Chapter 1: Introduction

1.1 Introduction

In daily life, human beings communicate with each other and interact with computers using gestures. As a kind of gesture, Sign Language (SL) is the primary communication media for deaf people. Everyday, millions of deaf people all over the world are using SL to get useful information and exchange ideas. Therefore, in recent years, SL recognition has gained a lot of attention and a variety of solutions have been proposed.

Signing as a tool makes deaf lives easier and facilitates their relationship with other hearing individuals. Sign Language allows deaf people to develop mentally and intellectually and helps them to develop an identity both individually and as a group. Furthermore it makes them break through their isolation for their greater integration into society. Sign Languages are not usually uniform, but have dialects that differ from country to country and region to region. As modern technology adapts to the needs of people with disabilities, information & communication technologies are becoming a more flexible medium for deaf people to communicate amongst themselves and with hearing individuals as well as enhance their level of learning / education. Experience has shown that education is a vital component of development. It immediately has an encouraging and enabling effect on the people concerned: they will be stimulated to participate, take initiative and use their skills and abilities more fully thereby ensuring an ongoing improvement in their quality of lives.

Everyone requires help in gaining and effectively using information and not just those individuals who have disabilities. In itself, information is neither accessible nor inaccessible; the form in which it is presented makes it so. People with hearing impairment are often handicapped because information is presented in a form inaccessible to them. So solutions that provide general access can provide benefit at a wider scale.

Sign languages are composed of two types of signs:

- Static signs
- Dynamic/Continuous signs

In static signs only hand shape and orientation is analyzed for recognition of signs. Whereas in case of dynamic or continuous signs, along with shape and orientation, direction of movement and sequence of gestures is also considered to completely understand full dynamic sign.

1.2 Pakistani Sign Language

Around 250,000 deaf Pakistanis most commonly communicate using Pakistan Sign Language (PSL), a visual-gestural language that emerged as a result of interaction with Urdu and other regional languages of the country. PSL is recognized as a community language of deaf Pakistanis and has been adopted as an integral part of total communication approach by the Directorate General of Special Education's (DGSE) 1990 official policy. Unfortunately however, owing to multiple variations introduced by language/ dialect differences across the country and economic and expertise constraints, sign language resources and interpreters have always been at a premium and the use of PSL generally as also in the deaf teaching approaches has not been coherent. Lot of research is carried out for automatic recognition of SLs of different countries, most prominently American Sign Language (ASL). Unfortunately PSL is always ignored. Making Pakistani Sign Language assisted learning obtainable through Internet can alleviate barriers to communication on a larger scale and help remove the handicap. Though foreign Sign Languages could be used to enrich the national/local language, they should never replace the national/local Sign Language and dialects. Thus making Pakistani Sign language easily available will serve this purpose.

There are many varieties of sign language in the region, including many pockets of home sign and informal sign languages. There is no consensus regarding which of these varieties constitute dialects of a language or separate languages, but several researchers have identified relatedness between the sign languages used in urban regions of India, Pakistan and Nepal [18] [19]. It is unknown whether this group is related to other languages of the subcontinent such as sign languages in Bangladesh or Sri Lanka.

For this thesis, only alphabets of PSL have been automatically recognized. Figure 1.1 shows alphabets of PSL. There are thirty seven alphabets in PSL. All alphabets are static right handed signs.

1.3 Literature Review

In taxonomies of communicative hand/arm gestures, Sign Language (SL) is often considered as the most structured form of gesture, while gestures that accompany verbal discourse are described as the least standardized. As non-SL gestures often consist of small limited vocabularies, they are not a useful benchmark to evaluate gesture recognition systems. However, SL on the other hand can offer a good benchmark to evaluate different gesture recognition systems because it consists of large and well-defined vocabularies, which can be hard to disambiguate by different systems. In real life, we can imagine many different useful applications for SLR such as:

- Sign-to-text/speech translation system or dialog systems for use in specific public domains such as airports, post offices, or hospitals. This is done by McGuire et al. [1], Akyol and Canzler [2].

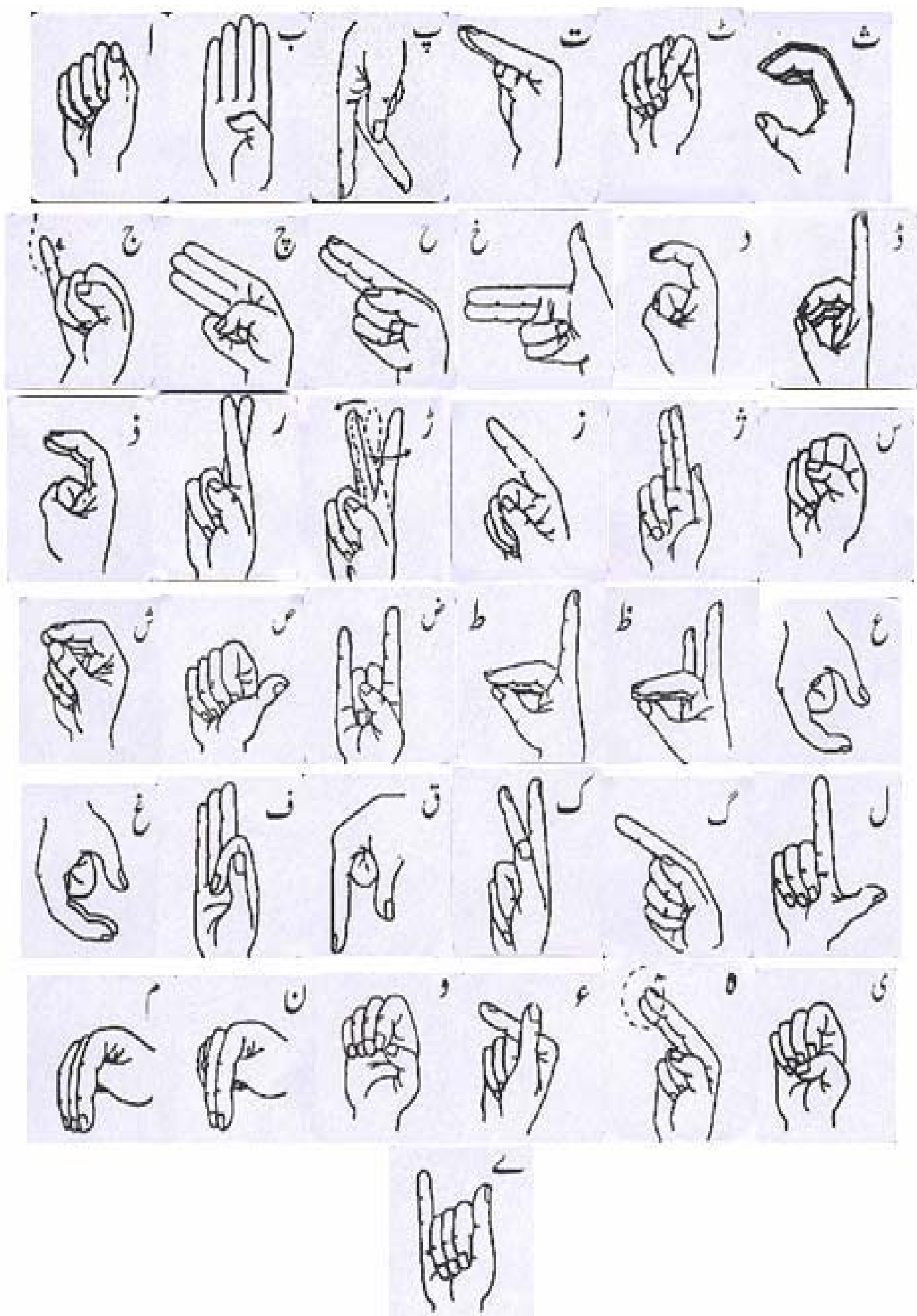


Figure 1.1: Alphabets of PSL

- In video communication between deaf people, instead of sending live videos, SLR can help to translate the video to notations which are transmitted and then animated at the other end to save bandwidth by Kennaway [3].
- SLR can help in annotating sign videos, as proposed by Koizumi et al. [4], for linguistic analysis to save a lot of human labour manually in ground truthing the videos.

According to the means of capturing features, SL recognition techniques can be classified into two groups: glove-based and vision-based. The former group of approaches requires users to wear data or color gloves. The glove enables the system to avoid or simplify the segmentation and tracking task. In comparison, the vision-based methods rely on computer vision techniques without needing any gloves. One difficulty in case of vision-based method is how to accurately segment and track hands. Researchers have proposed a large number of methods to recognize hand signs. These systems are based on different distinct features of signs to recognize the signs. Yoon et al. have used location, angles and velocity to recognize hand gestures[5]. Yin and Xie [6] used neural networks to classify hand signs. Depth silhouettes have also been used to recognize hand gestures by Salinas et.al. [7]. Dong, Wu and Hu have proposed quadratic curve based method to recognize gestures [8]. O. Al Jerrah et al. have used Neural networks to recognize Arabic sign languge [9]. Aleem Khalid et al. used statistical template matching for recognizing Pakistani Sign Language (PSL)[10]. Tanibata and Shimada used a combination of the color cue and template matching technique for their Japanese Sign Language (JSL) recognition system [11].

Many researchers have carried out researches for automatic recognition of SLs of different countries such as American Sign Language (ASL) [14], Arabic Sign Language [9], Malaysian Sign Language (Bahasa Isyarat Malaysia (BIM)) [12], Irish Sign Language [17], Japanese Sign Language (JSL) [11], Australian Sign Language (AUSLAN) [13], Dutch Sign Language [15], Chinese Sign Language [16] and many more. Unfortunately PSL could not get attention of researchers for its automatic recognition. I could find just one research publication on automatic recognition of PSL by Aleem Khalid et al. [10]. Technique used in this [10] research is very simple that is template matching. Inherent problems of template matching do not allow this research to cater for large vocabulary of PSL. So there is a need of more well structured, robust and better technique to be used for automatic recognition of PSL and this is the aim of this thesis.

1.4 Methodology

For this thesis, alphabets of PSL are automatically recognized. Block diagram of the methodology adopted for this thesis is shown in Figure 1.2.

RGB image of a hand showing sign is taken as input. Signer has to wear a colored glove on his hand. Segmentation of hands and fingers is based on the colors of the gloves. Ten features are extracted in feature extraction module. These features are angles between each finger-tip and its corresponding finger-joint. These extracted features are then passed to the classifier to recognize signs of alphabets of PSL. The classifier used for this thesis is fuzzy classifier. This classifier then classifies each signed alphabet and display the alphabet in written form as output that was fed to the system as a sign.

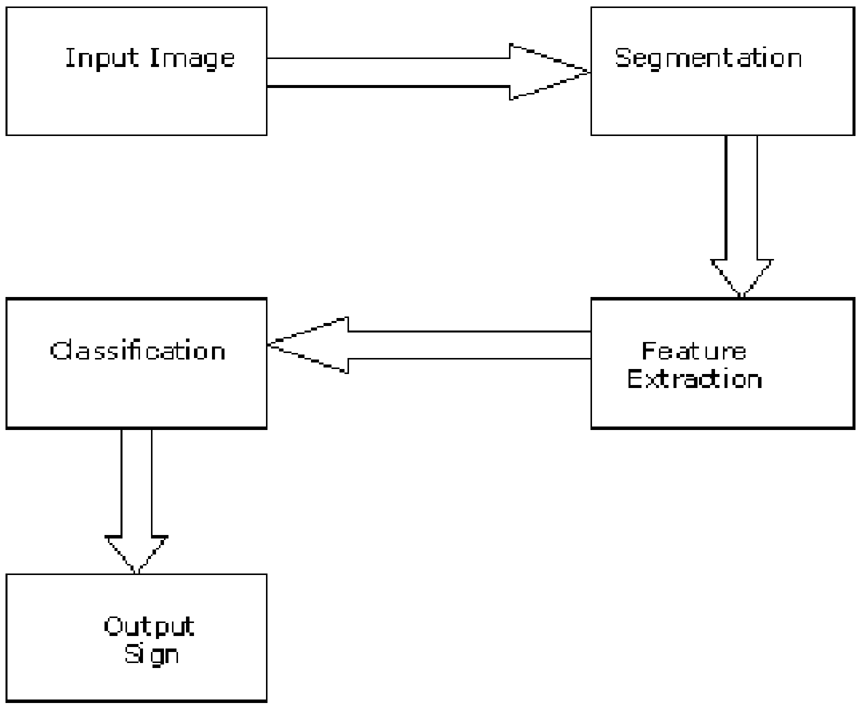


Figure 1.2: Block Diagram of PSL Recognizer

Chapter 2: Segmentation

2.1 Introduction

In the analysis of the objects in images it is essential that we can distinguish between the objects of interest and "the rest." This latter group is also referred to as the background. The techniques that are used to find the objects of interest are usually referred to as *segmentation techniques* - segmenting the foreground from background [20].

Different techniques are presented for improving the quality of the segmentation result. It is important to understand that there is no universally applicable segmentation technique that will work for all images and no segmentation technique is perfect.

Many human computer interaction applications use computer vision systems to track human hands. The segmentation of human hands is preliminary task in all these systems. For accomplishing objective of this thesis i.e. to recognize signs of PSL, accurate segmentation of hands from its background is very important. As segmentation accuracy determines the eventual success or failure of the computerized analysis procedure for the recognition of signs.

Different researchers proposed different methods for segmentation of hands for variety of applications. Broadly there are two segmentation techniques:

- Shape-based Segmentation
- Color-based Segmentation

For this thesis firstly skin-based segmentation technique, a color-based segmentation technique, was adopted to segment human hands but later on in the phase of feature extraction it was observed that the results obtained from this technique are putting many constraints on feature selection. So at that point in time this segmentation technique was replaced with another color-based technique. The later color-based technique uses color gloves for segmentation of human hands. The technique used for this proposed thesis i.e. color gloves based segmentation is discussed at the end of this chapter.

2.2 Shape-based Segmentation

Shape-based Segmentation is not very commonly used segmentation technique for human hands. Shape models are fitted to get the hand region. Along with shape models, curvatures of fingers are commonly used to segment out human hands. But it is observed that only shape can not give the accurate results, so many researchers included skin color information in their segmentation algorithms. When shape-based segmentation is used, normally some other information is also required for accurate segmentation such as position, motion or some color information etc. Shape-based hand segmentation accuracy is very much dependent upon the gesture of the hand. If, for example, Shape-based segmentation is considering the shape of the palm, number of fingers, orientation of fingers, curvature of fingers, shape of wrist and orientation of palm with wrist. If all these calculations have to make to identify a hand, it is very computing intensive process. For showing hand signs, hand have to be in different gestures, and there is very high probability of missing many above mentioned features in many signs. For example a sign shown in Figure 2.1, it is difficult to check the curvature of fingers or other features associated with fingers.

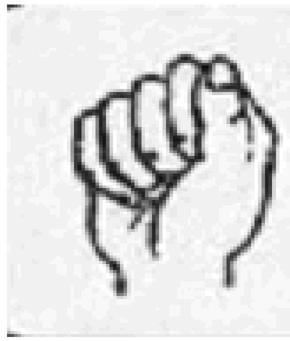


Figure 2.1: Hand gesture

So there can be a very high rate of inaccurate segmentation. These are the reasons, that researchers have normally avoided this technique of segmentation. Shape-based segmentation can be implemented using neural network, template matching etc.

2.3 Color-based Segmentation

Color segmentation takes a great attention because color is an effective and robust visual cue for characterizing an object from the others. There are two possibilities to use color information while segmenting human hand; Skin-color or colored marks on hands.

2.3.1 Skin-Color-based segmentation

A very commonly adopted method for human hand segmentation is skin color-based segmentation, as human skin color can be an important feature for hand regions in color images. Skin region segmentation in color image is gaining more and more attention of the researchers in recent years. As its applications range from object recognition, hand segmentation for gesture recognition [21], face detection and tracking [22] [23], filtering of objectionable images on web [24], visual tracking, human–computer interaction (HCI), human–robot interaction (HRI), vision-based robotics, visual surveillance, and so forth.

Human skin-color segmentation can be performed using variety of techniques such as neural network [28], stochastic models [26], fuzzy clusters [27], Discrete Cosine Transfer (DCT) [30], B-Spline curve modeling [31] and many more. Many researchers used other features of human hands along with skin color to have more accurate segmentation such as edge information [29], hand geometry and so forth.

However, color segmentation is not robust enough to deal with complex environments. Especially, changing illumination condition and complex background containing surfaces or objects with similar colors to a target are the major problems that limit its applications in practical real world. It is observed that accuracy of skin color model for segmenting human hands is very much dependent on given conditions. The factors that can affect the performance of skin-based segmentation include: illumination conditions, color model to be used, skin color tone, level of occlusion in background, viewing geometry of camera and many more. So robustness of the skin-based segmentation becomes very low.

For this thesis, hands were first segmented using skin-based segmentation technique. It is a matter of fact that while using human skin color model for segmentation, it is an important consideration to select appropriate color model. For this research different color models are analyzed to segment out human hands in a given RGB image with complex background, with variation in skin color tones and different illumination conditions. We used four color models for comparison: RGB, HSV, YUV, and Lab.

RGB (Red Green Blue) is most commonly known color model. RGB is basically an additive color model as it is based on the concept that what type of light is required to be emitted to get a particular color. RGB color model individually stores the values of red, green and blue colors. Normally cameras use this model. HSV (Hue Saturation Value) is a commonly used color model in computer graphics applications. It describes the color more closely to the way human perceive a color. YUV defines a color space in terms of one luma (Y) and two chrominance (U and V) components. Lab color space is an informal abbreviation of CIE 1976 (*L**, *a**, *b**) color space.It is said to be a color-opponent space with *L* dimension for luminance and two color-opponent dimensions a and b [25]. We observed that these color models changed their response with varying illumination conditions. Another observation is that skin tone variation does not have much effect on the response of these color models. The analysis of these color models suggested that if lighting conditions are good, HSV, Lab and YUV color models are very successful in segmenting human skin for indoor images. But if illumination conditions are not very good, YUV performs better than HSV. RGB is not the right choice for indoor images, no matter whatever lighting conditions are available. For outdoor images, if image is captured in proper sunlight, Lab color model is best choice for segmenting skin color pixels. YUV also performs better than other two color models for images in proper sunlight. But if the outdoor images have shadows on skin color pixels, YUV is best at segmenting human skin. HSV also quite accurately segments the human skin in shadowed images. So it can be concluded that color model selection for human skin color segmentation is highly dependent on the given conditions in which original image is captured. Segmentation results for different color models are summarized in Table 2.1 and 2.2.

	INDOOR IMAGES	
	Normal Light	
	Accuracy rate %	False rejection%
Lab	97.4	2.6
YUV	95.3	4.7
HSV	92.1	7.9
RGB	21	79

Table 2. 1: Segmentation results of indoor images

	OUTDOOR IMAGES

	Normal Light	
	Accuracy rate %	False rejection%
YUV	90.2	9.8
Lab	88.4	11.6
HSV	83.7	16.3
RGB	16.7	83.3

Table 2. 2:Segmentation results for outdoor images

After analysis of the color models, Lab color model was selected as to be used for segmentation. Figure 2.2 is showing the segmentation result using Lab color model for an indoor image given in Figure 2.1.

Figure 2. 1: Original Image

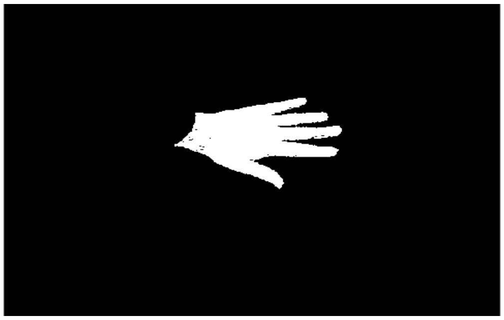


Figure 2. 2: Segmented Image

Normally when skin color is used as a feature to segment human skin regions, many constraints apply. These constraints are mainly on skin color tones and illumination conditions. These two factors are very crucial while defining human skin color range. With varying illumination conditions and variation in skin tones, a non adaptive skin range can not perform very well. So if these variations have to be catered for, multiple ranges have to be defined or some other features apart from skin color have to be added to properly perform segmentation of human skin regions. A more robust approach to segment human hands that can solve these problems is proposed. This robustness is achieved by defining a unique skin color range for every image. This image specific definition of skin color range makes segmentation more accurate. This adaptive approach is robust for illumination changes and skin tone variations, without considering additional features for segmentation. The results achieved from this adaptive technique are shown in Figure 2.3, 2.4, 2.5 and 2.6.

Figure 2. 3 : Original Image

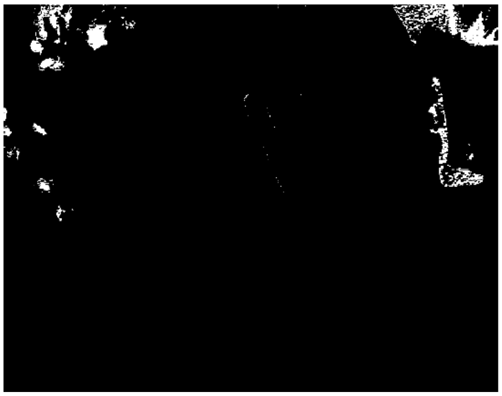


Figure 2. 4: non-adaptive segmentation

Figure 2. 5: Adaptive segmentation with MAX_ITR=5

Figure 2. 6: adaptive segmentation with MAX_ITR=15

The proposed adaptive approach of skin-Color-based segmentation is shown in Figure 2.7.

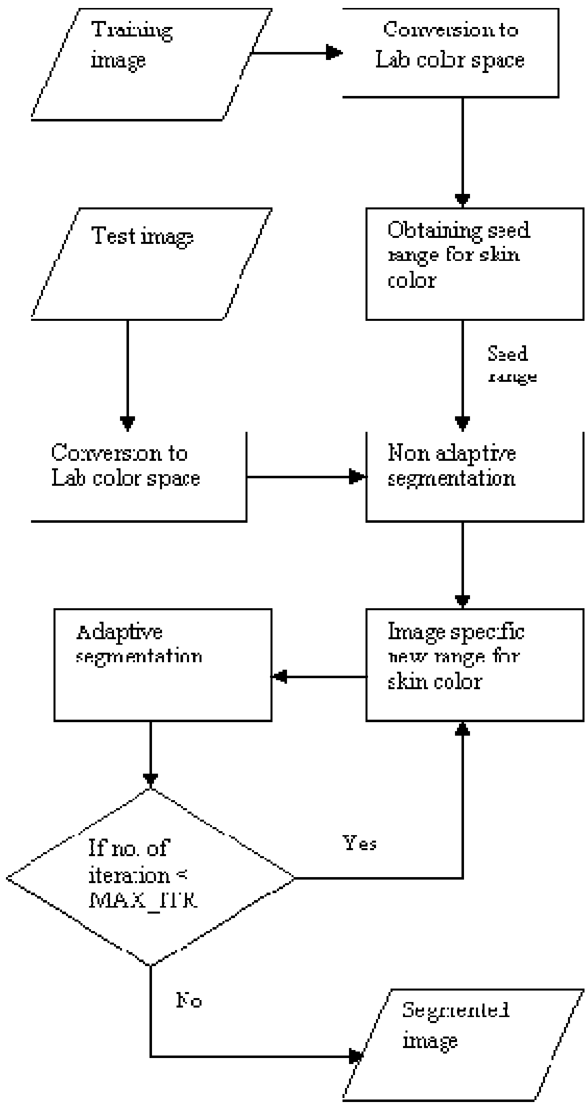


Figure 2. 7:Flow chart of adaptive skin Color-based segmentation

MAX_ITR is a constant defining upper limit for the iterations to get the adaptive range.

Skin-color-based technique is very frequently used by researchers for segmentation of hands for sign language recognition. As it is a natural clue to detect human hands without any external device dependence. But it puts a lot of constraints, some are discussed above that can effect the accuracy of the segmentation. If all these conditions are satisfied and segmentation results are hundred percent accurate, even then feature extraction from this segmented hand is very difficult and again put a lot of constraints on the classifier to be used to recognize sign language. These constraints are discussed in detail in a later chapter of classifier.

2.3.2 Color-mark-based Segmentation

Existing Human Computer Interaction HCI devices for hand gesture recognition fall into two categories: glove-based and vision-based systems. The glove-based system relies on electromechanical devices that are used for data collection about the gestures [32, 33, 34, 35, 36]. Here the person must wear some sort of wired gloves that are interfaced with many sensors. Then based on the readings of the sensors, the gesture of the hand can be recognized by a computer interfaced with the sensors. Because glove-based systems force the user to carry a load of cables and sensors, they are not completely natural the way an HCI should be. The second category of HCI systems has overcome this problem. Vision-based systems basically suggest using a set of video cameras, image processing, and artificial intelligence to recognize and interpret hand gestures [32]. These techniques are utilized to design visual-based hand gesture systems that increase the naturalness of human-computer interaction. The main attraction of such systems is that the user is not plagued with heavy wired gloves and has more freedom and flexibility. In pure visual-based hand gesture systems, segmentation is quite difficult and accuracy of such systems is very much dependent upon the gesture of the hand. This technique also put constraints on the size of the dictionary of the signs to be recognized. In wired, sensor-based gloves there is no issue of segmentation but it make system dependent on heavy external devices. So there should be a hybrid technique that can combine the advantages of both techniques. This is accomplished by using specially designed gloves with visual markers that help in determining hand postures, as presented in [37, 38, 39]. A good review about vision-based systems can be found in [40]. Once the data has been obtained from the user, the recognition system, whether it is glove-based or vision-based, must use this data for processing to identify the gesture.

Due to problems faced in the classification phase, the skin Color-based segmentation is replaced with color mark based segmentation. In this technique a colored glove is used to segment human hand. The gloves worn by the participants were marked with eleven different colors at different eleven regions as shown in Figure 2.8.

Each Acquired image is fed to the image processing stage in which color representation and image segmentation are performed for the gesture. By now, the color of each pixel in the image is represented by three values for red, green, and blue (RGB). In the image segmentation stage, the color information is used for segmenting the image into eleven regions representing the five finger-tips, five finger-joints and the wrist.

The intention of the finger tracker is to determine enough information to robustly track the position, orientation and pose of the user’s fingers. For real-time determination of these parameters without the need to constrain environmental conditions, marked gloves are used.

Figure 2. 8:colored glove

Colored marks on the finger-tips and finger-joints are marked in such a way that these marks are visible to camera in all types of gestures. Finger-tips are marked with colors all around instead of a single blob or cross mark on any one side of the finger. As if finger-tips are partially marked with a blob or cross mark, there are chances of turning away of this mark from the camera in many signs. Same is true for finger-joints.

The principal task the segmentation process has to perform is the estimation of the center of gravity for each marker. The center of gravity is computed from the weighted contributions of the pixels covered by the markers in the given colored image. We have implemented threshold based segmentation, because it is simple and able to work in real-time. Pixel values which are above a given threshold are used to estimate the center of gravity of the marker image.

Chapter 3: Feature Extraction

3.1 Introduction

In pattern recognition and in image processing, Feature extraction is a special form of dimensionality reduction. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant (much data, but not much information) then the input data will be transformed into a reduced representation set of features (also named features vector). Transforming the input data into the set of features is called *features extraction*. If the features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input [41]. These features are used by the pattern classifier to recognize different features.

Feature extraction involves simplifying the amount of resources required to describe a large set of data accurately. When performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power or a classification algorithm which overfits the training sample and generalizes poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy.

In many applications such as shape recognition, it is useful to generate shape features which are independent of parameters which cannot be controlled in an image. Such features are called invariant features. There are several types of invariance such as location invariance, rotation invariance, size invariance etc. So it is required to select such invariant features to increase efficiency and accuracy of the classifier using these features.

Feature selection in the design of pattern classifiers has three goals:

- To reduce the cost of feature extraction
- To improve the classification accuracy
- To increase invariance of features

There are some general dimensionality reduction techniques such as Principal components analysis, Semi-definite embedding, Multifactor dimensionality reduction, Non-linear dimensionality reduction, Isomap, Kernel PCA, Latent semantic analysis, Partial least squares [41]. But it is a common observation that best results are achieved when an expert constructs a set of application-dependent features. That’s why researcheres, working on development of techniques to recognize sign language, mostly donot rely on general dimensionality reduction techniques. Rather they go for features which are specific to the domain of sign language to get better accuracy and efficiency.

The performance of a general recognition system first depends on getting efficient features to represent pattern characteristics. Several methods of representing such hand gesture feature are proposed by the researchers. Features are extracted on the basis of key-points. So selecting appropriate key-points to extract accurate features is very important. Most commonly used features and key-points for static sign language recognition are hand edges, silhoutes, shape moments and finger tips. These features are discussed in this chapter in the light of three goals of feature selection in field of pattern classification explained above and at the end of this chapter, features (i.e. angles between finger-tips and finger-joints) selected for this thesis are discussed.

3.2 Shape Moments

Descriptors are some set of numbers that are produced to describe a given shape. The shape may not be entirely reconstruct able from the descriptors, but the descriptors for different shapes should be different enough that the shapes can be discriminated. Descriptor to describe shape using statistical properties are called shape *moments* [52].

Many researchers have gone for the selection of shape moments as features to classify hand gestures and sign language due to its strong inherent property of invariance. Yoon and Jo have used shape moments as feature to recognize alphabets of Korean sign language [51]. Shape moments are also selected as features to recognize sign language by Salleh and et.al[53]. Lange and et.al. also selected shape moments as features to recognize hand gestures [54].

Shape moments can be good as features to recognize sign language but these have certain limitations. If less number of moments are selected as features, accuracy of classifier using these features may suffer. On the other hand if number of descriptors increases, accuracy rate may improve but size of feature set increases and hence increase in computation cost. If size of vocabulary of signs large and there are many similar sign in the vocabulary, moments may not be proved as good features. Some shape moments show rotation invariance, but size invariance and location invariance is not possible to be achieved while using shape moments.

3.3 Edges

Edge detection is a terminology in image processing and computer vision, particularly in within the areas of feature detection and feature extraction, to refer to algorithms which aim at identifying points in a digital image at which the image brightness changes sharply or more formally has discontinuites [45].

The result of applying an edge detector to an image may lead to a set of connected curves that indicate the boundaries of objects, the boundaries of surface markings as well curves that correspond to discontinuities in surface orientation. Thus, applying an edge detector to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may be regarded as less relevant, while preserving the important structural properties of an image. If the edge detection step is successful, the subsequent task of interpreting the information contents in the original image may therefore be substantially simplified.

Many researchers has used edges as features to recognize sing language. Munib and et.al. used hand edges as feature which is later trained by neural network to recognize American Sign Language [42]. Hand contours are also used as a feature of hand to recognize hand gesture by Yasushi Hamada and et.al [43]. Classification is based on hand edges in research conducted by Jaklic and et.al. to recognize sign language.

Edge, as feature for sign language recognition, is a very simple technique and adopted by many researchers. Edges are easy to detect and there are many well-tested techniques to detect edges. These edges can be then used by any of the pattern classifier such as artificial neural networks, template matching and support vector machine and so on. But adopting such a simplified approach expunges many features of the sign language.

Edge as a feature for sign language recognition is analyzed in the light of three characteristics of a good feature extractor explained in introduction of this chapter. There are some limitations of this technique and due to these limitations edge features are usually combined with some other features to get better accuracy. If edges are taken as the basic feature, there are many other features to be taken along with this basic feature of edge such as curvatures, orientation, distances and many more. This lead to increase in size of the feature set. Computation cost for getting accurate edges of hands can be quite high as multiple techniques may be required to apply to get proper edges of hands. Similarly computation cost to extract all these features is quite high. If multiple features are selected carefully and then extracted accurately, high accuracy can be achieved for the classifier based on these features. Features based on edges or edges itself as feature don’t show any type of invariance i.e. location, size or rotation.

3.4 Silhouettes

A silhouette is a view of some object or scene consisting of the outline and a featureless interior, with the silhouette usually being black [46]. It is also defined as a drawing consisting of the outline of something, especially a human profile, filled in with a solid color [47].

Hand silhouettes as basic feature for sign language classifier are used by many researchers. Akahashi has used silhouettes to estimate human posture [48]. Chua and et.al. have also used hand silhouettes to recognize hand gestures [49]. Depth silhouettes are used for gesture recognition by Salinas and et.al. [50].

Hand silhouettes like edges usually selected as the basic feature and on the basis of this basic feature many other features are selected such as curvatures, orientation, distances between curvatures etc. As many other features are used with this basic feature, so size of feature set is normally large. Getting accurate silhouettes of hands may need heavy computation and if additional features are also considered, computation cost further increases. High accuracy rate for sign language classifier can be achieved if silhouettes are properly computed and additional features are accurately selected and extracted.

Property of invariance also lack in hand silhouette as features or features based on these silhouettes.

3.5 Finger-tips

The most commonly used key-points for feature extraction in the domain of sign language recognition are finger-tips. Finger-tips of a human hand are shown in Figure 3.1.

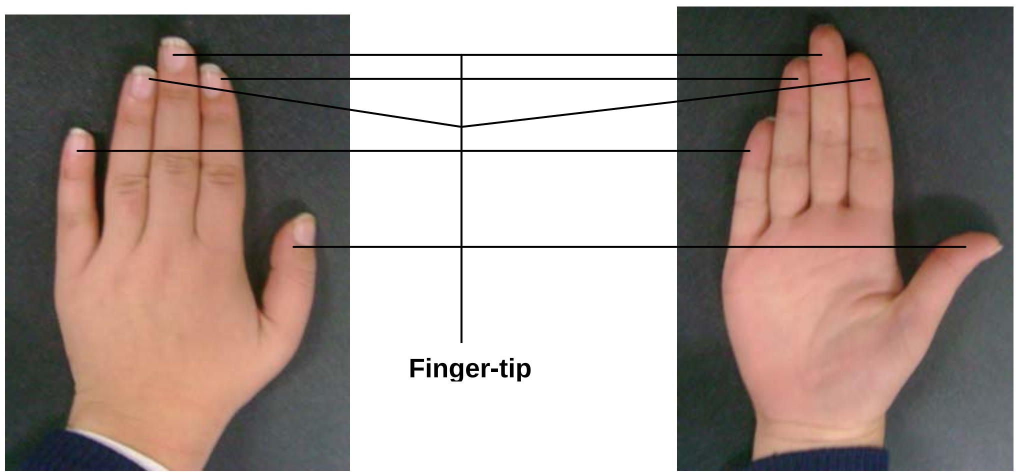


Figure 3.1: Finger_tips of hand

Finger positions are the key to recognize signs of a sign language and finger–tips can be used to identify different positions and orientations. Looking at the potential of finger-tips as the key feature many researchers have used these to extract features. Oka and et.al. have used finger-tips as the key point to recognize human hand gestures[56]. Finger-tips are also used to extract features for recognition of Arabic Sign Language by Assaleh [55]. Bedregal et al. have used finger-tips along with some other key points to recognize Brazilian Sign Language[57].

Features such as distances and angles between different finger-tips, their orientation, their relative position with respect to wrist etc. are based on the finger-tips which are passed to the classifier and it is observed that these features give results with high accuracy rate to recognize sign languages. But there are some limitations of this technique. It is very difficult to precisely get finger-tips in variety of hand gestures specially when relying purely on visual cues without any external device. Different techniques are utilized to correctly find the key points that are finger-tips in this case as whole system's accuracy is dependent on these. But high accuracy in finding finger-tips, without external device assistance, requires heavy computation. So such computation intensive techniques raise the feature extraction computation cost. Feature set based on finger-tips, which have capability to make classifier an accurate one, is normally not a smaller one making classification process more computation intensive. Assaleh and et.al.[55] have used colored gloves with marks on finger-tips to recognize Arabic Sign Language extracted thirty features from the segmented color regions. This number is three times higher than the number of features used for the proposed technique of this thesis.

3.6 Finger-tips and Finger-joints

Human beings are able to distinct signs in real world on the basis of the finger position and hand orientation. These two parameters are utilized in the proposed method of sign language recognition and are considered while selecting features for classifier. The features used for classification in this thesis are distance calculation for *bent* position of finger and angle between the finger-tip and corresponding finger-joint for all other finger positions. These angles have the capability to uniquely represent each sign.

To recognize alphabets of PSL, feature used in this thesis are angles between finger-tip and their corresponding finger-joint for all five fingers. Angles are provided as input to fuzzy inference system and then system identifies the position of each finger and on the basis of these positions sign recognition has been performed.

Feature extraction phase for this thesis is based on two steps:

- Point Calculation
- Angle Calculation

3.6.1 Point Calculation

Colored image of a hand wearing colored gloves is taken as input (shown in Figure 3.2) to point calculation module.

Figure 3.2: Marked glove

Each finger-tip and its corresponding joint have distinct color on the glove. One distinct colored marker has been used for wrist. On the basis of these colors, each finger-tip and each finger-joint is segmented. Then, each segment's centroid is calculated. These points are (x_{1f},y_{1f}) for joint of finger f and (x_{2f},y_{2f}) for corresponding finger-tip(Where f = {1,2,3,4,5}). These centroid points are then passed to next module to calculate angles.

3.6.2 Angle Calculation

Any possible position of a given finger can be defined with the help of its angle between its tip and joint. Fingers can be configured for a sign by changing its position different directions. Point calculation module calculates two points for each finger. For angle calculation between these two points, third point is also required. To calculate the third point (x_{3f}, y_{3f}), Equations 1 and 2 have been used.

$$x_{3f} = \begin{cases} x_{1f} & x_{2f} < x_{1f} \\ x_{1f} & x_{2f} > x_{1f} \\ x_{2f} & x_{2f} = x_{1f} \end{cases} \quad (1)$$

$$y_{3f} = \begin{cases} y_{1f} + | x_{1f} - x_{2f}| & x_{2f} < x_{1f} \\ y_{1f} - | x_{1f} - x_{2f}| & x_{2f} > x_{1f} \\ y_{2f} & x_{2f} = x_{1f} \end{cases} \quad (2)$$

After calculating three points, Euclidean distance has been calculated to get the measure of sides to be further used to measure the angles. Measure of sides has been calculated with the equations 3, 4 and 5.

$$a= ((x_{3f}-x_{2f})^2+(y_{3f}-y_{2f})^2)^{½} \quad (3)$$

$$b= ((x_{1f}-x_{3f})^2+(y_{1f}-y_{3f})^2)^{½} \quad (4)$$

$$c= ((x_{1f}-x_{2f})^2+(y_{1f}-y_{2f})^2)^{½} \quad (5)$$

Angles are calculated using cosine rule shown in equations 6, 7 and 8 (where A, B and C are the calculated angles).

$$a^2=b^2 +c^2- 2*b*c*\cos(A) \quad (6)$$

$$b^2=a^2 +c^2- 2*a*c*\cos(B) \quad (5)$$

$$c^2=a^2 +b^2- 2*a*b*\cos(C) \quad (8)$$

Plane for angles is shown in Figure 3.3.

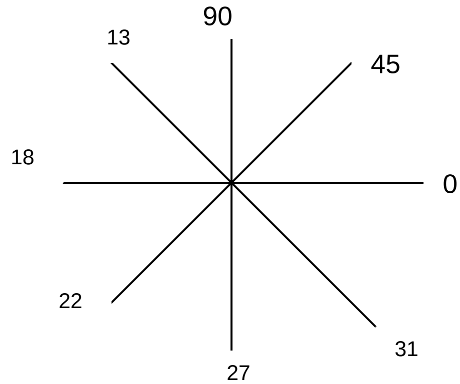


Figure 3.3: plane

Corresponding to the angles, the linguistic terms used in this thesis are shown in Figure 3.4

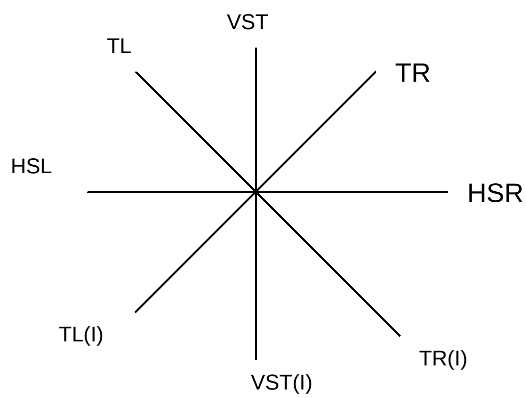


Figure 3.4: Linguistic terms for plane

Acronyms used in above figures are for: Horizontal Straight Right (HSR), Tilted Right (TR), Vertical Straight (VST), Tilted Left (TL), Horizontal Straight Left (HSL), Tilted Left Inverted (TLI), Vertical Straight Inverted (VSTI) and Tilted Right Inverted (TRI) and BENT. These positions are used by the classifier to recognize the sign.

There is a special type of finger position that can not be accurately defined with the help above mentioned criterion of angle calculation and that position is *bent* position of four fingers. Thumb bent position can be handled with above calculations of angles. For determining whether a finger is bent or not is checked by measuring and comparing two distances d_A and d_B.

d_A = Distance between finger-joint and thumb-joint

d_B = Distance between thumb-joint and wrist-mark.

Where distance between two points is not Euclidean distance, rather it is simply difference of x-coordinates of two points that can be given with the equation.

$X_{25}-X_{1f} <= X_w - X_{25}$

Where f = {1, 2, 3, 4, 5}

If $d_A < d_B$
Finger position is BENT
Else
Finger position is not BENT

If finger is BENT then a special angle (i.e. 380 in this thesis) assigned and passed to the classifier.

Feature selection in the design of pattern classifiers has three goals: to reduce the cost of extracting features, to improve the classification accuracy and to increase invariance in features. The feature selected for this thesis has achieved all these three goals. Such features has been selected which has reduced the size of feature set. Only six features, five angles of fingers and bent position has been used. Six features are comparatively very low in number to accurately recognize signs. Less number of features with low computational requirements for extracting features has enabled the proposed technique to minimize the cost of extracting features. As all possible finger positions are catered for in this thesis, so the reliability of the estimate of performance has been improved. These features have been also achieved size invariance, location invariance and rotation invariance. These features have shown their potential to give results with very high accuracy rates.

Chapter 4: Classification

4.1 Introduction

Classifiers are functions that use pattern matching to determine a closest match. They can be tuned according to examples, making them very attractive for use in Artificial Intelligence. These examples are known as observations or patterns. A class can be seen as a decision that has to be made. All the observations combined with their class labels are known as a data set [58]. Pattern classification is a sub-topic of machine learning. It can be defined as the act of taking in raw data and taking an action based on the category of the data [60]. Most research in pattern recognition is about two main categories of learning:

- Supervised Learning
- Unsupervised Learning

In supervised learning, each pattern belongs to a certain predefined class. Outcome is predicted on the bases of input features. In case of supervised learning, training data have both outcome and feature measurements, with this data a predictor or learner is built up. In case of unsupervised learning only features are observed, there is no a priori output and goal is to describe how data is organized or clustered [59].

There are many learning algorithms available in the field of pattern classification and people are still discovering new algorithms that they hope will work better. Any new learning algorithm, beside its theoretical foundation, needs to be justified in many aspects including accuracy and efficiency when applied to real life problems.A wide range of classifiers are available, each with its strengths and weaknesses. Classifier performance depends greatly on the characteristics of the data to be classified. There is no single classifier that works best on all given problems; this is also referred to as the "no free lunch" theorem. Various empirical tests have been performed to compare classifier performance and to find the characteristics of data that determine classifier performance. Determining a suitable classifier for a given problem is however still more an art than science.

In this chapter, an introduction is given to most commonly used pattern classifier for static hand signs alongwith their advantages and disadvantages and at the end the pattern classifier that is used for this thesis, Fuzzy classifier, is discussed.

4.2 Template Matching

Template matching is a technique in Digital image processing for finding small parts of an image which match a template image [64]. There are different approaches to accomplish template matching. Some are faster performers than others, and some find better matches.

The basic method of template matching uses a convolution mask (template), tailored to a specific feature of the search image, which we want detect. This technique can be easily performed on grey images or edge images. It is intuitively likely that the convolution output will be highest at places where the image structure matches the mask structure, where large image values get multiplied by large mask values [64].

Template matching is a very simple technique of pattern matching. So is used by many researchers to recognize sign language or hand gestures. Takanao and et.al. did use template matching technique for hand gestures tracking [65]. Aleem Khalid and et.al. used statistical template matching for recognizing PSL [66]. Tanibata and Shimada used a combination of the colour cue and template matching technique for their Japanese Sign Language (JSL) recognition system [67].

The template matching method involves matching a template of a defect-free model to the image of the object under inspection. Objects that are defect free ideally result in high response values from the classifier, while objects that are defective render low response values.The template matching method is considered to be the simplest method to recognize a particular pattern in the given image. It is simple to implement and to understand. But there are some problems with this method.One of the problems with this method is that as the size of the template increases, the computational cost increases significantly. This is particularly true when the positions of the objects to be identified are not known in advance because the position of a given object then has to be determined before applying the template. Another disadvantage of the template matching method is the need to construct a potentially large database with all possible templates of the objects and their positions. That’s why this method has lost its position as the top choice for reseracher as a classifier to recognize hand gestures or sign language

4.3 Artificial Neural network

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in union to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well. Structure of a simple ANN is shown in Figure 4.1.

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions [61]. Other advantages include:

a. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
b. Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.
c. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
d. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

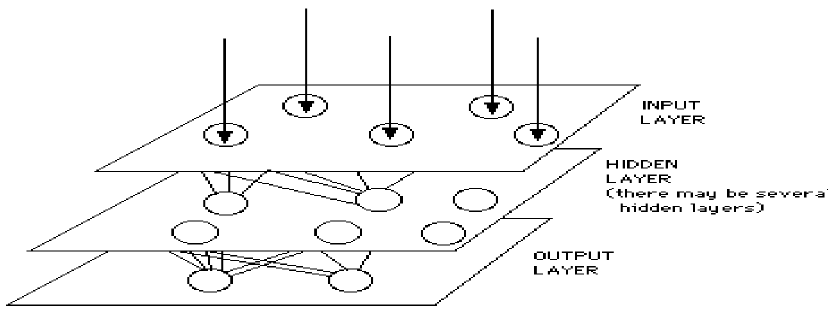


Figure 4. 1:A Simple Neural Network Structure

An important application of neural networks is pattern recognition. Pattern recognition can be implemented by using a neural network that has been trained accordingly. During training, the network is trained to associate outputs with input patterns. When the network is used, it identifies the input pattern and tries to output the associated output pattern. A simple example of pattern classification, with two classes (class 0 and class1) and two inputs (x1 and x2) can is shown in Figure 4.2.

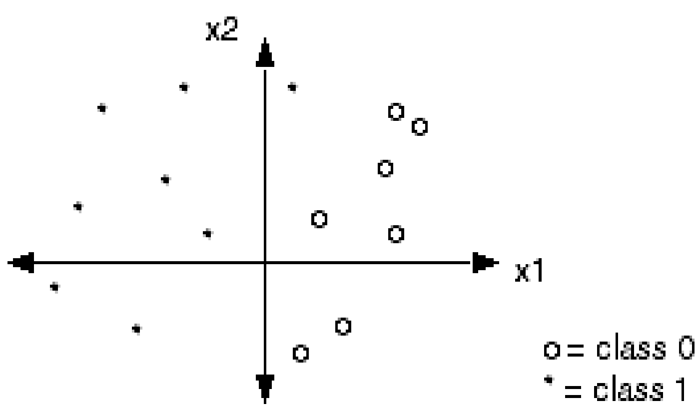


Figure 4. 2: A Simple example of pattern classification

The power of neural networks comes to life when a pattern that has no output associated with it, is given as an input. In this case, the network gives the output that corresponds to a taught input pattern that is least different from the given pattern.

ANN is most commonly used classifier for human hand sign recognition. It is used by O. Al Jerrah and et.al. to recognize Arabic sign languge [62], Xiaoming Yin and Ming Xie to identify hand gestures [63] and many more.

Neural networks are the classifier which is adopted by most of the researchers for sign language recognition because their ability to learn by example makes neural nets very flexible and powerful. There is no need to devise an algorithm in order to perform a specific task; i.e. there is no need to understand the internal mechanisms of that task. Along various other advantages of neural nets there are disadvantages too, neural networks cannot be programmed to perform a specific task; the examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. Also, network finds out how to solve the problem by itself, hence its operation can be unpredictable. Minimizing overfitting requires a great deal of computational effort. The individual relations between the input variables and the output variables are not developed by engineering judgment so that the model tends to be a black box or input/output table without analytical basis. The size of example set has to be large. Training time no matter it is offline is a sort of overhead to the system. So looking at above mentioned disadvantages neural network was not our choice as a classifiers for this thesis.

4.4 Support Vector Machine

Support Vector Machine (SVM*)* is a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. A special property of SVMs is that they simultaneously minimize the empirical classification error and maximize the geometric margin; hence they are also known as maximum margin classifiers [68]. Viewing the input data as two sets of vectors in an *n*-dimensional space, an SVM will construct a separating hyperplane in that space, one which maximizes the "margin" between the two data sets. To calculate the margin, two parallel hyperplanes are constructed, one on each side of the separating one, which are "pushed up against" the two data sets. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the neighboring datapoints of both classes. The hope is that, the larger the margin or distance between these parallel hyperplanes, the better the generalization error of the classifier will be.

Support Vector Machine (SVM) models are a close cousin to classical multilayer perceptron neural networks. Using a kernel function, SVM’s are an alternative training method for polynomial, radial basis function and multi-layer perceptron classifiers in which the weights of the network are found by solving a quadratic programming problem with linear constraints, rather than by solving a non-convex, unconstrained minimization problem as in standard neural network training [69].

In the parlance of SVM literature, a predictor variable is called an *attribute*, and a transformed attribute that is used to define the hyperplane is called a *feature*. The task of choosing the most suitable representation is known as *feature selection*. A set of features that describes one case (i.e., a row of predictor values) is called a *vector*. So the goal of SVM modeling is to find the optimal hyperplane that separates clusters of vector in such a way that cases with one category of the target variable are on one side of the plane and cases with the other category are on the other size of the plane. The vectors near the hyperplane are the *support vectors*. The Figure 4.3 [68] below presents an overview of the SVM process.

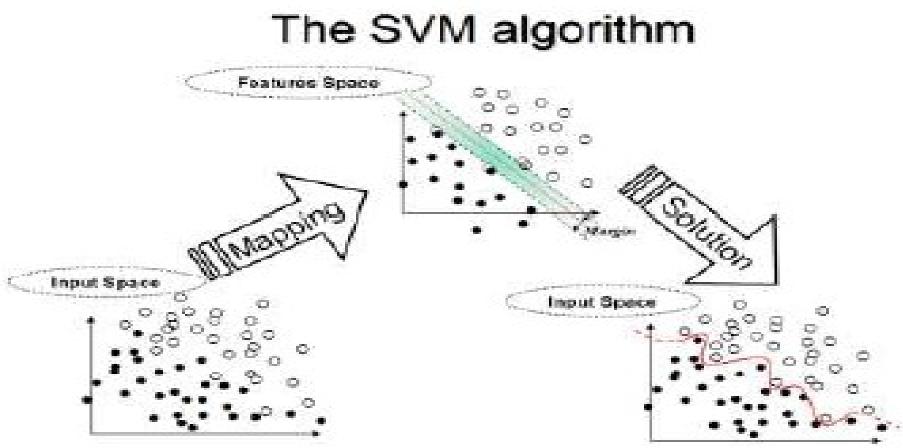


Figure 4. 3: A Simple SVM process

Many researcher has opted SVM as classifier for sign language recognition and hand gestures identification. Salinas and et.al. has used SVM as classifier to recognize human gestures [70]. SVM has been also used by T. Ko and et.al to recognize multimodal human conversation [71].

The key advantages of SVMs are the use of kernels, the absence of local minima, the sparseness of the solution and the capacity control obtained by optimizing the margin. Besides the advantages of SVMs - from a practical point of view - they have some drawbacks. Perhaps the biggest limitation of the support vector approach lies in choice of the kernel. Another limitation is speed and size, both in training and testing. Although SVMs have good generalization performance, they can be abysmally slow in test phase. Another important practical question, difficult to answer, is the selection of the kernel function parameters. From a practical point of view perhaps the most serious problem with SVMs is the high algorithmic complexity and extensive memory requirements of the required quadratic programming in large-scale tasks [72]. These disadvantages lead us for not selecting SVM as the classifier for recognition of PSL.

4.5 Fuzzy Classifier

Any classifier that uses fuzzy sets or fuzzy logic in the course of its training or operation is known as *fuzzy classifier*. Fuzzy classifiers are one application of fuzzy theory. Expert knowledge is used and can be expressed in a very natural way using linguistic variables, which are described by fuzzy sets.

Fuzzy Logic was initiated in 1965 by Lotfi A. Zadeh, professor for computer science at the University of California in Berkeley. Basically, Fuzzy Logic (FL) is a multivalued logic that allows intermediate values to be defined between conventional evaluations like true/false, yes/no, high/low, etc. Notions like rather tall or very fast can be formulated mathematically and processed by computers, in order to apply a more humanlike way of thinking in the programming of computers. A fuzzy system is an alternative to traditional notions of set membership and logic that has its origins in ancient Greek philosophy. The precision of mathematics owes its success in large part to the efforts of Aristotle and the philosophers who preceded him. In their efforts to devise a concise theory of logic, and later mathematics, the so-called "Laws of Thought" were posited [73].

Fuzzy logic starts with the concept of a fuzzy set. A *fuzzy set* is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership. To understand what a fuzzy set is, first consider the definition of a *classical set*. A classical set is a container that wholly includes or wholly excludes any given element. For example, the set of days of the week unquestionably includes Monday, Thursday, and Saturday. It just as unquestionably excludes butter, liberty, and dorsal fins, and so on.

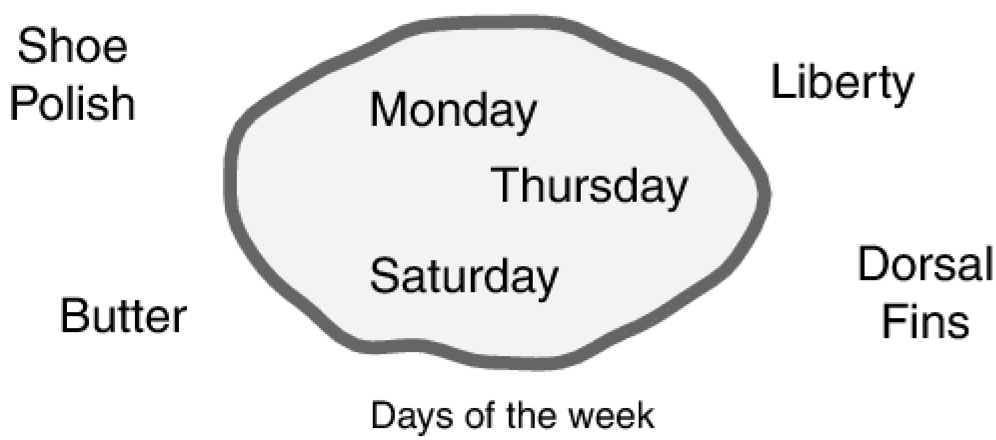


Figure 4. 4: An example of crisp set

This type of set is called a classical set because it has been around for a long time. It was Aristotle who first formulated the Law of the Excluded Middle, which says X must either be in set A or in set not-A. Another version of this law is: of any subject, one thing must be either asserted or denied. To restate this law with annotations: "Of any subject (say Monday), one thing (a day of the week) must be either asserted or denied (I assert that Monday is a day of the week)." This law demands that opposites, the two categories A and not-A, should between them contain the entire universe. Everything falls into either one group or the other. There is no thing that is both a day of the week and not a day of the week. Considering the set of days comprising a weekend. Figure 4.5 [75] classify the weekend days.

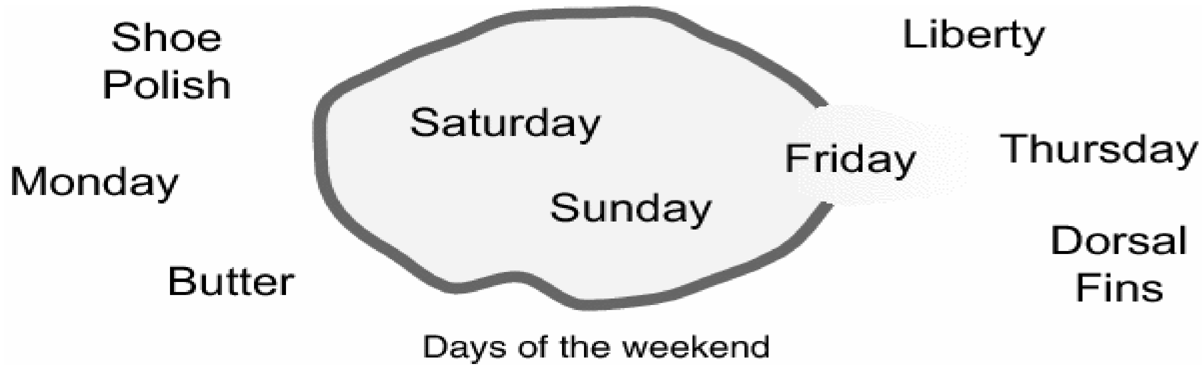


Figure 4. 5: Example of Fuzzy set

Most would agree that Saturday and Sunday belong, but what about Friday? It feels like a part of the weekend, but somehow it seems like it should be technically excluded. Thus, in Figure 5, Friday tries its best to "straddle on the fence." Classical or normal sets would not tolerate this kind of classification. Either something is in or it is out. Human experience suggests something different, however, straddling the fence is part of life. Of course individual perceptions and cultural background must be taken into account to define what constitutes the weekend. Even the dictionary is imprecise, defining the weekend as the period from Friday night or Saturday to Monday morning. This is the entrance of the realm where sharp-edged, yes-no logic stops being helpful. Fuzzy reasoning becomes valuable exactly when one has to work with how people really perceive the concept *weekend* as opposed to a simple-minded classification useful for accounting purposes only. More than anything else, the following statement lays the foundations for fuzzy logic. In fuzzy logic, the truth of any statement becomes matter of degree. Any statement can be fuzzy. The major advantage that fuzzy reasoning offers is the ability to reply to a yes-no question with a not-quite-yes-or-no answer. Humans do this kind of thing all the time, but it is a rather new trick for computers.

Reasoning in fuzzy logic is just a matter of generalizing the familiar yes-no (Boolean) logic. If true is given the numerical value of 1 and false the numerical value of 0, this value indicates that fuzzy logic also permits in-between values like 0.2 and 0.7453. A continuous scale time plot of weekend-ness shown in Figure 4.6 [75].

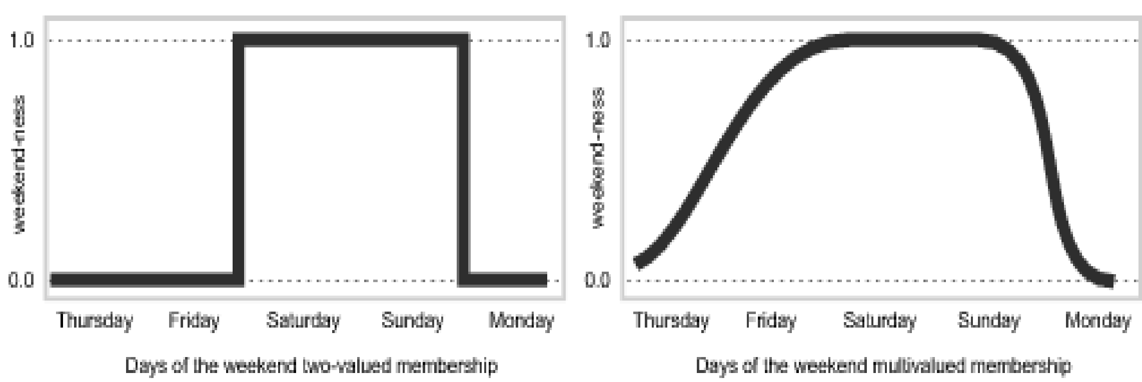


Figure 4. 6: Time plot of weekend-ness

By making the plot continuous, the degree is defined to which any given instant belongs in the weekend rather than an entire day. In the plot on the left, at midnight on Friday, just as the second hand sweeps past 12, the weekend-ness truth value jumps discontinuously from 0 to 1. This is one way to define the weekend, and while it may be useful to an accountant, it may not really connect with real-world experience of weekend-ness. The plot on the right shows a smoothly varying curve that accounts for the fact that all of Friday, and, to a small degree, parts of Thursday, partake of the quality of weekend-ness and thus deserve partial membership in the fuzzy set of weekend moments. The curve that defines the weekend-ness of any instant in time is a function that maps the input space (time of the week) to the output space (weekend-ness). Specifically it is known as a *membership function [75]*.

A *membership function* (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the *universe of discourse*, a fancy name for a simple concept. The only condition a membership function must really satisfy is that it must vary between 0 and 1. The function itself can be an arbitrary curve whose shape can be defined as a function that suits from the point of view of simplicity, convenience, speed, and efficiency [75].

The most important thing to realize about fuzzy logical reasoning is the fact that it is a superset of standard Boolean logic. In other words, if fuzzy values are kept at their extremes of 1 (completely true), and 0 (completely false), standard logical operations will hold. In fuzzy logic the truth of any statement is a matter of degree so classical truth tables required to be modified. One solution is that resolve the statement *A* AND *B*, where *A* and *B* are limited to the range (0,1), by using the function *min*(*A*,*B*). Using the same reasoning, OR operation can be replaced with the *max* function, so that *A* OR *B* becomes equivalent to *max*(*A*,*B*). Finally, the operation NOT *A* becomes equivalent to the operation 1-A. The figure 4.7 [75] uses a graph to show the same information.

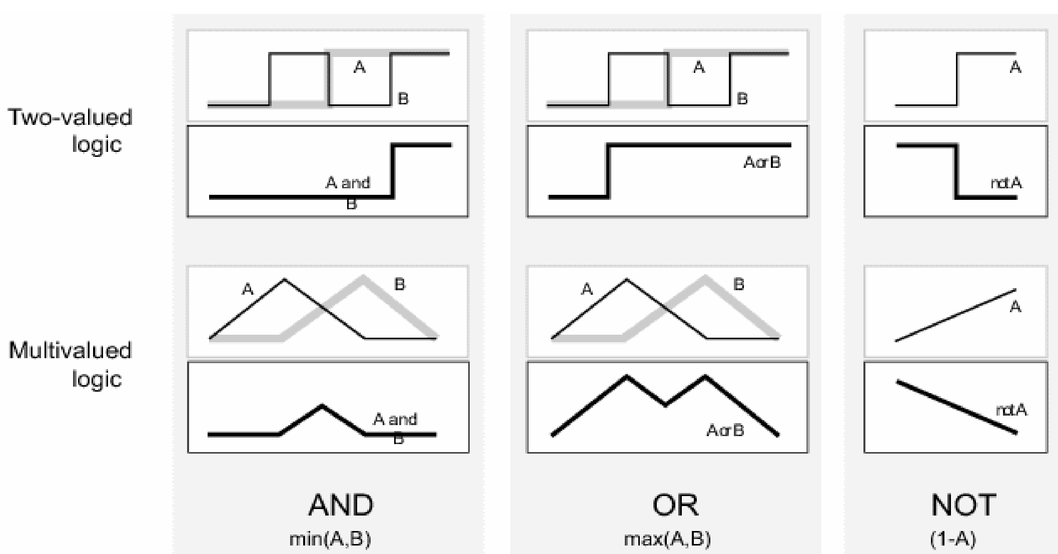


Figure 4. 7: Graph of Two_valued logic and Multivalued logic

In Figure 7, the logic is converted to a plot of two fuzzy sets applied together to create one fuzzy set. The upper part of the figure displays plots corresponding to the two-valued truth tables, while the lower part of the figure displays how the operations work over a continuously varying range of truth values *A* and *B* according to the fuzzy operations defined [75].

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. The process of fuzzy inference involves: Membership Functions, Logical Operations, and If-Then Rules. Two types of fuzzy inference systems are very commonly used: Mamdani-type and Sugeno-type. These two types of inference systems vary somewhat in the way outputs are determined [75]. Fuzzy inference systems have been successfully applied in fields such as automatic control, data classification, decision analysis, expert systems, and computer vision. Because of its multidisciplinary nature, Fuzzy Inference Systems (FIS) are associated with a number of names, such as fuzzy-rule-based systems, fuzzy expert systems, fuzzy modeling, fuzzy associative memory, fuzzy logic controllers, and simply (and ambiguously) fuzzy systems.

Mamdani's fuzzy inference method is the most commonly seen fuzzy methodology. Mamdani's method was among the first control systems built using fuzzy set theory. It was proposed in 1975 by Ebrahim Mamdani [76] as an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators. Mamdani's effort was based on Lotfi Zadeh's 1973 paper on fuzzy algorithms for complex systems and decision processes [20]. *Mamdani-type inference*, expects the output membership functions to be fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable that needs defuzzification. It is possible, and in many cases much more efficient, to use a single spike as the output membership function rather than a distributed fuzzy set.

Fuzzy inference process comprises of five parts: fuzzification of the input variables, application of the fuzzy operator (AND or OR) in the antecedent, implication from the antecedent to the consequent, aggregation of the consequents across the rules, and defuzzification.

The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. The input is always a crisp numerical value limited to the universe of discourse of the input variable and the output is a fuzzy degree of membership in the qualifying linguistic set (always the interval between 0 and 1).

After the inputs are fuzzified, degree is known to which each part of the antecedent is satisfied for each rule. If the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one number that represents the result of the antecedent for that rule. This number is then applied to the output function. The input to the fuzzy operator is two or more membership values from fuzzified input variables. The output is a single truth value.

Consequent is a fuzzy set represented by a membership function, which weights appropriately the linguistic characteristics that are attributed to it. The consequent is reshaped using a function associated with the antecedent (a single number). The input for the implication process is a single number given by the antecedent, and the output is a fuzzy set. Implication is implemented for each rule.

Decisions are based on the testing of all of the rules in a FIS, the rules must be combined in some manner in order to make a decision. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. Aggregation only occurs once for each output variable, just prior to the fifth and final step, defuzzification. The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable.

The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number. As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set.

Fuzzy Logic provides a different way to approach a control or classification problem. This method focuses on what the system should do rather than trying to model how it works. One can concentrate on solving the problem rather than trying to model the system mathematically, if that is even possible. On the other hand the fuzzy approach requires a sufficient expert knowledge for the formulation of the rule base, the combination of the sets and the defuzzification. In General, the employment of fuzzy logic might be helpful, for very complex processes, when there is no simple mathematical model (e.g. Inversion problems), for highly nonlinear processes or if the processing of (linguistically formulated) expert knowledge is to be performed.

Pattern recognition is a field concerned with machine recognition of meaningful regularities in noisy or complex environments. In simpler words, pattern recognition is the search for structures in data. In pattern recognition, a group of data is called a cluster. In practice the data are usually not well defined. That is, pattern recognition is , by its very nature, an inexact science. To deal with the ambiguity, it is helpful to introduce some “fuzziness” int o the formulation of the problem. For example, the boundary between clusters could be fuzzy rather than crisp; that is, a data point could belong to two or more clusters with different degrees of membership. In this way, the formulation is closer to the real-world problem and therefore better performance may be expected. This is the first reason for using fuzzy models for pattern recognition: the problem by its very nature requires fuzzy modeling (in fact, fuzzy modeling means more flexible modeling; by extending the zero-one membership to the membership in the interval [0,1], more flexibility is introduced). The second reason for using fuzzy models is that the formulated problem may be easier to solve computationally. This due to the fact that a nonfuzzy model often results in an exhaustive search in a huge space because some key variables can only take two values 0 and 1, whereas in a fuzzy model all the variables are continuous, so that derivatives can be computed to find the right direction for search [74].

For this thesis, Mamdani's fuzzy inference method of Matlab fuzzy toolbox is used. Ten angles are taken as input to FIS. All angles are fuzzified to check position of each finger. Positions specified in the proposed method are: Horizontal Straight Right (HSR), Tilted Right (TR), Vertical Straight (VST), Tilted Left (TL), Horizontal Straight Left (HSL), Tilted Left Inverted (TLI), Vertical Straight Inverted (VSTI) and Tilted Right Inverted (TRI) and BENT. Membership function of input angles is Gaussian. Membership function for position of index finger for side angle is shown in Figure 4.8 as an example.

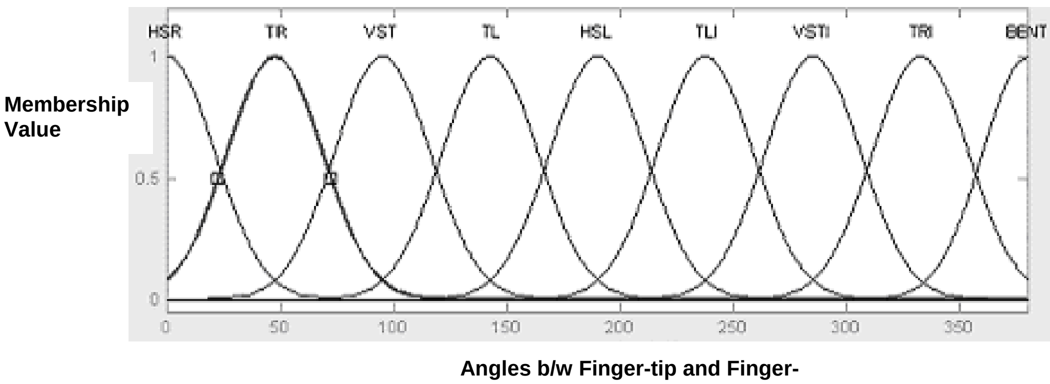


Figure 4. 8: Membership function for finger position

On x-axis of Figure 4.8, angle between finger-tip and finger-joint are given and on y-axis member ship value is shown. When position is determined by FIS, using membership function for finger position, FIS checks the rules to defuzzify the given input. Thirty seven rules in the fuzzy rule base for the proposed method are:

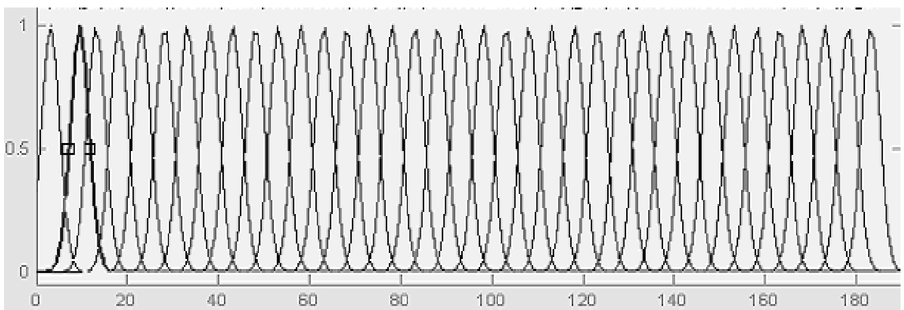
- If (angle1 is BENT) and (angle2 is BENT) and (angle3 is BENT) and (angle4 is BENT) and (angle5 is VST) then (sign is aliph)
- If (angle1 is VST) and (angle2 is VST) and (angle3 is VST) and (angle4 is VST) and (angle5 is TL) then (sign is bay)
- If (angle1 is VSTI) and (angle2 is TRI) and (angle5 is VSTI) then (sign is pay)
- If (angle1 is TL) and (angle2 is TL) and (angle5 is TL) then (sign is tay)
- If (angle1 is BENT) and (angle2 is BENT) and (angle3 is BENT) and (angle4 is BENT) and (angle5 is TL) then (sign is tai)
- If (angle1 is TL) and (angle2 is TL) and (angle3 is TL) and (angle4 is TL) and (angle5 is HSL) then (sign is say)
- If (angle1 is BENT) and (angle2 is BENT) and (angle3 is BENT) and (angle4 is VST) and (angle5 is TL) then (sign is jeem)
- If (angle1 is TL) and (angle2 is TL) and (angle3 is TL) and (angle4 is BENT) and (angle5 is TL) then (sign is chay)
- If (angle1 is TL) and (angle2 is TL) and (angle3 is BENT) and (angle4 is BENT) and (angle5 is TL) then (sign is haa)

- If (angle1 is HSL) and (angle2 is HSL) and (angle3 is BENT) and (angle4 is BENT) and (angle5 is VST) then (sign is khay)
- If (angle1 is TL) and (angle2 is not TL) and (angle3 is not BENT) and (angle4 is not BENT) and (angle5 is TL) then (sign is daal)
- If (angle1 is VST) and (angle2 is HSL) and (angle3 is HSL) and (angle4 is not VST) and (angle5 is TL) then (sign is dal)
- If (angle1 is TL) and (angle2 is TL) and (angle5 is HSL) then (sign is zal)
- If (angle1 is TL) and (angle2 is VST) and (angle3 is BENT) and (angle4 is BENT) and (angle5 is TL) then (sign is ray)
- If (angle1 is VST) and (angle2 is VST) and (angle3 is BENT) and (angle4 is BENT) and (angle5 is TL) then (sign is day)
- If (angle1 is TL) and (angle2 is not BENT) and (angle3 is BENT) and (angle4 is BENT) and (angle5 is TL) then (sign is zay)
- If (angle1 is VST) and (angle2 is VST) and (angle3 is BENT) and (angle4 is BENT) and (angle5 is VST) then (sign is yay)
- If (angle1 is BENT) and (angle2 is BENT) and (angle3 is BENT) and (angle4 is BENT) and (angle5 is HSL) then (sign is seen)
- If (angle1 is HSL) and (angle2 is HSL) and (angle5 is TL) then (sign is sheen)
- If (angle1 is BENT) and (angle2 is BENT) and (angle3 is BENT) and (angle4 is BENT) and (angle5 is HSR) then (sign is suad)
- If (angle1 is VST) and (angle2 is BENT) and (angle3 is BENT) and (angle4 is VST) and (angle5 is TL) then (sign is zuad)
- If (angle1 is VST) and (angle2 is HSL) and (angle3 is not HSL) and (angle4 is not VST) and (angle5 is TL) then (sign is tuay)
- If (angle1 is VST) and (angle2 is HSR) and (angle3 is HSR) and (angle4 is VST) and (angle5 is TR) then (sign is zuay)
- If (angle1 is TRI) and (angle2 is not TRI) and (angle5 is TRI) then (sign is ayn)
- If (angle1 is TRI) and (angle2 is TRI) and (angle5 is TRI) then (sign is ghayn)
- If (angle1 is BENT) and (angle2 is VST) and (angle3 is VST) and (angle4 is VST) and (angle5 is VST) then (sign is fay)
- If (angle1 is VSTI) and (angle2 is not TRI) and (angle5 is VSTI) then (sign is qaf)
- If (angle1 is VST) and (angle2 is VST) and (angle3 is BENT) and (angle4 is BENT) and (angle5 is VST) then (sign is caf)
- If (angle1 is TL) and (angle2 is BENT) and (angle3 is BENT) and (angle4 is BENT) and (angle5 is TL) then (sign is gaf)
- If (angle1 is VST) and (angle2 is BENT) and (angle3 is BENT) and (angle4 is BENT) and (angle5 is HSR) then (sign is laam)
- If (angle1 is VSTI) and (angle2 is VSTI) and (angle3 is VSTI) then (sign is meem)
- If (angle1 is VSTI) and (angle2 is VSTI) then (sign is noon)
- If (angle1 is BENT) and (angle2 is BENT) and (angle3 is BENT) and (angle4 is BENT) and (angle5 is TL) then (sign is wow)
- If (angle1 is TL) and (angle2 is TL) and (angle3 is BENT) and (angle4 is BENT) (angle5 is TL) then (sign is hay)
- If (angle1 is TL)and (angle5 is VST) then (sign is hamza)

- If (angle1 is BENT) and (angle2 is BENT) and (angle3 is BENT) and (angle4 is BENT) and (angle5 is TL) then (sign is cyay)
- If (angle1 is BENT) and (angle2 is BENT) and (angle3 is BENT) and (angle4 is VST) and (angle5 is VST) then (sign is byay)

For this thesis, Min. method has been used for fuzzy operator AND, Max. method has been used for OR operator. Implication method, Min., has been used with Max. aggregation method. There is only one output value of this FIS that is defuzzified value. The crisp output value i.e. sign, is calculated by the smallest of maximum (*som)* defuzzification method. Membership function for defuzzification is shown in Figure 4.9

Membership Value



Defuzzified value for sign

Figure 4. 9: Membership function for defuzzification

On x-axis of Figure 4.9, defuzzified values for each sign are given and on y-axis, membership function of the output value is shown. This defuzzified value is used to get the sign.

Chapter 5: Experiments & Results

5.1 Introduction

Any proposed design can be judged for its accuracy with the results of conducted experiments. Design of PSL recognizer for this thesis has been implemented and experiments are carried out to analyze the results. This chapter first discusses the set up for the experiments and then analysis of the results.

5.2 Experiments

Experiments have been conducted to check the validity of the design of the PSL recognizer. Two important components of the experiments are; the tool to implement the design and dictionary of signs to be tested for the proposed design. These two components are discussed in this chapter.

5.2.1 Fuzzy Logic Toolbox

For this thesis, tool used for the implementation of Fuzzy Inference System (FIS) for recognition of PSL is Fuzzy Logic Toolbox of Matlab 7.0. The Fuzzy Logic Toolbox extends the MATLAB technical computing environment with tools for the design of systems based on fuzzy logic. Graphical user interfaces (GUIs) guides through the steps of fuzzy inference system design. Functions are provided for many common fuzzy logic methods. The toolbox allows modeling complex system behaviors using simple logic rules and then implementing these rules in a FIS. Toolbox can be used as a stand-alone fuzzy inference engine. Alternatively, it can be used as fuzzy inference blocks in Simulink and simulate the fuzzy systems within a comprehensive model of the entire dynamic system. Like all MATLAB toolboxes, the Fuzzy Logic Toolbox can be customized. It can be used to inspect algorithms, modify source code, and add user defined membership functions or defuzzification techniques. It provides specialized GUIs for building FIS and viewing and analyzing results, membership functions for creating fuzzy inference systems, support for AND, OR, and NOT logic in user-defined rules, standard Mamdani and Sugeno-type fuzzy inference systems, automated membership function shaping through neuroadaptive and fuzzy clustering learning techniques, ability to embed a fuzzy inference system in a Simulink model, ability to generate embeddable C code or stand-alone executable fuzzy inference engines.

Due to the facility and ease of use provided by fuzzy logic toolbox, it was used to check the results of the system designed. FIS editor of the toolbox provides a user friendly customizable environment to set parameters of the system. FIS editor of this thesis is shown in Figure 5.1.

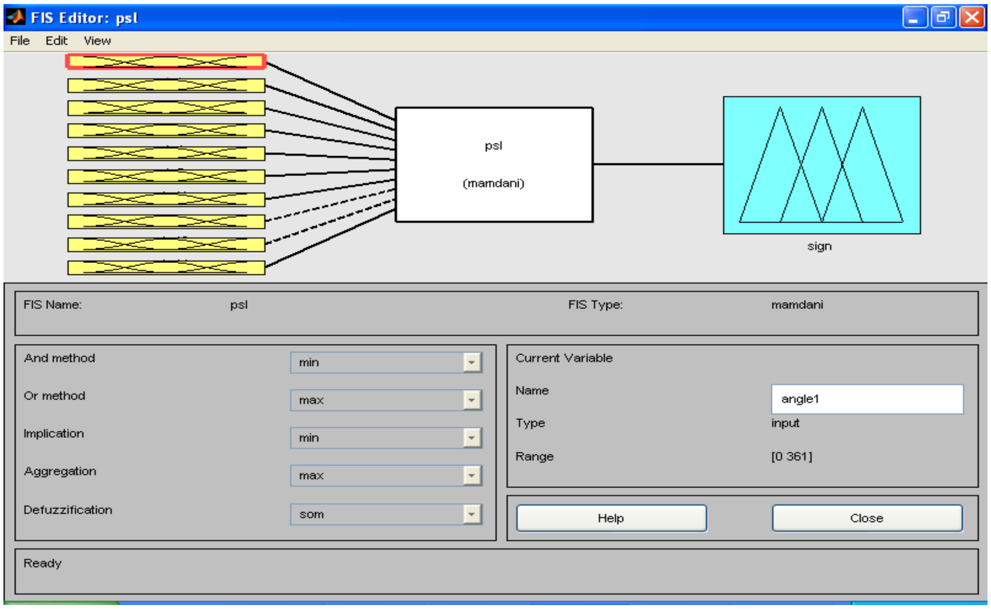


Figure 5. 1: FIS editor for PSL recognizer

Membership functions for the FIS can also be customized using fuzzy logic toolbox for input variables as well as for output variables of the FIS. Shape and other parameters for the variables can be customized according to the system. Membership function editor for one of the input variables is shown in Figure 5.2 and for output variable is shown in Figure 5.3.

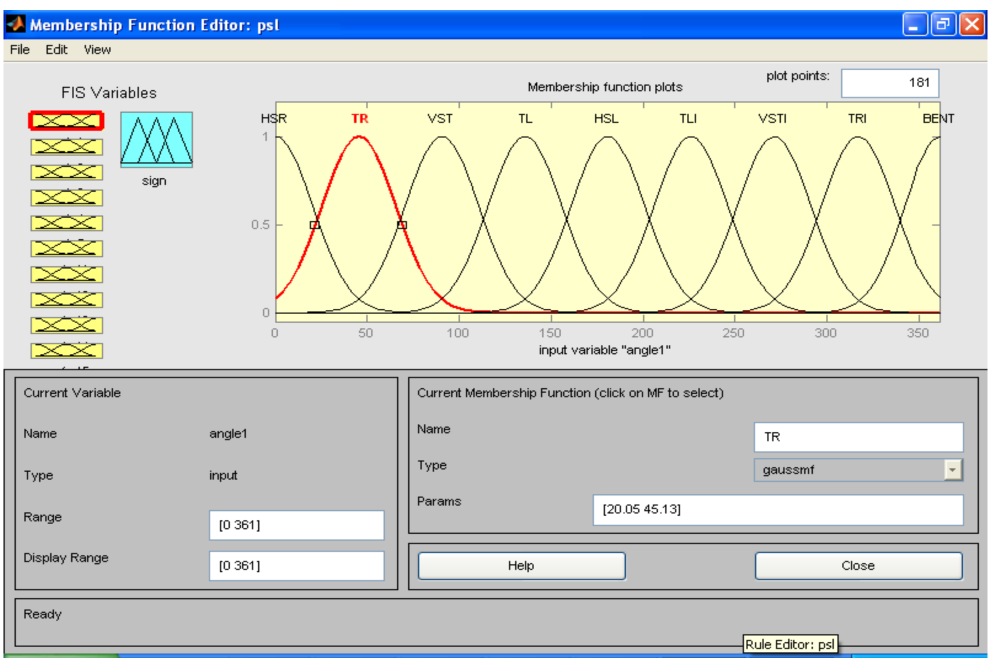


Figure 5. 2: Membership Function Editor for input variable

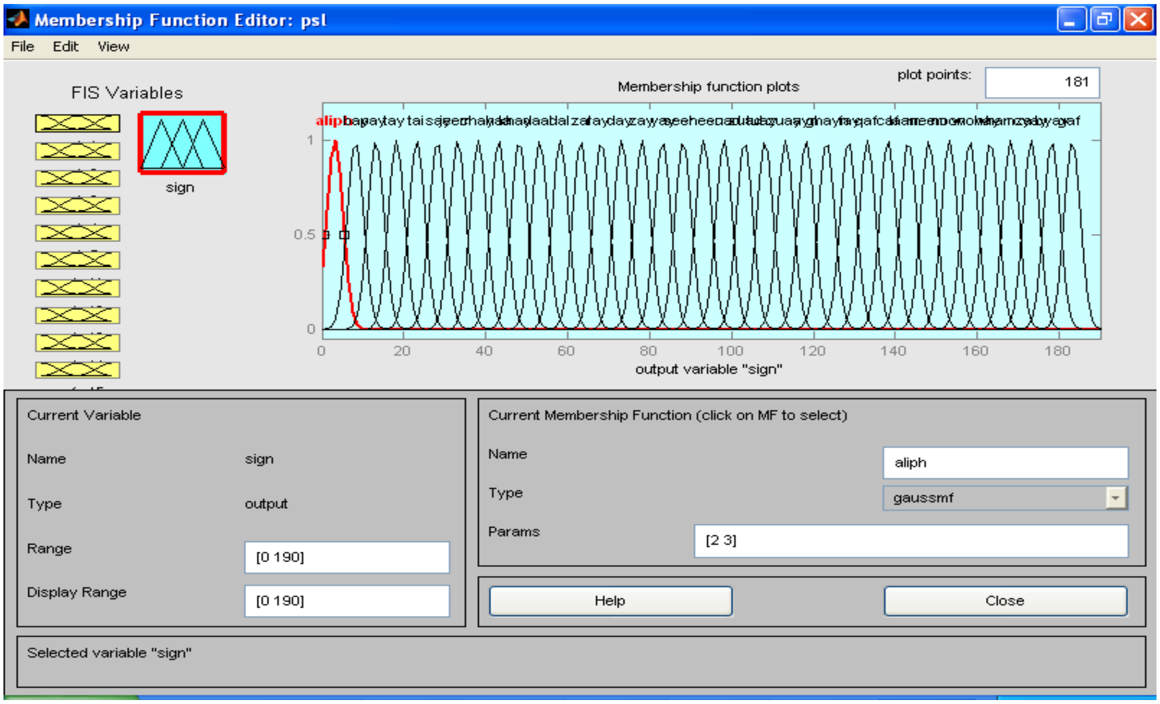


Figure 5. 3: Membership function editor for output variable

Once rules designed for a fuzzy classifier, these are very easy to feed into the toolbox and can be modified easily with a user friendly environment. Rule editor used for this thesis is shown in Figure 5.4.

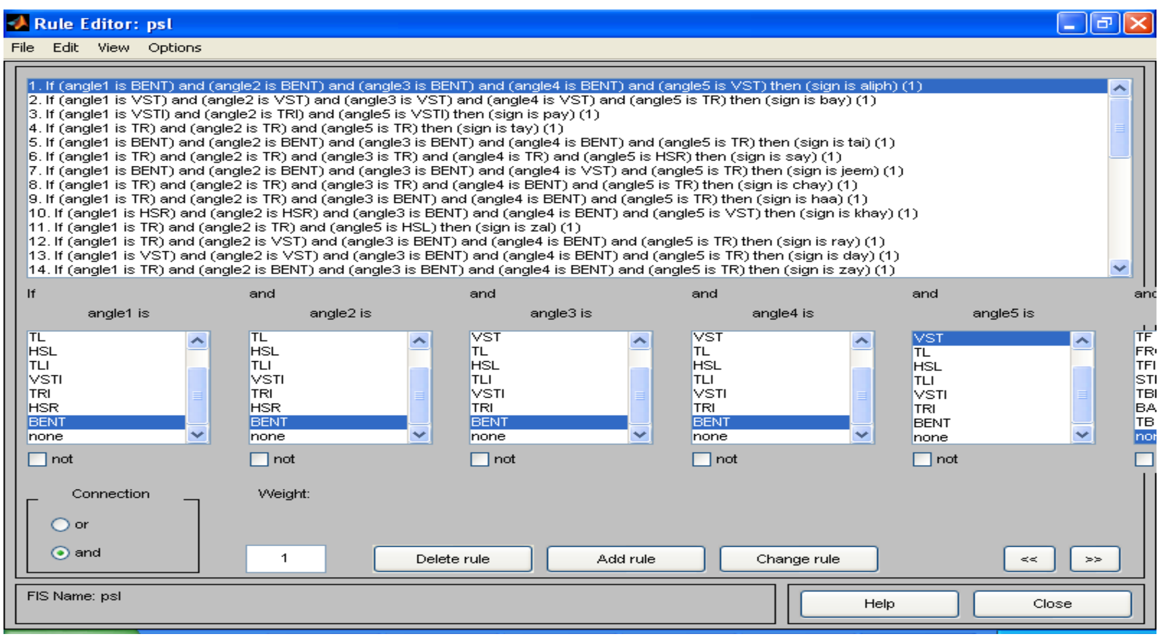


Figure 5. 4: Rule editor for PSL recognizer

For this thesis, fuzzy logic toolbox is integrated with source code written in sixteen other matlab files. These matlab files have been written to take input image and for calculation of input parameters required by the toolbox, toolbox returns its output again to a matlab source code file. Even without this integration this fuzzy logic toolbox can be used for testing rule base as it provide a rule viewer to analyze rules. Rule viewer for this thesis is shown in Figure 5.5.

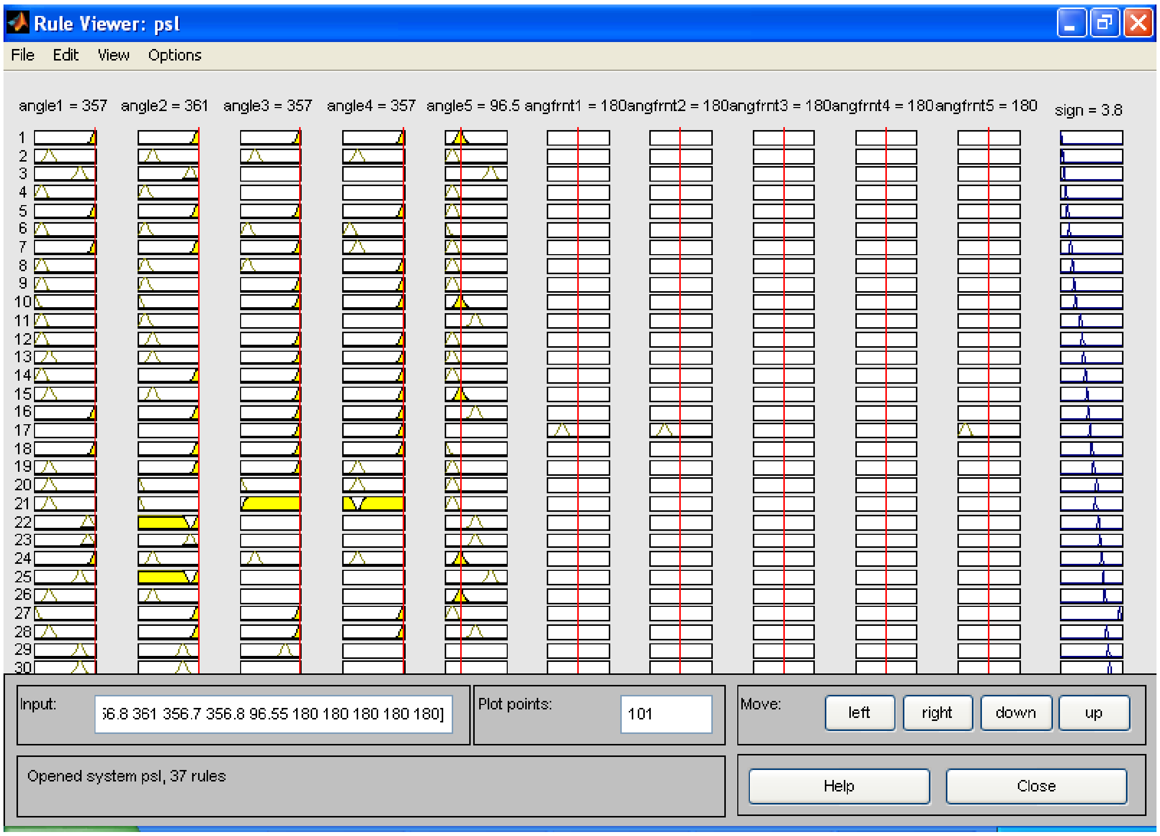


Figure 5. 5: Rule Viewer for PSL recognizer

5.2.2 Dictionary

Dictionary of signs used for experiments conducted to recognize PSL consist of only alphabets of PSL. There are thirty seven alphabets in PSL, all are static signs. Another important characteristic is that signs of all alphabets are right-handed gestures. Alphabets of PSL are shown in Figure 5.6.

Alphabets of PSL have different orientation of hands. All fingers of right hands are involved in signs. Fingers can have a variety of positions to adopt to show different sign. Palm is not always towards the viewer as it is the case in alphabets of many other sign languages. Signs of some alphabets are very similar to each other e.g. “Aliph” “seen” and “Tai”, “wow” and “choty yay”, “ain” and “ghain”, “dal” and “zal”, “meem” and “noon”, “day” and “caf”, “gaf” and “zay”. Some of the signs also required depth information, which is not possible to get in 2-D images, e.g. “sheen”, “hamza” and “hay”. All these characteristics of alphabets of PSL make the recognition task difficult. Fuzzy classifier has been used for this thesis, where rules have to be specified very carefully to give high accuracy. Making rule base for such a dictionary of signs was quite challenging, a lot of effort has been put to cater for all above mentioned problems. Hard work put in design of rule base lead to very high accuracy rates for classifier.

Figure 5. 6: Alphabets of PSL

Though dictionary of signs used for this thesis is not very large in number but this small subset of PSL have shown a lot of variety and hence can be a good benchmark to check the performance of classification for sign language.

5.3 Results and Analysis

Results of experiments are analyzed to have a look at the performance of any system. Results for PSL recognizer have been analyzed in detail in this section. A detailed analysis has been carried out with different views for PSL recognizer.

Experiments have been conducted on the above mentioned dictionary of PSL. Total 37 signs of PSL have been tested. Seven different people have performed signs of 37 alphabets hence total 259 signs have been used in experiments. Performers of the signs been ensured that sign should be accurately shown i.e. positions of the fingers, orientation of the hand and view of the palm should be according to the standard signs. Small variations in these parameters can be catered for by the inherent flexibility of the technique used. Out of these 37 signs only three signs could not be recognized with the proposed technique. Rests of the 34 signs are successfully recognized. Hence system achieved accuracy rate which can be considered as a very high rate while keeping in view the large variety dictionary possess.

All of the signs except two have been recognized successfully. Though signs, of alphabets of PSL, have variety of hand orientations, finger positions and visual similarities but even then PSL recognizer quite accurately performed. The results of PSL recognizer are summarized in Table 5.1.

Sign	**Visually Similar Signs**	**Recognition**
Aliph	Tai, seen, wow, choty yay, suad	Yes
Bay	-	Yes
Pay	Qaf	Yes
Tay	Haa, hay	Yes
Tai	Seen, aliph,wow, choty yay, suad	Yes
Say	-	Yes
Jeem	Bari yay	Yes
Chay	-	Yes
Haa	Hay, tay	Yes
Khay	-	Yes
Dal	Zal	Yes
Daal	Tuay, zuay	Yes
Zal	Dal	Yes
Ray	Day, caf	Yes
Day	Ray, caf	Yes
Zay	Gaf, hamza	Yes
Yay	-	Yes
Seen	Aliph, tai, suad, wow, choty yay	Yes
Sheen	-	Yes
Suad	Aliph, tai, seen, wow, choty yay	Yes
Zuad	-	Yes
Tuay	Daal, zuay	Yes
Zuay	Daal, tuay	Yes
Ain	Ghain	Yes

Ghain	Ain	Yes
Fay	-	Yes
Qaf	Pay	Yes
Caf	Ray, day	Yes
Gaf	Zay, hamza	Yes
Laam	-	Yes
Meem	Noon	Yes
Noon	Meem	Yes
Wow	Aliph, tai, seen, suad, choty yay	No
Hay	Haa, tay	No
Hamza	Gaf, zay	Yes
Choty yay	Aliph, tai, seen, suad, wow	No
Bari yay	Jeem	yes

Table 5. 1: Summary of results of PSL recognizer

For analysis of the results, groups of signs of PSL which are quite similar to each other, are discussed. These groups of signs are “aliph”, ”seen” and “tai”, “ain” and “ghain”, “dal” and “zal”, “meem” and “noon”, “day” and “caf”, “gaf” and “zay”.

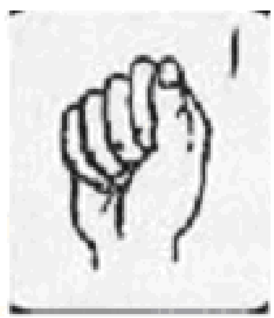


Figure 5. 7: Sign of "Aliph"

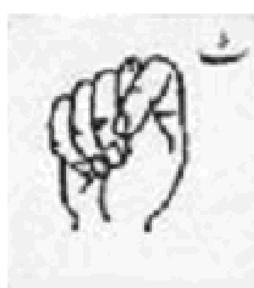


Figure 5. 8: Sign of "tai"

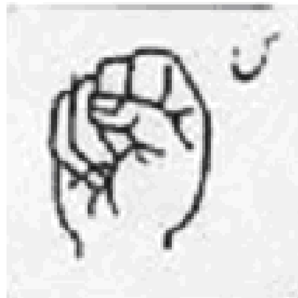


Figure 5. 9: Sign of "seen"

Signs of “aliph”, “tai” and “seen” are shown in Figure 5.7, 5.8 and 5.9 respectively. These three signs are quite similar to each other. Angles between finger-tips and corresponding finger-joints are exactly same for four fingers in all three signs. But thumb position is different in this group of signs. Normally such signs with minor differences are not distinguishable with classical classifiers. But with the PSL recognizer presented in this thesis, all these three signs have been accurately recognized. Distinction between signs in this group is possible due to the set of feature extracted in this technique. Position of four fingers in all these signs is BENT. The distinction is on the basis of thumb position i.e. VST (Vertical Straight) in “aliph”, TL (Tilted Left) in “tai” and HSL (Horizontal Straight Left). On the basis of these distinctive positions of thumb catered for in the recognizer, these signs have been correctly recognized.

Another pair of signs is very similar to each other and could be difficult to recognize with traditional classifiers. This pair consists of signs of “ain” and “ghain”.

Figure 5. 10: Sign of "ain"

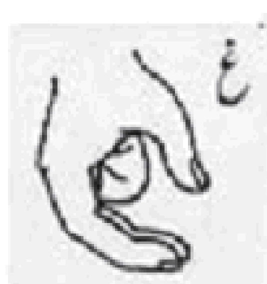


Figure 5. 11: Sign of "ghain"

Signs of “ain” and “ghain”, shown in Figure 5.10 and 5.11 respectively, are quite similar visually and very difficult for correct automatic recognition. Technique proposed in this thesis has shown its capablity of recognizing these two similar looking signs successfully. Positions of three fingers and thumb are exactly same for both signs. Thumb is TRI (Tilted Right Inverted) in both signs, index finger is TRI in both signs, little finger and ring finger are not visible in both signs the only difference between two signs is the position of middle finger. Middle finger is TRI for “ghain” whereas its not visible in sign for “ain”. This minor difference which could be confusing have not created any problem with the PSL recognizer for this thesis. These two signs have been successfully recognized by the technique presented in this thesis.

Another pair of signs in alphabets of PSL, which are very similar looking, is “dal” and “zal”.

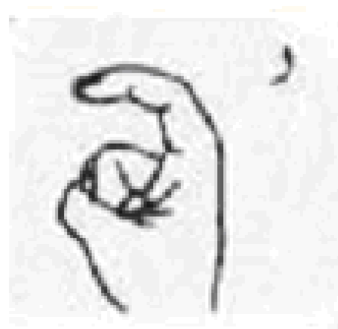


Figure 5. 12: Sign of "dal"

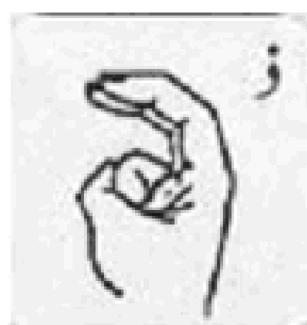


Figure 5. 13: Sign of "zal"

Figure 5.12 and Figure 5.13 have shown sign of “dal” and “zal”, which are very similar to each other. But these two signs are recognized quite accurately by the technique proposed in this thesis. Little finger and index finger are not visible in both the signs. Thumb has same position in both signs that is HSL. Index finger also has TL position for both the signs. Only distinguishable feature of the two signs is position of middle finger which is not visible in sign “dal” and it is TL for the sign of “zal”. PSL recognizer in this thesis has shown accurate recognition results for this pair of signs as well.

“meem” and “noon” are two other signs which are quite similar to each other and hence difficult to classify. Signs of these two alphabets are shown in Figure 5.14 and 5.15.

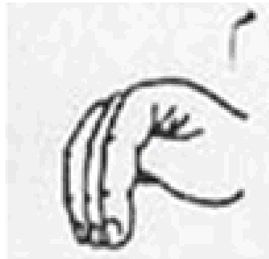


Figure 5. 14: Sign of "meem"

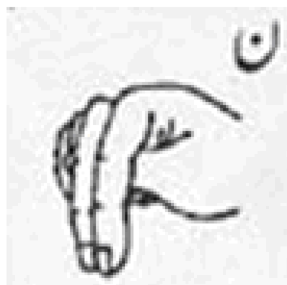


Figure 5. 15: Sign of "noon"

Sign of “meem” and “noon” are very similar to each other. The only difference between two signs is the position of ring finger. It is not visible in sign of “noon” whereas it is VSTI for sign of “meem”. Positions for the rest of the fingers in both the signs are exactly same. But even for this level of similarity among signs, PSL recognizer performed successfully.

Another pair, which is similar enough to be recognized incorrectly by the traditional classifiers, but is correctly recognized by the PSL recognizer for this thesis. This pair contains signs of “day” and “caf” shown in Figure 5.16 and 5.17 respectively.

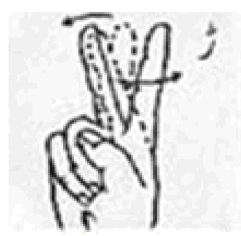


Figure 5. 16: Sign of "day"

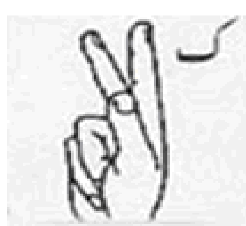


Figure 5. 17: Sign of "caf"

The signs of “day” and “caf” as shown in Figures 5.16 and 5.17 are very similar. Little and ring finger are BENT in both the signs. Index finger and middle finger are VST in both the signs of “day” and “caf”. The only distinction between the two signs is the position of the thumb which is VST in case of “caf” and TL for the sign of “day”. On the basis of this difference in thumb position, PSL recognizer successfully recognized these two signs.

Signs of “gaf” and “zay” also possess enough similarity in their positions of fingers to be classified incorrectly. Signs of “zay” and “gaf” shown in Figure 5.18 and 5.19 respectively.

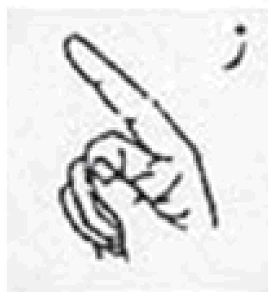


Figure 5. 18: Sign of "zay"

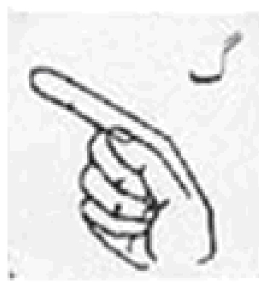


Figure 5. 19: Sign of "gaf"

Signs for “zay” and “gaf” have got very similar positions for the fingers. The only difference between two signs is the visibility of the middle finger. Middle finger is not visible in sign for “zay” whereas it is BENT for the sign “gaf”. On the basis of such minute difference PSL recognizer is capable of correct recognition. This shows the strength of the recognizer.
The signs which could not be recognized by the PSL recognizer are “choty yay” “wow” and “hay”.

Figure 5. 20: Sign of "wow"

Figure 5. 21: Sign of "choty yay"

Signs of “wow” and “choti yay” are very similar that’s why recognition of these signs was not correct in PSL recognizer. Features taken by classifier are only angles between finger-tip and corresponding finger-joint for all the five fingers. As it is evident from the Figure 5.7 and 5.8, that features selected are not enough for distinction between these two signs. Angles between finger-tip and their corresponding finger-joint of four fingers in both the signs are exactly same the only slight difference is in the position of the thumb. But this difference is so minute that cannot be identified with the PSL recognizer because angle between thumb-joint and thumb-tip in both the signs is about 45 degree. So to differentiate between these two signs, it is required to introduce some other features such as position of thumb relevant to other fingers that is above, below, right or left. But with the current features selected for recognition, distinction between these two signs is not possible.

Table 5.2 presents the analysis based on the visibility of the fingers in different signs of PSL alphabets.

Finger	**No of signs with visibility of the finger**
Index finger	37
Middle finger	37
Ring finger	33
Little finger	29
Thumb	35

Table 5. 2: Finger usage in sign recognition

Table 5.2 shows that index finger and middle finger is used in every sign whereas thumb is not visible in only two signs. Little and ring finger though used in lesser number of signs but that number is not that small to ignore these fingers while considering features. So it is required that positions of all of the fingers should be used in sign recognition and this is what done in the PSL recognizer.

It is an important consideration to select appropriate no of features to recognize signs. Positions of fingers are taken as the main features for classification in PSL recognition. Table 5.3 shows the analysis for different set of features i.e. positions of different fingers.

No. of Features	Feature set	No. of recognized sign
1	Index finger	0
1	Middle finger	0
1	Ring finger	0
1	Little finger	0
1	Thumb	0
2	Index finger, Middle finger	1
2	Index finger, Ring finger	0
2	Index finger, little finger	0
2	Middle finger, Ring finger	0
2	Middle finger, thumb	0
2	Ring finger, little finger	0
2	Ring finger, thumb	0
2	Little finger, thumb	0
2	Middle finger, little finger	0
2	Index finger, thumb	0
3	Index finger, Middle finger, Ring finger	1
3	Index finger, Middle finger, little finger	0
3	Index finger, Middle finger, thumb	6
3	Index finger, Ring finger, little finger	0
3	Index finger, Ring finger, thumb	0
3	Middle finger, Ring finger, little finger	0
3	Middle finger, Ring finger, thumb	0
3	Ring finger, Little finger, thumb	0
3	Index finger, little finger, thumb	0
4	Index finger, Middle finger, Ring finger, Little finger	0
4	Index finger, Middle finger, Ring finger, thumb	0
4	Index finger, Middle finger, little finger, thumb	0
4	Index finger, Ring finger, little finger, thumb	0
4	Middle finger, Ring finger, little finger, thumb	0

5	Index finger, Middle finger, Ring finger, little finger, thumb	29

Table 5. 3: Feature set analysis

Table 5.3 shows that all the five fingers should be considered while extracting features for sign language because majority of the signs require position of all the five fingers.

Chapter 6: Conclusions & Future Work

6.1. Conclusion

Automatic sign language recognition can be of great significance for communication with deaf people. Sign language recognition has also its application in virtual reality, human computer interaction, machine control in the industrial field. Researchers have proposed a large number of methods to recognize hand signs. Different techniques of automatic recognition for sign languages of different countries have been presented by many researchers. PSL is however ignored; only one research has been conducted for automatic recognition of a subset of PSL using template matching. In this thesis an automatic sign language recognizer has been presented to recognize alphabets of PSL. All alphabets are static right handed signs.

Segmentation is considered to be a very important phase in the domain of pattern classification. Different researchers proposed different methods for segmentation of hands for variety of applications. Broadly there are two segmentation techniques: Shape-based Segmentation, Color-based Segmentation. For this thesis firstly skin-based segmentation technique, a Color-based segmentation technique, was adopted to segment human hands but later on in the phase of feature extraction it was observed that the results obtained from this technique are putting many constraints on feature selection. So at that point in time this segmentation technique was replaced with another Color-based technique. The later Color-based technique uses color gloves for segmentation of human hands.

After segmentation another very important phase in sign recognition is feature extraction. Some commonly used features for hand gestures or sign language recognition have been discussed. These features are analyzed according to three properties of good features: to reduce the cost of feature extraction, to improve the classification accuracy, to increase invariance of features. A new set of features, which have been used by the PSL recognizer of this thesis, is presented. Features used by the PSL recognizer are angles between finger-tip and corresponding finger-joint of all the five fingers.

Four most commonly used classifiers for hand sign recognition have been discussed along with their pros, cons and suitability for different applications. Four classifiers: template matching, ANN, SVM and fuzzy classifier, have their own different techniques to classify signs. Classification technique used by each classifier has different computation cost and memory requirements. This determines their speed of classification. Accuracy rate of some classifiers is also dependent upon the size of dictionary of signs or gestures to be classifier. All of the four classifiers have their advantages that are discussed in chapter 4, but looking at all factors involved i.e. computation cost, speed, accuracy, training overheads, memory requirements and size of dictionary, the most suitable classifier for most of the applications is fuzzy classifier. So, the fuzzy classifier has been selected for the PSL recognizer for this thesis. Marked color glove has been used to segment hand and then these marks are also used to extract features to be used by the classifier to recognize sign. The only feature has been used by the FIS is angle between the finger-tip and finger-joint of the fingers. These angles have the capability to uniquely represent each sign. Results have shown that technique presented in this thesis is capable of recognizing a large dictionary of signs without any prior training and with very low computation requirements.

6.2. Future work

Looking at the potential of the system, it can be further developed for multidimensional objectives and applications. The method presented in this thesis only took static signs with right hand.

In future, PSL recognizer can also be modified for two handed static sign recognition systems. Further features would be required to be added to the system. Rules would also require addition to cater for new signs. Computation cost and memory utilization is very low for the PSL recognizer, so have the capability to incorporate much bigger dictionary with additional features.

PSL recognizer have very low computation cost. Its low computation process makes it a very good choice for recognizing continuous signs. This technique can be utilized for each frame and then classifier can be further modified to check inter-frame relation to recognize continuous signs.

PSL recognizer has very low computation requirements. So can be a very good candidate for real time systems. Online deaf communication systems, real-time gesture recognition systems and time-sensitive HCI systems and many other time-critical systems can use technique presented in this thesis.

References

[1] McGuire, R.M., Hernandez-Rebollar, J., Starner, T.,Henderson, V., Brashear, H., and Ross, D.S.: Towards a One-Way American Sign Language Translator. In: Proc. Intl Conf. Automatic Face and Gesture Recognition, pp. 620-625, 2004.

[2] Akyol, S. and Canzler, U.: An Information Terminal Using Vision Based Sign Language Recognition. In: Proc. ITEA Workshop Virtual Home Environments, pp. 61-68, 2002.

[3] Kennaway, R.: Experience with and Requirements for a Gesture Description Language for Synthetic Animation. In: Proc. Gesture Workshop, pp. 300-311, 2003.

[4] Koizumi, A., Sagawa, H., and Takeuchi, M.: An Annotated Japanese Sign Language Corpus. In: Proc. Intl. Conf. Language Resources and Evaluation, vol. III, pp. 927-930, 2002.

[5] Sub Yoon, Jung Soh, Younglae J. Bae and Hyun: Hand gesture recognition using combined features of location, angle and velocity. In: Science Direct, Pattern Recognition, Volume 34, Issue 7, 2001, Pages 1491-1501

[6]. Xiaoming Yin,and Ming Xie: Finger identification and hand posture recognition for human–robot interaction. In: Image and Vision Computing, Volume 25, Issue 8, 1 August 2007, Pages 1291-1300

[7] Rafael Muñoz-Salinas,R. Medina-Carnicer, F.J. Madrid-Cuevas and A. Carmona-Poyato: Depth silhouettes for gesture recognition. In: Pattern Recognition Letters, Volume 29, Issue 3, 1 February 2008, Pages 319-329

[8] Qiulei Dong, Yihong Wu, and Zhanyi Hu, P.J. Narayanan et al. (Eds.): Gesture Recognition Using Quadratic Curves. In: ACCV 2006, LNCS 3851, pp. 817–825, 2006.

[9] O. Al-Jarah, A. Shatnawi, and A.H. Halawani: Recognition of Gestures in Arabic Sign Language using Neural Networks. In: ACTA press, Artificial intelligence and soft computing, 2006.

[10] Aleem Khalid Alvi, M. Yousuf Bin Azhar, Mehmood Usman, Suleman Mumtaz, Sameer Rafiq, Razi Ur Rehman, Israr Ahmed: Pakistan Sign Language Recognition Using Statistical Template Matching. In: Proceedings of World Academy Of Science, Engineering and Technology, Volume 3 January 2005 Issn 1307-6884

[11] Tanibata, N., Shimada N.: Extraction of hand features for recognition of sign language words. In: Proceedings of the 15th International Conference on Vision Interface, pp. 391–398. Calgary, Canada (2002)

[12] Rini Akmeliawati, Melanie Po-Leen Ooi, Ye Chow Kuang: Real-Time Malaysian Sign Language Translation using Colour Segmentation and Neural Network. In: IMTC 2007 - Instrumentation and Measurement Technology Conference Warsaw, Poland, 1-3 May 2007
[13] Eun-Jung Holden, Gareth Lee, and Robyn Owens: Automatic Recognition of Colloquial Australian Sign Language. In: Proceedings of the IEEE Workshop on Motion and Video Computing (WACV/MOTION'05)

[14] Qutaishat Munib, Moussa Habeeb, Bayan Takruri and Hiba Abed Al-Malik: American Sign Language (ASL) recognition based on Hough transform and neural networks. In: Expert Systems with Applications Volume 32, Issue 1, January 2007, Pages 24-37

[15] G.A. ten Holt, P. Hendriks, T.C. Andringa, E.A. Hendriks, M.J.T. Reinders: Automatic Recognition of Dutch Sign Language. In: Proceedings of the twelfth annual conference of the Advanced School for Computing and Imaging, Lommel, Belgium, June 14-16, 2006, published by ASCI, Delft, pp 157-164.

[16] Chen, Y., et al., CSLDS: Chinese Sign Language Dialog System, in Proceedingsof the IEEE International Workshop on Analysis and Modeling of Faces and G.A. ten Holt – Automatic Recognition of Dutch Sign Language 8 Gestures AMFG'03. 2003, IEEE Computer Society: Nice, France. p. 236-238.

[17] G. Somers, R.N. Whyte**:** Hand posture matching for Irish Sign language interpretation. In: Proceedings of the 1st international symposium on Information and communication technologies, ACM International Conference Proceeding Series; Vol. 49

[18] Ulrike Zeshan: Sign Language of Indo-Pakistan: A description of a Signed Language. In: Philadelphia, Amsterdam, John Benjamins Publishing Co. (2000)

[19] Woodward, J (1993): The relationship of sign language varieties in India, Pakistan and Nepal. In: Sign Language Studies (78), 15–22, (1993).

[20] www.answers.com/topic/segmentation

[21] Xiaoming Yin, Ming Xie: Finger identification and hand posture recognition for human–robot interaction. In: Image and Vision Computing**,** Volume 25, Issue 8, 1, Pages 1291-1300, August 2007.

[22] Douglas Chai and King N. Ngan: Face Segmentation Using Skin-Color Map in Videophone Applications. In: IEEE transactions on circuits and systems for video technology, VOL. 9, NO. 4, June 1999.

[23] Stan Sclaroff, Vassilis Athitsos, Leonid Sigal: Skin Color-Based Video Segmentation under Time-Varying Illumination. In: IEEE Transaction on Pattern Analysis and Machine Intelligence Vol. 26, No. 7 pp. 862-877, July 2004.

[24] M. Fleck, D. Forsyth, and C. Bregler: Finding naked people. In: Proc. ECCV'96, pp. 592-602, April 1996.

[25] http://en.wikipedia.org

[26] Peter Gejgus, Jaroslav Placek, Martin Sperka: Skin color segmentation method based on mixture of Gaussians and its application in Learning System for Finger Alphabet. In: International Conference on Computer Systems and Technologies - CompSysTech'2004

[27] Kong Xiao, Liu Donghui and Shen Lansum: Segmentation of Skin Color Regions Based on Fuzzy Cluster. In: International Symposium on Intelligent Multimedia, Video and Speech Processing, October 2004, Hong Kong

[28] Hichem Sahbi and Nozha Boujemaa: Accurate Face Detection Based on Coarse Segmentation and Fine Skin Color Adaption. In: M. Tistarelli, J. Bigun, A.K. Jain (Eds.): Biometric Authentication, LNCS 2359, pp. 112–120, 2002.

[29] Son Lam Phung, Abdesselam Bouzerdoum, and Douglas Chai: Skin Segmentation Using Color and Edge Information. In: Symposium on Signal Processing and its Applications, 1-4 July 2003, Paris, France.

[30] Aamer .S.S.Mohamed, Ying Weng, Stan S Ipson, Jianmin Jiang: Face Detection based on Skin Color in Image by Neural Networks.

[31] Chiho Kim, Bum-Jae You , Mun-Ho Jeong and Hagbae Kim: Color segmentation robust to brightness variations by using B-spline curve modeling. In: Pattern Recognition Volume 41, Issue 1, January 2008, Pages 22-37

[32] V. I. Pavlovic, R. Sharma, and T. S. Huang: Visual interpretation of hand gestures for human-computer interaction: a review. In: IEEE Trans. Pattern Anal. Machine Intell., vol. 19, no. 7, pp. 677–695, 1997.

[33] S. S. Fels and G. E. Hinton: Glove-talk: a neural network interface between a data-glove and a speech synthesizer. In: IEEE Trans. Neural Networks, vol. 4, no. 1, pp. 2–8, 1993.

[34] D. J. Sturman and D. Zeltzer: A survey of glove-based input. In: IEEE Comput. Graph. Appl., vol. 14, no. 1, pp. 30–39, 1994.

[35] D. L. Quam: Gesture recognition with a dataglove. In: Proc.IEEE National Aerospace and Electronics Conference (NAECON'90), vol. 2, pp. 755–760, Dayton, Ohio, USA,May 1990.

[36] J. Eisenstein, S. Ghandeharizadeh, L. Huang, C. Shahabi, G. Shanbhag, and R. Zimmermann: Analysis of clustering techniques to detect hand signs. In: Proc. International Symposiumon IntelligentMultimedia, Video and Speech Processing (ISIMP'01), pp. 259–262, Hong Kong, China, May 2001.

[37] M. AL-Rousan and M. Hussain: Automatic recognition of Arabic sign language finger spelling. In: International Journal of Computers and Their Applications, vol. 8, no. 2, pp. 80–88, 2001, Special Issue on Fuzzy Systems.

[38] J. Davis and M. Shah: Gesture recognition. In: Tech. Rep. CSTR-93-11, Department of Computer Science, University of Central Florida, Orlando, Fla, USA, 1993.

[39] J. Davis and M. Shah: Visual gesture recognition. In: IEE Proceedings: Vision, Image and Signal Processing, vol. 141, no. 2, pp. 101–106, 1994.

[40] Y. Wu and T. S. Huang: Vision-based gesture recognition: a review. In: Proc. 3rd International GestureWorkshop (GW '99), pp. 103–115, Gif-sur-Yvette, France, March 1999.

[41] http://en.wikipedia.org/wiki/Feature_extraction

[42] Qutaishat Munib, Moussa Habeeb, Bayan Takrur and Hiba Abed Al-Malik: American sign language (ASL) recognition based on Hough transform and neural networks. In: Expert Systems with ApplicationsVolume 32, Issue 1, January 2007, Pages 24-37

[43] Yasushi Hamada, Nobutaka Shimada, Yoshiaki Shirai: Hand Shape Estimation under Complex Backgrounds for Sign Language Recognition. In: Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR'04)

[44] Ales Jaklic, Dora Vodopivec, Vito Komac: Learning Sign Language through Multimedia. In: Proceedings of the International Conference on Multimedia Computing and Systems, May 1995, IEEE, pp. 282-285

[45] http://en.wikipedia.org/wiki/Edge_detection

[46] http://en.wikipedia.org/wiki/Silhouette

[47] http://www.answers.com/topic/silhouette

[48] Kazuhiko Takahashi: Remarks on Human Body Posture Estimation From Silhouette Image. In: IEEJ Transaction of Electronics, Information and Systems, Vol. 125 (2005), No. 4 pp. 675-676

[49] Chin-Seng Chua, Haiying Guan and Yeong-Khing Ho: Model-based 3D hand posture estimation from a single 2D image. In: Image and Vision Computing, Volume 20, Issue 3, 1 March 2002, Pages 191-202

[50] Rafael Muñoz-Salinas, R. Medina-Carnicer, F.J. Madrid-Cuevas and A. Carmona-Poyato: Depth silhouettes for gesture recognition. In: Pattern Recognition Letters, Volume 29, Issue 3, 1 February 2008, Pages 319-329

[51] Yea-Hong Yoon and Kang-Hyun Jo: Hands Shape Recogniton Using Moment Invariant For the Korean Sign Language Recognition. In: Proceedings of the 7th Korea-Russia [niernationol Symposium. KORUS 2003

[52] Bryan S. Morse: Lecture on Shape Description (regions). In: Brigham Young University.

[53] Noor Saliza Mohd Salleh, Jamilin Jais, Lucyantie Mazalan, Roslan Ismail, Salman Yussof, Azhana Ahmad, Adzly Anuar, Dzulkifli Mohamad: Sign Language to Voice Recognition: Hand Detection Techniques for Vision-Based Approach. In: Current Developments in Technology-Assisted Education (2006)

[54] Mikhail Lange, Sergey Ganebnykh and Andrey Lange: Moment-Based Pattern Representation Using Shape and Grayscale Features. In: Lecture Notes in Computer Science, Volume 4477/2007, 2007.

[55] Khaled Assaleh, M. Al-Rousan: Recognition of Arabic Sign Language Alphabet Using Polynomial Classifiers. In: EURASIP Journal on Applied Signal Processing 2005:13, 2136–2145

[56] Kenji Oka, Yoichi Sato, Hideki Koike: Real-Time Finger Tracking and Gesture Recognition. In: IEEE Computer Graphics and Applications,Volume 22 , Issue 6 (November 2002) ,Pages: 64 - 71

[57] B.R.C. Bedregal, G.P. Dimuro, A.C.R. Costa: Hand Gesture Recognition in an Interval Fuzzy Approach. In: TEMA Tend. Mat. Apl. Comput., 8, No. 1 (2007), 21-31.

[58] http://en.wikipedia.org/wiki/Artificial_intelligence

[59] www.socher.org/index.php/Main/SummaryOfStatisticalMachineLearning

[60] http://en.wikipedia.org/wiki/Pattern_recognition

[61] Neural Networks, research report by Christos Stergiou and Dimitrios Siganos

[62] O. Al-Jarah, A. Shatnawi, and A.H. Halawani: Recognition of Gestures in Arabic Sign Language using Neural Networks. In: ACTA press, Artificial intelligence and soft computing, 2006.

[63] Xiaoming Yin and Ming Xie: Finger identification and hand posture recognition for human–robot interaction. In: Image and Vision Computing ,Volume 25, Issue 8, Pages 1291-1300, 1 August 2007

[64] http://en.wikipedia.org/wiki/Template_matching

[65] Takanao Inaguma, Koutarou Oomura, Hitoshi Saji, and Hiromasa Nakatani: Efficient Search Technique for Hand Gesture Tracking in Three Dimensions. In: BMCV 2000, LNCS 1811, pp. 594–601, 2000

[66] Aleem Khalid Alvi, M. Yousuf Bin Azhar, Mehmood Usman, Suleman Mumtaz, Sameer Rafiq, Razi Ur Rehman, Israr Ahmed : Pakistan Sign Language Recognition Using Statistical Template Matching. In: Proceedings of world academy of science, engineering and technology volume 3 January 2005 issn 1307-6884

[67] Tanibata, N., Shimada, N.: Extraction of hand features for recognition of sign language words. In: Proceedings of the 15th International Conference on Vision Interface, pp. 391–398. Calgary, Canada (2002)

[68] http://en.wikipedia.org/wiki/Support_vector_machine

[69] http://www.dtreg.com/svm.htm

[70] Rafael Muñoz-Salinas, R. Medina-Carnicer, F.J. Madrid-Cuevas and A. Carmona-Poyato: Depth silhouettes for gesture recognition. In: Pattern Recognition Letters, Volume 29, Issue 3, 1 February 2008, Pages 319-329.

[71] T. Ko, D. Demirdjian and T. Darrell**:** Untethered Gesture Acquisition and Recognition for a Multimodal Conversational System. In: ICMI’03, November 5–7, 2003, Vancouver, British Columbia, Canada.

[72] http://www.svms.org

[73] Martin Hellmann: Fuzzy Logic Introduction. Report on internet, March 2001.

[74] L-Xin Wang: A Course in Fuzzy Systems and Control. Publisher: Prentice Hall, ISBN: 0-13-540882-2

[75]http://www.mathworks.com/access/helpdesk/help/toolbox/fuzzy/index.html?/access/helpdesk/help/toolbox fuzzy/

[76] Mamdani, E.H. and S. Assilian: An experiment in linguistic synthesis with a fuzzy logic controller. In: International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975.

[77] Zadeh, L.A.: Outline of a new approach to the analysis of complex systems and decision processes. In: IEEE Transactions on Systems, Man, and Cybernetics, Vol. 3, No. 1, pp. 28-44, Jan. 1973.