TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

VOICE COMMAND BASED OBJECT RECOGNIZING ROBOT USING SPEECH AND IMAGE FEATURE EXTRACTION

(EG777EX)

By:

Bishal Heuju (069BEX409)

Bishal Lakha (069BEX410)

Dipkamal Bhusal (069BEX414)

Kanhaiya Lal Shrestha (069 BEX 417)

A PROJECT SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULLFILLMENT OF THE REQUIREMENT FOR THE BACHELOR'S DEGREE IN ELECTRONICS & COMMUNICATION ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

LALITPUR, NEPAL

(AUGUST, 2016)

LETTER OF APPROVAL

The undersigned hereby certify that they have read, and recommended to the Institute of Engineering for acceptance, this project report entitled "Voice Command Based Object Recognizing Robot using Speech and Image Feature Extraction" submitted by Bishal Heuju, Bishal Lakha, Dipkamal Bhusal and Kanhaiya Lal Shrestha in partial fulfilment of the requirements for the Bachelor's Degree in Electronics & Communication Engineering.

_____

Supervisor
Dr. Nanda Bikram Adhikari
Deputy Head
Department of Electronics & Computer Engineering,
Institute of Engineering, Pulchowk Campus,
Tribhuvan University, Nepal

_____

Internal Examiner
Dr. Basanta Joshi
Assistant Professor
Department of Electronics & Computer Engineering,
Institute of Engineering, Pulchowk Campus,
Tribhuvan University, Nepal

_____

External Examiner
Mr. Om Thapa
Chief Technical Officer
Vianet Communications Pvt. Ltd.

_____

Dr. Nanda Bikram Adhikari
Deputy Head
Department of Electronics & Computer Engineering,
Institute of Engineering, Pulchowk Campus,
Tribhuvan University, Nepal

_____

Dr. Dibakar Raj Pant
Head
Department of Electronics & Computer Engineering,
Institute of Engineering, Pulchowk Campus,
Tribhuvan University, Nepal

DATE OF APPROVAL: August, 2016

COPYRIGHT

ACKNOWLEDGEMENTS

We would like to express our deepest gratitude to the Department of Electronics and Computer Engineering, Central Campus Pulchowk for assigning major project as a part of the academic curriculum, which granted us an opportunity to choose our final year project. This project has helped us harness our knowledge in the practicality of the courses we studied in the engineering. During the course of this project, we studied the different underlying principles of speech processing and object recognition and the radical changes that these are bringing upon in our life. Implementation of algorithms and principles, embedded system characteristics and design issues, studied earlier in course were illustrated and hence clarified. We got an opportunity to apply the textual knowledge into real world problems.

We would like to thank out project supervisor Dr. Nanda Bikram Adhikari for providing us his insights in different fields and guiding us all the way. His encouragement and suggestion have played a vital role in timely implementation of the project.

ABSTRACT

This project provides a proof-of-concept of an assistant robot that bear two most important human senses-hearing and vision. The robot's task is defined to recognize the command 'identify' and then identify the object shown to it. On the command 'follow', the robot needs to track the object. The robot makes use of feature extraction from speech and images to recognize the given signal. The features from speech are extracted as Mel Frequency Cepstrum Coefficients (MFCC) and the features from the images are extracted as Speeded Up Robust Features (SURF). All the computational works are carried out in the main processor, Raspberry Pi. The robot can also be controlled by an android application.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF NOTATIONS AND ABBREVIATIONS

ASR	Automatic Speech Recognition
BoF	Bag of Features
DC	Direct Current
DCT	Discrete Cosine Transform
DTW	Dynamic Time Warping
FFT	Fast Fourier Transform
HMM	Hidden Markov Model
LPC	Linear Predictive Coding
MFCC	Mel Frequency Cepstral Coefficients
PCB	Printed Circuit Board
STT	Speech to Text
SURF	Speeded-Up Robust Features
USB	Universal Serial Bus

1. INTRODUCTION

1.1 Background

The human civilization is constantly thriving forward in the field of technology. We're told time and again by technologist and scientists that future has arrived. There is now no such thing as science fantasy. Whatever we've imagined with the new world is now and will be a reality. Anthropomorphic robots are one among such creations that are promised to be our everyday friend very soon. With imminent revolution in technology and robotics, scientists and engineers have predicted that very soon those robots will change the way we live our day-to-day life. Robots will recognize their owners by appearance and voice and understand the natural mode of communication. They will lift the burden of our work and take the roles of our cleaner, courier, nurse and assistants. Amazon, Google, Microsoft and Apple are currently working on the advanced robotic assistants with their own variations and they will certainly be the major players in the field as the outcomes arrive. All of these robots will have the necessary intelligence and sensing to fulfil the request of simple human commands for example "Bring me the coffee" or "Go and feed my pets".

In a simplistic sense, the home robots can be divided into three categories based on their purpose: (1) robots helping with dull and tedious household chores (2) robots taking on new roles in home (e.g. Education, entertainment) (3) robots in assisted living communities. All these kinds of robots will roam freely around our homes in whatever way they fit driven by a model and learning from the surrounding.

A lot of tech companies have come up with assistant robots in the market recently and as mentioned above, big silicon-valley companies are investing in the field. The steady development in the field of robotics has surely made the possibilities of service robots seem even closer and more compelling. Robots, that will have immense impact on our lives in the future, are slowly making their way today. Those service-robots are and will be human-alike. To communicate with the surrounding and the person, these robots need to be bear human perception, bear human-like senses. Of the five major senses of human, we rely a lot on hearing and vision for communication. To see the world around and judge what's happening and what needs to be done, these assistant robots need to bear human-like vision ability. They have to be able to identify the objects around. And to understand

the command of the user, the robots should be able to understand what we speak. They need to understand what the words of the command actually mean. Hence the prerequisites to develop an assistant robot is to implement speech and vision in machine.

This project is a proof-of-concept of interfacing speech and vision into a machine that can act as an assistant robot.

1.2 Objectives

The following are the major objectives of this final year project:

i. To develop an embedded system
ii. To implement speech recognition in machine
iii. To implement computer vision in machine
iv. To interface speech recognition and computer vision in the embedded system
v. To work in team and improve coordination and team working ability

1.3 Scope of Project

Service robots will have a huge impact on human life. A number of service robots have already been developed while some are in development. These service robots can be used as assistants in offices and homes. Speech recognition and object recognition feature on such robots has opened a wide area of applications.

Our project's key concept lies in implementing the fundamental idea of such robots; to recognize a voice command and recognize objects and/or follow commands. We have developed a simple three wheeler robot as our assistant that identifies the object showed by the user upon given command 'IDENTIFY' and track the object upon providing the command 'FOLLOW'. This robot can be controlled by an android application as well. The computational tasks are carried out by the Raspberry Pi.

Our system is speaker independent and isolated speech recognition with generic object recognition.

2. LITERATURE REVIEW

i. A thesis report was submitted to Centre for Automation Research, UMD Institute for Advanced Computer Studies (UMIACS) on a navigation and object location device for the blind. The goal of this project was to create navigation system for the blind. This project's key part was to identify the command of the visually impaired person and locate the object asked by him/her.

ii. A paper was published on International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering on the second volume, Issue 8, August 2013 that described a project on isolated speech recognition using the Mel Frequency Cepstral Coefficient and Dynamic Time Warping. The implementation of speech recognition was almost similar to our project.

iii. The paper published by Kumar and Chauhan, 2014, explains about a robot that works with the input given voice command. The system consists of three parts: voice recognition system, central controller system and the robot itself. The voice recognition system acts as an interface between the man and robot. The robot receives the voice as input signal and processes it and forward to the central controller and finally transmit the signal to the robot to execute the command.

iv. A paper called 'Object recognition using region detection and feature extraction' was published by E. Jauregi, E. Lazkano and B. Sierra from Robotics and Autonomous Systems Group, Computer Science and and Artificial Intelligence Department, University of Basque Country, Donostia. This paper deals with region detection and feature extraction method for object recognition which were implemented for door handle identification during robot navigation and extended to the signal identification problem.

3. THEORETICAL BACKGROUND

3.1 Speech Recognition

Speech is the most common mode of communication among human beings. It is very natural and efficient method of exchanging information. Speech carries information regarding gender, emotion and identity of the speaker. Since as early as 1930s, scientists have been researching ways and means to make computers able to record, interpret, and understand human speech.

Speech recognition is, in basic form, the process of converting a spoken language to words by computers or computerized devices. It is also known as “automatic speech recognition (ASR)”, “computer speech recognition” or “speech to text (STT)”. The common application of speech recognition can be found in car systems (installing an audio command based system to replace manual works, like play a radio station, call somebody), similar uses in office and home enabling a smart assistant, in military system, and in robotics (robotic assistant). Its applications also include voice user interfaces such as voice dialling, call routing, domotic (home automation) appliance control, search, etc. People often get confused over speaker recognition and speech recognition. Speaker recognition refers to identifying the speaker rather identifying what they said. Speech recognition on the other hand deals with identifying the words. It can be speaker dependent or speaker independent. Speaker recognition can be used to authenticate or verify the identity of a user.

Speech recognition can be classified into isolated word recognition and continuous speech recognition. In the isolated word recognition, speech is uttered in isolated words whereas in latter, speech is uttered continuously, thus making the recognition harder. The continuous speech recognition systems can further be classified as connected word recognition and conversational speech recognition. The former recognizes each word but has a limited vocabulary whereas the latter focuses on understanding the sentences and has a large vocabulary.

3.1.1 Speech recognition performance:

Most of the recognition systems depend on a form of training that acclimatizes the system either to the speech of a particular individual, or to a group of individuals. The necessity of

the system whether to operate for single or multiple users impacts the training method. Of all the forms of speech recognition system, a single word, pre-trained, single user system, is the simplest system to design with reasonable accuracy. Any system with higher level of functionalities then aforementioned will incur either design complexity, reduced accuracy or both.

Any speech recognition system will face the problems of background noise, microphone placement, and gain and so on. Having trained a system to a particular voice so that accuracy levels of over 90% can be achieved, an issue as simple as changing microphone placement or the addition of even quite low levels of background noise, could easily reduce accuracy by as much as 20%. So, a recognition system designed to work with headset-mounted microphone will perform better than capturing speech from a transducer placed away from the speaker.

3.1.2 Practical speech recognition:

All of the practical ASR systems have a lot of same process involved. The input signal is first cleaned up by a pre-processing system before a feature vector is extracted. The pre-processing may include filtering, windowing, normalizing, or segmentation.

After the pre-processing, features are extracted. The different methods to extract feature from a speech signal are explained below.

In the simplest of system, these features are compared in turn to a large set of stored features (an acoustic model). A distance measure (e.g. Euclidean) is computed to detect if they match. This can also be performed by hidden Markov model, assigning the probability of utterances. Beyond this, a language model can also be applied to weight the probabilities of the top few matches based upon their adherence to language rules.

With a very large acoustic model or language model, the searches can be really slow. So the size of the feature vector must be minimized where possible.

3.1.3 Approaches to automatic speech recognition:

There are generally three approaches to speech recognition theory, namely:

i. The acoustic-phonetic approach
ii. The pattern recognition approach

iii. The artificial intelligence approach [1]

The first approach, acoustic-phonetic is based on the theory of acoustic phonetics (speech sounds). It postulates that there exist finite, distinctive phonetic units in spoken language and that those unites are broadly classified based on their characteristics as a set of properties manifested in the speech signal, or spectrum over time. This was one of the earliest approach to speech recognition that was based on finding speech sounds and providing appropriate labels to them. Even though the acoustic properties are highly variable both with the speaker and with neighbouring sounds it is assumed that the rules governing the variability are straightforward and can be readily learned by a machine.

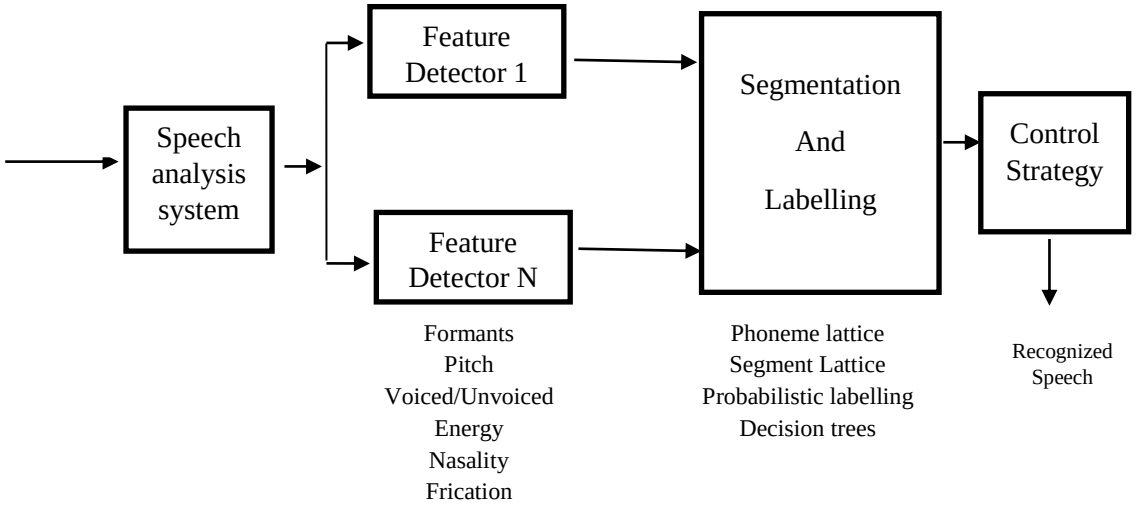


Figure 3.1 Block diagram of acoustic-phonetic speech recognition system

The first step in the acoustic phonetic approach is a spectral analysis of the speech that is combined with a feature detection. It converts the spectral measurement to a set of features that can describe the broad acoustic properties of the phonetic units. Then we carry out segmentation and labelling. In this process, the speech is first segmented into the stable acoustic regions and then one or more phonetic labels are attached to each segmented region. The last step in this approach attempts to determine a valid word from the phonetic label sequences. This approach has not been widely used commercially.

In pattern recognition approach, the speech patterns are directly used without feature determination as in acoustic phonetic approach. It uses a well formulated mathematical framework and establishes a consistent speech pattern representation, for reliable pattern comparisons. This approach consists of two steps-training and comparison. Speech

patterns that serve as the knowledge to the system are brought in through training steps. If enough versions of the patterns are collected in the training set, the training procedure should be able to characterize the acoustic properties of the pattern. In the comparison phase, the new speech's pattern is directly compared to the existing patterns in the training set. This method is extensively used now in speech recognition systems and we too have used pattern recognition for our project because of following advantages:

i. Simple to use
ii. Robustness and invariance to different speech vocabularies, users, feature sets
iii. Proven high performance

This approach is described in detail in proceeding sections.

Artificial intelligence based approach is a hybrid of both the approaches described above. It exploits the ideas and concepts of acoustic phonetic and pattern recognition methods. This method tries to imitate the human behaviour in visualisation, analyse and decision making of features into a machine. Learning and adapting over time is the key feature of this approach.

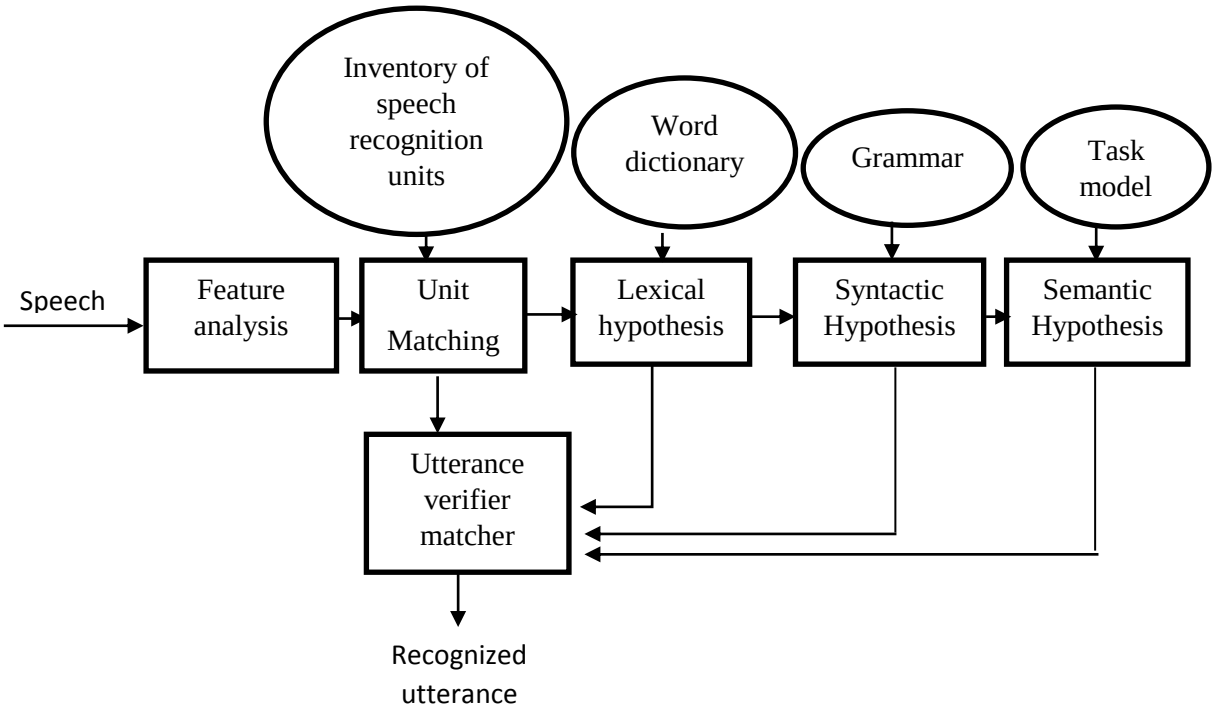


Figure 3.2 A top-down approach to knowledge integration for speech recognition

3.1.4 Introduction to pattern recognition technique:

Based on the speech pattern representation either in the form of speech template or statistical model, this technique is further classified into:

i. Template approach
ii. Stochastic approach

The idea of template matching is simple. We store the speech patterns as reference patterns, after devising some mathematical algorithms. Using the same algorithms, the features are extracted of the word uttered by the speaker and these are compared with the stored pattern to check a hit or a miss. Feature extraction algorithms any include Linear Predictive Coding, Discrete Fourier transform, Mel-frequency Cepstrum analysis. And for feature matching, dynamic time warping, vector quantization etc. can be used. These are described in detail in proceeding sections. In this method, the pre-recorded templates are fixed. So, to model the different variations in speech signal, we need to store a lot of templates for a single word. This becomes impractical. Template training and matching can become expensive as the vocabulary size increases. Template matching is also tediously speaker dependent and continuous speech recognition is not possible using this approach.

Pattern recognition exploits the fact that if the same words are uttered, then similar acoustic patterns appear. However, this is not always true. Some parts of pattern vary from occurrence to occurrence. This is especially prominent in connected words or continuous words recognition. Hence, if the recognizer can undertake the variability of the patterns then its ability to distinguish between words is likely to increase. Statistical model of ASR does just that. It uses a different way of defining the degree of fit between a word and speech data, as an alternative to the ‘cumulative distance’ as in pattern recognition. This is based on the notion of probability. Hidden Markov Model is the most popular stochastic model right now. This represents the current state of the art. In this model, instead of distance from a template, a *posteriori* probability that the model of the word could have produced the observed set of features vectors is calculated. The main drawback of statistical models is that they must make a priori modelling presumption, which is liable to be inaccurate, restrict the system’s performance.

3.1.4.1 Pattern Recognition-Template approach:

When a person utters a word, the word can be considered as the sequence of phonemes,

linguistic units. Because of the inevitable articulation, the acoustic patters associated with individual phones overlap in time. [2] So, for every word a speaker utters, the acoustic pattern appears in a very complicated form. But, if the same word is repeated time and again, the phonetic relationship and the mechanism of producing sound in the speaker, will generates similar pattern. There will be difference in a number of factors like the speed with which the word is spoken, pitch might be different, one word may be spoken more precisely than the other, etc. But there will be similarities between the word-pattern at a short-time window.

This method thus relies on comparing a template stored on the machine related to a certain word and comparing the speech against it while recognizing speech. Any incoming word is compared in turn with all the words stored in the machine and the one that is similar is the correct word.

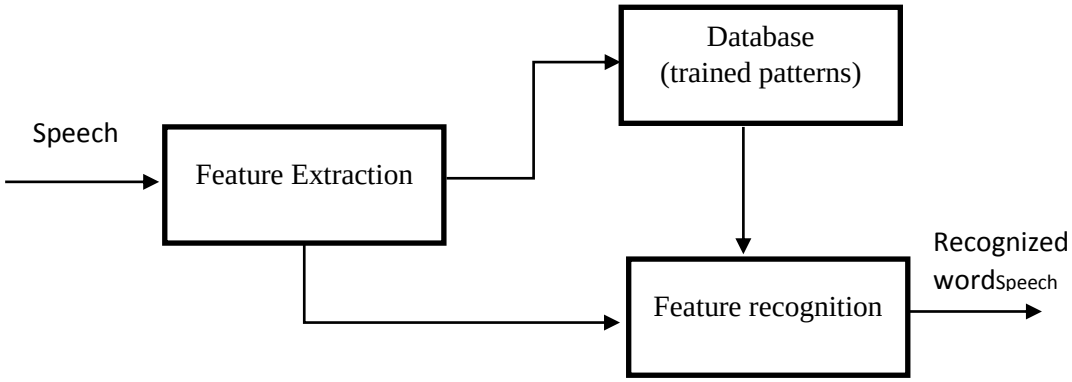


Figure 3.3 Block diagram of pattern recognition-template matching

3.1.4.1.1 Feature extraction methods:

As explained above pattern recognition consists of two important steps: generation and storing patterns in the machine and then comparison. It is thus very essential to select an appropriate feature extraction method to rightfully represent the acoustic speech signal. The different kinds of feature extraction methods are explained below:

1. Linear Predictive Coding (LPC):

This is a method for signal source modelling in speech it is often used a format extraction method however this is widely used in speech recognition system as well. This is one of the most powerful method for encoding quality speech at a low bit rate. The basic idea is that a specific speech sample at the current time is approximated by the linear combination of past examples.

2. Pure Fast Fourier Transform:

Direct use of vector coefficients of FFT power-spectrum of the speech signal is also possible as speech feature for speech recognition system. Pure FFT spectrum carries information about the speech signal than mimicking the human auditory system. If higher sampling rates is used, extra information is located in the higher frequency bans which are not considered salient in speech recognition.

3. MFCC:

MFCC is based on human hearing perceptions which cannot perceive frequencies above 1KHz. In other words, MFCC is based on the known variation of the human ear's critical bandwidth with frequency. [3] MFCC has two types of filter that are spaced linearly at low frequency below 100 Hz and logarithmic spacing above 1000Hz.

In order for the computer or computerized machine to process the signal it has to be digitized first. So, first the continuous-time signal is sampled and quantized. It gives a discrete signal. According to Nyquist-Shannon sampling theorem, a continuous time signal band-limited to a frequency f*max* needs to sample with a sampling frequency of at least 2fmax. This will allow the reconstruction of the signal. The sampling frequency and the feature vector size has a direct effect on recognition accuracy. So, a suitable sampling frequency must be decided. Since human speech has a relatively low bandwidth (mostly between 100 Hz-8 KHz), a sampling frequency of 16 KHz is sufficient for speech recognition tasks.

The overall process of MFCC can be explained in following steps: [4]

a. Pre-emphasis: In this process, the signal is passed through a filter that emphasizes higher frequencies i.e. to increase the signal energy content at higher frequencies. To perform pre-emphasis, we choose some value α between .9 and .1 and each value in the signal is re-evaluated using the formula:

$$y[n] = x[n] - \alpha * x[n-1] \tag{3.1}$$

b. Windowing and Framing: Though an audio signal constantly changes over time, we assume that on very short time-duration it doesn't change much i.e. remain statistically stationary. So we frame the signal into 20-40 milliseconds frames. Experiments have proven that the frames shorter than this duration won't give us enough samples for a reliable spectral estimate and if it is longer, the signal changes too much throughout the

frame. To obtain the frames, we multiply the speech signal with a windowing function. The windowing function divides the signal into a sequence of partial signals, with overlapping.

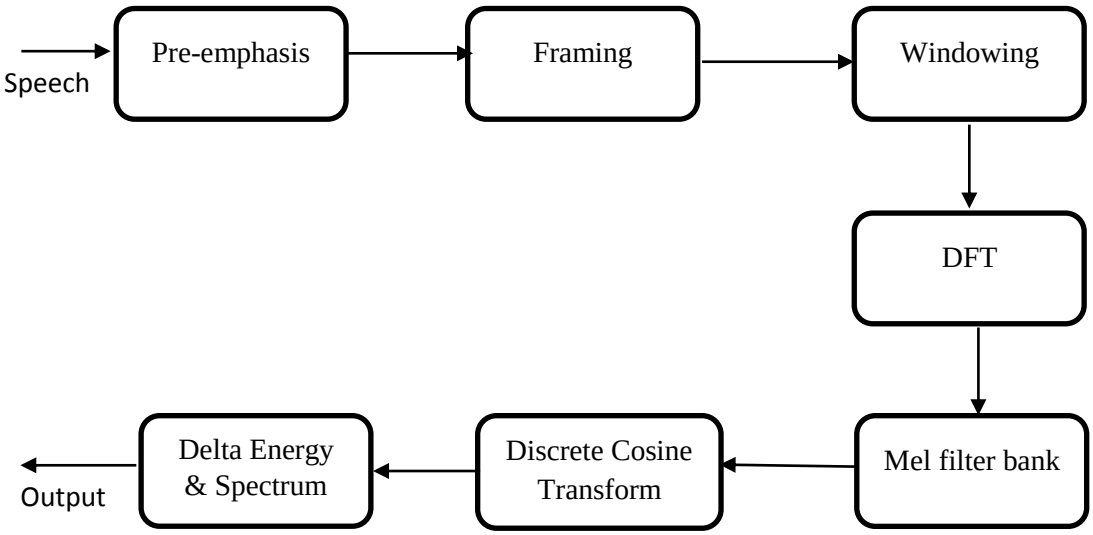


Figure 3.4 MFCC block diagram

c. Discrete Fourier Transform: The obtained signal is then converted into frequency domain by computing the discrete Fourier transform and calculating the power spectrum of each frame. This is motivated by the human cochlea which vibrates at different spots depending on the frequency of the incoming sounds. Depending on the location in the cochlea vibrates, different nerves fire informing the brain that certain frequencies are present.

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi kn}{N}} \quad \text{for } k = 0, 1 \ldots N-1 \tag{3.2}$$

where N is the number of sampling points in the speech frame

For implementation, Fast Fourier Transform (FFT) which is a variation of Discrete Fourier Transform optimized for speed, is used.

d. Mel-frequency spectrum: The frequency range obtained from FFT is very wide. But humans don't discern the difference between two closely spaced frequencies. This becomes pronounced as the frequencies increase. So, we compute Mel-frequency spectrum. For this the spectrum of the signal is filtered with N different band-pass filters and the power of each frequency band is computed. This filtering mimics, the human auditory system as explained earlier. In this filter bank, the first filter is very

narrow and gives an indication of how much energy exists near 0 Hz. As the frequencies get higher the filters get wider since the variations aren't concerned. The Mel scale tells us how to space the filter banks. Research suggests that too few and too many band-pass filters have a negative impact on the classification performance and that overlapping rectangular shaped filters achieve a better performance compared to the triangular shaped filter [5].

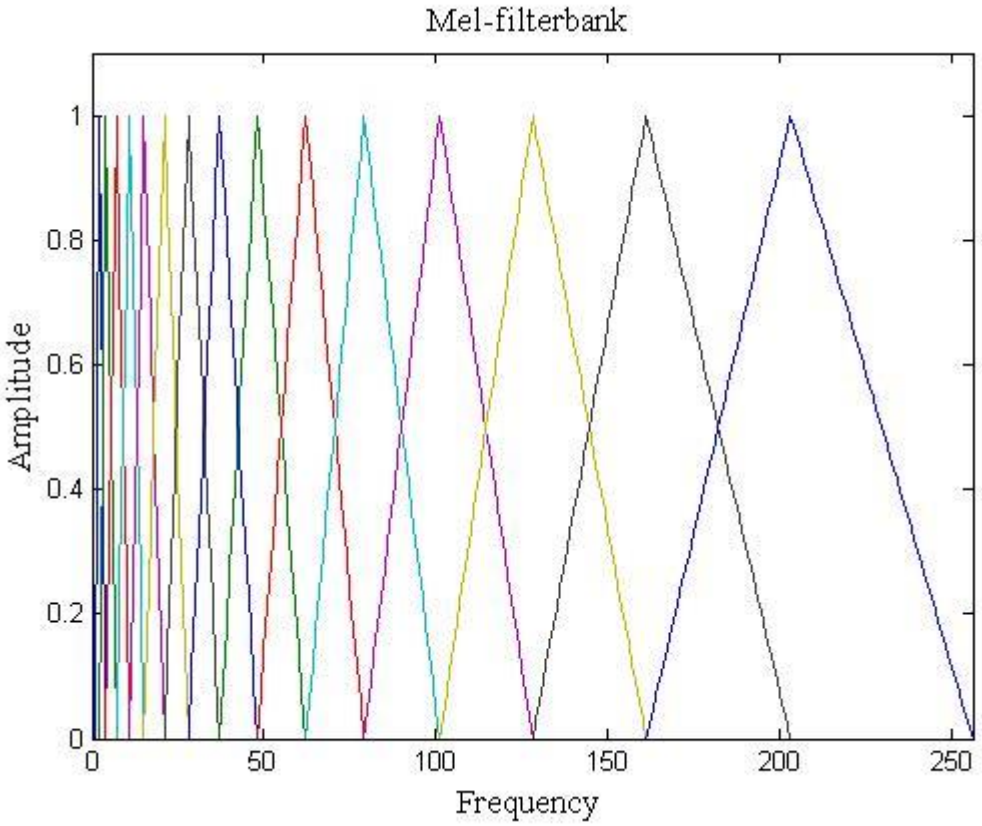


Figure 3.5 Mel Filter Bank

As shown in the Figure 3.5, a set of triangular filters are used to compute the filter spectral component so that the output of the process approximates to a Mel scale. Each filter's magnitude response is triangular in shape, and equal to unity at the centre frequency and decreases linearly to zero at the centre frequency of two adjacent filters.

The Mel scale related perceived frequency, or pitch, of a pure tone to its actually measured frequency. We humans are better at discerning small changes in pitch at low frequencies than they are at high frequencies. Incorporating this scale makes the features match more closely to what humans hear.

The formula for converting from frequency to Mel scale is:

$$M(f) = 1125 \ln(1 + \frac{f}{700}) \quad (3.3)$$

After the computation of Mel frequency spectrum, we take logarithm of them. This is because humans don't hear loudness on a linear scale. Generally, to double the perceived volume of a sound we need to put 8 times as much energy into it. This means that large variations in energy may not sound all that different if the sound is loud to begin with. This compression operation makes the features match more closely what humans actually hear.

e. Discrete Cosine Transform (DCT): The final step is to compute the DCT of the log filter bank energies. Since the filter banks are overlapped, the log of the energy we calculate are highly correlated to each other. The DCT de-correlates these energies and compresses the data size. The result of this conversion is indeed Mel Frequency Cepstrum Coefficient. The set of these coefficients is called acoustic vectors. Thus each speech signal is transformed into a sequence of acoustic vector. [6]

We use DCT-II, which can be implemented by the formula:

$$\text{Y[k]} = 2 * \sum_{n=0}^{N-1} x[n] * \cos\left(\pi * k * \frac{2*n+1}{2*N}\right) \text{ k=0, 1, 2....N} \quad (3.4)$$

To get an orthogonal matrix, we multiply again Y[k] by a scaling factor f,

$$f=\sqrt{1/(4 * N)} \text{ if k=0} \quad (3.5)$$

$$f=\sqrt{1/(2 * N)} \text{ otherwise} \quad (3.6)$$

3.1.4.1.2 Feature comparison:

How speech patterns are compared to determine their similarity is a key point in speech recognition system. Depending upon the nature of the recognition system and means from which the reference patterns are generated, pattern comparison can be done in a variety of ways. If the recognizer is an acoustic-phonetic recognizer where every analysis frame is classified according to a set of features, then the decision on absence or presence of these vowel features is made based on the values of acoustic parameters (e.g. Formants, energy in spectral bands, duration) either using a decision tree or a mathematical rule.

In the pattern based approach, speech is directly represented by the time sequence of

spectral vectors as obtained from the front-end spectral analysis. Let us consider a test pattern, T that consists of spectral frames over the duration of speech as

$$T= \{t1, t2, t3…..t\} \quad (3.7)$$

Similarly let R be a reference pattern such that

$$R= \{r1, r2, r3…..r\} \quad (3.8)$$

The goal of the pattern comparison is to determine the dissimilarity or distance between T and R, in order to verify that the reference pattern has minimum dissimilarity with the test pattern and the spoken word is the same word as reference word.

In the speech comparison phase, the reference feature vector is compared with the test sample. Since speech varies a lot either due to speed or timing, the durations of the word may change and this may not result in the matching even of the same sound. Dynamic Time Warping method, one that is widely used in the pattern recognition template, plays a vital role in such condition. This algorithm is used for measuring similarity between two-time series which may vary in time or speed.

Dynamic Time Warping is a time series alignment algorithm developed originally for speech recognition. It aims at aligning the two sequences of features vectors by warping the time axis iteratively until an optimal match between the sequences is found. According to Wikipedia, “In time series analysis, dynamic time warping (DTW) is an algorithm for measuring similarity between two temporal sequences which may vary in time or speed. For instance, similarities in walking patterns could be detected using DTW, even if one person was walking faster than the other, or if there were accelerations and decelerations during the course of an observation.”

Consider two sequences of vectors,

$$\text{Sequence } X= x1, x2, x3… \quad (3.9)$$

$$\text{Sequence } Y= y1, y2, y3… \quad (3.10)$$

These two sequences can be arranged in grid as shown in the Figure 3.6.

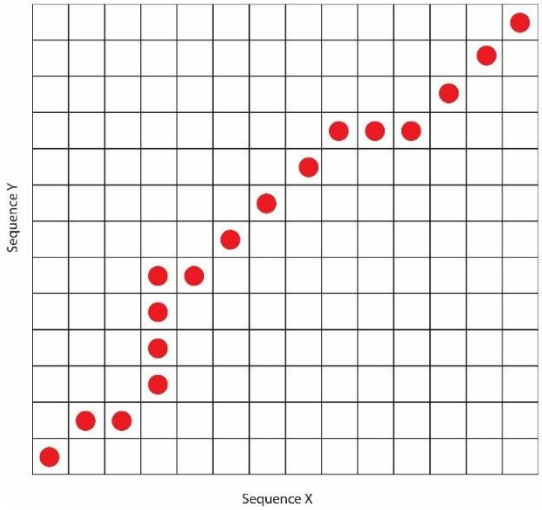


Figure 3.6 Plot of sequences X and Y

Inside each cell, a distance measure can be placed comparing the corresponding sequences elements. In order to find the best match, one need to find a path through the grid which minimizes the total distance between them. To do this, all possible routes through the grid must be found and for each the overall distance must be calculated.

The number of possible paths thorough the grids could be explosive. So there are some constraints on the warping function:

i. Monotonicity: The alignment path doesn't go back in time.
ii. Continuity: The alignment path doesn't jump in time index
iii. Boundary conditions: The boundary starts at the bottom left and ends at the top right
iv. Warping window: A good alignment path is unlikely to wander too far from the diagonal

Let us consider two signals X and Y with following features as stated below:

$$X= (1, 1, 2, 3, 2, 0) \quad (3.11)$$

$$Y= (0, 1, 1, 2, 3, 2, 1) \quad (3.12)$$

The plot of the two signals is given below:

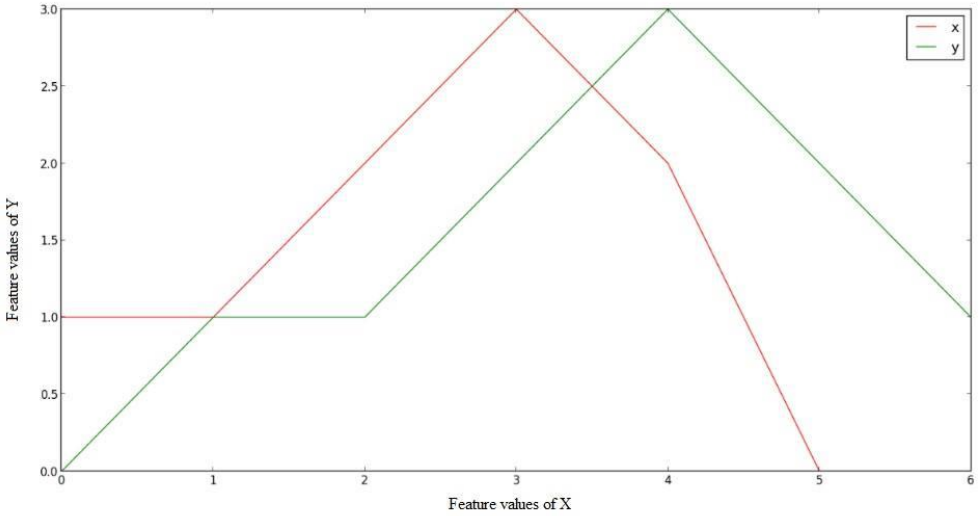


Figure 3.7 Plot of two signals X and Y

We can see in the graph that both the signals show somewhat similar behaviour; they both have a peak and around it, they slope downwards. They also vary in their speed and duration. Now using the DTW, we find how similar they actually are. We find the shortest possible route between them. For this, we find out the distance between all pairs of points in the twos signals assumed above. Lesser distances imply that these points can be matched together.

Using the Euclidean distances between the pair of points, we get the 'distance' matrix as:

```
array([
        [ 1, 1, 4, 9, 4, 0],
        [ 0, 0, 1, 4, 1, 1],
        [ 0, 0, 1, 4, 1, 1],
        [ 1, 1, 0, 1, 0, 4],
        [ 4, 4, 1, 0, 1, 9],
        [ 0, 0, 1, 4, 1, 1]
    ])                                                    (3.13)
```

Before finding the optimal path let's visualize the distance matrix of above:

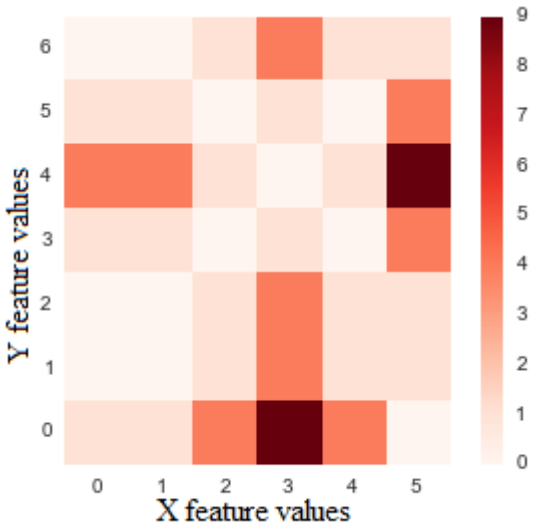


Figure 3.8 Visualization of distance matrix

The diagonal entries reveal the lowest distances in the Figure 3.8 above.

Now we find the path of minimum distance between that starts at (0, 0) and ends at top-right corner. We use the constraints stated as earlier to convert the problem into a dynamic programming.

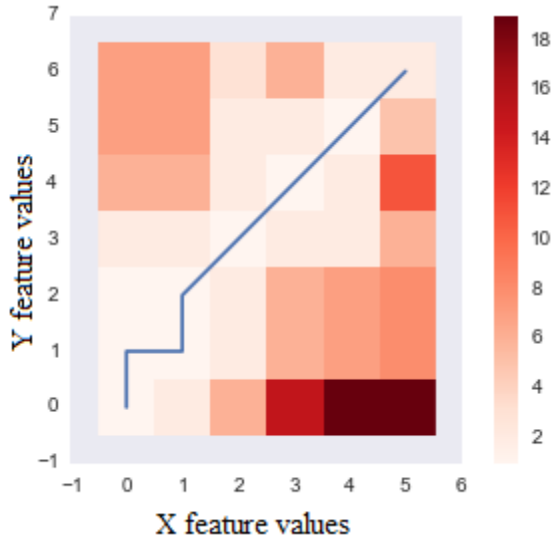


Figure 3.9 Optimum path between the sequences

The above plot shows the optimum warping path that minimizes the sum of distance along the path. [7]

From the calculations, the distance between these two signals is obtained as Figure 3.9 which shows the level of similarity between the two signals.

3.2 Object Recognition

Among five senses, vision is undoubtedly the one that man has come to depend upon above all others, and indeed the one that provides most of the data he receives. Not only do the input pathways from the eyes provide megabits of information at each glance but the data rates for continuous viewing probably exceed 10 megabits per second (Mbit/s). However, much of this information is redundant and is compressed by the various layers of the visual cortex, so that the higher centres of the brain have to interpret abstractly only a small fraction of the data. [8]

Another feature of the human visual system is the ease with which interpretation is carried out. We see a scene as it is—trees in a landscape, books on a desk, widgets in a factory. Human carries out these activities very easily and recognize wide range of objects, despite of variation in luminosity, size, shape, orientation, almost instantaneously. And not only recognition, but extraction of information from visual input is a simple task for human. But same is not true for computers. Emulating object recognition ability of human by computers is still a challenge, and many researches are being carried out in order to achieve that feat.

Various methods have been invented to mimic human's visual perception capabilities. In the field of object recognition several techniques have been purposed and applied. Some of them are discussed below:

a. Appearance based Method

Appearance based method include two phases. In first phase, a model is constructed from a set of reference images. It includes the appearance, orientation and illumination of image. In second phase, part of input image is extracted by segmentation of whole image. The recognition system then compares it with the reference image. [9]

b. Geometry-based Methods

In Geometry- based methods, the information about the objects is represented explicitly. The recognition is then interpreted by deciding whether a given image is a projection of the known (usually 3D) model of an object. Generally, two representations are needed: one to represent object model, and another to represent the image content. To facilitate finding a match between model and image, the two representations should be closely related. In the ideal case there will be a simple relation between primitives used to describe the model and

those used to describe the image. [9]

c. Recognition as a Correspondence of Local Features

In this method, objects are represented by a set of local features, which are automatically computed from the training images using various algorithms. The learned features are then organized into a database. When recognizing a test image, local features are extracted in a similar fashion as in the training images. Similar features are then retrieved from the database and the presence of objects is determined in the terms of the number of local correspondences. [9]

These methods are examples of specific object recognition. These methods are incapable of recognizing wide ranges of objects of same class. So a more general approaches are required. Bag of feature is one of many such approaches.

In Bag of feature model, each image is represented as a histogram of visual words. They come from Bag of word method used for information retrieval in text. In this model, an image is considered to be a document and patches of it are considered words. [10] The Bag of feature involves following steps:

3.2.1 Key point detection and description

3.2.1.1 Key-points Detector

It involves the selection of distinct (interesting) points on an image, such as corners, blobs, and T junctions. Key-points are extracted from both ‘train image’ and ‘test image’. One of the important properties of key-point detection is its repeatability, which mean same physical points can be found under different viewing condition.

3.2.1.2 Description

It involves representation of neighbourhood of every key-point by a feature vectors. The descriptor assists in search of similar element on an image. So the descriptor should be discriminative enough to distinguish between features of the object stored in the database. In addition, it should be invariant to (at least to some extent) to variation in object appearance. Descriptor is calculated in both train image and test image for object recognition.

There is a wide variety of detectors and descriptors currently in use. Due to speed and

robustness, **SURF** was used for object recognition in this project.

SURF

SURF is an acronym for 'Speeded Up Robust Feature'. SURF consists both detector and descriptor. In SURF, input image is analysed at different scales in order to guarantee invariance to change in scale. Then the detected interested points are provided with rotation and scale invariant descriptor. SURF combines Hessian-Laplace region detector with its own gradient orientation-based feature descriptor. [11] [12]

a. Key-point Detection in SURF

The approach used to find keypoints in SURF is Hessian-Laplace method. This leads to use of integral images, which reduces computational time drastically.

b. **Integral Image**

Integral image is the summed area table which value at (x, y) is the sum of all pixels above and to the left.

Mathematically,

$$I_z(x,y) = \sum_{i=0}^{i\leq x} \sum_{j=0}^{j\leq y} I(i,j) \tag{3.14}$$

The top left pixel value in the integral image is 0, and bottom-rightmost pixel of the integral image is thus the sum of all the original pixels for value. Using integral image value of any rectangular sum can be computed efficiently. It takes only three additions and four memory accesses to calculate the sum of intensities inside a rectangular region of any size while when integral image is not used, more additions and memory access are required.

c. Hessian matrix based interest points

The Hessian matrix is used to find interest points. It comprises second order partial derivatives. The determinant of Hessian matrix expresses the extent of the response and is an expression of the local change around the area. The hessian matrix is given as:

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma), & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \tag{3.15}$$

Where $L_{xx}(x,\sigma), L_{xy}(x,\sigma), L_{yy}(x,\sigma)$ are the convolution of the Gaussian second derivative $\frac{\delta^2}{\delta x^2} g(\sigma)$ with the image I in point x. The Gaussian kernels used for the hessian

matrix must be discretized and cropped before we can apply them. The SURF algorithm approximates these kernels with rectangular boxes, box filters which makes it possible to calculate the approximated convolution effectively for arbitrarily sized kernel utilizing the integral image.

$$Det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \tag{3.16}$$

The approximated and discrete kernels are referred to as D_{yy} for $L_{yy}(x,\sigma)$ and D_{xy} for $L_{xy}(x,\sigma)$. When using the approximated kernels to calculate the determinant of the Hessian matrix - we have to weight it with *w*, this is to assure the energy conservation for the Gaussians. The *w* term is theoretically sensitive to scale but it can be kept constant at 0.9.

To detect features across scale we have to examine several octaves and levels. The integral images allow the SURF algorithm to calculate the responses with arbitrary large kernels. Making interest regions scale invariant does pose two challenges, how to scale the approximated kernels and how this influences the possible values for σ. The kernels have to have an uneven size to have a central pixel and the rectangular areas. This challenges are addressed through Scale Space Representation.

The scale space is divided into octaves. An octave represents a series of filter response maps obtained by convolving the same input image with a filter of increasing size. In total, an octave encompasses a scaling factor of 2 (which implies that one needs to more than double the filter size). The construction of the scale space starts with the 9 ×9 filter, which calculates the blob response of the image for the smallest scale. Then, filters with sizes 15 ×15, 21 ×21, and 27 × 27 are applied, by which even more than a scale change of 2 has been achieved. But this is needed, as a 3D non-maximum suppression is applied both spatially and over the neighbouring scales. Hence, the first and last Hessian response maps in the stack cannot contain such maxima themselves, as they are used for reasons of comparison only. Therefore, after interpolation, the smallest possible scale is σ = 1.6 corresponding to a filter size of 12 × 12, and the highest to σ = 3.2. Similar considerations hold for the other octaves.

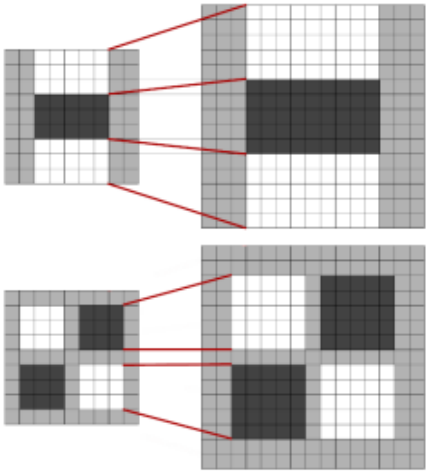


Figure 3.10 Increasing the size of kernels, and keeping the lobes correctly scaled

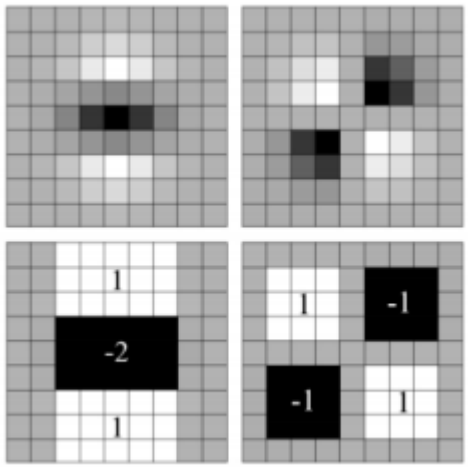


Figure 3.11 Discretized Gaussians and the approximations Dyy and Dxy

d. Interest Point description

The descriptor used in SURF describes the intensity content within the interest point neighbourhood based on first order Haar wavelet response in x and y direction.

Interest point description begins with fixing a reproducible orientation based on information from a circular region and the interest point. The square region aligned to the selected orientation is constructed and then SURF descriptor is extracted.

e. Orientation Assignment

To obtain invariance to image rotation, a reproducible orientation for the interest point is identified. For that, Haar wavelet responses in x and y direction within circular neighbourhood of radius 6s around interest point is calculated. Here, ‘s’ is the scale at

which interest point is detected. Use of integral image makes computation fast. Only six operations are needed to compute response in x and y.

Once wavelet response is calculated, the responses are represented as points in a space with horizontal strength along abscissa and vertical response along ordinate. Dominant response is estimated by calculating sum of all responses within a sliding orientation window size $\pi/3$. Then horizontal and vertical responses are summed. The two summed responses then yield a local orientation vector. The longest one determines the orientation of the interest point.

f. Descriptor based on Sum of Haar Wavelet Response

For the extraction of the descriptor, the first step consists of constructing a square region centred on the interest point and oriented along the orientation previously determined. The region is split up regularly into smaller 4 × 4 square sub-regions. This preserves important spatial information. For each sub-region, we compute Haar wavelet responses at 5×5 regularly spaced sample points. For reasons of simplicity, we call *dx* the Haar wavelet response in horizontal direction and *dy the* Haar wavelet response in vertical direction. "Horizontal" and "vertical" here is defined in relation to the selected interest point orientation.

Then, the wavelet responses *dx* and *dy* are summed up over each sub-region and form a first set of entries in the feature vector. In order to bring in information about the polarity of the intensity changes, we also extract the sum of the absolute values of the responses, |*dx*| and |*dy*|. Hence, each sub-region has a four-dimensional descriptor vector **v** for its underlying intensity structure

$$V = \sum dx, \sum dy, \sum |dx|, \sum |dy| \tag{3.17}$$

Concatenating this for all 4 × 4 sub-regions, this results in a descriptor vector of length 64. The wavelet responses are invariant to a bias in illumination (offset). Invariance to contrast (a scale factor) is achieved by turning the descriptor into a unit vector.

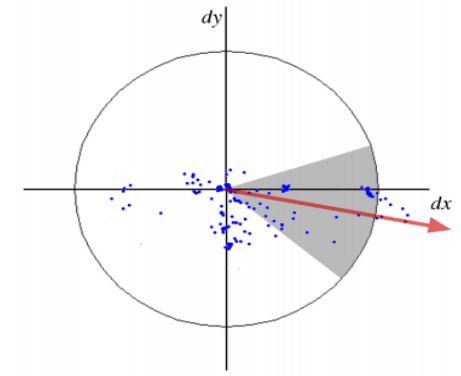


Figure 3.12 A sliding orientation window

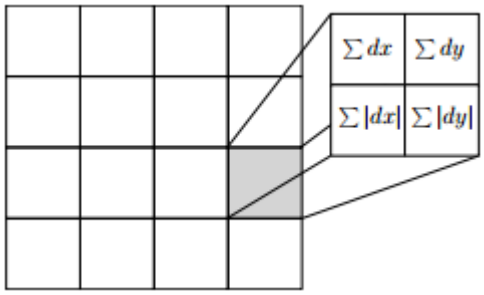


Figure 3.13 The layout of the image representation of a descriptor

3.2.2 Visual vocabulary

After computing key points and descriptors, next step involved in BoF is visual vocabulary formation. The calculated descriptors are used to generate the visual words which are later used to describe an object. The initial step in visual vocabulary formation is extraction visual words from images. This is achieved by the technique of clustering. Clustering of image descriptors provides the required words for visual vocabulary formation.

Clustering is division of data into groups of similar object. Each cluster (group) consists similarities between themselves and differences with another cluster. From perspective of machine learning, clustering is an unsupervised learning method. There are several algorithms for clustering data. Agglomerative algorithms, Divisive algorithms, K-means algorithm, Grid-based algorithms etc. are some. K-means clustering algorithm is a simple and efficient algorithm implemented to cluster descriptor vectors to obtain visual vocabulary.

In K-means clustering, first of all, to classify the given dataset, a certain number of clusters (let’s assume k) is fixed. Then, k centroids are defined for each clusters. These centroid

should be placed in such a way that distance between them is as large as possible. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no points are left, then the first step is completed. Then, k new centroids are calculated for the clusters formed from previous action. Now, the same activity of assigning dataset to nearest centroid is carried out This process continues until no more changes in location of k centroids is possible. Finally, this algorithm aims at minimizing an objective function, which in this case, is a squared error function. The objective function(J) is given as:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left| x_j^i - c_j \right|^2, \tag{3.18}$$

Where $\left| x_j^i - c_j \right|^2$ is chosen distance measure between a data point x_j^i and cluster centre c_j, is the indicator of n data points in their respective cluster centres. [13]

The basic k-means algorithm:

i. Select k data points as the initial centroids
ii. (Re) Assign all points to their closest centroids.
iii. Recomputed the centroid of each newly assembled cluster.
iv. Repeat steps 2 and 3 until centroid do not change.

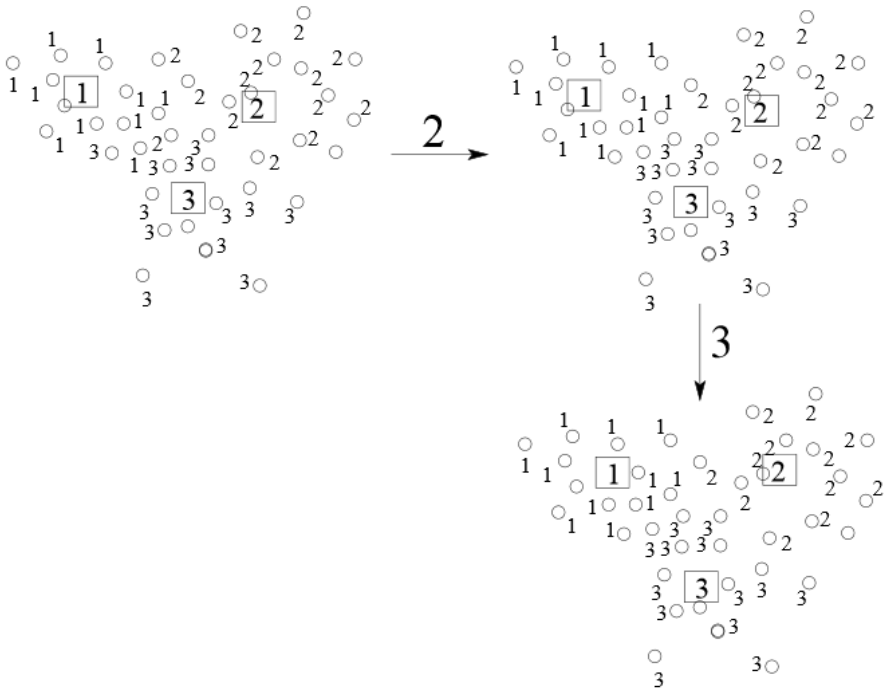


Figure 3.14 K-means clustering illustration

Thus formed clusters are visual words and collection of such visual words leads to formation of visual vocabulary. Then the visual vocabulary is used for quantizing features. A vector quantizer takes a feature vector and maps it to the nearest visual word in a visual vocabulary.

3.2.3 Classification

The generation of visual vocabulary is followed by image classification. The classification is done in order to assign input vector from image to one of two or more classes of objects. A decision rule is implied to divide input space (region occupied by input vectors) into decision regions separated by decision *boundaries.* [14]

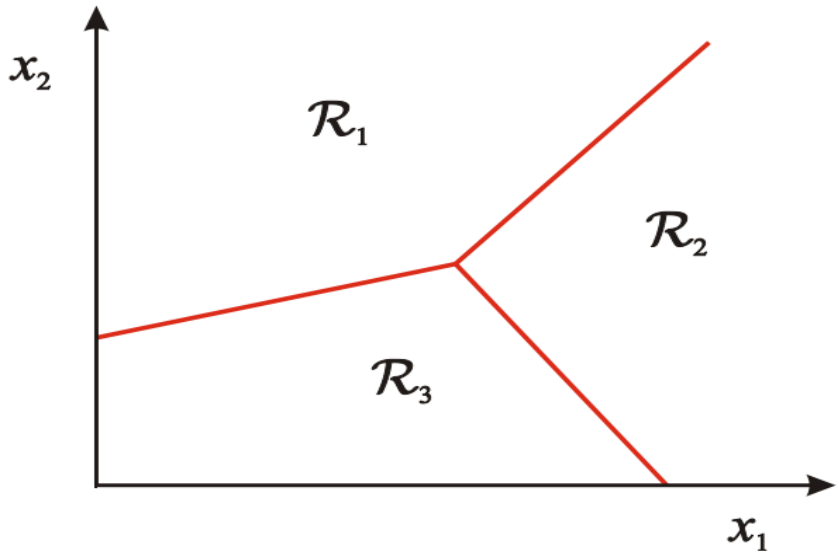


Figure 3.15 Input space divided into decision region

Several classifiers are available to carry out the classification task k-nearest neighbours, SVM etc. As SVM packages are publicly available and SVM framework is powerful, flexible and works very well in practice even with very small training sample size, SVM is implemented to carry out classification of bag of feature. [14] [15]

Support Vector Machine (SVM) is a supervised machine learning method used for both classification and regression problems. Here, it is used for classification problem. In this algorithm each data item is plotted in n-dimensional space, where n is available number of features, with the value of each feature being value of specific coordinate. Then a hyper-plane is found that differentiate the two classes.

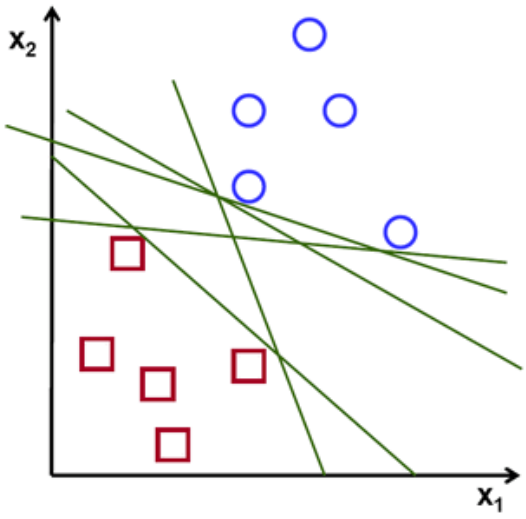


Figure 3.16 Possible hyper-planes

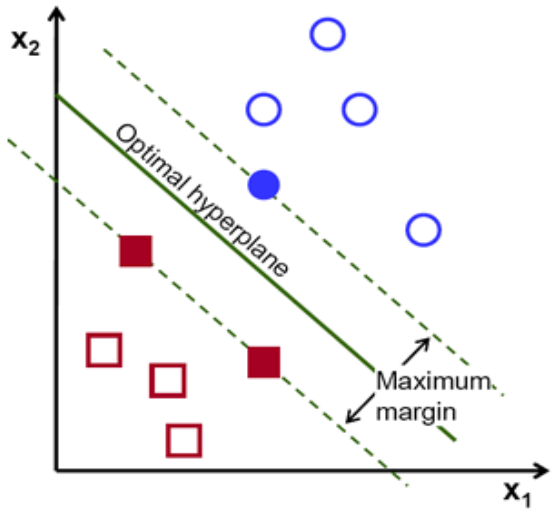


Figure 3.17 Optimal hyper-plane

There can be a number of hyper-plane separating two data set. In order to find the optimal hyper-plane, a margin is introduced. The distance between the line and the closest data points is referred as margin. The best or optimal line that can separate two data set is the line with largest margin. This is called Maximal-Margin hyper-plane or optimal hyper-plane.

For computation of optimal hyper-plane, first a hyper-plane is defined as f(x):

$$f(x) = ß_0 + ß^T x, \tag{3.19}$$

Where ß is known as weight vector and $ß_0$ is bias. Now, the optimal hyper-plane can be

defined as:

$$|ß_0 + ß^T x| = 1, \tag{3.20}$$

Where x represents training examples closet to hyper-plane. The training examples closet to the hyper-plane are called support vectors and the representation is called canonical hyper-plane Then, distance between a point x and a hyper-plane is computed given as:

$$\text{distance} = \frac{|ß_0 + ß^T x|}{||ß||} \tag{3.21}$$

For the canonical hyper-plane, the numerator is equal to one and the distance to support vectors is

$$distance_{support\ vector} = \frac{|ß_0 + ß^T x|}{||ß||} = \frac{1}{||ß||}, \tag{3.22}$$

Now, we know the margin (M) is twice the closest distance to the closest examples:

$$\text{M} = \frac{2}{||ß||} \tag{3.23}$$

At last, the maximization of M is carried out by applying Lagrangian optimization method.

Not all data are linearly separable. So the above mention method is insufficient for such data. The general idea to solve the problem is to map the original input space to some higher dimensional feature space where training set become separable. It is achieved by implying some lifting function φ (x). It is not required to explicitly computing the lifting transformation. [16] Instead a kernel function K is defined such that

$$K(x_i, x_j) = \varphi(x_i).\varphi(x_j)\ , \tag{3.24}$$

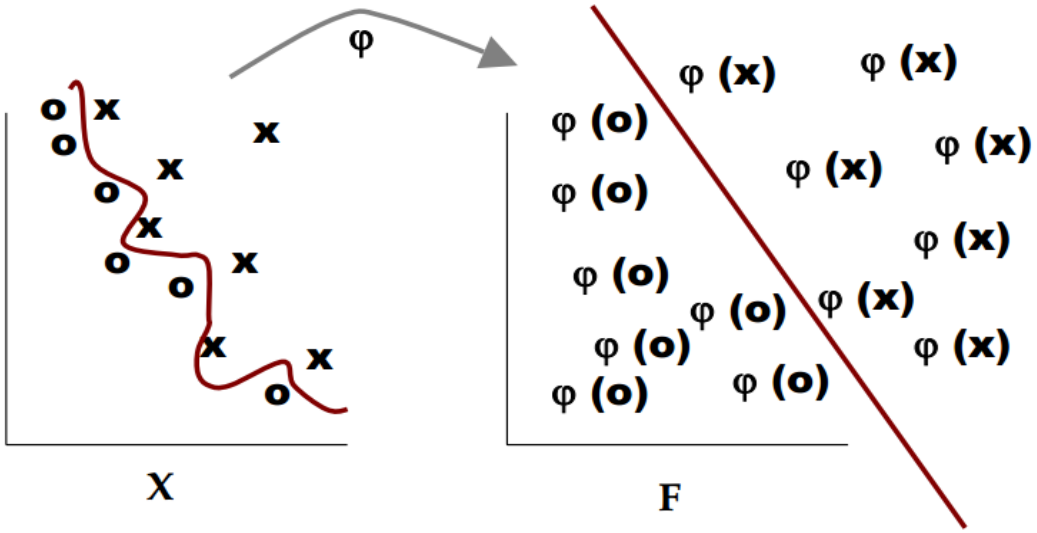


Figure 3.18 Applying lifting function to linearly inseparable data

The above mentioned SVM method is applicable only for binary class. It can be extended for multiple class by combining two-class SVMs. First of all, while training, SVM for each class I versus others is learned. Then while testing each SVM is applied to test example and assign to it the class of the SVM that returns the highest decision value. [16]

4. METHODOLOGY

4.1 System Diagram

4.1.1 System Block Diagram

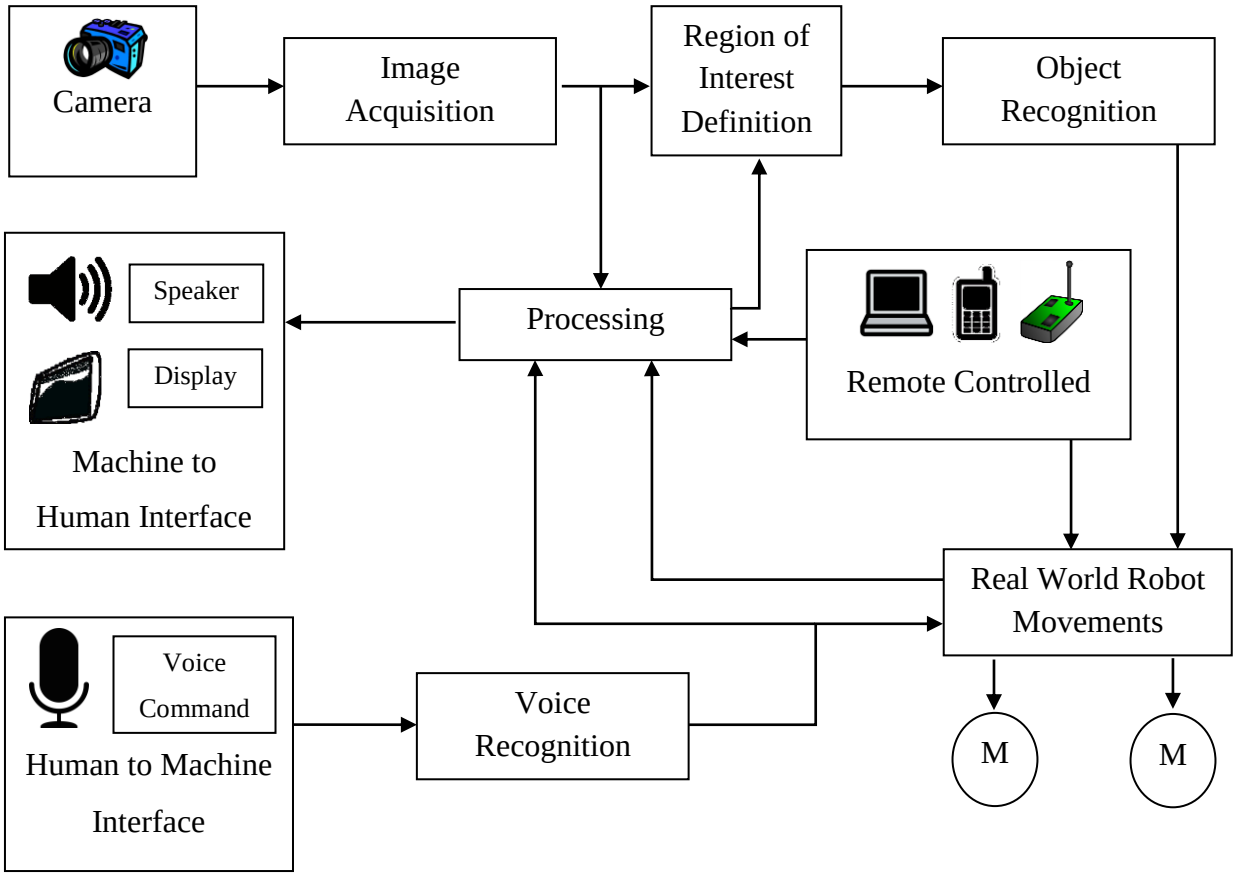


Figure 4.1 System Block Diagram

4.1.2 System Block Description

1. Human to Machine Interface: For the robot to act according to the human command, a microphone is used as a human to machine interface that will get the commands from humans to be heard by the robot. Voice command is the speech produced by us. It is in non-electrical form. A microphone picks up the signal to be recognized and converts into an electrical signal.

2. Machine to Human Interface: For the robot to inform the humans of any relevant decision it has made, based on various algorithms, the robot will make use of speakers (for synthesis of speech from robot as a form of information).

3. Voice Recognition: The speech signal form the microphone is analysed to produce a representation consisting of salient features in the speech. The speech pattern formed from the features is then compared to predefined speech pattern to determine a match of command.

4. Camera: The camera provides the vision to the robot. The camera acts as an eye for the robot, which will capture the images of the real world surroundings required for the detection of object commanded by the voice.

5. Image Acquisition: The camera captures a series of digital images regularly so that they can be processed to locate the object that is to be determined.

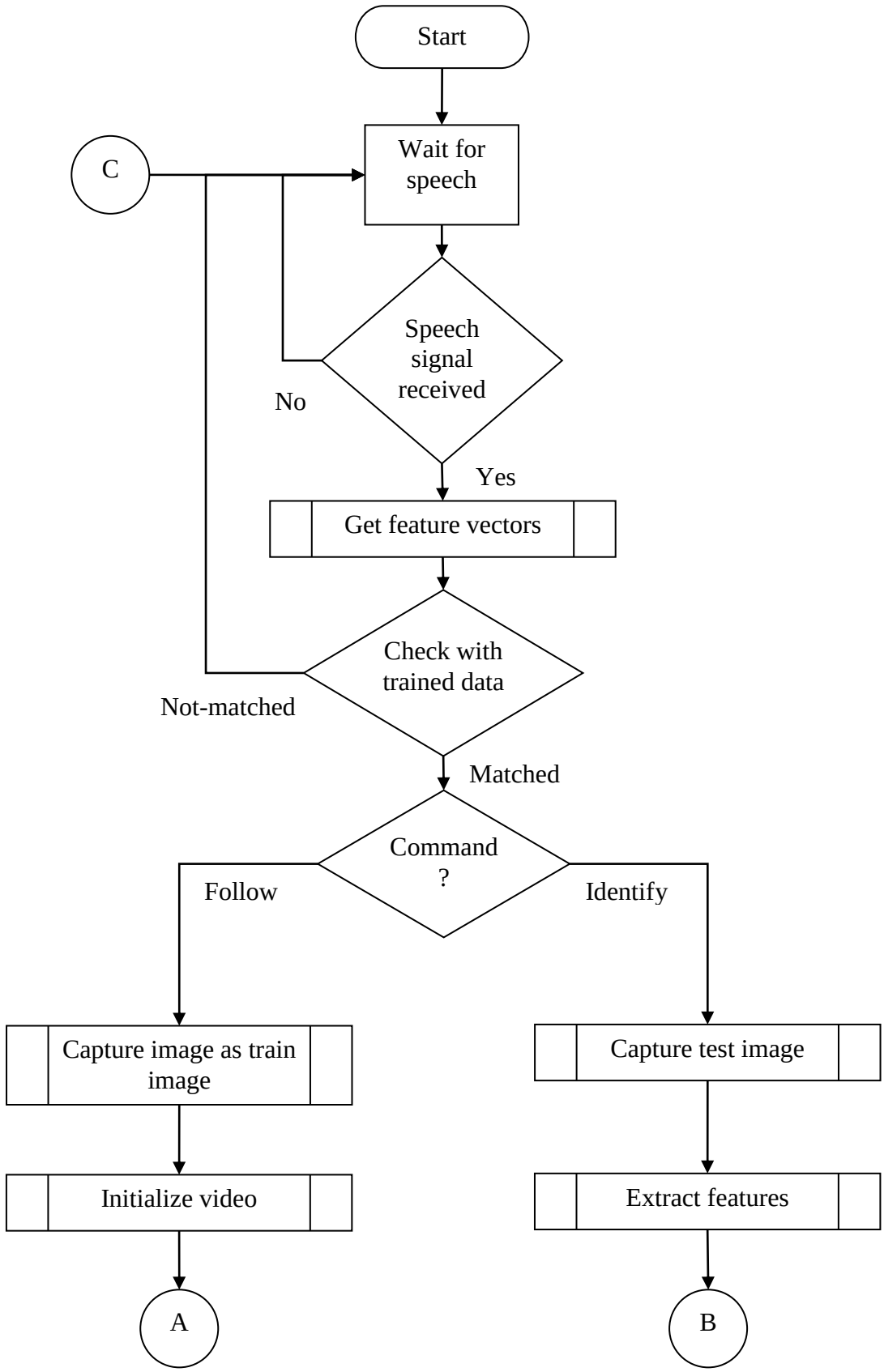
6. Region of Interest Definition: Based on the image acquired from the camera, real world movements of the robot and the voice command, the Region of Interest (ROI) is determined. The ROI determines the section of the image that needs to be processed to locate the object. The use of ROI improves the performance of object recognition.

7. Object Recognition: The images acquired from the camera are analysed to determine the salient features to recognize the object. The feature based object recognition approach has the benefit of recognizing even partially hidden objects with feature points exposed.

8. Remote Control: For the manual control of the robot, the remote control of the robot is necessary. It might even be useful while implementing machine learning in future scope to teach the robot some movements before it can actually start learning on its own.

9. Real World Robot Movements: The movement of the robot is generalized as real world movements which include the motion of the wheels and motion of the camera through motors controlled by the processor.

4.1.3 System Flow Chart



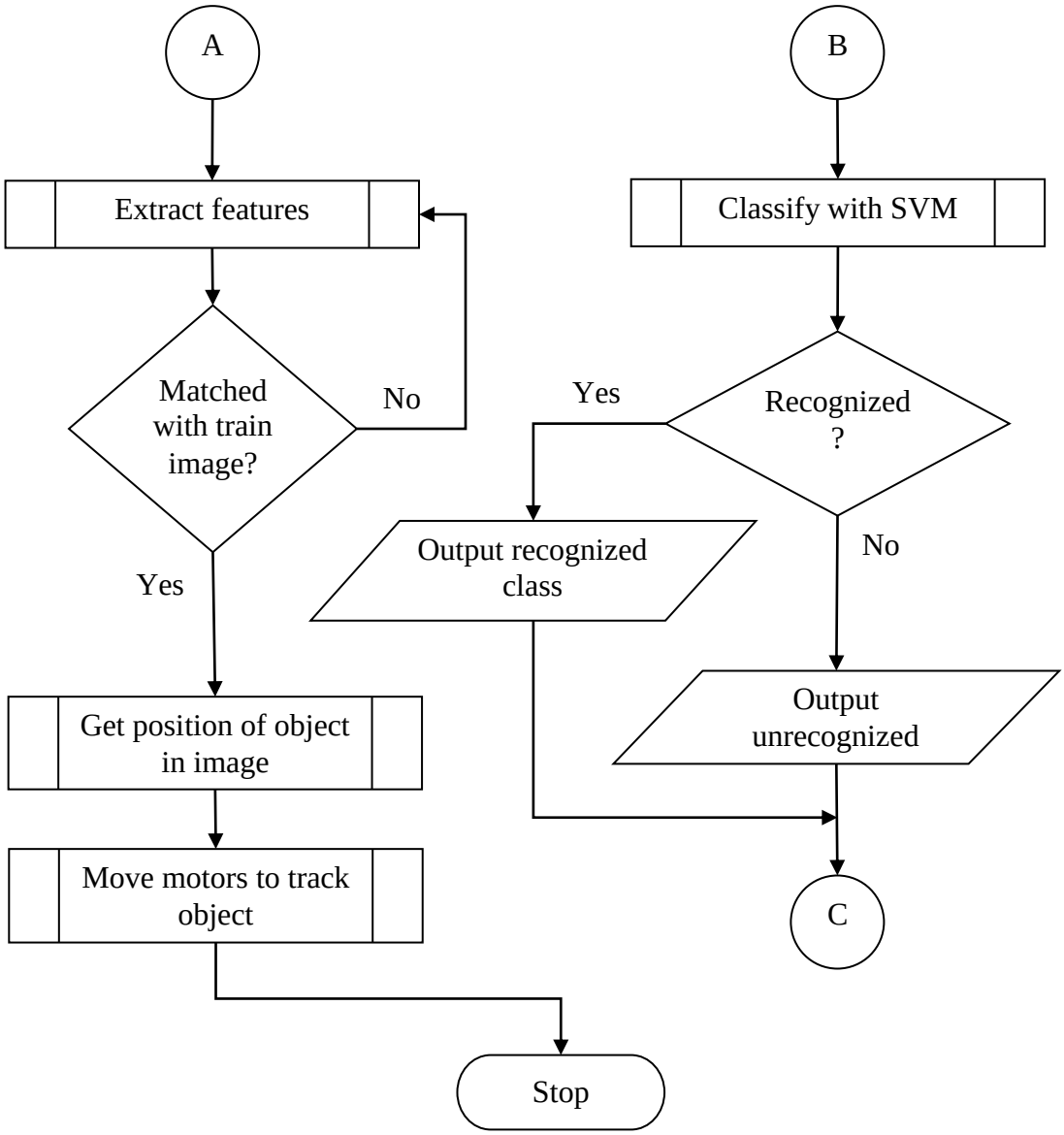


Figure 4.2 System flow chart

4.2 Speech Recognition Implementation

Because of the flexible nature, simple implementation and proven higher efficiency, we followed pattern recognition template approach for our isolated-word recognition system. We trained the robot to recognize two words: IDENTIFY and FOLLOW. To extract the feature vectors, MFCC was selected over the other feature extraction techniques. This was because of the advantages of MFCC discussed above in the literature overview.

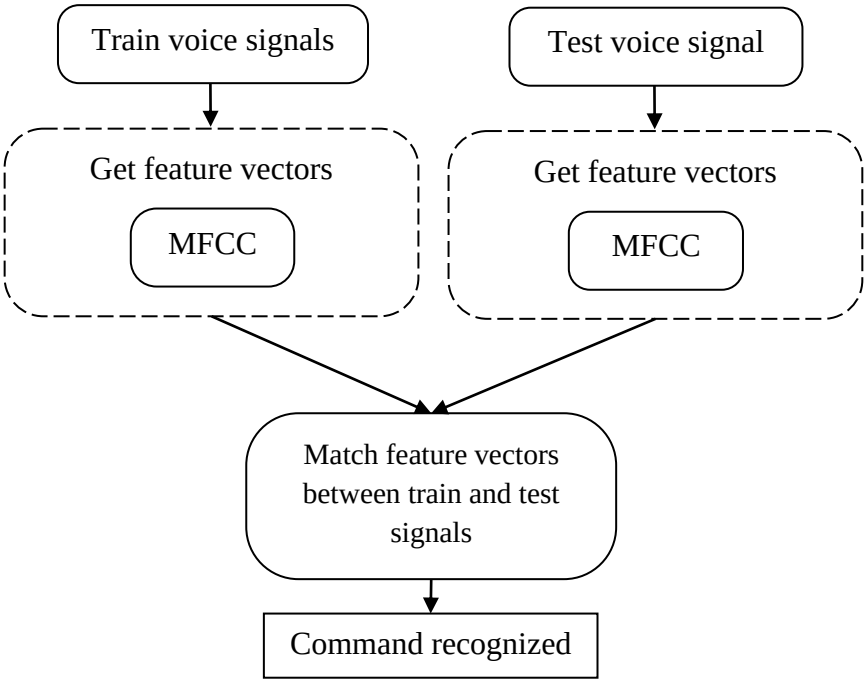


Figure 4.3 Block diagram of speech recognition

Mel Frequency Cepstral Coefficients can be obtained by first taking the FT of the signal, mapping the powers of the spectrum into the Mel scale, taking the logs of the power at each Mel frequency, and finally taking the discrete cosine transform of the log of the powers.

Using Python, the MFCC implementation was done and the cepstrums i.e. feature vectors were stored into a database to create a simple learning system. The implementations are explained in following three steps:

4.2.1 Audio samples and their feature vectors:

Before obtaining the feature vectors, we first had to attain the audio samples. We used the module pyAudio to record audio samples. This module provides us the ability to

manipulate an audio file and also set a threshold which only record when the input amplitude is larger than that threshold value i.e. in our system, the microphone always listens for a word and when the amplitude of a sound is greater than a certain level, it starts recording and when its silent (below a level), it stops recording.

The recorded data is sent to a pre-emphasis function that increases the signal energy at higher frequencies. The speech signal is sampled at 48KHz. And framing is performed

The FFT of length 512 is calculated thus obtaining the power spectrum of the signal. The energy of each frame is then calculated and fed to a log filter bank which gives the Cepstrum. Then the log of the Cepstrum are evaluated and DCT type-II performed which gives the feature vectors.

The graph of the word "IDENTIFY" in each of the step until its MFCC features are calculated and stored into a database, shown below:

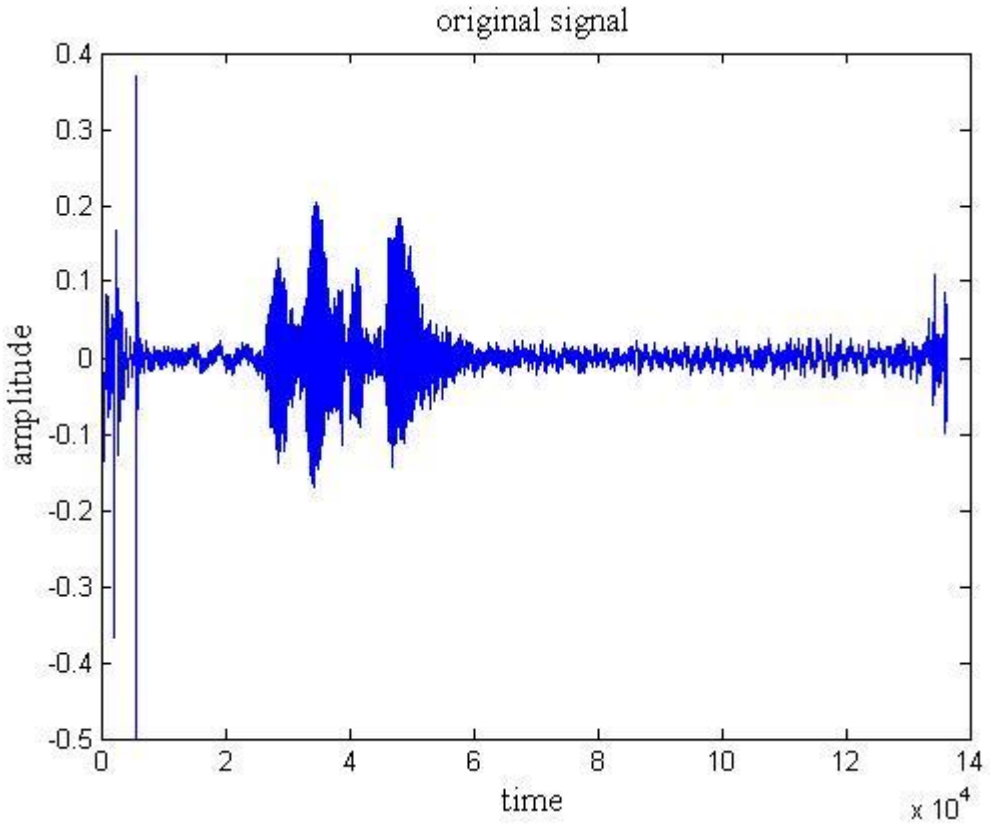


Figure 4.4 Original signal

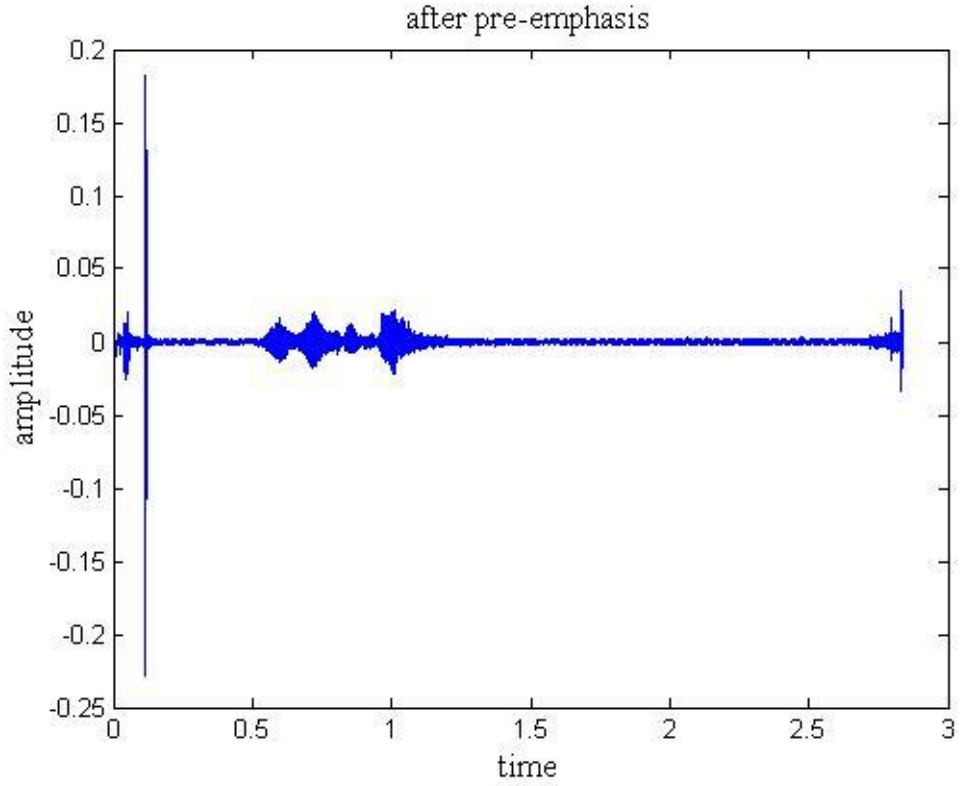


Figure 4.5 Signal after pre-emphasis

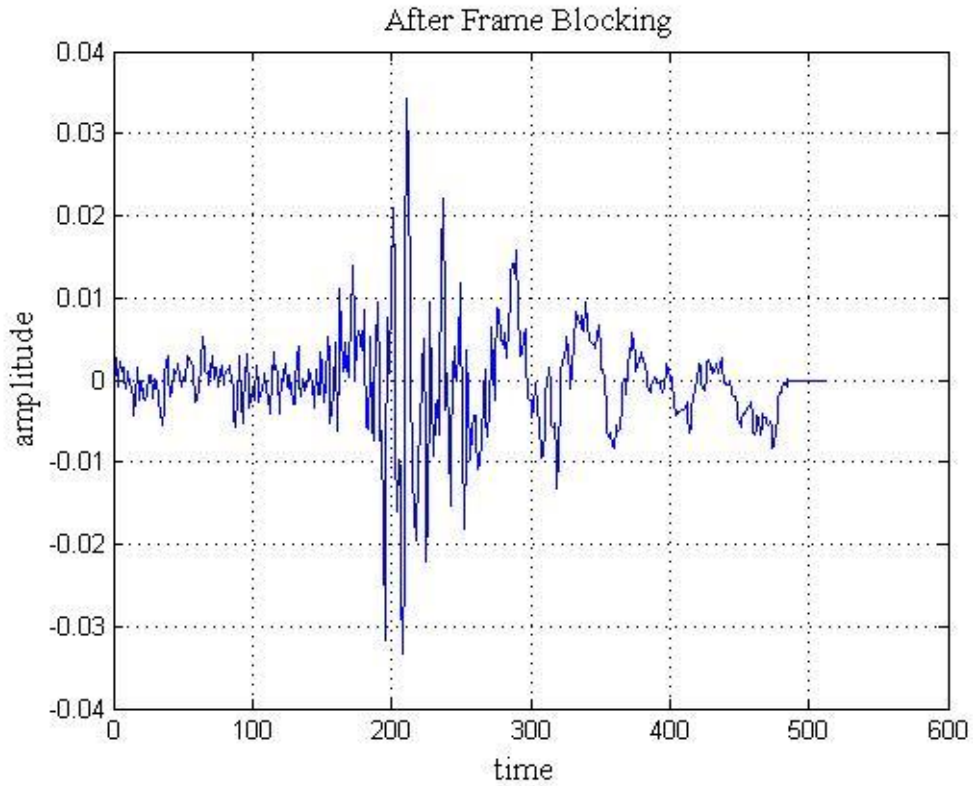


Figure 4.6 Signal after frame blocking

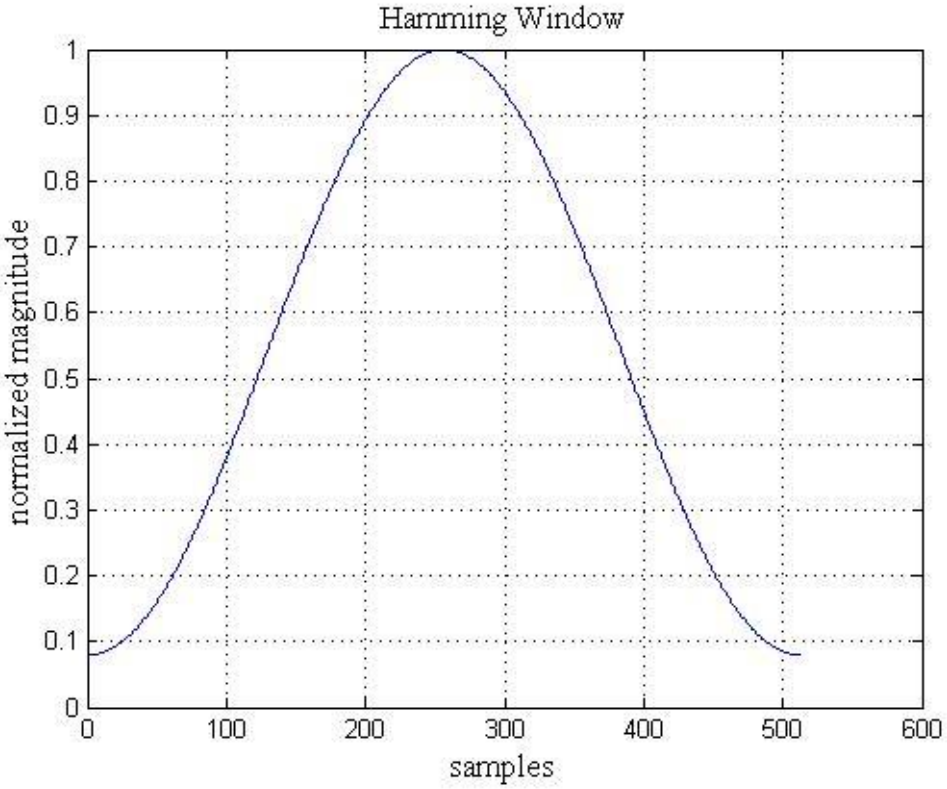


Figure 4.7 Hamming window

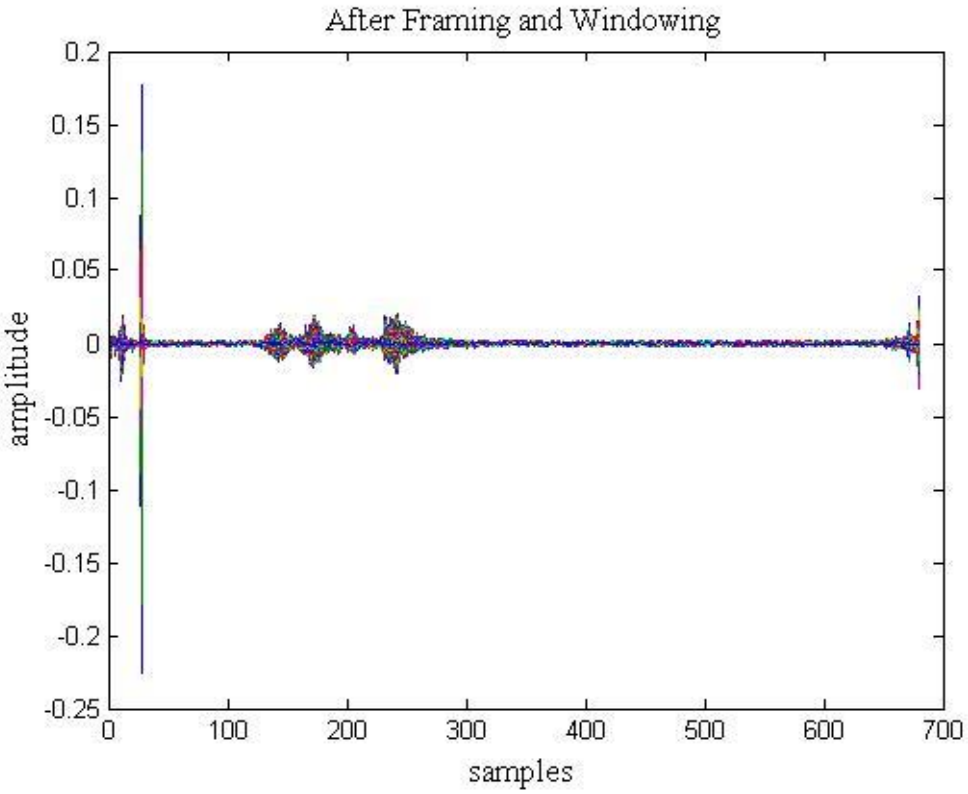


Figure 4.8 Signal after framing and hamming window

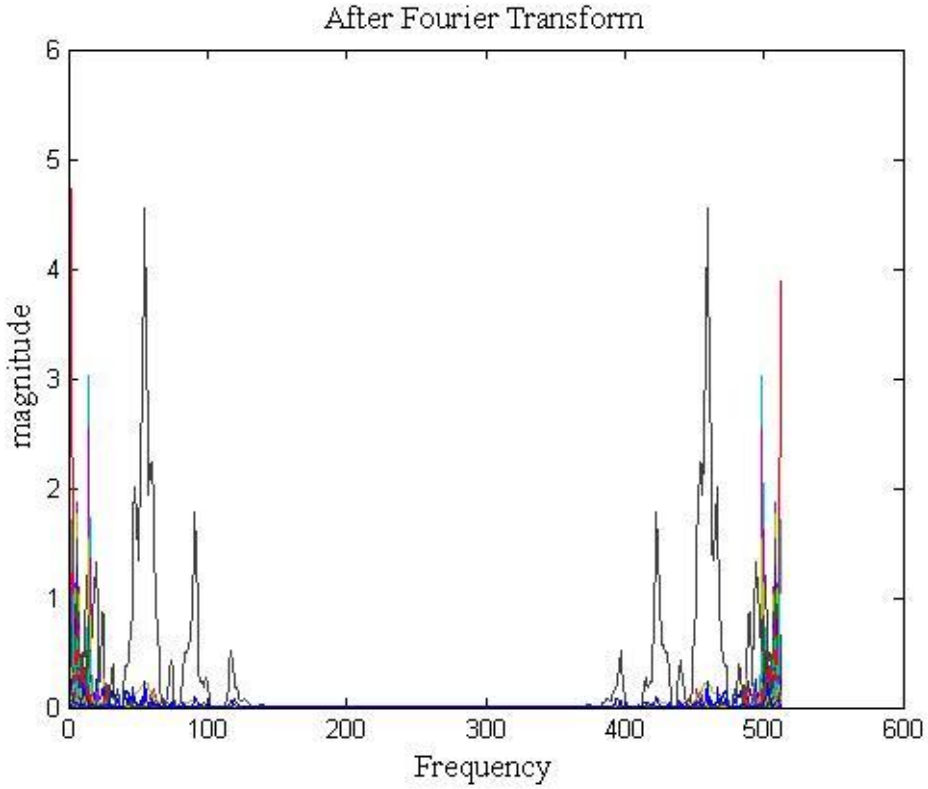


Figure 4.9 Signal after Fourier Transform

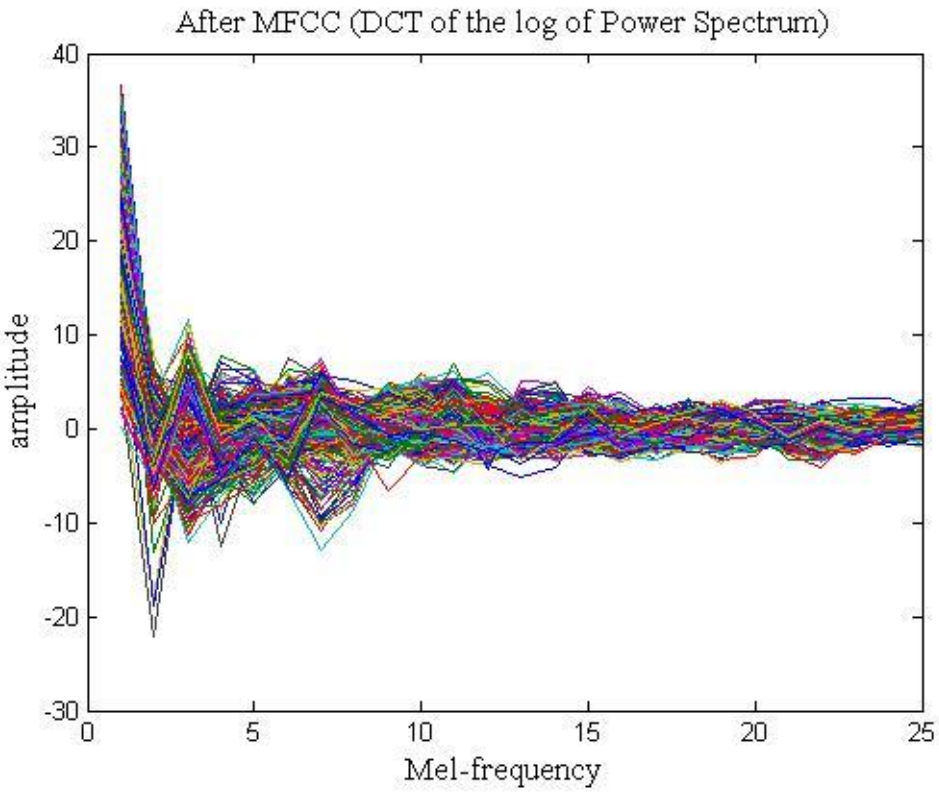


Figure 4.10 Signal after MFCC (DCT of the log of Power Spectrum)

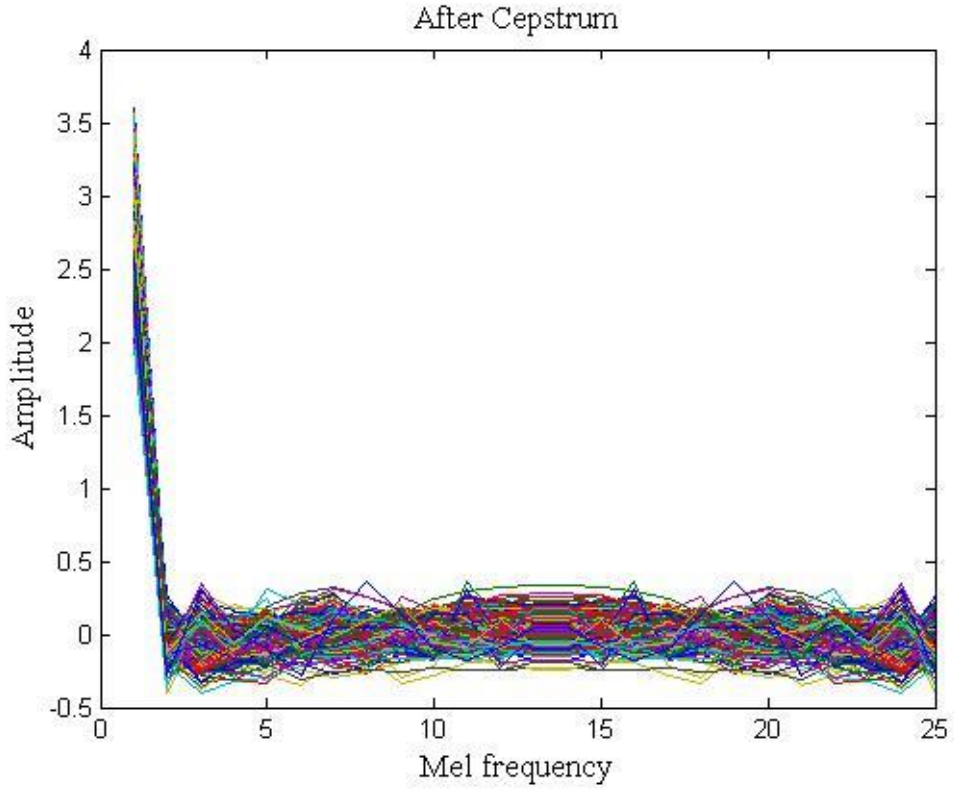


Figure 4.11 Signal after cepstrum

4.2.2 Training and Comparison:

System has to be trained to store the reference MFCC features for it to be able to recognize words. For this, the system uses the pyAudio library to take an input sample .wav file and generate MFCC values. These are then stored to the database. We trained the system a number of times to store a wide range of feature vectors representing a single word. The size of feature vectors did affect the recognition efficiency and it is discussed in the experiment section 7.1. This training creates a template of the reference patterns. To compare the patterns, similar steps of recording the speech sample is taken, its MFCC values are calculated and then it is compared to the stored values in the database. If there's a hit within a certain range of distance, the word is recognized. Our system is trained to identify the words, IDENTIFY and FOLLOW.

4.3 Computer Vision Implementation

4.3.1 Object Recognition

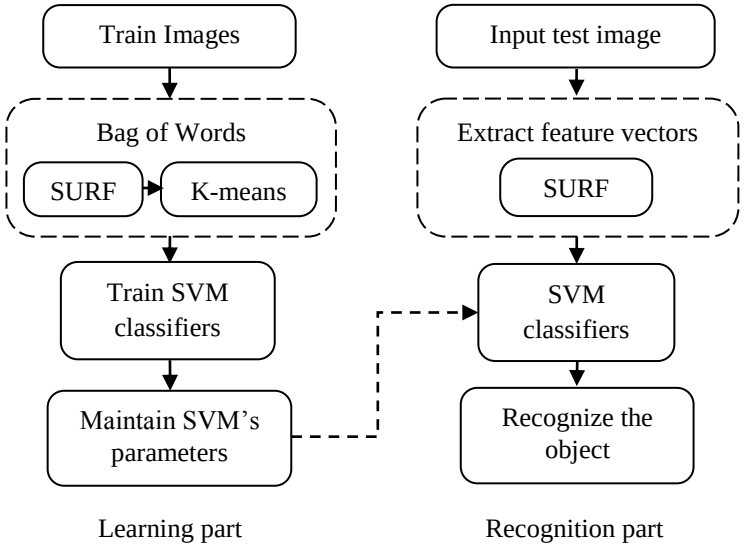


Figure 4.12 Block diagram of object recognition

The first phase of Bag-of-Feature method implementation is finding key points and descriptor from train images. SURF algorithm consists both key-points detector and descriptor. The descriptors vectors obtained from train images are then clustered using K-means algorithm. Centroid of each cluster represent a visual word and collection of such visual words form a visual vocabulary. Then each image is represented by frequency histogram where frequency refers to frequency of visual words. After image representation, image classification is implemented. The classification is performed using SVM.

4.3.2 Object Tracking

The robot can track and follow any object based on their features. For tracking the object, the robot captures the image of the object in front of it through the camera when it receives the command "follow". The image captured is stored as a train image. Then the SURF algorithm with brute force matching is used for feature based matching of the image captured by the camera.

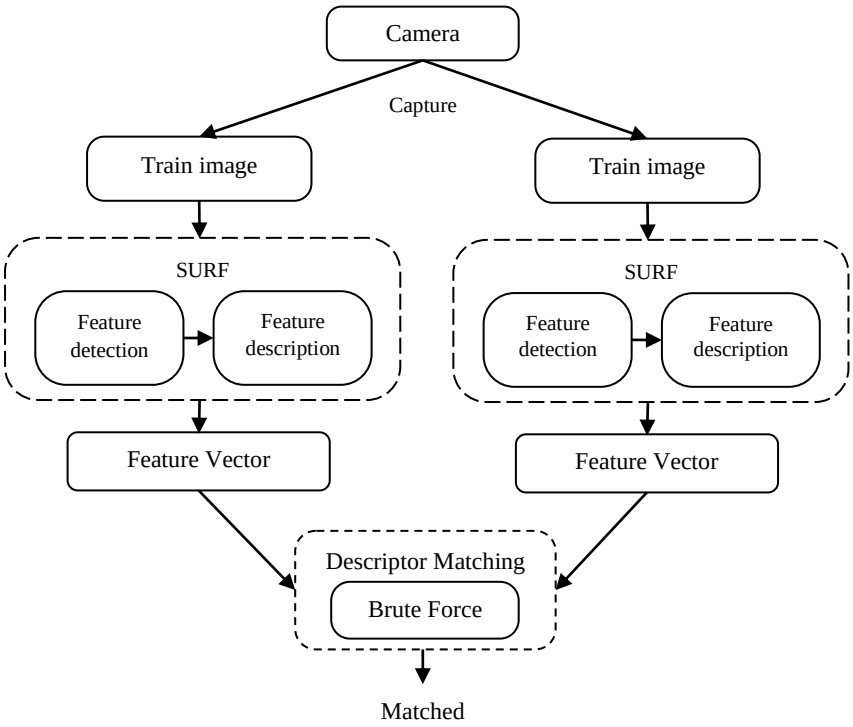


Figure 4.13 Object tracking block diagram

The robot is able to detect the position of the object to be tracked in the image based on its relative centre from the image frame centre. By calculating the offset, the robot moves its head and body to adjust to centre the object in the image. If the object is towards the left, the robot faces towards left to try to centre the object in the image. Similarly, the robot faces to right, up and down as the object is moved. The robot is also able to determine whether the object is near of far, but not the actual distance from the robot. The near or far position is approximated by calculating the area of the object bounding polygon. When the object is near, the bounding rectangle becomes larger because the object appears larger in the image. Similarly, when the object is far, the bounding rectangle becomes smaller as the object appears smaller. This provides the real world movements to the robot.

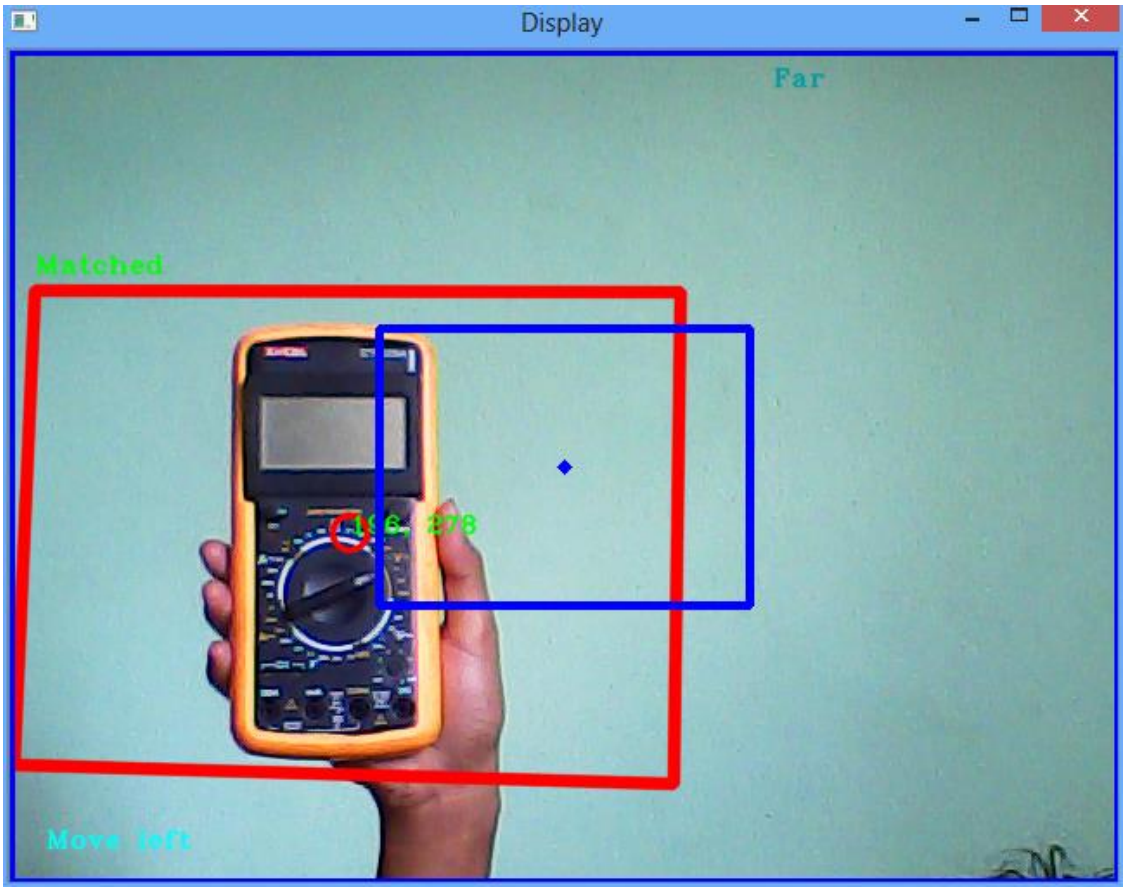


Figure 4.14 Tracking of object indicating the object is far and towards left

4.4 System Hardware Implementation

The physical movement of the robot is controlled by a separate microcontroller. The microcontroller used is Atmega32. The microcontroller receives the control signals through its serial port from the Raspberry Pi, and acts accordingly as instructed by the Raspberry Pi to the microcontroller.

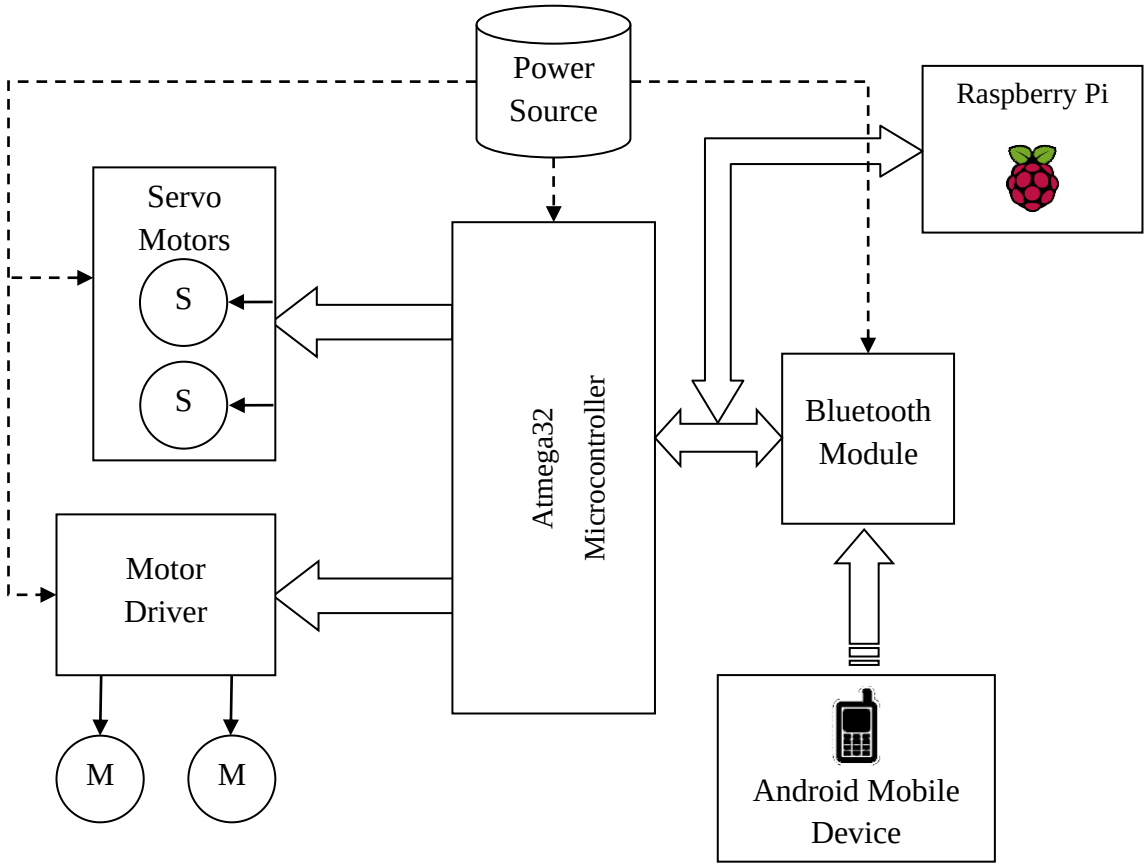


Figure 4.15 System hardware implementation

Figure 4.15 represents the control system of the robot. It includes two servo motors, denoted by 'S', which provides the pan and tilt motion to the head of the robot that mounts the camera. The two motors, denoted by 'M', provides the actual movement of the robot body in the environment. The simulation circuit of the system is shown in Figure 4.16.

The signals from the Raspberry Pi are generated based on the speech and image processing. Based on the speech signals and image signals, necessary commands are generated and sent as control signals to the control system.

As an example: To track an object, the speech signal 'follow' is fed into the Raspberry Pi. This signal is processed and recognized, and the robot captures the image of the object in front of it in order to get prepared for tracking it. The robot then starts to track the object and sends its position as control signal to the control system. Based on the control signals, the robot will move its head of whole body depending upon the necessity.

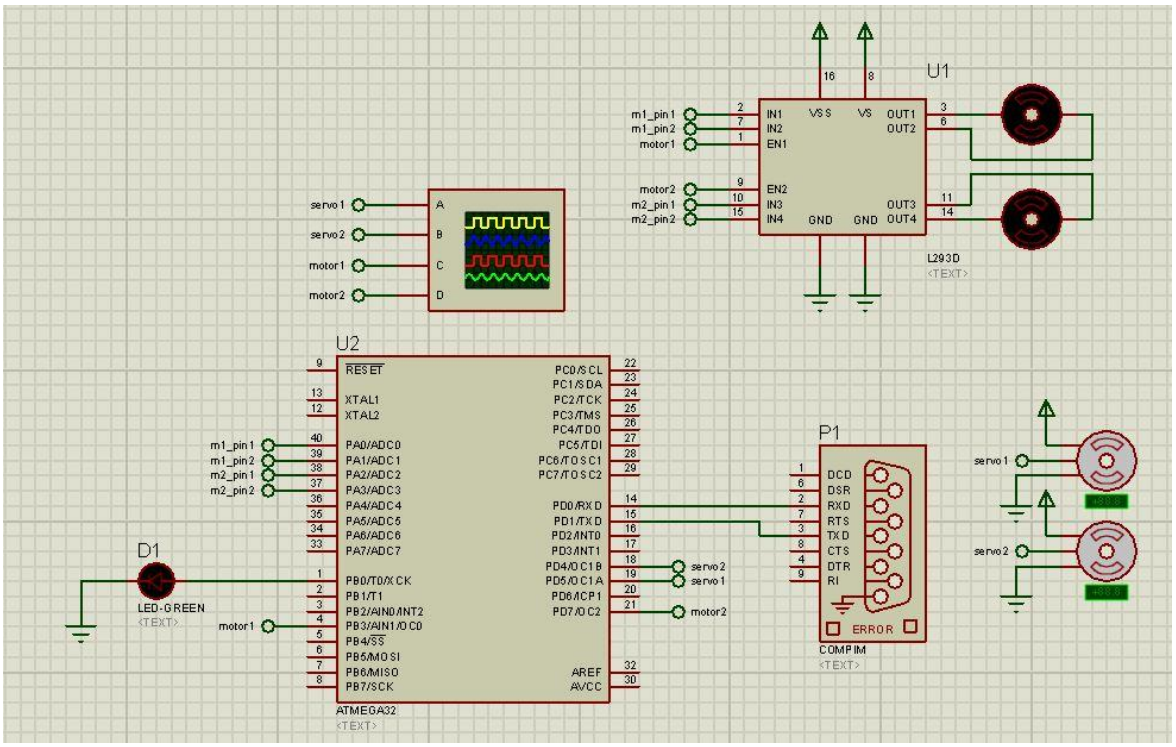


Figure 4.16 Proteus simulation of robot control system

The system hardware was designed to be assembled in a PCB. The PCB was designed using Altium Designer for the ease of assembling the control system of the robot hardware system. The PCB designed are as shown in the Figure 4.17.

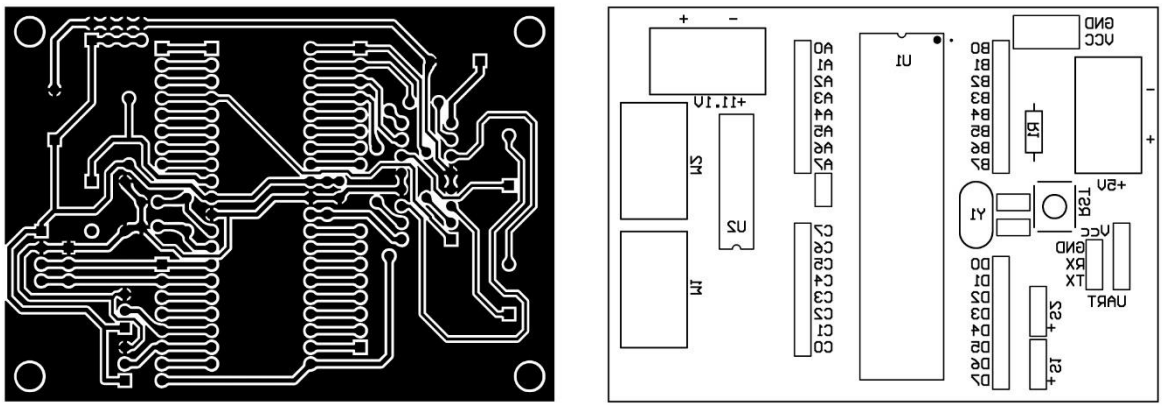


Figure 4.17 PCB (left) and top overlay (right) designed for the robot control system

4.5 Speech, Computer Vision and Hardware Integration

Both the speech and object recognition algorithms are implemented in the Raspberry Pi. The Raspberry Pi provides enough horsepower to perform the recognition within it. The speech processing and image processing tasks may run simultaneously or in succession. Voice commands such as ‘identify’, ‘follow’ will first cause speech recognition to take

place and then determine the need of image processing. Immediately after the command is recognized, image processing comes into action. The processed image signals generate required control signals for the robot control system if necessary.

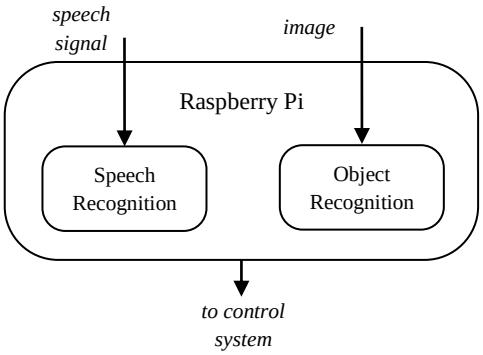


Figure 4.18 Integration of speech, computer vision and hardware of the robot system

5. RESOURCES USED

5.1 Microcontroller Tools

5.1.1 AVR Studio 6.0

Atmel Studio 6.0 is the integrated development platform for developing and debugging Atmel SMART ARM-based and Atmel AVR microcontroller (MCU) applications. Studio 6.0 supports all AVR and Atmel SMART MCUs. The Atmel Studio 6.0 IDP gives easy-to-use environment to write, build and debug your applications written in C/C++ or assembly code. It also connects to Atmel debuggers and development kits.

5.1.2 AVR Burn-O-Mat

The AVR Burn-O-Mat is a GUI for AVRdude-a popular command-line program for programming AVR microcontroller.

5.2 Simulation Tools

5.2.1 Proteus VSM

Proteus VSM brings Agile development into the embedded workflow. It enables rapid prototyping of both hardware design and firmware design in software, making it easy to make changes to both. Design, Test and Debug your embedded projects before a physical prototype is ordered.

5.3 PCB Designing Tool

5.3.1 Altium Designer

Altium Designer is an electronic design automation software package for printed circuit board, FPGA and embedded software design, and associated library and release management automation. It is developed and marketed by Altium Limited of Australia.

5.4 Python Resources

5.4.1 PyCharm Community Edition IDE

PyCharm is an Integrated Development Environment (IDE) used for programming in Python. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django. PyCharm is developed by the Czech company JetBrains. It is cross-platform working on Windows, Mac OS X and Linux. PyCharm has a Professional Edition, released under a proprietary license and a Community Edition released under the Apache License.

5.4.2 Modules

5.4.2.1 NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things like a powerful N-dimensional array object, sophisticated (broadcasting) function stools for integrating C/C++ and Fortran code useful linear algebra, Fourier transform, and random number capabilities.

5.4.2.2 SciPy

The SciPy library is one of the core packages that make up the SciPy stack. It provides many user-friendly and efficient numerical routines such as routines for numerical integration and optimization.

5.4.2.3 pyAudio

pyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library. With pyAudio, one can use python to play and record audio on variety of platform, such as GNU/Linux, Microsoft and Apple mac OS X. [17]

5.4.2.4 cPickle

The pickle module implements a fundamental, but powerful algorithm for serializing and de-serializing a Python object structure. “Pickling” is the process in which a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream is converted back into an object hierarchy. Pickling (and unpickling) is alternatively known as “serialization”, “marshalling,” or “flattening”, however, to avoid confusion, the terms used here are “pickling” and “unpickling”. The pickle module has an optimized cousin called the cPickle module. As its name implies, cPickle is written in C, so it can be up to 1000 times faster than pickle. [18]

5.4.2.5 pySerial

pySerial encapsulates the access for the serial port. It provides backends for Python running on Windows, OSX, Linux, BSD (possibly any POSIX compliant system) and IronPython. The module named “serial” automatically selects the appropriate backend.

5.4.2.6 Scikit-learn

It is a simple and efficient tool for data mining and data analysis which is built on NumPy and SciPy and matplotlib. It is an open source library, accessible to everybody, and reusable in various contexts and can be commercially used using BSD license. It provides tools for Clustering, Regression, Classification, Preprocessing, Model selection etc. [19]

5.4.2.7 Matplotlib

Matplotlib is a python 2D plotting library which produces publication quality figures in variety of hardcopy formats and interactive environments across platform. Using matplotlib, one can generate plots, histograms, power spectra, bar charts, error charts etc. [20]

5.5 Libraries Used

5.5.1 OpenCV

Open Source Computer Vision (OpenCV) has C++, C, Python and Java interfaces supported by Windows, Linux, Mac OS, iOS and Android designed for image processing and computer vision tasks. The library takes advantage of multi-core processing and hardware acceleration. Its usage ranges widely like for interactive art, mines inspection, robotics etc.

6. HARDWARE COMPONENTS USED

6.1 Atmega32

It is a high-performance, low-power Atmel 8-bit AVR RISC-based microcontroller combines 32KB of programmable flash memory, 2KB SRAM, 1KB EEPROM, an 8-channel 10-bit A/D converter, and a JTAG interface for on-chip debugging. The device supports throughput of 16 MIPS at 16 MHz and operates between 4.5-5.5 volt. By executing instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed.

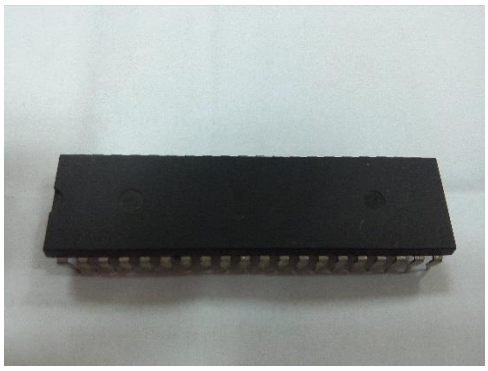


Figure 6.1 Atmega32 microcontroller

6.2 Raspberry Pi 2

The Raspberry Pi is a series of credit card-sized single-board computers developed in the United Kingdom by the Raspberry Pi Foundation. It has a 900MHz quad-core ARM Cortex-A7 CPU and 1GB RAM. In addition, it also has 4 USB ports, 40 GPIO pins, Full HDMI port, Ethernet port, combined 3.5mm audio jack and composite video, Camera interface(CSI), Display interface(DSI), Micro SD card slot and VideoCore IV 3D graphics core. [21]

Figure 6.2 Raspberry Pi 2

6.3 Raspberry Pi Camera

The Raspberry Pi camera module is a 5-megapixel camera. The camera module can be used to take videos as well as still photographs. [22]

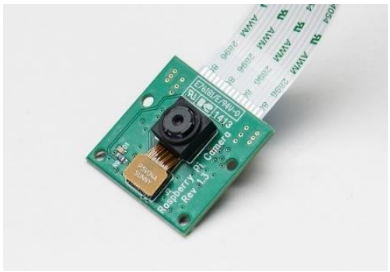


Figure 6.3 Raspberry Pi camera

6.4 DC Gear Motor

Geared DC motor is an extension of DC motor. A geared DC Motor has a gear assembly attached to the motor. The speed of motor is counted in terms of rotations of the shaft per minute and is termed as RPM. The gear assembly helps in increasing the torque and reducing the speed.

Figure 6.4 DC gear motor

6.5 Motor Driver

A motor driver is a little current amplifier; the function of motor drivers is to take a low-current control signal and then turn it into a higher-current signal that can drive a motor.

Figure 6.5 Motor driver module

6.6 Servo Motor

A servo motor is a rotary actuator that allows for precise control of angular position. It consists of a motor coupled to a sensor for position feedback. It also requires a servo driver to complete the system. The drive uses the feedback sensor to precisely control the rotary position of the motor. This is called closed-loop operation. By running the system closed-loop, servo motors provide a high performance alternative to stepper and AC induction motors.

Figure 6.6 Servo motor

6.7 Li-Po Battery

Lithium polymer battery or more correctly lithium ion polymer battery is a rechargeable battery of lithium-ion technology. Li-Pos work on the principle of intercalation and de-intercalation of lithium ions from a positive electrode material and a negative electrode material, with the liquid electrolyte providing a conductive medium. To prevent the electrodes from touching each other directly, a microporous separator is in between, which allows only the ions and not the electrode particles to migrate from one side to the other.

Figure 6.7 Li-Po battery

6.8 Buck Converter

A buck converter (step-down converter) is a DC-to-DC power converter which steps down voltage, while stepping up current), from its input to its output. It is a class of switched-mode power supply (SMPS) typically containing at least two semiconductors (a diode and

a transistor) and at least one energy storage element, a capacitor, inductor, or the two in combination. To reduce voltage ripple, filters made of capacitors (sometimes in combination with inductors) are normally added to such a converter's output and input.

Figure 6.8 Buck converter

6.9 Microphone

A microphone is a transducer that changes sound energy which exists as patterns of air pressure into patterns of electric current. Microphones use electromagnetic induction, capacitance change or piezoelectricity to produce an electrical signal from air pressure.

Figure 6.9 Condenser Microphone

7. EXPERIMENTS AND RESULTS

7.1 Speech Recognition Experiments

7.1.1 Effect of size of feature vectors:

To observe the effect of different sizes of feature vectors on speech signal recognition, we ran out an experiment to check the word recognition using different template sizes.

For the word IDENTIFY, we initially trained the system 8 times and the word was checked 10 times. There were 2 hits and 8 misses. We trained the system again and checked the recognition 14 times after it was trained for 17 times. There were 12 hits and 2 misses. On further training the system, up to 31 times, which generated a huge cluster of feature vectors, and checking the recognition, 26 comparisons were made, we found 12 hits and 14 misses. The similar process was repeated for the word FOLLOW. The findings of experiments on both the words are listed in the Table 7.1 below:

Word	No. of times trained	No. of comparisons	No. of hits	No. of misses	Accuracy (%)
IDENTIFY	8	10	2	8	20
IDENTIFY	17	14	12	2	85.7
IDENTIFY	31	26	12	14	46.15
FOLLOW	16	24	15	9	62.5
FOLLOW	25	15	15	0	100
FOLLOW	20	28	25	3	89.28

Table 7.1 Effect of size of feature vectors in speech recognition

Hence the speech recognition capacity of the system doesn't increase linearly with the feature vector size. Instead after a certain number of training of the system i.e. the size of feature vector to a certain level, the efficiency decreases. The change in efficiency is not linear, there are certain abrupt changes in efficiency compared to the change in feature vector size.

This is called over fitting and this model describes random error or noise instead of the underlying relationship. This condition violates Occam' razor, for example by including more parameters than are required or using a more complicated approach than required. Occam's razor implies that any given complex function is a *priori* less probable than any

given simple function. If the new, more complicated function is selected instead of the simple function and if there was not a large enough gain in training data fit to offset the complexity increase, then the new function 'overfits' the data and the complex overfitted function will likely perform worse than the simpler function on validation data outside the training dataset. [23]

7.1.2 Effect of different sampling rate:

To observe the effect of sampling rate on the speech recognition, an experiment was carried out to check the recognition efficiency using varying sampling rate. The obtained results are tabulated below for the word 'IDENTIFY':

Sampling rate: 48000 Hz

Word	No. of times trained	No. of comparisons made	No. of Hits	No. of misses	Accuracy (%)
IDENTIFY	8	10	2	8	20
IDENTIFY	17	14	12	2	85.7
IDENTIFY	31	26	12	14	46.15

Table 7.2 Effect of sampling rate 48KHz

Sampling rate: 16000 Hz

Word	No. of times trained	No. of comparisons made	No. of Hits	No. of misses	Accuracy (%)
IDENTIFY	7	10	3	7	33
IDENTIFY	18	16	9	7	56.25
IDENTIFY	31	19	12	7	63.15

Table 7.3 Effect of sampling rate 16KHz

Sampling rate: 8000 Hz

Word	No. of times trained	No. of comparisons made	No. of Hits	No. of misses	Accuracy (%)
IDENTIFY	10	10	2	8	20
IDENTIFY	16	10	2	8	20
IDENTIFY	24	10	6	4	60

Table 7.4 Effect of sampling rate 8KHz

When the speech was sampled at lower sampling rate than the earlier, the recognition efficiency was reduced. Sampling at lower sampling rate provides fewer number of samples thus the feature extracted from the speech might not have been enough to represent the signal. At 16 KHz and 8 KHz sampling rate, the over fitting was not observed even when system was trained for more than 20 times. Instead, feature values weren't enough to model the speech signal. Considering the efficiency observed at different sampling rate, we sampled our speech signal at 48 KHz.

7.2 Object Recognition Experiments

7.2.1 Collection of Train Image Datasets

For training the system for object recognition, 50 images for each category of class to be recognized were collected using online resources. A few of the datasets collected for training the system are shown here.

Figure 7.1 Examples of train dataset images

7.2.2 Collection of Test Images

As with train images, test images were also collected from online resources. 20 test images (10 with background and 10 without background) for each class to be recognized were collected for the experiment. Some of the test images are shown here.

Figure 7.2 Test images without background

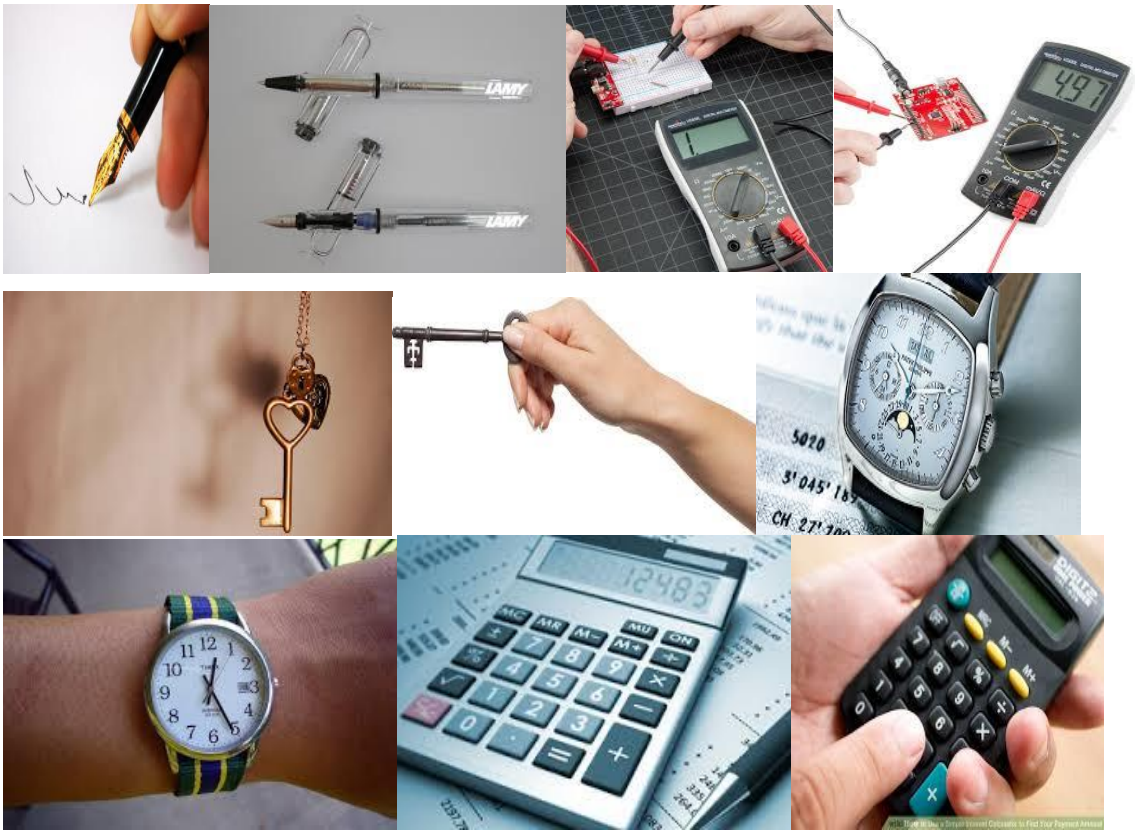


Figure 7.3 Test images with background

7.2.3 Effect of background in recognition

To observe the effect of background in object recognition, we collected 10 images with no background (white background) for each class. Then ran out a test to experiment how it effects the efficiency of the recognition.

We initially trained the system using 50 train images of ‘calculator’ and ran the test. For the second test, we trained the system with 50 train images of two classes ‘calculator’ and ‘key’ each and ran the test for recognition of calculator and key objects separately. Similarly, the experiments were conducted for 3, 4 and 5 classes in succession. The findings of experiments on both the words are listed in Table 7.5 below:

Train Objects	**Test Object**	**Recognition (out of 10)**		
		Recognized	False Recognition	Unrecognized
Calculator	Calculator	8	-	2
Calculator	Calculator	9	1	0
Key	Key	10	0	0
Calculator	Calculator	6	3	1
Key	Key	10	0	0
Multimeter	Multimeter	5	3	2
Calculator	Calculator	4	5	1
Key	Key	10	0	0
Multimeter	Multimeter	5	3	2
Pen	Pen	8	2	0
Calculator	Calculator	6	3	1
Key	Key	8	2	0
Multimeter	Multimeter	4	3	3
Pen	Pen	8	2	0
Watch	Watch	7	3	0

Table 7.5 Effect of background in object recognition

From the experiment, it was observed that having no background (white/plain background) increases the efficiency of the recognition. The images had pretty high accuracy in recognition with a little false recognition cases in multiclass recognition and some unrecognized cases. When comparing with the efficiency results in Table 7.7 for images

with background, this result is pretty good. Thus, less cluttered background causes the recognition efficiency to increase.

7.2.4 Effect of Varying Number of Train Images

To observe the effect of variation in the number of train images in object recognition, we collected 10, 20, 30, 40, 50 train images for 5 different classes each. Then ran out a test to observe the change in the efficiency of the recognition. During the test we also varied the number of keypoints extracted during training by varying the SURF Hessian threshold and observed the result for each cases.

The findings of experiments on both the words are listed in Table 7.6 below:

Train Object	**Threshold**	**Recognition (out of 10) with varied no. of train images**				
		50 train images	40 train images	30 train images	20 train images	10 train images
Calculator	100	7	5	6	8	5
	600	7	7	7	8	5
	1100	6	7	7	6	6
	1600	7	6	7	7	6
	2100	8	8	9	9	7
Key	100	5	5	5	5	5
	600	5	5	6	6	6
	1100	6	6	6	6	7
	1600	6	6	5	6	7
	2100	6	6	6	6	6
Multimeter	100	2	3	2	2	2
	600	4	3	3	3	3
	1100	4	3	2	2	2
	1600	4	4	2	2	1
	2100	3	2	1	2	1
Pen	100	6	6	6	6	6
	600	5	4	3	4	4
	1100	4	4	4	3	5

	1600	-	4	-	-	5
	2100	-	-	-	-	5
Watch	100	8	3	6	4	1
	600	7	4	5	4	1
	1100	5	3	4	4	1
	1600	4	3	4	4	0
	2100	5	3	4	3	2

Table 7.6 Effect of varying number of train images in object recognition

These results show that efficiency increase with no. of training datasets which is a pretty obvious case. Having more no. of images in the training dataset increases the odds of recognizing all varieties and forms of same object. For example, the calculator having large collection of all kinds of calculator including scientific and normal calculator is able to classify both kinds of calculators as 'calculators'. If the dataset had small no. of images containing only images of scientific calculators, then the system might not have been able to recognize normal calculators. Thus, in obvious ways larger datasets increases the efficiency of recognition.

Here we also observed that more keypoints (less threshold) gives higher accuracy in recognition which is prevalent in pen and watch. For calculator, key, multimeter however it is not the case but should have been so. This could because at higher threshold the lower no. of keypoints caused only the extreme keypoints to be detected which played important role in recognition and thus gave better recognition. But for larger datasets more keypoints would be better because with more keypoints gives the ability to recognize images with more precision.

7.2.5 Effect of Varying Number of Classification Class

To observe the effect of variation in the number of classification class in object recognition, we used 50 train images for each of 5 different classes. The system was initially trained with 2 classes. Then a test was run to classify test images among the two classes. Similarly, 3, 4 and 5 classes were used for training in succession. For each case we ran out the test to observe the change in the efficiency of the recognition. During the test

we also varied the number of keypoints extracted during training by varying the SURF Hessian threshold and observed the result for each cases.

The findings of experiments on both the words are listed in Table 7.7 below:

Train Objects	**Threshold**	**Test Object**	**Recognition (out of 10)**		
			Recognized	False Recognition	Unrecognized
Calculator Key	100	Calculator	6	0	4
	600		7	1	2
	1100		6	0	4
	1600		7	1	2
	2100		8	1	1
	100	Key	5	0	5
	600		6	0	4
	1100		6	1	3
	1600		6	1	3
	2100		5	0	5
Calculator Key Multimeter	100	Calculator	5	3	2
	600		7	1	2
	1100		7	1	2
	1600		6	1	3
	2100		7	1	2
	100	Key	5	3	2
	600		5	1	4
	1100		7	0	3
	1600		6	2	2
	2100		5	1	4
	100	Multimeter	2	2	6
	600		4	0	6
	1100		3	0	7
	1600		4	0	6

	2100		1	0	9
Calculator Key Multimeter Pen	100	Calculator	6	1	1
	600		7	0	3
	1100		6	1	3
	1600		-	-	-
	2100		-	-	-
	100	Key	4	3	3
	600		5	1	4
	1100		6	1	3
	1600		-	-	-
	2100		-	-	-
	100	Multimeter	1	2	7
	600		3	0	7
	1100		4	0	6
	1600		-	-	-
	2100		-	-	-
	100	Pen	4	4	2
	600		3	5	2
	1100		3	6	1
	1600		-	-	-
	2100		-	-	-
Calculator Key Multimeter Pen Watch	100	Calculator	6	2	2
	600		7	2	1
	1100		8	1	1
	1600		-	-	-
	2100		-	-	-
	100	Key	4	4	2
	600		5	1	4
	1100		4	3	3

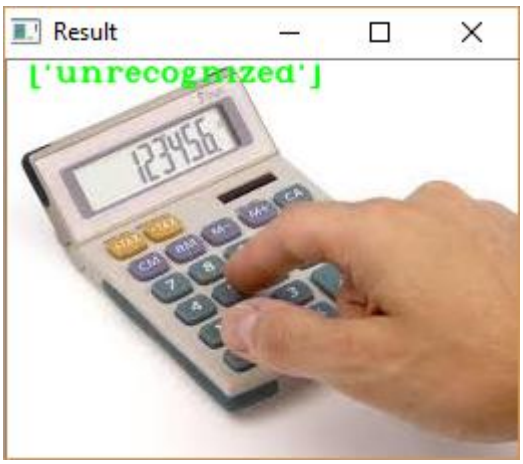
	1600		-	-	-
	2100		-	-	-
	100	Multimeter	1	2	7
	600		2	0	8
	1100		3	1	6
	1600		-	-	-
	2100		-	-	-
	100	Pen	3	6	1
	600		2	7	1
	1100		2	6	2
	1600		-	-	-
	2100		-	-	-
	100	Watch	1	8	1
	600		0	6	4
	1100		3	2	5
	1600		-	-	-
	2100		-	-	-

Table 7.7 Effect of varying number of classification classes in object recognition

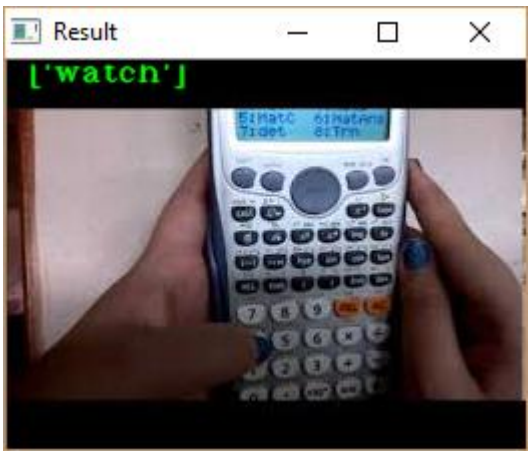
The result shows that no. of class variation caused rise in false recognition. This is because of the similarity between the keypoints obtained from the images from different classes. This is probably one of the hardest aspect of object recognition and still being researched on to effectively classify multiclass objects. With more keypoints, more false detections were also observed because the classifier is easily confused by the keypoints similarity between the classes.

7.2.6 Recognition Results

7.1 Object recognized

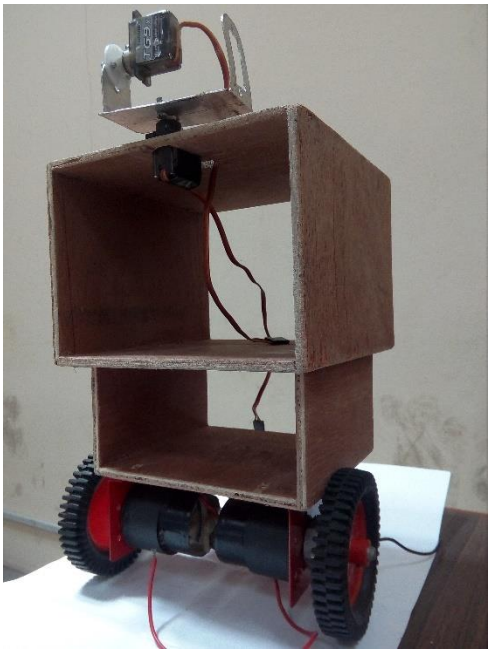


7.2 Object not recognized



7.3 Object false recognition

7.3 Hardware Results



7.4 Robot body structure

8. PROBLEMS ENCOUNTERED

8.1 Speech Recognition

i. The background noise degrades the recognition efficiency. The environment in which the training data was collected for the system had to be the same for an efficient recognition. During training, the noise caused a wide variation in data sets.

ii. Suitable feature vector size was required for an efficient recognition. An experiment had to be carried out to decide the required feature size.

iii. The Raspberry Pi does not have a built-in audio input port. So instead we had to use a USB soundcard to enable the audio input for processing.

iv. The setting up required for USB soundcard was problematic, and the soundcard and microphone seem to produce bad audio signal with noise embedded in the signal due to their poor quality which caused problem in recognition.

v. It wasn't easy to understand the visualization of MFCC values so as to interpret it.

8.2 Computer Vision

i. Bag of Features has no explicit use of configuration of visual word positions, so it was difficult to accurately recognize objects

ii. Bag of Features has poor localization of objects within an image, which made it problematic to track object. Instead we used feature based matching to track objects.

8.3 Hardware

i. During testing, the motor driver blew off consistently due to excessive current drawn by the motors. So we had to get motor driver of higher current rating.

8.4 Integration

i. Parallel processing of Speech and object recognition could not be carried out. Thus, we processed them in succession.

9. LIMITATIONS

- The system is trained to recognize only isolated word and the vocabulary is of two words.
- For feature comparison, DTW is followed which is less efficient than HMM. HMM is now widely used for feature comparison.
- The robot can only track the object down and no supplements has been provided to grab the objects.
- The object to be recognized is defined within the scope of a student’s room only and cannot recognize all objects.

10.BUDGET ANALYSIS

S.N.	Item	Qty.	Cost per Item (Rs.)	Total Cost (Rs.)
1.	Raspberry Pi with Camera Kit	1	15,000	15,000
2.	Li-Po Battery	1	5,000	5,000
3.	Li-Po Battery Charger	1	2,000	2,000
4.	Microcontroller	1	600	600
5.	DC Motors	2	1,500	3,000
6.	Servo Motors	2	1,500	3,000
7.	Motor Driver IC	3	300	900
8.	Buck Converter	1	550	500
9.	Miscellaneous	-	-	5,000
	Total			35,000

Table 10.1 Budget analysis

11.SCHEDULE ANALYSIS

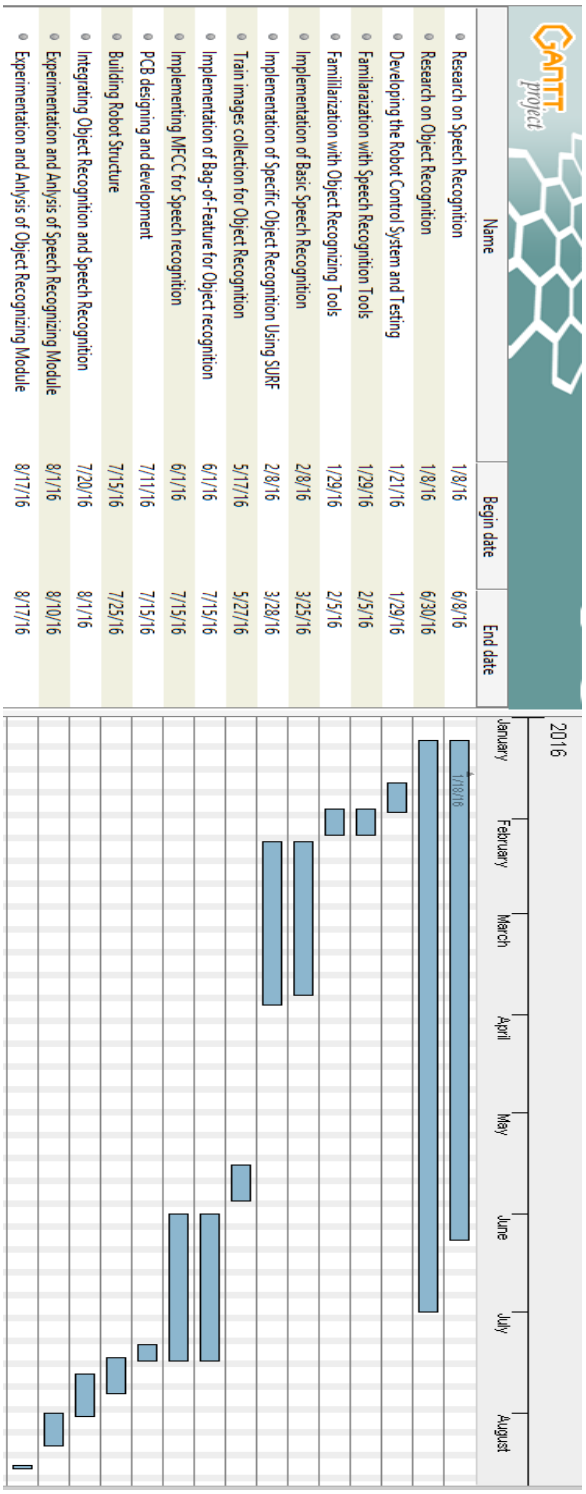


Figure 11.1 Gantt chart

12.FUTURE ENHANCEMENTS

For increasing the capacity of the system for a wider range of speech-recognition, our system has to be trained for different words and increase the vocabulary. This will enable the robots to identify a large number of words. Dynamic Time Warping method is used only for the template matching system which is considered to be inefficient than stochastic modelling. Hence, the state of art Hidden Markov Model can be used to increase the efficiency of the comparison. Presently, our system is able to identify only isolated words so this can be update to recognize continuous words or even implement natural language processing in the machine.

Similarly, for object recognition the datasets for training the classifier can be increased to recognize a large variety of objects. The collection of datasets could be a problem, but there exist many dataset resources online which can be used with appropriate license. This robot cannot track the object recognized using BoF method from the image which falls beyond the current scope of the project. For this, sliding window can be used to localize the recognized object within the image and track it.

The robot now is only designed to act with its head and body to fulfil the users command. Instead of just following the object, the robot can be developed with an end effector to provide the robot with an ability to grab the objects and carry out physical tasks such as bringing specific object. The robot can be designed to localize and navigate itself in the real world environment so that it can operate regardless of its position and obstacles.

13.CONCLUSION

This project aimed at implementing the speech recognition and object recognition in a machine using feature extraction techniques and provide a proof-of-concept of assistant robot. We have succeeded in building a mobile embedded system using Raspberry Pi that provides the robot two important senses, hearing and vision. To achieve this, we extracted speech feature using MFCC and object features suing SURF. The extracted features were matched with the reference database to perform the recognition. Though there are still many improvements that can be made to the system, this project succeeds in interfacing speech and vison to a machine.

REFERENCES

[1] B.-H. J. Lawrence Rabiner, Fundamentals of Speech Recognition, Princeton Hall International, 1993.

[2] W. H. J. Holmes, Speech Synthesis and Recognition, Taylor and Francis, 2004.

[3] s. s. Jamal Price, "Design an automatic speech," University of Maryland Estern.

[4] E. Gordon, Signal and Linear System Analysis, New York, USA: John Wiley & Sons Ltd., 1998.

[5] G. Z. Z. S. F. Zheng, "Comparison of Different Implementations of MFCC," *Journal of Computer Science & Technology,* vol. 16, pp. 582-589, 2001.

[6] A. B. I. Ahmad Kamarul, "Biomedical engineering labira-," UTM Jjohor.

[7] N. Batra, "Programatically understanding dynamic time warping," [Online]. Available: http://nipunbatra.github.io/2014/07/dtw/. [Accessed July 2016].

[8] E. Davies, in *Computer and Machine Vision: Theory, Algorithms, Practicalities*, Academic Press, 2012.

[9] J. M. a. S. Obdrazalek, "Object Recognition Methods Based On Transformation Covariant Features," Center for machine Perception, Czech Technical University, Prague.

[10] N. T. a. M. C. Marc T.law, Bag-of-Words Image Representation: Key Ideas and Further Insight, Switzerland: Springer International Publishing , 2014.

[11] H. Bay, A. Ess, T. Tuytelaars and L. V. Gool, "Speeded-Up Robust Features (SURF)," 2008.

[12] N. M. Drew Schmitt, "Object Classification and Localization Using SURF Descriptors," 2011.

[13] "Data Clustering Algorithm," [Online]. Available: https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm. [Accessed August 2016].

[14] "Introduction to SVM," [Online]. Available: http://docs.opencv.org/3.0-beta/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html. [Accessed August 2016].

[15] S. R. Gunn, “Support Vector Machines for Classification and Regression,” Faculty of Engineering,Science and Mathematics School of Electronics and Computer, 1998.

[16] “Non linear SVM,” [Online]. Available: http://svr-www.eng.cam.ac.uk/~kkc21/thesis_main/node14.html . [Accessed August 2016].

[17] “PyAudio,” [Online]. Available: https://people.csail.mit.edu/hubert/pyaudio/. [Accessed August 2016].

[18] “pickle — Python object serialization,” [Online]. Available: https://docs.python.org/2/library/pickle.html. [Accessed August 2016].

[19] “scikit-learn,” [Online]. Available: http://scikit-learn.org/stable/. [Accessed August 2016].

[20] “Matplotlib,” [Online]. Available: http://matplotlib.org/. [Accessed August 2016].

[21] “Raspberry Pi 2 Model B,” [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-2-model-b/. [Accessed May 2016].

[22] “Camera Module,” [Online]. Available: https://www.raspberrypi.org/products/camera-module/. [Accessed May 2016].

[23] D. M. Hawkins, “The problem of overfitting.,” *Journal of Chemical Information and Computer Sciences*, vol. 44, no. 1, pp. 1-12, 2004.

[24] “Practical Cryptography,” [Online]. Available: http://www.practicalcryptography.com/miscellaneous/machine-learning/tutorial-cepstrum-and-lpccs/. [Accessed July 2016].

[25] N.-l. S. M.-u. C. Tao Jia, “Moving object detection based on blob analysis,” IEEE, 2008.

[26] J. E. Solem, Programming Computer Vision with Python, United States: Creative Commons, 2012.

[27] J. Minichino and J. Howse, Learning OpenCV3 Computer Vision with Python, Birmingham: Packt Publishing Ltd., 2015.

[28] Y. LeCun, P. Haffner, L. Bottou and Y. Bengio, “Object Recognition with Gradient-Based Learning”.

[29] K. Grauman and B. Leibe, *Visual Object Recognition.*

[30] E. S. Gopi, Digital Speech Processing Using Matlab, Tamil Nadu, India: Springer, 2014.

[31] S. O. A. B. A. DRAPER, "INTRODUCTION TO THE BAG OF FEATURES PARADIGM FOR IMAGE CLASSIFICATION AND RETRIEVAL," 2010.

[32] E. Alpaydin, Introduction to Machine Learning, Massachusetts: The MIT Press, 2004.

[33] K. Ahuja and P. Tuli, "Object Recognition by Template Matching Using Correlations and Phase Angle Method," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 3, pp. 1368-1373, 2013.

[34] "Open Source Computer Vision," 18 December 2015. [Online]. Available: http://docs.opencv.org/3.1.0/d1/d73/tutorial_introduction_to_svm.html. [Accessed May 2016].

APPENDIX

A. Cepstrum

Cepstrum is basically a sequence of numbers that characterize a frame speech. The Cepstrum can be considered as being similar to the autocorrelation sequence. Let x(n) is the time domain signal, X(k) is the complex spectrum, P(k) is the power spectrum of x(n) and A(n) is the autocorrelation sequence of x(n).

By taking the DFT of x(n), we get the complex spectrum:

$$\text{DFT}(x(n)) \rightarrow X(k) \tag{A.1}$$

If we take the inverse DFT we get X(n) again.

To get the power spectrum, we take square of the absolute of the DFT of x(n).

$$|\,\text{DFT}(x(n))\,|^2 \rightarrow P(k) \tag{A.2}$$

IDFT of the power spectrum gives the autocorrelation sequence instead of the original sequence. And if we take the log of the power spectrum before the IDFT we actually get the spectrum. The plot of a speech signal, calculating power spectrum, autocorrelation and Cepstrum, is shown below. The signal is sampled at 8KHz frequency. [24]

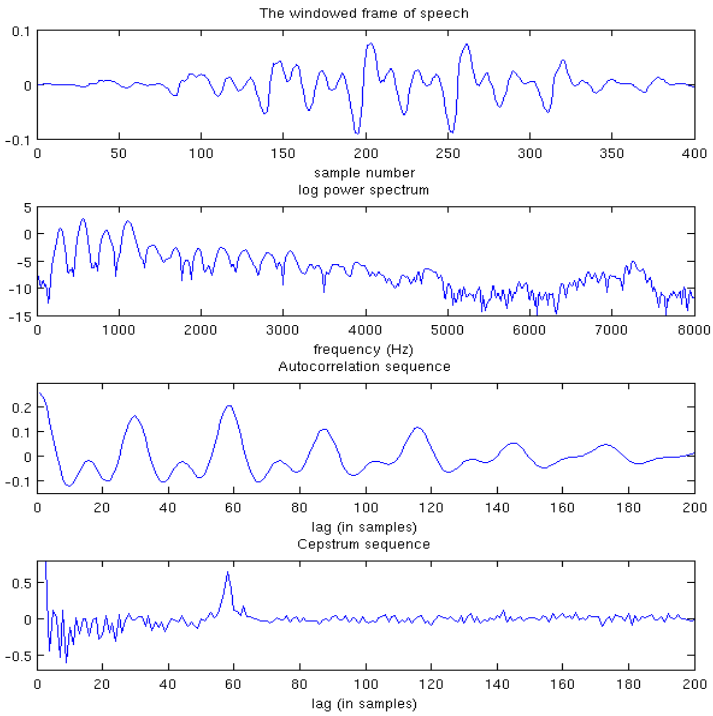


Figure A.1 Speech signal, power spectrum, autocorrelation sequence and cepstrum