

# **Hand Gesture Controlled Device Through Matlab**

*Dissertation submitted in  
Partial fulfilment of the requirement  
For the award of the degree of*

**Master of Technology**

*in*

**VLSI and Embedded System Design**

*by*

**Lovekesh Aggarwal**

**University Roll No. 2K14/VLS/14**

*Under the guidance of*

**Dr. S. Indu**

**Associate Professor,**

**Electronics and Communication Engineering Department, DTU**



**2014-2016**

**ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT**

**DELHI TECHNOLOGICAL UNIVERSITY**

**DELHI-110042, INDIA**



**Department of Electronics and Communication Engineering**

**Delhi Technological University**

**Delhi-110042**

**www.dce.edu**

### **CERTIFICATE**

This is to certify that the dissertation titled “**Hand gesture controlled device through matlab**” is a bonafide record of work done by **Lovekesh Aggarwal , Roll No. 2K14/VLS/14** at **Delhi Technological University, Delhi** for partial fulfilment of the requirements for the degree of Master of Technology in VLSI and Embedded System Design. This project was carried out under my supervision and has not been submitted anywhere else, either in part or full, for the award of any other degree or diploma to the best of my knowledge and belief.

Date: \_\_\_\_\_

**(Dr. S. Indu)**

**Associate Professor**

**Department of Electronics and Communication Engineering**

**Delhi Technological University**

## **ACKNOWLEDGEMENT**

This work would have not been possible without the support of many. First, it is my duty to thank God Almighty for making the completion of this research possible. Next, I would like to express my gratitude to my parents and my family for their continued support.

It gives me a great sense of pleasure to present the report of major project undertaken during M.tech fourth semester. This project is a continuation of minor project undertaken during M.tech second semester. I own special debt of gratitude to **Dr. S. Indu** (Associate Professor), Department of Electronics & Communication, Delhi Technological University, Delhi for her constant support and guidance throughout the course of my work. Her sincerity, thoroughness and perseverance have been a constant source of inspiration for me. It is only her cognizant efforts that my endeavours have seen light of the day.

I am also grateful to **Prof. Prem R. Chadha**, HOD, Electronics and Communication Engineering Department, DTU for his immense support. I will also like the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during my work.

I would also like to acknowledge Delhi Technological University for providing the right academic resources and environment for this work to be carried out.

Last but not the least I would like to express sincere gratitude to my friends and my colleagues for constantly encouraging me during the completion of work.

**Lovekesh Aggarwal**  
**University Roll no: 2K14/VLS/14**  
**M.TECH. (VLSI Design and Embedded System)**  
**Department of Electronics & Communication Engineering**  
**Delhi Technological University**  
**Delhi – 110042**

## INDEX

	Abstract	1
1.	Chapter 1 Introduction	2
	1.1 Overview	2
	1.2 Motivation	5
	1.3 Objectives to be achieved	5
	1.4 Project Description	6
	1.5 Issues and Challenges	7
	1.6 Project Scope	7
	1.7 Structure of Thesis	8
2	Literature Review	9
	2.1 Introduction	9
	2.1.1 Human Interface Device	9
	2.1.2 Hand Gesture	9
	2.1.3 Image Processing	10
	2.1.4 Feature Extraction	10
	2.1.5 Classification	10
	2.2 State of Art	11
3	Chapter 3 Image Acquisition and Database	13
	3.1 Introduction	13
	3.2 Hardware	13
	3.3 Software	14
	3.4 Image Database	14
4	Chapter 4 Image Processing	17
	4.1 Reading an Image	17
	4.2 Conversion of Image into Binary Image	17
	4.3 Extracting Skin Color	18
	4.4 Median Filtering	19
	4.5 Filtering the Noise and Removing small objects	20
	4.6 Other Processing Steps	21
	4.6.1. Centroid	21
	4.6.2 Orientation of Image	22
	4.6.3 Drawing a Boundary Box	22
	4.6.4 Thumb Detection	22

5	Features Extraction	24
	5.1 Objectives	24
	5.2 Theoretical Background	24
	5.3 HOG Descriptor	24
	5.4 Other Features	26
6	Classifiers	29
	6.1 Introduction	29
	6.2 K-Nearest Neighbours	29
	6.2.1 KNN Algorithm	30
	6.3 Other Classifiers	31
7	Applications	32
	7.1 Working Application	32
	7.1.1 Atmega 8L Microcontroller	33
	7.1.2 Zigbee	36
	7.1.3 ULN2803 Relay Driver	39
	7.1.4 LM 35 Series Temperature Sensor	39
	7.1.5 USB TO TTL	40
	7.1.6 LCD	41
	7.1.7 Schematic of final Hardware	42
	7.2 Other Applications	43
8	Results and Conclusion	44
	8.1 Results	44
	8.2 Conclusion	49
References		

## LIST OF TABLES

1.	TABLE 3.1 TRAINING SET	15
2.	TABLE 7.1 BAUD RATE CALCULATION	34
3.	TABLE 7.2 ADC PRESCALAR SELECTIONS	36

## LIST OF FIGURES

1.	FIG 1.1 GRAPHICAL OUTLINE OF THESIS	8
2.	FIG. 2.1 SCHEMATIC VIEW OF GESTURE RECOGNITION SYSTEM	10
3.	FIG 3.1 LOGITECH C310 HD CAMERA	13
4	FIG. 4.1 CONVERTING RGB DIRECTLY TO B&W IMAGE	18
5	FIG. 4.2 CONVERTING RGB TO B&W IMAGE BY SKIN COLOR EXTRACTION	19
6	FIG. 4.3 REMOVING SALT & PEPPER NOISE	19
7	FIG 4.4 CENTROID OF IMAGE	22
8	FIG 4.5 THUMB DETECTION	23
9	FIG. 5.1 64X128 PIXELS WINDOW(HOG DESCRIPTOR)	24
10	FIG. 5.2 CROPPED IMAGE OF CELL C1(HOG DESCRIPTOR)	25
11	FIG. 5.3 GRADIENT ORIENTATION OF CELL C1(HOG DESCRIPTOR)	25
12	FIG. 5.4 HISTOGRAM OF ORIENTED GRADIENT H1 FOR CELL C1	25
13.	FIG. 5.5 ACCUMULATE HOGs OVER THE BLOCK TO CREATE HOG DESCRIPTOR	25
14.	FIG. 5.6 CENTROID METHOD OF FINDING FINGERS	27
15.	FIG.6.1 SIMULATION TIME IN CASE OF SVM	29
16.	FIG.6.2 SIMULATION TIME IN CASE OF KNN	29
17.	FIG. 6.3. EXAMPLE OF K-NN CLASSIFICATION	30
18	FIG. 7.1 FINAL HARDWARE	32
19	FIG. 7.2 PIN DIAGRAM OF ATMEGA 8L	33
20	FIG. 7.3 UCSRA REGISTER	34
21	FIG. 7.4 UBRR REGISTER	34
22	FIG. 7.5 UCSRB REGISTER	35

23	FIG. 7.6 ADMUX REGISTER	35
24	FIG. 7.7 ADSCRA REGISTER	35
25	FIG. 7.8 ULN 2803 PIN DIAGRAM	39
26	FIG. 7.9 INTERNAL STRUCTURE OF ULN2803	39
27	FIG. 7.10 USB TO TTL	41
28	FIG.7.11 LCD	41



## LIST OF FLOW DIAGRAM

1.	PROPOSE MODEL	6
2.	FLOW CHART OF MEDIAN FILTER ALGORITHM	20
3.	NUMBER OF PEAKS	26
4.	NUMBER OF FINGERS	38

## **ABSTRACT**

We propose a vision based Hand gesture Recognition system that recognizes hand gestures that is useful especially for physically challenged people, and provides recognized character or number as text and corresponding sound. A lot of work has been done towards developing intelligent and natural interfaces between human being and computer systems. Hand Gesture recognition is a fast developing area in the field of Human Computer Interaction. Present day system developers not only aim to develop the fully functional systems, but also aim for systems with more intuitive interfaces, allowing for easier user/system interaction. The gesture recognition is a process in which the gestures or postures of human body parts are identified and are used to control computer and other electronic appliances. The most contributing reason for the emerging gesture recognition is that they can create a simple communication path between human and computer called as HCI (Human Computer Interaction). This thesis is confined to a identification of hand postures to establish a man-machine interaction and to control a simple hardware wirelessly. The hand region in the image is detected and data regarding the detected image is stored in different groups. In this approach, the input which is an image or a frame from a video can be obtained from web camera or any other camera. Data consists of HOG (Histogram of an Oriented Graph) extracted features and different classification techniques have been used to classify the detected image from the trained groups. Skin detection system is more efficient as compare to converting color image into binary image. This is a simple yet efficient approach.

## CHAPTER 1 INTRODUCTION

### 1.1 OVERVIEW

Since the introduction of the most common input computer devices not a lot have changed. This is probably because the existing devices are adequate. It is also now that computers have been so tightly integrated with everyday life, that new applications and hardware are constantly introduced. The means of communicating with computers at the moment are limited to keyboards, mice, light pen, trackball, keypads etc. These devices have grown to be familiar but inherently limit the speed and naturalness with which we interact with the computer. As the computer industry follows Moore's Law since middle 1960s, powerful machines are built equipped with more peripherals. Vision based interfaces are feasible and at the present moment the computer is able to "see". Hence users are allowed for richer and user-friendlier man-machine interaction. This can lead to new interfaces that will allow the deployment of new commands that are not possible with the current input devices. Plenty of time will be saved as well. With current advances in technology, we see a rapidly increasing availability, but also demand, for intuitive HCI. Devices are not only controlled by mouse and keyboard anymore, but we are now using gesture controlled devices in public areas and at our homes. Distant hand gestures in particular removed the restriction to operate a device directly; we can now interact freely with machines while moving around. In this work, we are interested in HCI that does not force users to touch a specific device or to wear special sensors, but that allows for unrestricted use. While there is a great variety of HCI techniques to interact with distant virtual objects, e.g. to select menu entries, there is still a lack for intuitive and unrestricted text input. Although there are ways to input text by using a virtual keyboard on a display or by speech recognition, there are situations where both are not suitable: the rest requires interaction with a display. Users must speak which is not always possible. demanding on the surroundings, especially for physically challenged people. In present days a lot of research has been taking place to develop a simple and natural interaction between human and computer. The ways in which one can interact with computer are either by using devices like keyboard, mouse or via audio signals, while the former always needs a physical contact and the latter is prone to noise and disturbances . With the advancement in image processing a simple and efficient mode of communication with computer can be developed. By using image processing techniques a vision based interaction can be established. Gesture recognition and processing is the best example for vision based interaction . Gesture recognition is a field which interprets the gestures of human body parts like face and hands. Gesture recognition interfaces which are used as way of natural communication between man and machine give rise to a variety of applications such as equipment free remote control systems, sign language interpretation and many more . The keyboard and mouse are currently the main interfaces between man and computer. In other areas where 3D information is required, such as computer games, robotics and design, other mechanical devices such as roller-balls, joysticks and data-gloves are used. Humans communicate mainly by vision and sound, therefore, a man-machine interface would be more intuitive if it made greater use of vision and audio recognition. Another advantage is that the

user not only can communicate from a distance, but need have no physical contact with the computer. However, unlike audio commands, a visual system would be preferable in noisy environments or in situations where sound would cause a disturbance. The visual system chosen was the recognition of hand gestures. The amount of computation required to process hand gestures is much greater than that of the mechanical devices, however standard desktop computers are now quick enough to make this project — hand gesture recognition using computer vision — a viable proposition. A gesture recognition system could be used in any of the following areas:

- Man-machine interface: using hand gestures to control the computer mouse and/or keyboard functions. An example of this, which has been implemented in this project, controls various keyboard and mouse functions using gestures alone.
- 3D animation: Rapid and simple conversion of hand movements into 3D computer space for the purposes of computer animation.
- Visualisation: Just as objects can be visually examined by rotating them with the hand, so it would be advantageous if virtual 3D objects (displayed on the computer screen) could be manipulated by rotating the hand in space
- Computer games: Using the hand to interact with computer games would be more natural for many applications.
- Control of mechanical systems (such as robotics): Using the hand to remotely control a manipulator.

The impendent of virtual environments brings in a whole new set of problems for user interfaces. The unveiling of 3D objects and worlds in which the user is engrossed allows such people as scientists, engineers, doctors and architects to envision composite structures and systems with eminent degrees of quality and naturalism. Shutter glasses furnish a stereo or 3D view of the scene, which is no longer confined to a desktop monitor, but may be a large table, projection screen or room. The limiting component in these systems currently is the fundamental interaction. Virtual environments seek to produce a world where the interaction experiences are real. Current mechanical, acoustic and magnetic input devices track the user and provide control of movement, selection and manipulation of objects in virtual scenes. Several tools are purported and used so far to make such interaction more and more prompt and effortless. Touch screens are the most widely used example: though the ramification of the underlying system is hidden from the user, and makes it possible for a user to point to the choices as he could do in real life. The cost associated to it is the major limitations of the aforesaid technology other limitations may be size, requirement of a physical location, and other intrinsic limitation to 2D. Other more innovative devices proposed for virtual reality include gloves or wearable tools such as mechanical sensors, actuators and micro cameras [1]. They are capable to handle 3D worlds, making it natural and realistic, and also provide in some implementations tactile sensations. Regrettably, their cost is usually very high, and thus the user acceptance confined, hence making them more desirable for professional applications such as a flight simulator or remote surgery equipment .

However these interactions are often limited and non rational, while the devices are awkward, unmanageable and prone to distortion from the physical environment. We are interested in formulating an alternative, natural interface that more intimately models the way we interact

with the real world. The user should be able to reach out, grab, point and move 3D objects just as we do with real objects.

These challenges open a new direction for human-computer interaction which combined with computer vision techniques and it is possible to build an advanced input devices. The computer vision devices can be implemented and upgrade to the new input devices in the future. It gives the input command to the computer rather than just a function of taking photo or record video. We can do more implementation to transform the computer vision devices to become an input command device to reach the function as keyboard or mouse. One of the ways to give signal to computer vision devices is by using hand gesture. More specifically hand gesture is used as the signal or input modality to the computer. Certain signal can be recognized by computer as an input of what computer should do. These will benefits the entire user without using a direct device and can do what they want as long as the computer vision device can sense it. These make computer user easier than using the keyboard or mouse. The future computer or laptop may eliminate the use of keyboard and mouse by substituting with a vision-based interpretation devices.

Interaction between humans comes from different sensory modes like gesture, speech, facial and body expressions [2]. A primary goal of virtual environments (VE) is to provide natural, efficient, powerful, and flexible interaction. Gesture as an input modality can help meet these requirements. Human gestures are certainly natural and flexible, and may often be efficient and powerful, especially as compared with alternative interaction modes. This chapter will cover automatic gesture recognition, particularly computer vision based techniques that do not require the user to wear extra sensors, clothing or equipment. The specifications also involve accuracy of detection and recognition for real life applications [3].

The traditional two-dimensional (2D), keyboard- and mouse- oriented graphical user interface (GUI) is not well suited for virtual environments. Synthetic environments provide the opportunity to utilize several different sensing modalities and technologies and to integrate them into the user experience. Devices which sense body position and orientation, direction of gaze, speech and sound, facial expression, galvanic skin response, and other aspects of human behavior or state can be used to mediate communication between the human and the environment. Combinations of communication modalities and sensing devices can produce a wide range of unimodal and multimodal interface techniques. The potential for these techniques to support natural and powerful interfaces for communication in VEs appears promising. If interaction technologies are overly obtrusive, awkward, or constraining, the user's experience with the synthetic environment is severely degraded. If the interaction itself draws attention to the technology, rather than the task at hand, or imposes a high cognitive load on the user, it becomes a burden and an obstacle to a successful VE experience. Therefore, there is focused interest in technologies that are unobtrusive and passive.

In this paper, we are concerned in recognition of hand gestures. The inputs are taken from web camera or any other camera. These images are subjected to modest MATLAB algorithms which pre-process the image and detect the skin. The platform used for the

programming is MATLAB. Initially, we used R2007a but that raised problem in executing the commands of video capturing and taking snapshots from running video. Then finally we moved to R2013a version which supports these commands and it runs in 64-bit computers.

## **1.2 MOTIVATION**

I had chosen this project with an interest of learning the direct interaction of humans with the consumer electronic devices . This takes the user experience to a whole new level. The gesture control technology would reduce our dependence on the age old peripheral devices hence it would reduce the overall complexity of the system. Initially this technology was considered in the field of gaming (like Xbox Kinect), but the application of motion/gesture control technology would be more diverse if we apply it to our other electronics like computers ,televisions, etc., for our day to day purposes like scrolling , selecting, clicking etc. So, we decided to build an introductory software implementation of the device which would eventually provide instructions to a hardware wirelessly.

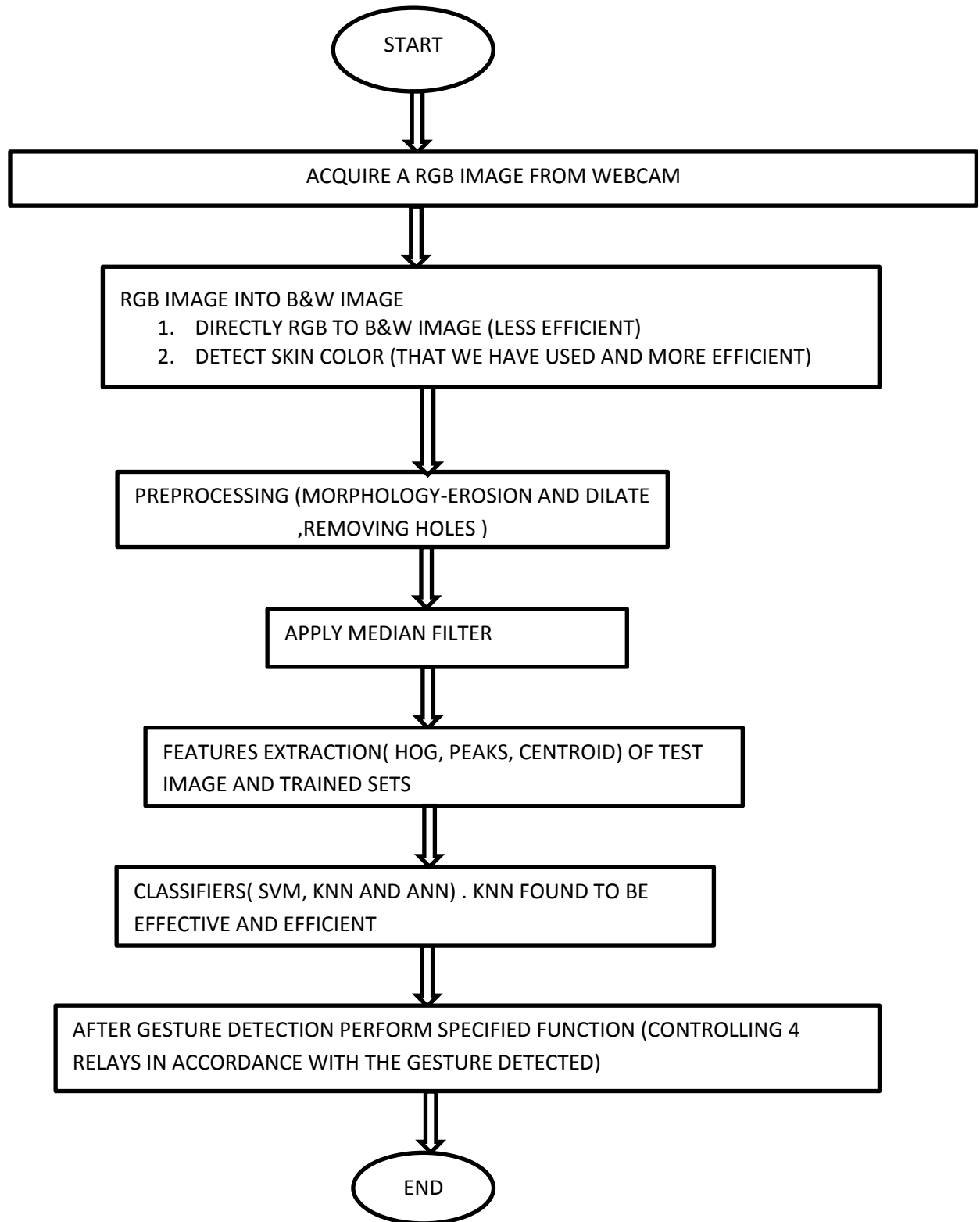
## **1.3 OBJECTIVES TO BE ACHIEVED:**

The main objective of this paper shall be

1. To develop a hand gesture recognition system.
2. That system should be capable enough to detect hand gestures correctly and to translate detected gestures into set of instructions provided to a hardware wirelessly.

## 1.4 PROJECT DESCRIPTION

### PROPOSED MODEL



## 1.5 ISSUES AND CHALLENGES

The first challenge was to detect the hand correctly with the help of webcam . I need a software capable of detecting the object by preprocessing that image. Many are available but I decided to go with the MATLAB software because of its availability and also because of the lots of help content available on the internet regarding this software. First I try to convert the RGB image directly Black and white image but converting directly RGB image into Black and White image resulted into an image which is little distorted .So I had to learn about skin detection techniques and other image processing techniques like Image smoothening, Noise removal and Reduction. Now, after detecting the image of hand correctly and mapping the gestures, I had to learn to use Windows API in order tune the software with the nircmd.exe. It was an easy task but in order to make a good application of my system my mentor advised me to make a hardware which work in accordance with the detected gestures. So, in short, there was a steep learning curve.

As simple webcam is unable to take good quality snapshot of my hand so there was a need to go for high end webcam and also to optimize our software and its functionality . Also, for testing my project I had a primary requirement of having a black background .

The main problems I normally encounter in making a hand gesture recognition system are camera orientation and the reliability of the system Considering the above mentioned points in this paper we propose a better system which shall take care of all the technical shortcomings.

1. It will detect the gesture correctly even if there is slightly change in the orientation of the gesture detected.
2. It will detect the gesture correctly in different backgrounds.
3. Control hardware wirelessly (controlling 4 relays).

## 1.6 PROJECT SCOPE

After recognizing the gesture MATLAB will perform specified task. Example of tasks

- Paticular gesture can write an alphabet in Microsoft word and save that word file in a specified folder
- Detected gesture can pronounce an corresponding alphabet and pronunciation of alphabet can be done in two ways:
  - 1) By using google translator.
  - 2) By using window system
- Can perform system functions on windows operating system using nircmd.exe like stop and play music, lower down the volume , move the pointer etc.
- Can control hardware wirelessly.



## 1.7 STRUCTURE OF THESIS

In second chapter there is a brief description of previous works that have been done in this area . Third chapter is all about the software and hardware that I have used in this project. This chapter deals with the image acquisition devices and gives information about the trained data set i.e. the gestures which my system is capable of detecting. In fourth chapter various preprocessing techniques that I have adopted during the course of my project and also the techniques which I found best among the rest and that I have used in this project . Information related to the extracted features from the preprocessed image has been discussed in the fifth chapter. HOG is sufficient for my project that I have found during the later course of my project has been described in this chapter. And also there is a brief description about the extraction features that I had earlier employed before I start using HOG descriptors. In sixth chapter different classifiers have been discussed and brief description about the classification technique that I have adopted and it is tried to explain why I have chosen this classifier. In seventh chapter I have discussed about different applications of my project , the ones which I have performed and also about the potential applications of my system. In eighth chapter the results and conclusion I got after implementing my system have been discussed.

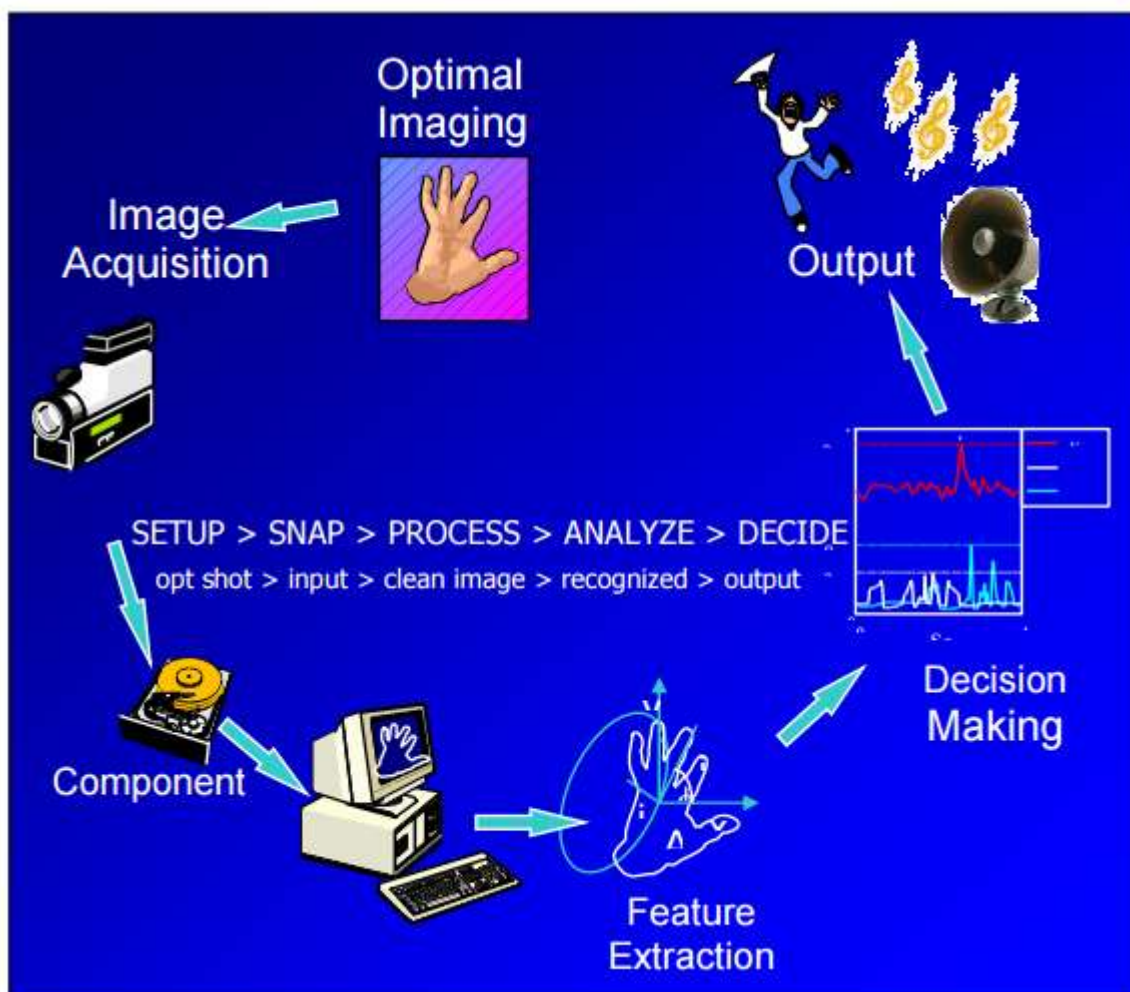


FIG 1.1 GRAPHICAL OUTLINE OF THESIS

## **CHAPTER 2 LITERATURE REVIEW:**

### **2.1 INTRODUCTION**

#### **2.1.1 HUMAN INTERFACE DEVICE**

A human interface device or HID is a type of computer device that interacts directly with and most often takes input from humans and may deliver output to humans . Common HIDs are:- Keyboard Mouse, Trackball, Touchpad, Laptop eraser Graphics tablet Joystick, Gamepad The working group named as the Human Interface Device class at the suggestion of Tom Schmidt of DEC because the proposed standard supported bi-directional communication. To have interactive input device is important to meet up problems come with it. For example using remote control is very handy. But by losing the remote control device, the purpose of easing started to disappear. Another example yet different perspective is to have a very dynamic system that can be configured programmatically instead of hardware mending.

#### **2.1.2 HAND GESTURE**

When we look in a wider extent, gestures came out from human body is natural. From Concise Oxford English Dictionary, gesture is a movement of part of the body, especially a hand or head, to express an idea or meaning . As for this project, the development portion is to develop a new practical technology for alternative human interface. This gesture definition has been narrowed down to only hand and is given particular attention on details that Asian hands has. Human do most things by using hand. The uniqueness covers from picking the smallest object to even do specialize work or even giving meaningful gestures to convey information. Take traffic police for instance, by just moving his hand and come out with different gesture each explaining go, stops or wait to the drivers.

There are two basic approaches in gesture recognition:

1. The top-down approach, where a previously created model of collected information about hand configurations is rendered to some feature in the image coordinates. Comparing the likelihood of the rendered image with the real gesture image is then used to decide whether the gesture of the real image corresponds to the rendered one.
2. The bottom-up approach, which extracts features from an input image and uses them to query images from a database, where the result is based on a similarity measurement of the database image features and the input features.

The disadvantage of the first approach is that it seems to use a high computational effort in order to achieve robust recognition. The second approach however requires an adequate preprocessing in order to achieve a reliable segmentation. This report mainly keeps the focus on the latter approach since this seems to be the commonly used one. The whole process of static gesture recognition can be coarsely divided into four phases, as shown in Figure 2.1. Each phase performs a specific task, whose result is passed to the

next phase. The commonly used techniques for each phase are described in the following subsections.

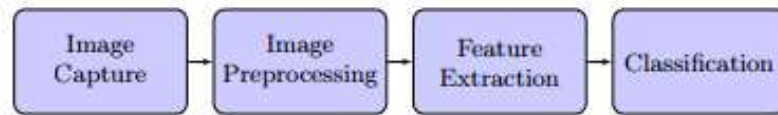


FIG. 2.1 SCHEMATIC VIEW OF GESTURE RECOGNITION SYSTEM

### 2.1.3 IMAGE PROCESSING

Image processing is any form of information processing for which the input is an image, such as photographs or frames of video; the output is not necessarily an image, but can be for instance a set of features of the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal processing techniques to it

### 2.1.4 FEATURE EXTRACTION

The aim of this phase is to find and extract features that can be used to determine the meaning of a given gesture. Some interesting techniques are presented later on in this section. Ideally such a feature, or a set of such features, should uniquely describe the gesture in order to achieve a reliable recognition. Therefore, different gestures should result in different, good discriminable features. Furthermore, shift and rotation invariant features lead to a better recognition of hand gestures even if the hand gesture is captured in a different angle.[4],[5],[6],[7]

### 2.1.5 CLASSIFICATION

The classification represents the task of assigning a feature vector or a set of features to some predefined classes in order to recognize the hand gesture. In previous years several classification methods have been proposed and successfully tested in different recognition systems. In general, a class is defined as a set of reference features that were obtained during the training phase of the system or by manual feature extraction, using a set of training images. Therefore, the classification mainly consists of finding the best matching reference features for the features extracted in the previous phase. This section presents an overview of the most commonly used methods in different hand gesture recognition systems. [7],[8].

## 2.2 STATE OF ART

In earlier days hand gesture detection was done using mechanical devices to obtain information of the hand gesture [9]. One of the most widely used and accepted examples for hand gestures recognition is data glove. Evolution of computer hardware improved a lot of in present scenario this also effects the performance of computing. Enhancements of gesture recognition has replaced the role of data gloves to non wearable devices due to its naturalness without using any device this is quite user friendly in human computer interaction. One of the major drawbacks of data glove is that it is cumbersome with the limitation of hand movement.

Also vision is one of the major six physical senses that computer must be instantiated perceptibly when communicated to humans [10]. So vision based approaches are preferred more than wearable devices in hand gesture recognition. Generally there are three stages in most of the gesture recognition systems. The three stages may be enumerated as image pre –processing tracking and recognition stage [11] .In tracking, there are several researchers who have done the similar research like Viola-Jones based cascade classifier, commonly used for face tracking in rapidly image processing [12]. Cascade classifiers are currently considered more robust pattern detection against the noises and lighting conditions as well [13].

For tracking Viola-Jones and several other researchers have developed algorithms used for face tracking in rapid image processing like HAAR cascade classifier. This is presently one of the robust detection techniques under different constraints like noise [13].Gesture as input of human computer interaction based applications is an emerging field in which many researchers have worked and proposed different practical techniques. Jain [14] implemented a vision based hand gesture pose estimation based application for mobile devices. Pavlovic et al. [15] accomplished in their work that the gestures of users must be explained logically for developing a good human computer interaction based system .Though the present technologies of gesture recognition are not feasible in providing the logical explanations to the human gestures. Some of the major challenges evolved in due course of time are the complexness and robustness of human gestures. Another hand gesture recognition method based on input-output Hidden Markov Models of tracking skin color blobs was proposed by Marcel et al. [16]. Controlling VLC media player using hand gesture recognition is done in real time environment using vision based techniques [17]. The sign language tutoring tool studied by Aran et al. [18] which their research designed to teaching the fundamental of the sign language in interactive way.

Liu and Lovell [19] implemented a technique for real time tracking of hand capturing gestures with the use of a web camera, personal computer and image processing algorithms making it more users friendly. Chen et al. [20] implemented hidden Markov model technique for training the hand gesture for recognizing the hand postures. Nevertheless, this model is more complicated in training the hand gesture equated with Cascade classifiers. Lee et al. [21] developed a Virtual Office Environment System (VOES), in which avatar is used navigate and interact with other participants.

Contemporary works in hand gesture recognition by many researchers show that hand gesture system can also be practically implemented into several type of application

systems and various environment . Ahn et al. [22] developed an interactive way of slide show presentation system in the virtual environment. Research in hands, gestures and movement helps in developing models of the human body. This makes it possible to solve the challenges from mathematical viewpoint. How so ever, these techniques proposed are excessively complex and sophisticated for typical application scenarios. Generally, pattern recognition methodologies are capable of solving the problem with humbler hardware and computation necessities. In the present research effort, we will consider these aspects by taking it as a reference to a smart interaction environment of virtual object manipulation and control. Here the user can execute different actions that translate into a command in an intelligent system and further execute the user requirements into practical actions.

## CHAPTER 3 IMAGE ACQUISITION AND DATABASE

### 3.1 INTRODUCTION

The input images are captured by a webcam placed on a table or laptop. The system is demonstrated on a conventional PC/ Laptop computer running on Intel I3 Processor with 4GB of RAM. The system developed can process hand gestures at an acceptable speed. Given a variety of available image processing techniques and recognition algorithms, we have designed our preliminary process on detecting the image as part of our image processing. The system starts by capturing a hand image from signer with a webcam setup towards certain angle with black background.

### 3.2 HARDWARE

Web camera captures motion images with resolution 640 times 480 pixels and capability capturing 30 frames per seconds. Preliminary, this particular camera can go up to 25 frames per second only. Now, the capability is bigger and not impossible to increase in the near future. To enhance the capturing ability, the camera used is using CCD as a sensor. CCD stands for Charge Couple Device, which is not regarded as more superior against CMOS or Complementary Metal Oxide Semiconductor type of sensor. With three layer of CCD, each capturing each plane consisting of Red, Blue and Green respectively giving a more vibrant and sharp images. This factor also contributing to the resolution and frames per second performance on capturing images. CCD is less prone to noise.

In this project I have used Logitech HD webcam C310[23] that have a capability of acquiring 5MP still photos as my laptop webcam unable to provide me a good quality pictures and also that with USB camera it is much easier for me to focus a required area as compared to Laptop camera . These were the two reasons which made me switch to USB camera.



FIG 3.1 LOGITECH C310 HD CAMERA

### **3.3 SOFTWARE**

For image processing I have used software MATLAB. The name MATLAB stands for matrix laboratory. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

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

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or Fortran. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.[24]

### **3.4 IMAGE DATABASE**

Image database can have different formats. Images can be either hand drawn, digitized photographs or a 3D dimensional hand. Photographs were used, as they are the most realistic approach. Images came from two main sources. Various databases on the Internet and photographs I took with a digital camera. This meant that they have different sizes, different resolutions and sometimes almost completely different angles of shooting. The database itself was constantly changing throughout the completion of the project as it was it that would decide the robustness of the algorithm. Therefore, it had to be done in such way that different situations could be tested. I have developed a specific application for my system, before reaching this application I have developed many other applications which also I will discuss in later chapters.

TRAINING SET TABLE 3.1



It will off the '1<sup>st</sup>' relay



It will on the '1<sup>st</sup>' relay



It will off the '2<sup>nd</sup>' relay



It will on the '2<sup>nd</sup>' relay



It will off the '3<sup>rd</sup>' relay





It will on the '3<sup>rd</sup>' relay



It will off the '4<sup>th</sup>' relay



It will on the '4<sup>th</sup>' relay

## CHAPTER 4 IMAGE PROCESSING

The basic aim of this chapter is to optimally prepare the image obtained from the previous chapter in order to extract the features in the next phase

### 4.1 READING AN IMAGE

I have used following commands to access a training folder and also to read an image.

```
(A) folder=dir('C:\Users\lovekesh\Desktop\matlabunderstandmtech\Training');
```

This command will create a structure having name folder with fields

name  
date  
bytes  
isdir  
datenum

From this command we have nx1 structure, where n = number of folders +2;

Extra 2 folders are

‘.’ Current location  
‘..’ previous location

```
(B) im= imread(str)
```

reads a grayscale or color image from the filespecified by the string FILENAME. If the file is not in the current directory, or in a directory on the MATLAB path, specify the full pathname.

### 4.2 CONVERSION OF IMAGE INTO BINARY IMAGE

We have converted the image into a binary image by using a function im2bw. In this we basically find a particular level by using a matlab command graythresh

This *MATLAB* function computes a global threshold (level) that can be used to convert an intensity image to a binary image with im2bw i.e. intensities above threshold level will be considered as 1(white) and all the intensities below threshold level will be considered as 0(black).

```
gr=rgb2gray(im);
```

It will convert RGB Image into GRAY image.

Then we find threshold and convert that gray image into black and white image or binary image

```
th=graythresh(gr);
```

```
bw=im2bw(gr,th);
```

But above process sometimes gives distorted image and also it is giving more than one objects so in that case it would be very difficult to find boundary if we need to but In this project I don't need to find boundary conditions because I have used HOG

features . Earlier, before extracting HOG features I had found boundary conditions to find the peaks and if we have more than one object and also if the boundary has many discontinuities it is very difficult to have correct boundary coordinates.



FIG. 4.1 CONVERTING RGB DIRECTLY TO B&W IMAGE

### 4.3 EXTRACTING SKIN COLOR

Skin detection can be defined as detecting the skin colour pixels in an image. It is a fundamental step a wide range of image processing application such as face detection, hand tracking and hand gesture recognition. Skin detection using colour information has recently gained a lot of attention, since it is computationally effective and provides robust information against scaling, rotation and partial occlusion. Skin detection using colour information can be a challenging task, since skin appearance in images is affected by illumination, camera characteristics, background and ethnicity.

So first step towards skin color extraction is to do color balancing.

$$mR = 1/(\text{mean}(\text{mean}(R)));$$

$$mG = 1/(\text{mean}(\text{mean}(G)));$$

$$mB = 1/(\text{mean}(\text{mean}(B)));$$

Smallest Avg Value (MAX because we are dealing with the inverses)

$$\text{maxRGB} = \max(\max(mR, mG), mB);$$

Calculate the scaling factors

$$mR = mR/\text{maxRGB};$$

$$mG = mG/\text{maxRGB};$$

$$mB = mB/\text{maxRGB};$$

Secondly I took snapshot of my hand and by using command imtool I tried to find the Cb and Cr pixels of that image by converting RGB image into YCbCr .

Thirdly by using find command I tried to find those location which matches to my skin tone turning that are into white and outer region black.



FIG. 4.2 CONVERTING RGB TO B&W IMAGE BY SKIN COLOR EXTRACTION

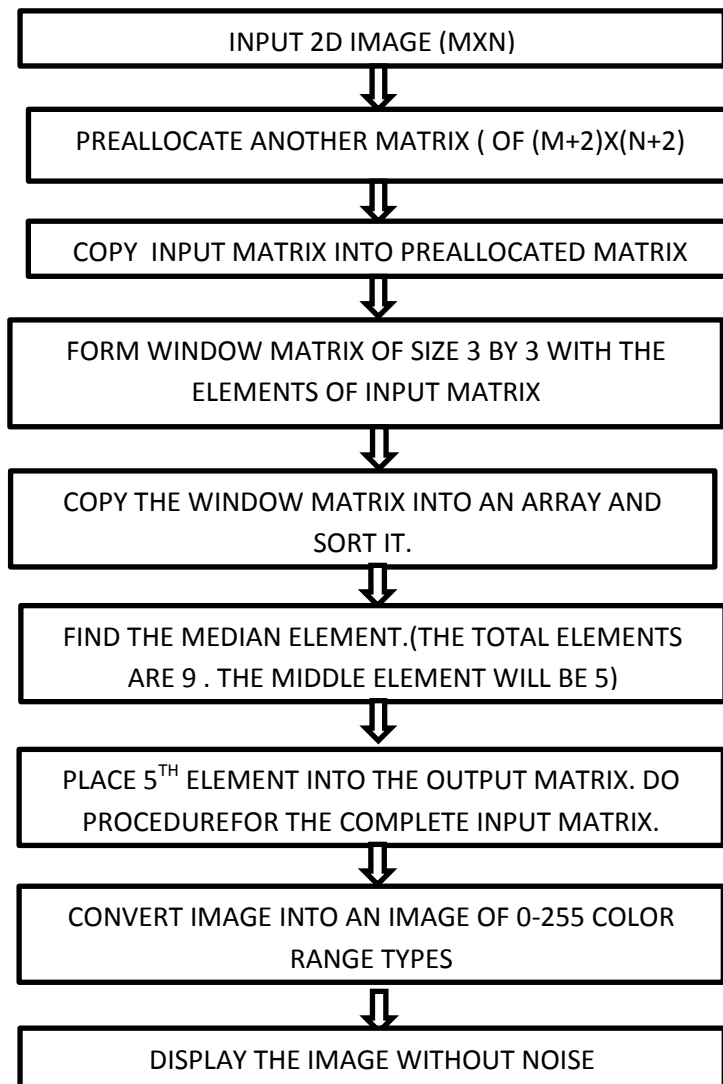
#### 4.4 MEDIAN FILTERING

To clear salt and pepper noise in which salt is white '1' and pepper black '0'.



FIG. 4.3 REMOVING SALT & PEPPER NOISE

## FLOWCHART OF MEDIAN FILTER ALGORITHM



### 4.5 FILTERING THE NOISE AND REMOVING SMALL OBJECTS

For filtering the noise we have used a two matlab functions

- `bwareaopen`
- `imfill`

To get the best accurate number of the object detected in the image, all the areas other than the pointer need to be removed. To do this, we have used MATLAB's "*bwareaopen*" function.

Function can be used as;

$$BW2 = bwareaopen(BW, P)$$

Where; it removes from a binary image all connected components (objects) that have fewer than P pixels, producing another binary image, BW2. We set a threshold 500 pixels for this study.

Although following processing steps I have not used in my final project but I had used them before using HOG features extraction during initial days of implementing my system.

#### 4.6. OTHER PROCESSING STEPS

##### 4.6.1. CENTROID

In mathematics and physics, the centroid or geometric center of a two-dimensional region is, informally, the point at which a cardboard cut-out of the region could be perfectly balanced on the tip of a pencil (assuming uniform density and a uniform gravitational field). Formally, the centroid of a plane figure or two dimensional shape is the arithmetic mean ("average") position of all the points in the shape. The definition extends to any object in  $n$ -dimensional space: its centroid is the mean position of all the points in all of the coordinate directions. we calculate the centroid for differentiating various hand pattern based on number of fingers. By assuming number of fingers and finger position as distinct from location of the centroid in the hand image we can differentiate many hand gestures. The centroid of a non-self-intersecting closed polygon

defined by  $n$  vertices  $(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})$  is the point  $(C_x, C_y)$ , where

$$C_x = \frac{1}{6A} \sum_{i=0}^{n-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$

$$C_y = \frac{1}{6A} \sum_{i=0}^{n-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$

and where  $A$  is the polygon's signed area

$$A = \frac{1}{2} \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i)$$

In these formulas, the vertices are assumed to be numbered in order of their occurrence along the polygon's perimeter, and the vertex  $(x_n, y_n)$  is assumed to be the same as  $(x_0, y_0)$ . Note that if the points are numbered in clockwise order the area  $A$ , computed as above, will have a negative sign; but the centroid coordinates will be correct even in this case. we calculate the centroid for differentiating various hand pattern based on number of fingers. By assuming number of fingers and finger position as distinct from location of the centroid in the hand image we can differentiate many hand gestures. The partition of the hand is very important. Hand gesture patterns which contain the index finger will fall in the left hand side of the centroid. And other hand patterns which contain little finger will find in the right side of the centroid location in image. Centroid always computed at the geometric center of the image and it is also called as center of mass if the image is uniformly distributed. Centroid is calculated using the image moment, which is the weighted average of pixel's intensities of the image.

Here we have used a function `s = regionprops(L, 'Centroid');`



FIG 4.4 CENTROID OF IMAGE

Red point in the image is the Centroid of the image

#### 4.6.2. ORIENTATION OF IMAGE

In this we will find out whether the orientation of the image is horizontal or the vertical i.e. whether the thumb is up down or right left. Also we have make use of the fact that all the horizontal images we have start form extreme left means we always have a yboundary==1 for the case of horizontal image. In our program if the image is of vertical orientation it will display “vertical orientation” and if image is of horizontal orientation it will display a message “horizontal orientation”

#### 4.6.3. DRAWING A BOUNDARY BOX

After removing all the connected components (objects) other than the pointer, using MATLAB’s “*bwlabel*” function the pointer can be labeled. In other words the region can be detected. To get the properties of the region such as center point or bounding box etc., MATLAB’s built in “*regionprops*” function can be used as;

`STATS = regionprops(BW, properties)`

Where; it measures a set of properties for each connected component (object) in the binary image, BW. The image BW is a logical array; it can have any dimension. In our application since we need to get the center of the object and the bounding box, we used only these two properties.

#### 4.6.4.THUMB DETECTION

In this we make use of theory explained above regarding bounding box. In this we calculated white pixel count of left ,white pixel count of right , white pixel count top part , white pixel count of lower part and compare it with the thmbpercent which is

`thmbpercent=real(0.0690*whiteCntAll);`

suppose (white pixel count of left)<( thmbpercent=real(0.0690\*whiteCntAll))

then thumb is at left

suppose (white pixel count of right)<( thmbpercent=real(0.0690\*whiteCntAll))

then thumb is at right

suppose (white pixel count of top) < (thumbpercent = real(0.0690 \* whiteCntAll))  
then thumb is at top  
suppose (white pixel count of bottom) < (thumbpercent = real(0.0690 \* whiteCntAll))  
then thumb is at bottom



When thumb is at left



When thumb is at right



When thumb is up

FIG 4.5 THUMB DETECTION



## CHAPTER 5 FEATURES EXTRACTION

### 5.1 OBJECTIVES

Histogram of oriented Gradient descriptors, are the feature descriptors used in computer vision and the image processing for purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image.

### 5.2 THEORETICAL BACKGROUND

HOG features have been introduced by Navneet Dalal and Bill Triggs[25] who have developed and tested several variants of HOG descriptor, with different spatial organization, gradient computation and normalized methods.

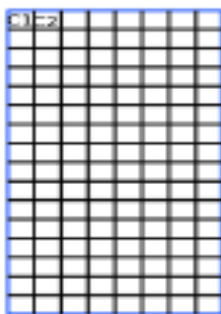
The essential idea behind the HOG descriptors is that local object appearance and shape within an image can be described as by distribution of intensity gradients or edge detection . The implementation of HOG descriptors can be achieved by dividing the image into small connected regions called cells and for each cell compiling HOG directions or edge detections for the pixels within the cell. The combination of these histograms represent the descriptor. For better performance , the local histograms can be contrast- normalized by calculating a measure of the intensity across bigger region of the image called block and then using the value to normalize all cells within the block. This normalization resulted in improved invariance to changes illumination or shadowing.

### 5.3 HOG DESCRIPTOR

Following HOG configurations are for human detection

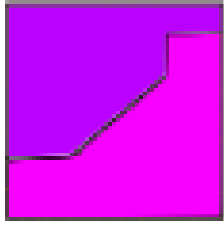
- Window :64x128 pixels
- Histogram Bins:9 , each bin is 20 Degree. Gradient orientation range 0-180 degree. Orientation sign is ignored.
- Cell:8x8 pixels
- Block:16x16 pixels
- BlockStride/BlockShift: 8pixels

#### 1. Calculation of HOG for all cells

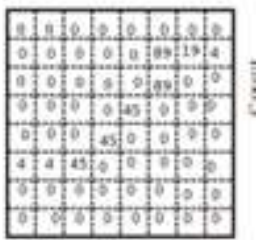


64X128 Pixels w1 window taken from the image

FIG. 5.1



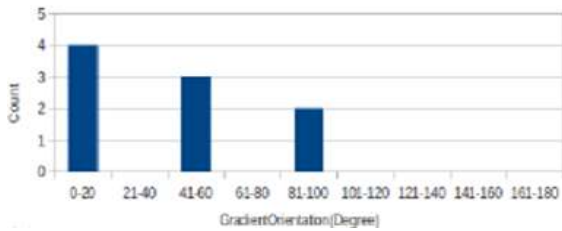
Cropped image of cell C1  
FIG. 5.2



Gradient orientation of cell C1

For the x direction  $x = [0 -1 1]$  For the y direction  $y = [0 -1 1]^T$  which is the same as x but transposed and multiplied by  $-1$   
 $dx = p(x+1) - p(x-1)$   $dy = p(y+1) - p(y-1)$   
 $orientation = \text{atan}(dy/dx)$

FIG. 5.3



$h1 = \{4, 0, 3, 0, 2, 0, 0, 0, 0\}$

Histogram of oriented gradient h1 for cell C1

FIG. 5.4

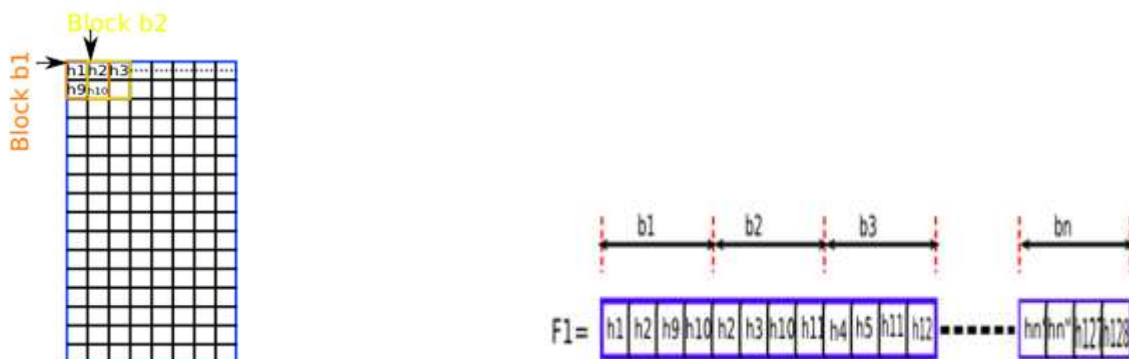


FIG. 5.5 Accumulate HOGs over the block to create HOG descriptor

Dimensions/Size of HOG Descriptor Vector (F1) will be (Number of blocks\*Cells of Block\* Number of bins per Histogram).=105\*4\*9=3780.

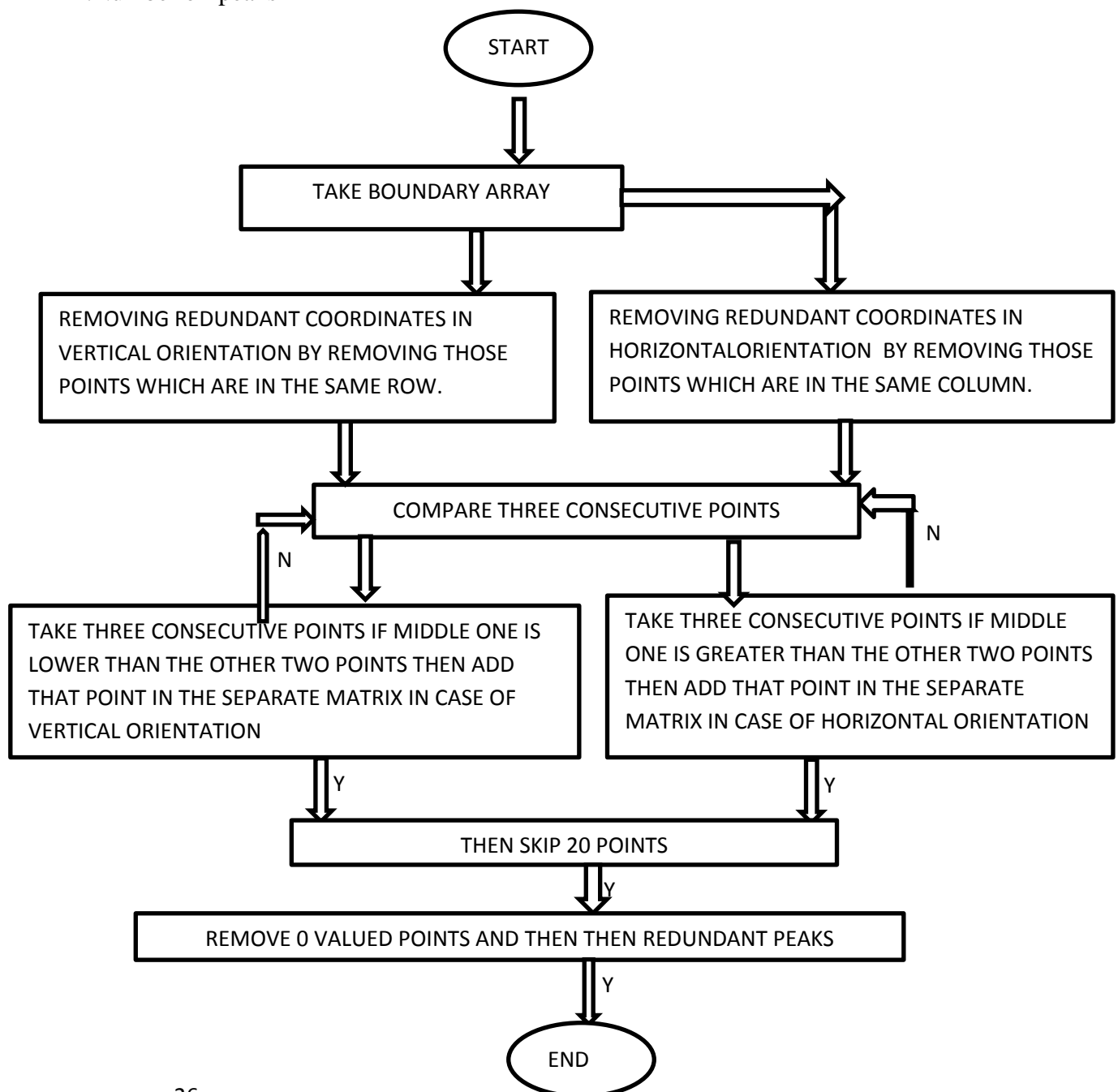
Number of blocks=((Width of window/BlockStrid in X Direction)-1)\*((Height of window/ BlockStrid in Y Direction)-1)=((64/8)-1)\*((128/8)-1)=7\*15=105.

Shift block b1 by BlockStride(here 8 pixels) to create block b2 and shift b2 by BlockStride to create block b3 and so on in horizontal and vertical direction.

We can also find the peaks in processed image but after using the HOG features in this system we do not need to find the peaks but during my early days in developing this system I had made a function for finding the peaks .It was helpful in finding the number of fingers

### 5.4 OTHER FEATURES

#### 1.Number of peaks



## 2. FINDING NUMBER OF FINGERS

### FIRST WAY

We have find number of fingers by using peaksh command in matlab which find number of peaks in a boundary of a image.

But sometimes it consider folded fingers also peak so this error we have removed it by by using a distance command.

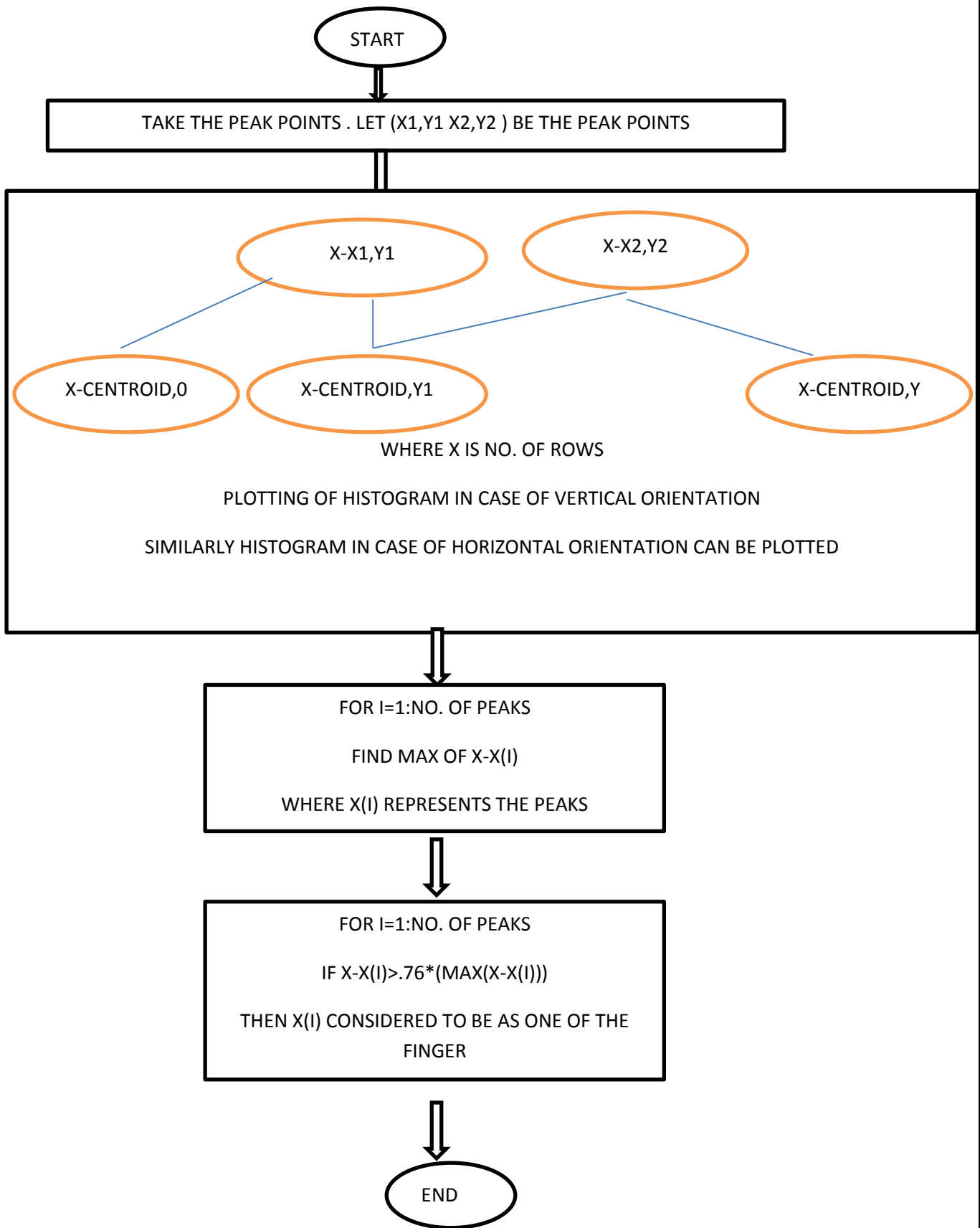
Distance command basically find the distance between the centroid and the peak. Suppose the maximum distance is  $d_{stm}$  then all those peaks which have distance from centroid suppose  $dist(i)$  such that  $d_{stm} - dist(i) > 20$  we ignore that peaks.



FIG. 5.6 CENTROID METHOD OF FINDING FINGERS

Red points on fingers are peaks and red point on thumb is centroid

### SECOND WAY



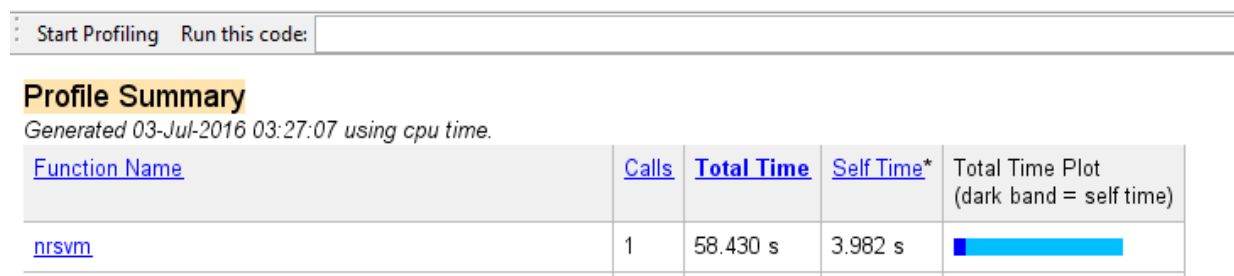
## CHAPTER 6 CLASSIFIERS

### 6.1 INTRODUCTION

Three classifiers can be used

1. KNN (k-Nearest Neighbors)
2. ANN (artificial neural network)
3. SVM (support vector machines)
4. HMM (hidden markov model)

I have used KNN as it is easy to use as compare to others classifiers. SVM is also easy to apply but first problem I have encountered is that predefined function for SVM can classify only two groups. After that I have made multisvm function for SVM to classify between different groups but it is taking lot of time as compare to KNN classification.



Start Profiling Run this code:

**Profile Summary**  
Generated 03-Jul-2016 03:27:07 using cpu time.


Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
<a href="#">nrsvm</a>	1	58.430 s	3.982 s	

FIG.6.1 SIMULATION TIME IN CASE OF SVM



**Profile Summary**  
Generated 03-Jul-2016 03:32:49 using cpu time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
<a href="#">nrknn</a>	1	22.013 s	3.369 s	

FIG.6.2 SIMULATION TIME IN CASE OF KNN

Other classifier can be used HMM (Hidden Markov Model)

Out of all classifiers I have used KNN as it is not only easy to apply it is also giving less computation time as compare to others.

### 6.2 K- NEAREST NEIGHBOURS

This classification method uses the feature-vectors gathered in the training to find the k nearest neighbors in a n-dimensional space. The training mainly consists of the extraction of (possible good discriminable) features from training images, which are then stored for later classification. Due to the use of distance measuring such as the euclidian or manhattan distance, the algorithm performs relatively slowly in higher dimensional spaces or if there are many reference features. In [26], an approximate nearest neighbors classification was proposed, which provides a better performance.

### 6.2.1 KNN ALGORITHM

The training examples are vectors in a multidimensional feature space and each one is classified that is labelled by using one of the classifiers present and during the training phase features of the test points are stored and they are classified into one of the predefined classes. For example we are having many groups of labelled samples. Nature of the items present in the groups is of homogeneous in nature. Let's assume we have an unlabelled sample which is needed to be classified. In this KNN algorithm will be very useful.

K nearest neighbours is a very simple algorithm. It stores all the features of all the available samples and it classifies new samples by a majority vote of its K neighbours.

The efficiency of the model increases if we have more well defined classes. The efficiency of the system depends upon the value of K. KNN has also a drawback. Suppose we have a sample and needed to be classified into a predefined classes. So we create circle around the sample of some radius. And in that area we try to find the number of other classes present in that area and the class which is in a majority will win. But there might be a case in which class which is very frequent will try to dominate the classification of a new example presented. We can overcome this problem if we weight the classifications by taking into an account the distance between the test point and to each of the KNN around it i.e. multiply class of each of the k nearest neighbour by the inverse of the distance between the test point and the k nearest neighbours. There is also an alternative way to overcome this problem by SOM ( self organizing map),in which each node represents a centre of a cluster of test points which are having same class i.e. having same features and it is regardless of the majority k nearest neighbours surrounding test point so K nearest neighbour can be applied to the self organizing map.

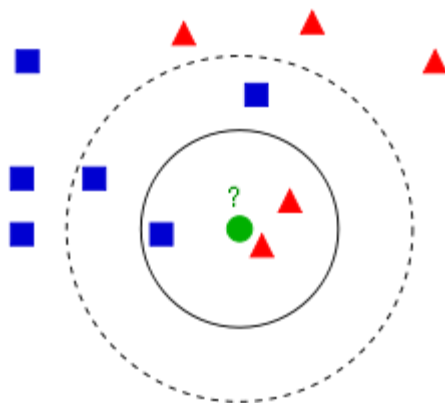


FIG. 6.3. EXAMPLE OF K-NN CLASSIFICATION

Now in the above figure we classify the green circle into one of the classes surrounding that point. We can see the circle around the green circle in solid line in which we are having two red triangles and a blue box. So in above figure we have  $k=3$ . If the value of k is 5 then we can see the dashed line circle having three blue boxes two red triangles. So if we have  $k=3$  then the class of the test point green circle is same of that red triangle and if  $k=5$  then we classified the green circle into the class of blue square.

### 6.3 OTHER CLASSIFIERS

Other classifiers that can be used are explained below

**Hidden Markov Models.** The Hidden Markov Model (HMM) classifiers belong to the class of trainable classifiers. It represents a statistical model, in which the most probable matching gesture-class is determined for a given feature vector, based on the training data. In [27], HMMs were successfully used to distinguish up to 40 different hand gestures with an accuracy of up to 91.9%. In order to train the HMM, a Baum-Welch re-estimation algorithm, which adapts the internal states of the HMM according to some feedback concerning the accuracy, was used.

**Multi Layer Perceptron.** A Multi Layer Perceptron (MLP) classifier is based on a neural network. Therefore, MLPs represent a trainable classifier (similar to Hidden Markov Models). They use three or more layers of neurons that are all connected. During the training phase, the weights of the connections between the neurons are adapted, based on the feedback that describes the difference between the output and the expected result. In [28], a MLP classifier was used to recognize 26 different ASL gestures with a recognition rate of up to 98.7%, depending on the number of features used to describe the gesture.

**Support Vector Machine** Machine learning is known as subfield of artificial intelligence. Through machine learning we can develop methods for enabling a computer to learn. Over the period there are so many techniques developed for machine learning. Support vector machine (SVM) has been firstly introduced by Vapnik and gained popularity because of its exiting feature such as better empirical performance. Support vector machine (SVM) is a classification and regression technique that uses machine learning theory to maximize the accuracy of prediction. Support vector machines is discussed for two-class problems. First, support vector machines, in which training data are linearly separable in the input space and the second one support vector machines for the case where training data are not linearly separable and map the input space into the high dimensional feature space to enhance linear separability in the feature space. For a two-class problem, a support vector machine is trained so that the direct decision function maximizes the generalization ability



## CHAPTER 7 APPLICATION

### 7.1 WORKING APPLICATION

The working application of my system and which also my objective is the hardware having 4 relays controlled by the microcontroller ATMEGA 8L which is getting instruction from MATLAB detected gestures.

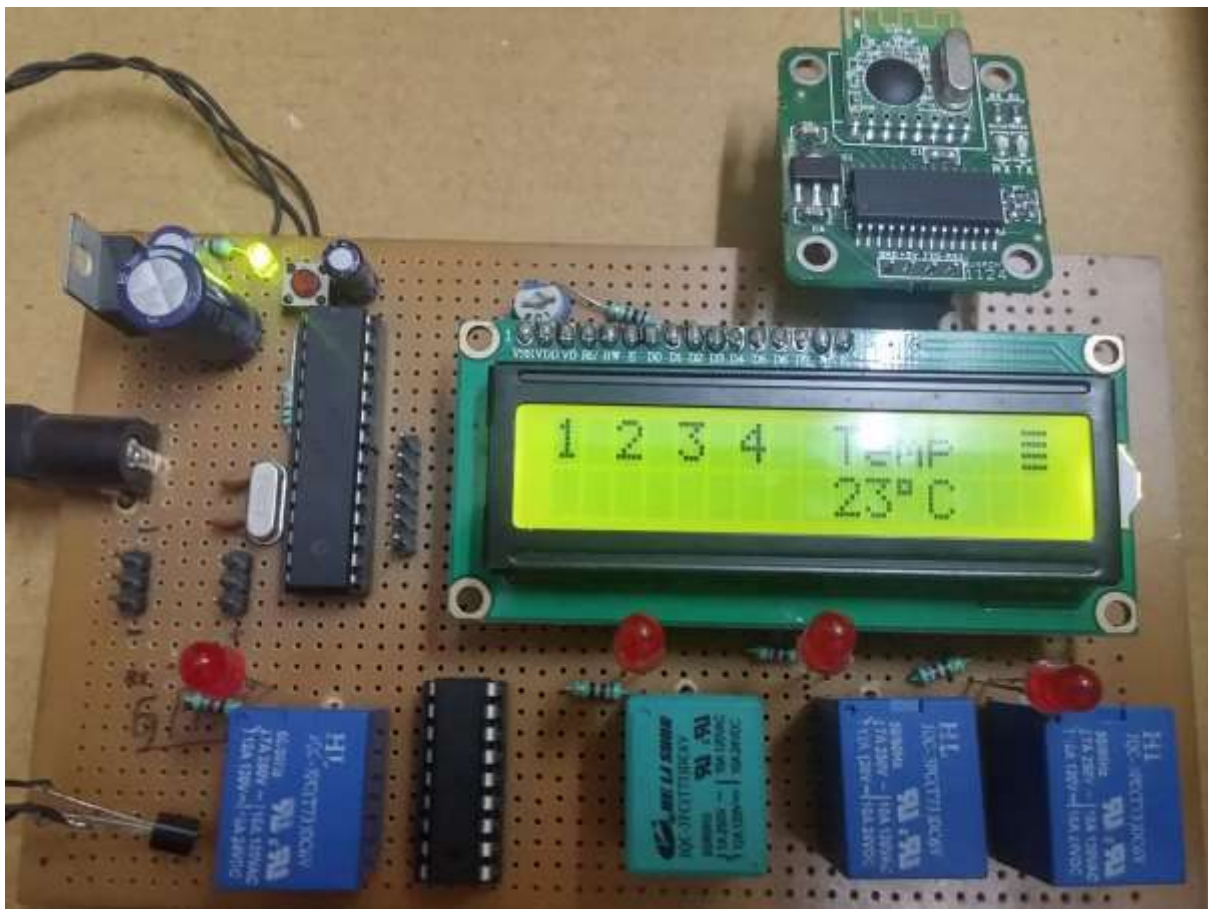


FIG. 7.1 FINAL HARDWARE

Below are the list of the components and some brief description of the components that I have used in my hardware.

### 7.1.1 ATMEGA 8L MICROCONTROLLER



FIG. 7.2 PIN DIAGRAM OF ATMEGA 8L

The ATmega[29] series is a very popular series. I have used ATmega 8L series due to its popularity and the availability of the content regarding this on the net. In case if I found any issues regarding this microcontroller , just google regarding this issue would be enough and also my requirements was not that high and ATmega was sufficient for developing my system. It's an 8 bit microcontroller and a CMOS device so it requires very low power . It's based on RISC architecture thereby speeding up the processing and execute powerful instructions in a single clock cycle, so ATmega8 can achieves throughputs 1 MIPS per MHz, allowing optimize power consumption versus the processing speed. As Atmega is a microcontroller there are lots of content regarding this but in this topic I have mentioned only about the selected topics which I needed to go through before getting started with the microcontroller and also the topics which are needed to implement my system.

In this project I have worked with a baud rate of 9600 . To set baud rate of USART register , we have to use following formula

$$\text{BAUDRATE} = ((F\_CPU)/(\text{BAUD} * 16UL) - 1)$$

Where BAUD= 9600

Operating Mode	Equation for Calculating Baud Rate <sup>(1)</sup>	Equation for Calculating UBRR Value
Asynchronous Normal mode (U2X = 0)	$BAUD = \frac{f_{osc}}{16(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{osc}}{8(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{osc}}{2(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{2BAUD} - 1$

TABLE 7.1 BAUD RATE CALCULATION

I have used **USART normal mode operation**. In this mode of communication, the data is transmitted/received asynchronously, i.e. we do not need (and use) the clock pulses, as well as the XCK pin. The data is transferred at the BAUD rate we set in the UBRR register. This is similar to the UART operation.

So for operation in USART normal mode we have to keep the U2X pin 0 in the UCSRA register.

Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

FIG. 7.3 UCSRA REGISTER

In this case, the 16-bit UBRR register is comprised of two 8-bit registers – UBRRH (high) and UBRRL (low).

Load upper 8-bits of the baud to UBRR register

Load lower 8-bits of the baud to the UBRR register

Bit	15	14	13	12	11	10	9	8	
	URSEL	-	-	-	UBRR[11:8]				UBRRH
	UBRR[7:0]								UBRRL
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

FIG. 7.4 UBRR REGISTER

To turn on the transmission and reception in the UCSRB register

RXEN=1 AND TXEN =1

For a condition where, whenever any data is received, an interrupt will be fired by the CPU, then we have to keep RXEN=1 in the UCSRB register.

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

FIG. 7.5 UCSRB REGISTER

Now , in my project I am sending 8 bits at a time

So, UCSZ0=1 and UCSZ1=1 , for 8 bit mode

UDRE bit in UCSRA is very important whenever I send or receive data.

The UDRE Flag indicates if the transmit buffer (UDR) is ready to receive new data. If UDRE is one, the buffer is empty, and therefore ready to be written. The UDRE Flag can generate a Data Register Empty Interrupt (see description of the UDRIE bit in UCSRB register). UDRE is set after a reset to indicate that the Transmitter is ready.

Initialization of ADC

REFS0=0 in the ADMUX register for reference voltage=5v

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

FIG. 7.6 ADMUX REGISTER

Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

FIG. 7.7 ADSCRA REGISTER

**ADC Prescaler Select Bits** – The prescaler (division factor between XTAL frequency and the ADC clock frequency) is determined by selecting the proper combination from the following in the ADSCRA (as shown above)

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

TABLE 7.2 ADC PRESCALAR SELECTIONS

### CONDITIONS FOR ON AND OFF OF 4 RELAYS

I have send 'a' , 'b' , 'c' , 'd' , 'w' , 'x' , 'y' , 'z' wirelessly to a hardware with the help of matlab.

case 'a' : PORT\_ON(relay1)

case 'b' : PORT\_ON(relay2)

case 'c' : PORT\_ON(relay3)

case 'd' : PORT\_ON(relay4)

case 'w' : PORT\_OFF(relay1)

case 'x' : PORT\_OFF(relay2);

case 'y' : PORT\_OFF(relay3)

case 'z' : PORT\_OFF(,relay4);

#### 7.1.2 ZIGBEE

I have used ZIGBEE[30] both at the transmitter and the receiver side to have two way communication .It works at 2.4 Ghz frequency in a half duplex mode . It can automatically switch between the receive and transmit mode and also there is a LED which indicates the mode in which the RF modem is. We can adjust the baud rate of Receive and Transmit serial data. The baud rate can be of 9600, 4800, 2400 and 19200 bps at 5V or 3V level for direct interfacing to microcontrollers. This RF modem is basically used in applications where there is a need of two way wireless data transmission. High data rates can be employed in ZIGBEE and also this RF modem has a long range. The communication protocol that ZIGBEE employs is of self controlled and for user interface it is completely transparent.

## Features

- Switching between TX and RX mode is automatic
- FSK technology, half duplex mode, robust to interference.
- 2.4 GHz band, no need to apply frequency usage license.
- Protocol translation is self controlled, easy to use.
- High sensitivity, reliable transmission range.
- Standard UART interface, TTL (3-5V) logic level.
- Stable, small size, easier mounting.
- No tuning required, PLL based self tuned.
- Error checking (CRC) of data in built.

## Applications

Applications are not limited to a certain level but because of being cost-effective, low-power battery and wireless connectivity, this Zigbee technology is used in almost every appliance if not in all.

Zigbee technology is programmed in a chip form and is used in many devices to function automatically. For controlling and monitoring a whole factory unit while sitting in one cabin is possible by using Zigbee technology. It centralizes all the units in one place and enables the remote monitoring.

- Sensor Networks / Data collection
- Wireless metering
- Access control / Identity discrimination
- IT home appliance
- Smart house products / Security Systems
- Remote control / Remote measurement system
- Weather stations

## Typical Application

For setting up the wireless communication between the microcontroller/ microprocessor to the computer , this RF modem is very useful.

## Specifications

- Working voltage is around 5v (min 4.5 V and max. 9V)
- Frequency 2.4 Ghz
- Output RF Power 1 dBm
- Typical Operating Range: 30-50 meters
- UART baud rate (8 bit data, no parity, 1 stop bit) 9600/4800/2400/19200 bps

## Pin definition

- RXD = Input or receive pin: Input serial data having 3 to 5V of logic level and generally it is connected to TXD pin of microcontrollers.

- TXD = Output or transmit pin: Output serial data having 3 to 5V of logic level and generally it is connected to RXD pin of microcontrollers.
- +5V = 5V regulated supply input.
- GND = Ground pin and it shares the common ground with microcontroller.

### Operation

The RF module works in a half duplex mode i.e. either in a receiving mode or in a transmitting mode but does not work in both mode simultaneously . So after transmission it switch automatically to receiving mode. There is a LED which indicates the mode in which the RF modem is. The received data is checked by CRC . In a transmission mode and any data which is need to be transmitted is input to a buffer for next transmission cycle. This RF modem also has a internal buffer for incoming data so that it is not corrupted with the data which is needed to be transmit . After power on ,TX LED will blink and indicates the initialization of the ZIGBEE. The RX LED is directly on TX OUT pin to indicate that the data is received and sent to the output pin.

### Setting of Baud Rate

To configure the baud rate it is needed to be set when the Zigbee is off. If we try to modify during power up the module will still operate in the previous configuration. Solder the pads to ON the jumper. And by default it will works in 9600 bps and has no jumper in the unit.

B1 and B2 are two jumpers on the pcb.

- B1 = OFF B2 = OFF 9600 bps (**Default**)
- B1 = ON B2 = OFF 4800 bps
- B1 = OFF B2 = ON 2400 bps
- B1 = ON B2 = ON 19200bps

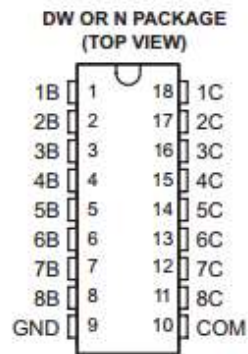
### Frequency channel setting

We can employ our Zigbee in different channel . To communicate between the two Zigbee they must be working at same channel. We can use the unit in four different channels. By default it works in channel 1 only and has no jumper in the unit. To configure it, it is needed to be set when the Zigbee is off. If we try to modify during power up the module will still operate in the previous configuration.

- F2 = OFF F1 = OFF is Channel #1 (**Default**)
- F2 = OFF F1 = ON is Channel #2
- F2 = ON F1 = OFF is Channel #3
- F2 = ON F1 = ON is Channel #4

### 7.1.3. ULN2803 RELAY DRIVER

The ULN2803A device is a high-voltage, high-current .The device consists of (Single Output) eight NPN Darlington pairs



**Pin Functions**

PIN		TYPE	DESCRIPTION
NAME	NO.		
<1:8>B	1 - 8	I	Channel 1 through 7 darlington base input
<1:8>C	18 - 11	O	Channel 1 through 7 darlington collector output
GND	7	—	Common Emmitter shared by all channels (typically tied to ground)
COM	8	I/O	Common cathode node for flyback diodes (required for inductive loads)

FIG. 7.8 ULN 2803 PIN DIAGRAM

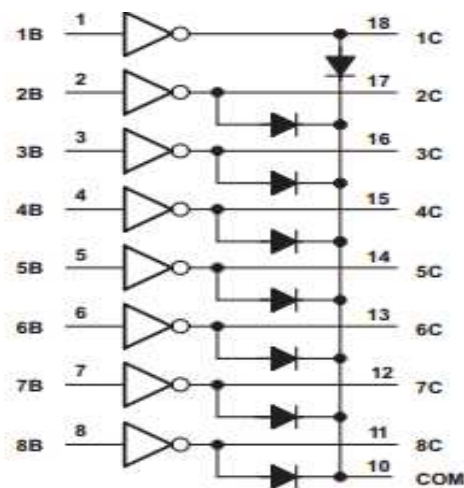


FIG. 7.9 INTERNAL STRUCTURE OF ULN2803

### 7.1.4. LM 35 SERIES TEMPERATURE SENSOR

Features

- Calibrated Directly in Celsius (Centigrade) temperature
- Linear + 10-mV/°C Scale Factor



- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full –55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost Due to Wafer-Level Trimming
- Operates from 4 V to 30 V
- Less than 60-μA Current Drain
- Low Self-Heating, 0.08°C in Still Air
- Non-Linearity Only  $\pm\frac{1}{4}$ °C Typical,
- Low-Impedance Output, 0.1  $\Omega$  for 1-mA Load

#### Description

The LM35[31] series are precision integrated-circuit devices with an output voltage linearly proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of  $\pm\frac{1}{4}$ °C at room temperature and  $\pm\frac{3}{4}$ °C over a full –55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the wafer level.

#### Application

- Power Supplies
- Battery Management
- HVAC
- Appliances

The difference between an LM335 and LM34 and LM35 temperature sensors is the LM335 sensor gives out the temperature in degrees Kelvin, while the LM35 sensor gives out the temperature in degrees Celsius and the LM34 sensor gives out the temperature in degrees Fahrenheit.

#### 7.1.5 USB TO TTL

The USB TTL Serial cables are a range of USB to serial converter cables which provide connectivity between USB and serial UART interfaces



FIG. 7.10 USB TO TTL

#### 7.1.6 LCD (LIQUID CRYSTAL DEVICE)

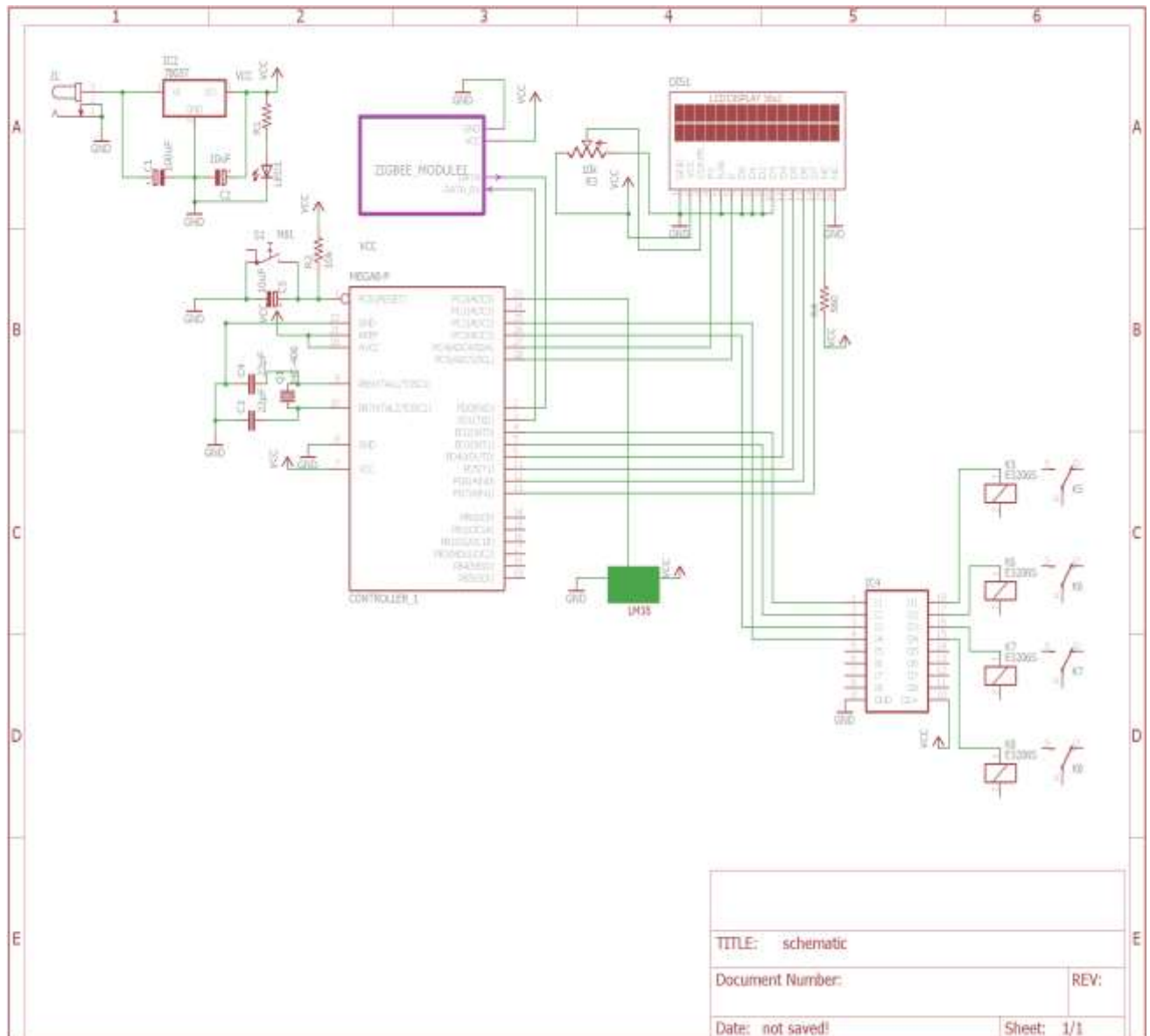


FIG. 7. 11 LCD

We always use devices made up of Liquid Crystal Displays (LCDs) like computers, digital watches and also DVD and CD players. They have become very common and have taken a giant leap in the screen industry by clearly replacing the use of Cathode Ray Tubes (CRT). CRT draws more power than LCD and are also bigger and heavier. In this we have two rows and sixteen columns. In this there are only two main things to see

1. Lcd4\_Write\_String(to write on LCD)
2. Lcd4\_Set\_Cursor(to set Cursor)

## 7.1.7 SCHEMATIC OF MY FINAL HARDWARE



## 7.2 OTHER APPLICATIONS

Other tasks that code can perform

Nircmd

- MUTE SYSTEM VOLUME
- UNMUTE SYSTEM VOLUME
- OPEN CDROM

Also make use of JAVA.AWT.ROBOT to move cursor diagonally

Write an alphabet in Microsoft word using actx control function

Pronunciation of alphabet by using

- Google Translator

```
vlc1 = actxcontrol('VideoLAN.VLCPlugin.2', [1 1 0 0]);  
vlc1.Volume = 50; % Set Volume  
vlc1.AutoLoop = 0; % Set Autoloop  
vlc1.playlist.add(['http://translate.google.com/translate_tts?tl=', 'en', '&q=', str] ); %  
Unofficial text to speech by google  
vlc1.playlist.play; % play it
```

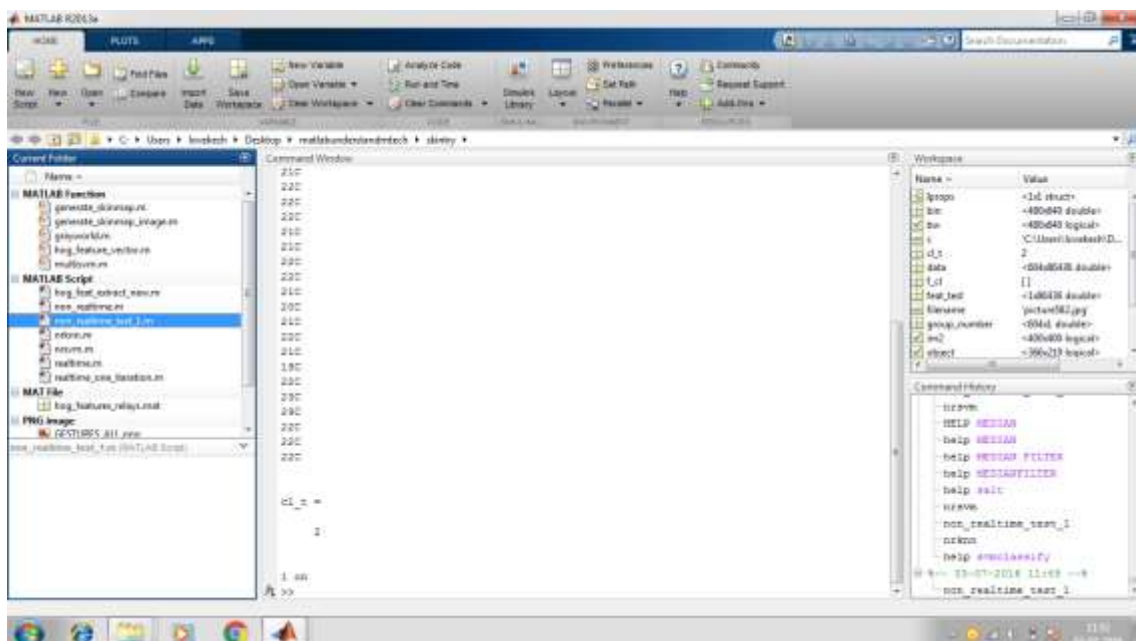
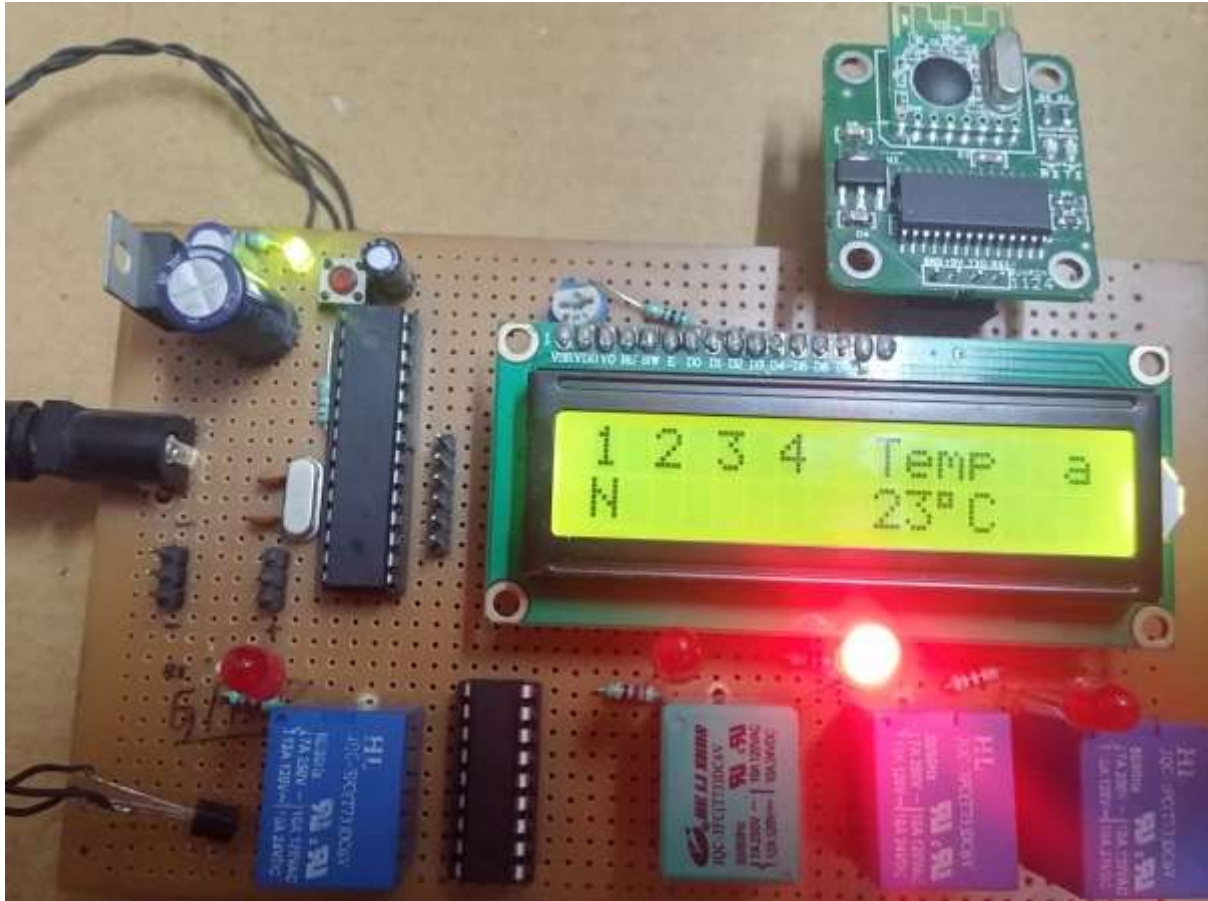
- System Speech Synthesizer using NET.addAssembly

```
userPrompt = 'What do you want the computer to say?';  
titleBar = 'Text to Speech';  
defaultString = 'Hello World! MATLAB is an awesome program!';  
caUserInput = inputdlg(userPrompt, titleBar, 1, {defaultString});  
if isempty(caUserInput)  
    return;  
end; % Bail out if they clicked Cancel.  
caUserInput = char(caUserInput); % Convert from cell to string.  
NET.addAssembly('System.Speech');  
obj = System.Speech.Synthesis.SpeechSynthesizer;  
obj.Volume = 100;  
Speak(obj, caUserInput)
```

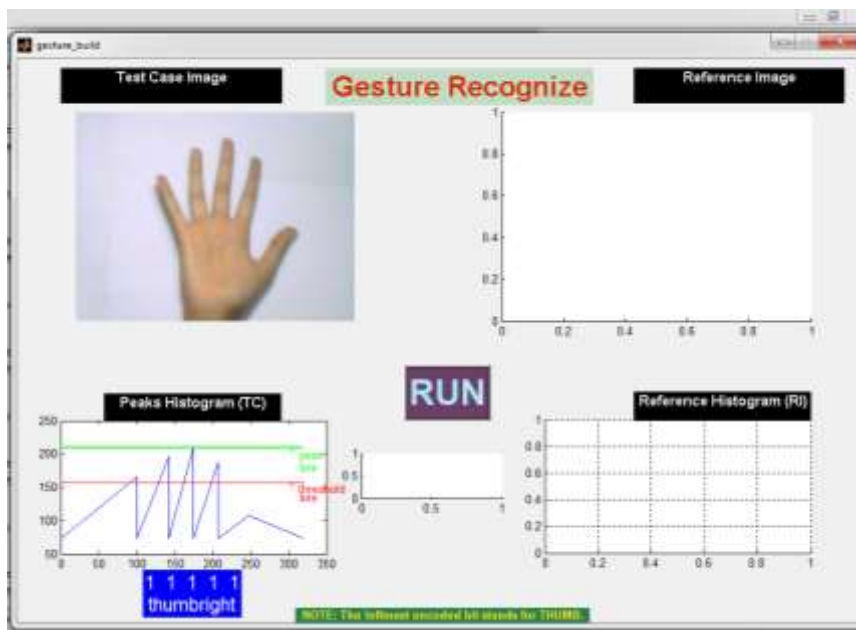
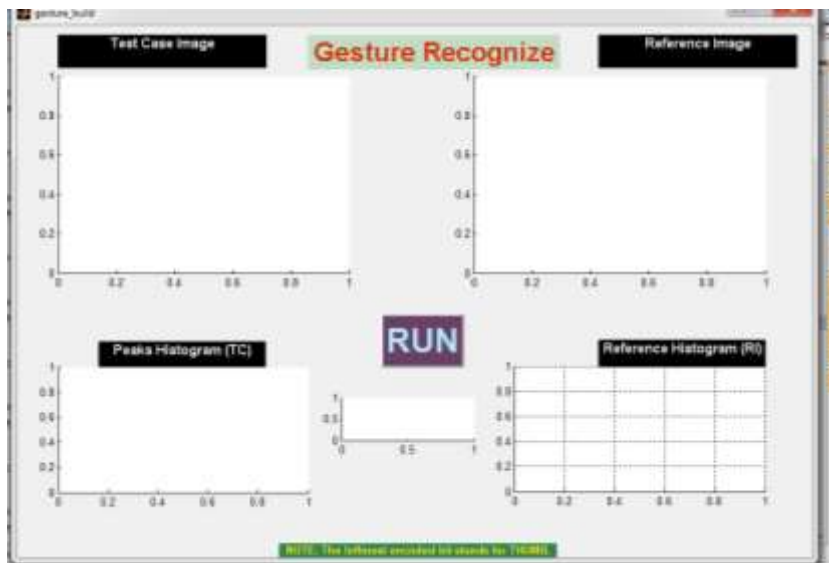
## CHAPTER 8 RESULTS AND CONCLUSION

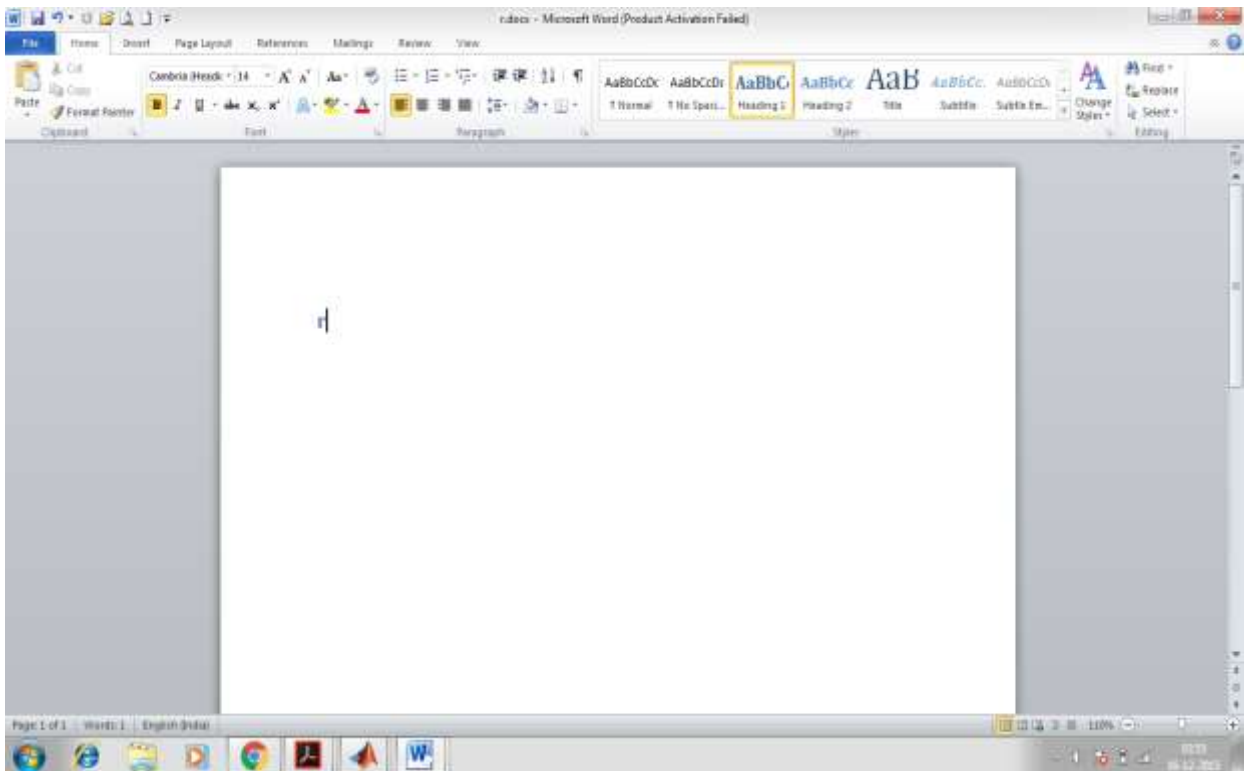
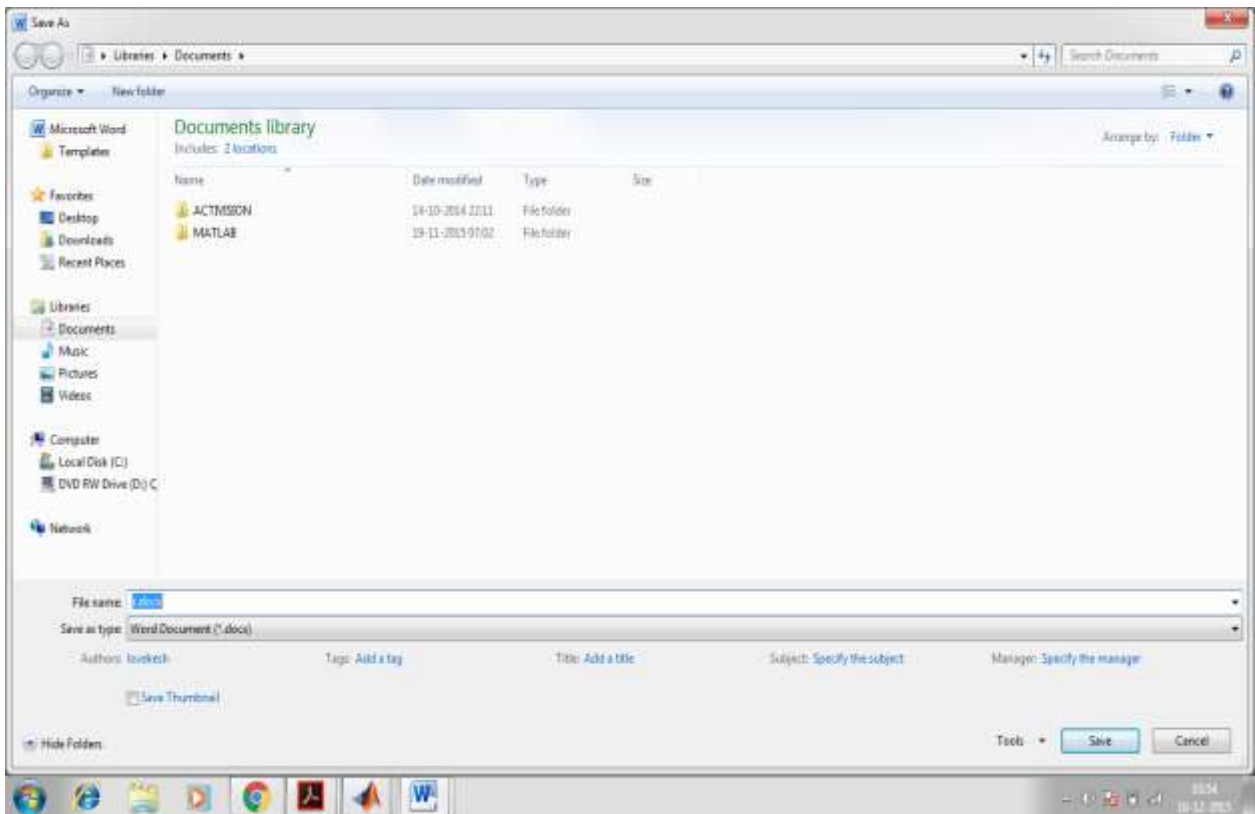
### 8.1 RESULTS

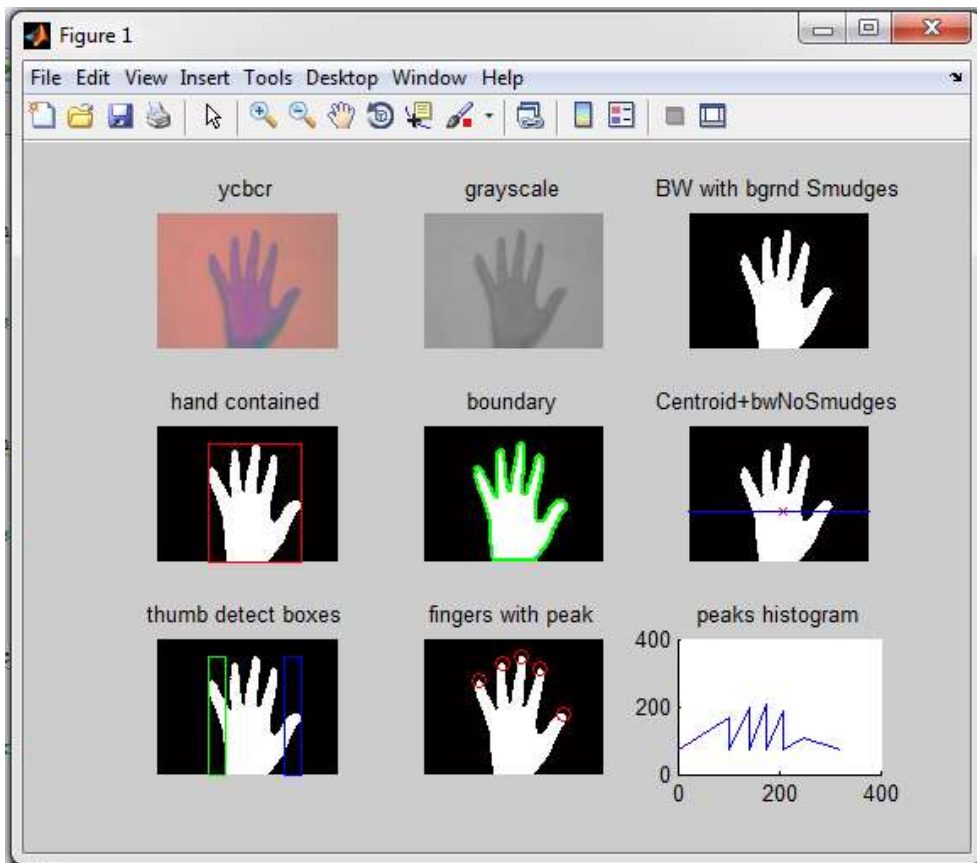
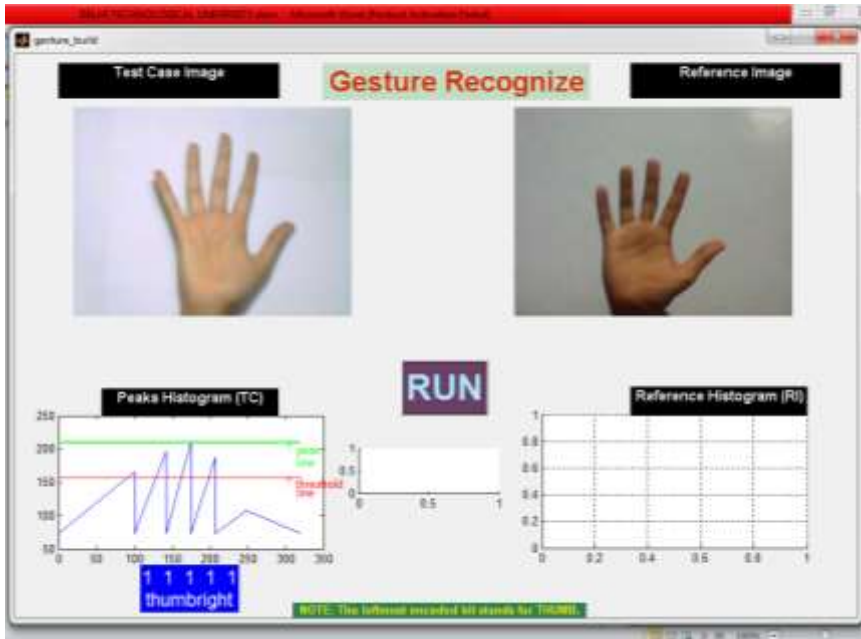
When I send 'w'



Other results which are not needed to implement the system but these I have found during early days of my work and give more information about the image and could play a vital role in other application are given below

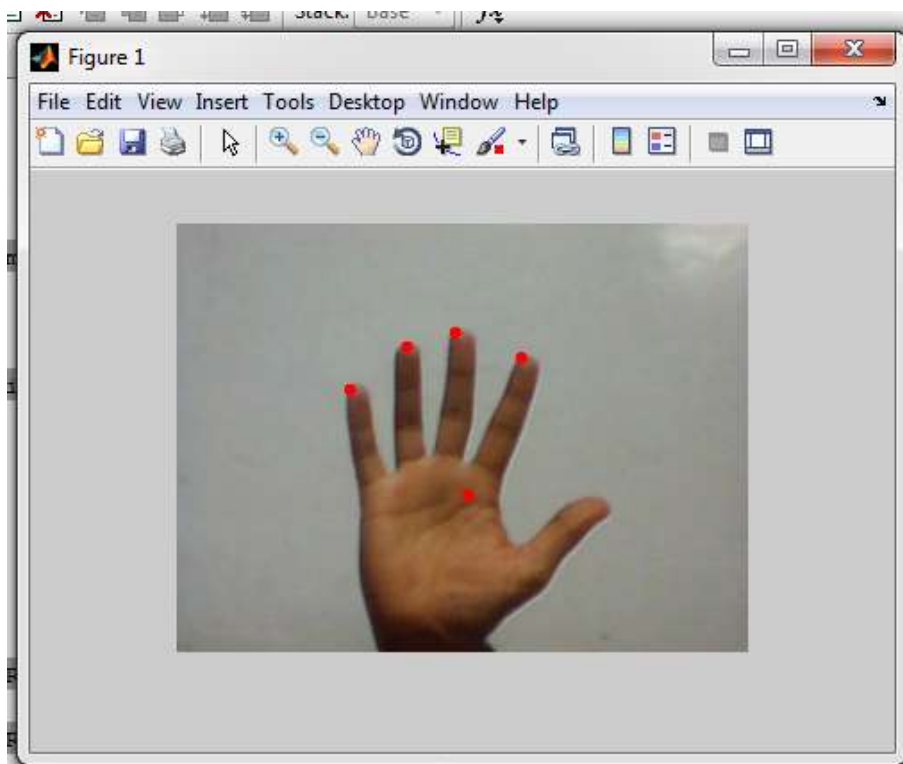




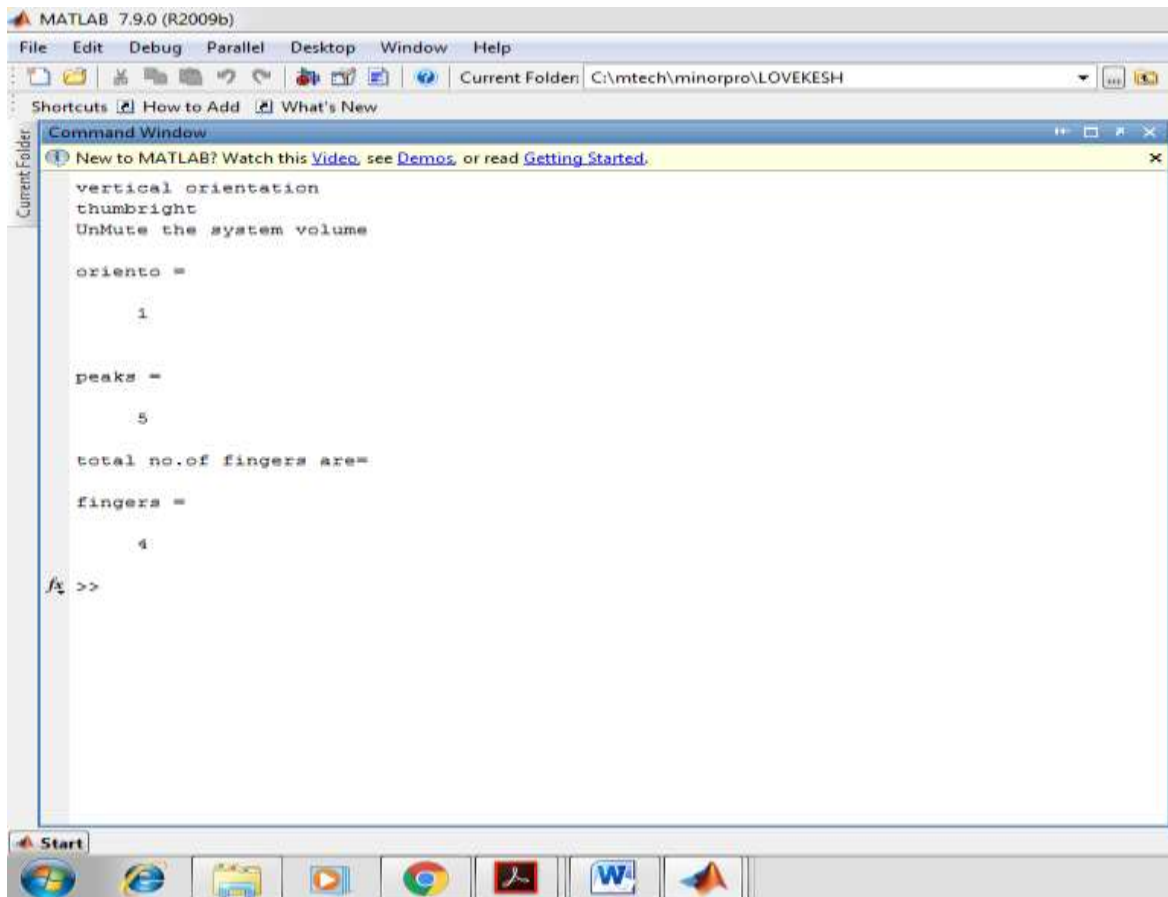




```
MATLAB 7.9.0 (R2009b)
File Edit Debug Parallel Desktop Window Help
Current Folder: C:\mtech\minorpro\LOVEKESH
Shortcuts How to Add What's New
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
31 175
peaksup =
5
err =
5
finga =
1 1 1 1 1
flag =
1 1 1 1 1
str =
s
Document will be saved in C:\mtech\minorpro\LOVEKESH\TextDoc.doc
Elapsed time is 17.785835 seconds.
>>
```



Showing four fingers and a centroid



## 8.2 CONCLUSION

In present environment a number of facilities and various modes for providing input to any application are available. It is though unfortunate that with the ever increasing smart environments and corresponding input technologies still not many applications are available which are controlled using current and smart facility of providing input which is by hand gesture. The most important advantage of the usage of hand gesture based input modes is that using this method the user can interact with the application from a distance without using the keyboard or mouse. The application of manipulating objects through hand gestures in virtual environment is being proposed and implemented in the present paper provides a suitable efficient and user friendly human computer interface. With the help of this application the user can interact with the virtual objects using hand gesture instead of any other physical input devices .As the application provides the flexibility to the users and specifically physically challenged users to define the gesture according to their feasibility and ease of use.

The idea of the project got started by finding number of peaks and the orientation of image. But this ideas were not giving satisfactory results that we wanted. Skin detection is much effective way over the direct conversion from RGB to binary image. Most of the time skin detection method gives one object and perfect outline of my palm. My supervisor and I

had the idea of trying to use HOG features extraction in conjunction with the classifiers(KNN) . I have used other classifiers also but found KNN to be effective and efficient. Another advantage of using HOG features in conjunction with KNN is that we can add as many training samples as many we want without putting any other extra effort. The Histogram-based and Explicitly threshold skin detection methods were evaluated and based on the results, the Histogram method was deemed as more accurate. As far as the orientation algorithm is concerned it can be further improved. The main problem is how good differentiation one can achieve. My major goal was speed and the avoidance of special hardware. The background, which must preferably be black to get accurate results. MATLAB is slower but allows its users to work faster and concentrate on the results and not on the design. My project is working fine and giving accurate results most of the time but it is giving 100% accurate result when the program run for only one iteration . In infinite iteration it is not giving 100% success rate .

## REFERENCES

- [1] Conic, N., Cerseato, P., De & Natale, F. G. B., (2007), “Natural Human- Machine Interface using an Interactive Virtual Blackboard”, In Proceeding of ICIP 2007, pp.181-184.
- [2] Ismail, N. A., & O’Brien, A., (2008), “ Enabling Multimodal Interaction in Web-Based Personal Digital Photo Browsing”, Proceedings of the International Conference on Computer and Communication Engineering , Kuala Lumpur, Malaysia, May 13-15, pp. 907-910.
- [3] Pang, Y. Y., Ismail, N. A., & Gilbert, P. L. S., (2010), “ A Real Time Vision-Based Hand Gesture Interaction”, Fourth Asia International Conference on Mathematical Analytical Modelling and Computer Simulation, pp. 237-242.
- [4] L. Bretzner, I. Laptev, and T. Lindeberg. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In Proc. Face and Gesture, pages 423–428, 2002.
- [5] C.-C. Chang, J.-J. Chen, W.-K. Tai, and C.-C. Han. New approach for static gesture recognition. *Journal of Information Science and Engineering*, 22(5):1047–1057, September 2006.
- [6] J. Segen and S. Kumar. Fast and accurate 3d gesture recognition interface. In ICPR '98: Proceedings of the 14th International Conference on Pattern Recognition-Volume 1, page 86, Washington, DC, USA, 1998. IEEE Computer Society.
- [7] H. Zhou, D. Lin, and T. Huang. Static hand gesture recognition based on local orientation histogram feature distribution model. In CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 10, page 161, Washington, DC, USA, 2004. IEEE Computer Society
- [8] T. Starner and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *Pattern Analysis and Machine Intelligence*, IEEE Transactions, 20(12):1371–1375, December 1998.
- [9] Kortum, P., (2008) “HCI Beyond the GUI: Design for Haptic, Speech, Olfactory, and Other Nontraditional Interfaces” Morgan Kaufmann Publishers, pp. 75-106.
- [10] Conic, N., Cerseato, P., De & Natale, F. G. B., (2007), “Natural Human- Machine Interface using an Interactive Virtual Blackboard”, In Proceeding of ICIP 2007, pp.181-184.
- [11] Moeslund, T. B., & Norgaard, L. (2002) “A brief overview of hand gestures used in wearable human computer interfaces”, Technical report, Aalborg University, Denmark.
- [12] Viola & Jones, (2001), “Rapid object detection using boosted cascade of simple features”, In Proceedings of Computer Vision and Pattern Recognition, pp. I-511 - I-518.
- [13] Chen, Q., Coredea, M. D., Petriu, E. M., Varkony, A. R., Koczy, I., & Whalen, T.E., (2009), “ Human Computer Interaction for Smart Applications Using Hand Gesture and Facial Expressions,” *International Journal of Advanced Media and Communication*, vol. 3c.1/2, pp. 95-109.
- [14] Jain, G. (2009), “Vision-Based Hand Gesture Pose Estimation for Mobile Devices”, University of Toronto.

- [15] Pavlovic, V., Sharma, R., & Huang, T.S. (1997), "Visual interpretation of hand gestures for humancomputer interaction: A review." IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), 7(19):pp. 677–695.
- [16] Marcel, S., Bernier, O., Viallet, J. E., & Collobert, D (2000), "Hand Gesture Recognition using InputOutput Hidden Markov Models." In Proc. of the FG'2000 Conference on Automatic Face and Gesture Recognition.
- [17] Rautaray, S.S., & Agrawal, A. (2010), "A Novel Human Computer Interface Based On Hand Gesture Recognition Using Computer Vision Techniques", In Proceedings of ACM IITM'10, pp.292-296.
- [18] Aran, O., Ari, I., Benoit, F., Campr, A., Carrillo, A. H., Fanard, Akarun, L., Caplier, a., Rombaut, M., & Sankuru, B., (2006) "Sign Language Tutoring Tool", eNTERFACE 2006, The Summer Workshop on Multimodal Interfaces, Croatia.
- [19] Liu, N., & Lovell, B. (2001) "Mmx-accelerated realtime hand tracking system" In Proceedings of IVCNZ.
- [20] F. Chen, C. Fu, & C. Huang, 2003 , "Hand gesture recognition using a real-time tracking method and hidden Markov models" Image and Vision Computing, pp. 745-758.
- [21] Lee, C. S., Ghyme, S. W., Park, C. J., Wohn, K., (1998) "The Control of avatar motion using hand gesture", In Proceeding of Virtual Reality Software and technology (VRST), pp. 59-65.
- [22] Ahn, S. C., Lee, T. S., Kim, I. J., Kwon, Y. M., & Kim, H. G. (2004), "Computer Vision-Based Interactive Presentation System," Proceedings of Asian Conference for Computer Vision.
- [23] <http://www.logitech.com/en-in/product/hd-webcam-c310>
- [24] <http://cimss.ssec.wisc.edu/wxwise/class/aos340/spr00/whatismatlab.htm>
- [25] N. Dalal, B.Triggs: Histogram of Oriented gradients for human detection, IEEE Computer Society Conference on Computer Vision and Pattern Recognition ,2005.
- [26] H. Zhou, D. Lin, and T. Huang. Static hand gesture recognition based on local orientation histogram feature distribution model. In CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 10, page 161, Washington, DC, USA, 2004. IEEE Computer Society
- [27] T. Starner and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. Pattern Analysis and Machine Intelligence, IEEE Transactions, 20(12):1371–1375, December 1998.
- [28] S. G. Wysoski, M. V. Lamar, S. Kuroyanagi, and A. Iwata. A rotation invariant approach on static-gesture recognition using boundary histograms and neural networks. In Proceedings of the 9th International Conference on Neural Information Processing (ICONIP, pages 2137–2141, 2002.
- [29] <http://maxembedded.com/2011/06/the-adc-of-the-avr/>
- [30] <http://www.sunrom.com/p/rf-serial-data-link-uart-2-4-ghz>
- [31] <http://www.ti.com/lit/ds/symlink/lm35.pdf>