



HÖGSKOLAN  
Dalarna

# **Hand Gesture Detection & Recognition System**

**Muhammad Inayat Ullah Khan**

**2011**

**Master  
Thesis  
Computer  
Engineering  
Nr: E4210D**



# DEGREE PROJECT

## Computer Engineering

Programme	Reg number	Extent
Masters Programme in Computer Engineering - Applied Artificial Intelligence	E4210D	15 ECTS
Name of student	Year-Month-Day	
Muhammad Inayat Ullah Khan	2011-12-08	
Supervisor	Examiner	
Pascal Rebreyend	Mark Dougherty	
Company/Department	Supervisor at the Company/Department	
Computer Engineering	Pascal Rebreyend	
Title		
Hand Gesture Detection and Recognition System		
Keywords		
Hand gesture, Recognition, Detection, Diagonal Sum, Pre-processing, Feature extraction, Skin Modelling, Labelling		

## ABSTRACT

The project introduces an application using computer vision for Hand gesture recognition. A camera records a live video stream, from which a snapshot is taken with the help of interface. The system is trained for each type of count **hand gestures** (one, two, three, four, and five) at least once. After that a test gesture is given to it and the system tries to recognize it.

A research was carried out on a number of algorithms that could best differentiate a hand gesture. It was found that the diagonal sum algorithm gave the highest accuracy rate. In the preprocessing phase, a self-developed algorithm removes the background of each training gesture. After that the image is converted into a binary image and the sums of all diagonal elements of the picture are taken. This sum helps us in differentiating and classifying different hand gestures.

Previous systems have used data gloves or markers for input in the system. I have no such constraints for using the system. The user can give hand gestures in view of the camera naturally. A completely robust hand gesture recognition system is still under heavy research and development; the implemented system serves as an extendible foundation for future work.

## ACKNOWLEDGMENT

To accomplish even the simplest task, the help of Allah Almighty is required and my thesis, with its complexities and difficulties, was not an easy task. After completing the task of its production, I am thankful to Allah The Most Powerful that He helped at every step and never let the courage waver even in the face of the worst problem.

My thesis supervisor, **Pascal Rebreyend** was also very supportive of everything. He gives me tremendous support and help throughout, even though it took up a lot of his precious time.

I wish to acknowledge all my university teachers who support and guide me throughout this degree specially **Dr. Hasan Fleyeh, Siril Yella, Mark Dougherty** and **Jerker Westin**. I am also thankful to my family who support me in my studies.

**TABLE OF CONTENTS**

<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 DIGITAL IMAGE PROCESSING.....	1
1.2 BIOMETRICS.....	1
1.3 HAND GESTURE DETECTION AND RECOGNITION.....	2
1.3.1 DETECTION .....	2
1.3.2 RECOGNITION .....	3
1.4 MOTIVATION .....	4
1.5 SCOPE .....	4
1.6 SOFTWARE TOOLS .....	4
1.7 OBJECTIVES.....	5
 <b>CHAPTER 2: LITERATURE REVIEW</b>	 <b>6</b>
2.1 LIGHTING .....	7
2.2 CAMERA ORIENTATIONS AND DISTANCE .....	8
2.3 BACKGROUND SELECTION .....	8
2.4 DIFFERENT RECOGNITION APPROACHES .....	8
2.41 PEN-BASED GESTURE RECOGNITION.....	8
2.42 TRACKER-BASED GESTURE RECOGNITION.....	9
2.43 DATA GLOVES .....	9
2.44 BODY SUITS.....	10
2.45 HEAD AND FACE GESTURES.....	10
2.46 HAND AND ARM GESTURES.....	11
2.47 BODY GESTURES .....	11
2.48 VISION-BASED GESTURE RECOGNITION .....	12
 <b>CHAPTER 3: METHODOLOGY</b>	 <b>14</b>
3.1 PROJECT CONSTRAINTS .....	14
3.2 THE WEBCAM SYSTEM (USB PORT).....	15
3.3 BRIEF OUTLINE OF THE IMPLEMENTED SYSTEM.....	15

---

3.31 PRE-PROCESSING .....	16
3.31.1 SKIN MODELING .....	16
3.31.2 REMOVAL OF BACKGROUND .....	18
3.31.3 CONVERSION FROM RGB TO BINARY .....	18
3.31.4 HAND DETECTION .....	18
3.32 FEATURE EXTRACTION ALGORITHMS .....	19
3.33 REAL TIME CLASSIFICATION .....	20
 <b>CHAPTER 4: FEATURE EXTRACTIONS</b>	 <b>21</b>
4.1 NEURAL NETWORKS .....	21
4.2 ROW VECTOR ALGORITHM .....	22
4.3 EDGING AND ROW VECTOR PASSING ALGORITHM .....	24
4.4 MEAN AND STANDARD DEVIATION OF EDGED IMAGE .....	25
4.5 DIAGONAL SUM ALGORITHM .....	26
4.6 GRAPHICAL USER INTERFACE (GUI) .....	27
4.61 GUI DESIGN .....	27
4.62 NN TRAINING .....	28
4.63 PERFORMANCE OF NN .....	28
4.64 DETECTION AND RECOGNITION .....	29
 <b>CHAPTER 5: RESULTS AND DISCUSSION</b>	 <b>30</b>
5.1 ROW VECTOR ALGORITHM .....	30
5.2 EDGING AND ROW VECTOR PASSING ALGORITHM .....	30
5.3 MEAN AND STANDARD DEVIATION OF EDGED IMAGE .....	31
5.4 DIAGONAL SUM ALGORITHM .....	31
5.5 PROCESSING TIME .....	32
5.6 ROTATION VARIANT .....	32
5.7 EXPERIMENT AND ANALYSIS .....	33
5.71 EFFECT WITH DIFFERENT SKIN TONES .....	34
5.72 EFFECT OF TRAINING PATTERN .....	34
5.73 GESTURE RECOGNITION .....	35

5.8 FAILURE ANALYSIS .....	36
5.9 CONVENTIONAL TEST STRATEGIES APPLIED .....	37
5.91 UNIT TESTING .....	37
5.92 INTEGRATION TESTING .....	37
5.93 RECOVERY TESTING .....	37
5.94 SENSITIVITY TESTING .....	37
 <b>CHAPTER 6: CONCLUSION AND FUTURE WORK</b>	 <b>38</b>
6.1 CONCLUSION .....	38
6.2 FUTURE WORK.....	39
 <b>REFERENCES.....</b>	 <b>40</b>

---

**LIST OF FIGURES**

<b>TABLE OF CONTENTS.....</b>	<b>VIII</b>
FIGURE: 1.1: LIGHTING CONDITION AND BACKGROUND .....	2
FIGURE 1.2: HAND GESTURE RECOGNITION FLOW CHART .....	3
FIGURE 2.1: THE EFFECT OF SELF SHADOWING (A) AND CAST SHADOWING (B) [25].....	7
FIGURE 3.1: SYSTEM IMPLIMENTATION .....	15
FIGURE 3.2: DIFFERENT ETHNIC GROUP SKIN PATCHES .....	17
FIGURE 3.3: REMOVAL OF BACKGROUND.....	18
FIGURE 3.4: LABELING SKIN REGION.....	19
FIGURE 3.5: REAL TIME CLASSIFICATION .....	20
FIGURE 4.1: NEURAL NET BLOCK DIAGRAM .....	21
FIGURE 4.2: NN FOR ROW VECTOR AND EDGING ROW VECTOR .....	212
FIGURE 4.3: NN FOR MEAN AND STANDARD DEVIATION.....	212
FIGURE 4.4 ROW VECTOR OF AN IMAGE .....	23
FIGURE 4.5: ROW VECTOR FLOW CHART .....	23
FIGURE 4.6: EDGING AND ROW VECTOR FLOW CHART.....	24
FIGURE 4.7: MEAN & S.D FLOW CHART .....	25
FIGURE 4.8: DIAGONAL SUM .....	26
FIGURE 4.9: DIAGONAL SUM FLOW CHART .....	26
FIGURE 4.10: GRAPHICAL USER INTERFACE .....	27
FIGURE 4.11: NN TRAINING.....	28
FIGURE 4.12: PERFORMANCE CHART.....	268
FIGURE 4.13: GRAPHICAL USER INTERFACE OUTPUT .....	269
FIGURES 5.1: PERFORMANCE PERCENTAGE .....	31
FIGURES 5.2: DEGREE OF ROTATION CLASSIFICATION .....	313
FIGURES 5.3: DEGREE OF ROTATION MISCLASSIFICATION .....	31
FIGURES 5.4: ETHNIC GROUP SKIN DECTION .....	314
FIGURES 5.5: MIX TRAINING PATTERN.....	315
FIGURES 5.6: CLASSIFICATION PERCENTAGE .....	316



**LIST OF TABLES**

**TABLE OF CONTENTS.....IX**

**TABLE 1: PROCESSING TIME (TESTING)..... 32**

# CHAPTER 1

## INTRODUCTION

Recent developments in computer software and related hardware technology have provided a value added service to the users. In everyday life, physical gestures are a powerful means of communication. They can economically convey a rich set of facts and feelings. For example, waving one's hand from side to side can mean anything from a "happy goodbye" to "caution". Use of the full potential of physical gesture is also something that most human computer dialogues lack [14].

The task of hand gesture recognition is one the important and elemental problem in computer vision. With recent advances in information technology and media, automated human interactions systems are build which involve hand processing task like hand detection, hand recognition and hand tracking.

This prompted my interest so I planned to make a software system that could recognize human gestures through computer vision, which is a sub field of artificial intelligence. The purpose of my software through computer vision was to program a computer to "understand" a scene or features in an image.

A first step in any hand processing system is to detect and localize hand in an image. The hand detection task was however challenging because of variability in the pose, orientation, location and scale. Also different lighting conditions add further variability.

### 1.1 DIGITAL IMAGE PROCESSING

Image processing is reckoned as one of the most rapidly involving fields of the software industry with growing applications in all areas of work. It holds the possibility of developing the ultimate machines in future, which would be able to perform the visual function of living beings. As such, it forms the basis of all kinds of visual automation.

### 1.2 BIOMETRICS

Biometric systems are systems that recognize or verify human beings. Some of the most important biometric features are based physical features like hand, finger, face and eye. For instance finger print recognition utilizes of ridges and furrows on skin surface of the palm and fingertips. Hand gesture detection is related to the location of the presence of a hand in still image or in sequence of images i.e. moving images. Other biometric features are determined by human behavior like voice, signature and walk. The way humans generate sound for mouth, nasal cavities and lips is used for voice recognition. Signature recognition looks at the pattern, speed of the pen when writing ones signature.

---

## 1.3 HAND GESTURE DETECTION AND RECOGNITION

### 1.3.1 DETECTION

Hand detection is related to the location of the presence of a hand in a still image or sequence of images i.e. moving images. In case of moving sequences it can be followed by tracking of the hand in the scene but this is more relevant to the applications such as sign language. The underlying concept of hand detection is that human eyes can detect objects which machines cannot with that much accuracy as that of a human. From a machine point of view it is just like a man fumble around with his senses to find an object.

The factors, which make the hand detection task difficult to solve, are:

#### Variations in image plane and pose

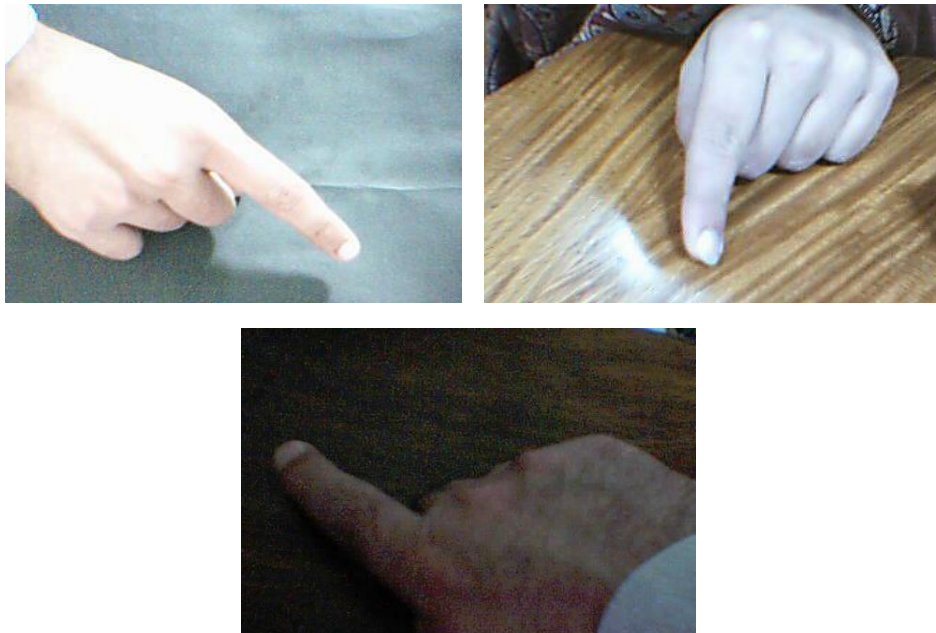
The hands in the image vary due to rotation, translation and scaling of the camera pose or the hand itself. The rotation can be both in and out of the plane.

#### Skin Color and Other Structure Components

The appearance of a hand is largely affected by skin color, size and also the presence or absence of additional features like hairs on the hand further adds to this variability.

#### Lighting Condition and Background

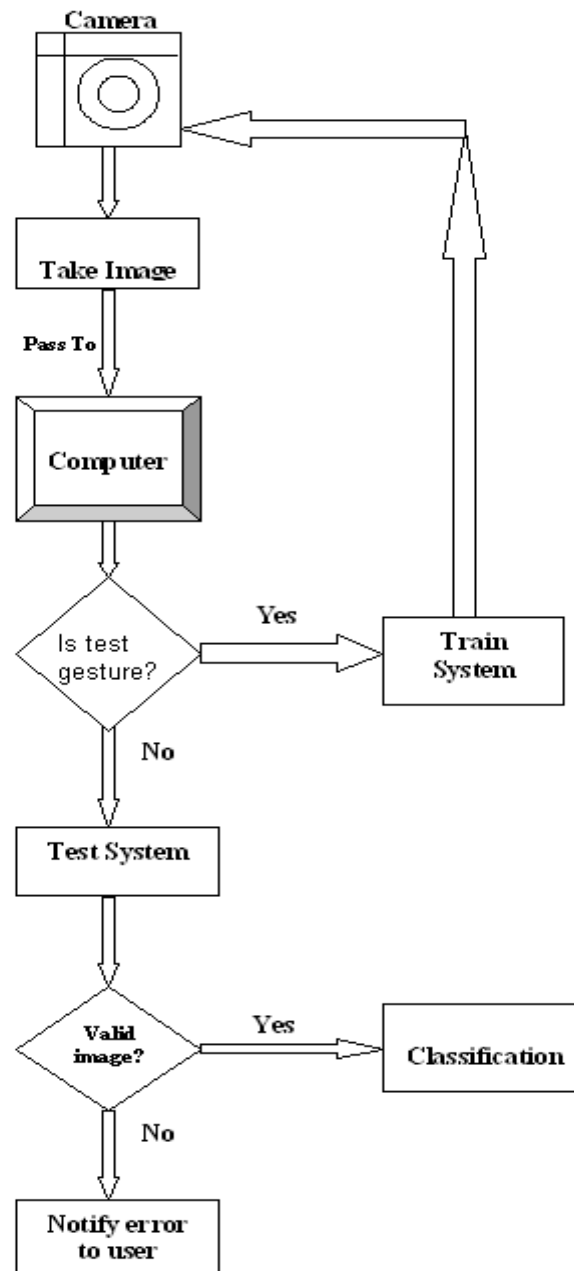
As shown in Figure 1.1 light source properties affect the appearance of the hand. Also the background, which defines the profile of the hand, is important and cannot be ignored.



**Figure: 1.1: Lighting Condition and Background**

### 1.3.2 RECOGNITION

Hand detection and recognition have been significant subjects in the field of computer vision and image processing during the past 30 years. There have been considerable achievements in these fields and numerous approaches have been proposed. However, the typical procedure of a fully automated hand gesture recognition system can be illustrated in the **Figure 1.2** below:



**Figure 1.2: Hand Gesture Recognition Flow Chart**

## 1.4 MOTIVATION

Biometric technologies make use of various physical and behavioral characteristics of human such as fingerprints, expression, face, hand gestures and movement. These features are then processed using sophisticated machines for detection and recognition and hence used for security purposes. Unlike common security measures such as passwords, security cards that can easily be lost, copied or stolen; these biometric features are unique to individuals and there is little possibility that these pictures can be replaced or altered.

Among the biometric sector hand gesture recognition are gaining more and more attention because of their demand regarding security for law enforcement agency as well as in private sectors such as surveillance systems.

In video conferencing system, there is a need to automatically control the camera in such a way that the current speaker always has the focus. One simple approach to this is to guide the camera based on sound or simple cues such as motion and skin color.

Hand gestures are important to intelligent human and computer interaction to build fully automated systems that analyze information contained in images, fast and efficient hand gesture recognition algorithms are required.

## 1.5 SCOPE

The scope of this project is to build a real time gesture classification system that can automatically detect gestures in natural lighting condition. In order to accomplish this objective, a real time gesture based system is developed to identify gestures.

This system will work as one of futuristic of Artificial Intelligence and computer vision with user interface. Its create method to recognize hand gesture based on different parameters. The main priority of this system is to simple, easy and user friendly without making any special hardware. All computation will occur on single PC or workstation. Only special hardware will use to digitize the image (Digital Camera).

## 1.6 SOFTWARE TOOLS

Due to the time constraint and complexity of implementing system in C++, the aim was to design a prototype under MATLAB that was optimized for detection performance. A system that accepted varying inputs of different sizes and image resolutions was implemented; constructing a well coded and documented system for easier future development.

## **1.7 OBJECTIVES**

First objective of this project is to create a complete system to detect, recognize and interpret the hand gestures through computer vision

Second objective of the project is therefore to provide a new low-cost, high speed and color image acquisition system.

## CHAPTER 2

### LITERATURE REVIEW

#### INTRODUCTION

Hand gesture recognition research is classified in three categories. First “**Glove based Analysis**” attaching sensor with gloves mechanical or optical to transduces flexion of fingers into electrical signals for hand posture determination and additional sensor for position of the hand. This sensor is usually an acoustic or a magnetic that attached to the glove. Look-up table software toolkit provided for some applications to recognize hand posture.

The second approach is “**Vision based Analysis**” that human beings get information from their surroundings, and this is probably most difficult approach to employ in satisfactory way. Many different implementations have been tested so far. One is to deploy 3-D model for the human hand. Several cameras attached to this model to determine parameters corresponding for matching images of the hand, palm orientation and joint angles to perform hand gesture classification. Lee and Kunii developed a hand gesture analysis system based on a three-dimensional hand skeleton model with 27 degrees of freedom. They incorporated five major constraints based on the human hand kinematics to reduce the model parameter space search. To simplify the model matching, specially marked gloves were used [3].

The Third implementation is “**Analysis of drawing gesture**” use stylus as an input device. These drawing analysis lead to recognition of written text. Mechanical sensing work has used for hand gesture recognition at vast level for direct and virtual environment manipulation. Mechanically sensing hand posture has many problems like electromagnetic noise, reliability and accuracy. By visual sensing gesture interaction can be made potentially practical but it is most difficult problem for machines.

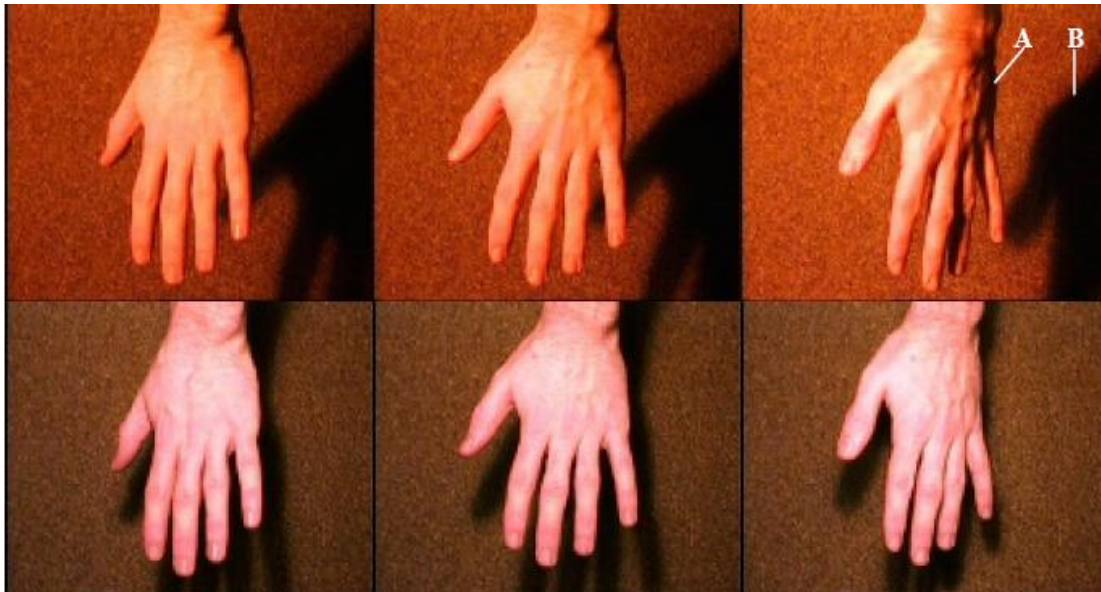
Full American Sign Language recognition systems (words, phrases) incorporate data gloves. *Takashi and Kishino* discuss a Data glove-based system that could recognize 34 of the 46 Japanese gestures (user dependent) using a joint angle and hand orientation coding technique. From their paper, it seems the test user made each of the 46 gestures 10 times to provide data for principle component and cluster analysis. The user created a separate test from five iterations of the alphabet, with each gesture well separated in time. While these systems are technically interesting, they suffer from a lack of training [1, 2].

Excellent work has been done in support of machine sign language recognition by *Sperling and Parish*, who has done careful studies on the bandwidth necessary for a sign conversation using spatially and temporally sub-sampled images. Point light experiments (where “lights” are attached to significant locations on the body and just these points are

used for recognition), have been carried out by *Poizner*. Most systems to date study isolate/static gestures. In most of the cases those are fingerspelling signs [13].

## 2.1 LIGHTING

The task of differentiating the skin pixels from those of the background is made considerably easier by a careful choice of lighting. According to Ray Lockton, if the lighting is constant across the view of the camera then the effects of self-shadowing can be reduced to a minimum [25]. (See Figure 2.1)



**Figure 2.1: The effect of self shadowing (A) and cast shadowing (B) [25].**

The top three images were lit by a single light source situated off to the left. A self-shadowing effect can be seen on all three, especially marked on the right image where the hand is angled away from the source. The bottom three images are more uniformly lit, with little self-shadowing. Cast shadows do not affect the skin for any of the images and therefore should not degrade detection. Note how an increase of illumination in the bottom three images results in a greater contrast between skin and background [25].

The intensity should also be set to provide sufficient light for the CCD in the camera. However, since this system is intended to be used by the consumer it would be a disadvantage if special lighting equipment were required. It was decided to attempt to extract the hand information using standard room lighting. This would permit the system to be used in a non-specialist environment [25].



## **2.2 CAMERA ORIENTATIONS AND DISTANCE**

It is very important to be careful about direction of camera to permit easy choice of background. Two good and more effective approaches are to point the camera towards wall or floor. Lighting was standard room; intensity of light would be higher and shadowing effects lower because camera was pointed downwards. The distance of the camera from the hand should be such that it covers the entire gesture mainly. There is no effect found on the accuracy of the system if the image is a zoomed one or not; the principle is to cover the entire hand area majorly.

## **2.3 BACKGROUND SELECTION**

Another important aspect is to maximize differentiation that the color of background must be different as possible from skin color. The floor color in the work used was black. It was decided to use this color because it offered minimum self-shadowing problem as compared to other background colors.

## **2.4 DIFFERENT RECOGNITION APPROACHES**

The different recognition approaches studied are as follows:

### **2.41 PEN-BASED GESTURE RECOGNITION**

Recognizing gestures from two-dimensional input devices such as a pen or mouse has been considered for some time. The early Sketchpad system in 1963 used light-pen gestures, for example. Some commercial systems have used pen gestures since the 1970s. There are examples of gesture recognition for document editing for air traffic control, and for design tasks such as editing splines.

More recently, systems such as the OGI Quick Set system have demonstrated the utility of pen-based gesture recognition in concert with speech recognition to control a virtual environment. Quick Set recognizes 68 pen gestures, including map symbols, editing gestures, route indicators, area indicators, and taps.

Oviatt has demonstrated significant benefits of using both speech and pen gestures together in certain tasks. Zeleznick and Landay and Myers developed interfaces that recognize gestures from pen-based sketching [3].

There have been commercially available Personal Digital Assistants (PDAs) for several years, starting with the Apple Newton, and more recently the 3Com Palm Pilot and various Windows CE devices. Long, and Rowe survey problems and benefits of these gestural interfaces and provide insight for interface designers. Although pen-based gesture

recognition is promising for many HCI environments, it presumes the availability of, and proximity to, a flat surface or screen. In virtual environments, this is often too constraining – techniques that allow the user to move around and interact in more natural ways are more compelling [3].

## 2.42 TRACKER-BASED GESTURE RECOGNITION

There are many tracking system available commercially which can be used for gesture recognition, primarily tracking eye gaze, hand gesture, and overall body and its position. In virtual environment interaction each sensor has its own strengths and weaknesses. Gestural interface eye gaze can be useful, so I focus here on gesture based input from tracking the hand and the body.

## 2.43 DATA GLOVES

For communication and manipulation people use their hand for wide variety of tasks. Hands including wrist with approximately 29 degrees of freedom are very dexterous and extremely expressive and quite convenient. In variety of application domain, hand could be used for control device with sophisticated input, providing real time control with many degrees of freedom for complex tasks. Sturman analyzed task characteristics and requirements, hand action capabilities, and device capabilities, and discussed important issues in developing whole-hand input techniques [4].

Sturman suggested taxonomy of whole-hand input that categorizes input techniques along two dimensions: Classes of hand actions that could be *continuous* or *discrete* and interpretation of hand actions that could be *direct*, *mapped*, or *symbolic* [4].

Given interaction task, can be evaluated as to which style best suits the task. Mulder presented an overview of hand gestures in human-computer interaction, discussing the classification of hand movement, standard hand gestures, and hand gesture interface design [3].

For the measurement position of hand configuration there are many commercially devices available to calculate the degree of precision, accuracy and completeness. These devices include exoskeleton and instrumented gloves mounted on hand and figure that are known as “Data gloves”. Few advantages of data gloves, direct measurement of hand and finger parameters, provision of data, high sampling frequency, easy if use, line of sight, low cost version and translation independency feature of data.

However with advantages of data glove there are few disadvantages like difficulty in calibration, reduction in range of motion and comfort, noise in inexpensive system, expensiveness of accurate system. Moreover it’s compulsory for user to wear cumbersome device.

Many projects have used hand input from data gloves for “point, reach, and grab” operations or more sophisticated gestural interfaces. Latoschik and Wachsmuth present a

multi-agent architecture for detecting pointing gestures in a multimedia application. Väänänen and Böhm developed a neural network system that recognized static gestures and allows the user to interactively teach new gestures to the system. Böhm *et al.* extend that work to dynamic gestures using a Kohonen Feature Map (KFM) for data reduction [3].

The HIT Lab at the University of Washington developed Glove GRASP, a C/C++ class library that allows software developers to add gesture recognition capabilities to SGI systems, including user-dependent training and one- or two-handed gesture recognition. A commercial version of this system is available from General Reality [3].

## **2.44 BODY SUITS**

Process of small place of strategically dots placed on human body, people can perceive patterns such as gestures, activities, identities and other aspects of body. One way of approach is recognition of postures and human movements is optically measure of 3D position such as markers attached to body and then recovers time varying articulate structure of body. This articulated sensing by position and joint angles using electromechanically sensors. Although some of system require small ball or dot placed top user clothing. I prefer body motion capture by “body suits” generically.

Body suits have advantages and disadvantages similar to those of data gloves. At high sampling rate it provides reliable results but they are cumbersome and very expensive. Non-trivial calibration. Several cameras used by optical system which is typically offline data process, lack of wires and tether is major disadvantages.

## **2.45 HEAD AND FACE GESTURES**

When people interact with one another, they use an assortment of cues from the head and face to convey information. These gestures may be intentional or unintentional, they may be the primary communication mode or back channels, and they can span the range from extremely subtle to highly exaggerate. Some examples of head and face gestures include: nodding or shaking the head, direction of eye gaze, raising the eyebrows, opening the mouth to speak, winking, flaring the nostrils and looks of surprise, happiness, disgust, anger, sadness, etc [5].

People display a wide range of facial expressions. Ekman and Friesen developed a system called FACS for measuring facial movement and coding expression; this description forms the core representation for many facial expression analysis systems [6].

A real-time system to recognize actions of the head and facial features was developed by Zelinsky and Heinzmann, who used feature template tracking in a Kalman filter framework to recognize thirteen head/face gestures [6].

Essa and Pentland used optical flow information with a physical muscle model of the face to produce accurate estimates of facial motion. This system was also used to generate spatio-temporal motion-energy templates of the whole face for each different expression – these templates were then used for expression recognition [3].

## **2.46 HAND AND ARM GESTURES**

These two parts of body (Hand & Arm) have most attention among those people who study gestures in fact much reference only consider these two for gesture recognition. The majority of automatic recognition systems are for deictic gestures (pointing), emblematic gestures (isolated signs) and sign languages (with a limited vocabulary and syntax). Some are components of bimodal systems, integrated with speech recognition. Some produce precise hand and arm configuration while others only coarse motion [3].

Stark and Kohler developed the ZYKLOP system for recognizing hand poses and gestures in real-time. After segmenting the hand from the background and extracting features such as shape moments and fingertip positions, the hand posture is classified. Temporal gesture recognition is then performed on the sequence of hand poses and their motion trajectory. A small number of hand poses comprises the gesture catalog, while a sequence of these makes a gesture [3].

Similarly, Maggioni and Kämmerer described the Gesture Computer, which recognized both hand gestures and head movements. There has been a lot of interest in creating devices to automatically interpret various sign languages to aid the deaf community. One of the first to use computer vision without requiring the user to wear anything special was built by Starner, who used HMMs to recognize a limited vocabulary of ASL sentences. The recognition of hand and arm gestures has been applied to entertainment applications [3].

Freeman developed a real-time system to recognize hand poses using image moments and orientations histograms, and applied it to interactive video games. Cutler and Turk described a system for children to play virtual instruments and interact with life like characters by classifying measurements based on optical flow [3].

## **2.47 BODY GESTURES**

This section includes tracking full body motion, recognizing body gestures, and recognizing human activity. Activity may be defined over a much longer period of time than what is normally considered a gesture; for example, two people meeting in an open area, stopping to talk and then continuing on their way may be considered a recognizable activity. Bobick proposed taxonomy of motion understanding in terms of: Movement – the atomic elements of motion, Activity – a sequence of movements or static configurations and Action – high-level description of what is happening in context.

Most research to date has focused on the first two levels [3].

The Pfinder system developed at the MIT Media Lab has been used by a number of groups to do body tracking and gesture recognition. It forms a two-dimensional representation of the body, using statistical models of color and shape. The body model provides an effective interface for applications such as video games, interpretive dance, navigation, and interaction with virtual characters [3].

Lucente combined Pfinder with speech recognition in an interactive environment called Visualization Space, allowing a user to manipulate virtual objects and navigate through virtual worlds [3].

Paradiso and Sparacino used Pfinder to create an interactive performance space where a dancer can generate music and graphics through their body movements – for example, hand and body gestures can trigger rhythmic and melodic changes in the music [3].

Systems that analyze human motion in virtual environments may be quite useful in medical rehabilitation and athletic training. For example, a system like the one developed by Boyd and Little to recognize human gaits could potentially be used to evaluate rehabilitation progress [3].

Davis and Bobick used a view-based approach by representing and recognizing human action based on “temporal templates,” where a single image template captures the recent history of motion. This technique was used in the Kids Room system, an interactive, immersive, narrative environment for children [3].

Video surveillance and monitoring of human activity has received significant attention in recent years. For example, the W4 system developed at the University of Maryland tracks people and detects patterns of activity [3].

## **2.48 VISION-BASED GESTURE RECOGNITION**

The most significant disadvantage of the tracker-based systems is that they are cumbersome. This detracts from the immerse nature of a virtual environment by requiring the user to put on an unnatural device that cannot easily be ignored, and which often requires significant effort to put on and calibrate. Even optical systems with markers applied to the body suffer from these shortcomings, albeit not as severely. What many have wished for is a technology that provides real-time data useful for analyzing and recognizing human motion that is passive and non-obtrusive. Computer vision techniques have the potential to meet these requirements.

Vision-based interfaces use one or more cameras to capture images, at a frame rate of 30 Hz or more, and interpret those images to produce visual features that can be used to interpret human activity and recognize gestures [3].

Typically the camera locations are fixed in the environment, although they may also be mounted on moving platforms or on other people. For the past decade, there has been a significant amount of research in the computer vision community on detecting and recognizing faces, analyzing facial expression, extracting lip and facial motion to aid speech recognition, interpreting human activity, and recognizing particular gestures [3].

Unlike sensors worn on the body, vision approaches to body tracking have to contend with occlusions. From the point of view of a given camera, there are always parts of the user's body that are occluded and therefore not visible – e.g., the backside of the user is not visible when the camera is in front. More significantly, self-occlusion often prevents a full view of the fingers, hands, arms, and body from a single view. Multiple cameras can be used, but this adds correspondence and integration problems [3].

The occlusion problem makes full body tracking difficult, if not impossible, without a strong model of body kinematics and perhaps dynamics. However, recovering all the parameters of body motion may not be a prerequisite for gesture recognition. The fact that people can recognize gestures leads to three possible conclusions: we infer the parameters that we cannot directly observe, we don't need these parameters to accomplish the task, and we infer some and ignore others [3].

Unlike special devices, which measure human position and motion, vision uses a multipurpose sensor; the same device used to recognize gestures can be used to recognize other objects in the environment and also to transmit video for teleconferencing, surveillance, and other purposes. There is a growing interest in CMOS-based cameras, which promise miniaturized, low cost, low power cameras integrated with processing circuitry on a single chip [3].

Currently, most computer vision systems use cameras for recognition. Analog cameras feed their signal into a digitizer board, or frame grabber, which may do a DMA transfer directly to host memory. Digital cameras bypass the analog-to-digital conversion and go straight to memory. There may be a preprocessing step, where images are normalized, enhanced, or transformed in some manner, and then a feature extraction step. The features – which may be any of a variety of two- or three-dimensional features, statistical properties, or estimated body parameters – are analyzed and classified as a particular gesture if appropriate [3].

This technique was also used by us for recognizing hand gestures in real time. With the help of a web camera, I took pictures of hand on a prescribed background and then applied the classification algorithm for recognition.

## CHAPTER 3

### METHODOLOGY

#### INTRODUCTION

There have been numerous researches in this field and several methodologies were proposed like Principle Component Analysis (PCA) method, gradient method, subtraction method etc. PCA relates to Linear transformation consist on statistical approach. This gives us powerful tool for pattern recognition and data analysis which mostly used in image processing techniques for data (compression, dimension and correlation). Gradient method is also another image processing technique that detect colour patches applying low pass filters is also known as edge detection method. Subtraction method is very simple that subtract input image pixel to another image or constant value to provide output. I have also studied different approaches to hand gesture recognition and came to know that implementation of such techniques like PCA and Gradient method is complicated, we can produce same output as these techniques gives us by simple and easy implementation. So, I have tried four different algorithms and finally selected the one, which was most efficient i.e. diagonal sum algorithm. This algorithm is able to recognize maximum gestures correctly.

#### 3.1 PROJECT CONSTRAINTS

I propose a vision-based approach to accomplish the task of hand gesture detection. As discussed above, the task of hand gesture recognition with any machine learning technique suffers from the variability problem. To reduce the variability in hand recognition task we assume the following assumptions:

- Single colored camera mounted above a neutral colored desk.
- User will interact by gesturing in the view of the camera.
- Training is must.
- Hand will not be rotated while image is capturing.

The real time gesture classification system depends on the hardware and software.

##### Hardware

- Minimum 2.8 GHz processor Computer System or latest
- 52X CD-ROM drive
- Web cam (For real-time hand Detection)

---

**Software**

- Windows 2000(Service Pack 4),XP, Vista or Windows 7
- Matlab 8.0 or latest (installed with image processing toolbox)
- Vcapg2.dll (Video Capture Program Generation 2)
- DirectX 9.0 (for supporting Vcapg2)

**3.2 THE WEBCAM SYSTEM (USB PORT)**

Below is the summary of the specifications of the camera which this system required:

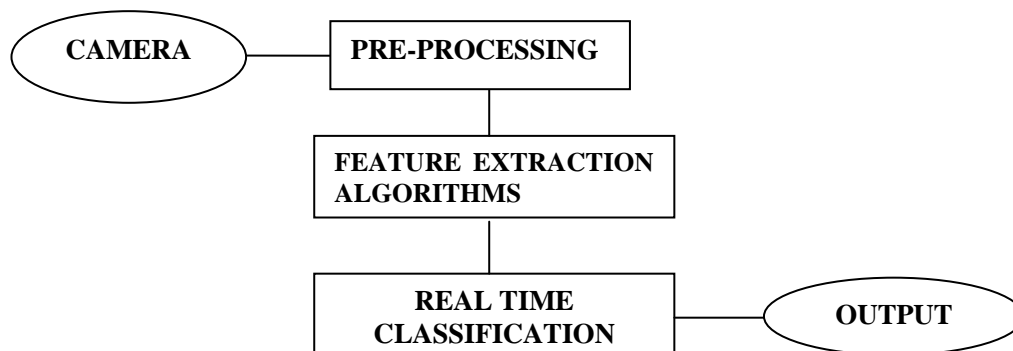
**Resolution:** 640x480  
**Video frame rate:** 30fps @640x480  
**Pixel depth:** Minimum 1.3-mega pixels  
**Connection port:** USB

In my project web cam was attached via USB port of the computer. The web cam worked by continually capturing the frames. In order to capture a particular frame, the user just need to select the particular Algorithm METHOD button on the interface and the hand was detected in the particular frame. The web cam took color pictures, which were then converted into grayscale format. The main reason of sticking to grayscale was the extra amount of processing required to deal with color images.

**3.3 BRIEF OUTLINE OF THE IMPLEMENTED SYSTEM**

Hand gesture recognition system can be divided into following modules:

- Preprocessing
- Feature extraction of the processed image
- Real time classification



**Figure 3.1: System Implementation**



### 3.31 PRE-PROCESSING

Like many other pattern recognition tasks, pre-processing is necessary for enhancing robustness and recognition accuracy.

The preprocessing prepares the image sequence for the recognition, so before calculating the diagonal Sum and other algorithms, a pre-processing step is performed to get the appropriate image, which is required for real time classification. So it consists of some steps. The net effect of this processing is to extract the hand only from the given input because once the hand is detected from the given input it can be recognized easily. So pre-processing step mainly consists of following tasks:

- **Skin Modeling**
- **Removal of Background**
- **Conversion from RGB to binary**
- **Hand Detection**

#### 3.31.1 Skin Modelling

There are numerous method used for skin detection such as RGB (Red, Green, Blue), YCbCr (Luminance Chrominance) and HSV (Hue, Saturation, Value).

- **RGB:**

RGB is a 3D color space pixel where each pixel has combination of three colors Red, Green and Blue at specific location. This technique widely used in image processing for identifying skin region.

- **YCbCr (Luminance Chrominance):**

This color space is used in digital video color information represent two color Cb and Cr. Cb is difference between Blue and Cr is difference between Red component references of value. This is basically RGB transformation to YCbCr for separation of luminance and chrominance for color modelling.

- **HSV (Hue, Saturation and Value):**

In HSV, Hue detect dominant color and Saturation define colourfulness whilst Value measure intensity or brightness. This is well enough to choose single color but it ignores complexity of color appearance. It trade off computation speed mean computationally expensive and perceptual relevance.




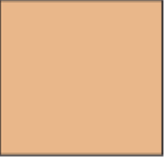


**My approach for this thesis is to work with RGB to binarization techniques to Explicitly Defined skin Region.**

- **Skin Detection:**

The skin color detection is one of important goal in hand gesture recognition. Skin color detection decision rules which we have to build that will discriminate between skin portion and non-skin portion pixels. This is accomplished usually by metric introduction, which measure distance of the pixel color. This metric type is knows as skin modelling.

- **Explicitly Defined Skin Region**

Following are some common ethnic skin groups and there RGB color space:

					
<b>European</b>	<b>Middle Eastern</b>	<b>Eastern</b>	<b>Asian</b>	<b>Lt. Black</b>	<b>Dk. Black</b>
R=245 G=218 B=204	R=237 G=191 B=166	R=211 G=141 B=111	R=233 G=183 B=138	R=197 G=132 B=92	R=96 G=59 B=43

**Figure 3.2: Different Ethnic Group Skin Patches**

To build a skin classifier is to define explicitly through a number of rules the boundaries of skin color cluster in some color space. The advantage of this method is the simplicity of skin detection rules that leads to the construction of very rapid classifier. For Example [7]

(R,G,B) is classified as skin if:  
 $R > 95$  and  $G > 40$  and  $B > 20$  and  
 $\max\{R,G,B\} - \min\{R,G,B\} > 15$  and  
 $|R-G| > 15$  and  $R > G$  and  $R > B$

In this classifier threshold defined to maximize the chance for recognizing the skin region for each color. If we see in **Figure 3.2** that Red color in every skin sample is greater than 95, Green is greater than 40 and Blue is greater than 20 in. So threshold can make this classifier easily detect almost all kind of skin.

This is one of the easiest methods as it explicitly defines skin-color boundaries in different color spaces. Different ranges of thresholds are defined according to each color space components in as the image pixels that fall between the predefined ranges are considered

as skin pixels. The advantage of this method is obviously the simplicity which normally avoids of attempting too complex rules to prevent over fitting data. However, it is important to select good color space and suitable decision rules to achieve high recognition rate with this method. [8]

### 3.31.2 Removal of Background

I have found that background greatly affects the results of hand detection that's why I have decided to remove it. For this I have written our own code in spite of using any built-in ones.



Figure: 3.3: Removal of Background

### 3.31.3 Conversion from RGB to Binary

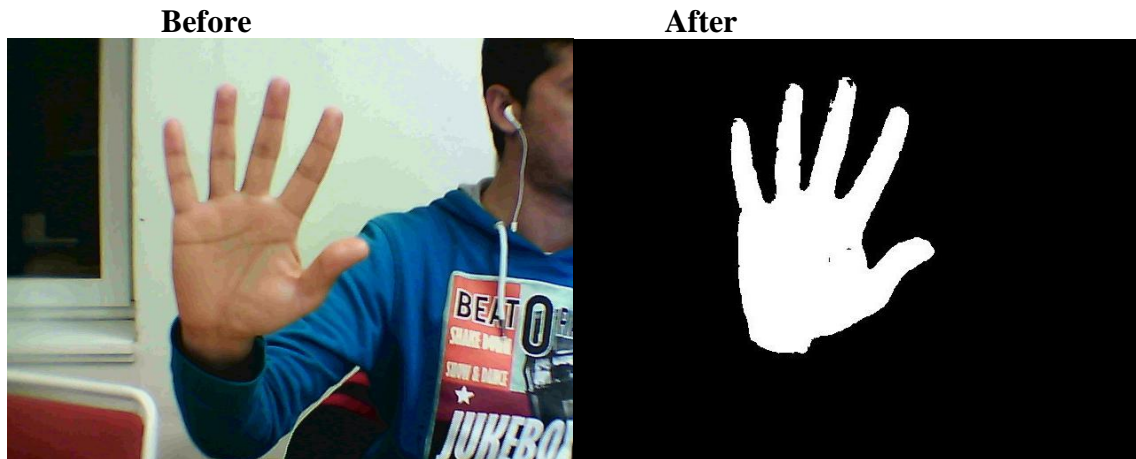
All algorithms accept an input in RGB form and then convert it into binary format in order to provide ease in recognizing any gesture and also retaining the luminance factor in an image.

### 3.31.4 Hand detection

Image could have more than one skin area but we required only hand for further process. For this I choose criteria image labeling which is following:

- **Labeling:**

To define how many skin regions that we have in image is by labelling all skin regions. Label is basically an integer value have 8 connecting objects in order to label all skin area pixel. If object had label then mark current pixel with label if not then use new label with new integer value. After counting all labelled region (segmented image) I sort all them into ascending order with maximum value and choose the area have maximum value which I interested because I assume that hand region in bigger part of image. To separate that region which looked for, create new image that have one in positions where the label occurs and others set to zero.



**Figure 3.4: Labeling Skin Region**

### **3.32 FEATURE EXTRACTION ALGORITHMS**

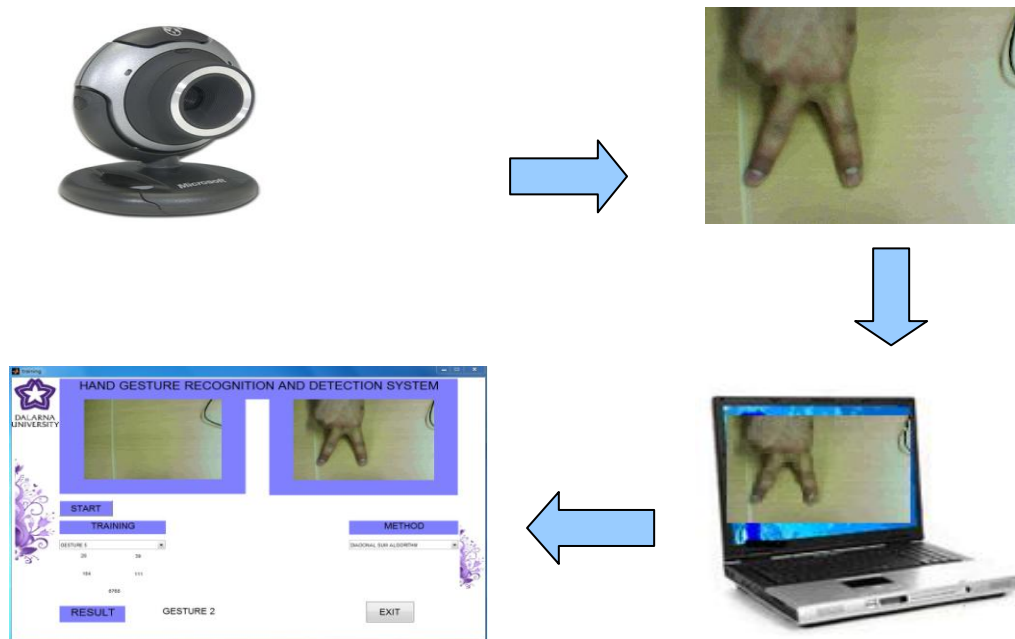
There are four types of algorithms that I studied and implemented namely as followings:

- **Row vector algorithm**
- **Edging and row vector passing**
- **Mean and standard deviation of edged image**
- **Diagonal sum algorithm**

For details of these algorithms, see Chapter4.

### 3.33 REAL TIME CLASSIFICATION

**Figure 3.5** shows the concept for real time classification system. A hand gesture image will be passed to the computer after being captured through camera at run time and the computer will try to recognize and classify the gesture through computer vision.



**Figure 3.5: Real Time Classification**

In real time classification the system developed tries to classify the gestures not saved before but given at the run time. The system first trains itself with the user count gestures at the run time and then tries to classify the newly given test gestures by the user. The algorithm used by the system for real time classification is the diagonal sum algorithm.

## CHAPTER 4

### FEATURE EXTRACTIONS

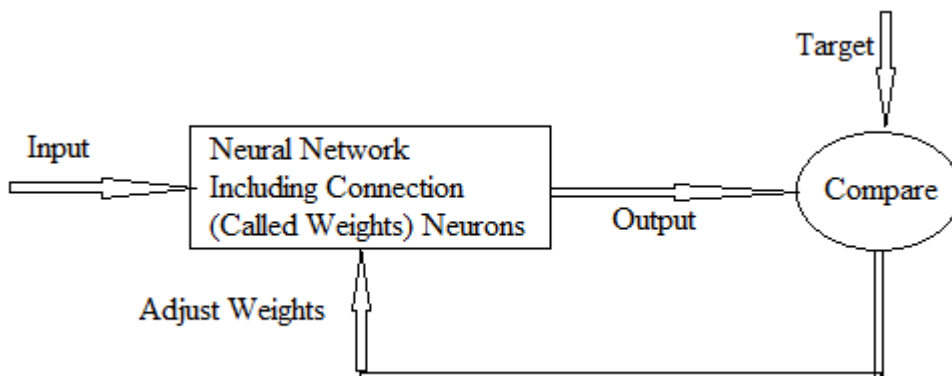
#### INTRODUCTION

In this chapter, I described the detail of all the four features extraction algorithms. First I would like to discuss neural network used in first three algorithms.

#### 4.1 NEURAL NETWORKS

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements [9].

Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown in Figure 4.1 below. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used, in this supervised learning to train a network [9].



**Figure 4.1 Neural net Block Diagram**

Neural networks have been trained to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, and vision and control systems. Today neural networks can be trained to solve problems that are difficult for conventional computers or human beings [9].

Once data ready for representation then next step is to design NN for training and testing data. In first two algorithms Row Vector and Edging and Row Vector passing algorithm have three layers feed forward network: Input, Hidden and Output. Number of neuron in Input is 640 which are equal to number of features extracted from each of algorithm and one neuron for Output layer for skin class to be recognized. But for Mean and standard deviation there are only two input which is also equal to extracted features from this algorithm. Neural network Architecture has number of parameter such as learning rate (lr), number of epochs and stopping criteria which is based on validation of data. Training of Mean Square Error at output layer which is set trial values and which is set by several experiments.

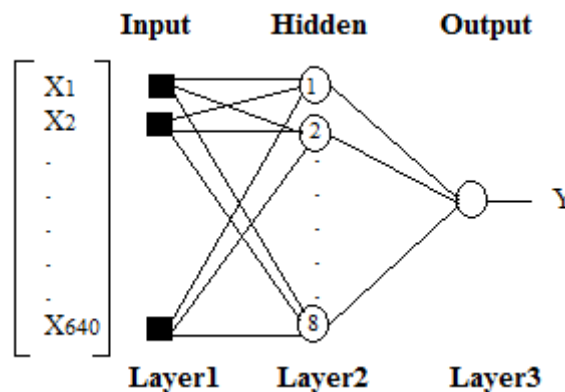


Figure 4.2: NN for Row Vector and Edging Row Vector

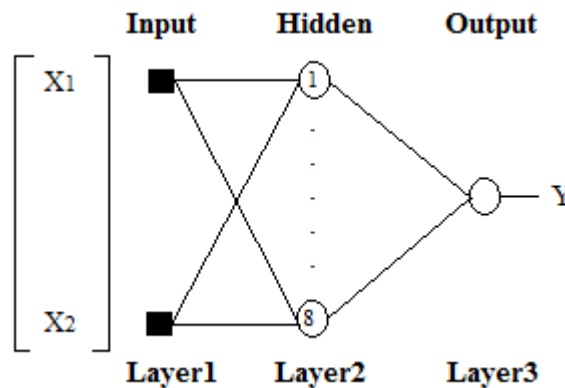
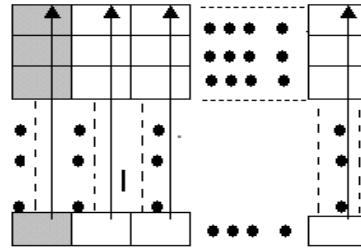


Figure 4.3: NN for Mean and S.D

## 4.2 ROW VECTOR ALGORITHM

We know that behind every image is a matrix of numbers with which we do manipulations to derive some conclusion in computer vision. For example we can calculate a row vector of the matrix. A row vector is basically a single row of numbers with resolution  $1 \times Y$ , where  $Y$  is the total no of columns in the image matrix. Each element in the row vector represents the sum of its respective column entries as illustrated in **Figure 4.4:**



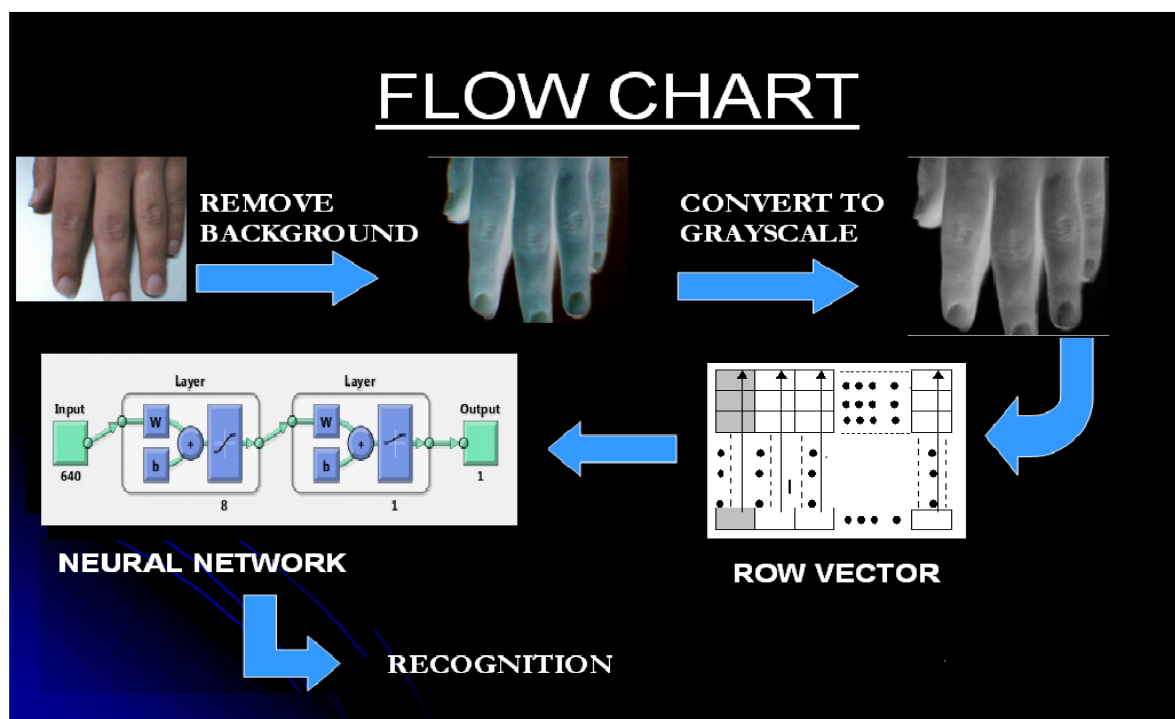
**Figure 4.4 Row vector of an image**

The first algorithm I studied and implemented makes use of the row vector of the hand gestures. For each type of hand gesture, I took several hand images, do skin modeling, labeling, removed their background and RGB to binary conversion in the preprocessing phase, calculated their row vectors and then trained the neural network with these row vectors. Ultimately, the neural network was able to recognize the row vectors that each gesture count can possibly have. Hence, after training, the system was tested to see the recognition power it had achieved.

Mathematically, we can describe the image for training or testing purpose given to the neural network as:

Input to neural network = Row vector (After image Preprocessing)

The flowchart of the algorithm is given below in **Figure 4.5**:



**Figure 4.5: Row Vector Flow Chart**



### 4.3 EDGING AND ROW VECTOR PASSING ALGORITHM

In the pre-processing phase of this algorithm, I do preprocessing, skin modeling and removed the background etc. of the gesture image taken. This image was converted from RGB into grayscale type. Gray scale images represent an image as a matrix where every element has a value corresponding to how bright/dark the pixel at the corresponding position should be colored.

For representing the brightness of pixels there are two ways for represent numbers, First class called Double class that assign floating numbers (“decimals”) between 0 and 1 for each pixel. The zero (0) value represent black and value one (1) corresponds to white. The second class known as unit8 that assign integer between 0 and 255 for brightness of pixel, zero (0) correspond to black and 255 for white. The unit8 class requires less storage than double roughly 1/8.

After the conversion of the image into grayscale, I took the edge of the image with a fixed threshold i.e. 0.5. This threshold helped us in removing the noise in the image. In the next step, a row vector of the edged image was calculated. This row vector is then passed on to the neural network for training. The neural network (NN) is later on tested for the classification of the gestures.

Mathematically, the input to the neural network is given as:

$$\text{Input to NN} = \text{Row vector [Edge (Grayscale image)]}$$

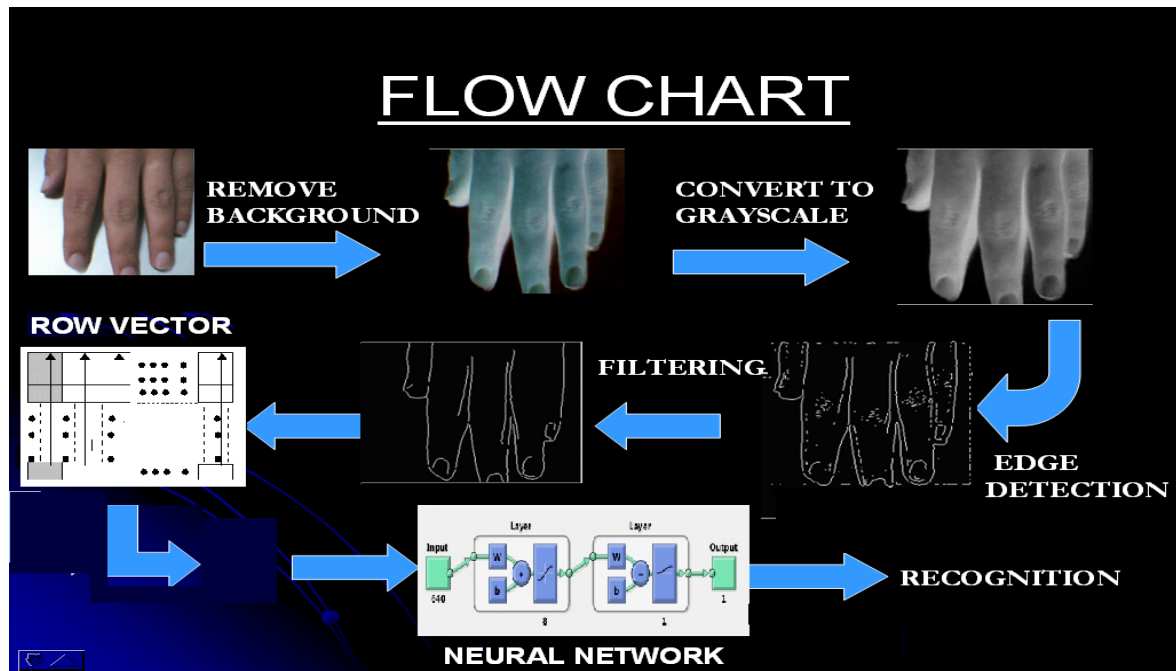


Figure 4.6: Edging and Row Vector Flow Chart

#### 4.4 MEAN AND STANDARD DEVIATION OF EDGED IMAGE

In the pre-processing phase, doing several step like removing the background and RGB image is converted into grayscale type as done in the previous algorithm. The edge of the grayscale image is taken with a fixed threshold i.e. 0.5 then calculate the mean and standard deviation the processed image.

Mean is calculated by taking a sum of all the pixel values and dividing it by the total no of values in the matrix. Mathematically, it is defined as:

$$\bar{X} = \sum_{i=1}^n X_i / n$$

Stand Deviation can calculate from mean which is mathematically defined as:

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}$$

The mean and standard deviation of each type of count gesture are given to the neural network for training. In the end, the system is tested to see the success rate of classification this technique provides. Mathematically, the input given to the neural network is defined as:

Input to NN= Mean (Edge (Binary image)) + S.D (Edge (Binary Image))

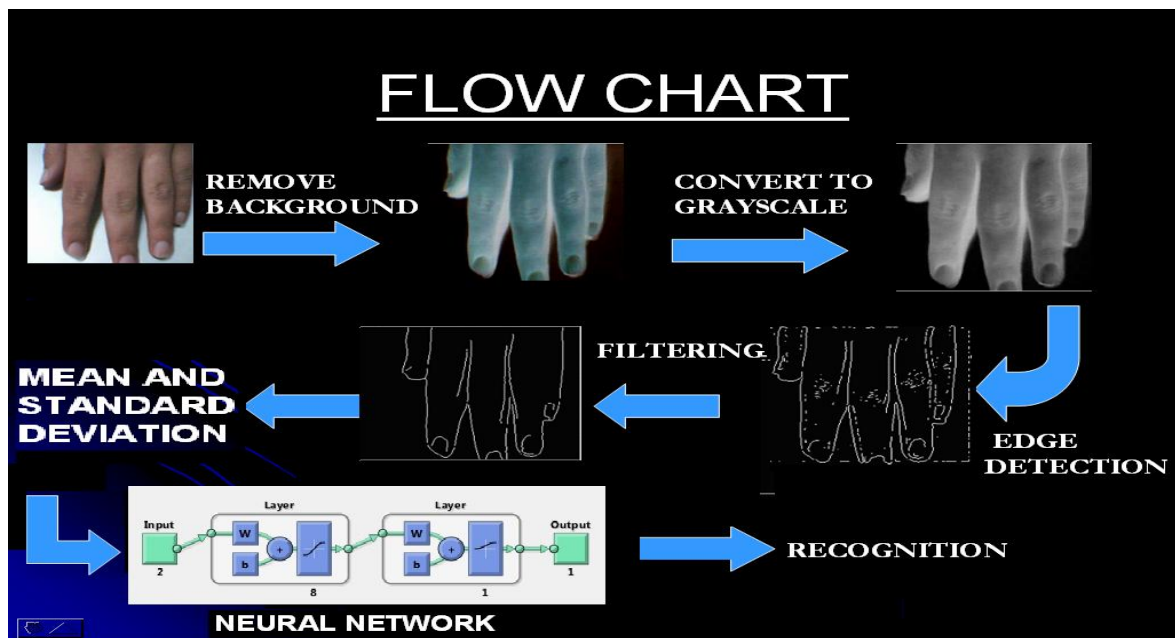


Figure 4.7: Mean & S.D Flow Chart

## 4.5 DIAGONAL SUM ALGORITHM

In the pre-processing phase, doing mentioned steps in methodology, skin modeling removal of the background, conversion of RGB to binary and labeling. The binary image format also stores an image as a matrix but can only color a pixel black or white (and nothing in between). It assigns a 0 for black and a 1 for white. In the next step, the sum of all the elements in every diagonal is calculated. The main diagonal is represented as  $k=0$  in Figure 4.7 given below; the diagonals below the main diagonal are represented by  $k<0$  and those above it are represented as  $k>0$

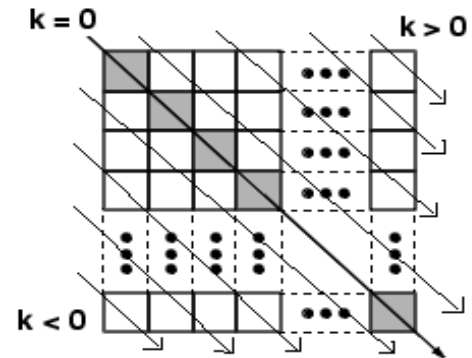


Figure 4.8: Diagonal Sum

The gesture recognition system developed through this algorithm first train itself with the diagonals sum of each type of gesture count at least once, and then its power could be tested by providing it with a test gesture at real time. Mathematically, the input given to the system at real time is given as:

$$X_i = \sum_{i=1}^n \text{Diagonals}$$

$$\text{Input} = \sum_{i=1}^n X_i$$

The flowchart of the algorithm is given below in Figure 4.9:

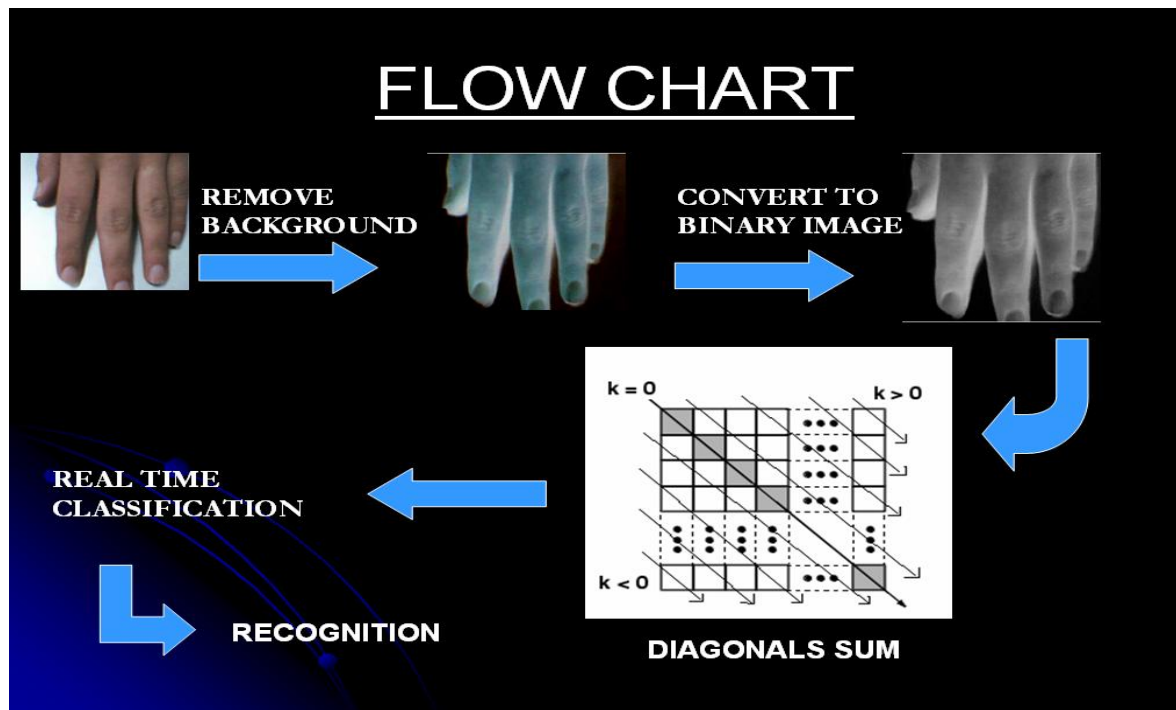


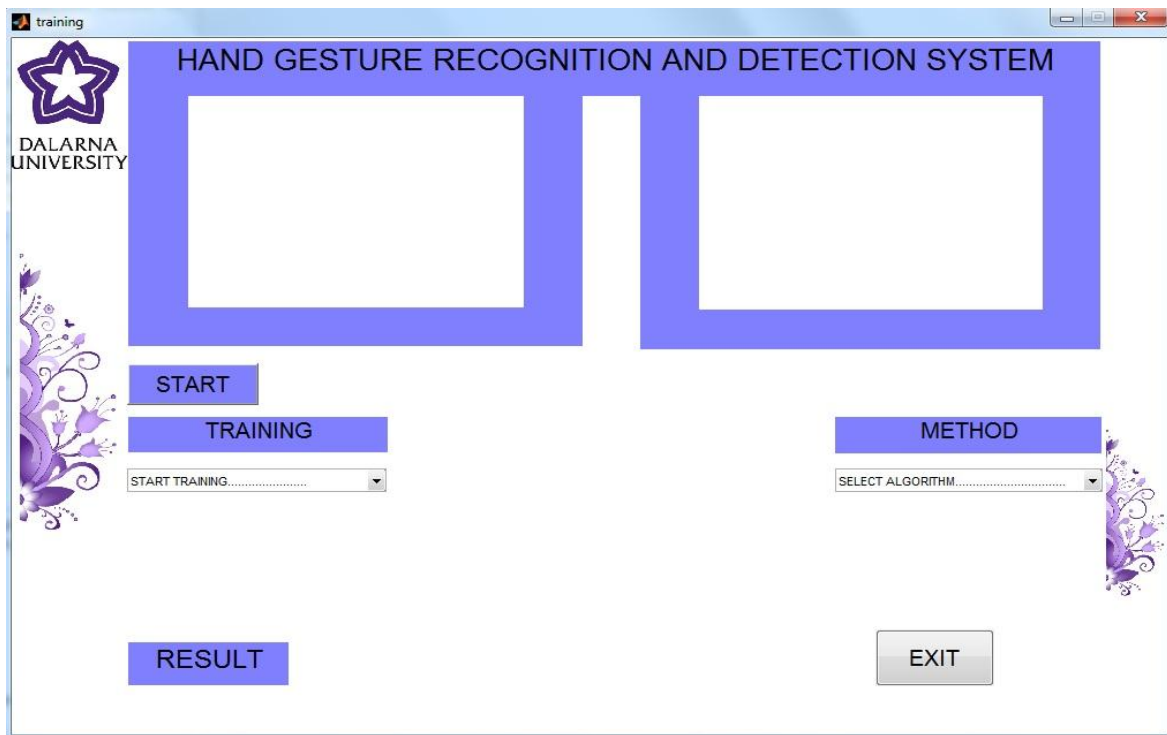
Figure 4.9: Diagonal Sum Flow Chart

## 4.6 GRAPHICAL USER INTERFACE (GUI)

GUIDE is MATLAB's Graphical User Interface Development Environment. GUIDE use for GUIs containing various style figure windows of user interface objects. For creating GUI each object must be programmed to activate user interface.

### 4.6.1 GUI DESIGN

The next stage was to design a GUI such that it reflected the GUI requirements stated above. Following Figure 4.9 shows the GUI design:



**Figure 4.10: Graphical User Interface**

**START** button activates the web cam. User can view his/her hand in the first box, which is above the start button. On selecting any gesture from drop down list under the **TRAINING** button, image will be captured and displayed in the right hand box.

My first step is training for which we capture different images and select respective options of gesture numbers from the drop down menu in order to train the system. When an option is chosen from the drop down menu, the user is asked to enter a name for the training image. Recognition process can now be started by capturing a test gesture and then clicking the any of algorithm under **METHOD** button. This displays a save window that stores your test gesture by the name you give it. After that a progress bar indicates the processing of the system (i.e. preprocessing and recognition phase). The result of the

system will appear in front of the **RESULT** textbox. **EXIT** button enables the user to quit the MAT LAB.

#### 4.62 Neural Network Training If NN algorithm selected:

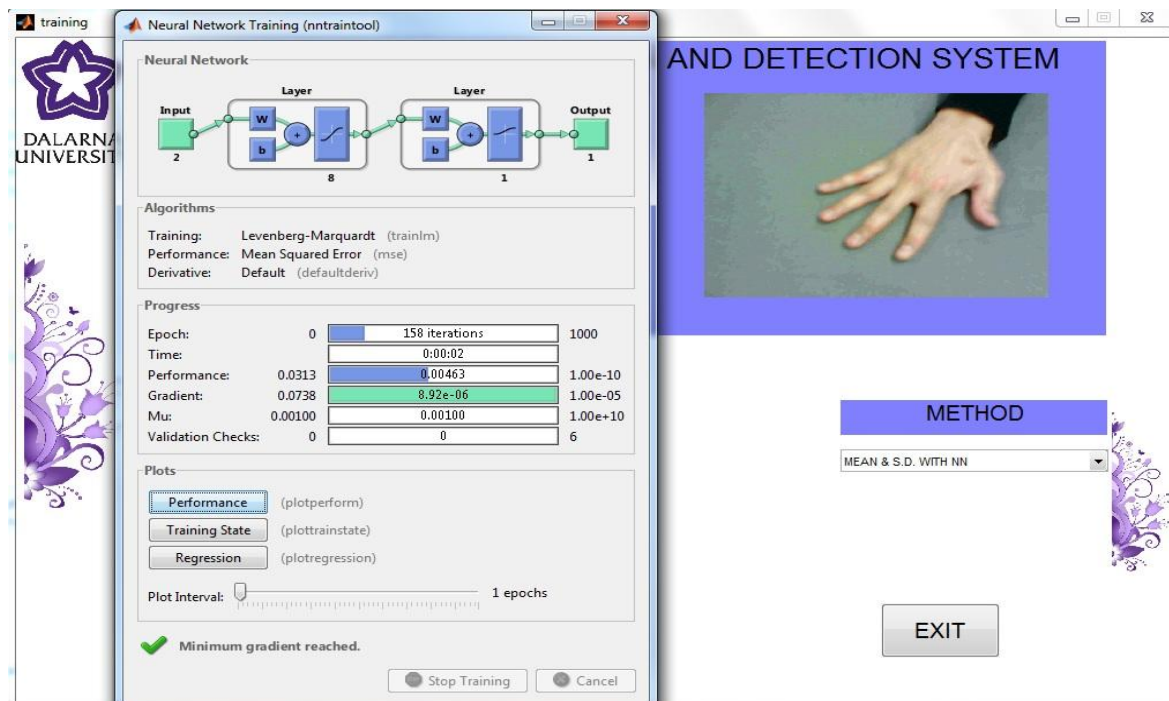


Figure 4.11: NN Training

This neural network training system will pop up when we select Row Vector algorithm, Edging and Row Vector and Mean and Standard Deviation algorithm. This is not for Diagonal Sum algorithm. Diagonal Sum classified real time.

#### Performance of NN

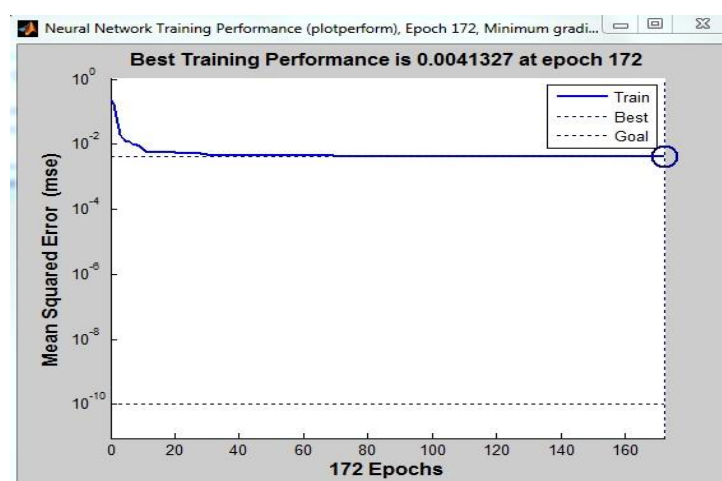
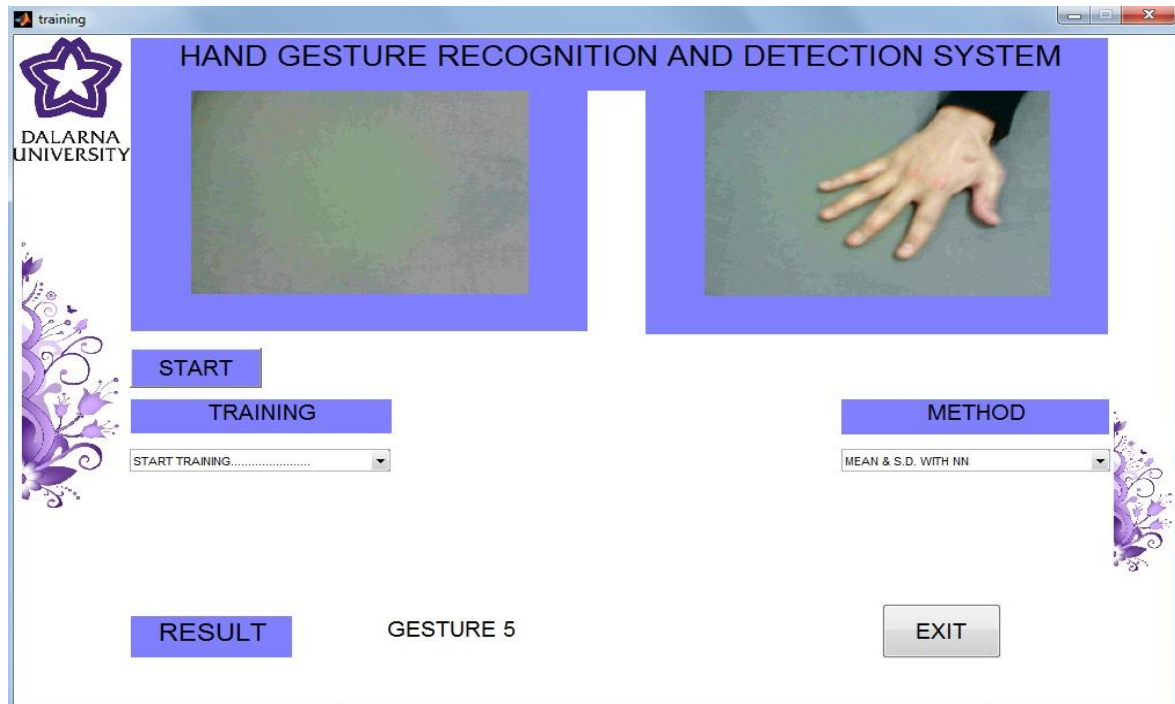


Figure 4.12: Performance Chart

#### 4.64 Detection and Recognition of a gesture



**Figure: 4.13: Graphical User Interface Output**

For Diagonal Sum algorithm we need to select 5 different gestures by selecting drop down menu under **TRAINING** for real time training and for recognition select Diagonal Sum Algorithm under **METHOD** drop down menu. There is no neural network for this algorithm but remaining 3 algorithms have neural network for train the system.



## CHAPTER 5

### RESULTS AND DISCUSSION

#### INTRODUCTION

The hand gesture recognition system has been tested with hand images under various conditions. The performance of the overall system with different algorithms is detailed in this chapter. Examples of accurate detection and cases that highlight limitations to the system are both presented, allowing an insight into the strengths and weaknesses of the designed system. Such insight into the limitations of the system is an indication of the direction and focus for future work.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. It helps us in uncovering errors that were made inadvertently as the system was designed and constructed. We began testing in the ‘small’ and progressed to the ‘large’. This means that early testing focused on algorithms with very small gesture set and we ultimately moved to a larger one with improved classification accuracy and larger gesture set.

#### 5.1 ROW VECTOR ALGORITHM

The detection rate of the system achieved through this algorithm was 39%. It was noticed that the performance of the system improved as the data set given to neural network (NN) for training was increased. For each type of gesture, 75 images were given to the system for training. At the end, the system was tested with 20 images of each kind of gesture. The results of the algorithm are given below in **Figure 5.1**.

The row vector algorithm failed to give satisfactory results because the parameter (row vector) of two different pictures happened to be the same for different gestures. This resulted in wrong classification of some of the gestures and also it takes too much time in training process so, a need was felt for improvement in the parameter passed to the neural network (NN). This resulted in the evolution of my edge and row vector-passing algorithm.

#### 5.2 EDGING AND ROW VECTOR PASSING ALGORITHM

The detection rate of the system achieved through this algorithm was 47%. It was noticed that the performance of the system improved as the data set for training was increased. For each type of gesture, 75 images were given to the system for training. At the end, the system was tested with 20 images of each kind of gesture. The results of the algorithm are given below in **Figure 5.1**.

The introduction of edge parameter along with the row vector gained an improvement in performance but the self-shadowing effect in edges deteriorated the detection accuracy and

it was again thought to improve the parameter quality passed to the neural network (NN). It also have drawback of time consuming it take more than normal time for training process. This gave birth to mean and standard deviation of edged image algorithm.

### 5.3 MEAN AND STANDARD DEVIATION OF EDGED IMAGE

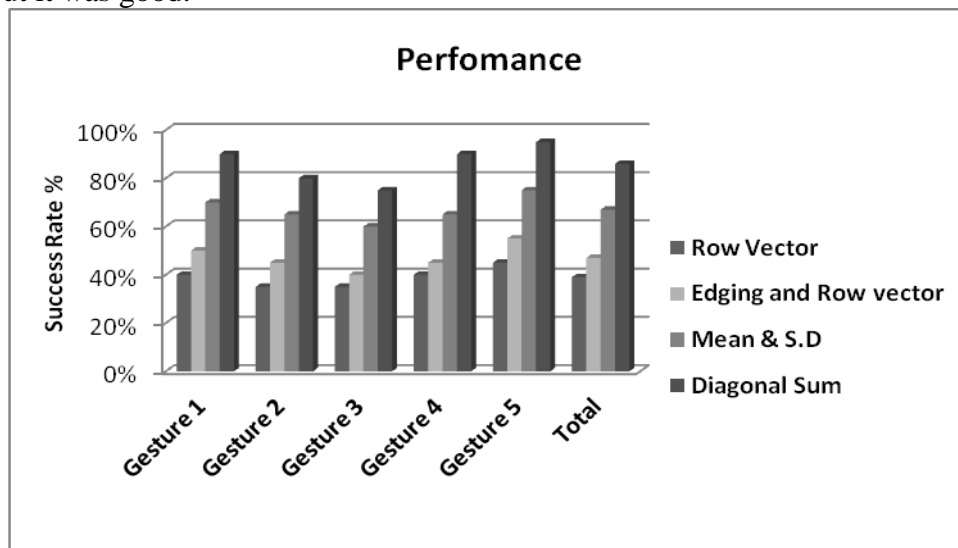
The detection rate of the system achieved through this algorithm was 67%. It was noticed that the performance of the system improved as the data set for training was increased. For each type of gesture, 75 images were given to the system for training. At the end, the system was tested with 20 images of each kind of gesture. The implementation details of the algorithm are given below in **Figure 5.1**.

The mean and standard deviation algorithm did help us in attaining an average result and also it take less time for training process but still the performance was not very good as I want. The reason was majorly the variation in skin colors and light intensity.

### 5.4 DIAGONAL SUM ALGORITHM

The poor detection rate of the above algorithms resulted in the evolution of diagonal sum algorithm, which used the sum of the diagonals to train and test the system. This is real time classification algorithm. User need to train system first and then try to recognized gesture. Every gesture at least once user should have to give system for training process. The detection rate of the system achieved through this algorithm was 86%. For each type of gesture, multiple images were given to the system for training. After every training process system were tested 20 times. At the end, the system was tested with 20 images of each kind of gesture. The results of the algorithm are given below in **Figure 5.1**.

The diagonal sum algorithm also demanded improvement as its detection accuracy was not 100% but it was good.



**Figures 5.1: Performance Percentage**



## 5.5 PROCESSING TIME

Evaluation of time in an image processing and recognition procedure is very important for result and performance, which shows tendency in all techniques which I used to recognized hand gestures. There are few factors that prejudiced on the results such as quality of image, size of image (e.g. 648x480) and the parameters of recognition techniques or algorithm. In first three algorithms e.g. Row Vector, Edging and Row vector and Mean and Standard Deviation, neural network used for training and testing but there is real time classification training and testing without NN in Diagonal Sum, So its takes less time. Therefore time counted for training phase. Given image to system for testing it's include training, testing, feature extraction and recognition of that particular image Following are comparison of processing time for all algorithms:

Algorithms	Row Vector (NN)	Edging and Row vector (NN)	Mean & S.D (NN)	Diagonal Sum
Processor Speed	Intel 2.53 GHz	Intel 2.53 GHz	Intel 2.53 GHz	Intel 2.53 GHz
Test Image Size	640x480	640x480	640x480	640x480
Time (s)	240 sec	240 sec	180 sec	60 sec

**Table 1: Processing Time (Testing)**

## 5.6 ROTATION VARIANT

Influence of rotation in same gesture at different degree is also important role in gesture recognition process. Let consider first three methods e.g. Row vector, Edging and Row Vector and Mean & S.D. There are multiple images of different people with different degrees of angle in training database so neural network have ability of learning which can classify with different variation of gesture position. Increasing training pattern gives more effective result because I observe that neural network able to generalize better as if we increase number of gestures patterns made by different people, there by better ability to extract features of specific gesture rather than feature of same gesture which made by single person. The main motivation of neural network in pattern recognition is that once network set properly trained by learning then system produce good result even existence of incomplete pattern and noise.

In real time classification e.g. Diagonal Sum rotation influence does matter but it depends on degree of rotation. It can easily classify if degree of rotation if between 10 to 15 degree which you can see below **Figure 5.2**. But if the degree of rotation is more that this then it is possible that it misclassify. Let suppose during training process if we are going to give gesture to system which originate vertically and for testing it's on vertically see below **Figure 5.2**. So it is possible that diagonal sum value will change and output also misclassify. This is because when we try to recognize gesture in real time, determining problem during training when one gesture end other begins, so this is main cause of misclassification.



**Figure 5.2: Degree of Rotation classification**



**Figure 5.3: Degree of Rotation misclassification**

## 5.7 EXPERIMENTS AND ANALYSIS

Performed experiment shows the achieved result and estimate gesture recognition system projected in chapter 4. The experiment divided into two categories to better analyze system performance and capabilities. The more general approach to work with differently user independent system developed to interact with multi users with different kind of skin colors and hands shapes. It is very important approach to attempt for independent multi-user system. The system can be used by various users.

Two main aims for this work to detect hand and recognition of hand gesture with neural network and real classification. The first aim to detect hand with different skin tones, using explicitly defined skin region. Secondly gesture recognition with neural network and real classification by different algorithms. This system designed to test the hypothesis that detection and recognition rate would increase as:

- Hand detection with different skin tones
- More training pattern are used to train neural network
- Gesture recognition

The analysis of each experiment which mentioned above is presented here one by one according to above sequence.

### 5.71 EFFECT WITH DIFFERENT SKIN TONES:

Hand detection with different ethnic group experiment designed to test hypothesis that detection rate would increase. If we are able to detect skin then ultimately hand will detect. For using explicitly defined skin region classifier makes this possible to detect almost all kind of skins with different ethnic group.

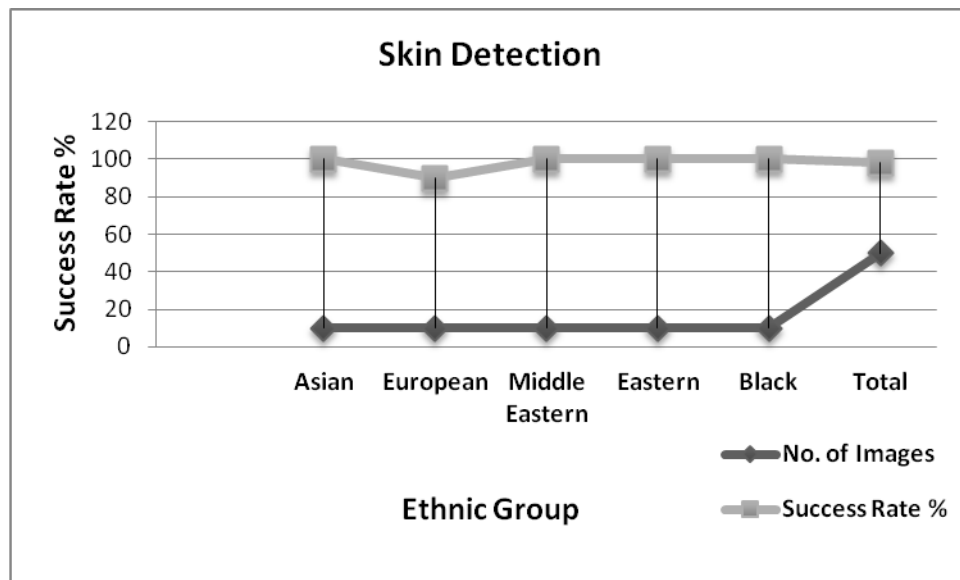


Figure 5.4: Ethnic Group Skin Detection

**Figure 5.4** is a graph showing that effect of using explicitly skin classifier with different ethnic group. 10 images of each group given to this and it gives almost 100% result. It has ability to detect almost all kind of skin tones. Using explicitly defined skin region makes possible and experiment have confirmed this with different skin tones.

Since it was observed that the detection accuracy varied with different skin colors and light intensity but using skin classifier make system efficient can have ability to detect different skin tones.

### 5.72 EFFECT OF TRAINING PATTERN:

This experimental hypothesis was to decreased misclassification as the number of training pattern increased. There are multiple images of different people with different degrees of angle in training database so neural network have ability of learning which can classify with different variation of gesture position. Increasing training pattern gives more effective results.

Initially in neural network try to train network with 30 images as 6 images of individual gesture but classification result was less and it is observed that as we increase number of

pattern of different people classification result increased which we can see **Figure 5.5** by increasing training set classification result improved.



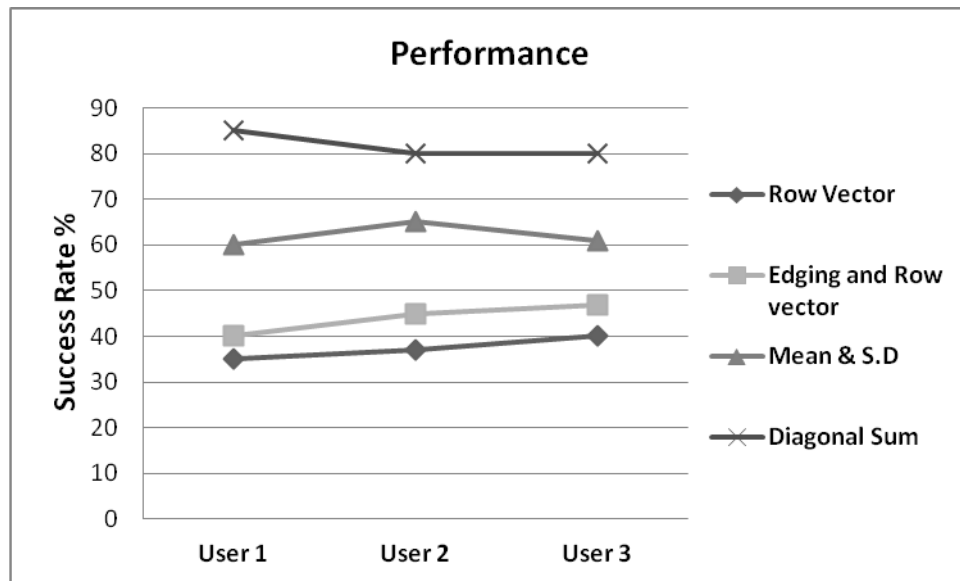
**Figure 5.5: Mix Training Pattern**

This experiment shows that neural network able to generalize better as if we increase number of gestures patterns made by different people, there by better ability to extract features of specific gesture rather than feature of same gesture which made by single person. The main motivation of neural network in pattern recognition is that once network set properly trained by learning then system produce good result even existence of incomplete pattern and noise.

### 5.73 GESTURE RECOGNITION:

This experimental hypothesis was to recognize of gestures that user gave to system either with training with neural network and real time classification. The database to test hand gesture recognition system created 75 static images with different background. The static images size 640x480 pixels collected from camera. I grabbed gestures from different people and implement different methods of recognition with more than one user ultimately system has ability to recognize gestures of different peoples.

The accuracy percentage measured per hand gesture from different users. In **Figure 5.6** prove the effectiveness of all method implemented. Classification of each gesture can be seen from following results. In following results comparison that classification percentage varies with different methods.



**Figure 5.6: Classification Percentage**

This experiment shows that system have ability to recognize hand gestures with multiple users with different skin colors. Hand gestures classification accuracy can see while classifying between five gestures. These are not enough result to express any conclusive trend for gesture recognition experiment but most important thing is that gesture recognition was possible using neural network and real time classification.

## 5.8 FAILURE ANALYSIS

### ORIENTATION/ ROTATION

There are few factors which could decrease efficiency of system specially when we talk about real time classification problem occur during training when one gesture end other begins, when user try to recognize gesture with different degree the system fail to recognize the gesture.

In different hand images the hand could have different orientation; this may affect hand recognition accuracy because when we get images from camera to testing and training distance of camera from hand and rotation of wrist makes difference, so it is essential to work with many degrees of freedom as possible in order to make recognition process realistic.

If we talk about orientation we must deal this thing in processing phase doing framing of every image by finding main axis of hand and calculate orientation and reorientation. For the rotation of wrist there are many way to deal with e.g. COG (Centre of Gravity). In processing phase we could dealt with every training and testing images by rotating clock or anti clockwise by making absolute orientation point.

## **5.9 CONVENTIONAL TEST STRATEGIES APPLIED**

### **5.91 UNIT TESTING**

The unit test focused on the internal processing logic. All statements in a module have been exercised at least once. The interface module was tested to ensure that information properly flowed into and out of the program unit under test.

### **5.92 INTEGRATION TESTING**

Integration testing is a technique for constructing the software architecture and conducting tests to uncover errors with interface. The objective of testing was to crosscheck for components fully functional or not according to design. Thus I integrated all my unit components and saw if the system worked as a whole properly or not. The information flows between the components were checked once again.

### **5.93 RECOVERY TESTING**

System fails in many ways but recovery must be properly performed. For example when a person trains the system with his gestures and the system fails to do so, it gives an error message indicating that the system was not properly trained and it keeps on doing so until it gets a valid gesture (that's acceptable to the system for differentiation). When the system is properly trained, only then we can expect that it will give us accurate results.

### **5.94 SENSITIVITY TESTING**

Invalid input classes that may cause instability or improper processing. It was found during sensitivity testing that if the system was once fully trained for all the gesture types, it gave accurate results, otherwise if it were just trained for a single or two gestures and then tested, it performs erroneous processing.

## CHAPTER 6

### CONCLUSION AND FUTURE WORK

#### 6.1 CONCLUSION

This chapter summarizes my work at every stage of the project. At the time I started my thesis, I had a brief idea of how I will bring it from a topic on the paper to a real product. Due to knowledge of Computer Vision and Biometric subjects I had background in the image-processing field but not at expert level but my constant effort helped me to go through and succeed eventually.

As required in every project, research is of utmost importance. So, I spent the pretty much time in going through the background literature. I looked at various approaches of doing my thesis and developed four different methods: Row vector algorithm, Edging and row vector passing algorithm, Mean and standard deviation of edged image and Diagonal sum algorithm.

Each of these algorithms was tried with neural networks and have higher performance rate in the ascending order respectively.

The first limitation that was discovered in all the algorithms used with neural networks was that their performance depended on the amount of training dataset provided. The system worked efficiently after being trained by a larger dataset as compared to a smaller dataset.

The Row vector algorithm used initially was a very vague approach adopted for classification as it was found through experiments that the row vectors of two different images could happen to be the same.

In the edging and row vector-passing algorithm, the edging parameter was introduced in addition to the row vector to improve the gesture classification accuracy but it was found that due to self-shadowing effect found in edges, the detection rate was not sufficiently improved.

The next parameters tried for classification were mean and standard deviation. They also failed to give satisfactory results (i.e. above 60%) but still they were among the best parameters used for detection with neural networks.

Due to the unusual behavior of neural network with all the mentioned parameters, the diagonal sum algorithm was finally implemented in real time. The system was tested with 60 pictures and the detection rate was found to be 86%. The strengths and weaknesses of gesture recognition using diagonal sum have been presented and discussed. With the

implemented system serving as an extendible foundation for future research, extensions to the current system have been proposed.

## 6.2 FUTURE WORK

The system could also be made smart to be trained for only one or two gestures rather than all and then made ready for testing. This will require only a few changes in the current interface code, which were not performed due to the shortage of time.

One time training constraint for real time system can be removed if the algorithm is made efficient to work with all skin types and light conditions which seems impossible by now altogether. Framing with COG (Centre of gravity) to control orientation factor could make this system more perfect for real application.

The system's speed for preprocessing could be improved if the code is developed in VC/VC.Net



## REFERENCES

- [1] J.Jenkinwinston (96207106036), M.Maria Gnanam (96207106056), R.Ramasamy (96207106306), Anna University of Technology, Tirunelveli: **Hand Gesture Recognitionsystem Using Haar Wavelet.**
- [2] Laura Dipietro, Angelo M. Sabatini, *Senior Member, IEEE*, and Paolo Dario, *Fellow, IEEE*, **A Survey of Glove-Based Systems and Their Applications.**
- [3] Kay M. Stanney **HANDBOOK OF VIRTUAL ENVIRONMENTS Design, Implementation, and Applications, Gesture Recognition Chapter #10** by Matthew Turk
- [4] Daniel Thalman, **Gesture Recognition Motion Capture, Motion Retargeting, and Action Recognition**
- [5] Hatice Gunes, Massimo Piccardi, Tony Ja, 2007, Research School of Information Sciences and Engineering Australian National University **Face and Body Gesture Recognition for a Vision-Based Multimodal Analyzer**
- [6] J. Heinzmann and A. Zelinsky **Robust Real - Time Face Tracking and Gesture Recognition**
- [7] Vladimir Vezhnevets, Vassili Sazonov, Alla Andreeva, Moscow State University **A Survey on Pixel-Based Skin Color Detection Techniques,**
- [8] TEY YI CHI, Universiti Teknologi Malaysi, **FUZZY SKIN DETECTION**
- [9] Robyn Ball and Philippe Tissot, Texas A&M University, **Demonstration of Artificial Neural Network in Matlab**
- [10] Howard Demuth, Mark Beale, **Neural Network Toolbox**
- [11] Ari Y. Benbasat and Joseph A. Paradiso in MIT Media Laboratory, Cambridge **An Inertial Measurement Framework for Gesture Recognition and Applications**
- [12] Peggy Gerardin, Color Imaging Course Winter Semester 2002-2003, **Color Image Segmentation**
- [13] Michael Nielsen, Moritz Störring, Thomas B. Moeslund, and Erik Granum, March 2003, **A procedure for developing intuitive and ergonomic gesture interfaces for man-machine interaction.**

[14] Oğuz ÖZÜN<sup>1</sup>, Ö. Faruk ÖZER<sup>2</sup>, C. Öncel TÜZEL<sup>1</sup>, Volkan ATALAY, A. Enis ÇETİN<sup>2</sup>, Dept. of Computer Engineering, Middle East Technical University, Ankara, Turkey, Dept. of Electrical Engineering, Bilkent University, Ankara, Turkey Faculty of Engineering, Sabanci University, Istanbul, Turkey, **Vision based single stroke character recognition for wearable computing.**

[15] Melinda M. Cerney<sup>1</sup>, Judy M. Vance, March 28, 2005, **Gesture Recognition in Virtual Environments: A Review and Framework for Future Development.**

[16] Ashutosh Saxena, Aditya Awasthi, Vaibhav Vaish, **and SANKET: Interprets your Hand Gestures.**

[17] Yoichi Sato, Institute of Industrial Science, the University of Tokyo, Makiko Saito Hideki Koike, Graduate School of Information Systems, University of Electro-Communications, Tokyo **Real-Time Input of 3D Pose and Gestures of a User's Hand and Its Applications for HCI**

[18] Mohammad Al-aqrabawi Fangfang Du, 11<sup>th</sup> March' 2000, **Human Skin Detection Using Color Segmentation**

[19] Manuel Cabido Lopes Jos'e Santos-Victor Instituto de Sistemas e Rob'otica, Institution Superior T'ecnico, Lisbon, Portugal in IROS Workshop on Robot Programming by demonstration, Las Vegas, SA, Oct 31<sup>st</sup>, 2003 **Motor Representations for Hand Gesture Recognition and Imitation**

[20] Raymond Lockton and Andrew W. Fitzgibbon, Department of Engineering Science University of Oxford. **Real-time gesture recognition using deterministic boosting**

[21] Didier Stricker, Didier Stricker, June 15th 2006 **Hand gesture recognition gradient orientation histograms and eigenvectors methods**

[22] Steven Daniel Lovell, February 2005, **A System for Real-Time Gesture Recognition and Classification of Coordinated Motion**

[23] Klimis Symeonidis, Submitted to the School of Electronic and Electrical Engineering On August 23,2000 **Hand Gesture Recognition Using Neural Networks.**

[24] William Buxton, Eugene Fiume, Ralph Hill, Alison Lee, Carson Woo in Computer Systems Research Group, Department of Computer Science, University of Toronto **Continuous hand-gesture driven input.**

[25] Ray Lockton, Balliol College, Oxford University, **Hand Gesture Recognition Using Computer Vision.**