image detection using YCbCr space color model as skin detection. In *Proceedings of the 2011 American Conference on Applied Mathematics and the 5th WSEAS International Conference on Computer Engineering and Applications*, pages 123-128. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, 2011.
- [30] Wikipedia. Radial basis function network, 2012.  
[http://en.wikipedia.org/wiki/Radial\\_basis\\_function\\_network](http://en.wikipedia.org/wiki/Radial_basis_function_network), accessed Mar. 2012.
- [31] R. B. Wilbur, and A. C. Kak, Purdue RVL-SLLL American Sign Language Database, School of Electrical and Computer Engineering Technical Report TR-06-12, Purdue University, W. Lafayette, IN, 2006.
- [32] D. Chai, and K. N. Ngan. Face segmentation using skin-color map in videophone applications. *IEEE Trans. on Circuits Syst. Video Technol*, 9:551-564, 1999.



**APPENDIX**

**GLOSSARY**

**ASL:** American Sign Language  
**GSL:** German Sign Language  
**CSL:** Chinese Sign Language  
**AUSLAN:** Australian Sign Language  
**ArSL:** Arabic Sign Language  
**3D:** Three Dimensions  
**2D:** Two Dimensions  
**RBFNN:** Radial Basis Function Neural Network  
**FPS:** Frames Per Second  
**TTS:** Text To Speech  
**ANN:** Artificial Neural Network  
**HMM:** Hidden Markov Model  
**LSF:** Old French Sign Language  
**BSL:** British Sign Language  
**NN:** Neural Network  
**RBF:** Radial Basis function  
**LDA:** Linear Discriminant Analysis  
**PCA:** Principal Component Analysis  
**RGB:** Red Green Blue  
**YCbCr:** Luminance Chrome blue Chrome red  
**SAPI:** Speech Application Programming Interface  
**MATLAB:** Matrix Laboratory