5-2015

Visual Speech Recognition using Histogram of Oriented Displacements

Sujeeth Selvam Kumaravel
Clemson University

Visual Speech Recognition using Histogram of Oriented Displacements

---

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Electrical Engineering

---

by
Sujeeth Selvam Kumaravel
May 2015

Accepted by:
Dr. John Gowdy, Committee Chair
Dr. Robert Schalkoff
Dr. Carl Baum

Abstract

Lip reading is the recognition of spoken words from the visual information of lips. It has been of considerable interest in the Computer Vision and Speech Recognition communities to automate this process using computer algorithms. In this thesis, we have developed a novel method involving describing visual features using fixed length descriptors called Histogram of Oriented Displacements to which we apply Support Vector Machines for recognition of spoken words. Using this method on the CUAVE database we have achieved a recognition rate of 81%.

Dedication

I dedicate this research effort to my parents, whose support helped me go through grad school. My words cannot describe my appreciation for my father Kumaravel Subbiah and my mother Shanthi Kumaravel for their love, support and endless care about my studies and life.

Acknowledgments

It is a pleasure to convey my sincere thanks in this humble acknowledgement to all who supported me. In the first place I would like to express my sincere appreciation and gratitude to Dr. John Gowdy for his supervision, advice and guidance from the very early stages of this research. It was a wonderful experience working with you. I would like to thank Dr. Robert Schalkoff for his advice and encouragement for this research. I would like to thank Dr. Carl Baum for his advice and encouragement for this research. I am grateful to these professors in every way and would like to continue our collaboration in the future. I sincerely thank Creed Johnson for his contribution to this research. Many thanks to my lab mate Shamama Afnan at the Speech Processing Laboratory for sharing her research experience with me and for her wonderful friendship. I am grateful to the Faculty of Holcombe Department of Electrical and Computer Engineering for supporting my research with excellent facilities.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Lip reading is a technique used to understand or interpret speech with only the visual signal of mouth movement and without the acoustic signal. It is a technique mastered by people with hearing disabilities. The ability to lip read allows a person with hearing impairments to communicate with others and engage in social activities which otherwise would be difficult. Automating the process of lip reading has applications in human computer interaction (HCI). Recent advances in the fields of computer vision, pattern recognition and signal processing have led to a growing interest in automating this challenging task of lip reading. Automating the ability to lip read, a process referred to as visual speech recognition (VSR) or speech reading, could open the door for other novel related applications.

VSR has received a great deal of attention in the last decade for its potential use in applications such as human-computer interaction (HCI), audio visual speech recognition (AVSR), speaker recognition, sign language recognition and video surveillance. Its main aim is to recognize spoken words by using only the visual signal that is produced during speech production. Hence VSR deals with the visual domain of speech and involves image processing, object detection, pattern recognition, artificial intelligence, statistical modeling etc.

There are two main approaches to the VSR problem, each with its own strengths and weaknesses:

- visemic approach
- holistic approach.

1.1 Visemic Approach

The traditional and most common approaches to VSR are based on visemes. A viseme is one of a sequence of mouth dynamics (mouth shapes and movements) in the visual domain that are required to generate a phoneme. Hence a viseme represents a part of the word. Several problems arise in using visemes in VSR, such as low number of visemes (between 10 and 14), compared to the number of phonemes (between 45 and 53). Visemes cover only a small subspace of the mouth dynamics in the visual domain. These problems contribute to the poor performance of the traditional VSR systems. Hence using the visemic approach is like quantizing a signal which results in loss of information.

1.2 Holistic Approach

The holistic approach considers the signature of a whole word rather than only a part of it like the visemic approach does. This approach is a good alternative to the visemic approach for automatic lip-reading. The major problem of this approach is that for a complete English language lip reading system, we need to train all the English language words in the dictionary. But it can be effective if trained on a specific domain of words, e.g., numbers, zip codes, cities.

Speech perception is a multimodal process and involves information from more than one sensory modality. The McGurk effect [1] shows that visual articulatory information is integrated into our perception of speech automatically and unconsciously. For example, a visual /ga/ combined with an auditory /ba/ is often heard as /da/. This effect is shown to be very robust and knowledge about it seems to have very little effect on one′s perception of it.

Interest in machine lip reading began to emerge in the mid 1980s when it was shown that visual lip information extracted from a speaker′s lips can enhance the performance of an automatic speech recognition system, especially in a noisy environment. Petajan was the first to investigate the problem of machine lip reading or visual speech recognition [2]. Recently, it has also been shown that the dynamics of the speaker′s lip during speech articulation provides useful biometric information for speaker recognition.

Visual speech recognition involves three steps:

- Lip segmentation

- Feature extraction

- Classifier design.

Although significant research effort and many technological advances have been made recently, visual speech recognition is still far from practical deployment. Unlike the relatively mature field of automatic speech recognition, there are still many unsolved theoretical and algorithmic issues in visual speech recognition. For example, the problems of lighting, shadow, pose, facial hair, camera resolution and so forth make reliable segmentation and extraction of lip feature a difficult task. The problem is further compounded by the difficult and variable environments these recognition systems tend to operate in. There is also little theoretical study on the amount of phonetic/linguistic information that can be extracted from the speaker′s lips for speech recognition. Various aspects of this research area include lip segmentation from video sequence, lip feature extraction and modeling, feature fusion and classifier design for visual speech recognition and speaker verification.

Audio-visual speech recognition (AVSR) is concerned with recognizing speech using both audio signal and the visual signal of mouth movement. When there is a high amount of acoustic noise present in the recognition environment, audio only speech recognition may produce poor results. It has been found that including the visual signal of lip movements in speech recognition improves the recognition accuracy. But it is also of interest to determine what kind of accuracy can be obtained with visual-only speech recognition. This kind of a system can be useful when there is no audio available in the recognition scene or if the audio signal is very highly degraded. In this thesis, we work only with the visual signal in order to determine the recognition accuracy.

Chapter 2

Background

The focus of most audio visual speech recognition systems is to find effective ways of combining video with existing audio-only speech recognition (ASR) systems[3]. However, in some cases it is difficult to extract useful information from audio. An example is a voice controlled car stereo system. The user has to be able to play, pause, switch stations and tracks using simple commands. This allows the driver to keep his attention on the road and hands on the wheel. In this case, the voice is corrupted by the car engine and traffic noise and also by the car stereo audio itself, so almost all useful speech information is in the video. However, only a few authors have focused on the problem of visual-only speech recognition as a stand alone problem. In this chapter, we will review the work performed by some authors in visual speech recognition.

2.1 Previous Work

2.1.1 Visual Speech Recognition Using Support Vector Machines

In [4], Gordan et al., proposed a visual speech recognition network based on Support Vector Machines (SVM) [5, 6]. Each word was described as a temporal sequence of visemes [7]. Each viseme was described by a support vector machine, and the temporal character of speech was modeled by integrating the support vector machines as nodes into a Viterbi decoding lattice [8].

The basic visual models were viseme models, and the visual word models were obtained by a combination of the basic models into a temporal dynamic sequence. Speech was modeled as

a temporal sequence of symbols corresponding to the different phones produced and SVMs were employed as nodes in a Viterbi lattice. The nodes of a Viterbi lattice are supposed to generate posterior probabilities of the corresponding symbols to be emitted [8]. The output of an SVM was converted into a posterior probability by a sigmoidal mapping as

$$P(y = +1|f(\mathbf{x})) = \frac{1}{1+exp(a_1 f(\mathbf{x})+a_2)}$$

where $f(\mathbf{x})$ is the hyperplane equation of an SVM. a_1 and a_2 can be derived from the training set using maximum likelihood estimation[9].

A network of parallel SVMs was built where each SVM was trained to classify patterns in a particular class. The pattern $\mathbf{x}_k$ is assigned to class C_l according to a maximum a posteriori rule:

$$P(y_l = 1|f_l(\mathbf{x}_k)) = \arg\max_j P(y_j = 1|f_j(\mathbf{x}_k))$$

where the probabilities were given by SVM outputs.

In the temporal domain the word models were represented starting from the visemic model and from the total number of frames T in the word pronunciation, by assuming that the duration of each viseme in the word pronunciation is variable, but necessarily not zero. A temporal network of models corresponding to the different durations of visemes was represented by the Viterbi algorithm [8] containing as many states as the number of frames T in a word. Each word in the vocabulary was represented by a Viterbi lattice. Such a Viterbi lattice for the word one is shown in Figure 2.4. Each node in the lattice is the probability that the corresponding symbol o_k is emitted at time instant k. This probability was denoted $b_{o_k k}$ and was generated by the corresponding SVM. $a_{o_k o_{k+1}}$ denotes the transition probability from the state that generates o_k to the state that generates o_{k+1}.

The probability that a word w_d, d=1 to D, where D is the number of words in the vocabulary, was produced following a path l in the lattice was calculated as

$$p_{d,l} = \prod_{k=1}^{T} b_{o_k k} \prod_{k=1}^{T-1} a_{o_k o_{k+1}|d,l}$$

and the probability the word w_d was produced was taken to be the maximum of the $p_{d,l}$. The maximum $p_{d,l}$ value is denoted as p_d. The most probable word, that is whose probability p_d is maximum is recognized.

This system was evaluated on a task of recognizing the first 4 English digits spoken by 12 speakers from the Tulips1 [10] database. 12 visemes were used in the recognition and SVMs with polynomial kernel of degree 3 were used. The rectangular region of interest around the mouth is taken and downsampled into a 16x16 image. The 256 gray level values were taken and along with their 256 temporal derivatives, each mouth image was represented by a feature vector of length 512.

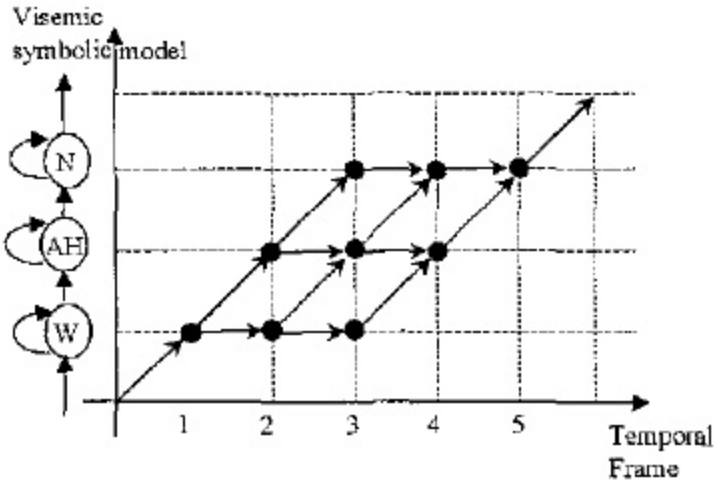

Figure 2.1: SVM Viterbi Lattice [4]

A recognition rate of **90.6%** was obtained, but this paper performed the recognition of only 4 digits. But it proved the suitability of SVMs for visual speech recognition.

2.1.2 Visual Speech Recognition With Loosely Synchronized Feature Streams

In [11], Saenko et al. presented an approach to detecting and recognizing spoken isolated phrases using solely the visual input. They adopted an architecture that first employs discriminative detection of visual speech and articulatory features and then performs recognition using a model that accounts for the loose synchronization of the feature streams. Discriminative classifiers detect the subclass of lip appearance corresponding to the presence of speech and further decompose it into features that correspond to the physical components of articulatory production. These components evolve in a semi-independent fashion, and conventional viseme based approaches fail to capture the resulting co-articulation effects. This paper presented a Dynamic Bayesian Network (DBN) with multi-stream structure and observations consisting of articulatory feature classifier scores which can model varying degrees of co-articulation. It evaluated a visual-only recognition system on a command utterance task and showed comparative results on lip detection and speech/nonspeech classification as well as recognition performance against a baseline system.

The articulators (e.g., lips and tongue) can evolve asynchronously from each other, especially in spontaneous speech, producing varying degrees of co-articulation. Existing systems treat speech

Figure 2.2: Bilabial Closure During Production of Words "romantic" and "academic" [11]

as a sequence of atomic viseme units and so they require many context dependent visemes to deal with co-articulation[12]. In this paper, the authors have modeled the multiple underlying physical components of human speech production or Articulatory Features (AF)[13]. The varying degrees of asynchrony between the AF trajectories are represented using a multi-stream model.

In this paper, the authors describe an end-to-end visual-only approach to detecting and recognizing spoken phrases, including visual only detection of spoken phrases. They have used articulatory features (AF) as an alternative to visemes and a DBN [14, 15] for recognition with multiple loosely synchronized feature streams. The observations of the DBN are the outputs of the discriminative AF classifiers. The authors evaluated their approach on a dataset containing 20 car-stereo control commands.

The appearance of the mouth can be heavily influenced by the asynchrony between articulatory gestures. This occurs when the articulatory features not involved in the production of the current sound evolve asynchronously. The Figure 2.1 shows an example of such de-synchronization in two snapshots taken at the moment of complete lip-closure during pronunication of *romantic* and *academic*.

Suppose the phoneme /m/ is to be modeled in these two different utterances as a single context-independent viseme. Both images would be considered to belong to a single class (the bilabial viseme) and to have the same open/closed feature value (fully closed). But their appearance is different because in the second context the mouth is 25% wider. There is also contextual variation because, in *romantic* the lip rounding of /ow/ lingers during the lip closure. So modeling lip rounding and lip opening as two separate articulatory features would capture more information that just modeling the /m/ viseme. Allowing the features to proceed through their trajectories asynchronously would account for these types of effects. An alternative way to model such variability

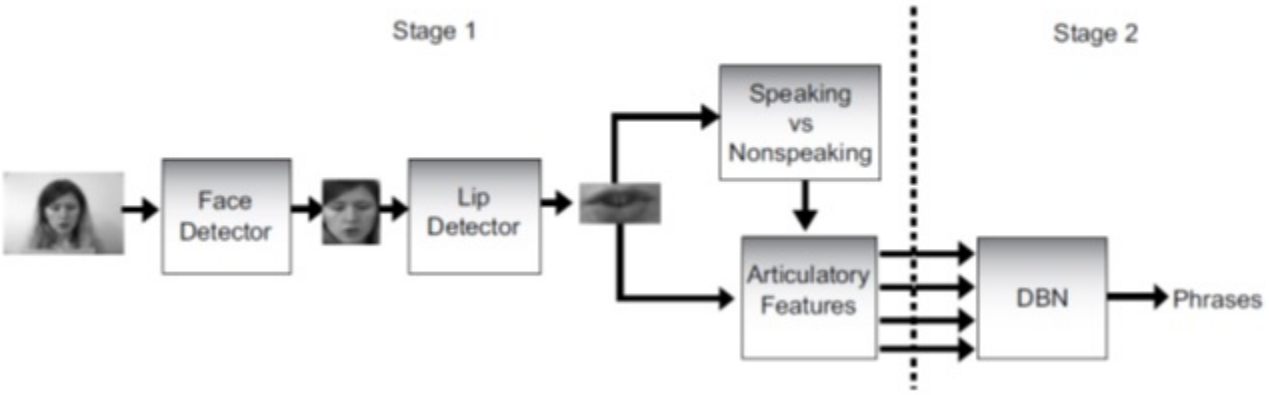

Figure 2.3: System Block Diagram[11]

is to use context dependent units. However, visual coarticulation effects can span three or more visemes, requiring a large number of context dependent models.

The system consisted of two stages as shown in Figure 2.2. The first stage is a cascade of discriminative classifiers that first detects speaking lips in the video sequence and then recognizes components of lip appearance corresponding to the underlying articulatory processes. This stage of the system extracts articulatory features from the input video sequence. In visual modality only visual articulators can be modeled. A restricted articulatory feature set corresponding to the configuration of lips was chosen. The features used were Lip Opening (LO), discretized into closed, narrow, medium and wide states; Lip Rounding (LR), discretized into rounded and unrounded states; and Labio-Dental (LD) which is a binary feature indicating whether the lower lip is touching the upper teeth such as to produce /f/ or /v/. Other articulatory features that are distinguishable from the video such as the tongue movements were not incorporated in this paper. The AF detection stage is implemented as a cascade of discriminative classifiers each of which uses the result of the previous classifier to narrow down the search space. Support Vector Machines (SVMs) are used as the discriminative classifiers. In the cascade, the first classifier detects the presence and location of the face in the image. If a face was detected, the second classifier searches the lower region of the face for lips. Once the lips have been located, they are classified as either speaking or non-speaking. This is accomplished in two steps. First, motion is detected and then a speaking lip classifier is applied to determine whether the lips are moving due to speech or some other activity.

The final set of classifiers decompose the detected speech into several articulatory features. To evaluate the lip detector, a subset of the AVTIMIT dataset[16] was used with the videos of 20 speakers speaking English sentences. The authors collected their own dataset consisting of videos of 3 speakers speaking similar sentences (*speech* dataset) and the videos of the same 3 speakers with videos of them making non-speech movements (*nonspeech* dataset). The *speech* and *nonspeech* datasets were used to train and test the speaking-lip classifier. The *speech* dataset was used to train the AF classifiers and viseme classifier. The SVM lip detector detected lips in 99% of the frames. Normalized image difference energy was calculated over consecutive frames and then low pass filtered over a one-second window with a 2 Hz cutoff frequency. To the filtered output, a threshold is applied to determine whether the lips are moving. For the frames which were classified as containing moving lips, an SVM classifier is used to determine if the movement corresponds to speech activity. Its output was median filtered using a half-second window to remove outliers. This classifier achieved 98.2% detection rate of speaking lips. SVM classifiers were used for the three articulatory features. A viseme SVM classifier was also used as a baseline. One-vs-all multiclass SVM formulation was used. So 6 SVM classifiers were trained: 4 for LO; one for LR and one for LD. One SVM for each of 6 visemes was also trained. The input vectors to the SVMs were produced by first resizing the lip image into 32 by 16 pixels. A discrete cosine transform (DCT) is then applied to the image and 512 coefficients are obtained. PCA transform is applied to reduce the dimensionality of the vector to 75 components. Radial basis function (RBF) kernels were used for SVMs.

The second stage is a DBN that recognizes the phrase while explicitly modeling the possible asynchrony between these articulatory features. This stage of the system is a short phrase recognizer that models the visual speech in terms of underlying articulatory processes. The recognizer uses a Dynamic Bayesian Network with a multi-stream structure and observations consisting of the AF classifier outputs from the previous stage. The model is implemented as a Dynamic Bayesian Network due to the semi-independent evolution of the AF streams. Figure 2.3 shows three frames of the DBN.

The model consists of three parallel HMMs, one per AF, where the joint evolution of the HMM states is constrained by the synchrony requirements imposed by the variable c^1 and c^2 as seen in Figure 2.4. The following figure shows a conventional single stream viseme HMM which is used as a baseline system for comparison.

The model allows the AFs to proceed through their trajectories at different rates. This

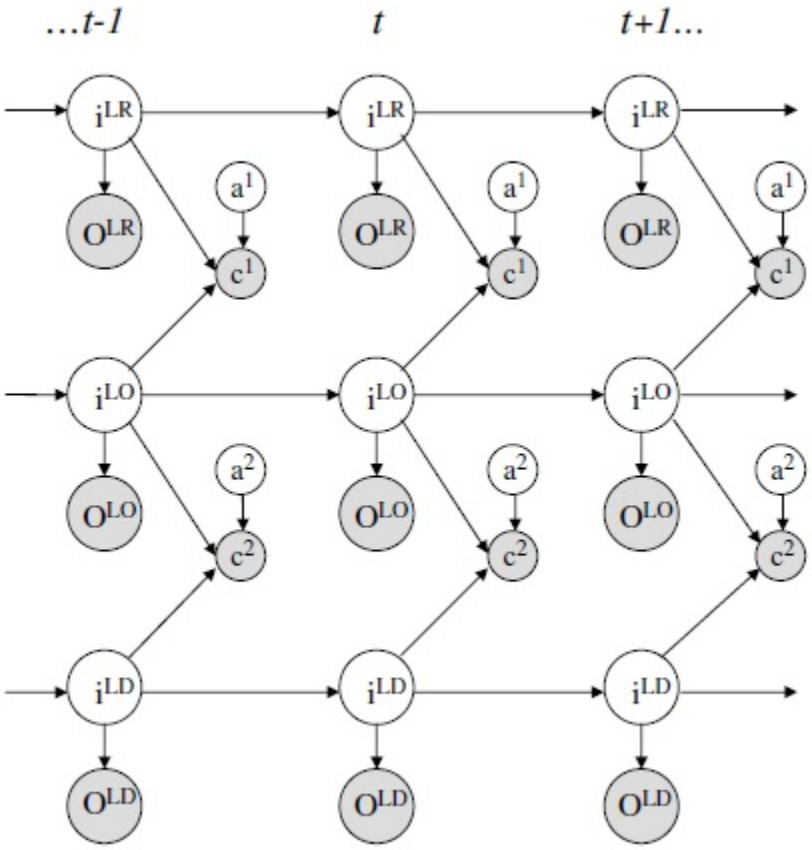

Figure 2.4: DBN for Articulatory Feature Based VSR[11]

asynchrony is not completely unconstrained. Sets of trajectories that are more synchronous are more probable than less synchronous ones and a hard constraint on the maximum degree of asynchrony is imposed. i_t^F is the index into the state sequence of feature stream F at time t. If stream F is in the n^{th} state of a given word at time t, then $i_t^F = n$. The degree of asynchrony between two feature streams F_1 and F_2 at time t is defined as $|i_t^{F_1} - i_t^{F_2}|$. The probabilities of varying degrees of asynchrony are given by the distributions of the a^j variables. Each c_t^j variable checks that the degree of asynchrony between its parent feature streams is a_t^j. This is done by having the c_t^j variable always being observed with value 1 with distribution

$$P(c_t^j = 1 | a_t^j, i_t^{F_1}, i_t^{F_2}) = 1$$
$$if \quad and \quad only \quad if \quad |i_t^{F_1} - i_t^{F_2}| = a_t^j$$

and 0 otherwise, where $i_t^{F_1}$ and $i_t^{F_2}$ are the indices of the feature streams c_t^j. For example, for c_t^1, $F_1 = LR$ and $F_2 = LO$.

For each stream the observations O^F are the SVM margins for that feature and the observation model is a Gaussian mixture. Whole word models are used. A separate DBN for each phrase in the vocabulary is trained with i^F ranging from 1 to the maximum number of states in the word. Recognition corresponds to the phrase whose DBN has the highest Viterbi score. To perform recognition with the model, standard DBN inference algorithms [17] were used. The observation models, the per-feature state transition probabilities and the probabilities of asynchrony between streams are learned via maximum likelihood using the Expectation-Maximization algorithm[17].

The DBN component of the system was evaluated on the recognizing isolated phrases. A set of 20 commands that could be used to control an in-car stereo system was chosen. The videos of 2 speakers speaking these commands was collected (*commands* dataset).

The six decision values of the SVMs were used as the observations. The AF based DBN was evaluated with some asynchrony between the feature streams. LR and LO streams were allowed to de-synchronize by upto one index value (one state) as are LO and LD streams. The two asynchrony probabilities $p(a^1 = 1)$ and $p(a^2 = 1)$ are learned from the training data. This model uses whole-word units and Gaussian mixture models (GMMs) of observations (single Gaussians with tied diagonal covariance matrices). This resulted in a visual speech recognition rate of **65.8%**. The results showed that the approach was better at accounting for variation in speech that is faster than the speech used in training.

2.1.3 Visual Speech Recognition Using DCT features

In [18], Hong et al., focused on dimensionality reduction strategies for DCT based features [19, 20, 21, 22, 23] for VSR. PCA [24, 25, 19, 20, 26, 27] was applied to extract DCT coefficients. This combination utilized the advantages of these two transforms. DCT was used to differentiate frequencies while PCA was used to select the most important components in the DCT feature vector.

The ROI containing the mouth area was downsampled to a 32x16 image. The image transform 8x8 block-based DCT was applied to this image. This method divided the 32x16 image into 8 non-overlapping blocks of 8x8 and applied the DCT transform to each of the blocks. This gave a 512 dimensional vector as output. The dimensionality of this vector was reduced to 80 by using PCA. The resulting vector was normalized and used as a feature vector for classification using Semi-Continuous Hidden Markov Model (SCHMM) [28] which was set with 6 states and 8 modes per state. The system was tested using the HIT-BiCAVDB database containing speech videos from 10 speakers. Speaker Dependent recognition tests were conducted. A visual speech recognition rate of **68.8%** was obtained.

2.1.4 GA-based Informative Feature for Visual Speech Recognition

In [29], Ukai et al. proposed a feature called GA-based Informative Feature (GIF) and applied it to Visual Speech Recognition. The feature extraction method consisted of two transforms which converted an input vector to GIF. The two transformation matrices were obtained using Genetic Algorithm (GA) and the training data.

The two transformation matrices, which were denoted A and B, were computed as explained in the following. Let there be C classes and the training set be denoted $R = \{\mathbf{r}_n\}$. For an i^{th} class, for an input vector $\mathbf{x}$, the following linear classifier was assumed:

$$f(\mathbf{x}; \mathbf{a}_i) = (\sum_{j=1}^{N} a_{i,j} x_j) + a_{i,N+1}$$

where $\mathbf{a}_i = (a_{i,1}, ..., a_{i,N}, a_{i,N+1})$ are the classifier parameters and a part of the matrix A. These classifier parameters were calculated using a GA as explained in the following. An initial population G_0 with K individuals was created with each individual having $N + 1$ chromosomes each of which encoded a classifier parameter. For the k^{th} individual $\mathbf{v}_k$ in the h^{th} generation G_h, a fitness function $E(\mathbf{v}_k)$ was calculated as follows:

$$E(\mathbf{v}_k) = \sum_{n=1}^{\|R\|} l_n . sgn(f_i(\mathbf{r}_n; \mathbf{a}))$$

where $\mathbf{a}$ was a parameter set obtained by decoding $\mathbf{v}_k$ and l_n was a transcribed label that equaled 1 if $\mathbf{r}_n$ belonged to a class, or -1 otherwise. Conventional GA operations were applied to form a next generation G_{h+1}; elitist selection and inheritance were applied to copy a certain individual to G_{h+1}; for genetic diversity, mutation and crossover operations were also conducted to generate a new individual. These operations were repeated for $h = 0$ to $h = F - 1$ and a final population G_F was obtained. From G_F, the individuals having the K/I highest fitness values were extracted and added to a candidate population G_C. By repeating these steps I times, the selection of candidate population G_C was completed. Now G_C was taken as the initial generation and all the above steps were repeated and the best fit individual $\mathbf{v}$ was obtained. The transformation parameter set $\mathbf{a}$ was subsequently obtained by decoding $\mathbf{v}$. These steps enabled the computation of the matrix A.

An input vector $\mathbf{x}$ was converted into an intermediate vector $\mathbf{y}$ using matrix A as follows:

$$\mathbf{y} = A(\mathbf{x}^T 1)^T.$$

For the i^{th} class, a mean vector μ_i, was calculated as

$$\mu_i = 1/\|R_i\| \sum_{\mathbf{r} \in R_i} A(\mathbf{r}^T 1)^T$$

where R_i was a subset of the training set in which all vectors belonged to i^{th} class. For the vector $\mathbf{y}$, a linear transformation g was defined as:

$$g(\mathbf{y}; \mathbf{b}_m) = \sum_{j=1}^{C} b_{m,j} y_j$$

where $\mathbf{b}_m = (b_{m,1}, ...b_{m,C})$ indicated classifier parameters and a part of matrix B. For m=1, the parameter set $\mathbf{b}_1$ is optimized by applying GA with fitness function modified as:

$$E(\mathbf{v}_k) = var(w_1, ..., w_C)$$

where $w_i = g(\mu_i; \mathbf{b})$. Here $\mathbf{b}$ is obtained by decoding $\mathbf{v}_k$. For $m = 2$, $\mathbf{b}_2$ is optimized so as to maximize a variance just as same as $\mathbf{b}_1$ under the constraint that the inner product between $\mathbf{b}_1^T$ and $\mathbf{b}_2^T$ is zero. Similarly, for any m, the m^{th} parameter set $\mathbf{b}_m$ was calculated in the same way. These steps enabled the computation of the matrix B.

The intermediate vector $\mathbf{y}$ is further converted into a vector $\mathbf{z}$ as:

$$\mathbf{z} = B\mathbf{y}.$$

This vector $\mathbf{z}$ was the proposed feature GIF.

The authors used the CENSREC-1-AV database [30] for conducting lipreading experiments. The database contained digit utterances. 59x35 mouth ROI images were obtained by lip detection techniques. These detected lip images were then resized to 29x17. A 493 dimensional input vector was obtained by enumerating the intensity values in the image from left-top to right-bottom. By

applying the above two transformation matrices, the 10-dimensional GIF output vector was obtained. To the 10-dimensional GIF vectors, first-order and second-order derivatives were added making the feature vectors 30 dimensional. Hidden Markov Model (HMM) was used for recognition experiments. A visual HMM was built using the Baum-Welch training algorithm, and recognition results were obtained using Viterbi algorithm. HMMs were built for all the digits. Each digit HMM consisted of 16 states having 8 Gaussian mixtures. In these tests, a visual speech recognition rate of **59.79%** was obtained.

2.1.5 Visual Speech Recognition Using Zernike Moments

In [31], Borde et al. computed visual features using Zernike moments. These features were normalized and their dimensions were reduced using PCA. Viola Jones face detection algorithm with AdaBoost was used to detect face in images. Subsequently, mouth ROI was extracted from these detected face images. These mouth ROIs were preprocessed. The preprocessing steps included conversion to gray scale and filtering. From the filtered image, a gray scale threshold was computed and the mouth ROI was converted into a binary image. This enabled getting the actual ROI which contained only the lip region.

From these processed ROIs, Zernike moments, which describe shape information, were computed. These moments are the mapping of an image onto a set of complex Zernike polynomials which form a complete orthogonal set on the unit disk with $x^2 + y^2 = 1$.

$$Z_{mn} = \frac{m+1}{\pi} \int_x \int_y I(x,y)[V_{mn}(x,y)]dxdy$$

where m is the order of the Zernike polynomial, n is the angular dependency, $I(x,y)$ is the image. The Zernike polynomials $V_{mn}(x,y)$ are expected in polar coordinates using radial polynomial (R_{mn}) as per the following equations:

$$V_{mn}(r,\theta) = R_{mn}(r)e^{-jn\theta}$$

$$R_{mn}(r) = \sum_{s=0}^{\frac{m-|n|}{2}} (-1)^s \frac{(m-s)!}{s!(\frac{m+|n|}{2}-s)!(\frac{m-|n|}{2}-s)!} r^{m-2s}.$$

For each frame, Zernike moments of upto order 9 were calculated. So there were 9 feature values for each frame. Each word in the database contained 52 frames in the corresponding video. So for one word, a 468 dimensional feature vector was computed. The dimensionality of this feature vector was reduced by using PCA. The authors used the vVISWa database consisting of video

sequences of isolated utterances of 12 city names spoken by 10 speakers to test their method of visual speech recognition. Each speaker spoke each word 10 times. Euclidean distance classifier was used for classification. A recognition rate of **63.88%** was obtained.

2.1.6 Visual Speech Recognition Using kNNR

The most important paper that has been used as the main reference in this thesis is [32]. In this work, the mouth ROI is detected first. 8 features were extracted from the ROI in each frame of the input video. The feature extraction process is explained in the following.

The rectangular ROIs height and width were taken as two features. The mouth ROI was transformed into the frequency domain using Discrete Wavelet Transform (DWT). This results in four wavelet subbands LL_i, LH_i, HL_i, HH_i for each ROI. The mutual information between two consecutive ROIs were defined as follows.

$$M(X;Y) = \sum_x \sum_y p(x,y) log(\frac{p(x,y)}{p(x)p(y)})$$

where

X = mouth ROI in the current frame

Y = mouth ROI in the previous frame

$p(x,y)$ = joint probability mass function (pmf) of X and Y

$p(x)$ = marginal pmf of X

$p(y)$ = marginal pmf of Y.

Instead of calculating the mutual information in the spatial domain, it is calculated in the frequency domain in each of the four subbands and the average of the four values is taken as the mutual information feature for a particular ROI as follows:

$$M_i = \frac{M(LL_i;LL_{i-1})+M(HL_i;HL_{i-1})+M(LH_i;LH_{i-1})+M(HH_i;HH_{i-1})}{4}$$

where

$LL_i, HL_i, LH_i; HH_i$ are wavelet subbands of the mouth ROI in the current frame and

$LL_{i-1}, HL_{i-1}, LH_{i-1}; HH_{i-1}$ are wavelet subbands of the mouth ROI in the previous frame.

The Quality measure of one mouth ROI in reference to the previous mouth ROI was calculated as follows:

$$Q = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2+\sigma_y^2)(\bar{x}^2+\bar{y}^2)}$$

where

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$
$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$
$$\sigma_x^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$$
$$\sigma_y^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2$$
$$\sigma_{xy}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})$$

where

X = current ROI

Y = previous ROI

N = number of pixels in the mouth ROI image.

A Quality feature is also calculated in the frequency domain in each of the four subbands and the average of the four values is taken as the quality feature value for a particular ROI as follows.

$$Q_i = \frac{Q(LL_i;LL_{i-1})+Q(HL_i;HL_{i-1})+Q(LH_i;LH_{i-1})+Q(HH_i;HH_{i-1})}{4}$$

where

Q_i = Quality measure for i^{th} ROI.

HL, LH and HH subband coefficients have a Laplace distribution with a mean = 0. If the coefficients are away from the mean by more than the subband standard deviation, those are called significant because they are more likely associated with a significant image feature like an edge or a corner. HL coefficients correspond to vertical features and LH coefficients correspond to horizontal features. From this the ratio of vertical features to horizontal features can be calculated. This ratio is computed as follows.

$$R = \frac{\sum_x \sum_y \begin{cases} 0, & (HL_{median} - \sigma_{HL}) \le HL(x,y) \le (HL_{median} + \sigma_{HL}) \\ 1, & otherwise \end{cases}}{\sum_x \sum_y \begin{cases} 0, & (LH_{median} - \sigma_{LH}) \le LH(x,y) \le (LH_{median} + \sigma_{LH}) \\ 1, & otherwise \end{cases}}$$

where

HL_{median} and LH_{median} are medians of HL and LH subbands, respectively

$HL(x, y)$ and $LH(x, y)$ are coefficients of HL and LH subbands, respectively

σ_{HL} and σ_{LH} are standard deviations of HL and LH subbands, respectively.

Sobel edge detector was applied to the ROI. The horizontal filter S_h highlights the horizontal edges of the ROI and the vertical filter S_v highlights the vertical edges of the ROI. The ratio between the amount of vertical edges to the amount of horizontal edges is calculated as follows.

$$ER = \frac{\sum_{x=1}^{W}\sum_{y=1}^{H}\sum_{i=-1}^{1}\sum_{j=-1}^{1}|ROI(x+i,y+j)S_v(i+1,j+1)|}{\sum_{x=1}^{W}\sum_{y=1}^{H}\sum_{i=-1}^{1}\sum_{j=-1}^{1}|ROI(x+i,y+j)S_v(i+1,j+1)|}$$

where

$ROI(x, y)$ = intensity value at (x,y) in the ROI

H = height of the ROI

W = width of the ROI.

The amount of red color in the ROI indicates the appearance of tongue. The amount of tongue that appears is calculated as the ratio of the amount of red color and the size of the ROI as follows:

$$RC = \frac{\sum_{x=1}^{W}\sum_{y=1}^{H} Red(ROI(x,y))}{WH}$$

where

$Red(ROI(x, y))$ - Red component value at (x,y) in the ROI.

The amount of appearance of teeth was also taken as a feature. The ROI image was converted into 1976 CIELAB color space (L*,a*,b*)[33] and 1976 CIELUV color space (L*,u*,v*)[33] and a teeth pixel was defined as follows.

$$t = \begin{cases} 1, & a* \leq (\mu_a - \sigma_a) \\ 1, & u* \leq (\mu_u - \sigma_u) \\ 0, & otherwise \end{cases}$$

where

μ_a, σ_a = mean and standard deviation of a*

μ_u, σ_u = mean and standard deviation of u*.

The number of teeth pixels is calculated as

$$T = \sum_{x=1}^{W}\sum_{y=1}^{H} t(x, y).$$

So, mouth ROI height, width, mutual information, quality, ratio of vertical to horizontal features, ratio of vertical edges to horizontal edges, amount of tongue and amount of teeth are calculated as features. For each of these features, weights are assigned according to the relative importance of that feature. This weight was computed as the ratio of the word recognition rate obtained with only that feature and the sum of the word recognition rates of all features. D_i is the distance of the i^{th} feature vector of a test sample from the i^{th} feature vector of a training sample. This distance for each feature vector is calculated using Dynamic Time Warping. The weighted average of these eight distances is taken as the distance of a test sample from the training sample.

After computation of the distance of a test sample from the training samples, the k nearest neighbor rule (kNNR) has been used with k = 3 to recognize a test sample. Speaker dependent tests were conducted on a database of 27 speakers. An overall recognition rate of **76.38%** was obtained.

Chapter 3

Goal for Research and Methods Used

This chapter describes the goal of this particular research and the specific methods used to achieve this goal

3.1 Statement of the Problem

The goal of this thesis is to develop a novel method to perform visual speech recognition using input from the database developed in Clemson University called Clemson University Audio Visual Experiments (CUAVE)[34]. The database contains audio and video files of the speech of 33 speakers. There are 17 male speakers and 16 female speakers. For this thesis only the video files in the database were used as input because visual speech recognition needs to be performed.

The camera is positioned such that a speaker is directly facing it. Each speaker speaks the words 0 to 9. He/she repeats this for five consecutive times. Hence, each speaker speaks 50 words. In this thesis, the video corresponding to each word is taken as the input in an isolated manner. So isolated word speech recognition has been performed. Also, speech from a speaker was taken for training the system and speech from the same speaker has been taken for testing.the system. So speaker dependent speech recognition has been performed. Hence the goal is to develop a computer program that takes the video of a word as input and gives a label, referring to which word has been

Figure 3.1: Feature Points Detected on a Face Image

spoken, as output.

3.2 Approach

This section explains the specific methods used in this thesis.

3.2.1 Extraction of Mouth Region of Interest

A video input contains a sequence of frames. In each of these frames, on the speaker's face, a set of feature points are detected. An example of this detection is shown in the Figure 3.1. In particular, 18 points are detected on the mouth of the speaker.

These 18 points allow the segmentation of the mouth region alone. The rectangular region of the image bounded by the top most point, the bottom most point, the left most point and the right most point of these 18 points contains the mouth region of interest (ROI). This rectangular

Figure 3.2: Mouth Region of Interest (ROI)

region is cropped from a frame. Such a mouth ROI is shown in Figure 3.2.

3.2.2 Feature Extraction

From each of the ROI images, features are extracted. The following three features are extracted from each ROI:

1. Mouth ROI Height
2. Mouth ROI Width
3. Appearance of tongue.

The method of extraction of each of these features is explained in the following.

3.2.2.1 Mouth ROI Height

The number of pixels along the row dimension of the mouth ROI image is taken to be the Mouth ROI Height.

3.2.2.2 Mouth ROI Width

The number of pixels along the column dimension of the mouth ROI is taken to be the Mouth ROI Width.

MATLAB is used in the experiments performed in this thesis. In MATLAB, the spatial coordinate system is used for images in which the spatial resolution is $\frac{1}{10000}^{th}$ of a pixel dimension. For example, an image position of (156.1824,123.4876) is perfectly valid. This coordinate system is used in finding the Mouth ROI Height and the Mouth ROI Width.

3.2.2.3 Appearance of Tongue

The amount of red color in the ROI represents the amount of lips and tongue. The amount of red color contributed by the lips in the ROI will stay constant. So, as this value changes from the

ROI in one frame to the next, it gives an indication of how much of the tongue is visible. The sum of the red component values of all RGB pixels in the mouth ROI image is computed. This sum is normalized by the number of pixels in the image. This gives the following measure of the amount of red color in the ROI:

$$RC = \frac{\sum_{x=1}^{h} \sum_{y=1}^{w} Red(ROI(x,y))}{hw}$$

where

ROI - mouth ROI

h - mouth ROI height

w - mouth ROI width.

3.2.2.4 Trajectory of Features

In an input video, from the ROI in each frame, these 3 features are calculated. Hence for each video, a time series of 3-dimensional feature vectors is obtained. To visualize how the features height, width and appearance of tongue vary together with frames, they are plotted in a 3D plot. Such a plot will result in a trajectory as shown in Figure 3.3.

3.3 Histogram of Oriented Displacements

3.3.1 Variable Lengths of Feature Trajectories

Videos containing different words are of different durations. Also different speakers speak at different speeds. Even the same speaker may speak the different renditions of the same word with different durations. Hence the number of frames in different videos are different. So the features extracted from a video will also be of different lengths. Figures 3.4 and 3.5 shows examples of the height feature corresponding to two different attempts of the word 0. Figure 3.4 shows that the word has 20 frames and Figure 3.5 shows that the word has 19 frames. So the 3D trajectories of these different words will also be of different lengths.

3.3.2 Histogram of Oriented Displacements

To compare the different trajectories for recognition, they have to be described by fixed length descriptors. Hence each trajectory has to be converted to a fixed length vector. The technique

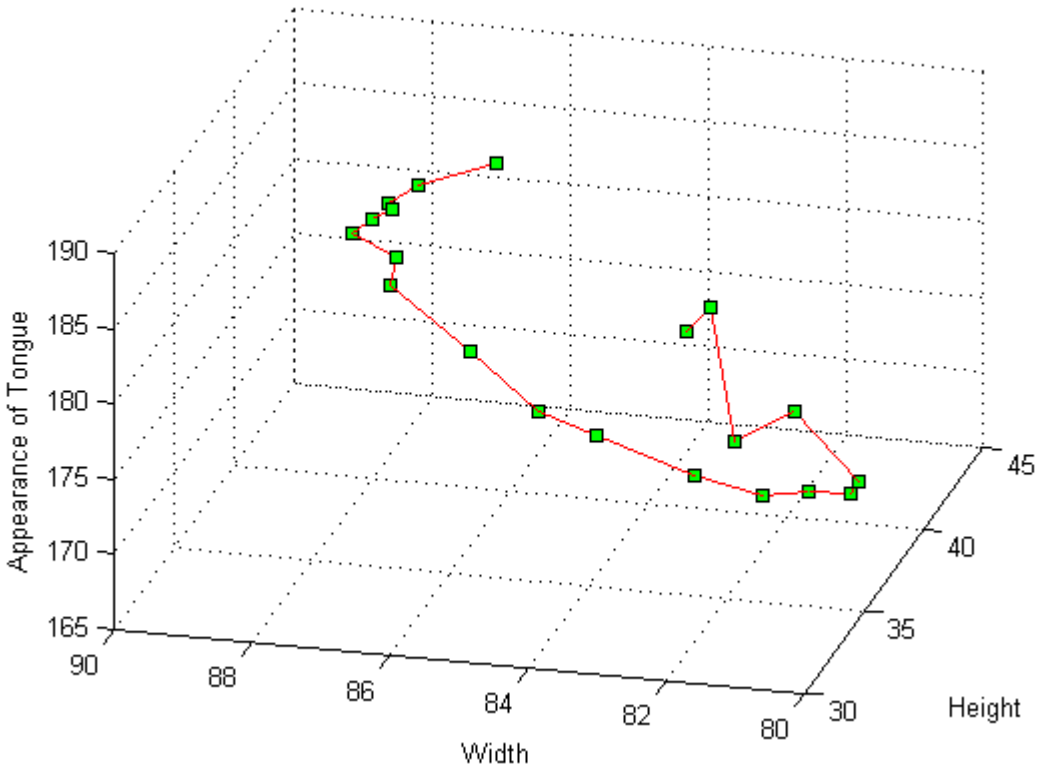

Figure 3.3: 3-Dimensional Trajectory of Features

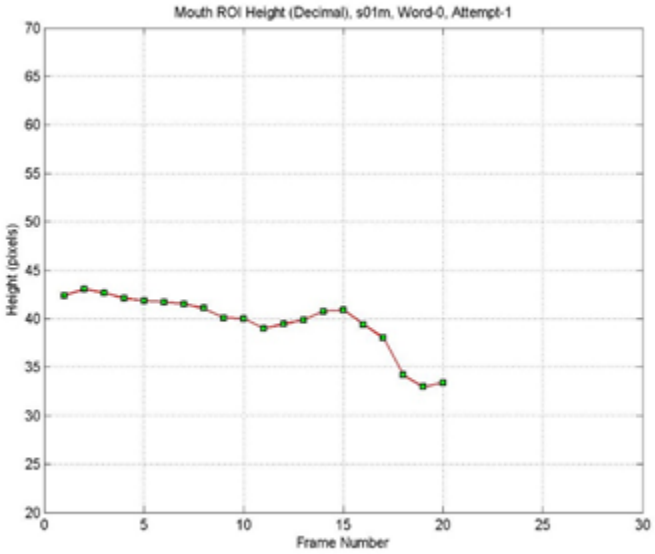

Figure 3.4: Word Zero, Attempt-1

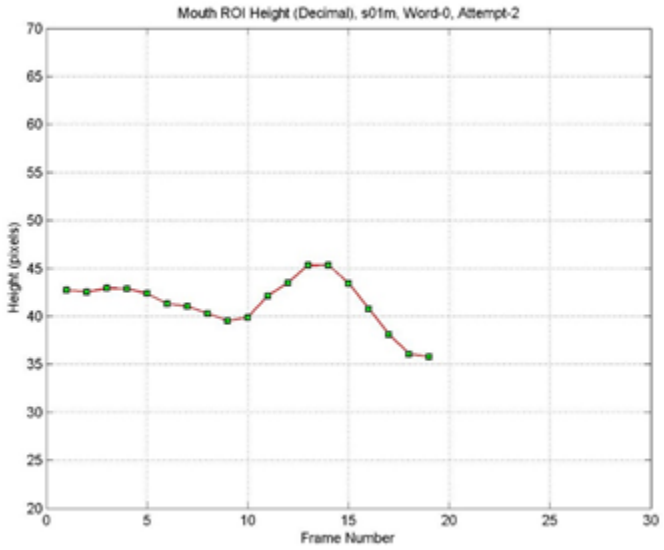

Figure 3.5: Word Zero, Attempt-2

Angle range	Quantized value
[0,45)	45
[45,90)	90
[90,135)	135
[135,180)	180
[180,225)	225
[225,270)	270
[270,315)	315
[315,360)	360

Table 3.1: Angle Quantization

used in this thesis for accomplishing this is computing the Histogram of Oriented Displacements (HOD) [35] for each trajectory.

The method of computation of HOD is explained as follows. Consider a 2-dimensional trajectory. It is a sequence of points. Let it be denoted as $T = \{P_1, P_1, P_3, ..., P_n\}$ where P_t is the 2D position at time t. For each pair of points (P_t, P_{t+1}), the slope of the line connecting these two points is calculated as follows.

$$slope = \frac{P_{t+1.y} - P_{t.y}}{P_{t+1.x} - P_{t.x}}.$$

From this slope, the angle between the two points θ can be calculated which will be between 0° and 360°. These θ values are quantized into 8 values. If θ lies between 0° and 45°, it is quantized into 45° and if it lies between 45° and 90°, it is quantized into 90° etc. Table 3.1 shows this quantization process. A histogram of the quantized values of θ is created. For each θ, the corresponding histogram bin is calculated as follows:

$$bin = \frac{\theta * 8}{360}.$$

The length of the line between the two points is then added to the specific histogram bin. Thus, the histogram accumulates the lengths of the consecutive moves in the corresponding orientations. In this way, a 2D trajectory can be converted into a vector of fixed length 8. An example of such a histogram is shown in Figure 3.6.

3.3.3 Temporal Pyramid

Dealing with the trajectory as a whole misses the temporal information. To capture this information, a temporal pyramid approach is used. The trajectory is split into multiple levels. In

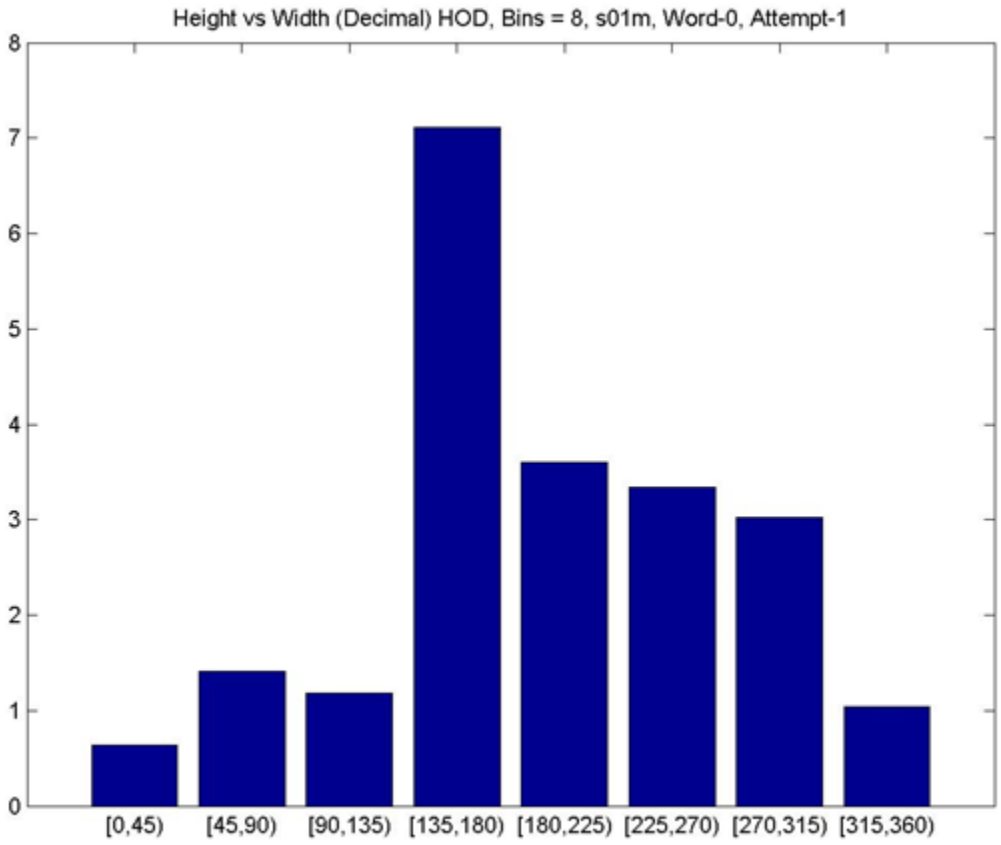

Figure 3.6: Histogram of Oriented Displacements

level 1, the whole trajectory is considered and the histogram is computed for it. In level 2, the trajectory is split into 2 halves and two histograms are computed, one for each of the two halves. In level 3, the trajectory is split into 4 quarters and four histograms are computed. Hence a total of 7 histograms are computed. The final descriptor of the 2D trajectory is the concatenation of these 7 histograms. With each histogram having 8 bins, the final descriptor's length is 7*8=56.

3.3.4 Descriptor for the 3D Trajectory

The descriptor for a 3D trajectory is obtained by concatenating the individual descriptors of the three 2D projections. Hence, the final descriptor of a 3D trajectory describing each word is of length 3*56=168. So the video of each word is described by a 168 dimensional vector. We call these vectors **Word Vectors**. An example of such a word vector is shown in Figure 3.7.

3.4 Recognition

A Support Vector Machine (SVM) is used to recognize the word vectors. The SVM finds the optimal hyperplane that would separate two classes with minimum number of misclassifications. The optimal hyperplane is the one that maximizes the margin between itself and the vectors closest to the hyperplane called support vectors. The margin maximization problem is formulated into a Lagrange multipliers problem which ultimately can be represented as a convex quadratic optimization problem. From the solution of this optimization problem the parameters of the hyperplane can be computed. SVM can be used to fit a hyperplane to both linearly separable and non linearly separable data. It is explained in detail in the following. The following subsections (3.4.1 and 3.4.2) explaining SVM have been taken from [36].

3.4.1 Linearly Separable Binary Classification

Let the training set have L training points, each with D dimensions and each one of them belongs to one of two classes, $y_i = +1$ or $y_i = -1$. So our training data is of the form:

$$\{\mathbf{x}_i, y_i\} \quad where \quad i = 1, ..., L, \quad y_i \in \{+1, -1\}, \quad \mathbf{x} \in R^D. \tag{3.1}$$

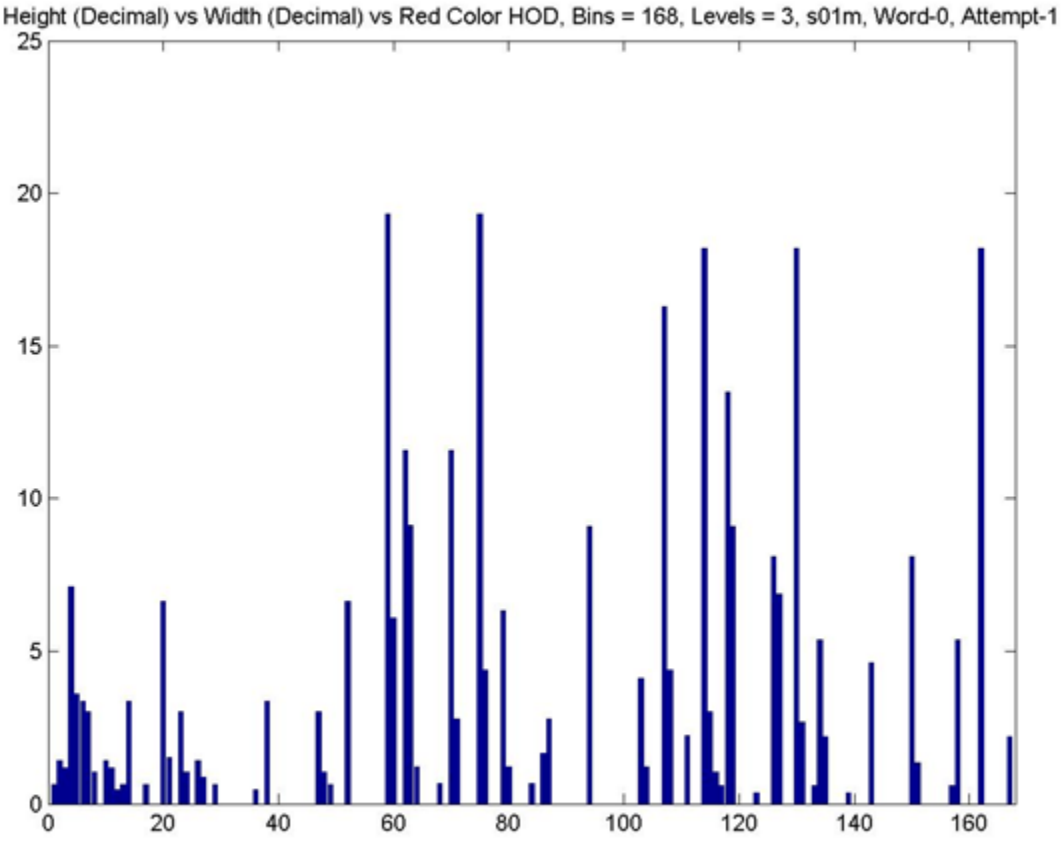

Figure 3.7: Word Vector

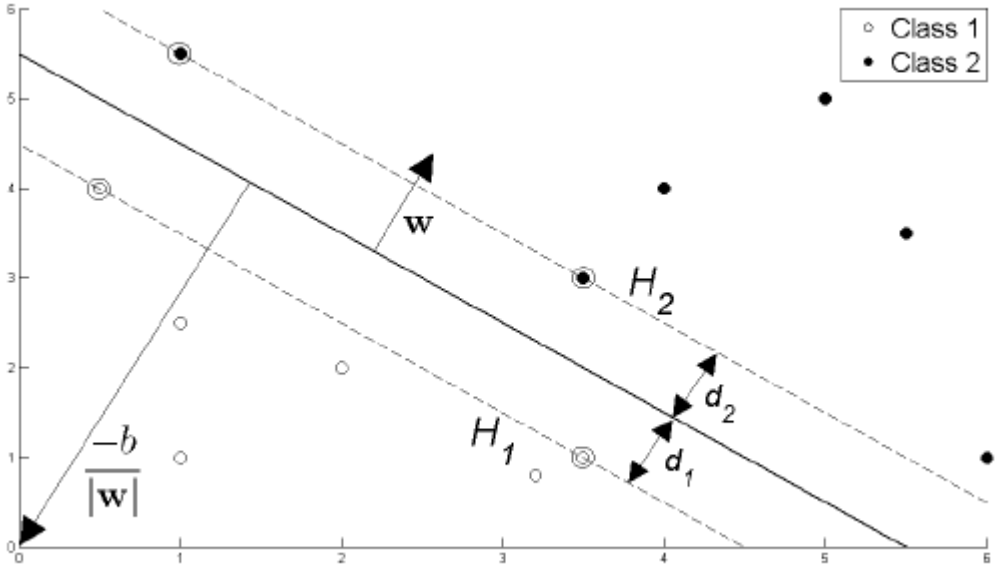

Figure 3.8: Hyperplane Through Two Linearly Separable Classes [36]

Consider the data to be linearly separable. The hyperplane separating the two classes can be described by $\mathbf{w}^T\mathbf{x} + b = 0$ where

1. $\mathbf{w}$ is normal to the hyperplane
2. $\frac{b}{\|\mathbf{w}\|}$ is the perpendicular distance from the origin to the hyperplane.

See Figure 3.8 for an illustration. Support vectors are the examples closest to the separating hyperplane, and the aim of SVM is to orientate the hyperplane in such a way as to be as far as possible from the closest members of both classes. Implementing SVM involves to finding $\mathbf{w}$ and b such that the following equations hold:

$$\mathbf{x}_i{}^T\mathbf{w} + b \geq +1 \quad for \quad y_i = +1 \tag{3.2}$$

$$\mathbf{x}_i{}^T\mathbf{w} + b \leq -1 \quad for \quad y_i = -1. \tag{3.3}$$

These equations can be combined into

$$y_i({\mathbf{x}_i}^T\mathbf{w} + b) - 1 \geq 1 \quad \forall i. \tag{3.4}$$

In Figure 3.8, the support vectors on either classes lie on two planes H_1 and H_2. d_1 is the distance between H_1 and the hyperplane and d_2 is the distance between H_2 and the hyperplane. The hyperplane must be equidistant from these two planes, so $d_1 = d_2$. This quantity is called the SVM's margin. In order to orientate this hyperplane to be as far from the support vectors as possible, this margin needs to be maximized.

Using vector geometry, this margin is equal to $\frac{1}{\|\mathbf{w}\|}$ and maximizing it subject to the constraint in (3.3) is equivalent to finding:

$$min\|\mathbf{w}\| \quad such \quad that \quad y_i({\mathbf{x}_i}^T\mathbf{w} + b) - 1 \geq 1 \quad \forall i. \tag{3.5}$$

Minimizing $\|\mathbf{w}\|$ is equivalent to minimizing $\frac{1}{2}\|\mathbf{w}\|^2$. We must therefore find:

$$min\frac{1}{2}\|\mathbf{w}\|^2 \quad such \quad that \quad y_i({\mathbf{x}_i}^T\mathbf{w} + b) - 1 \geq 1 \quad \forall i. \tag{3.6}$$

In order to perform this minimization, the Lagrange multipliers method is followed. Let α_i, i = 1,...,L such that $\alpha_i \geq 0, \quad \forall i$ be the lagrange multipliers.

$$L_P = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{L} \alpha_i[y_i({\mathbf{x}_i}^T\mathbf{w} + b) - 1] \tag{3.7}$$

$$L_P = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{L} \alpha_i y_i({\mathbf{x}_i}^T\mathbf{w} + b) - \sum_{i=1}^{L} \alpha_i. \tag{3.8}$$

We wish to find the $\mathbf{w}$ and b which minimizes and the α_i, i=1,...,L that maximize (3.7). We can do this by differentiating L_P with respect to $\mathbf{w}$ and b and setting the derivatives to zero:

$$\frac{\partial L_P}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{L} \alpha_i y_i \mathbf{x}_i \tag{3.9}$$

$$\frac{\partial L_P}{\partial b} = 0 \Rightarrow \sum_{i=1}^{L} \alpha_i y_i = 0. \tag{3.10}$$

Substituting (3.9) and (3.10) in (3.8) gives the expression (3.11). This has to be maximized with respect to α_i, i=1,...L.

$$L_D = \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j \quad s.t. \quad \alpha_i \geq 0 \quad \forall i, \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{3.11}$$

$$L_D = \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i H_{ij} \alpha_j \quad where \quad H_{ij} = y_i y_j {\mathbf{x}_i}^T \mathbf{x}_j \tag{3.12}$$

$$L_D = \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \boldsymbol{\alpha}^T H \boldsymbol{\alpha} \quad \alpha_i \geq 0 \forall i, \sum_{i=1}^{L} \alpha_i y_i = 0. \tag{3.13}$$

This new formulation L_D is called the dual form of the Primary L_P. This L_D has to be maximized, that is, we need to implement the following:

$$\max_{\alpha} [\sum_{i=1}^{L} \alpha_i - \frac{1}{2} \boldsymbol{\alpha}^T H \boldsymbol{\alpha}] \quad s.t. \quad \alpha_i \geq 0 \forall i, \sum_{i=1}^{L} \alpha_i y_i = 0. \tag{3.14}$$

This is a convex quadratic optimization problem, and solving this will give $\boldsymbol{\alpha}$ and then (3.9) will enable the calculation of the appropriate value of $\mathbf{w}$.

A datapoint $\mathbf{x}_s$ is called a support vector if

$$y_s({\mathbf{x}_s}^T \mathbf{w} + b) = 1. \tag{3.15}$$

These support vectors are those that correspond to α_i>0. For all other datapoints α_i will be 0. Substituting the expression for $\mathbf{w}$ from (3.9), we get the following expression:

$$y_s(\sum_{m \in S} \alpha_m y_m \mathbf{x}_m . \mathbf{x}_s + b) = 1 \tag{3.16}$$

where S is the set of all support vectors. Simplifying this, we get an expression for b:

$$b = y_s - \sum_{m \in S} \alpha_m y_m \mathbf{x}_m . \mathbf{x}_s. \tag{3.17}$$

It is better to take an average over all the support vectors, to find the optimal value of b:

$$b = \frac{1}{N_s} \sum_{s \in S} (y_s - \sum_{m \in S} \alpha_m y_m \mathbf{x}_m . \mathbf{x}_s). \tag{3.18}$$

Now, the optimal values for **w** and b that will result in the hyperplane are available.

3.4.2 Binary Classification for Non-linearly Separable Data

To find a hyperplane for non linearly separable data, constraints (3.2) and (3.3) are relaxed slightly to allow for misclassified points by introducing a positive slack variable ξ_i, i=1,...,L:

$$\mathbf{x}_i{}^T\mathbf{w} + b \geq +1 + \xi_i \quad for \quad y_i = +1 \tag{3.19}$$

$$\mathbf{x}_i{}^T\mathbf{w} + b \leq -1 + \xi_i \quad for \quad y_i = -1 \tag{3.20}$$

$$\xi_i \geq 0 \quad \forall i. \tag{3.21}$$

These conditions can be combined into:

$$y_i(\mathbf{x}_i{}^T\mathbf{w} + b) - 1 + \xi_i \geq 0 \quad where \quad \xi_i \geq 0 \quad \forall i. \tag{3.22}$$

In this formulation, the data points that are on the incorrect side of the hyperplane have a penalty that increases with the distance from it. The number of misclassifications should be minimized and the following optimization should be performed:

$$min\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{L}\xi_i \quad s.t. \quad y_i(\mathbf{x}_i{}^T\mathbf{w} + b) - 1 + \xi_i \geq 0 \quad \forall i \tag{3.23}$$

where the parameter C controls the trade-off between the slack variable penalty and the size of the margin. This optimization is reformulated into a Lagrangian as follows:

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{L}\xi_i - \sum_{i=1}^{L}\alpha_i[y_i(\mathbf{x}_i{}^T\mathbf{w} + b - 1 + \xi_i)] - \sum_{i=1}^{L}\mu_i\xi_i. \tag{3.24}$$

This Lagrangian has to be minimized with respect to **w**, b and ξ_i and maximized with respect to $\boldsymbol{\alpha}$ (where $\alpha_i \geq 0$, $\mu_i \geq 0$, $\forall i$). Differentiating with respect to **w**, b and ξ_i and setting the derivatives to 0 gives:

$$\frac{\partial L_P}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{L} \alpha_i y_i \mathbf{x}_i \tag{3.25}$$

$$\frac{\partial L_P}{\partial b} = 0 \Rightarrow \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{3.26}$$

$$\frac{\partial L_P}{\partial \xi_i} = 0 \Rightarrow C = \alpha_i + \mu_i. \tag{3.27}$$

Substituting these in (3.24), we get L_D which has the same form as (3.13) above. (3.27), together with $\mu_i \geq 0, \forall i$ implies $0 \leq \alpha_i \leq C, \forall i$. So we need to find:

$$\max_{\alpha}[\sum_{i=1}^{L} \alpha_i - \frac{1}{2}\boldsymbol{\alpha}^T H \boldsymbol{\alpha}] \quad s.t. \quad 0 \leq \alpha_i \leq C \quad \forall i \quad and \quad \sum_{i=1}^{L} \alpha_i y_i = 0. \tag{3.28}$$

After the appropriate values of α_i, i =1,...,L are found, $\mathbf{w}$ is calculated. b is calculated in the same way as in (3.18), but the set of support vectors used to calculate b are determined by finding the indices for which $0 < \alpha_i <$C. This is how a hyperplane is found in the case of a non linearly separable dataset. Once the hyperplane has been found, a new datapoint $\mathbf{x}_u$ is classified by evaluating $y_u = sgn(\mathbf{w}^T \mathbf{x}_u + b)$.

3.4.3 Kernels

Datasets which are not linearly separable may become linearly separable when mapped to a higher dimensional space.Kernel methods map the data into higher dimensional spaces in the hope that in these new higher dimensional spaces, the data may become easily separable or better structured. But the mapping need not be explicitly computed because of the kernel trick. As can be seen from (3.14), the function that needs to be maximized involves a matrix denoted H where $H_{ij} = y_i y_j {\mathbf{x}_i}^T \mathbf{x}_j$ (from (3.12)). Hence the algorithm involves only the inner product between two data vectors. Hence, even after mapping the data to higher dimensional space, we need to consider only the inner product in that space, so there is no need of explicit computation of the mapping itself. If ϕ is the mapping, then the inner product is expressed as:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j). \tag{3.29}$$

This function is called the kernel function. There are different kernel function available. When $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i{}^T \mathbf{x}_j$, it is called a linear kernel. In this thesis, linear kernel has been used.

3.4.4 Multiclass Classification

In a recognition problem with K classes, with K >2, K(K-1)/2 binary classifiers are trained. Each classifier receives the training samples from a pair of classes and learns to distinguish between those two classes. At prediction time, a voting scheme is applied. All K(K-1)/2 classifiers are applied to the new sample and the class that received the highest number of "+1" predictions gets predicted by the overall classifier. This approach is called one-vs-one approach to multiclass classification. This approach is used in this thesis to perform classification of a spoken word into 10 different classes.

Chapter 4

Results

Experiments were conducted using the methods explained in the previous chapter and recognition results were collected. In this chapter, experiments are explained in detail and the results are reported.

4.1 Experiments

4.1.1 Explanation of the Database

Clemson University Audio Visual Experiments (CUAVE) database [34] has audio and video files containing the speech activity of 33 speakers. This database was developed in Clemson University. As described earlier, there are 17 males and 16 females. The speakers look straight into the camera and say the words 0 to 9 consecutively for five times. Hence each speaker speaks 50 times. The video files containing the individual words spoken by these speakers are cut using MATLAB and are used as inputs. From these video inputs word vectors were collected.

4.1.2 Speaker Dependent Experiments

Speaker dependent recognition experiments were performed. This means that the training of the system was performed using training data from a single speaker and the testing data from the same speaker was used to perform classification of the words. This was repeated for all 33 speakers.

4.1.3 k-Fold Validation

Cross validation is a technique used to validate a model developed for a statistical problem like classification or regression. It is a validation technique used to assess how a model will generalize to unknown data. Suppose we have a model with unknown parameters and a dataset to which the model should be fit, that is, the training data. The fitting process optimizes the parameters so that the model fits the training data as well as possible. Cross validation is a way to predict the fit of this model to unseen data. One round of cross validation involves partitioning the dataset into complementary subsets. One subset is used to train the system and the system is tested on the other subset and test results are collected. Multiple rounds of this process are performed on different partitions and the results are averaged over all rounds. The overall result is a valid statistical percentage.

k-fold validation is a type of cross validation method in which the original dataset is partitioned into k equal size subsamples. Of the k subsamples, one subsample is taken as the validation set or testing set and the remaining k-1 subsamples are taken to be the training set. This cross validation process is repeated k times (folds) by taking each of the k subsamples exactly once as the testing set. The k recognition results are then averaged and the average is the final recognition result. In this thesis, a value of 5 is taken for k. Hence 5-fold validation is performed.

There are 50 word vectors extracted from each of the speakers. According to 5-fold validation, this dataset is partitioned into 5 subsets of 10 word vectors each. 5 rounds of recognition are performed on these subsets. In the first round, one subset of 10 word vectors is taken as the testing set and the remaining 4 subsets containing 40 word vectors are collectively taken as the training data and recognition result are collected. In the second round, a different subset is taken as the testing data and the remaining subsets are taken as the training data and the recognition result is collected. This process is repeated for 5 times and collectively 5 recognition percentages are available. These are averaged to find the recognition percentage for a speaker. This process is performed for all 33 speakers.

4.2 Results

In this section, the recognition results are reported. Table 4.1 shows these results. The first column of this table shows the speaker. The second column through the sixth column give the

recognition rate for each of the five rounds in 5-fold validation. The last column gives the average of the 5 recognition rates, and this is the final recognition rate for the corresponding speaker. The average of the recognition rates of all 33 speakers is **81.03%**.

Speaker	**fold 1**	**fold 2**	**fold 3**	**fold 4**	**fold 5**	**Average**
Speaker 1	100	90	100	80	100	94
Speaker 2	80	90	100	100	100	94
Speaker 3	70	100	70	90	60	78
Speaker 4	60	80	60	70	60	66
Speaker 5	80	90	90	80	90	86
Speaker 6	100	80	70	80	90	84
Speaker 7	70	80	70	70	60	70
Speaker 8	90	80	90	90	70	84
Speaker 9	70	80	70	90	90	80
Speaker 10	80	90	90	80	90	86
Speaker 11	80	70	80	70	90	78
Speaker 12	60	70	60	80	70	68
Speaker 13	80	80	100	90	100	90
Speaker 14	70	80	80	100	80	82
Speaker 15	80	90	90	90	80	86
Speaker 16	80	70	100	90	80	84
Speaker 17	90	50	50	70	90	70
Speaker 18	60	80	70	80	80	74
Speaker 19	60	60	80	70	70	68
Speaker 20	80	40	80	80	90	74
Speaker 21	60	70	60	70	50	62
Speaker 22	80	100	90	80	90	88
Speaker 23	100	100	100	70	90	92
Speaker 24	70	100	100	100	90	92
Speaker 25	70	90	70	90	70	78
Speaker 26	70	80	80	70	70	74
Speaker 27	70	80	80	60	90	76
Speaker 28	70	80	90	100	90	86
Speaker 29	70	90	80	90	100	86
Speaker 30	90	60	100	80	80	82
Speaker 31	100	80	90	90	90	90
Speaker 32	80	80	80	80	80	80
Speaker 33	100	90	100	80	90	92

Table 4.1: Recognition Results

Chapter 5

Conclusions and Discussion

5.1 Answering the Research Questions

Visual Speech Recognition (VSR) is an area of research that involves collaborative efforts among multiple disciplines including speech processing, image processing, computer vision and pattern recognition. VSR has the potential to improve the accuracy of audio-only speech recognition systems especially when the level of noise is high. It has applications in Human Computer Interaction (HCI) such as visually recognizing passwords while logging in. VSR systems are part of Audio-Visual Speech Recognition (AVSR) systems which combine the decisions from audio speech recognition and visual speech recognition to give the final recognition result which results in improved speech recognition accuracy. These systems have the potential to be implemented in consumer electronics devices such as smartphones, car stereo systems, etc., and hence have commercial viability, too.

A critical part of the VSR systems is feature extraction and recognition. The specific methods used to extract features and the recognition algorithms used are major factors in determining the final accuracy of the system. In this thesis, these questions have been studied. The specific methods followed can be summarized as follows.

5.1.1 Fixed Length Description of Features

Mouth ROI height, mouth ROI width and amount of appearance of tongue have been extracted from each frame of the input video. A method called Histogram of Oriented Displacements

(HOD) has been applied to describe a sequence of these features extracted from each word using a fixed length descriptor. These fixed length descriptors were called word vectors.

5.1.2 Recognition of Words

Support vector machine (SVM) with linear kernel has been used to recognize the word vectors. Cross validation using 5-fold validation has been used to collect the recognition results.

5.1.3 Experimental Results

Speaker dependent recognition experiments have been performed on the CUAVE database that contains speech videos of words 0 to 9. An average recognition rate of 81.03% was obtained over 33 speakers.

5.2 Contributions

Feature extraction method has been the major topic of interest in this research. For the first time in VSR research, the Histogram of Oriented Displacements (HOD) method has been used to describe the features extracted. Given the good recognition accuracy obtained, it can be concluded that the HOD method is suitable for VSR.

Also, the suitability of Support Vector Machine (SVM) has been studied. SVM proves to be an effective algorithm that can be used in VSR.

5.3 Recommendations for Future Work

In this thesis, speaker dependent recognition has been performed. In the future, speaker independent recognition using the HOD feature extraction method should be studied.

The developed VSR system should be incorporated in an AVSR system and tested for the overall recognition accuracy. Such a study will give insights into the suitability of the developed VSR system in AVSR.

SVM using kernels other than the linear kernel like RBF kernel, polynomial kernel or sigmoidal kernel should be used for recognition and the corresponding recognition accuracies obtained should be studied.

Recognition algorithms other than SVM, such as k Nearest Neighbor Rule (kNNR) and, artificial neural network (ANN) algorithms could be used to recognize the word vectors. kNNR suffers from the curse of dimensionality. When the dimensionality of the feature vectors is high, its performance becomes poor. Hence the dimensionality of the word vectors, which is 168 in this thesis, has to be reduced when using kNNR. Such a study will give insights into the suitability of those algorithms for VSR using the HOD feature extraction method.

Bibliography

[1] H. McGurk and J. MacDonald, "Hearing lips and seeing voices," *Nature*, vol. 264, pp. 746–748, Dec. 1976.

[2] E. D. Petajan, *Automatic Lipreading to Enhance Speech Recognition (Speech Reading)*. PhD thesis, Champaign, IL, USA, 1984. AAI8502266.

[3] C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, and D. Vergyri, "Large-vocabulary audio-visual speech recognition: a summary of the johns hopkins summer 2000 workshop," in *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on*, pp. 619–624, 2001.

[4] M. Gordan, C. Kotropoulos, and I. Pitas, "Visual speech recognition using support vector machines," in *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*, vol. 2, pp. 1093–1096 vol.2, 2002.

[5] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.

[6] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[7] C. Benoît, M. T. Lallouache, T. Mohamadi, and C. Abry, "A set of French visemes for visual speech synthesis," in *Talking Machines: Theories, Models and Designs* (G. Bailly and C. Benoît, eds.), pp. 485–504, North-Holland, Amsterdam: Elsevier Science Publishers B. V., 1992.

[8] S. J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book Version 3.4*. Cambridge University Press, 2006.

[9] J. Platt, "Probabilistic outputs for support vector machines and comparison to regularize likelihood methods," in *Advances in Large Margin Classifiers* (A. Smola, P. Bartlett, B. Schoelkopf, and D. Schuurmans, eds.), pp. 61–74, 2000.

[10] J. R. Movellan, "Visual speech recognition with stochastic networks.," in *NIPS* (G. Tesauro, D. S. Touretzky, and T. K. Leen, eds.), pp. 851–858, MIT Press, 1994.

[11] K. Saenko, K. Livescu, M. Siracusa, K. Wilson, J. Glass, and T. Darrell, "Visual speech recognition with loosely synchronized feature streams," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, pp. 1424–1431 Vol. 2, Oct 2005.

[12] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. Senior, "Recent advances in the automatic recognition of audiovisual speech," *Proceedings of the IEEE*, vol. 91, pp. 1306–1326, Sept 2003.

[13] S. King, T. A. Stephenson, S. Isard, P. Taylor, and A. Strachan, "Speech recognition via phonetically featured syllables.," in *ICSLP*, ISCA, 1998.

[14] J. Gowdy, A. Subramanya, C. Bartels, and J. Bilmes, "Dbn based multi-stream models for audio-visual speech recognition," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 1, pp. I–993–6 vol.1, May 2004.

[15] K. Livescu and J. Glass, "Feature-based pronunciation modeling for speech recognition," in *Proc. HLT/NAACL*, 2004.

[16] T. J. Hazen, K. Saenko, C.-H. La, and J. R. Glass, "A segment-based audio-visual speech recognizer: Data collection, development, and initial experiments," in *Proceedings of the 6th International Conference on Multimodal Interfaces*, ICMI '04, (New York, NY, USA), pp. 235–242, ACM, 2004.

[17] K. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning*. University of California, Berkeley, 2002.

[18] X. Hong, H. Yao, Y. Wan, and R. Chen, "A pca based visual dct feature extraction method for lip-reading," in *Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP '06. International Conference on*, pp. 321–326, Dec 2006.

[19] G. Potamianos, H. Graf, and E. Cosatto, "An image transform approach for hmm based automatic lipreading," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, pp. 173–177 vol.3, Oct 1998.

[20] P. Duchnowski, M. Hunke, D. Busching, U. Meier, and A. Waibel, "Toward movement-invariant automatic lip-reading and speech recognition," in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, vol. 1, pp. 109–112 vol.1, May 1995.

[21] P. Scanlon and R. Reilly, "Feature analysis for automatic speechreading," in *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on*, pp. 625–630, 2001.

[22] M. Heckmann, K. Kroschel, C. Savariaux, and F. Berthommier, "DCT-based video features for audio-visual speech recognition," in *Proceedings of International Conference on Spoken Language Processing*, pp. 1925–1928, 2002.

[23] A. Nefian, L. Liang, X. Pi, X. Liu, and K. Murphy, "Dynamic bayesian networks for audio-visual speech recognition," in *Eurasip journal on applied signal processing*, pp. 1274–1288, November 2002.

[24] G. I.Chiou and J.-N. Hwang, "Lipreading from color video," *Image Processing, IEEE Transactions on*, vol. 6, pp. 1192–1195, Aug 1997.

[25] C. Bregler and Y. Konig, "Eigenlips for robust speech recognition," in *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, vol. ii, pp. II/669–II/672 vol.2, Apr 1994.

[26] S. Dupont and J. Luettin, "Audio-visual speech modeling for continuous speech recognition," *Multimedia, IEEE Transactions on*, vol. 2, pp. 141–151, Sep 2000.

[27] J. Luettin and N. A. Thacker, "Speechreading using probabilistic models," *Computer Vision and Image Understanding*, vol. 65, no. 2, pp. 163 – 178, 1997.

[28] K. Riedhammer, T. Bocklet, A. Ghoshal, and D. Povey, "Revisiting semi-continuous hidden markov models," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4721–4724, March 2012.

[29] N. Ukai, S. Takumi, S. Tamura, and S. Hayamizu, "GIF-LR: ga-based informative feature for lipreading," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2012, Hollywood, CA, USA, December 3-6, 2012*, pp. 1–4, 2012.

[30] S. Tamura, C. Miyajima, N. Kitaoka, T. Yamada, S. Tsuge, T. Takiguchi, K. Yamamoto, T. Nishiura, M. Nakayama, Y. Denda, M. Fujimoto, S. Matsuda, T. Ogawa, S. Kuroiwa, K. Takeda, and S. Nakamura, "CENSREC-1-AV: an audio-visual corpus for noisy bimodal speech recognition," in *Auditory-Visual Speech Processing, AVSP 2010, Hakone, Kanagawa, Japan, September 30 - October 3, 2010*, p. 6, 2010.

[31] P. Bordea, A. Varpeb, R. Manzac, and P. Yannawara, "Recognition of isolated words using zernike and MFCC features for audio visual speech recognition," *CoRR*, vol. abs/1407.1165, 2014.

[32] A. B. Hassanat, "Visual words for automatic lip-reading," *CoRR*, vol. abs/1409.6689, 2014.

[33] I. X-RIte, *A Guide to Understanding Color Communication*. X-Rite, 1993.

[34] E. Patterson, S. Gurbuz, Z. Tufekci, and J. Gowdy, "Cuave: A new audio-visual database for multimodal human-computer interface research," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 2, pp. II–2017–II–2020, May 2002.

[35] M. A. Gowayyed, M. Torki, M. E. Hussein, and M. El-Saban, "Histogram of oriented displacements (hod): Describing trajectories of human joints for action recognition," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pp. 1351–1357, AAAI Press, 2013.

[36] T. Fletcher, "Support vector machines explained," 2008.