Design of ANFIS System for Recognition of Single Hand and Two Hand Signs for Indian Sign Language

Shweta Dour
K J Somaiya College of Engineering ,
University of Mumbai

J.M.Kundargi
K J Somaiya College of Engineering ,
University of Mumbai

ABSTRACT

Sign language develops independently from the spoken language of the region . The sign language used in India is commonly known as Indian Sign Language (ISL). A functioning sign language recognition system can provide an opportunity for a deaf/mute person to communicate with non-signing people without the need for an interpreter. Our system deals with images of bare hands, which allows the user to interact with the system in a natural way. In doing so, we have designed a collection of ANFIS networks, each of which is trained to recognize one sign gesture. Features of the input gesture of the sign are extracted obtaining feature vector. The recognition algorithm translates each quantitative value of the feature into fuzzy sets of linguistic terms using membership functions. The membership functions are formed by the fuzzy partitioning of the feature space into fuzzy equivalence classes, using the feature cluster centers generated by the subtractive clustering technique. The subtractive clustering algorithm and the least-squares estimator are used to identify the fuzzy inference system, and the training is achieved using the hybrid learning algorithm.

General Terms

Sign Language , Hand Gestures , Sign language recognition system .

Keywords

Indian Sign Language (ISL), Adaptive Neuro Fuzzy Inference System(ANFIS) , Subtractive Clustering.

1. Introduction

Sign languages are not universal. There is no internationally recognized and standardized sign language for all deaf people. As is the case in spoken language, every country has got its own sign language with high degree of grammatical variations. The sign language used in India is commonly known as Indian Sign Language (henceforth called ISL). ISL produces static as well as dynamic gestures respectively and to recognize them needs to solve several kinds of challenges like two handed signs , sometimes both hands moving, sometimes one hand is moving fast another slow, different hand shapes, contacting the body etc. Most of ISL gestures being considered here are alphabets of the ISL.

Linguistic studies on ISL were started around 1978 and it has been found that ISL is a complete natural language, instigated in India, having its own morphology, phonology, syntax, and grammar . The research on ISL linguistics[1] and phonological studies get hindered due to lack of linguistically annotated and well documented ISL data . There is a limited number of hearing people who are competently able to communicate in sign language. Sign language interpreters can be used to aid communication between deaf and hearing people but this is often difficult due to the limited availability and high cost of interpreters. These difficulties in communication between hearing and deaf people can lead to problems in the integration of deaf people into society and conflicts with a self-determined lifestyle for hearing impaired people. As it has been specified above, ISL is not only used by the deaf people but also by the hearing parents of the deaf children, the hearing children of deaf adults and hearing deaf educators . Therefore the need to build a system that can associate signs to the words of spoken language, and which can further be used to learn ISL, is significant.

Indian Sign Language (ISL) is a visual-spatial language [2] which provides linguistic information using hands, arms, face, and head/body postures. The signer often uses the 3D space around his body to describe an event . Unlike spoken languages where the communication medium is dependent on sound, in sign language, the communication medium depends upon the visual channel.

ISL signs can be generally classified[3] into three classes: One handed, two handed, and non-manual signs. Fig. 1.1 shows the overall Indian sign hierarchy.

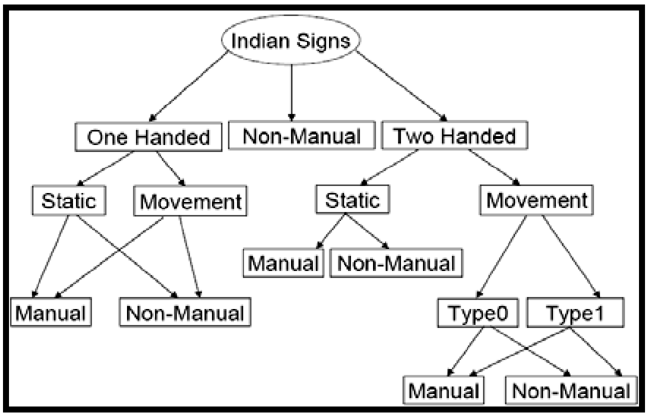


Fig. 1.1 ISL Type Hierarchy

One handed signs: the one handed signs are represented by a single dominating hand. One handed signs can be either static or movement related. Each of the static and movement signs is further classified into manual and non-manual signs.

Two hand signs: As in the case of one hand signs, similar classification can be applied to two handed signs. However, two handed signs with movements can be further distinguished as:

Type0: Signs where both hands are active

Type1: Signs where one hand (dominant) is more active compared to the other hand (non-dominant) .

2. Database Representations of Indian Sign Language

In order to approach the problem of translating signs in to text it is necessary to create a database of videos of different signs by multiple signers. Unlike American Sign Language or British sign language, Indian sign language does not have a standard database that is available for use. Hence we have created our own database of Indian signs in collaboration with Indian Deaf Society [4] .

The complete database representation for alphabets of Indian Sign Language is as shown in Figure 2.1 which shows alphabets A to Z for ISL wherein including signs of both single hand and two hand signs.

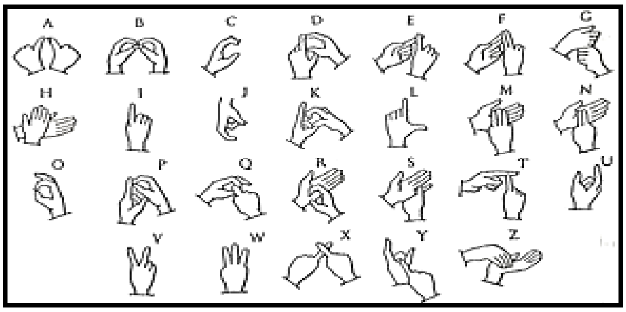


Figure 2.1 Database Representation for ISL

The complete database of all 26 alphabets of ISL created by us for our work is as shown in Figure 2.2.We have created and worked upon a total of 5 different datasets with different combinations of signs for the ISL . These datasets include datasets of signs having both single and two hand gesture , only single hand gestures , and also a dataset having only two hand gestures representing the alphabets of ISL created by multiple signers of Indian signs. The experimental setup consists of a light background and preferably the signer wearing a light colored clothes. This controlled environment reduces tracking and segmentation problems. The RGB videos are acquired using a i-ball C8.0 web camera at a resolution of 1024X768 pixels. Higher resolution incorporates longer delays in the video acquisition process and higher execution times. The videos are acquired under normal lighting conditions to simulate real-time environment. Natural lightning conditions are required to be present during the video acquisition process.

A total of 4 different signers volunteered where each signer is asked to repeat the sign twenty five times under different conditions such that we provide a total of 100 training samples of each alphabet. A video acquisition process is subjected to many environmental conditions such as position of the camera, distance from the camera, illumination or lightening conditions available and presence of noise.

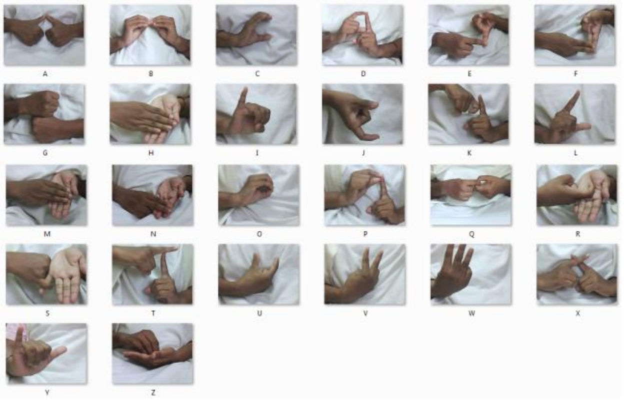


Fig. 2.2 Created Database for our work

3. Proposed Method for Recognition of Sign

In our work we have presented a system based on hand feature extraction in combination with a multi-layer fuzzy neural-network based classifier . The hand gesture area is separated from the background by using skin detection and segmentation method of skin color, then a contour of hand image can be used as a feature that describe the hand shape.

As such, the general process of the proposed method is composed of three main parts:-

1) A preprocessing step to focus on the gesture. This step is creating a skin-segmented binary image by using a threshold value of probability, if the probability of a pixel in skin likelihood image is more or equal to estimated threshold value, it supposed that this pixel represents skin color, if not it supposed that this pixel does not represent skin color. The skin color pixels are white and the other ones are black in the skin – segmented image.

2) A feature extraction step that use the hand contour of the gesture image ,The hand contour will act as the feature of the gesture. The feature extraction aspect of image analysis seeks to identify inherent characteristics, or features of objects found within an image. These characteristics are used to describe the object, or attribute of the object, prior to the subsequent task of classification. For posture recognition, (static hand gestures) features such as fingertips, finger directions and hand’s contours can be extracted. But such features are not always available due to self-occlusion and lighting conditions. Feature extraction is a complex problem, and often the whole image or transformed image is taken as input. Contour detection process consists of two steps: first find the edge response at all points in an image using gradient computation and in the second step modulate the edge response at a point by the response in its surround.

3) A classification step. The unknown gesture's feature will be produced and entered to the fuzzy neural network .The gesture recognition process diagram is illustrated in Figure 3.1,the hand region obtained after the pre-processing stage and it will be used as the primary input data for the feature extraction step of the gesture recognition algorithm.

Our classification process based on the use of single MLP, with hybrid training algorithm. In the feature extraction stage the hand contour is resized in order to make it appropriate for neural network input**,** then it entered to the classification stage. The recognition process consist of two phases , training and classification, as shown in the Figure 3.1

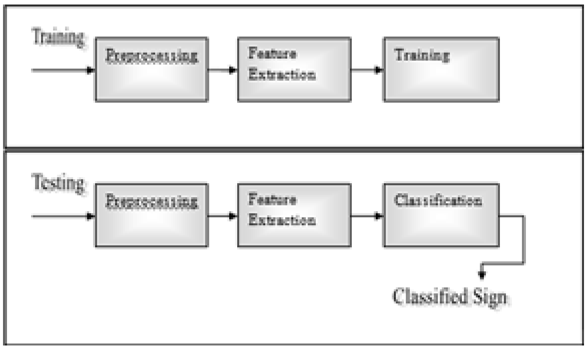


Fig. 3.1 The Recognition process

Features for each hand shape are represented in the form of crisp IF-THEN rules that are extracted from the values of synaptic weights of the corresponding trained Fuzzy Neural Network.[5]. Each crisp IF-THEN rule is then fuzzified by employing a special membership function in order to represent the degree to which a pattern is similar to the corresponding antecedent part. For the recognition step, a Fuzzy Neural Network will be used. We propose fuzzy control concepts and techniques to determine adaptively the learning parameters so that the performance of the neural network will be improved. A general architecture for such system is shown in figure 3.4.

When an unknown gesture is to be classified, each sample of the unknown gesture is tested by each fuzzy rule[6]. The accumulated similarity associated with all samples of the input is computed for each hand gesture in the vocabulary, and the unknown gesture is classified as the gesture yielding the highest accumulative similarity. Based on the method we can implement a small-sized dynamic hand gesture recognition system.

We have introduced, a hand posture and gesture modeling and recognition system, which can be used as an interface to make possible communication with smart environment (intelligent space) by simple hand gestures. The system transforms preprocessed data of the detected hand into a fuzzy hand-posture feature model[7] by using fuzzy neural networks and based on this model determines the actual hand posture applying fuzzy inference[8]. Finally, from the sequence of detected hand postures, the system can recognize the hand gesture of the user.

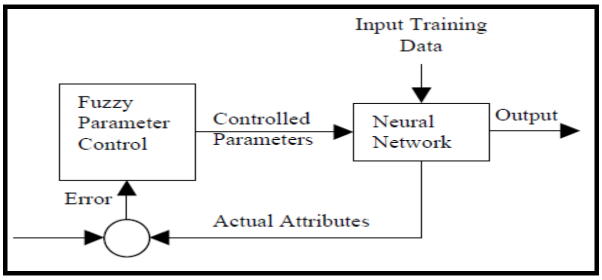


Figure 3.2 : System using a fuzzy controller for the adaptation of neural learning parameter

At any instant of time, the hand gesture can be represented by its position and orientation[9] of the hand, and the configuration of the fingers. From the signal point of view, we may say that a sequence of regularly sampled gestures constitute the motion of the hands and fingers. This sequence of gestures will be termed as dynamic gestures. The dynamic gestures as a whole present very complicated forms of data and may not be easily characterized by conventional means, such as image features. To initiate any dynamic gesture, there should be some starting point in space; that is to say, the hands should start their motions from some initial positions. In our proposed work, gestures are required using a desktop camera observing a stationary background, under room lighting. For the purpose of further processing , images will be represented as 2D vectors. All images will be considered to be the binarized, canonicalized versions as in Figure 3.3(a).

We wish to recognize gestures based on a previous acquired training set. Let us assume that we have K gestures, and for the k^{th} gesture we have obtained M_k training examples; N is the number of image pixels 32x32 here which are numbered 1,2…1024 from left to right and top to bottom. Any 4 x 4 pixels is called a subpattern and the subpatterns are numbered 1, 2… 64 (Fig.3.3. (b)).

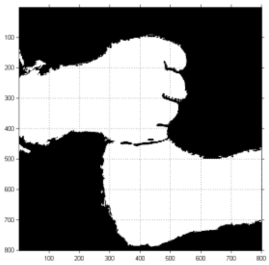
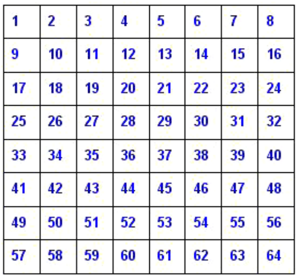


Figure.3.3 (a) An example pattern



1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Figure.3.3 (b) Its 64 nominal subpatterns

3.1 Proposed Algorithm for Recognition

The algorithm for Fuzzy Neural Network performs the following steps to recognize any pattern:

1) Learns all the training patterns and remembers their subsections;

2) Divides the new input pattern into 64 segments as described above;

3) Compares each of the segments individually against the corresponding segments of the trained patterns to compute any match;

4) Finds the total number of segments that matched for any pattern and finds the closest overall match.

3.2 Network creation and training gesture

For convenience, we use the following notation in referring to nodes: let N_i be a pattern node. Then $N_{i,\ j}$ denote the j^{th} subpattern node of N_i . $N_{i,\ j,(d\ x\ ,d\ y\ )}$ denotes the sampling-subpattern node of $N_{i,j}$ that takes care of the input subpattern, which is (d_x, d_y). Away from the subpattern. As usual, X is in the horizontal direction and Y is in the vertical direction. A positive (negative) d_x denotes a right (left) shift and a positive (negative) d_y denotes a down (up) shift of the subpattern. We may call $N_{i,\ j,(\ d\ x\ ,d\ y\ )}$ as the (d_x, d $_y$)th sampling- subpattern node of $N_{i,j}$. We define a subnet of a pattern node N_k to be the sub-network consisting of the pattern node N_k, its sub pattern nodes and sampling-subpattern nodes, together with all the associated links.

Suppose we are given a set of training patterns. Each pattern is represented by a row matrix G of 1024 pixels, and each sub pattern by a row matrix I_j of 16 pixels, $1 \le j \le 64$, namely,

$P_j = [P_{j1}, P_{j\,2}, \ldots\ldots\ P_{j64}]$ ....(1)

and

$G = [P_1, P_2, \ldots\ldots, P_k]$ ....(2)

Where $P_{j,k}$ is the normalized gray level of the corresponding pixel, i.e, $P_{j,k}$, $\in\{-1,1\}$and $1\le k\le 64$, $1\le j\le 64$ With 1 representing white and –1 representing black. For convenience, we represent the input to a sampling-subpattern node $N_{i,\ j,(d\ x\ ,d\ y\ )}$ by $P_j\ (d_x\ ,\ d\ _y\ )$, P_j (0,0) may be abbreviated as P_j Since we are doing supervised classification, it is assumed that the class of each input training pattern is known.

As mentioned earlier, each sub pattern node stores a node weight shared by all its sampling subpatttern nodes. For a subpattern node $N_{i,j}$ its node weight $W_{i,j}$ is defined to be

$$W_{i,j} = \left[W_{i,j,1}, W_{i,j,2}, \ldots\ldots\ldots W_{i,j,64}\right] \text{.......................(3)}$$

Where $W_{i,j,k} \in Z$, $1 \le k \le 64$, is an integer. Suppose an input training pattern G with class C is presented to the network. Each sampling-subpattern node $N_{i,j}(d_x, d_y)$ computes its output $Out_{i,j}(d_x, d_y)$ by

$$Out_{i,j}(d_x, d_y) = \frac{W_{i,j} * P_j^T(d_x, d_y)}{\alpha + \left|W_{i,j}\right|} \text{..................(4)}$$

Where $\left|W_{i,j}\right| = \sum_{k=1}^{64} W_{i,j,k}$ the superscript T stands for matrix transposition and α≥0. Since each element of $P_j^T(S_x, S_y) \in \{-1,1\}$ is either 1 or –1, the following relationship holds:

$$\left|W_{i,j}\right| \le W_{i,j} * P_j^T(d_x, d_y) \le \left|W_{i,j}\right| \text{..................}\mathbf{(5)}$$

Therefore, we have $-1 \le Out_{i,j}(d_x, d_y) \le 1$ Apparently, measure the similarity between and the node weight stored in . The more is similar to the stored weight , the closer is to 1. On the contrary, the more is different from

measure the similarity between $P_j\ (d_x, d_y)$ and the node weight $W_{i,j}$ stored in $N_{i,j}$. The more $P_j\ (d_x, d_y)$ is similar to the stored weight $W_{i,j}$, the closer $Out_{i,j}(d_x, d_y)$ is to 1. On the contrary, the more is $P_j\ (d_x, d_y)$ different from $W_{i,j}$ the closer $Out_{i,j}(d_x, d_y)$ is to –1. All the outputs of $N_{i,j}(d_x, d_y)$ *(sampling-subpattern nodes)* are sent to respective $N_{i,j}$ *(subpattern nodes).* Each sub pattern node $N_{i,j}$ takes the maximum value of $Out_{i,j}(d_x, d_y)$ *(all its input),* by the way

$$Out_{i,j} = \max\left(Out_{i,j(-d,-d)}, \ldots\ldots\ldots Out_{i,j(0,0)}, \ldots\ldots\ldots Out_{i,j(d,d)}\right)$$

...................(6)

and sends this value to its pattern node N_i the way $O_{i,\ j}$ is computed reflects the concept of recognition by parts. Also, this account for the tolerability of GRFNN of deformation, noise, and shift in position. Obviously, -1 ≤ $Out_{i,j}$ ≤ 1 for every possible i and j.

Let the Priority index Pr_i, for the pattern node N_i be defined by

$$\Pr_i = \sum_{j=1}^{64}(3\times Out_{i,j} - 2)^{1/3}$$

..................................(7)

With j ranging over the 64 subpatterns nodes of Ni. Using priority indexes makes the training procedure more efficient. The priority indexes of all pattern nodes stored in decreasing order and placed in the priority list. Suppose the largest priority index in the priority list is Pr_k. Let the pattern node corresponding to P_k be N_k, the class for N_k be C_k and N_k's vigilance be ρ_k.

Compute the following matching degree M_k for N_k

$$M_k = \frac{\sum_{j=1}^{64}(\omega_{k,j} \wedge Out_{i,j} + 1)}{\sum_{j=1}^{64}(\omega_{k,j} + 1)}$$

....(8)

Where $\omega_{k,,j}$ is link weight between N_k and $N_{k,j}$. The operator Λ is defined as $\omega_{k,j} \wedge Out_{i,j} = \min(\omega_{k,j}, Out_{i,j})$.

Since $\omega_{k,j} \wedge Out_{i,j} \le \omega_{k,j}\,; \omega_{k,j} \ge$-, we have $0 \le M_k \le 1.$

4. ANFIS Architecture

The main objective of ANFIS is to optimize the parameters of a given fuzzy inference system by applying a learning procedure using a set of input-output data pairs (called training data). The parameter optimization is done in a way such that the error measure between the desired and the actual output is minimized.

ANFIS is a multilayer feed-forward network that uses neural network learning algorithms and fuzzy logic to map an input space to an output space. Five layers are used to construct this inference system. Each layer contains several nodes described by the node function. Adaptive nodes, denoted by squares, represent the parameter sets that are adjustable in these nodes, whereas fixed nodes, denoted by circles, represent the parameter sets that are fixed in the system. The output data from the nodes in the previous layers will be the input in the present layer. There are two types of fuzzy inference system (FIS) described in the literature [10]. The most important difference between the two systems is the definition of the consequence parameter. The consequence parameter in Sugeno FIS[11] is either a linear equation, called "first-order Sugeno FIS", or constant coefficient. The Sugeno FIS is used in the present work.

The architecture of ANFIS is a feed forward network that consists of 5 layers . Fig4.1. shows the equivalent ANFIS architecture for a two input Sugeno-type fuzzy inference system. A rule in the first order Sugeno FIS has the form:

If x is Ai and y is Bi , then fi = $p_i x + q_i y + r_i$.

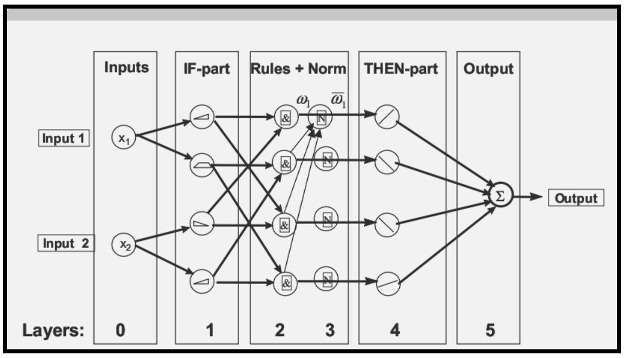


Fig. 4.1. ANFIS architecture for a two-input, two-rule Sugeno FIS.

Layer 1 Every node in this layer is an adaptive node with a node function where *x* (or *y*) is the input to node *I* and *Ai* (or *Bi*) is a linguistic label and O_i^1 is the membership grade of fuzzy set *A* (= *A1*, *A2*, *B1* or *B2*) and it specifies the degree to which the given input *x* (or *y*) satisfies the quantifier *A*. The membership function for *A* can be parameterized membership function as given in equation 1 or normally known as {*ai, bi, ci*} is the parameter set.The output of a node in the first layer specifies to which degree a given input, x, satisfies a quantifier, A, i.e., the function of the node i in this layer is a membership function for the quantifier, Ai , of the form

$$O_i^1 = \mu Ai\ (x) \text{………………………}(1)$$

Each membership function has a set of parameters that can be used to control that membership function. For example, a Gaussian membership function that has the form

$$\mu Ai\ (x) = e^{-((x-ci)\sigma i)^2} \text{……………………..}(2)$$

has two parameters, ci and σi .

Tuning the values of these parameters will vary the membership function, which means a change in the behavior of the FIS. Parameters in this layer are referred to as *premise parameters* .

Layer 2 Every node in this layer is a fixed node , whose output is the product of all the incoming signals Each node output represents the firing strength of a rule.In the second layer, the output of a node represents a firing strength of a rule. The node generates the output (firing strength) by multiplying the signals that come on its input, i.e.,

$$w_i = \mu Ai\ (x) \times \mu Bi\ (y) \text{………………………}(3)$$

Layer 3 Every node in this layer is a fixed node . The *i*th node calculates the ratio of the *i*th rule's firing strength to the sum of all rules' firing strengths. outputs of this layer are called normalized firing strengths.The function of a node in the third layer is to compute the ratio between the i th rule's firing strength to the sum of all rules' firing strengths:

$$\overline{w_i} = \frac{w_i}{w_1 + w_2} \text{……………………………}(4)$$

wi is referred to as the *normalized firing strength*.

Layer 4 Every node *I* in this layer is an adaptive node with a node function. Where *wi* is a normalized firing strength from layer 3 and { *pi* ,*qi* ,*ri* } is the parameter set of this node. Parameters in this layer are referred to as *consequent parameters* .

. In the fourth layer, each node has a function of the form:

$$O_i^4 = \overline{w_i} f_i = w_i (p_i x + q_i y + r_i) \text{…………..}(5)$$

Layer 5 The single node in this layer is a fixed node labeled Σ, which computes the overall output as the summation of all incoming signals. For simplicity, we assume that the fuzzy inference system under consideration has two input *x* and *y* and one output *z.*

The overall output is computed in the fifth layer by summing all the incoming signals, i.e.,

$$O_i^5 = f = \sum_i \overline{w_i} f_i = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} \text{……………}(6)$$

During the learning process, the premise and consequent parameters are tuned until the desired response of the FIS is achieved .

4.1 Subtractive clustering

As a first step towards extracting the initial fuzzy model the subtractive clustering technique is applied to the input output data pairs, which are obtained from the system which is to be modelled. The cluster estimation technique[12] helps in locating the cluster centers of the input output data pairs. This in turn helps in the determination of the rules which are scattered in input output space, as each cluster center is an indication of the presence of a rule. In addition to this it also helps to determine the value of the premise parameters. This is important because an initial value, which is very close to the final value, will eventually result in the quick convergence of the model towards its final value during the training session with neural network. In this clustering technique the potential of all the input output data points are calculated as a function of their Euclidian distance from all the other data points. The points having a potential above a certain preset value are considered as cluster centers. After the cluster centers are ascertained the initial fuzzy model[13] can be subsequently extracted as the centers will also give an indication of the number of linguistic variables. The cluster estimation method for determining the number of rules and initial rule parameters is briefly described below. This technique is used when there is not a clear idea on how many clusters there should be for a given set of data, substractive clustering method can be used for estimating the number of clusters and the cluster centers in a set of data[13]. It assumes each data point is a potential cluster center and calculates a measure of the potential for each data point based on the density of surrounding data points. The algorithm selects the data point with the highest potential as the first cluster center and then destroys the potential of data points near the first cluster center. This process of acquiring a new cluster center and destroying the potential of surrounding data points is repeated until the potential of all data points falls below a threshold.

To extract rules[14] from data, we first separate the training data into groups according to their respective classes. Consider a group of *n* data points {*x*1,*x*2,...,*xn*} for a specific class, where *xi* is a vector in the input feature space. We consider each data point as a potential cluster center for the group and define a measure of the potential of data point *xi* to serve as a cluster center point, xi , as

$$\mathbf{Pi} = \sum_{j=1}^{n} e^{-\|x_{i-}x_j\|^{2/(r_{a/2})^2}} \quad \ldots\ldots\ldots\ldots\ldots\ldots..(7)$$

where r_a is a positive constant

||.|| denotes the Euclidean distance[15], and "*ra"* is a positive constant. Thus, the measure of the potential of a data point is a function of its distances to all other data points in its group. A data point with many neighboring data points will have a high potential value. The constant "*ra"* is effectively a normalized radius defining a neighborhood; data points outside this radius have little influence on the potential. Note that because the data space is normalized "*ra"*= 1.0 is equal to the length of one side of the data space. After the potential of every data point in the group has been computed, we select the data point with the highest potential[16] as the first cluster center. Let *xi* be the location of the first cluster center and *P*i be its potential value. We then revise the potential of each data point *xi* in the group .Then, the density measure for each data point, xi , is reduced according to the formula:

$$\mathbf{Pi} = \mathbf{P}_i - P_{c1} e^{-\|x_i - x_{c1}\| 2/(r_{b/2})^2} \quad \ldots\ldots\ldots\ldots\ldots..(8$$

)

where xc1 is the point selected as the first cluster center, Pc1 is its density measure, and rb is a positive constant.

Thus, we subtract an amount of potential from each data point as a function of its distance from the first cluster center. The data points near the first cluster center will have greatly reduced potential, and therefore will unlikely be selected as the next cluster center for the group. The constant "*rb"* is effectively the radius defining the neighborhood which will have measurable reductions in potential. To avoid obtaining closely spaced cluster centers, we typically choose "*rb"* = 1.25 "*ra"*. When the potential of all data points in the group has been reduced according to Eq. (8), we select the data point with the highest remaining potential as the second cluster center. We then further reduce the potential of each data point according to their distance to the second cluster center. Note that data points close to the cluster center will have significantly reduced density measure so that they are not likely to be selected as the next cluster center. The next cluster center is chosen and the density measure is reduced again. This process is repeated until a stopping criterion is met.

4.2 Feature extraction

Selecting good features is a crucial step in any object recognition system. One may ask why not using the image itself as a feature? There are two reasons for not doing so. First of all, the high dimensionality of the image makes it unreasonable to use the image as a feature. Secondly, many of the information embedded in the image data are redundant and some may be not useful. Therefore, the objective of the feature extraction phase is to represent the image by a set of numerical features that correspond to the useful information, remove the redundancy of image data, and reduce its dimensionality. For a set of features to be considered reasonable, they must satisfy the following [17]:

(1) Images for objects from the same class must have very similar features.

(2) Features for objects from different classes must be noticeably different.

(3) Features must be scale, translation, and rotation invariant, which means that they must be able to recognize objects regardless of their size, location, and orientation.

Our feature extraction scheme uses the border information, the center of area, and the gesture direction to extract a feature vector for the gesture. The approach depends on originating vectors from the center of area to the portion of the border that bears the most important information about the gesture (fingertips region). The lengths of the vectors are taken as the corresponding features of the gesture. If cx and cy are the coordinates of the center of area, and bx and by are the coordinates of a point on the border, the length of the vector is given by:

$$l_{cb} = \sqrt{(\mathrm{bx - cx})^2 + (\mathrm{by - cy})^2}$$

and the vector direction is given by:

$$\theta_{cb} = \tan^{-1}\left(\frac{by - cy}{bx - cx}\right)$$

Two problems were found here. First, how can we determine the useful part of the border, and second, what number of vectors should be used? To determine the useful portion of the border, the direction of the gesture is used. We find that the information discriminating gestures in Indian sign language lies in the region between the angles:

$$\left(90 - \theta_g\right) \le \theta \le \left(113 + \theta_g\right)$$

where µg is the gesture direction. Consequently, only vectors with directions that satisfy the above inequality are used. The region specified by this equation corresponds to the region of fingertips and is derived based on the investigation of the hand gestures of several people. In this phase, the features extracted from the image data are recognized as a specific gesture. We use Adaptive Neuro-Fuzzy Inference Systems (ANFIS) as the underlying architecture for the recognition process. The adopted ANFIS architecture is of the type that is functionally equivalent to the first order Sugeno-type fuzzy inference system. For each of the gestures, an ANFIS model is built and trained to recognize the corresponding gesture. The resulting architecture is some times called *MANFIS* (Many ANFIS) .Each ANFIS model is trained to produce a value of 1 as an output if the data presented at its inputs corresponds to the gesture that the model is associated with, and a value of 0 otherwise.

The recognition process is done by presenting the features of the image to be classified to each of the 64 ANFIS models. This will result in 64 different responses. A voting scheme is applied to determine the class to which the image belongs.

The class (gesture) that is associated with the ANFIS model with the response closest to the value of 1 is chosen as the class to which the image under investigation belongs. An important issue that determines the effectiveness of this phase is the construction of the ANFIS models. The process of constructing an ANFIS model for a specific class involves two steps:

- Identification of a fuzzy model (Fuzzy Inference System) for that class, and
- Training the model using ANFIS.

These two steps are discussed in the following subsections.

4.3 Training fuzzy models

Once a fuzzy model for a class is identified, an ANFIS network equivalent to the model can be built as discussed in Section 2. The network is then trained using the hybrid learning algorithm such that the desired response is achieved. The hybrid learning algorithm combines gradient descent method and least-squares estimate (LSE) to identify parameters. The problem with using the gradient descent method alone is that it is generally slow and likely to become trapped into local minima . If an adaptive network is linear in some of the network’s parameters, LSE can be used to identify these parameters. For ANFIS, we can observe that when the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters. So the LSE can be used to identify these parameters. To apply the hybrid learning algorithm, each epoch consists of two passes; *forward pass* and *backward pass*. In the forward pass, the premise parameters are fixed and the consequent parameters are identified by LSE. In the backward pass, the error signals are propagated backwards and the premise parameters are updated using gradient descent. The consequent parameters identified this way are optimal (under the condition that the premise parameters are fixed) . This means that the hybrid learning algorithm will converge faster since it reduces the search space of the pure backpropagation.

5. Data sets used for our work

The data set used for training and testing the recognition system comprises of images for five different sets of signs of ISL out of the the 26 signs shown in Fig. 2.2. Samples for each gesture were taken from 4 different volunteers. For each signing gesture 100 samples each were used for training purpose. Training was performed until either an error goal of 0.2 is achieved, and maximum of 3 epochs is reached. The samples were taken from different distances from the camera, and with different orientations. This way, we were able to obtain a data set with cases that have different sizes and orientations, so that we can examine the capabilities of our feature extraction scheme.

Dataset 1: A, B, C, D, E, G, I, O, U, Q, X (Single and two hand signs)

.

Dataset 2: F, L, W, V, Y, H, J, N. (Single and two hand signs)

Dataset 3: K, M, T, P, R, S, Z. (Only two hand signs)

Dataset 4:A, X, G, B, D, E, K, T, R, Z. (Ten two hand signs)

Dataset 5: C, I, J, L, O, U, V, W, Y. (Nine Single hand signs)

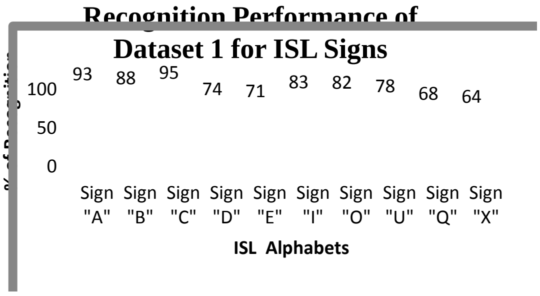
Here Dataset 1 , Dataset 2 and Dataset 3 combine to recognize all the 26 alphabets of ISL wherein these datasets comprise of both single hand and two hand signs mix in the dataset.

Dataset 3 comprise of ten two hand signs only that is a database having one type of data all of two hand signs .

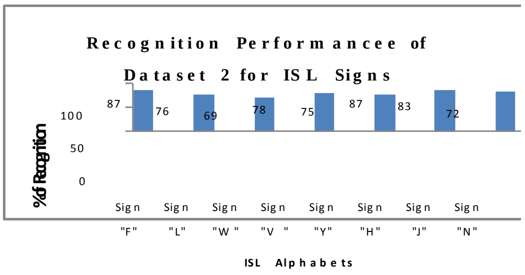
Dataset 4 comprise of 9 single hand signs only that is a database having one type of data all of single hand signs .

The Recognition Rate

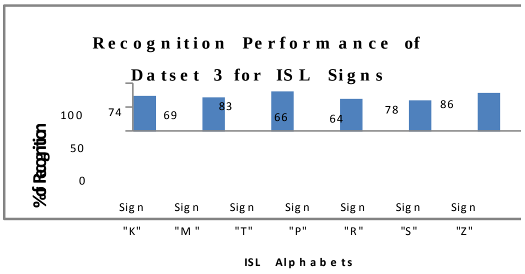
Dataset 1:



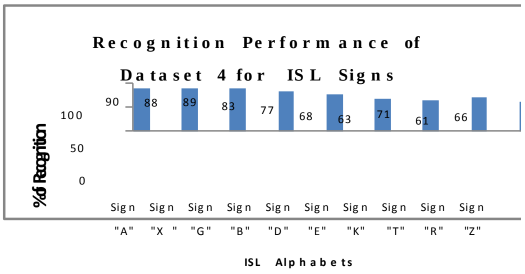
Dataset 2:



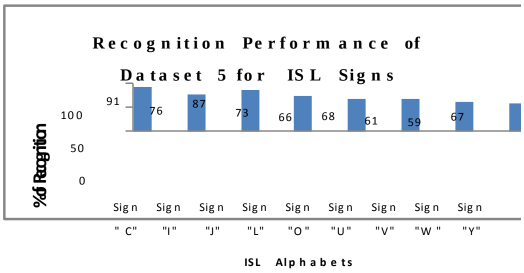
Dataset 3:



Dataset 4:



Dataset 5:



6. Results and Discussion

Having used Datasets in variety of combinations with single hand and two hand signs mix in some dataset and only single hand and only two hand signs in some dataset we have been able to recognize in all of 26 signs of ISL alphabets database which included both single hand and two hand signs , in our set of recognized signs we have included both types of signs and we conclude that the system designed by us is robust in a way that it can give recognition for both single hand and two hand signs of ISL if taken together as well as for only single hand or only two hands signs taken for experimental purpose. Most of the misclassified samples correspond to the gestures that are similar to each other. As an example, shows the gestures “E” and “F”. Because these gestures are similar, their corresponding features are also similar. Therefore, it is probable for a sample of the gesture “E” to be classified as “F” or vice versa.

REFERENCES

[1]Vasishta M., Woodward J., DeSantis S. 1998, “An Introduction to Indian Sign Language”, All India Federation of the Deaf (Third Edition).

[2] Tirthankar Dasgupta, Sambit Shukla, Sandeep Kumar,Synny Diwakar, Anupam Basu,“A Multilingual Multimedia Indian Sign Language Dictionary Tool”, The 6’th Workshop on Asian Language Resources, pp. 57-64, 2008.

[3] Anup Nandy, Jay Shankar Prasad, Pavan Chakraborty, G. C. Nandi, Soumik Mondal, “Classification of Indian Sign Language In Real Time”, In the proceedings of International Journal on Computer Engineering and Information Technology (IJCEIT), Vol. 10, No. 15,pp. 52-57, Feb. 2010.

[4]Indian Sign language, empowering the deaf, Indian Sign Language Database<http://www.deafsigns.org>.

[5]Binh, N.D .. T. Ejima, *"Hand gesture recognition using fuzzy neural network,"* in Proc. of ICGST Int. Conf. on Graphics, Vision and Image Processing, pp. 1-6,Cairo, Egypt,2005.

[6]Mu-Chun Su, *A Fuzzy Rule-Based Approach to Spatio-Temporal Hand Gesture Recognition*, IEEE Transactions on systems, man and Cybernetics-Part C Applications and Reviews, VOL. 30, NO. 2, MAY 2000

[7] Jong-Sung Kim, Won Jang, and Zeungnam Bien A Dynamic Gesture Recognition System for the Korean Sign Language (KSL) IEEE Transactions On Systems, Man, *And* Cybernetics-Part B: Cybernetics, Vol. 26, No. 2, April 1996

[8] Balazs Tusor ,Annamaria R. Varkonyi-K6czy, Circular Fuzzy Neural Network Based Hand Gesture and Posture Modeling IEEE Transactions On Instrumentation And Measurement, Vol. 60, No. 5, May 2011

[9]Omar Al-Jarrah , Alaa Halawani, Recognition of gestures in Arabic sign language using neuro-fuzzy systems Artificial Intelligence 133 (2001) 117–138

[10] Sugeno, M. "Industrial applications of fuzzy control", Elsevier Science Pub. Co, (1985).(18)

[11]Takagi T, Sugeno M, 1985. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15: 116–132.

[12]Sin, S. K., and De Figueiredo. 1993. Fuzzy System Designing Through Fuzzy Clustering and Optimal preDefuzzification. Proc. IEEE International Conference on Fuzzy Systems. 190-195.

[13] Gomez, A. F., M. Delgado, and M. A. Vila. 1999. About the Use of Fuzzy Clustering Techniques for Fuzzy Model Identification. *Fuzzy Set and Systems*. 106: 179-188.

[14] Demirli, K., S. X. Cheng, and P. Muthukumaran. 2003. Subtractive Clustering Based Modeling of Job Sequencing with Parametric Search. *Fuzzy Sets and Systems*. 137: 235-270.

[15]Surmann, H., and A. Selenschtschikow. 2002. Automatic Generation of Fuzzy Rule Bases: Example I. Proc. of the NF2002 First International ICSC on Neuro-Fuzzy Technologies. January 2002.

[16]S.L. Chiu, Fuzzy model identification based on cluster estimation, J. Intelligent and Fuzzy Systems 2 (3) (1994) 267–278.

[17]A. Khotanzad, J.-H. Lu, Classification of invariant image representations using a neural network, IEEE Transactions Acoustics, Speech, and Signal Processing 38 (1990) 1028–1038.