CHAPTER 1

A relation is something which requires understanding the feeling of others

PREAMBLE

This project is given the name “TO BUILD A MODEL FOR IMPLEMENTING AUTOMATED LIP READING TECHNIQUE WHICH INVOLVES LIP MOTION FEATURE TO TEXT CONVERSION”.

A speech recognition system has three major components: feature extraction, probabilistic modelling of features and classification. In literature, the general approach is to extract the principle components of the lip movement in terms of the lip shape based properties in order to establish a one-to-one correspondence between phonemes of speech and visemes of lip shape. Several modelling and classification aspects for speech reading are studied and compared to come up with a compact and faster algorithm to implement proposed problem statement.

This project aims at robustly tracking the lip region based on color and shape information. With this assumption that the lip localization part can be achieved, this project begins with the videos being shot focused basically to the lip area. The lip shape or contour is uniquely described using the lip feature vector which eventually forms part of the lip feature vector accounting to a single syllable. The experimental results have been tested on different frames under constraints of illumination condition and head motion.

1.1 INTRODUCTION

The process of understanding human behaviour is been interesting problem since the generation. The invention of computers and further, imaging algorithms opened up new avenues to automate the concept of understanding the human behaviour. The role computing knowledge in behaviour management has brought a paradigm shift to not only automate the engineering theoretical areas but, also provided insight for Medical area of research.
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

The present project is one specific area of understanding human behaviour since it is known from ages that Face is the index of human attitude which will not only talk about the present emotion but also gives certain amount of information about the incident which might accord with the person before meeting that individual or it might also give information about the opinion of the person on the individual how is communicating or will communicate.

Further, since it would not be possible to understand or bring out the complete details of face since 80-90% of human population will have common features. Thus the in present project, we have considered LIP formation and LIP movement as a parameter to understand the expression of the speaker.

The Concept of knowing what deaf and dumb person speaks is literally a million rupee question for someone like us who could not even understand what someone is speaking when both the communicating individual or Group is blessed with all the parameters required for communication.

The basic motivation of the present project lies in this crux of the problem, the project tries to understand the human speech by capturing the Lip based on imaging algorithms and also a embedding of speech with the image thus creating a training model.

The basic idea was to develop effective system for speech reading application. The input to our present system is the video of a person who is speaking some alphabet which is recorded using a suitable camera. The video thus obtained will be segmented to frames which are used for designing and implementing the experimental models.

Further, the extracted images will be processed using MATLAB ver 7.0 software to correlate every lip image to its equivalent feature vector. The final output of this system is the text display of the alphabet spoken by the speaker.

![Fig 1.1 Basic design of lip motion to text conversion](image-url)
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

Robust and accurate analysis of lip features requires coping with the large variation in appearance across subjects and the large appearance variability of a single subject caused by changes in lighting, pose, and facial expressions. Mouth features play a central role for automatic face recognition, facial expression analysis, lip readings and speech processing. Accurately and robustly tracking lip motion in image sequences is especially difficult because lips are highly deformable, and they vary in shape, color and relation to surrounding features across individuals.

This project aims at robustly tracking the lip region based on color and shape information. With the assumption that the lip localization part has been achieved, this project begins with the videos being shot focused basically to the lip area. The lip shape or contour is uniquely described using the lip feature vector which eventually forms part of the lip feature vector accounting to a single syllable. The results have been tested on different frames under constraints of illumination condition and head motion.

➢ BACKGROUND:

▪ Neural Network

The concept of Computing has created many opening in all fields of engineering but in computer science the relevance of artificial neural networks are computational models inspired by animal central nervous systems (in particular the brain) that are capable of machine learning and pattern recognition. They are usually illustrated as systems of interconnected "neurons" that can compute values from inputs by feeding information through the network.

For example, in a neural network for handwriting recognition, a set of input neurons may be activated by the pixels of an input image representing a letter or digit. The activations of these neurons are then passed on to a weighted and transformed by some function determined by the network's designer, to other neurons, etc., until finally an output neuron is activated that determines which character was read.

Like any machine learning methods, neural networks have been used to solve a wide variety of tasks that are hard to solve using ordinary rule-based programming, including computer vision and speech recognition.
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

1.2 PROBLEM STATEMENT

Sounds are rhythmic nodes created by vocal cords. Each human being has an individual pattern which can discriminate him/her from others. In case of imitation also an individual likes to reproduces a pattern of sound which other person produces with ease.

In the present work, we want to understand the pattern in which a individual creates a sound for expressing a specific letter. Further, a standard pattern is created which will be knowledge base for the comparison of the expressed sound with the knowledge base. Sounds can be made with the lips in the same position. Fast speech, poor pronunciation, bad lighting, faces turning away, hands over mouths, moustaches and beards make lip reading more difficult or even impossible.

The concept of image matching is taken up to understand the expressed sound which makes it a unique process, since it is a understood parameter that irrespective of the individuals vocal cord the movement of lips has to be same to for pronouncing the letters. This idea has been taken in the present project to bring out the different patterns for the pronunciation.

1.2.1 Existing Technology

- As there will be illumination variation during the process of talking few of them talk slowly few speak loudly, Hence there will be difference in the lip motion or movements.
- A lot of research work is being done on lip-reading competing speaker, challenge of multispeaker, presence and absence of lipstick and moustache, front view lip-reading.
- Many techniques have been proposed for lip-reading. Different methods are used to extract features.
- Many features related to characters were studied and recognition techniques based on those features were suggested.

1.2.2 Proposed System

In this project, we are going to convert lip motion features to text conversion where in the input is the static images or real time videos from which we are going to
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

extract frames from these videos and contour the region for lip extraction from which we get the output as text.

As per the knowledge of imaging we feel that when a normal person speaks the number of frames would be around 650, but we have considered 135 frames for the present work.

1.3 THE SOLUTION

The present work is concerned with designing a system to determine the letter spoken by the speaker using a video consisting of lip movements. The solution proposed in this work is to determine vowels of English language spoken by many speakers but of the same region. To provide a tool for deaf and dumb people for better communication.

1.4 LITERATURE SURVEY

The literature survey for lip motion features was performed to make a study on following approaches:
1. Extraction of Grid based motion feature.
2. Contour based motion feature.
3. Lip shape features.

1.4.1 Extraction of Grid based motion feature:

The grid-based motion feature extraction, as described in [4] the dense motion estimation over a uniform grid of size Gx*Gy on the extracted lip region image is first performed. The motion matrices, Vx and Vy, are separately transformed via 2-D-DCT.

Transforming the motion data into DCT domain has two advantages. First, it serves as a tool to reduce the feature dimension by filtering out the high frequency components of the motion signal. Second, DCT de-correlates the feature vector so that the discriminative power of each feature component can independently be analysed.
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

![Diagram of motion estimation process](image)

\[ f_{GRD} = \{ f_x^1, f_y^1, f_x^2, f_y^2, ..., f_x^M, f_y^M \} \]

**Fig.1.2 Grid Based Motion feature**

**1.4.2 Contour based motion feature:**

In the contour-based lip motion representation, only motion vectors computed on the pixels along the extracted lip contour are taken into account and the rest is discarded. In this case, the two sequences of x and y motion components on the contour pixels are separately transformed using one-dimensional DCT Fig. 1.2 depicts the procedure for extraction of contour-based lip motion features as proposed in [2].

**1.4.3 Lip shape features:**

The contour-based lip motion feature vector \( f_{ctr} \) can be fused with lip shape parameters to improve the representation. They used this to denote the lip shape feature vector by \( f_{shp} \). Thus, in the lip shape-based approach the first step requires parameterizing the lip shape with four cubic polynomial and two line segments. In order to assure translation and rotation invariance, we represent the lip shape in terms of horizontal and vertical distances between the sampled points. The concatenation of lip shape parameters with contour-based motion information is illustrated in Figure 1.3 and is as referenced in [5].
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

![Lip Shape features](image)

**Comparison of three methods:**

A detailed comparison of above mentioned alternatives as performed in [1], and as observed in fig 1.4 shows that the best recognition rate is achieved with combination of contour-based motion features with the shape features. On the other hand, performance of contour-based motion features starts to degrade after a certain value of the feature dimension N.

![Comparison of previous methods](image)

**1.5 OBJECTIVES**

To design algorithm that can be used-

- Classify and recognize the vowels with the visual features obtained from the lip images using lip reading techniques.
- To effectively trace the lip contour region.
- To develop a simple system for speech reading applications.
- To create an aiding tool for deaf and dumb people for better communication.
1.6 ADVANTAGES

- A two-way ‘text supported’ video communication for to-and-fro interaction between deaf/dumb and normal people.

In this application there will be communication between two individuals through webcam or any other media. The person who is deaf will see the text on his screen as the other person on the other side will speak. The repetition will apply for the deaf or dumb person speaking on this side who cannot produce audio so the lip movement will recognize the speech spoken.

- An electronic support Tool for professional Lip readers:

This will be helpful in true sense for the professional lip readers. The lip readers who predict the conversation between two individual when the audio of the conversation is not available. This tool will help them to analyse the conversation through movement of the lips.

1.7 LIMITATIONS

- Varying illumination conditions.
- Presence of moustache.
- Selection of feature vector.
- Resizing the images for uniformity.
- Text recognition algorithm.

1.8 ORGANISATION FOR REPORT

The present report is divided into seven major chapters the present chapter brings out the overview of the project, the existing methods and the problem statement. In second chapter we have included software specifications of our project including the functional overview, user characteristics/roles, input/output requirement. Then we have briefly described about project cycle. Third chapter deals with the project description, color model, edge detection, morphology, neural network and block diagram of our project. The Fourth chapter deals with the implementation detail such as the software
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion
tools and coding part. *Fifth chapter* deals with the testing part, different testing techniques like functional, black box testing, then *sixth chapter* deals with the snapshots and outputs of our project for the different inputs. And finally in *chapter seven* we have concluded by indicating scope of the future work.
CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

In this chapter we have included software requirement specification of our project including the functional overview, user characteristic/roles, input requirement. Then we have output requirement which describe the proper output of the project. After that we have briefly described about the project cycle according to which our project work has been done.

2.1 HARDWARE REQUIREMENTS

Hardware requirements needed for the image processing mainly are noted below:

Processor : Intel core2 Duo onwards
RAM : 512 MB RAM (minimum)
Hard Disk : 5 GB (minimum)
Monitor : VGA Compatible
Keyboard : Standard QWERTY keyword
Mouse : Standard mouse

2.2 SOFTWARE REQUIREMENTS

Software requirements needed for the image processing mainly are noted below:

- **OS- Windows family:** we use windows platform as the windows is very user friendly and has very good GUI (graphical user interface). Also windows family support numerous applications required for the development of the project.

- **MATLAB 7.0 and above:** MATLAB is an important and main software required for the lip motion feature to text conversion.

  MATLAB is a high level technical computing language and interactive environment for algorithm development, data visualization, data analysis and numeric computation. Using the MATLAB product, you can solve technical computing problems faster than with traditional programming languages, such as C, C++ and FORTRAN.

  You can use MATLAB in a wide range of application, including signal and image processing, communication, control design, test and measurement, financial modelling.
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extends the MATLAB environment to solve particular classes of problems in these application areas.

MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications and distribute your MATLAB algorithms and applications.

2.3 FUNCTIONAL OVERVIEW

The movement of the articulator organ produces human speech. In speech recognition, there are two types of source of information, acoustic signal and visual signal. Most researches in speech recognition have been done using acoustic signal only. Lip reading, however, has the problem that not all words can be uniquely visualized. For example, words such as ‘ben’, ‘men’ and ‘pen’, are very similar with respect to the lip movements. But even though the visual system is normally not specially designed for decoding and recognizing word information, the lip movements contain enough information for the visual categorization of speech. Several approaches to visual lip reading have been described. Despite the progress that has been achieved, still more research is required to improve recognition accuracy.

In a noisy environment where voice recognition software tends to under-perform, using lip reading approach increases the accuracy. Previous works have tried to compare the two images

By classic image comparison techniques. But those works achieved low accuracy rate. Use of neural network in lip reading technique improves the accuracy rate. In this technique the problem is divided into clusters and four different distances: horizontal and vertical viz. H-1 (horizontal distance between inner lip points), H-2 (horizontal distance between outer lip points), V-1 (vertical distance between inner lip points) and V-2 (vertical distance between outer lip points) calculated. This study used neural networks clustering technology in lip reading applications and its statistical approach based on the most commonly used words and their intensive database comprising of just their distances making the search and processing faster and more accurate as compared to traditional image comparison techniques. This study also compares the similar sounding words and
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion displays each one of them as an option. The results suggest that their neural networks model performs better than the others. Data clustering deals with the problem of classifying a set of N objects into groups so that objects within the same group are more similar than objects belonging to different groups. In this study neural network is used to predict the letter or word spoken by the subject. The video of the word spoken by the subject is taken and is split in to various .jpeg images using Advanced X Video converter, Version 5.0.3 tool.

They have taken minimum of two and maximum of five images for a letter or a word as suited. The lip area on the image is highlighted using the spline curve by marking the control points and then the grid is superimposed on the image so that the distances can be taken accurately. The lip area is cropped from the complete image and then the various distances i.e. H1, H2, V1, V2 are stored in the database. Then these distances are summed and for each image and the resultant sum compare with sums of the relevant images of the particular letter or the word and send for further clustering that takes place with the help of Neuro Solutions using MATLAB. In testing, the test image passed through above procedure and we get the test sum which is to be compared with the original entries of the database. The comparison is done by keeping in consideration a buffer of 25 units for manual error constraints. After the comparison takes place the prediction of the letter or the word done through the classified entries of the clusters.

2.4 INPUT REQUIREMENTS

Input requirements needed for the image processing mainly are briefly noted below:

**Video Input:** The input of our system is video of an individual’s lip motion while speaking an alphabet. The video is captured so as to focus on the lip movement information. The video is taken under certain standardized conditions such as uniform illumination, steady face alignment.

**Frames Extraction:** The video which is taken as input first is then converted into frames. The video can be converted into frames by the method namely: using MATLAB.

**Pre-Processing:** Before the contour extraction method is applied to the original image, it is required that it undergoes a sequence of transformations that eventually highlight the
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

lip region. The pre-processing requirement performs such enhancements and color transforms on each frame image extracted.

2.5 OUTPUT REQUIREMENTS

Output requirements needed for the image processing mainly are briefly noted below:

**Lip Region Extraction**: The region of interest is the lip area. We use the edge detection methods to trace the lip contour. This process thus provides the boundary pixel information constituting the lip contour.

**Feature vector**: The lip contour for computational purposes can be better represented using significant lip properties. These properties can be stacked up to form the Lip Feature Vector. A number of such feature vectors when combined to account for the frames constituting a single alphabet forms the Motion feature Vector.

2.6 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software system or its component. A function is described as a set of inputs.

A user logs onto the MATLAB. Then open the folder containing the code and then run the code. If needed show the path of the video file. Then user runs the specific code then we obtain the frames which are extracted and calculate the motion feature vectors and then compare the values of motion feature vectors with the database using neural network.

2.7 PROJECT CYCLE

Our plan to include following stages in our project cycle:

Fig 2.1: Schematic diagram of project cycle
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

**Requirement Analysis:** The requirements of the implementation of our project are thoroughly analysed. Various requirements to build the project are been listed here and it includes both hardware and software requirements.

The requirements are collected as Input requirements and Hardware and Software requirements. And Functional and Non Functional requirements.

**Design:** After realizing and acquiring the requirement next we move to designing of the project. Model for the proposed method is defined and designed. According to the architecture the system is divided into various modules for easy designing and implementation.

**Module Designing:** The identified modules are being designed according to the working of the proposed system. Various classes are designed according to the modules.

**Implementation:** All the modules are integrated and implemented using C programming language. The code is written in MATLAB.

**Testing:** After the complete implementation the software will be validated and verified.
CHAPTER 3

SYSTEM DESIGN

This chapter deals with the project description which gives brief description of our project, system design color model, edge detection, morphology, neural network and block diagram of our project.

3.1 PROJECT DESCRIPTION

In this project, we are carrying out the lip motion feature to text conversion by robustly tracking the lip region based on color and shape information. With the assumption that the lip localization part has been achieved, this project begins with the videos being shot focused basically to the lip area. The lip shape or contour is uniquely described using the lip feature vector which eventually forms part of the lip feature vector accounting to a single syllable. The results have been tested on 20 different speakers under constraints of illumination condition and head motion.

3.1.1 Color Model

This is also known as colour space. Colour models are different ways in which colour information is stored. The most common and obvious colour model in the RGB colour model. The reason for having more than one colour model is that some operations are easier to implement if we move away from the RGB model.

RGB: An RGB image, sometimes referred to as a true color image, is stored as an $m$-by-$n$-by-3 data array that defines red, green, and blue color components for each individual pixel. An RGB MATLAB array can be of class double, uint8, or uint16. In an RGB array of class double, each color component is a value between 0 and 1. A pixel whose color components are (0, 0, 0) is displayed as black, and a pixel whose color components are (1, 1, 1) is displayed as white. The three color components for each pixel are stored along the third dimension of the data array. For example, the red, green, and blue color components of the pixel (10,5) are stored in RGB(10,5,1), RGB(10,5,2), and RGB(10,5,3), respectively.

NTSC: The brightness component of television signals uses the National Television System Committee (NTSC) color encoding scheme.
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

NTSC uses a luminance-chrominance encoding system invented in 1938 by Georges Valensi. The three color picture signals are divided into Luminance (derived mathematically from the three separate color signals (Red, Green and Blue)) which takes the place of the original monochrome signal and Chrominance which carries only the color information. This allows black-and-white receivers to display NTSC color signals by simply ignoring the chrominance signal.

### 3.1.2 Edge Detection

When we look at an image, we immediately observe 3 basic types of discontinuities in the grey level viz. Points, lines and edge. An edge can be defined as a set of connected pixels that form a boundary between two disjoint regions. Variation of scene features, usually brightness, give rise to edge. In other words, edges are representations of the discontinuities of the scene intensity function. There could be various reasons such as type of material, surface texture, lightning condition, etc., which play an important role in forming these discontinuities.

The purpose of detecting sharp changes in image brightness is to capture important events and changes in properties of the world. It can be shown that under rather general assumptions for an image formation model, discontinuities in image brightness are likely to correspond to:

- Discontinuities in depth,
- Discontinuities in surface orientation,
- Changes in material properties and variations in scene illumination.

The edges obtained from natural images are usually not at all ideal step edges. Instead they are normally affected by one or several of the following effects:

- Focal blur caused by a finite depth-of-field and finite point spread function.
- Penumbral blur caused by shadows created by light sources of non-zero radius.
- Shading at a smooth object.

There are many methods for edge detection, but most of them can be grouped into two categories, search-based and zero-crossing based. The search-based methods detect edges by first computing a measure of edge strength, usually a first-order derivative expression such as the gradient magnitude, and then searching for local directional maxima of the gradient magnitude using a computed estimate of the local orientation of...
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

the edge, usually the gradient direction. The zero-crossing based methods search for zero crossings in a second-order derivative expression computed from the image in order to find edges, usually the zero-crossings of the Laplacian or the zero-crossings of a non-linear differential expression. As a pre-processing step to edge detection, a smoothing stage, typically Gaussian smoothing, is almost always applied.

3.1.3 Morphology

Morphology is the study of the shape and form of objects. Morphological image analysis can be used to perform

- Object extraction
- Image filtering operations, such as removal of small objects or noise from an image.
- Image segmentation operations, such as separating connected objects

Measurement operations, such as texture analysis and shape description. The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries.

- **Erosion:**

  With $A$ and $B$ as sets in $Z^2$, the erosion of $A$ by $B$ is defined as

  $$A - B = \{ z \mid (B) z \subseteq A \} \quad (1)$$

  The erosion of $A$ by $B$ is the set of all points $z$ such that $B$, translated by $z$, is contained in $A$. We will assume that the set $B$ is a structuring element. In the previous example (slide 6), since “$B$ had to be contained in $A$” is equivalent to “$B$ not sharing any common elements with the q g y background”, the erosion can be expressed in the following form:

  $$A - B = \{ z \mid (B) z \cap A = \emptyset \} \quad (2)$$

  where $A^c$ is the complement of $A$ and $\emptyset$ is the empty set. Erosion removes pixels on object boundaries.

- **Dilation:**

  With $A$ and $B$ as sets in $Z^2$, the dilation of $A$ by $B$ is defined as:

  $$A + B = \{ z \mid (B^c) z \cap A \neq \emptyset \} \quad (3)$$
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

This equation is based on reflecting $B$ about its origin, and shifting this reflection by $z$. The dilation of $A$ by $B$ then is the set of all displacements $z$, such that and $A$ overlap by at least one element. Therefore, the dilation can also be expressed as

$$A + B = \{ z \mid (B \hat{)} z \cap A \subseteq A \} \quad (4)$$

As before, we assume that $B$ is a structuring element and $A$ is the set (image object) to be dilated.

**Opening and closing:**

As we’ve seen, dilation expands the components of an image while the erosion shrinks them. 

*Opening* generally smoothes the contour of an object and eliminate thin protrusions.

*Closing* also tends to smooth sections of contours but fusing narrow breaks and long, thin guls and eliminating small holes and filling gaps in the contour.

The opening of a set $A$ by structuring element $B$ is defined as

$$A \ast B = (A - B) + B \quad (5)$$

Therefore, the opening $A$ by $B$ is the erosion of $A$ by $B$, followed by a dilation of the result by $B$.

Similarly, the closing of a set $A$ by structuring element $B$ is defined as

$$A \cdot B = (A + B) - B \quad (6)$$

Therefore, the closing of $A$ by $B$ is the dilation of $A$ by $B$, followed by the erosion of the result by $B$.

Morphological Opening removes small objects from the neighbourhood of the boundary of region of interest in the image. Closing removes small holes from the boundary and smoothenst it.

### 3.1.4 Neural Network

**BACKPROPAGATION:**

An abbreviation for "backward propagation of errors" this is a common method of training artificial neural networks. From a desired output, the network learns from many inputs,

It requires a dataset of the desired output for many inputs, making up the training set. It is most useful for feed-forward networks (networks that have no feedback, or
simply, that have no connections that loop). Backpropagation requires that the activation function used by the artificial neurons (or "nodes") be differentiable.

The back propagation learning algorithm can be divided into two phases: propagation and weight update.

**Phase 1: Propagation:**

Each propagation involves the following steps:

1. Forward propagation of a training pattern's input through the neural network in order to generate the propagation's output activations.
2. Backward propagation of the propagation's output activations through the neural network using the training pattern target in order to generate the deltas of all output and hidden neurons.

**Phase 2: Weight update:**

For each weight-synapse follow the following steps:

1. Multiply its output delta and input activation to get the gradient of the weight.
2. Subtract a ratio (percentage) of the gradient from the weight.

This ratio (percentage) influences the speed and quality of learning; it is called the *learning rate*. The greater the ratio, the faster the neuron trains; the lower the ratio, the more accurate the training is. The sign of the gradient of a weight indicates where the error is increasing, this is why the weight must be updated in the opposite direction.

Repeat phase 1 and 2 until the performance of the network is satisfactory.

### 3.2 PROCESSING STEPS

The below block diagram show the brief flow of our project where in it consist of seven blocks these are as follows: video input, frame extraction, pre-processing block, lip region extraction, feature extraction, discrimination analysis, text display.

**Video Input:**

This block is the input of our system, which contains the video of an individual’s lip motion while speaking an alphabet. The video is captured so as to focus on the lip movement information. The video is taken under certain standardized conditions such as uniform illumination, steady face alignment.
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

**Frame Extraction:**

The video obtained is then converted into frames. The video can be converted into frames by method namely: MATLAB.

**Pre-processing:**

Before the contour extraction algorithm is applied to the original image, it is required that it undergoes a sequence of transformations that eventually highlight the lip region. The preprocessing block performs such enhancements and color transforms on each frame image extracted.

![Fig 3.1 Processing Steps for text display](image)

**Lip Region Extraction:**

The region of interest is the lip area. We use the edge detection methods to trace the lip contour. This block thus provides the boundary pixel information constituting the lip contour.

**Feature Vector:**

The lip contour for computational purposes can be better represented using significant lip properties. These properties can be stacked up to form the Lip Feature Vector. A number of such feature vectors when combined to account for the frames constituting a single alphabet forms the Motion feature Vector.
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

**Neural Network:**

Multilayer neural networks are capable of performing just about any linear or nonlinear computation. This block is thus responsible for providing a solution to the nonlinear problem of relating the feature vectors to the corresponding class or alphabet.

### 3.3 DATABASE DESIGN

The speech recognition task can be formulated as either an open-set or a closed-set identification problem. In the closed-set identification problem, a reject scenario is not defined, and an unknown observation is classified as belonging to one of the R registered pattern classes. In the open-set problem, the objective is, given the observation from an unknown pattern, to find whether it belongs to a pattern class registered in the database or not; the system identifies the pattern if there is a match and rejects otherwise. Hence, the problem can be thought of as an R+1 class identification problem, including a reject class.

In this phase, a database addressing a limited vocabulary speech-reading application is created. 20 individuals are put to speak certain number of alphabets and this data will be stored and processed to get motion feature vector corresponding to every alphabet uttered for each individual separately.

The speech-reading database $Ds$ constructed includes 100 different phrases, i.e., $R = 100$. Each phrase is an alphabet with three repetitions. The number of source speakers for this is 20. The $Ds$ database is partitioned into two disjoint sets $Ds1$ and $Ds2$, one for training and the other for testing.

- Each alphabet was spoken for a repetition of 3 times.
- The person was asked to close lips after every alphabet for differentiation.
- Camera’s video resolution = 640x480.
- Frames/sec = 30 fps.
- Average time for 5 alphabets (a, e, i, o, u) = 22 sec.
3.4 FLOW CHART

Capture test video

Load video in MATLAB

Extract frames

Select every 5th frame

Apply gamma correction on rgb image

Convert rgb to ntsc color model

Select saturation component

Convert image into binary

Apply opening and closing

Extract lip region by applying edge detection function
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

Fig 3.2 Flow chart
3.5 DATA FLOW DIAGRAM

Designing with the Data Flow Diagrams involves creating a model of the system. The entities and attributes are a model of the states of the system. Processes model the rules of a System. The stimuli and response are modelled by Data Flows. All of these models are combined into one graphic model called a Data Flow Diagram.

Fig 3.3 Data Flow Diagram

The Data flow diagram shows that user can do different actions such as login to the system and write the codes to the generated questions. Administrator can create the tests and questions. Administrator view the user lists and evaluate the c programs.

**Data flow diagram (DFD)** is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

3.6 SEQUENCE DIAGRAM

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

![Sequence Diagram](image)

**Fig 3.4 Sequence Diagram**
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

1. Load video to Matlab
2. Extract frames
3. Invalid video
4. Select 5th frames
5. Invalid frame selected
6. Apply gamma correction
7. Convert RGB to NTSC color model
8. Select saturation component
9. Convert to binary image
10. Extract lip region by edge detection algorithm
11. Apply morphological operations
12. Creating boundary box over the lip contour
13. Get lip properties/features
14. Invalid properties/features
15. Average of all lip features
16. Apply neural network on average feature
17. Display text
18. Invalid text display
CHAPTER 4

IMPLEMENTATION

4.1 SOFTWARE TOOLS USED

MATLAB 7.0 and above: MATLAB is an important and main software required for the processing of the images.

MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis and numeric computation. Using the MATLAB product, you can solve technical computing problems faster than with traditional programming languages, such as C, C++ and FORTRAN.

4.2 IMPLEMENTATION DETAILS

In the coding part we have several modules the main modules of lip motion feature to text conversion are Image acquisition and enhancement and color mapping, Morphological operation and binarization of images, motion feature vector for single alphabet and averaged feature vector, Neural network.

4.2.1 Coding

function varargout = guidetemplate0(varargin)
% GUIDETEMPLATE0 MATLAB code for guidetemplate0.fig
% GUIDETEMPLATE0, by itself, creates a new GUIDETEMPLATE0 or raises the existing singleton*.
% H = GUIDETEMPLATE0 returns the handle to a new GUIDETEMPLATE0 or the handle to the existing singleton*.
% GUIDETEMPLATE0('CALLBACK',hObject,eventData,handles,...) calls the local function named CALLBACK in GUIDETEMPLATE0.M with the given input arguments.
% GUIDETEMPLATE0('Property','Value',....) creates a new GUIDETEMPLATE0 or raises the existing singleton*. Starting from the left, property value pairs are
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

% applied to the GUI before guidetemplate0_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to guidetemplate0_OpeningFcn via varargin.
% Begin initialization code

% Begin initialization code

% Begin initialization code

---
Executes just before guidetemplate0 is made visible.

function guidetemplate0_OpeningFcn(hObject, eventdata, handles, varargin)
    global vid1;
    % This function has no output args, see OutputFcn.
    % hObject    handle to figure
    % eventdata    reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    % varargin    command line arguments to guidetemplate0 (see VARARGIN)
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

% Choose default command line output for guidetemplate0
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes guidetemplate0 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

function varargout = guidetemplate0_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

varargout{1} = handles.output;

% --- Executes on button press in Compute.
function Compute_Callback(hObject, eventdata, handles)
    global play;
    clc;
    handlesactivex1.URL = play;
    inpt=mmreader(handlesactivex1.URL);
    cd('D:\final year project\Karthik\Frames')
    otptfol = fullfile(cd, 'framesAkashI');
    if ~exist(otptfol, 'dir')
        mkdir(otptfol);
    end
    frames=inpt.numberOfFrames;
    frameswritten=0;
    y=zeros(frames);
    for frame=1:frames
        y=read(inpt,frame);
        otptBaseFileName = sprintf('%3.3d.jpg', frame);
        outputFullFileName = fullfile(otptfol, otptBaseFileName);
        imwrite(y, outputFullFileName, 'jpg');
        frameswritten = frameswritten + 1;
    end
    progressIndication = sprintf('Wrote %d frames to folder "%s"',frameswritten, otptfol);
    disp(progressIndication);
    set(handles.codestatus1,'String',progressIndication);

    cd('D:\final year project\Karthik\Frames\framesAkashI')
    imagefiles = dir('*.jpg');
    nfiles = length(imagefiles);   % Number of files found
    z=5;
    k=0;
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

$$l=1; \ g=0; \ u=1; \ v=0;$$

$$imat=0; \ op=0; \ cl=0; \ e=1; \ f=1; \ bound=0;$$

\[
\text{for } i=1:nfiles \\
\quad \text{if } \text{mod}(i,z)==0 \\
\quad \quad k=k+1; \\
\quad \quad j=i; \\
\quad \text{end} \\
\text{if } k~==0 \\
\quad \text{if } l<=k \\
\quad \quad \text{currentfilename} = \text{imagefiles}(j).\text{name}; \\
\quad \quad \text{currentimage} = \text{imread(currentfilename);} \\
\quad \quad \text{images}{l} = \text{currentimage}; \\
\quad \quad l=l+1; \\
\quad \text{end} \\
\text{end} \\
\text{end}
\]

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

$$y=1;$$

\[
\text{for } m=1:k \\
\quad \text{hgamma} = \text{video.GammaCorrector}(1.3, \ 'Correction', \ 'gamma'); \\
\quad \text{x1}=\text{images}{m}; \\
\quad \text{x1}=\text{padarray}(x1, [24 24]); \\
\quad \text{y1} = \text{step(hgamma, x1}); \\
\quad \text{t} = \text{rgb2ntsc}(y1); \\
\quad \text{r1} = \text{t}(:,:,3); \\
\quad \text{l} = \text{graythresh}(r1); \\
\quad \text{bw1} = \text{im2bw}(r1,l); \\
\quad \text{r} = \text{bw1}; \\
\quad \text{bo} = \text{imopen}(r, \text{strel('diamond',7)}); \% 8 4
\]
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

```matlab
bg = imclose(bo, strel('disk', 20)); % 20
ed = edge(bg);

bwc = bwmorph(ed, 'shrink', Inf);
bw = bwareaopen(bwc, 400);  % remove object less than 400 pixels
bwe = bwmorph(bw, 'clean');
im = imfill(bwe, 'holes');

bo2 = imopen(r, strel('disk', 9));
bg2 = imclose(bo2, strel('disk', 10));

x2 = im2double(rgb2gray(x1));
x3 = x2 + bwe;

[B, L] = bwboundaries(im, 'noholes'); % im
[B9, L9] = bwboundaries(bg2, 'noholes');

c = length(B(:));  c9 = length(B9(:));
stats = regionprops(im, 'Area', 'Image');  % measure area of the lips
stats9 = regionprops(bg2, 'Area', 'Image');
if c ~= 0
    g = g + 1;
    for n = 1: c9
        area9(n) = (stats9(n).Area);
    end

[mlen9, ind9] = max(area9);
si9 = stats9(ind9, 1).Image;
se = strel('disk', 5);
si92 = imdilate(si9, se);
% subplot(1, 2, 2); imshow(si92);
```

Dept. of CSE, GMIT, Davangere
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

\[
[B92,L92] = bwboundaries(si92,'holes');
\]
\[
c92=length(B92(:));
\]
\[
bound(g)=c92;
\]
\[
if g==1
\]
\[
if c92== 1
\]
\[
cl=1; close1=1;
\]
\[
else
\]
\[
imat(e,f)=g;
\]
\[
f=f+1;
\]
\[
op=1;
\]
\[
end
\]
\[
end
\]
\[
if g>1
\]
\[
if c92== 1
\]
\[
if cl== 0
\]
\[
imat(e,f)=g ; %f+
\]
\[
e=e+1; f=1;
\]
\[
end
\]
\[
cl=1; op=0;
\]
\[
end
\]
\[
if c92>1
\]
\[
if op==1
\]
\[
f=f;
\]
\[
else
\]
\[
op=1;
\]
\[
imat(e,f)=g-1;
\]
\[
f=f+1;
\]
\[
end
\]
\[
imat(e,f)=g;
\]
\[
f=f+1; cl=0;
\]
\[
end
\]
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

for n=1:c
    [mlen,ind]=max(stats(n).Area);
    boundary = B{n};
    area = stats(ind).Area;               % obtain the area calculation corresponding to label 'k'

    si=stats(1,1).Image;
    si2=imfill(si,'holes');

    if m==1
        [m1 n1]=size(si2);
        resize=imresize(si2,[150 250]);
        nm=150; nn=250;
        rx = 150/m1; ry = 250/n1;
    end

    [m2 n2] = size(si2);
    nm = m2 * rx; nn = n2 * ry;
    resize=imresize(si2,[nm nn]);

    [B1,L1] = bwboundaries(resize,'noholes');
    boundary1 = B1{1};
    delta_sq1 = diff(boundary1).^2;
    perimeter1 = sum(sqrt(sum(delta_sq1,2)));   % estimate of the object's perimeter
    stats1 = regionprops(resize,'Area','Perimeter'); % obtain the area calculation corresponding to label 'k'

    area1 = stats1(ind).Area;
    metric1 = 4*pi*area1/perimeter1^2;
    metric_string1 = sprintf('%2.3f',metric1);
%    text(resize(1,2)-50,resize(1,1)+13,metric_string1,'Color','r',...
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

```matlab
% 'FontSize',14,'FontWeight','bold');
pff(m,1)=metric1; %1. form factor
pa(m,1)=area1; %2. area

pdist_h(m,1)=nn; %3. horizontal dist
pdist_v(m,1)=nm; %4. vertical dist

width=nn/2; height=nm;
rect=[1 1 width height];
resize2 = imcrop(resize, rect);

[B2,L2] = bwboundaries(resize2,'noholes');
boundary2 = B2{1};

j12=B2{1,1}(1,1); i12=B2{1,1}(1,2);
[j22,in2]=min(B2{1,1}(:,1)); i22=B2{1,1}(in2,2);
[i32,in3]=max(B2{1,1}(:,2)); j32=max(B2{1,1}(in3));
[j42,in4]=max(B2{1,1}(:,1)); i42=max(B2{1,1}(in4,2));

p1=[i12 j12]; p2=[i22 j22];
p3=[i32 j12]; p4=[i42 j42];
pt1=[i12 i32]; pt2=[i12 i22]; pt3=[j12 j12]; pt4=[j12 j22];

ptheta1(m) =(atan2(abs(det([p3-p1;p2-p1])),dot(p3-p1,p2-p1)) * 180/pi);
%5. upper angle
ptheta2(m) =(atan2(abs(det([p3-p1;p4-p1])),dot(p3-p1,p4-p1)) * 180/pi);
%6. lower angle

end
end

disp('Feature Extracted');
progressIndication = 'Feature Extracted';
```
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

set(handles.codestatus2,'String',progressIndication);

% % % % % % % % % % % % % % % % % % % %
% % % % % % % % % % % % % % % % % % % %
{am an}=size(imat);
count(1:am)=an;
for i=1:am
  for j=1:an
    if imat(i,j)==0
      count(i) = count(i) - 1;
    end
  end
index = imat(i,1);
index1 = imat(i,count(i));

ii1=1; ii2=1; ii3=1; ii4=1; ii5=1; ii6=1;
upff=0; upa=0; updist_h=0; updist_v=0; uptheta1=0; uptheta2=0;
for k = index:index1
  upff(ii1)=pff(k);
  ii1=ii1+1;
  upa(ii2)=pa(k);
  ii2=ii2+1;
  updist_h(ii3)=pdist_h(k);
  ii3=ii3+1;
  updist_v(ii4)=pdist_v(k);
  ii4=ii4+1;
  uptheta1(ii5)=ptheta1(k);
  ii5=ii5+1;
  uptheta2(ii6)=ptheta2(k);
  ii6=ii6+1;
end
apff1(i,1) = mean(upff)*10;
apa1(i,1) = mean(upa)/100;
aph1(i,1) = mean(updist_h);
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

\[
\begin{align*}
\text{apv1}(i,1) &= \text{mean}(\text{updist}_v); \\
\text{aptu1}(i,1) &= \text{mean}(\text{uptheta}1); \\
\text{aptd1}(i,1) &= \text{mean}(\text{uptheta}2);
\end{align*}
\]

end

\text{progressIndication} = 'Motion feature extracted';

% set(handles.codestatus3,'String',progressIndication);

\text{disp('Motion feature extracted');}

\text{feature} = \text{cat}(2,\text{apff1},\text{apa1},\text{aph1},\text{apv1},\text{aptu1},\text{aptd1});

cd('D:\final year project\Karthik\neural_net')

load net_6682.mat;

\text{feature} = \text{feature}';

\text{Y} = \text{round}(\text{sim}(\text{net},\text{feature}));

\text{lny}=\text{length}(%Y);

\text{Y}

\text{for} i=1:\text{lny}

\text{switch}(\text{Y}(i))

\text{case} 1

\text{x}(i)= \text{char}(97);

\text{case} 2

\text{x}(i)= \text{char}(101);

\text{case} 3

\text{x}(i)= \text{char}(105);

\text{case} 4

\text{x}(i)= \text{char}(111);

\text{case} 5

\text{x}(i)= \text{char}(117);

\text{otherwise}

\text{x}(i)= \text{char}(45);

\text{end}

\text{end}

\text{x}=\text{x}';
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

disp(x);
% t=uitable('position',[528 145 115 300]);
% set(t,'Data',x);
set(handles.alphabetop,'String',x);
% hObject    handle to Compute (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes during object creation, after setting all properties.
function video_CreateFcn(hObject, eventdata, handles)
% hObject    handle to video (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: popupmenu controls usually have a white background on Windows.
%    See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: listbox controls usually have a white background on Windows.
%    See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

```matlab
set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc  &&  isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes during object creation, after setting all properties.
function codestatus1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to codestatus1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% --- Executes on selection change in output.
function output_Callback(hObject, eventdata, handles)
% hObject    handle to output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: contents = cellstr(get(hObject,'String')) returns output contents as cell array
%        contents{get(hObject,'Value')} returns selected item from output
% --- Executes during object creation, after setting all properties.
function output_CreateFcn(hObject, eventdata, handles)
% hObject    handle to output (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

```matlab
% Hint: listbox controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

### 4.2.2 PROGRAMMING TECHNIQUES

- **Database design process:**

  It is fair to say that database play a critical role in almost all areas where computers are used, including business, electronic commerce, engineering, medicine, law, education and library. A database is collection of a related data.

  A database has the following implicit properties:

  - A database represents some aspect of the real world, sometimes called the mini-world or the Universe Of Discourse (UOD) changes to the mini world are reflected in the database.
  
  - A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.

  - A database is designed, built and populated with data for a specific purpose. It is an intended group of users and some preconceived application which these users are interested.

  Database Management System (DBMS) is a collection of programs that enables users to create and maintain a database. DBMS is a general-purpose software system that facilities the process of defining, constructing, manipulating and sharing database among various users and application. Defining a database involves the specifying the data types, structures and constraints of the data to be stored in the database. The database definition or descriptive information is also stored in the database in the form of dictionary; it is called Meta data constructing the database is the process of storing the data on the storage medium that is controlled by the DBMS.
CHAPTER 5

TESTING

This chapter deals with the discussion of the scope of the project, different testing techniques like functional, black box testing.

Testing is a general approach to the testing process rather than a method of devising particular system or component testing. Different testing strategies may be adopted depending on the type of system to be tested and the development process used. Once source code has been generated software must be tested to uncover as many errors as possible before delivery to the customer. Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and code generation.

5.1 SCOPE

The system should not be tested as signal, monolithic unit. The testing process should therefore proceeding stages where testing is carried out incrementally in conjunction with system implementation. Therefore an iterative one with information being fed back from the last stages to earlier part of the process.

5.2 FUNCTIONAL TESTING

Testing Objectives

- Testing is a process of executing program with the intent of finding the error.
- A good test case is one that has a high probability of finding an as yet undiscovered error.
- Successful test is one that uncovers an as yet discovered error.

Testing Principles

- All testing should be traceable for customer requirement.
- Tests should be planned large before testing begins.
- Testing should being “in the small” and progress towards “in the large”.
- To be most effective, an independent third party should conduct testing.
To build a model for implementing automated lip reading which involves Lip motion
feature to text conversion

Testability

Software testability is simple how a computer program can be tested. The
characteristics lead to testing software as follows.

- Operability
- Observability
- Controllability
- Decomposability
- Simplicity
- Stability
- Understand ability

5.3 BLACK BOX TESTING

Black box testing relies on the specification of the system or component, which is
being tested to drive the test case. The system is a black box whose behaviour can only be
determined by studying its inputs and the related outputs.

In this testing, the application is taken through the series of inputs which are both
valid and invalid. Our application is provided with some invalid data and the performance
is tested. For invalid input the application is provided with suitable error checking and
messages. System is also checked with some valid inputs for which the application is
responding remarkably. System is working properly for all possible valid and invalid
inputs accordingly, hence tested as black box.

5.4 WHITE BOX TESTING

White box testing is verification technique software engineers can used to
examine if their code works as expected. White-box testing is testing that takes into
account the internal mechanism of a system or component. White-box testing is also
known as structural testing, clear box testing, and glass box testing. The connotations of
“clear box” and “glass box” appropriately indicate that you have full visibility of the
internal workings of the software product, specifically, the logic and the structure of the
code.
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

Using the white-box testing techniques as software engineer can design test cases that

(1) Exercise independent paths with in a module or unit.
(2) Exercise logical decisions on both their true and false side.
(3) Execute loops at their boundaries and within their operational bounds.
(4) Exercise internal data structures to ensure their validity.

With white-box testing, we run the code with pre-determined input and check to make sure that the code produces pre-determine output. Each time we write a code module, we write test cases for it based on the guidelines.

Basis path testing is a means for ensuring that all independent paths through a code module have been tested. An independent path is any path through the code that introduces at least one new set of processing statements or a new condition. Basis path testing provides a minimum, lower-bound on test cases that need to be written. To introduce the basis path method, we will draw a flow graph of a code segment.

5.5 DATA FLOW TESTING

In data flow based testing, the control flow graph is annotated with the information about how the program variables are defined and used. Different criteria exercise with the varying degrees of precision how a value assigned to a variable is used along different control paths. A reference notation is a definition-use pair, which is a triple of (d, u, V) such that V is a variable, d is a note in which V is defined and u is a node in which V is used. There exists a path between d and u in which the definition of V in d is used in u.

5.6 FAILURE TEST CASES

As with black-box test cases, you must think about the kinds of things users might do with your program. Look at the structure of your code and think about every possible way a user might break it. We need to be smart enough to think of your particular code and how people might outsmart it (accidentally or intentionally).

We must augment our test cases to handle these cases.

Some suggestions follows:
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

> We must look at every input into the code you are testing. Do we handle each input if it is incorrect?
> We must look at code from a security point of view. Can a user overflow a buffer, causing a security problem?
> We must look at every calculation. Could it possible to create an overflow? Have we protected from possible division by zero?

The importance of software testing and its impact on software cannot be underestimated. Software testing is a fundamental component of software quality assurance and represents a review of specification, design and coding. The greater visibility of software system and the cost associated with the software failure are motivating factors for planning, through testing. It is not uncommon for a software organization to spent 40% of its effort on testing.

### 5.7 TEST RESULTS

We analysed the lip motion feature by giving various images of lips and finding their lip properties such as form factor, area, horizontal distance, vertical distance, up and down angles as input. Output is measured in terms of the class value and associated with the class value the vowels are displayed.

The averaged motion feature vector along with the respective class is provided as the training data for neural network.

Two sets of class values used: \{0.2:: 1\} and \{1::5\}, Selected class values: \{a=1, e=2, i=3, o=4, u=5\}

<table>
<thead>
<tr>
<th>Form Factor</th>
<th>Area</th>
<th>Horizontal distance</th>
<th>Vertical distance</th>
<th>Up angle</th>
<th>Down angle</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.720381</td>
<td>24037</td>
<td>250</td>
<td>150</td>
<td>42.02627</td>
<td>30.96379</td>
<td>1</td>
</tr>
<tr>
<td>0.720292</td>
<td>27288</td>
<td>261.7521</td>
<td>162.9496</td>
<td>44.02898</td>
<td>34.8085</td>
<td>2</td>
</tr>
<tr>
<td>0.740887</td>
<td>28963</td>
<td>268.1624</td>
<td>168.3453</td>
<td>43.51599</td>
<td>28.90918</td>
<td>3</td>
</tr>
<tr>
<td>0.375121</td>
<td>32359</td>
<td>261.7521</td>
<td>208.2734</td>
<td>65.22486</td>
<td>43.65738</td>
<td>4</td>
</tr>
<tr>
<td>0.772757</td>
<td>29962</td>
<td>262.8205</td>
<td>167.2662</td>
<td>49.08562</td>
<td>30.12028</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5.1 Training Data
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

➢ REGRESSION PLOT

![Regression Plot](image)

**Fig 5.1 Regression Plot**

A regression plot attempts to show the effect of adding another variable to a model already having one or more independent variables.

➢ Simulating Test Data:

• The net, consisting of hidden layer weights and biases providing the best result for training and testing of neural network is saved.

• We use this net for recognition of classes for the test data provided.
CHAPTER 6

RESULTS

6.1 SNAPSHOTs OF THE PROJECT

6.1.1 GUI:

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders—just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

MATLAB GUIs can be built in two ways:

1) Using GUIDE (GUI Development Environment), an interactive GUI construction kit.
2) Create code files that generate GUIs as functions or scripts (programmatic GUI construction).

Most GUIs generate or use data that is specific to the application. These mechanisms provide a way for applications to save and retrieve data stored with the GUI. With GUI data:

- You can access the data from within a callback routine using the component's handle, without needing to find the figure handle.
- You do not need to create and maintain a hard-coded name for the data throughout your source code.
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

![GUI: Video processing](image1.png)

**Fig 6.1: GUI: Video processing**

The fig 6.1 depicts to select the filename of the video to be played on clicking compute button where the video is of a person speaking vowels.

![GUI: Recognized text](image2.png)

**Fig 6.2: GUI: Recognized text**

The fig 6.2 depicts the code status such as feature extracted and motion feature extracted and the text such as vowels are displayed.
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

6.2 OUTPUT FOR DIFFERENT SAMPLE DATA

6.2.1 Lip region extraction:

Before the contour extraction algorithm is applied to the original image, it is required that it undergoes a sequence of transformations that eventually highlight the lip region.

6.2.2 Color models:

The very first step to study an image is to observe the R, G, and B components of the color image.

![Fig 6.2.1 Red Component](image1)
![Fig 6.2.2 Green Component](image2)
![Fig 6.2.3 Blue Component](image3)
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

A visual observation shows that the Green component of the image presents the highest differentiation between the lip and skin portion. On the other hand, the skin and the lip color essentially lie in the red band of the RGB colormap Thus it required an interplay of different colormap systems and their relation with the RGB colormap.

6.2.3 NTSC colormap images:

Fitting solution to the project requirement: Saturation component of the NTSC colormap.

\[ Q=(0.211 \times R) - (0.523 \times G) + (0.312 \times B) \]

The above equation maps the R, G and B components of the original image to its saturation component.
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

6.2.4 Image enhancement:

Prior to color mapping:

The R, G, B components individually were processed before applying the mapping function.

Post color mapping:

Further processes can be aided by using some enhancement techniques applicable to grayscale images.

6.2.5 Gamma correction:

The ‘gamma correction’ forms the first part of the enhancement requirement applied prior to color mapping. The function Video.GammaCorrector non-linearly transforms the input image pixel values to the output.

6.2.6 Post mapping, ‘imadjust’ is used
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

\text{imadj}(I) \text{ maps the intensity values in grayscale image } I \text{ to new values in } J \text{ such that } 1\% \text{ of data is saturated at low and high intensities of } I. \text{ This increases the contrast of the output image } J.

After highlighting the necessary lip portion, the image is converted to its \textit{binary equivalent}.

![Original image](image1.png)  ![Binary Equivalent](image2.png)

**Fig 6.2.11 Original image**  **Fig 6.2.12 Binary Equivalent**

### 6.3 DIFFERENT PHASES OF PROCESSING

<table>
<thead>
<tr>
<th>Image</th>
<th>Stage</th>
<th>Process</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image 1" /></td>
<td>1</td>
<td>From video by Selecting a Frame</td>
<td>To Understand the Lip Motion</td>
<td>A Color image with the size acceptable in capturing the Lip Portion</td>
</tr>
<tr>
<td><img src="image2.png" alt="Image 2" /></td>
<td>2</td>
<td>Performing Saturation</td>
<td>To Extract the feature</td>
<td>The Lip formation or Lip Contour</td>
</tr>
</tbody>
</table>
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Converting to Binary Image</td>
<td>The image contains the Feature properties resembling the LIP feature are obtained</td>
<td>A portion Acceptable to Edge detection</td>
</tr>
<tr>
<td>4</td>
<td>Opening Operation</td>
<td>To obtain the smoothing effect</td>
<td>To get the clear lip portion or Location</td>
</tr>
<tr>
<td>6</td>
<td>Next level of Smoothing</td>
<td>To get higher level of LIP Component</td>
<td>Removes the Unwanted Noise</td>
</tr>
<tr>
<td>7</td>
<td>Edge detection</td>
<td>To get the Boundary line of the required Feature</td>
<td>A clear discrimination between the object of reference and unwanted features</td>
</tr>
<tr>
<td>8</td>
<td>Morphological operation</td>
<td>To eliminate unwanted portion in the image</td>
<td>A clear picture of required image</td>
</tr>
</tbody>
</table>
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Filling Operation</td>
<td>To fill the image region</td>
</tr>
<tr>
<td>10</td>
<td>Overlapping image</td>
<td>The original image and the edge detected images are overlapped</td>
</tr>
<tr>
<td>11</td>
<td>Display value of form factor</td>
<td>The calculated value of form factor is displayed</td>
</tr>
<tr>
<td>12</td>
<td>Indication of points</td>
<td>Points are marked to get the distance between left to right and top to bottom extremities</td>
</tr>
</tbody>
</table>

Table 6.1 Different phases of processing
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

CHAPTER 7

CONCLUSION

In this project, we have investigated different lip shape extraction algorithms to decide over the current approach involving initial RGB to NTSC color map conversion to provide acceptable discrimination between the lip portion and the skin area. The various possible challenges like uneven illumination and moustache conditions were overcome under the pre-processing part of our algorithm.

Under the Lip Motion recognition part, the tool used to discriminate among classes is the neural network. Study of results indicate an accuracy of only satisfactory levels. The task at hand is to implement other multi-valued logic based models and compare them to come up with the best suitable model.

The present project could be more appropriate in case of deaf people on one hand are unable to adhere to the voices, while on the other hand the dumb people fail to effectively express themselves owing to their vocal disability. Hence, we can aid the deaf and dumb by providing them with a device which does not involve voice but only lip motion and hence ease the communication.

7.1 SCOPE OF FUTURE WORK

Though this work recognizes the vowels well enough, there are plenty of space to improve the recognition accuracy. In future the accuracy can be increased by using some other features and English phoneme or visemes can be identified following this method or it can be used to recognize a word or sentence. Lip segmentation can be improved by Recognition of vowels from facial images using lip reading Technique School of introducing geometric or color based automatic lip segmentation in future.

This method can be improved from speaker and language dependent to speaker independent lip-reading system. In future we can work with:

- All 26 alphabets and numbers, words
- Robust image registration
- Product to aid deaf and dumb
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

ABBREVIATIONS

- National Television System Committee - NTSC
- Red, Green and Blue - RGB
To build a model for implementing automated lip reading which involves Lip motion feature to text conversion

BIBLIOGRAPHY

7. www.mathworks.in
8. www.imageprocessingblog.com