CHAPTER 1
INTRODUCTION

1.1 HUMAN COMPUTER INTERFACE SYSTEM

Human – Computer interaction is inevitable in this era because of the enormous application of the computer systems in every field of life. The world would come to a halt without the human computer interaction. They both are necessarily, complementary to each other. From the early days the interaction between them has happened through some devices, they are call as interface devices, for instance, key board and mouse [1] [9]. Complex communication with the computer systems is possible with the emerging state of the art techniques, which has made the computer systems even more intelligent. It has been possible due to the result oriented efforts made by researchers and professionals for creating flourishing human computer interfaces [2]. Undoubtedly, the complexities of human needs have grown wildly and continues to grow so, so the complex programming ability and intuitiveness are necessary attributes of computer programmers to survive in a competitive surroundings. The computer programmers have been incredibly successful in easing the communication between computers and human. The new products, with newest of the technologies have emerged easing the human computer interaction. For instance, it has helped in facilitating tele-operating, robotics as well as better human control over complex work systems such as cars, planes and monitoring systems [48]. Initially, the computer programmers were avoiding complex programs as the focus was more on speed rather than other modifiable features. However, a move towards a user friendly environment has motivated them to revert to the center area [3].

The thought is to make computers understand the human language, human gestures, understand speech, facial expressions and develop a user friendly human computer interfaces (HCI). Gestures are the non-verbally exchanged information. A person can perform numerous gestures at a time. Since human gestures are perceived through vision, it is a subject of great interest for computer vision researchers. Our project aims to determine the different human gestures by creating an HCI. Encoding of these gestures into an objective language well understood by the computers has to be done and it demands a complex programming algorithm. An overview of our gesture recognition system has been given to gain knowledge.

1.2 GESTURES

There are innumerable gestures a human can do, with their own meaning and applications, and so it is hard to settle on a specific useful definition of gestures. Therefore, a statement can only specify a particular domain of gestures. Many researchers have tried to define gestures but their actual meaning is still arbitrary and gesture specific. Bobick and Wilson [2] have defined gestures as the motion of the body that is intended to communicate with other agents. For a successful communication, it is pertinent that both the sender and receiver must have the same set of information for a particular gesture.

All in all, a gesture is defined as an expressive movement of body parts which has a particular message which is to be communicated between a sender and a receiver. A gesture is scientifically categorized into two distinctive categories: dynamic and static [1]. A dynamic gesture is intended to change over a period of time whereas a static gesture is observed at only that particular instant of time. For instance, a waving hand means goodbye is an example of dynamic gesture and the stop sign is an example of static gesture. Thus, it is necessary to interpret all the static and dynamic gestures over a period of time so as to understand a full message [49]. This complex process is called gesture recognition. Hence we can put it in this way; the gesture recognition is the process of detecting, recognizing and then interpreting a continuous sequential gesture from the given set of data [10].

1.3 GESTURE BASED APPLICATIONS

These days Gestures are used in almost every field like medical monitoring, computer gaming, examine devices or an object etc. There are various applications based on gestures. Some of them are as follows

3D Design:

CAD (computer aided design) is an HCI which provides a platform for interpretation and manipulation of 3-Dimensional inputs which can be the gestures. Manipulating 3D inputs with a mouse is a time consuming task as the task involves a complicated process of decomposing a six degree freedom task into at least three sequential two degree tasks. Massachuchetttes institute of technology [3] has come up with the 3DRAW technology that uses a pen embedded in polhemus device to track the pen position and orientation in 3D.A 3space sensor is embedded in a flat palette, representing the plane in which the objects rest .The CAD model is moved synchronously with the users gesture movements and objects can thus be rotated and translated in order to view them from all sides as they are being created and altered.

Tele presence:

There may raise the need of manual operations in some cases such as system failure or emergency hostile conditions or inaccessible remote areas. Often it is impossible for human operators to be physically present near the machines [4]. Tele presence is that area of technical intelligence which aims to provide physical operation support that maps the operator. Arm to the robotic arm to carry out the necessary task, for instance the real time ROBOGEST system [5] constructed at University of California, San Diego presents a natural way of controlling an outdoor autonomous vehicle by use of a language of hand gestures [1]. Tele presence area has its application in various field like space, undersea mission, medicine manufacturing and in maintenance of nuclear power reactors.

Virtual reality:

Like an object can be examined by rotating them with the hand similarly it would be beneficial if a 3D object (displayed on a monitor) could be manipulating by rotating the hand in space. Virtual reality is applied to computer-simulated environments that can simulate physical presence in places in the real world, as well as in imaginary worlds. Recent virtual reality environments are primarily visual experiences, displayed either on a computer screen or through special stereoscopic displays [6]. Some methods include additional sensory details, such as sound through speakers or headphones. Hand recognition to interact with computer games would be more natural for many applications like gaming applications.

Sign Language:

American Sign Language (ASL) is a visual language. With signing, the brain processes linguistic information through the eyes. The shape, placement, and movement of the hands, as well as facial expressions and body movements, all play important parts in conveying information. ASL language is a predominant language for deaf communities in US [6]. Sign languages are the most raw and natural form of languages could be dated back to as early as the advent of the human civilization, when the first theories of sign languages appeared in history. It has started even before the emergence of spoken languages. Since then the sign language has evolved and been adopted as an integral part of our day to day communication process. Now, sign languages are being used extensively in international sign use of deaf and dumb, in the world of sports, for religious practices and also at work places [7].

A simple gesture with one hand has the same meaning all over the world and means either ’hi’ ‘goodbye’. Many people travel to foreign countries without knowing the official language of the visited country and still manage to perform communication using gestures and sign language.

Above application shows that gestures are being used almost all over the world. In a number of applications around the world gestures are used as means of communication [1]. A predefined set of gestures are used in airport to communicate with the pilots to get off and on the runway. Almost every sport people use gestures to communicate their decision.

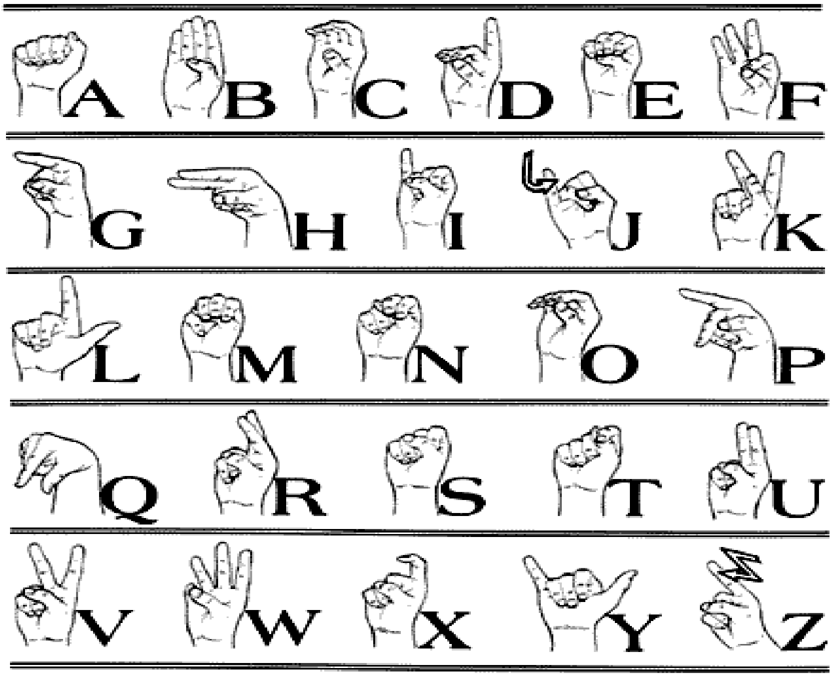


Fig (1.1) American Sign Language Symbol [49]

The recognition of gestures representing words and sentences as they do in American and Danish sign language [8] undoubtedly represents the most difficult recognition problem of those applications mentioned before. A functioning sign language recognition system could provide an opportunity for the deaf to communicate with non-signing people without the need for an interpreter. It could be used to generate speech or text making the deaf more independent. Unfortunately there has not been any system with these capabilities so far. In this project our aim is to develop a system which can classify sign language accurately.

1.4 LITERATURE SURVEY

Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body (Mitra and Acharya 2007). Gestures can be classified based on the moving body part (Fig. 1.1). There are two types of hand gestures; static and dynamic gestures. Static hand gestures (hand postures/poses) are those in which the hand position does not change during the gesturing period. Static gestures mainly rely on the shape and the flexure angles of the fingers. In dynamic hand gestures (hand gestures), the hand position is temporal and it changes continuously with respect to time. Dynamic gestures rely on the hand trajectories, scales and orientations, in addition to the shape and fingers flex angles. Dynamic gestures, which are actions composed of a sequence of static gestures, can be expressed as a temporal combination of static gestures.

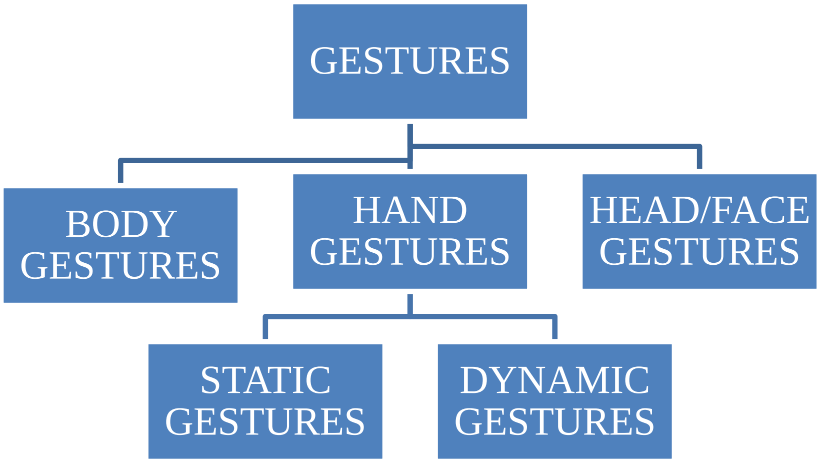


Fig (1.2) Type of Gestures [1]

Static Gestures are characterized with

- Shape
- Texture
- Skin color
- Finger flex's angle

Dynamic Gestures are characterized with

- Shape
- Texture
- Skin color
- Finger's flex angle
- Hand trajectory
- Scale
- Orientation

There exist several reviews on hand modeling, pose estimation, and hand gesture recognition (Erol et al. 2007; Ong and Ranganath 2005; Pavlovic et al. 1997; Wu and Huang 1999). The tools used for vision based hand gesture recognition can be classified into three categories (Fig. 1(b)). They are (1) Hidden Markov Model (HMM) based meth- ods (Lee and Kim 1999; Yoon et al. 2001; Ramamoor- thy et al. 2003; Just and Marcel 2009; Chen et al. 2003; Yang et al. 2007), (2) Neural network (NN) and learning based methods (Pramod Kumar et al. 2010a, 2010c, 2011; Alon et al. 2009; Su 2000; Licsar and Sziranyi 2005; Ge et al. 2008; Yang et al. 2002; Yang and Ahuja 1998; Zhao et al. 1998; Teng et al. 2005; Hasanuzzamana et al. 2007; Eng-Jon and Bowden 2004), and (3) Other methods (Graph algorithm based methods (Pramod Kumar et al. 2010b; Tri- esch and Malsburg 1996a, 1998, 2001), 3D model based methods (Athitsos and Sclaroff 2003; Ueda et al. 2003; Yin and Xie 2003; Lee and Kunii 1995), Statistical and syntactic methods (Chen et al. 2008; Wang and Tung 2008), and Eigen space based methods (Patwardhan and Roy 2007; Daniel et al. 2010)).

A systematic approach to building a hand appearance detector is presented in Kolsch and Turk (2004). The paper proposes a view specific hand posture detection algorithm based on the object recognition method proposed by Viola and Jones. A frequency analysis based method is utilized for instantaneous estimation of class separability, without the need for training. The algorithm is applied for the detection of six hand postures. Wu et al. proposed an algorithm for view independent hand posture recognition (Wu and Huang 2000). The suitability of a number of classification methods is investigated to make the algorithm view independent. The work combined supervised and unsupervised learning paradigms to propose a learning approach called Discriminant-EM (D-EM). The D-EM uses an unlabeled dataset to help supervised learning to reduce the number of labeled training samples. The image datasets utilized to test the above algorithms have simple (uniform) and relatively similar backgrounds, and these works did not address the issues with complex backgrounds (which contain clutter and other distracting objects). An algorithm for the recognition of hand postures in complex natural environments is useful for the real-world applications of interactive systems. Triesch and Malsburg (1996a, 2001) addressed the complex background problem in hand posture recognition using elastic graph matching (EGM). Bunch graph method (Triesch and Malsburg 1996a) is utilized to improve the performance in complex environments. In graph algorithms, the entire image is scanned to detect the object, which increases the computational bur- den. In addition, in a bunch graph each node is represented using a bunch of identical node features which further de- creases the processing speed. Athitsos et al. proposed an- other algorithm to estimate hand pose from cluttered images (Athitsos and Sclaroff 2003). The algorithm segmented the image using skin color, and it needs fairly accurate estimates of the center and the size of the hand. The above algorithms cannot deal with complex backgrounds which contain skin colored regions, and large variations in the hand size.

1.5 THE PROPOSED APPROACH

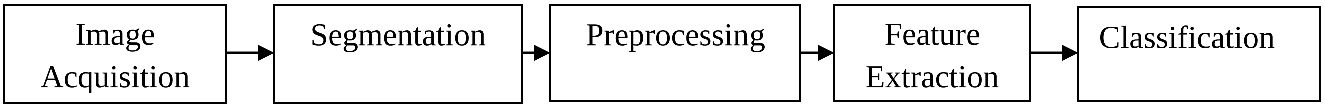


Fig (1.3) Block Diagram of hand gesture recognition system

The proposed algorithm makes use of an image based analysis that is based on the different human gestures that a person can encounter in day to day life. This thesis focuses on the detection and recognition of hand gestures in a natural environment, and it is notably the most difficult to implement in a satisfactory way. Several different approaches have been thought and tested so far. The posture of the human hand is modeled as in a three dimensional space [18] and this model is matched to the images of the hand by one or more cameras, and then the related joint angles and orientation of the hands are estimated. These parameters are then applied to the hand gesture algorithm to perform gesture classification. Secondly, we could simply capture the image of the concerned hand gesture using a camera and then extract some features which are used as input in classification algorithm for classification [19]. Thirdly, we could make use of an existing data base, apply preprocessing and extract some feature and those features are used as input in the required algorithm for classification.

In this project we have used third method for modeling the system. Similarity to skin map is utilized to detect and identify the hand region in complex background images. This information is extracted using feature based visual attention. The features utilized are based on shape, texture and color, which are extracted from the specific hand gestures and then they form the basic components which are used to recognize the other similar gestures.

To start with, the skin based features are extracted from a map which represents the similarity of pixels to human skin color, using a computational model [19]. Then, the color features are extracted by the discretization of chrominance color components in HSV and YCbCr color spaces, and the similarity to skin map. Subsequently, the hand region is extracted by the segmentation of input image using the threshold similarity to skin map. Next, the texture feature is extracted using Gabor filter. Gabor filters are convolved with images to acquire desirable hand gesture features. Segmentation and morphological filtering techniques are applied on images in preprocessing phase before using Gabor filter bank to obtain refined texture and shape feature. Now as we have gathered all the concerned features, the principal components analysis (PCA) method is then used to reduce the dimensionality of the feature space. With the reduced features, SVM is trained and exploited to perform the hand gesture recognition. Consequently, the hand postures are recognized using shape and texture based features (of the hand region), with a Support Vector Machines (SVM) classifier.

In hand gesture recognition system we have taken database from standard hand gesture database, NUS-HAND SET database [20]. The proposed algorithm is reliable against complex and skin color backgrounds as the segmentation of hand region is done using the attention mechanism, which makes an efficient use of the combination of color, shape as well as texture features. The experimental results show that the algorithm has a person independent performance. The proposed algorithm has robustness against variation in hand size and its position in the image and it is naturally the case, as we have worked upon the general dataset.

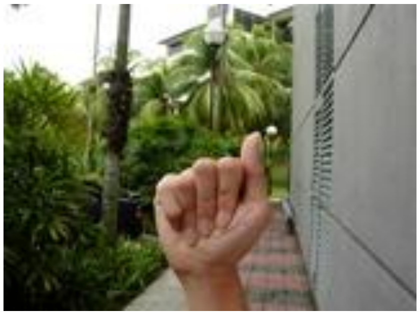
1.6 DATABASE DESCRIPTION

In this project, we have worked on the standard database of NUS hand posture dataset-II [18]. The given postures are obtained from 40 subjects with various hand sizes, and with different ethnicities. Also, the images have a variety of indoor as well as outdoor complex backgrounds. The hand postures have wide intra class variations in hand sizes and appearances. Furthermore, the database also contains a set of background images which is used to test the hand detection capability of the proposed algorithm. The recognition algorithm is tested with the new dataset using a 10 fold cross validation strategy.

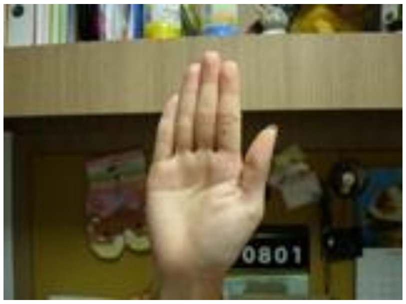
The NUS hand posture dataset consists of 11 classes of postures, 25 sample images per class, which are captured by varying the position and size of the hand within the image frame. Both grayscale and color images are available (160×120 pixels). Additionally, the hand postures are selected in such a way that the inter class variation in the appearance of the postures is less, which makes the recognition task challenging

In this project segmentation o perations are performed on gray scale image. From the database 11 gestures are used with 25 different backgrounds. The system works offline recognition i.e. We give test image as input to the system and system tells us which gesture image we have given as input. The system is purely data dependent.

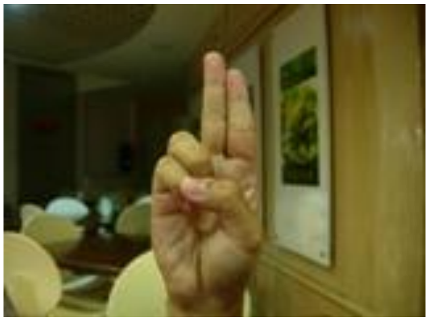
We take RGB color image and converted it into gray scale image here for ease of segmentation problem. The resolution of grabbed image is 160*120. Each of the gestures/signs is performed in front of a complex background. Each gesture is performed at various scales, translations, and a rotation in the plane parallel to the image-plane [15]. There are total 275 images, 25 images per gesture.



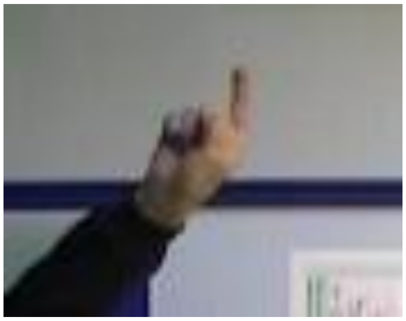
A



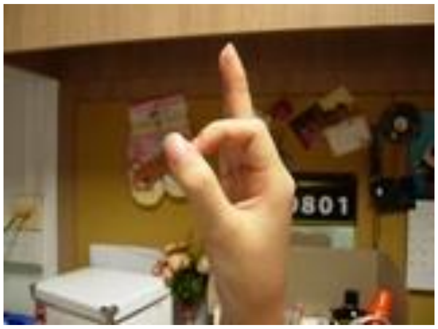
B



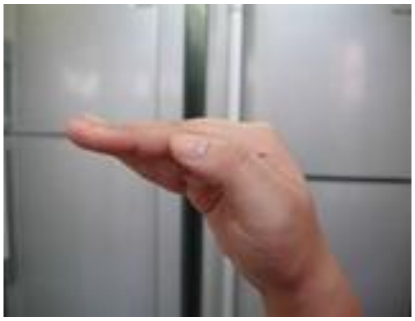
C



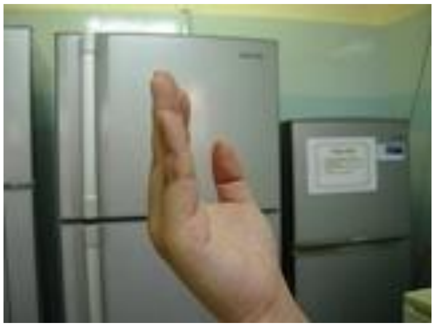
D



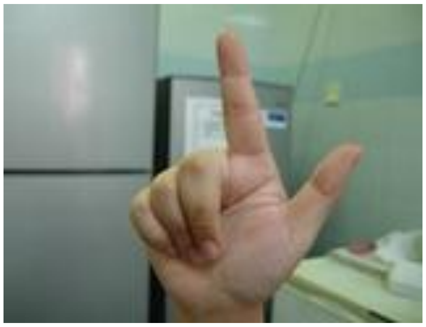
E



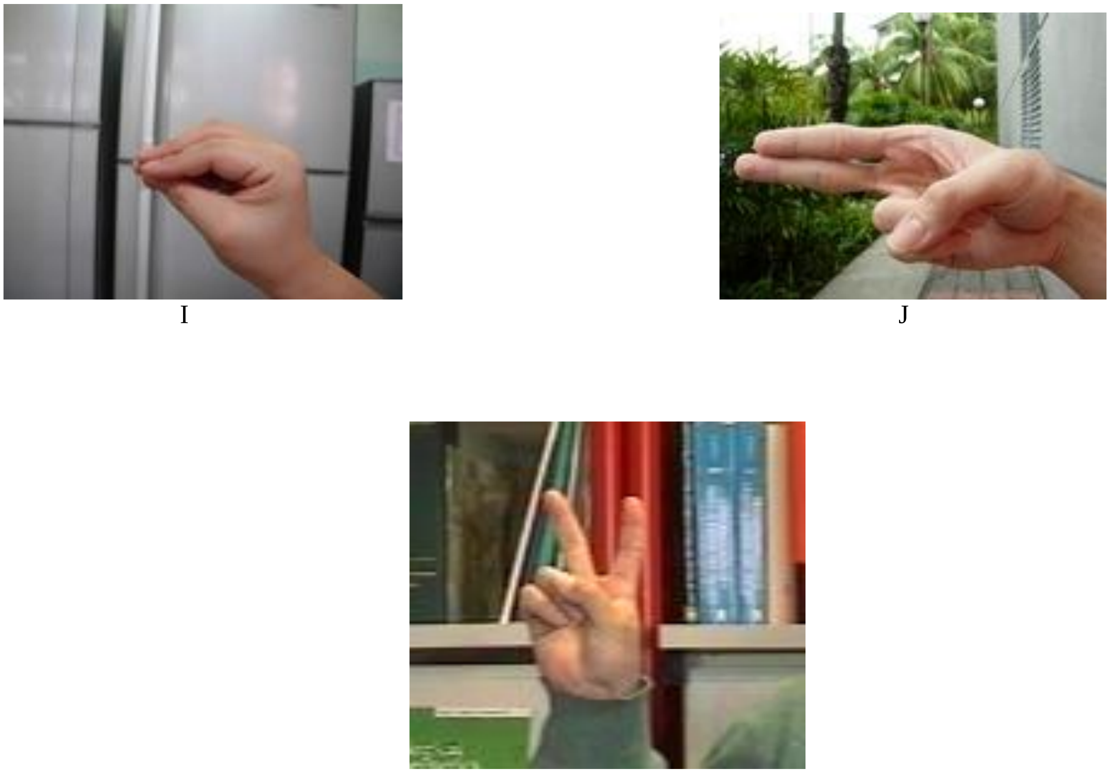
F



G



H



I

J

V

Fig 1.4 Samples of Images from database [18]

1.7 THESIS OUTLINE

In **Chapter 2** various color spaces are discussed which have been used in this project.

In **Chapter 3** preprocessing of gesture recognition system is described. Preprocessing consist image acquisition, segmentation and morphological filtering methods. We have taken our database from NUS database. RGB images are converted into binary image consisting hand or background .Morphological filtering techniques are used to remove noises from images so that we can get a smooth image.

In **Chapter 4** feature extraction methods is described .We have used Gabor filter to obtain our prime features. Canny edge detection technique is used to detect the border of hand in image.

In **Chapter 5** we explained different technique of classification of hand gesture using linear classifier, Support Vector machine.

In **Chapter 6** we concluded our work and discussed about its future scope.

CHAPTER 2
VARIOUS COLOR SPACE MODEL

2.1 INTRODUCTION

One of the most important information that could be extracted while performing recognition of a hand gesture is the skin region detection. This is because, once the skin region within the image is known, it automatically gives the information about the posture of the hand. But, to make the recognition withstand to the large variations in appearance of skin that may occur, like in shape, color, occlusion, intensity, etc, is the main and challenging task. For instance, there are several objects that may be commonly confused as skin, like wood, copper. Also, the imaging noise can appear as speckles of skin like color. On the whole, the human skin is described by a combination of red and yellow and brown and there is also a specific range of hue for skin and saturation that represent skin-like pixels [33]. For example saturation is more for a yellowish colored skin, while for a dark colored skin, it is less. Hence, the main goal of skin detection and classification is to build a decision rule that discriminate between skins and non skin pixels [48]. So the skin color pixels must be identified in a way so that most of the different types and ranges of values fall in a specific color space. Skin pixels must have high detection while the amount of non-skin pixels classified as skin should be minimized.

A number of image processing models can be applied for skin detection. Human skin color would be used to identify and differentiate the skin. Human skins have a characteristic color and so the hand gesture recognition system may be based on skin color identification. But there have been many problematic issues in the domain of skin detection, like, the model of precise skin color distribution, the choice of color space, and also the way of mechanizing color segmentation research for the detection of human skin [9]. We have in our project, focused on pixel based skin recognition that is, classifying each pixel either as skin or non-skin. We know that, each pixel is considered to be an individual unit within an image. So, pixel-base skin recognition gives a high level of accuracy at the detection phase of the process. Due to its robustness and efficiency, some color models are used extensively in skin detection. To process various color spaces and skin detection based on them, it is necessary to start with understanding of the basics of human vision and some of the commonly used terms [17].

Color Information

Whatever color we see associated with different objects is the one which is being reflected by it. Thus, human vision is a means of perceiving the outside world by reception and processing of electromagnetic radiation that is the light in the range known as the visible spectrum. Visible light wave spectrum covers the wavelengths in the range of 380 nm to 780 nm (VIBGYOR). The sensation of color comes from the different frequencies of light rays. The lesser frequency radiation is perceived as being on the red end of the color spectrum and the higher frequency are perceived as being toward the violet end.

Luminance Response

Luminance is simply describes the amount of light an area appears to be emitting, transmitting, or reflecting as perceived by the human eye.

Hue and Saturation Response

Hue describes the tint or the extent that the light being perceived corresponds to a combination of the following colors: red, green, or blue. Saturation refers to the purity of the perceived color in terms of the amount of whiteness in the color. As an example, pink and red can have the same hue but red is highly saturated while pink is much less saturated.

Now getting knowledge about the human perception of color, brightness and intensity, let us move forward with the color spaces:

2.2 RGB (Red, Green, Blue)

The primary spaces are based on the tri-chromatic theory, assuming that it is possible to match any color by mixing appropriate amounts of three primary colors. Three primary colors red(R), green (G), and blue (B) are used. Combining RGB we can obtain cyan = G+B, magenta = R+B, and yellow = R+G. We can create perceptions of a wide range of colors with different linear combinations of RGB primaries [1]. The main advantage of this color space is simplicity. However, it is not perceptually uniform. The RGB color space is conceptually a cube with one axis representing red, one representing green, and one representing blue. It does not separate luminance and chrominance, and the R, G, and B components are highly correlated. Now if we desire to pick one specific color used in an image, so for the similarity measure Euclidian distance among the colors would be determined and then checked or simply hit and tried, which is a fairly lengthy exercise.

2.3 HSV (Hue, Saturation, Value)

The perceptual spaces try to quantify the subjective human color perception by using the intensity, the hue and the saturation components [17]. It expresses hue with dominant color of an area. Saturation measures the colorfulness of an area in proportion to its brightness. And the value is related to the color luminance, i.e., the brightness of light is measured in terms of Luminance. This model discriminates luminance from chrominance. Chrominance is the color information in a signal. So chrominance information defines the color (hue and saturation), but not the brightness. This is a more responsive method for describing colors, and because the brightness is independent of the color information and therefore this is very useful model for computer vision. This separation of the luminance component from chrominance information has advantages in applications such as image processing [19]. So allowing the user to choose a range for the hue, a range for the saturation, and a range for the value gets us reasonable results. This is easier for the user, but this model gives poor result where the brightness is very low.

2.4 YCbCr (Luminance-Chrominance)

The luminance-chrominance spaces are where one component represents the luminosity and the two others the chromaticity [2] [19] [17]. The luminosity information, that is, the measure of the brightness of the light is given by Y component. The chromaticity information, that is, the color information, is given in the Cb and Cr components. Hue and saturation can be calculated by the following transformation, as given by equations (1) and (2),

$$\mathrm{S} = \sqrt{|\mathrm{Cb}|^2 + |\mathrm{Cr}|^2} \qquad (1)$$

$$\mathrm{H} = \tan^{-1} (|\mathrm{Cr}|/|\mathrm{Cb}|) \qquad (2)$$

H represents hue, which is defined as the angle of vector in YCbCr color space. S represents saturation, which is defined as the mode of Cb and Cr. This color space separate RGB into luminance and chrominance information and are useful in compression applications.

Therefore, now we have discussed about several color spaces but for skin detection those color spaces are taken into consideration, which separate the chromaticity from the luminance components of color. This is due to the fact that by employing chromaticity-dependent components of color only, some degree of robustness to illumination changes can be achieved [17]. Having selected a suitable color space, the simplest approach for detecting what constitutes skin color is to employ bounds on the coordinates of the selected space. These bounds are selected by examining the distribution of skin colors in a preselected set of images. Once the limits for the skin color is applied the skin region can easily be extracted from the original image, and thus that typical hand gesture can then be known.

The aim of this brainstorming discussion of the color spaces is to overcome the challenge of skin color detection for natural interface between human and machine. So to detect the skin color under dynamic background the study of various color models was done for pixel based skin detection. Since, the hand gesture recognition, due to challenges of vision based methods, such as varying lighting condition, complex background and skin color detection; variation in human skin color complexion required the robust development of algorithm for natural interface [16]. Nevertheless, the color is a very powerful descriptor for object detection. So for the segmentation purpose color information was used, which is invariant to rotation and geometric variation of the hand provided no occlusion is there. Human perceives characteristics of color component such as brightness, saturation and hue component than the percentage of primary color red, green, and blue. Color models are useful for to specify a particular color in standard way. Different people have different types of skin colors, however it is found that the major difference does not lie in their chrominance but to large extent it is determined by intensity. Ergo, from our discussion we can take into consideration that the simplest color models useful for intensity invariant skin detection are HSV and YCbCr. As a matter of fact, the HSV model and YCbCr model are an effective mechanism to determine human skin based on hue-saturation, and luminosity-chromaticity respectively [19].

CHAPTER 3
SEGMENTATION & PREPROCESSING

3.1 INTRODUCTION

Preprocessing is one of the most important steps in a hand gesture recognition system. A total of 275 images, with 25 signs in 11 different classes have been taken from the NUS database [16] which is a standard in gesture recognition. Preprocessing is applied to images so as to make it more objective for the machine before we can extract features from hand images. Preprocessing mainly consists of two steps:

- Segmentation
- Morphological filtering

Image segmentation is an important image processing, and its aim is to partition the image into perceptually similar regions. For example, if we desire to find if there is a chair or person inside an image, we perhaps need image segmentation to separate out objects and analyze each of the objects. It basically addresses two problems, one is the criteria for good partition and the other is the method for efficient partition. We do the segmentation of an image before the image feature extraction, image compression and image pattern recognition. Although there are several proposed methods but still none is a robust method till date.

The step we have performed for the effective image segmentation is to convert RGB image into binary image via gray scale image so that we can have only two objects in image one is hand and other is background [11]. For this, skin similarity measure is used and gray scale images are converted into binary image consisting hand or background [25]. But no image is free from noise, be it external or internal. For that reason, we apply morphological image processing operations for the removal of these kinds of noise. Morphological techniques consist mainly of four operations: dilation, erosion, opening and closing.

3.2 SEGMENTATION

Image segmentation is a conventional and transcendental problem in computer vision applications. It simply refers to partitioning of an image into several mutually exclusive subsets such that each of the subset relates to a meaningful part of that specific image. As it is one of the vital steps of the several computer vision applications, the quality of segmentation of image greatly influences the performance of the complete system. An extensive and exhaustive amount of literature on the efficient segmentation of an image has been published over the past few decades. Some of them have critically achieved extraordinary success and became popular in a wide range of applications. Thus to start with, a very good segmentation technique needs to select a adequate threshold of gray level for extract hand from the background, this means that background shouldn't have any part of hand while the reverse must also be taken care of. Usually, the selection of an appropriate and effective working segmentation algorithm depends mainly on the application field and type of images in use. Similarity to skin map is applied for image segmentation in the images related to hand gesture recognition system. Next step involves the calculation of the difference between skin color pixel and input image and then finally the threshold is applied. Meaning there by, the pixels which have differences greater than the threshold value are discarded.

Initially we collected thirteen different skin color patterns from the database. Skin color patterns include Indian, German, Turkish, Chinese, American etc skin colors. All these thirteen samples are in RGB form so convert these samples into YCbCr form to get chrominance values of each of the sample set. Make group of the values of Cb and Cr from the converted YCbCr images as shown in equations (3) and (4).

Fig (3.1) Sample of skin from different countries

$$Cr = [cr1\ cr2\ cr3\ cr4\ cr5\ cr6\ cr7\ cr8\ cr9\ cr10\ cr11\ cr12\ cr13] \tag{3}$$

$$\text{Cb} = [\text{cb1 cb2 cb3 cb4 cb5 cb6 cb7 cb8 cb9 cb10 cb11 cr12 cb13}] \qquad (4)$$

Determine the mean of each group and called it as rmean and bmean respectively. This will give Skin Similarity Map. Now convert the test image into YCbCr form and compute the size of converted image.

Fig (3.2) YCbCr color space image

Let the size of the converted image matrix is M×N.

Calculate the difference matrix. Difference matrix can be calculated by calculating difference between the chrominance value and the calculated mean value from the sample set. Apply this for each of the pixel in the test input image. Using this difference matrix calculate Skin likelihood image by using following formula:

$$\text{Likely\_skin} = \frac{1}{2\pi} e^{-\left(0.5 \text{ x } (\text{rbcov })^{-1} \text{x}^{\text{T}}\right)} \qquad (5)$$

where,

$$\text{rbcov(cb,cr)} = \text{E[(Cb-bmean)*(Cr-rmean)]} \qquad (6)$$

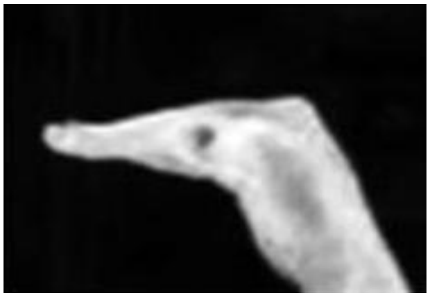


Fig (3.3) Skin Likelihood Image

Next step is to apply thresholding to the skin likelihood image. OTSU method is used to segment out the hand from the image, and thus the threshold value is calculated from OTSU technique.

OTSU's method involves the search for the threshold that gives the minimum intra-class variance, which is defined as the weighted sum of variances of the two classes:

$$\sigma_{\omega}^{2}(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \qquad (7)$$

Weights ω_1 and ω_2 are the probabilities of the two classes separated by a threshold t and σ_1^2 and σ_2^2 are the variances of these classes. OTSU shows that minimizing the intra-class variance is the same as maximizing inter-class variance.

$$\sigma_b^2(t) = \sigma^2 - \sigma_{\omega}^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \qquad (8)$$

Which is expressed in terms of class probabilities ω_i and class meansμ_i The class probability $\omega_1(t)$ is computed from the histogram as t :

$$\omega_1(t) = \textstyle\sum_0^t p(i) \qquad (9)$$

While the class mean $\mu_1(t)$ is:

$$\mu_1(t) = \textstyle\sum_0^t p(i)x(i) \qquad (10)$$

Where x(i) is the value at the center of the i^{th} histogram bin. Similarly, you can compute $w_1(t)$ and $\mu_i(t)$ on the right-hand side of the histogram for bins greater than t. The class probabilities and class means can be computed iteratively. This idea yields an effective algorithm.

Algorithm

1. Compute histogram and probabilities of each intensity level
2. Set up initial $\omega_i(0)$ and $\mu_i(0)$
3. Step through all possible thresholds t = 1 maximum intensity
    1. Update ω_i and μ_i
    2. Compute $\sigma_b^2(t)$
4. Desired threshold corresponds to the maximum $\sigma_b^2(t)$
5. You can compute two maxima (and two corresponding thresholds). $\sigma_{b1}^2(t)$ is the greater max and $\sigma_{b2}^2(t)$ is the greater or equal maximum.

$$\text{Desired threshold} = (\text{threshold1}+\text{threshold2})/2 \qquad (11)$$

Fig (3.4) Segmented Image

3.3 MORPHOLOGICAL FILTERING

If we take a closer look to the segmented image after applying the OTSU algorithm on the original RGB image we find that the segmentation is not perfectly done. The background can have some background noise which has an intensity of '1' in a binary image and the hand gesture can have gesture noise which has an intensity of '0' in a binary image [9]. We need to remove these errors as they create problem in proper hand gesture recognition. A morphological filtering [4] approach has been applied using sequence of dilation and erosion to obtain a smooth, closed, and complete contour of a gesture. Morphology deals with the shape or structure of an object. Morphological techniques probe an image with a small shape or template called a structuring element. Morphology is a tool for extracting image components that are useful for representation and description of region shape, boundary, skeleton, convex hull etc [11]. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighborhood of pixels. Morphological operation returns an image in which the pixel has a non-zero value only if the test is successful at that location in the input image.

The structuring element is a small binary image, with a small matrix of pixels, each with a value of zero or one. The dimension of the matrix specifies the size of the structuring element. However the shape of the structuring element is specified by the pattern of ones and zeros. An origin of the structuring element is usually one of its pixels, although generally the origin can be outside the structuring element.

1	1	1
1	1	1
1	1	1

ORIGIN

Table (3.1) 3X3 Square Structuring Element

In the morphological dilation and erosion, the structuring element is moved over the actual image and the morphological computations are performed. The value of the pixels in an output image is obtained by following a set of rules on the neighbors in the input image [21]. The dilation and erosion operation on a binary image A and with a structuring element B defined as follow [22].

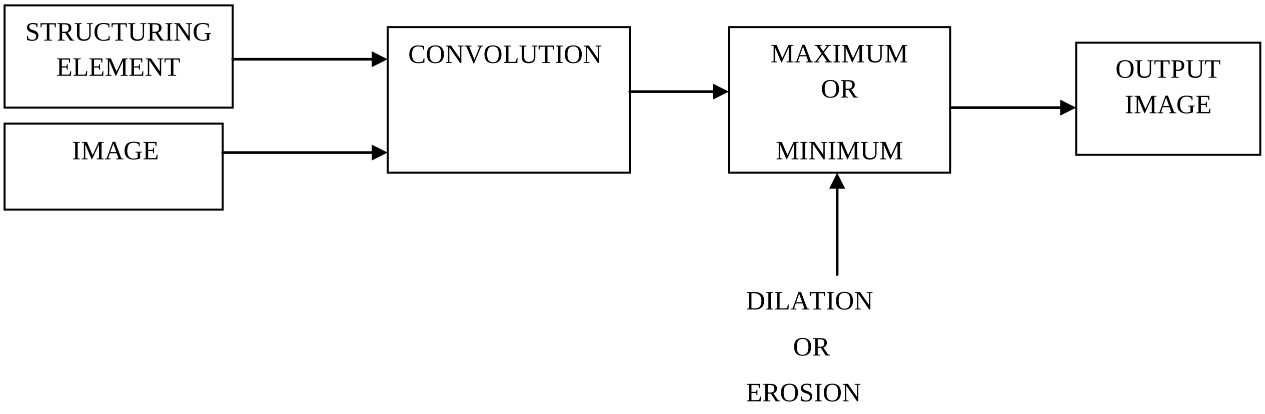


Fig (3.5) Erosion and Dilation Block Diagram

- Dilation:

The dilation of an image by a structuring element has the following three effects on the image:

- Filling of gaps
- Removal of noise
- Expansion of the object boundary

The mathematics of the dilation process can be easily explained as follows:

If A and B are sets in the 2-D integer space Z^{2}: $x = (x^{1},x^{2})$ and Ø is the empty set, then dilation of A by B is defined by:

$$A\oplus B = \{z \mid (B\hat{}) z \cap A \neq \emptyset\} \qquad (12)$$

Where B ^ is the reflection of B. In dilation process first we obtain the reflection of B about its origin and then we shift the reflection by x [23].The condition of dilation of A by B is set of all x The condition is such that for dilation [24] of A by B is set off all x displacement such that B ˆ and an overlap at least one nonzero element. Set B is commonly referred to as the structuring element [24]. The value of the output pixel is the maximum value of all the pixels in the input pixel's neighborhoods. In any of the pixels is set to the value 1, the output pixel is set to 1. We consider each of the background pixels in the input image, to compute the dilation of a binary input image by the structuring element. For each input or background pixel we superimpose the structuring element with the input image so that the origin of the structuring element coincides with the input pixel position. Now if one of the pixels in the structuring element coincides with a foreground pixel of the image, foreground value is set to the input pixel. However, if all the pixels in the image are background, then only the background value is set to the input pixels.

For our example 3×3 structuring element, the effect of this operation is to set to the foreground color any background pixels that have a neighboring foreground pixel. Such pixels must lie at the edges of white regions, and so the practical upshot is that foreground regions grow (and holes inside a region shrink). Dilation is the dual of erosion i.e. dilating foreground pixels is equivalent to eroding the background pixels.

- Erosion:

The erosion of an image by a structuring element has the following three effects on the image:

- Removal of external noise
- Boundary pixels get eliminated

The mathematics of the erosion process can be easily explained as follows:

The erosion of A by B is

$$A\Theta B = \{x|(B)x \subseteq A\} \qquad (13)$$

The erosion of A by B is the set of all point x such that B, translated by A, is contained in A [23]. Thus the value of the output pixel is minimum value of all the pixels in the input pixel's neighborhood. In binary image, if any of the pixels is set to 0, the output pixel is set to 0 [27].

- Opening:

The opening of A by B is obtained by the erosion of A by B, followed by dilation of the resulting image by B.

$$A \circ B = (A \otimes B) \oplus B \qquad (14)$$

In the opening operation the external noise is eliminated [28]. The opening of A by B is simply the erosion of A by B followed by dilation of the result by B.

- Closing:

The closing of set A by structuring element B is

$$A \bullet B = (A \oplus B) \otimes B \qquad (15)$$

In the opening operation the internal noise is eliminated. Closing also tends to smooth section of contours but [26],it generally fuses narrow breaks and long thin gulfs, eliminates small holes and fills gaps in the contour [26].

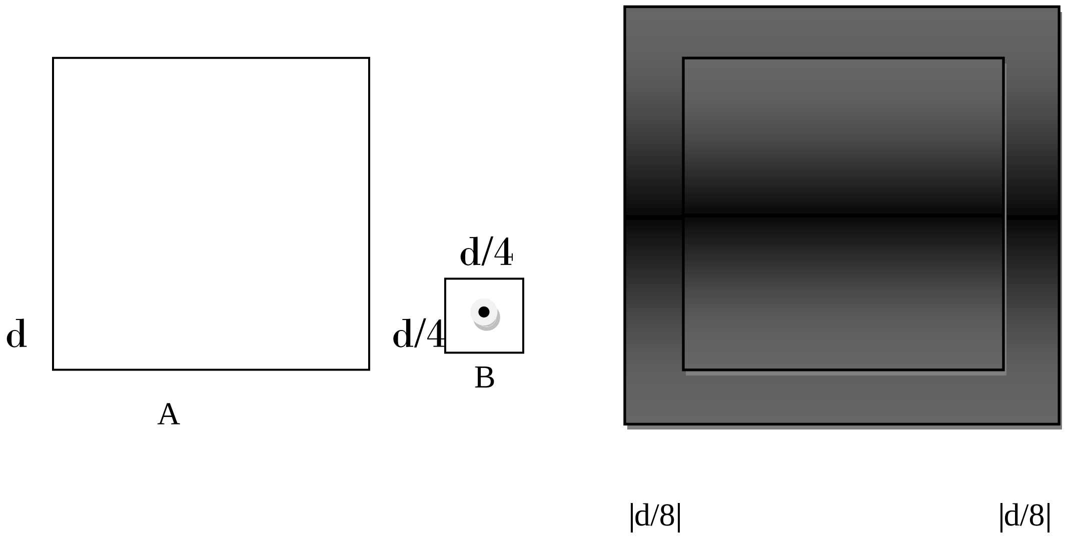


Fig (3.6) Dilation Process [26]

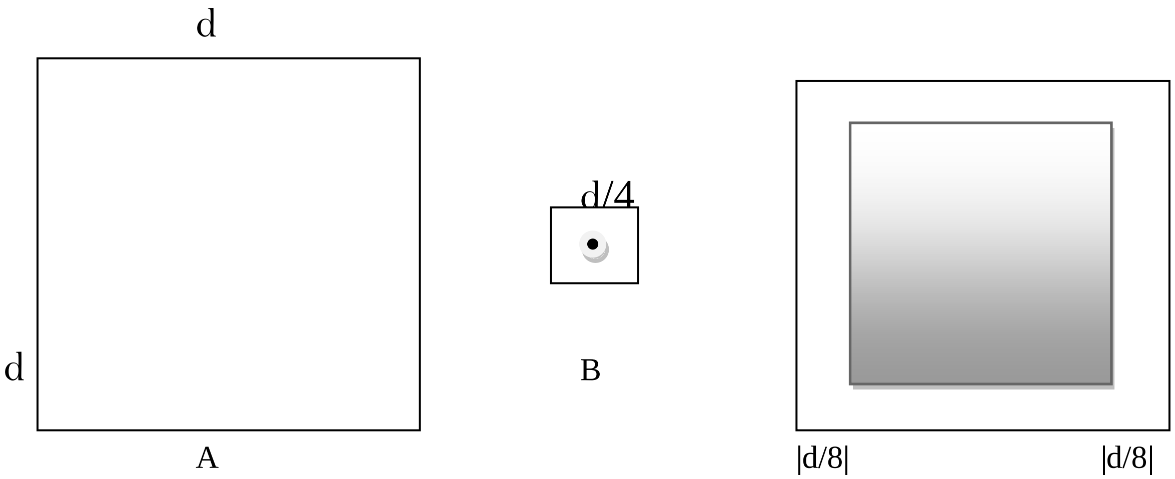


Fig (3.7) Erosion Process [26]

3.4 CONCLUSION

In this chapter preprocessing of related images is described. Preprocessing involves firstly image acquisition, then segmentation of image and then finally the morphological filtering. The RGB image is converted into binary image consisting hand or background and the OTSU algorithm [22] is used for segmentation purpose. Later, noise is removed using Morphological techniques.

CHAPTER 4
FEATURE EXTRACTION

4.1 INTRODUCTION

In this chapter we will discuss the feature extraction for the purpose of gesture recognition. Feature extraction is very important in terms of giving input to a classifier .Our prime features are color, shape and texture feature .In feature extraction first we have to find edge of the segmented and morphological filtered image . Canny edge detector algorithm is used to find the edge which leads us to get boundary of hand in image.

4.2 CANNY EDGE DETECTOR

Edge contains some of the most valuable information in an image. There are definitely several applications where this information can be used. For instance, the edges can be used to measure the size of an object, or to isolate a specific object from the background, or to recognize and classify objects. In image processing finding edges is fundamental problem because edge defines the boundaries of the objects in an image. An edge can be defined as sudden or strong change in the intensity or a local discontinuity in the pixel values that exceeds a given threshold value. Thus, by finding the edge in any image we are reducing amount of data but we are preserving the shape as an edge is simply the observable difference in pixel values. The Canny edge detection algorithm is also known as the optimal edge detector. Canny [28], enhanced the edge detection by following a list of criteria. Firstly, the error rate, which means that the edges occurring in images should not be missed and that there must be no responses to non-edges [29]. Secondly, the edge points must be well localized. Substantially, the distance between the actual edge and the edge pixels as found by the detector is to be at a minimum. Thirdly, the response of the edge detector must be unique for a single edge [29]. This is necessary because the two criteria were not enough to completely eliminate the possibility of multiple responses to an edge. Based on these three criteria, the canny edge detector firstly eliminates the noise and smoothen the image. Next, it finds the image gradient [31] to show the regions with high spatial derivatives [32]. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum. Hysteresis is used to track along the remaining pixels that have not been suppressed to further reduce the gradient array. Hysteresis has a very important significance as it uses two values of thresholds. Suppose the magnitude is lesser than the first threshold, and then it is set to zero. That means, it is made a non edge. However if the magnitude is above the second threshold which is the higher of the two thresholds, then it is made an edge. But if the magnitude lies between the two threshold values, then if there is a path from this pixel to a pixel with a gradient above second threshold, it is set to zero [31]. Thus this process can be put it in the following steps:

Step 1: Before trying to locate and detect any edges, noise must be filtered out from the input image. Gaussian filter is used extensively in image processing for smoothening of the images, and also it can be computed using a simple mask [3]. Hence, Gaussian smoothing is used in canny edge detection as a sub operation and this can be performed using standard convolution method [4]. The mask through which convolution of image is to be done is typically smaller than the actual image [3]. Consequently, operation on pixels at a time is done when the mask is swept over the image. The sensitivity of the detector for noise depends upon the size of the Gaussian window, larger the Gaussian mask, lower is the sensitivity of detector towards noise. While with the increase in size of the Gaussian mask, the localization error also increases. An example of a 5*5 Gaussian filter is given below [31]:

1/273	1	4	7	4	1
	4	16	26	16	4
	7	26	41	26	7
	4	16	26	16	4
	1	4	7	4	1

Table (4.1) 5 * 5 Gaussian filter Example

Step 2: Gradient of the smoothened and noise free image n, the next step is to find the edge strength by taking the gradient of the image. The Sobel operator performs a 2-D spatial gradient measurement on an image [32]. Then, the approximate absolute gradient magnitude (edge strength) at each point can be found. The Sobel operator uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). They are shown below:

-1	0	+1
-2	0	+2
1	0	+1

Gx

+1	0	+1
+1	0	+1
+2	0	+2

Gy

Table 4.2 Gradient example

$$\mathrm{G} = \sqrt{\mathrm{Gx}^2 + \mathrm{Gy}^2} \qquad (16)$$

From this the edge gradient and the direction can be determined [31]

$$\Theta = \arctan\left(\mathrm{Gy}^2/\mathrm{Gx}^2\right) \qquad (17)$$

Step3: Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image [31]. So if the pixels of a 5x5 image are aligned as follows:

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Table 4.3 Image segment (5*5)

Then, it can be seen by looking at pixel whose value is "1", there are only four possible directions when describing the surrounding pixels - 0 degrees (in the horizontal direction), 45 degrees (along the positive diagonal), 90 degrees (in the vertical direction), or 135 degrees (along the negative diagonal). So now the edge orientation has to be resolved into one of these four directions.[31]

step4: After the edge directions are known, non-maximum suppression now has to be applied. Non maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image [31].

4.3 GABOR FILTER BANK

After edge detection we get a boundary of hand in image. Wavelet transform can extract spatial and frequency information from a given signal. As compare other wavelet transforms, the Gabor wavelet has impressive properties in terms of mathematics and biology. Among various wavelet Gabor also provides optimal resolution in time as well as in frequency domain. Gabor wavelet models are very well known in simulations of the receptive profile of visual cortical cells [12]. The simple cells of the visual cortex of mammalian brains can be best model as self- similar 2D Gabor wavelets [48]. Gabor filters can derive multi- orientation information from a hand gesture images at different scales. The derived information it is local in nature. Basic approach while constructing a Gabor filter for hand gesture or in any other application such as face recognition or emotion detection is to construct a filter bank with different scales and orientations. Hence Gabor filters are used for features detection at various angles and scales [13-17]. The main Principle of Gabor filters is that they can capture visual properties, such as spatial locality, orientation selectivity, and spatial frequency characteristics [38-39]. Due to these characteristics many applications choose Gabor filter for feature representation. In 2D Gabor filter is represented as multiplication of a 2D Gaussian and a complex sinusoidal function sometimes also called as a complex exponential function. The general expression is: The Gabor filter is represented as [43]:

$$F(x, y) = \exp\left(- \frac{(x_0^2 + \gamma^2 y^2))}{2\,\sigma^2}\right) \times \cos(2\pi x_0 / \lambda) \qquad (18)$$

$$x_0 = x\cos\Theta + y\sin\Theta \qquad (19)$$

$$y_0 = -x\sin\Theta + y\cos\Theta \qquad (20)$$

λ shows the wavelength of the filter. The standard deviation σ of the Gaussian factor determines the size of the receptive field. Spatial frequency bandwidth can be calculated as σ/ λ, which determines the numbers of visible parallel stripes zone. In the proposed method this ratio has been fixed to σ/ λ=0.50. Ψ- Phase offset to get the symmetry of the kernel in terms of origin. γ- Aspect ratio this gives the ellipticity of the receptive field. Here, in this proposed method, local Gabor filter bank with 3 different scale and 5 different orientations σ = {1, 2, 3} and θ= {0^o, 45^o, 90^o, 135^o} is being used. After the calculation of Gabor function will be displayed on the output window.

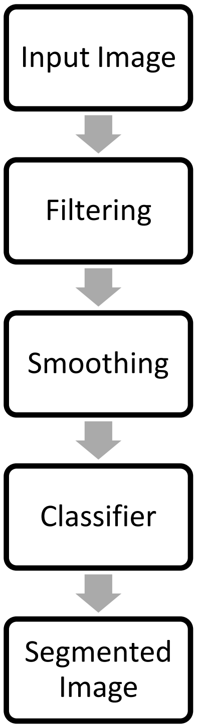


Fig (4.1) Flow Chart for the Texture Analysis through Gabor Filter

These steps can be easily understood as:

Firstly the input image is Gabor filtered which creates differentiable intensity profile. Then it is smoothened by a Gaussian filter which essentially brings the intensities in the discrete levels. Now as the intensity can be conspicuously differentiated. This is fed into a classifier, which classifies the different textures and hence a texture based segmented image is generated.

Basically Gabor filter bank with 5 different scales and 8 different orientations is being used where input image is convolved with these 40 Gabor filters (5 scales and 8 orientations). But here the image is being convolved with 12 filters to reduce the complexity. As per the above discussion the higher the value of σ makes the image blur so here only three scale values are being selected to reduce the complexity and also to get better results [36,37]. The responses on the orientation [0, pi] are complex conjugate of the responses on orientation [pi, 2pi] so by selecting orientation [0, pi] the computation is being reduced to half. The figure 4.2 shows the kernel response and figure 4.3 shows the magnitude of the applied Gabor filter.

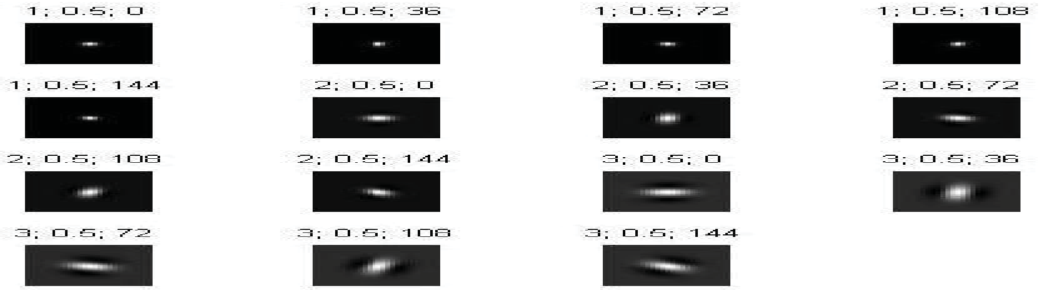


Fig (4.2) Kernel response with different scales and orientations

Figure (4.3) Gabor filter response of a typical hand gesture

Some properties of Gabor filters:

- A tunable band pass filter
- Similar to a STFT or windowed Fourier transforms
- Satisfies the lower-most bound of the time-spectrum resolution (uncertainty principle)
- It’s a multi-scale, multi-resolution filter
- Has selectivity for orientation, spectral bandwidth and spatial extent.
- Has response similar to that of the Human visual cortex (first few layers of brain cells)
- Used in many applications – texture segmentation; iris, face and fingerprint recognition.

- Computational cost often high, due to the necessity of using a large bank of filters in most applications.

As in case of gesture recognition while using Gabor filter for the feature extraction its output vector has high dimensionality. If the input image is of 160x120 pixels after convolution with Gabor filter bank of 3 scales and 4 orientations the dimensionality will be 12x 160x 120=230400. The 2D Gabor filtered images is converted into a pattern vector this process is repeated for all other 12 responses. In the last after getting all the pattern vectors for 12 responses concatenate all of them either in rows or columns.

4.4 RESULTS

Fig (4.4) Input to the Gabor filter bank

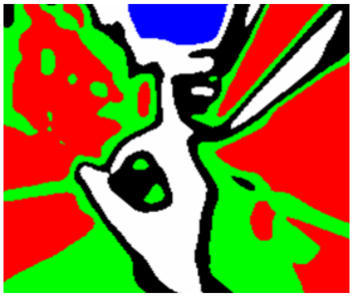


Fig (4.5) Gabor Filter Output

4.5 PCA INTRODUCTION

The PCA is a procedure of finding the so called principal components of observed data presented by a large random vector, i.e. Of components of a smaller vector which preserves principal features of observed data. In particular, this means that the original vector can be reconstructed from the smaller one with the least possible error.

As many recognition systems use PCA (Principal components analysis) [34, 35]. PCA is an effective method which can reduce the dimensionality of the data and can effectively extract the required information of the image. It provides data which has no redundancy as Gabor filter wavelet are not orthogonal wavelets [16]. In case of image processing the complexity of grouping the images can be reduced. As among other dimension reduction methods PCA is the fastest algorithm [35] so it will reduce the time complexity. PCA outperform when

Dataset is small so in this paper by using local Gabor filter bank it enhance the computation efficiency of the PCA.

4.6 THE PCA THEORY

Principal component analysis in signal processing can be described as a transform of a given set of n input vectors (variables) with the same length K formed in the n-dimensional vector $x = [x_1, x_2, ...x_n]^T$ into a vector y according to

$$y = A\ (x - m_X) \qquad (21)$$

This point of view enables to form a simple formula but it is necessary to keep in the mind that each row of the vector x consists of K values belonging to one input. The vector m_X is the vector of mean values of all input variables defined by relation [41],

$$m_x = \frac{1}{K} \sum_{k=1}^{K} x_k \qquad (22)$$

Matrix A is determined by the covariance matrix C_X. Rows in the A matrix are formed from the eigenvector of C_X ordered according to corresponding eigen values in descending order. The evaluation of the C_X matrix is possible according to relation [41],

$$C_x = E\{(x - m_x)(x - m_x)^T\} = 1/K\ \sum (x_k\, x_k^T - m_x\, m_x^T) \qquad (23)$$

As the vector x of input variables is n-dimensional it is obvious that the size of C_X is n x n. The elements $C_X(i, i)$ lying in its main diagonal are the variances, of x and the other values $C_X(i, j)$ determine the covariance between input variables x_i, x_j.

$$C_X\ (i, i) = E\{(x_i - m_i)^2\} \qquad (24)$$

$$C_X(i, j) = E\{(x_i - m_i)(x_j - m_j)\} \qquad (25)$$

Between input variables x_i, x_j. The rows of A are ortho normal so the inversion of PCA is possible according to relation

$$x = A^T y + m_X \qquad (26)$$

The kernel of PCA has some other interesting properties resulting from the matrix theory which can be used in the signal and image processing to fulfill various goals as mentioned below.

4.7 PCA FOR IMAGE COMPRESSION

Data volume reduction is a common task in image processing. There is a huge amount of algorithms [3], [4], [5] based on various principles leading to the image compression. Algorithms based on the image color reduction are mostly lossy but their results are still acceptable for some applications. The image transformation from color to the gray-level (intensity) image I belongs to the most common algorithms. Its implementation is usually based on the weighted sum of three color components R, G, B according to relation,

$$I = w_1R + w_2G + w_3B \qquad (27)$$

The R, G and B matrices contain image color components, the weights w_i were deter- mined with regards to the possibilities of human perception [2]. The matrix A is replaced by matrix A_l in which only l largest (instead of n) eigen values are used for its forming. The vector $\hat{x}$ of reconstructed variables is then given by relation,

$$\hat{x} = A^T y + m_x \qquad (28)$$

True-color images of size M x N are usually saved in the three-dimensional matrix P with size M x N x 3 which means that the information about intensity of color components is stored in the 3 given planes. The vector of input variable x can be formed as the n=3-dimensional vector of each color. Forming three 1-dimensional vectors $x_{1,2,3}$ from each plane P(M, N,i) with the length of M.N can be advantageous for better understanding and programming. The covariance matrix C_x and corresponding matrix A are then evaluated and the 3-dimensional reconstructed vector $\hat{x}$ can be called as the first, the second and the third component of the given image. The image obtained by reconstruction with matrix A_1 contains the majority of information so this image should have the maximum contrast.

CHAPTER 5

CLASSIFICATION

5. 1 SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a classification and regression prediction tool that uses machine learning theory to maximize predictive accuracy while automatically avoiding over-fit to the data. Machine learning is basically a computer program that can learn from experience with respect to some class of the tasks and performance measure.

Support Vector machines can be defined as systems which use hypothesis space of a linear discriminant functions in a high dimensional feature space, trained with a learning algorithm from optimization theory that each machine learning algorithms aimed to learn representations of simple functions. Hence, the goal of learning is to output a hypothesis that performed the correct classification of the training data and all the machine learning algorithms are designed to find such an accurate fit to the data. The ability of a hypothesis to correctly classify data not in the training set is known as its generalization. SVM implements a learning bias derived from statistical learning theory [40].

Consider a simple two class problem. A machine has to learn the boundary between the two classes. So that given a new sample it should indicate if it belongs to class 1 or class 2. There are many linear classifiers (hyper planes) that separate the data. Figure (5.1) shows the three of possible hyper planes. While figure (5.2) clearly explains that there can be infinite number of hyper-planes that can distinctively separate the two classes. However, only one of these achieves maximum separation between the samples of the two distinct classes.

5.2 NEURAL NETWORKS AS A SOLUTION

Most neural networks are designed to find a separating hyper plane This is not necessarily optimal In fact many neural networks start with a random line and move it, until all training points are on the right side of the line This inevitably leaves training points very close to the line in a non-optimal way. There are however now approaches in which a large margin classifier, i e a line approaching the optimal is sought [44].

5.3 SUPPORT VECTOR MACHINES AS A SOLUTION

Support Vector Machines use geometric properties to exactly calculate the optimal separating hyper-plane directly from the training data. They also introduce methods to deal with non-linearly separable cases, i.e. where no straight line can be found, and cases in which there is noise in the training data, then some of the training examples are wrong.

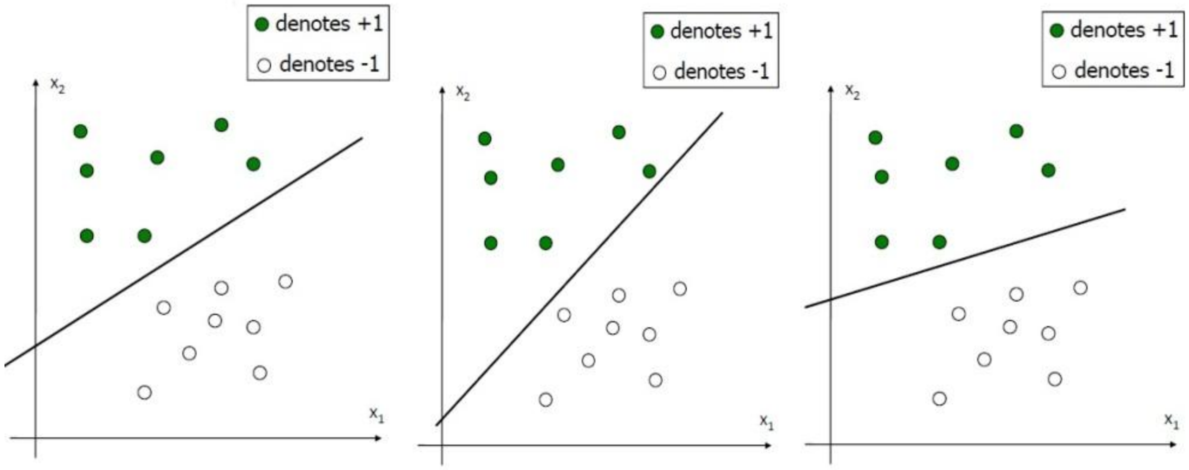


Figure (5.1): Different Hyper planes separating the classes [45]

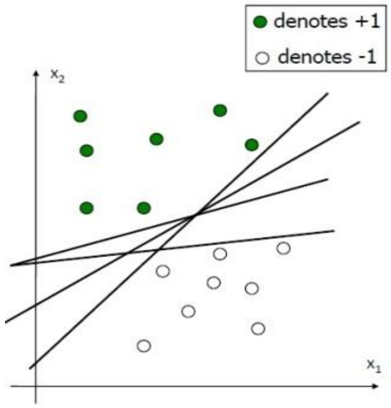


Figure (5.2): Infinite number of hyper planes for separation of classes

Thus there can be several such hyper-planes that may separate the two classes prominently. But not all of them are good hyper-planes as some points are very near the hyper-plane and points near them might be classified differently although intuitively they should be the same class.

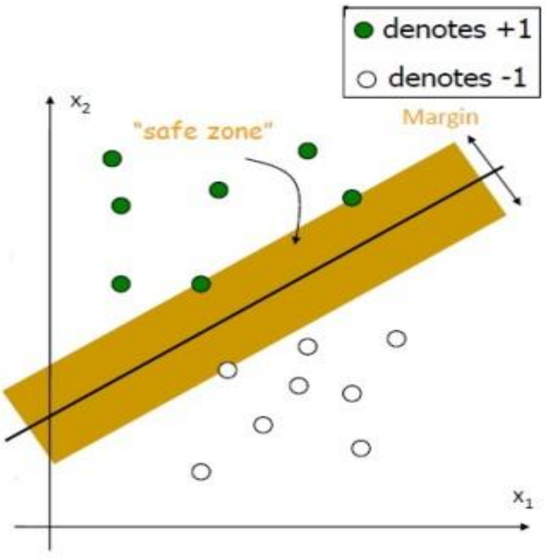


Fig (5.3): Hyper-plane with the maximum margin [45]

Therefore it would be preferable to have a hyper- plane with a large margin or even the largest possible margin. The reason we need it is because if we use a hyper plane to classify, it might end up closer to one set of datasets compared to others and we do not want this to happen and thus we see that the concept of maximum margin classifier or hyper plane as an apparent solution. This maximum margin hyper plane is known as the optimal separating hyperplane as shown in figure (5.3). Margin is defined as the width that the boundary could be increased by before hitting a data point. The linear discriminant function (classifier) with the zone of maximum margin is the best.

Suppose x is a vector point and w is weight and is also a vector. Now using the equation of a straight line, we have,

$$\omega^{T}x^{+} + b = 1, \text{ line which just touches class 1} \qquad (29)$$

$$\omega^{T}x^{-} + b = -1, \text{ line which just touches class 2} \qquad (30)$$

$$\omega^{T}x + b = 0, \text{ maximum-margin separator} \qquad (31)$$

Among all possible hyper planes, SVM selects the one where the distance of hyper plane is as large as possible. This maximum-margin separator is determined by a subset of the data points. Data points in this subset are called “support vectors” as shown in figure (5.4).

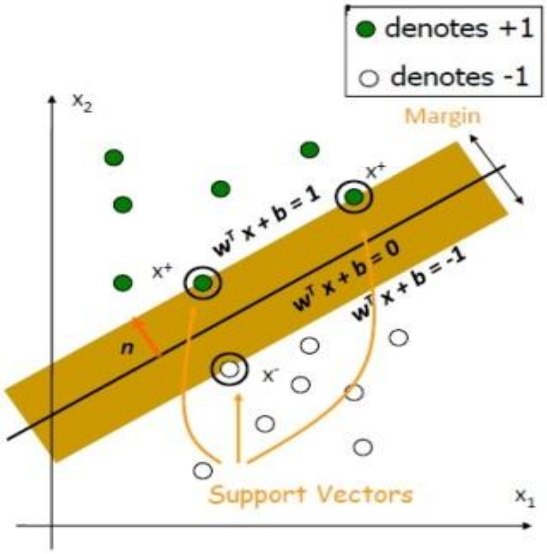


Fig (5.4): Maximum-margin separator and support vectors [45]

Distance of closest point on hyper-plane (support vectors) to origin can be found by maximizing the x as x is on the hyper plane. Similarly for the other side points we have a similar scenario. Thus solving and subtracting the two distances we get the summed distance from the separating hyper-plane to nearest points. Maximum Margin (M), is given by,

$$M = (x^{+} - x^{-}).\, n$$

$$= (x^{+} - x^{-}).\frac{\omega}{\|\omega\|} = \frac{2}{\|\omega\|} \qquad (32)$$

To find the optimal separating hyper-plane we have to find the hyper-plane which satisfies the above condition and maximizes the minimum distance between the hyper-plane and any sample of the training data. It is the sum of the distances from the nearest two points to the separating hyper-plane, which has to be maximized [45].

The above illustration is the maximum linear classifier with the maximum range. In this context it is an example of a simple linear SVM classifier. Critically examining the maximum margin concept, we have another reason is that even if we've made a small error in the location of the boundary this gives us least chance of causing a misclassification [42]. The other advantage would be avoiding local minima and better classification. A classification task usually involves with training and testing data which consist of some data instances. Each instance in the training set contains one target values and several attributes. The goal of SVM is to produce a model which predicts target value of data instances in the testing set which are given only the attributes.

The major strengths of SVM are the training is relatively easy. No local optimal, unlike in neural networks [40]. It scales relatively well to high dimensional data and the trade-off between classifier complexity and error can be controlled explicitly [30].

5.4 RESULTS

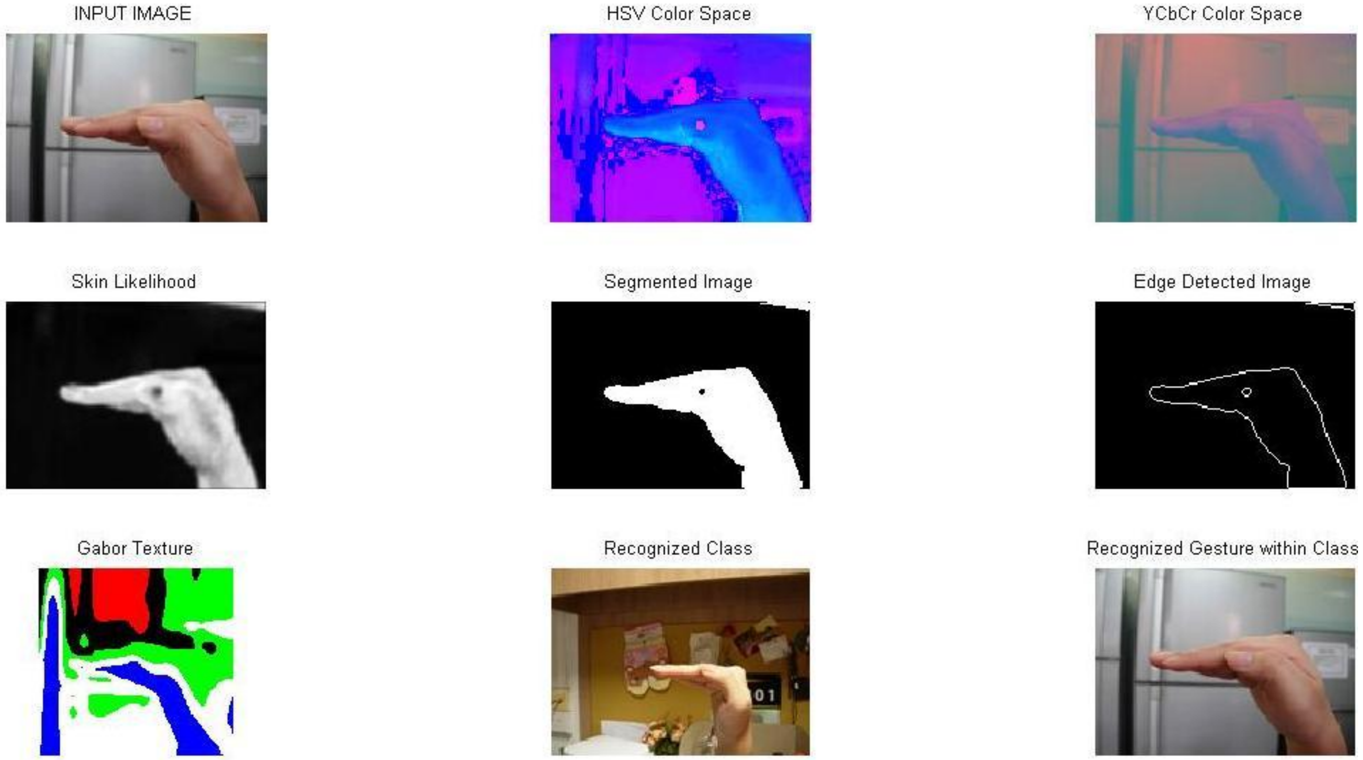


Fig (5.5): Output of various operations applied on Input Image to fetch the Recognized Gesture

5.5 CLASSFICATION RESULT USING SVM CLASSIFIER

A table of confusion (sometimes also called a confusion matrix), is a table with specified rows and columns that reports the number of false positives, false negatives, true positives, and true negatives, in relation to the Actual Data and Predicted Data [30]. This allows more detailed analysis than mere proportion of correct guesses (accuracy). Accuracy is not a reliable metric for the real performance of a classifier, because it will yield misleading results if the data set is unbalanced (that is, when the number of samples in different classes vary greatly). Accuracy can be of little help, if classes are severely unbalanced. Furthermore, accuracy assumes equal cost for both kind of errors (FP and FN).

	Predicted Class		
Actual Class		Class = Yes	Class = No
	Class = Yes	TP	FN
	Class = No	FP	TN

Table (5.1) Accuracy analysis table

The True Positive (TP) and True Negative (TN) are the correct classifications. But, False Positive (FP) and False Negative (FN), give the incorrectly classified data. False Positive (FP) is the lower risk. It’s similar to the false probability in the Bayesian Theory. It means, that the dataset doesn’t belong to the actual class, but the classifier has recognized it as belonging to that class. It is just a precautionary classification, and so this error has comparatively low degree of cost. However, False Negative (FN) has a high a higher degree of cost associated with it. It’s similar to the probability of miss in the Bayesian Theory. It means that the dataset belongs to the actual class, but the classifier has not taken it into that class. Hence, the classifier has missed the correct data to be recognized as the actual class. It shows the poor classification, and thus has the risk of maximum error.

TABLE (5.2): Confusion Matrix for the classification of the different hand gestures classes.

IINPUT GESTURE CLASS		RECOGNIZED GESTURE CLASS																								
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**RR**
	1	93	3	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93
	2	1	94	3	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94
	3	0	0	98	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98
	4	0	2	0	96	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96
	5	3	0	2	0	95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95
	6	0	0	0	0	0	98	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98
	7	0	0	0	0	0	0	96	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96
	8	0	0	0	0	0	0	0	97	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97
	9	0	0	0	0	0	1	0	1	98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98
	10	0	0	0	0	0	0	0	5	1	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94
	11	0	0	0	0	0	0	0	0	0	0	92	4	0	4	0	0	0	0	0	0	0	0	0	0	0	92
	12	0	0	0	0	0	0	0	0	0	0	7	93	0	0	0	0	0	0	0	0	0	0	0	0	0	93
	13	0	0	0	0	0	0	0	0	0	0	2	0	95	0	3	0	0	0	0	0	0	0	0	0	0	95
	14	0	0	0	0	0	0	0	0	0	0	0	4	4	90	2	0	0	0	0	0	0	0	0	0	0	90
	15	0	0	0	0	0	0	0	0	0	0	7	0	0	0	93	0	0	0	0	0	0	0	0	0	0	93
	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	99	0	0	0	1	0	0	0	0	0	99
	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	91	3	2	3	0	0	0	0	0	91
	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	89	1	5	0	0	0	0	0	89
	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	92	8	0	0	0	0	0	92
	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	93	0	0	0	0	0	93
	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98	0	0	0	2	98
	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97	0	3	0	97
	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95	5	0	95
	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	94	0	94
	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	0	95	95

Accuracy Percentage = (Accurately recognized gestures/ Total recognized gestures) * 100%

= (2365/2500) * 100%

= 94.6%

5.6 RESULT AND DISCUSSION

In our testing, we took total 275 images to classify a hand gesture. With our training images, which have well smoothed contour, the recognition accuracy is 94.6%, which is better than recognition accuracy calculated in HMM [10] and intensity based methods [23, 33]. In our project used skin detection method gives invariance against varying lightening condition and different backgrounds. Our proposed method is seen as rotation and orientation invariant. It has been observed that PCA (Principal Component Analysis) used for dimensionality reduction, gives better result as compare to LDA [36] for the large positional variation of images.

Our system has strict requirement on edge smoothing. When we try to classify those images that turn out to have large noise, the recognition accuracy drops quickly to around 40%. So we need to develop better contour smoothing algorithm. As compared to our proposed method Intensity based method has less computing complexity and is less sensitive to for small edge noises [46].

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

From all the explanation and application of the theories and algorithms relevant in hand gesture recognition, we have taken out our own hand gesture algorithm, which involves classification of a gesture by image acquisition, efficient use of different color spaces, segmentation, morphological techniques, skin similarity image representation and finally the application of SVM classifier to recognize the hand gesture. This is a new technique, which is competent even in complex backgrounds. It is also invariant of lateral orientation, which has a lot of significance as it makes the hand gesture recognition system realistic. Moreover, our technique is also independent of the intensity of light in the image.

This system shows some difficulty in recognizing the hand gesture when the background is similar to the skin color. It is also dependent on the skin segmentation algorithm, as if the hand portion is not segmented properly then the hand gesture will not be recognized accurately. Besides having computational complexity, this technique also finds difficulty when the camera resolution is low, which makes the input image of poor quality, and hence the accurate recognition rate falls.

However, even though the technique finds some problem, but those limitations fall way short of the robustness and state of the art nature of our method.

6.2 FUTURE WORK

As we have explained in the above section, our technique finds little difficulty in few situations. So our future work mainly focuses on sorting out these limitations:

- Our method deals with recognition of the hand gestures off-line so work can be done to do it for real time. Then, our hand recognition system can be useful in many fields like robotics, human computer interaction.
- Reduction of complexity leads us to a less computation time. So, the Support Vector Machine can be modified to reduce the complexity. Then the reduced complexity provides us less computation time, and we may easily go for real time recognition.

REFERENCES

[1] Pramod Kumar Pisharady Prahlad Vadakkepat Ai Poh Loh, “Attention Based Detection and Recognition of Hand Postures Against Complex Backgrounds” Int J Comput Vis(2013).

[2]Chaves-González, J. M., Vega-Rodrígueza, M. A., Gómez- Pulidoa, J. A., & Sánchez-Péreza, J. M. (2010). Detecting skin in face recognition systems: a colour spaces study. Digital Signal Processing, 20(03), 806–823.

[3] Henrik Birk and Thomas Baltzer Moeslund, “Recognizing Gestures from the Hand Alphabet Using Principal Component Analysis”, Master’s Thesis, Laboratory of Image Analysis, Aalborg University, Denmark, 1996.

[4] Andrew Wilson and Aaron Bobick, “Learning visual behavior for gesture analysis,” In Proceedings of the IEEE Symposium on Computer Vision, Coral Gables, Florida, pp. 19-21, November 1995.

[5] Jennifer Schlenzig, Edward Hunter, and Ramesh Jain, “Recursive spatio-temporal analysis: Understanding Gestures”, Technical report, Visual Computing Laboratory, University of San Diego, California, 1995.

[6] Daniel, K., John, M., & Charles, M. (2010). A person independent system for recognition of hand postures used in sign language. *Pattern Recognition Letters*, *31*, 1359–1368

[7] Christopher Lee and Yangsheng Xu, “Online, interactive learning of gestures for human robot interfaces” Carnegie Mellon University, The Robotics Institute, Pittsburgh, Pennsylvania, USA, 1996.

[8] Richard Watson, “Gesture recognition techniques”, Technical report, Trinity College, Department of Computer Science, Dublin, July, Technical Report No. TCD-CS-93-11, 1993.

[9] Thomas G. Zimmerman , Jaron Lanier , Chuck Blanchard , Steve Bryson , Young Harvill, “A hand gesture interface device”, SIGCHI/GI Proceedings, conference on Human

factors in computing systems and graphics interface, p.189-192, April 05- 09, , Toronto, Ontario, Canada, 1987.

[10] Hyeon-Kyu Lee and Jin H. Kim," An HMM-Based Threshold Model Approach for Gesture Recognition" IEEE transactions on pattern analysis and machine intelligence, vol. 21, no. 10, october 1999.

[11] Rick Kjeldsen and John Kender,"Finding skin in color images", In Proc. IEEE Int. Conf. on autom. Face and Gesture Recognition, pages 3 12-3 17, 1996

[12] Etsuko Ueda, Yoshio Matsumoto, Masakazu Imai, Tsukasa Ogasawara. "Hand Pose Estimation for Vision- based Human Interface", IEEE Transactions on Industrial Electronics, Vol. 50, No. 4, pp. 676–684,2003.

[13] Chan Wah Ng, Surendra Ranganath, "Real-time gesture recognition system and application", Image Vision Comput, 20(13-14): 993-1007, 2002.

[14] Claudia Nölker and Helge Ritter, "Visual Recognition of Continuous Hand Postures", IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, vol.31, no.1, February 2001

[15] Bowden and Sarhadi, "Building temporal models for gesture recognition" British Machine Vision Conference, pages 32-41, 2000.

[16] Matthew A. Turk and Alex P. Pentland, " Face recognition using eigenfaces", IEEE Society Conference on Computer Vision and Pattern Recognition, pages 586–591, Lahaina, Maui, Hawaii, June 3-6 1991.

[17] J. Edward Jackson, "A Users Guide to Principal Components", Wiley Series in Probability and Mathematical Statistics, A Wiley-Interscience Publication, 1st edition, 1991.

[18] http://www.ece.nus.edu.sg/stfpage/elepv/NUS-HandSet/.

[19] E. Welch, R. Moorhead, J. K. Owens, "Image Processing Using The HSI Color Space," IEEE, 1991.

[20] S. M. Khaled, M. S. Islam, M. G. Rabbani, M. R. Tabassum, A. U. Gias, M. M. Kamal, H. M. Muctadir, A. K. Shakir, A. Imran, and S. Islam, "Combinatorial Color Space Models

for Skin Detection in Sub-continental Human Images," Springer-Verlag Berlin Heidelberg, IVIC 2009, LNCS 5857, pp. 532–542, 2009.

[21] A. S. Ghotkar, G. K. Kharate, "Hand Segmentation Techniques to Hand Gesture Recognition for Natural Human Computer Interaction," International Journal of Human Computer Interaction (IJHCI), Volume (3): Issue (1): 2012.

[22] Prateem Chakraborty, Prashant Sarawgi, Ankit Mehrotra, Gaurav Agarwal, Ratika Pradhan, "Hand Gesture Recognition:A Comparative Study" Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol II MECS 2008, 19-21 March, 2008, Hong Kong.

[23] N.OTSU, "A Threshold Selection Method from Gray-Level Histograms", IEEE transactions on systems, man, and cybernetics, vol. smc-9, no. 1, January 1979.

[24] Lalit Gupta and Suwei Ma "Gesture-Based Interaction and Communication: Automated Classification of Hand Gesture Contours", IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 31, no. 1, February 2001

[25] E. R. Dougherty, "An Introduction to Morphological Image Processing", Bellingham, Washington: SPIE Optical Engineering Press, 1992.

[26] L. Gupta and T. Sortrakul, "A Gaussian mixture based image segmentation algorithm," Pattern Recognit., vol. 31, no. 3, pp. 315–325, 1998.

[27] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human computer interaction: A review," IEEE Trans. Pattern Anal. Machine Intell., vol. 19, pp. 677–694, July 1997.

[28] L. Gupta, T. Sortrakul, A. Charles, and P. Kisatsky, "Robust automatic target recognition using localized boundary representation," Pattern Recognit., vol. 28-10, pp. 1587–1598, 1995.

[29] Vladimir I. Pavlovic, Rajeev Sharma, Thomas S. Huang, "Visual Interpretation of Hand Gestures for Human- Computer Interaction: A Review", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 677-695, July 1997

[30] Vikramaditya Jakkula, “Tutorial on Support Vector Machine (SVM)” School of EECS, Washington State University, Pullman 99164.

[31] M. A. amin and H. Yan, Sign Language Finger Alphabet Recognition from Gabor-PCA Representation of hand gestures, presented at the Proceeding of the sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 2007.

[32] R.PATIL and P. D. S. M.B., Dimensionality Reduction of Satellite images using Principal Component analysis, International Journal of Communication Engineering Applications-IJCEA, vol. 02, July- Oct 2011.

[33] R. C. Gonzales and R. E. Woods. Digital Image Processing. Prentice Hall, second edition, 2002. 795 pages, ISBN 0-201-18075-8.

[34] M. Petrou and P. Bosddogianni. Image Processing: The Fundamentals. John Wiley and Sons, Inc., UK, 2000. 335 pages, ISBN 0-471-99883-4.

[35] H. H. Barret. Foundations of Image Science. John Wiley & Sons, New Jersey, U.K., third edition, 2004.

[36] Lalit Gupta and Suwei Ma “Gesture-Based Interaction and Communication: Automated Classification of Hand Gesture Contours” , IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 31, no. 1, February 2001.

[37] Shigeo Abe, “Support Vector Machines for Pattern Classification, second edition”, Kobe University, Graduate School of Engineering, 2nd edition, Springer-Verlag London Limited 2005, 2010.

[38] http://www.cs.rug.nl/~petkov/publications/journals

[39] Xuewen wang XiaoQuing Ding, Changsong liu, “ Gabor filter –based feature extraction for character recognition”.

[40] O. L. Mangasarian, David R. Musicant, “Lagrangian Support Vector Machines”, Journal of Machine Learning Research, 161-177 (2001).

[41] M. Mudrova´, A. Proch´azka, “Principal Component analysis in Image Processing” Institute of Chemical Technology, Prague Department of Computing and Control Engineering.

[42] Vikramaditya Jakkula,, “Tutorial on Support Vector Machine (SVM)” ,Vikramaditya Jakkula, School of EECS, Washington State University, Pullman, www.ccs.neu.edu/course/cs5100/resources/SVMTutorial.doc.

[43 Joni-Kristian Kamarainen, “Gabor Features in Image Analysis” Machine Vision and Pattern Recognition Laboratory, Lappeenranta University of Technology (LUT Kouvola), 2008.

[44] J. Weston, C. Watkins, “ support vector machines”, Proceedings of ESANN99, Belgium, 1999.

[45] Liu Yucheng and Liu Yubin, “Incremental Learning Method of Least Squares Support Vector Machine”, International Conference on Intelligent Computation Technology and Automation” VCL-94-104, 2010

[46] Xingyan Li, “Vision Based Gesture Recognition System with High Accuracy” Department of Computer Science the University of Tennessee Knoxville, TN 37996-3450.

[47] http://asi.insa-rouen.fr/enseignants/~arakotom/toolbox/index.html

[48] R.Dhanabal, V.Bharathi, G.Prithvi Jain, Ganeash Hariharan, P.Deepan Ramkumar , Sarat Kumar Sahoo, “Gabor Filter Design for Gesture Recognition Using Matlab and Verilog HDL” R.Dhanabal et al. / International Journal of Engineering and Technology (IJET),2007

[49] Sanjay Meena “A study on hand gesture recognition technique”, 2010