Segment, Track, Extract, Recognize and Convert Sign Language Videos to Voice/Text

P.V.V.Kishore
Dept. of ECE, Research Scholar
Andhra University College of Engineering
Andhra Pradesh, INDIA

P.Rajesh Kumar
Dept. of ECE, Associate Professor
Andhra University College of Engineering
Andhra Pradesh, INDIA.

Abstract— **This paper summarizes various algorithms used to design a sign language recognition system. Sign language is the language used by deaf people to communicate among themselves and with normal people. We designed a real time sign language recognition system that can recognize gestures of sign language from videos under complex backgrounds. Segmenting and tracking of non-rigid hands and head of the signer in sign language videos is achieved by using active contour models. Active contour energy minimization is done using signers hand and head skin colour, texture, boundary and shape information. Classification of signs is done by an artificial neural network using error back propagation algorithm. Each sign in the video is converted into a voice and text command. The system has been implemented successfully for 351 signs of Indian Sign Language under different possible video environments. The recognition rates are calculated for different video environments.**

Keywords- Sign Language; Multiple Video object Segmentations and Trackings; Active Contours; skin colour, texture, boundary and shape; Feature Vector; Artificial Neural Network.

I. INTRODUCTION

Sign language is a vision based language of hearing impaired people, which involves the use of hands, the face and body. Sign language recognition system works on five essential parameters; hand shapes, hands movement, hand and head orientation, hand and head location and facial expressions. Out of these five parameters the foremost fundamental requirement is hand shapes. The second most important parameters are hand movements, hand and head orientation and their location in the video frame. These parameters can be incorporated into a sign language recognition system to improve the performance of the system by segmenting and tracking hands and head of the signer in the video. Segmented hand shapes and their locations are used to design a feature vector. The created feature vector is used to train the neural network. In the last decade extensive research had been done to understand hand gestures from the movement of human hands [1,2]. Compared to tracking techniques to rigid objects [3], tracking non-rigid objects such as bare human hands are a very complex and challenging task.

Going back to early days of sign language, Stokoe et.al [4] showed that signs were made of basic articulatory units. Initially, they were referred as cheremes. At present, they are being referred as phonemes, in resemblance with words in spoken language.

The major difficulty in sign language recognition, compared to speech recognition, is to recognize [5] different communication attributes of a signer, such as hands and head movement, facial expressions and body poses simultaneously. All these attributes are considered as a good recognition system.

The second major problem, faced by sign language recognition system designers, is tracking the signer in the clutter of other information available in the video. This is addressed by many researchers as signing space [6]. A sign language space can be created with entities such as humans or objects stored in it around a 3D body centered space of the signer. The entities are executed at a certain location and later referenced by pointing to the space. Another major challenge faced by researchers is to define a model for spatial information, containing the entities, created during the sign language dialogue.

Additional difficulties arise in the form of background in which signer is located. Most of the methods, which have been developed so far, are used simple backgrounds [7,8,9,10] in controlled set-up such as simple backgrounds, special hardware like data gloves[11,12,13], restricted sets of actions, restricted number of signers, resulting different problems in sign language feature extraction.

Our research is directed towards recognizing gestures of Indian Sign language [33] under real time conditions such as varied lighting, different backgrounds and make the system independent of the signer. For segmentation and tracking, we have employed active contour models, which are capable of segmenting and tracking non rigid hand shapes and head movements. Features are extracted from segments of hand and head shapes including tracking information, in the form of hand locations from each video frame. This feature vector is used to train the artificial neural network, the network outputs and determine the voice command relevant to the trained sign.

Active contours, popularly known in the research community as ‘snakes’, is an active research area with applications to image and video segmentation predominantly to locate object boundaries. They are also used for video object tracking applications. Active contours come under the category of model based segmentation and tracking methods, which are giving good results in the last few years [14,15,16]. The active contours were first introduced by Terzopoulos [17,18]. The basic idea, behind active contours, is to start with a curve anywhere in the image and move the curve in such a way that it sticks to the boundaries of the objects in the image. Thus it separates the background of the image from its objects. The original "snakes" algorithm was prone to topological disturbances and is exceedingly susceptible to initial conditions. However, with invention of level sets, [19] topological changes in the objects are automatically handled.

For tracking non-rigid moving objects, the most popular model based technique, is active contours [20]. Active contours can bend themselves based on the characteristics of the objects in the image frame. Previously created active contour models were only capable of tracking an object in a video sequence with a static background [21]. Jehan-Besson et.al [22] tracked objects in successive frames by matching the object intensity histogram using level sets. Though, the tracking error was in minimum level, it increased the object experiences intensity transformation due to noise or illumination changes. Almost all active contours methods discussed above were suffered from the problems related to cluttered backgrounds, lacking in texture information and occlusions from other objects in a video sequence. These problems can cause the performance of tracking non-rigid objects to decrease drastically.

Segmentation and tracking results are fused together to form a feature matrix, which is unique from the methods proposed for sign language feature extraction in [23, 24,25]. Finally, an artificial intelligence approach is proposed to recognize gestures of sign language as in [26,27,28,29]. Hidden Markov Models (HMM) were used extensively for classification for sign language [30, 31, 32].

The rest of the paper is organized as follows: sect. 2 introduces the proposed system for recognizing gestures of Indian Sign Language (INSLR), Sect.3 presents proposed segmentation and tracking using active contours, Sect.4 gives the creation of feature matrix, Sect.5 pattern classification using neural network, Sect.6 discusses experiments under various test conditions, and the last Sect.7, briefly concludes with future scope of our research.

II. Proposed System For Gesture Recognition

The proposed system has four processing stages namely, hand and head segmentation, hand and head tracking, feature extraction and gesture pattern classification. Fig. 1 shows the block process flow diagram of the proposed method.

From the Fig. 1 we can comprehend the overall working process of the system. Video segmentation stage divides the entire frames of images of the signers into hands and head segments of the signer. Hand and head tracking module gives the location of each hand and head in each video frame. Shape features, extracted from segmentation and location information from tracking, are fused into a feature matrix for each sign and are optimized before saving to the database. This process is repeated for all the available signs using a video of a signer. The sign language database can be restructured to add new signs into the database.

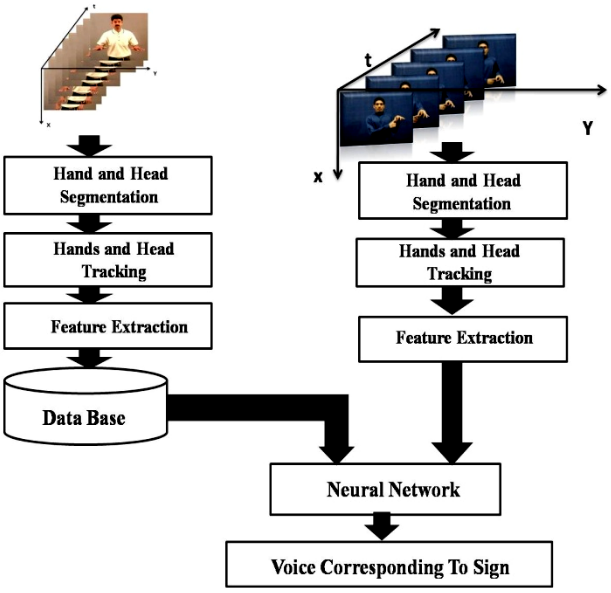

Figure 1. Sign Language Recognition System Architecture

Testing is done by giving a sign video sequence that is not initially trained and does not have a feature vector in the database. 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 video database of Indian signs in collaboration with Indian Deaf Society [33] and shanthi ashram school for deaf children. We have created 480 signs of alphabets, numbers, words and sentences by multiple signers of Indian sings and list is growing. A total of twenty different signers volunteered under different conditions with a total number of 4800 gesture videos for a total of 480 signs. To test the robustness of our system we also used sign language videos from YouTube, which are under different video backgrounds and in real environments taking the total video database count to 6800. Neural network is trained using the feature vector from the database and tested using the feature vector from the input of the system. The output of the neural network results in a voice and text message to the corresponding input sign.

III. Segmentation and Tracking

A. Active Contours-Level Sets

James A. Sethian and Stanley Osher [34] presented boundaries of $\mho(x)$ implicitly and model their propagation using appropriate partial differential equations. The boundary is given by level sets of a function $\phi(x)$. In level sets method, the interface boundary is characterized by a zero level set function $\phi(x) = 0, where\ \phi: \mathbb{R}^{\mathbf{2}} \to \mathbb{R}$. $\mho$ is defined for all values of x,

$$\mho = \{\phi(x) = 0, x \in \mathbb{R}^2\} \qquad (1)$$

The sign of $\phi(x)$ defines whether x is inside the contour $\mho$ or external to it. The sets $\mho^{Int} = \{x, \phi(x) \leq 0\}$ and $\mho^{Ext} = \{x, \phi(x) > 0\}$. The level set evolves based on the curvature κ of the image objects and assuming the curve moving towards the outward normal $\vec{\boldsymbol{n}}$ defined in terms of parameter ϕ as

$$\boldsymbol{\kappa} = \nabla.\left[\frac{\nabla\phi}{|\nabla\phi|}\right] \quad and \quad \vec{\boldsymbol{n}} = \frac{\nabla\phi}{|\nabla\phi|} \tag{2}$$

Usually the curve $\mho$ evolution is a time dependent process and the time dependent level set function $\phi: \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$ as $\mho(\mathrm{t}) = \{\phi(x,t) = 0, x \in \mathbb{R}^2\}$. One way to solve is to approximate spatial derivatives of motion and update the position of the curve over time. This method of solving the level sets is prone to unsteadiness due to erroneousness detection of position of the curve.

A different approach was proposed from the theory of level sets in [34]. Start with a zero level set $\phi(x) = 0$ of higher dimension function and entrench the object curvature. Initializing the level set function ϕ at $t = 0$, we have

$$\phi(x, t = 0) = \pm d \tag{3}$$

Where $'d'$ is signed distance function (sdf) from x to the curvature of the image object. If d is a positive value, x is outside the object boundary. if d is a negative value, x is inside the object boundary. The goal is to construct an equation for evolution of $\phi(x,t)$ to embrace the object boundaries from zero level set $\phi(x) = 0$. we can propagate the zero level set $\phi(x) = 0$, by solving a convection equation containing the velocity field v, which propagates all the level sets as

$$\phi_t + v.\nabla\phi = 0 \tag{4}$$

The motion is in normal velocity of the curve which is given by eq.2 as $v = F\vec{\boldsymbol{n}} = F\frac{\nabla\phi}{|\nabla\phi|}$. Inserting in eq.4 we get a level set equation of the form

$$\phi_t + F|\nabla\phi| = 0 \tag{5}$$

Eq.5 is a type of Hamilton-Jacobi equation. The speed term F is dependent of object curvature $\boldsymbol{\kappa} = \nabla.\left[\frac{\nabla\phi}{|\nabla\phi|}\right]$, which can be formulated as

$$F(\boldsymbol{\kappa}) = F_0 + F_1(F) \tag{6}$$

Eq.6 drives the contour to level out with the high curvature regions together with a diffusion term.

B. Hands and Head Segmentation and Tracking

This section presents the video object segmentation, tracking of hands and head simultaneously from a range of video backgrounds under different lighting conditions with diverse signers. A video sequence is defined as a sequence of image frames $\mathrm{I(x,y,t)}: \mathrm{D} \to \mathrm{R}$ where the images change over time. Alternatively, a succession of image frames can be represented as $I_{(n)}$ where $0 \leq n \leq \infty$. The basic principle behind our proposed segmentation and tracking technique is to localize the segment and track one or more moving objects of the n^{th} frame from the cues available from previous segmented and tracked objects in frames $I_{(1)}, I_{(2)}, \dots\dots\dots.. I_{(N)}$ such that subsequent contours $\Gamma^1, \Gamma^2 \dots\dots\dots \Gamma^N$ are available. The sign videos are composed of many moving objects along with the hands and head of the signer. Signer's hands and head are considered as image foreground denoted by $I^f_{(n)}$ and rest of the objects are as the image background $I^b_{(n)}$, for the image $\mathrm{I_{(n)}}$ in the video sequence. Foreground contour of the hands and head might be denoted by $\Gamma^f_{(n)}$. Our proposed video segmentation algorithm segments hands and head of signers using colour, texture, boundary and shape information about the signer given precedent understanding of hand and head shapes from $\boldsymbol{I^f_{(n-1)}}$ and $\boldsymbol{I^b_{(n-1)}}$..

C. Colour and and Texture Extraction Modules

Colour plays a vital role in segmentation of complex images. There are various colour models, but the RGB colour model is most commonly used for video acquisition. Processing on a RGB colour frame increases the size of feature vector and thereby making the tracking process lethargic. Instead of working with gray scale images which store intensity information about each pixel, we used each of the three colour planes separately to extract each colour feature vector. This allows us to work with only one plane at a time depending on the background colour level. We choose manually the colour plane which highlights the human object from a background of clutter. Once a colour plane is identified, texture features are calculated using coorelogram of pixel neighbourhood [35,36]. Texture is an irregular distribution of pixel intensities in an image. Allam.et.al [37] established that co-occurrence matrix (CM's) produce better texture classification results than other methods. Gray Co-occurrence matrix (GLCM) presented by Haralick.et.al [38] is most effectively used algorithm for texture feature extraction for image segmentation.

Let us consider a color plane of our original RGB video. The R color plane is now considered as $M \times N$ R coded 2D image. The element of co-occurrence matrix $C_{d,\theta}$ defines the joint probability of a pixel x_i of R color intensity $\boldsymbol{r_i}$ at a distance d and orientation θ to another pixel x_j at R color intensity $\boldsymbol{r_j}$.

$$C_{d,\theta}(r_i, r_j) = Pr\{I(z_1) = r_i \wedge I(z_2) = r_j : |z_1 - z_2|_\theta = d\} \tag{7}$$

where $|z_1 - z_2|_\theta$ gives the distance between pixels. For each co-occurrence matrix, we calculate four statistical properties: contrast (C), correlation (CO), energy (EN) and homogeneity (H) defined as follows

$$CO = \sum_{i,j} \frac{(i-\mu_i)(j-\mu_j)}{\sigma_i \sigma_j} C_{d,\theta}(r_i, r_j) \tag{8}$$

$$C = \sum_{i,j} |i-j|^2\, C_{d,\theta}(r_i, r_j) \tag{9}$$

$$EN = \sum_{i,j} \left(C_{d,\theta}(r_i, r_j)\right)^2 \tag{10}$$

$$H = \sum_{i,j} \frac{C_{d,\theta}(r_i, r_j)}{1 + |i-j|} \tag{11}$$

The logic, in which the above parameters are used as texture feature, is described as follows; the contrast represents inertia and variance of the texture the correlation term gives correlation between different elements of GLCM, CO is high for more complex textures. From eq.41 μ_i and μ_j mean values along i and j directions.σ_i and σ_j represent variances, energy term describes the consistency of the texture, homogeneity is taken as a measure of coarsenesses of the texture. We used four different orientations $\theta = \{0,45,90,135\}$ and two distance measures $d = (1, -1)$ for calculation of GLCM matrix.

Finally, a feature vector $f^{vect}(\boldsymbol{x})$ is produced which is a combination of any one or all of the color planes and texture vector. Thus $f^{\boldsymbol{vect}}(\boldsymbol{x}) = \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}) \ldots\ldots\ldots . f_n(\boldsymbol{x})\}$ the feature vector contains color and texture values of each pixel in the image. This is a five dimension feature vector containing the first vector for any of the three color planes and the next four vectors for texture. We can also use all three color planes to represent color, and then the feature vector becomes a seven dimension feature vector.

Most of the image sequences contain many classes of colour and texture. Hence, we classify them as background and foreground pixels using Mixture of Gaussians (GMM) [39] clustering algorithm.

Given n- dimensional feature vector $\mathrm{f}^{\mathrm{vect}}(\mathrm{x})$ and the GMM algorithm classify this n-vector into $k-$categories. Thus, we model the object and background in an image frame using two mixture of Gaussian probability density functions (pdf): M^{ho} and M^b. The probability of finding a value of feature vector in the reference frame i.e. the first frame $I^{(1)}$is given by

$$p_{in}\left(\frac{f^{vect}(x)}{O_{hh}}\right) = \sum_{i=1}^{C_{obj}} w_{i,t} * p\left(\frac{f^{vect}(x)}{O_{hi}}\right) \quad for\ x \in Object \tag{12}$$

$$p_{out}\left(\frac{f^{vect}(x)}{B_{bk}}\right) = \sum_{i=1}^{C_{bck}} w_{j,t} * p\left(\frac{f^{vect}(x)}{b_{bj}}\right) \quad for\ x \in Background \tag{13}$$

where $p_{in}(f^{vect}(x)/O_{hh})$ and $p_{out}(f^{vect}(x)/B_{bk})$ represent mixture probabilities of human object and background. O_{hh}represents object hand and head portions. B_{bk} represents video background. $p(f^{vect}(x)/O_{hi})$ and $p\left(f^{vect}(x)/b_{bj}\right)$terms represent probabilities of multivariate guassians of object and background. C_{obj} and C_{bck} denote the number of clusters in the object and background mixtures. Weight terms $w_{i,t}$ and $w_{j,t}$ gives the estimate of the mixing parameters of two mixture models C_{obj}andC_{bck}. Maximum Likelihood Estimation (MLE) [40] algorithm is used to calculate the parameters $O_{hh} = \left(O_{hi}, w_{i,t}\right) \forall\ 1 \le i \le C_{obj}$ and $B_{bk} = \left(b_{bj}, w_{j,t}\right) \forall\ 1 \le j \le C_{bck}$. The MLE consists of estimating the unknown parameters of two mixtures by minimizing the likelihood functions

$$\lambda^0 = \prod_{x \in \boldsymbol{\mathcal{I}}_{\boldsymbol{f}}^{(\boldsymbol{1})}} p_{in}(f^{vect}(x)/O_{hh}) \tag{14}$$

$$\lambda^b = \prod_{x \in \boldsymbol{\mathcal{I}}_{\boldsymbol{b}}^{(\boldsymbol{1})}} p_{out}(f^{vect}(x)/B_{bk}) \tag{15}$$

The estimation problem comes down to solve the following equations

$$O_{hh} = arg \max_{O_{hh}}[\lambda^0] \tag{16}$$

$$\overline{B_{bk}} = arg \max_{B_{bk}}[\lambda^b] \tag{17}$$

We will assume, at this point, is that objects in the video sequence pretty much remain same when compared to background, that varies due to camera movement or changes in background scenes. This can be taken care by periodically updating the background clusters with some threshold if the changes in consecutive background frame cross the specified threshold.

For a pixel x in any frame $I_{(n)}$ where $n > 1$, the above procedure is used to calculate the object mixture probability $\overline{p_{\imath n}(x)}$ and background mixture probability $\overline{p_{out}(x)}$ on a neighbourhood of pixels which is eight in this paper. To track the objects in the current frame $I_{(n)}$, the initial contour position in previous frame $I_{(n-1)}$ is moved towards new object boundaries in current frame $I_{(n)}$ by minimizing the following energy functional of the level set

$$\begin{aligned} E^{CT}(I^f) = \iint_{\jmath f} D\left(p_{in}, \overline{p_{\imath n}(x)}\right) dx \\ + \iint_{\jmath b} D(p_{out}, \overline{p_{out}(x)})\, dx \end{aligned} \tag{18}$$

Where p_{in} and p_{out} are object and background PDF's in the reference frame $I_{(1)}$. $\overline{p_{\imath n}(x)}$ and $\overline{p_{out}(x)}$ are object and background PDF's in the current frame $I_{(n)}$. $D\left(p_{in}, \overline{p_{\imath n}(x)}\right)$and $D(p_{out}, \overline{p_{out}(x)})$ represent KL-distance symmetric which is computed between two PDF's by the following equation

$$\begin{aligned} D\left(p_{in}, \overline{p_{\imath n}(x)}\right) = \frac{1}{2}\Bigg[\int_z p_{in}(z) * \log(p_{in}(z)/\overline{p_{\imath n}(z)})dz \\ + \int_z \overline{p_{\imath n}(z)} * \log(p_{in}(z)/\overline{p_{\imath n}(z)})dz\Bigg] \end{aligned} \tag{19}$$

Where the integrals are calculated over domain z. The energy function in eq.18 is used to track the object boundaries in $I_{(n)}$by calculating the local statistics of the pixels within the object separating them from background. We can implement this by assigning pixel x to object in the current frame when $D\left(p_{in}, \overline{p_{\imath n}(x)}\right) > D(p_{out}, \overline{p_{out}(x)})$ and to the background otherwise.

D. Object Boundary Module

In the earlier module the focus was on extracting region information with the objective of minimizing the object contour energy function, which segments the objects of interest in the video sequence. But poor lightning can impact image region information in a big way. Hence, we extracted boundary edge map of the image objects which only depends on image derivatives. The way out would be to couple the region information in the form of colour and texture features to boundary edge map to create a good tracking of image objects.

We define the boundary B^o as pixels that are present in edge map of the object. The boundary pixels can be calculated by using gradient operator on the image. To align the initial contour Γ^0 from previous frame to the objects in the current frame to pixels on the boundary we propose to minimize the following energy function

$$E^B(I^f) = \int_{arc(Length\ og\ Obj)} g\big(\mathcal{B}^O(x)\big)dx \qquad (20)$$

Where $arc(Length\ of\ obj)$ is the length of the object boundary. The function g is an edge detection function. The boundary energy reaches to a minimum when the initial contour aligns itself with the boundary of the objects in the image. The minimization of energy in eq.20 also results in a smooth curve during evolution of the contour [41].

E. Shape Information Module

Before Even with colour, texture and boundary values of pixels in the image, the greatest challenge comes when object pixels and background pixels share the same colour and texture information. This happens because we are trying to track non- rigid objects that are hands and head of the signer along with finger positions and orientations, which change frequently in sign video. The problem influences the propagation of contour and results in meagre tracking of non-rigid objects in video sequences. The active contour can be influenced by giving information regarding the shape of the object computed from the previous frames. The method followed in [42] is used to construct the influence of shape of non-rigid objects in the image sequence. As for the first frame $I_{(1)}$, where prior shape information is not available, we just used the region and boundary information for segmentation. The segmented objects in frame one are used to initialize contours in the next frame for tracking. For $I_{(n)} \forall\, n = 1$ the tracking of $I_{(n)}$ is given by the level set contour $\Gamma^{(n)}$ which minimizes the energy function

$$E^S = \min_T \int_{\Gamma^{int}} \phi_0(T^{-1}x)dx \qquad (21)$$

where the minimum is calculated over Euclidian Similarity Transformations $T: \mathbb{R}^2 \to \mathbb{R}^2$ which is a combination of translational and rotational parameters. Minimizing over groups of transformations to achieve rigid object interactions was proposed by chan and zhu [33]. We propose to use a non-rigid shape influence term in this paper. Now let us recollect Γ, which indicates the active contour and Γ^0 be the active contour for the shape from the first frame. Let $\phi: \mathbb{R}^2 \to \mathbb{R} = \phi(x)$ be a level set distance function associated with contour Γ, and $\phi_0: \mathbb{R}^2 \to \mathbb{R} = \phi_0(x)$ is a level set function with contour Γ^0 from first image frame $I_{(1)}$. Let x be a pixel in the image space R fixed, $\phi(x) = \phi(\mho; x)$ is actually a function of contour Γ. The initial contour Γ aligns itself with the object contour Γ^0 in the first frame that is the initial contour for the next frame in the video sequence coming from the contour in the previous frame. Hence the shape interaction term proposed in this paper has the from

$$E^S(I^f) = \int_{\Gamma^{int}} \phi_0(x)dx \qquad (22)$$

Thus by applying shape energy to the level set we can effectively track hands and head in sign videos and we could differentiate between object contour modifications due to motion and shape changes.

F. Level Set Energy Function For segmentation and Tracking Combining Colour, Texture, boundary and Shape Modules

By integrating the energy functions from colour, texture, boundary and shape modules we formulate the following combined energy functional of the active contour as

$$E_{int}(I^f) = \alpha E_{CT}(I^f) + \beta E^O(I^f) + \gamma E^S(I^f) \qquad (23)$$

where α, β, γ are weighting parameters that provide stability to contribution from different energy terms. All terms are positive real numbers. The minimization of the energy function is done with the help of Euler-Lagrange equations and realized using level set functions. The resultant level set formulation is

$$\frac{d\phi^n(x,t)}{dt} = \big((\alpha R_{CT}(\phi^n) + \beta R^b(\phi^n) + \gamma R^S(\phi^n)\big)\|\nabla\phi^n\| \qquad (24)$$

where

$$R_{CT}(\phi^n) = -D\big(p_{in}, \overline{p_{in}(x)}\big) + D\big(p_{out}, \overline{p_{out}(x)}\big) \qquad (25)$$

$$R^b(\phi^n) = g(B^O(x)) + \nabla .\, g(B^O(x)) \left[\frac{\nabla\phi}{|\nabla\phi|}\right] \qquad (26)$$

$$R^S(\phi^n) = \phi_0(x) \qquad (27)$$

The numerical implementation of eq.24 is computed using narrowband algorithm [43]. The algorithm approximates all the derivatives in eq.24 using finite differences. The level set function is reinitialized when the zero level set clutches the boundary of the object in the image frame.

IV. Feature Vector

After Successful segmentation and tracking the feature extraction, a feature vector is being created, that is stored in the database at training stage as templates or can be used as inputs to pattern classifiers such as Hidden Markov Models (HMM), Neural Networks, Fuzzy Inference Systems or Decision Trees, which have limited memory. Hence, it becomes important to design a feature matrix that optimizes the performance of the classifier.

The feature matrix f^{Mat} derived from a video sequence is a fusion of hand and head segments representing shapes in each frame along with their location in the frame. The shape information for an nth frame is presented as ϕ^n , which is a binary matrix equal to frame size. Tracking active contour results in location of hands and head contours in each frame and giving their location $(x^{(n)}, y^{(n)})$ values.

Temporal dimension is mostly neglected during feature extraction in video processing. In other words, video is considered as an extension of images. In addition to 2D nature of image, the temporal dimension is managed using a technique called temporal pooling [44]. Here, the extracted features are temporally pooled into one feature value for the whole video sequence as illustrated in fig. 2. Temporal pooling is largely done by averaging. Other simple statistical techniques are employed such as standard deviation, median or minimum/maximum.

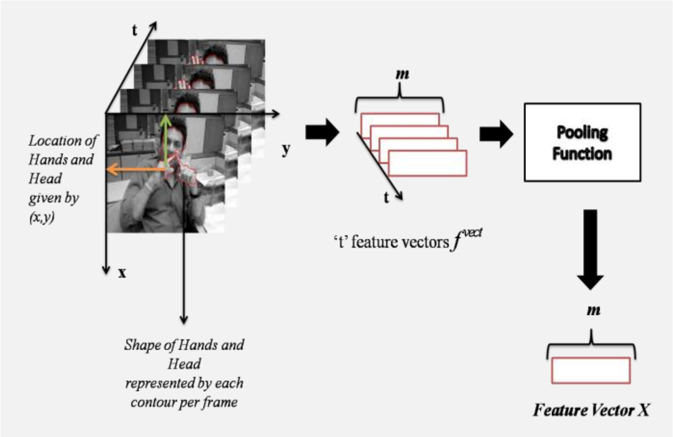
t
y
x
Location of Hands and Head given by (x,y)
m
t
't' feature vectors f^{vect}
Pooling Function
m
Shape of Hands and Head represented by each contour per frame
Feature Vector X

Figure 2. Feature Vector Design using Temporal Pooling

For an nth frame in the video sequence, the first row of feature vector f^{vect} consists of pixel values with the shape of active contour i.e. the pixels representing segmented head and hand shapes. The second and third row consists of $(x^{(n)}, y^{(n)})$ location information of those pixels that are segmented. The feature vector for each frame is a three dimensional vector f^{vect} representing shape and location information about each segmented hand and head contours. For an entire video in a sign language recognition system, each f^{vect} of three dimension becomes f^{Mat} feature matrix stacked in the form shown in fig. 2. A row in the feature matrix represents 3D f^{vect} for a particular frame and it is represented with time't'.

Temporal pooling is engaged to reduce the dimensionality of the feature vector for a particular video sequence. Averaging on each row of the f^{Mat} with respect to number of pixels in a frame leave us with a new reduced feature vector f^{Nvect} , which uniquely represents a particular video sequence. Finally each row in the new feature matrix f^{NMat} consists of feature vector representing each sign video sequence, stored in the database as templates.

V. Pattern Recognition-Neural Network

For handling large data such as in our system it is customary to use solutions that are fast enough without losing its accuracy. One of the few systems that can handle our large data matrices is Feed-Forward artificial neural network (ANN). Artificial neural network (ANN) is extensively used in detection of cancer [45], classifications [46], face recognition [47] and finger print recognition [48] ,to name a few. The feature matrix f^{NMat} is given as input for training the feed-forward neural network shown in fig. 3. Generally, a Feed-Forward neural network is a combination of three layers of neurons: input layer, hidden layer and output layer. The neurons in these layers are activated by using a nonlinear sigmoid activation function. Let $x_{i,j}(itr)$ where $1 \le i \le N$ *and* $1 \le \text{j} \le M$, be the input to the neural network derived from feature matrix f^{NMat} . Where M and N denote the number of columns and rows of f^{NMat} . *itr* is the number of iterations called Epochs in neural network terminology. The neural network outputs are denoted by $y_{i,j}(itr)$ where $1 \le i \le N$ *and* $1 \le \text{j} \le M$.

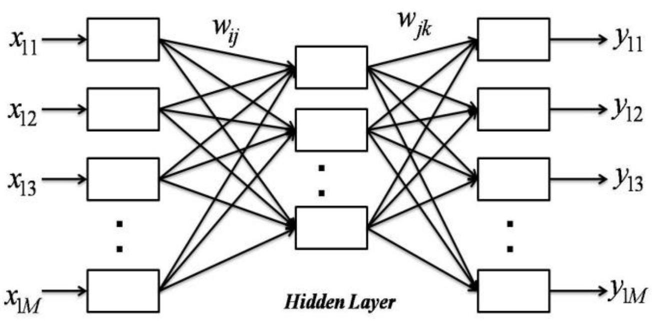

Figure 3. Neural Network Architecture

The back propagation algorithm follows the following steps in determining the output as discussed in [49]. The output of jth unit in the first hidden layer is calculated from the values in the hidden layer as

$$y_j(itr) = f(\sum_{i=1}^{M} x_i(itr)w_{ij}(itr) - \Theta_j) \qquad (28)$$

Where M is number of neurons in the hidden layer, w_{ij} is the weight vector connecting ith unit in between hidden layer and input layer and jth unit in between hidden layer and output layer. Θ is the threshold value of jth unit in hidden layer. $f(\bullet)$ is the activation function, which we choose as sigmoid function, defined as

$$f(\bullet) = \frac{1}{1 + e^{-x_{ij}(itr)}} \qquad (29)$$

The output layer output is calculated as

$$y_k(itr) = f(\sum_{j=1}^{M} x_{jk}(itr)w_{jk}(itr) - \Theta_j) \qquad (30)$$

where M is the number of neurons in the j^th^ layer for k^th^ neuron.

During back propagation process the error gradient from the output layer is calculated from the relation

$$\Delta_k(itr) = y_k(itr)[1 - y_k(itr)]e_k(itr) \tag{31}$$

where $e_k(itr)$ is the error in the output layer given as

$$e_k(itr) = y_{Tk}(itr) - y_k(itr) \tag{32}$$

where $y_{Tk}(itr)$ is the desired or targeted output of the system. Adjustment of the weights of the neurons is done iteratively using the equation

$$w_{ij}(itr+1) = w_{ij}(itr) + \mu y_j(itr) + \Delta_k(itr) \tag{33}$$

Where μ is adaptive learning rate of the neural network during training stage. After (k-1)th iteration and the kth feed-forward propagation, let the output vector be $y_{ij} = [y_{1j}, y_{2j}, y_{3j}, \dots\dots, y_{ij}]^T$ and the target vector $y_{Tk} = [y_{1k}, y_{2k}, y_{3k}, \dots\dots, y_{Tj}]^T$. The expression for adaptive learning rate is

$$\mu = \frac{\alpha}{e^{\left(-\frac{1}{j}\sum_{i=1}^{M}(y_{ij} - y_{Tk})^2\right)}} \tag{34}$$

where the value of α is. $0 \le \alpha \le 1$ The learning rate is controlled by error obtained during iterations.

The error gradient for the hidden layer is calculated using the relation

$$\Delta_j(itr) = y_j(itr)[1 - y_j(itr)]\sum_{k=1}^{i}\Delta_k(itr)w_{jk}(itr) \tag{35}$$

The above discussed procedure is run iteratively to reach the expected output.

VI. Experiments and Results

The first frame $I_{(1)}$ is segmented by calculating the feature vector consisting of single colour plane (R or G or B) and texture information along with boundary edge map. Then the proposed method is applied to the remaining frames of the video sequence. The segmentation and tracking result of the previous frames is used as a mask or initial contour for the current frame along with location of the mask and so on. The energy minimization function in eq.23 is employed to process the level sets and to produce an optimal contour for segmentation and tracking for each frame. In all video sequences initial contour is cropped manually that can be placed near to the object of interest.

In the first experiment we started with a video sequence which is shot in a lab using web cam with dark background and with an additional constraint that signer should also wear a dark shirt. This video sequence is part of the database we have created for sign language recognition project. Our Sign language database consists of 351 signs with 8 different signers. Figure 4 (frame size 320×480) shows our segmentation and tracking algorithm with values of $\alpha = 0.3, \beta = 0.5$ and $\gamma = 0.2$. The object and background clusters are made of three and two clusters.

The experiments are performed in R colour plane. As such we can do it any colour plane or with colour videos. The problem with full colour sequences pertaining to sign language is that the sign language videos contain large sequence of frames with lot of information to be extracted. Fig. 4(a) shows four frames from a video sequence of the signer performing a sign related to English alphabet 'X'. This simple sign contains 181 frames. Fig. 4(b) shows the results obtained from our proposed method. The inclusion of prior shape term in the level sets segments the right finger shape in spite of being blocked by the finger from left hand. This segmentation and tracking result will help in good sign recognition. Fig. 4(b) and 4(c) shows the effectiveness of the proposed algorithm against the Chan-Vese (CV) model in [50].

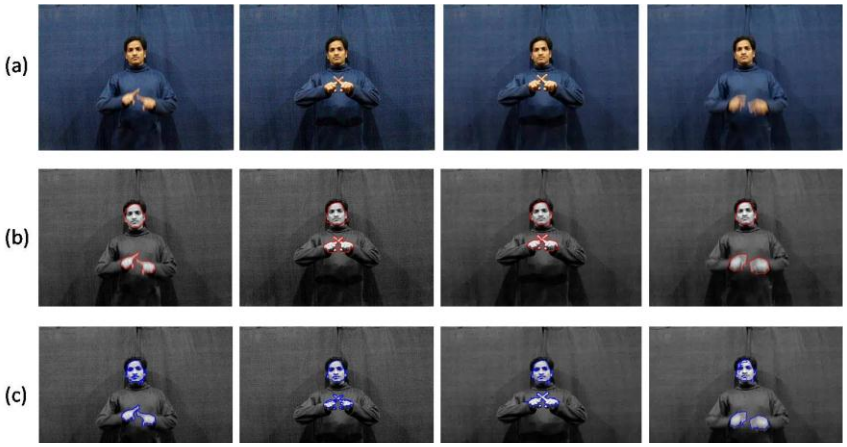

Figure 4. Experiment one showing our proposed segmentation algorithm on sign language videos under laboratory conditions. Frames 10, 24, 59,80 are shown. Row (a) shows four original frames. Row (b) shows the results from proposed algorithm and Row (c) Shows results of CV algorithm in [50].

The results of segmentation and tracking are fused into a feature vector and a database is created using 351 signs with 4 different signers. The size of feature matrix f^{NMat} for the neural network is 1404×80. The Neural network can classify 351 signs. Hence the target vector of neural network is 351×80 .

The neural network, used, is feed forward and back propagation network, which has 1404 instances of input and 351 instances of output neurons and 652 neurons in the hidden layer as shown in fig. 5. The weights and bias for each neuron are initialized randomly and network is ready for training. The training process requires a set of examples of proper network behavior, network inputs and target outputs. During training the weights and biases of the network are iteratively adjusted to minimize the network performance function, which in case of feed forward networks, is mean square error (MSE).

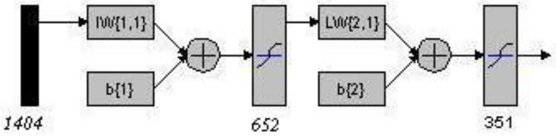

Figure 5. Neural Netwok used for testing sign videos with simple backgrounds.

The neural network shown in fig. 5 is trained for simple backgrounds using four sets of sign videos for each gesture consisting of different signers. The epochs versus error graph for training the network is shown in fig. 6.

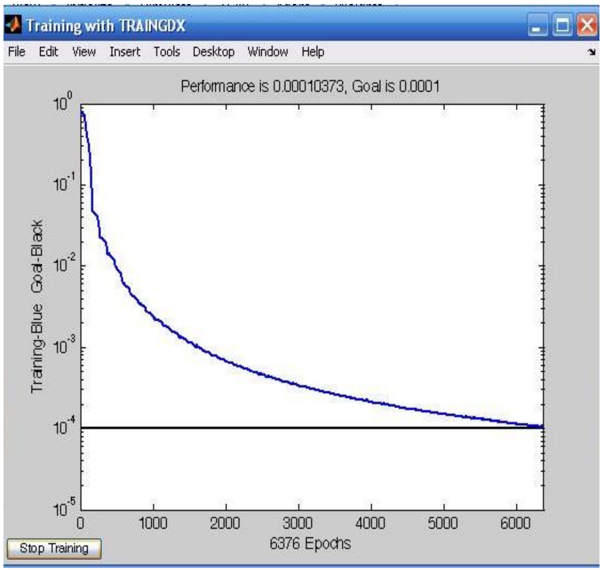

Figure 6. Neural Netwok Training graph for simple backgrounds with four samples per gesture.

The recognition rate is defined as the ratio of the number of correctly classified signs to the total number of signs:

$$\text{Re}\,cognition\ Rate(\%)=\frac{Perfectly\ Classified\ Signs}{Total\ Signs\ Used}\times 100 \quad (36)$$

TABLE I. RESULTS OF TESTING USING GESTURES WITH SIMPLE BACKGROUNDS

Number of input neurons : 1404
Number of hidden neurons: 652
Number of output neurons: 351
Activation Function: sigmoid
Learning rate: 0.05 Momentum Factor: 0.9
Error Tolerance: 0.0001
Number of training samples used: **1404**
Number of testing samples: **2106**

Data	Total Number of Samples	Correctly Recognized samples	Performance Rate of the Neural Network (%)
Training With 4 Samples/Gesture			
Training	**1404**	**1404**	100%
Testing	**2106**	**2009**	95.39%
Total	3510	122	97.23%

The recognition rate is defined as the ratio of the number of correctly classified signs to the total number of signs:

$$\text{Re}\,cognition\ Rate(\%)=\frac{Perfectly\ Classified\ Signs}{Total\ Signs\ Used}\times 100 \quad (36)$$

Table. I gives the results of testing the proposed network. For simple backgrounds the neural network is classifying signs with a recognition rate of 97.23%.

We observe occlusions of hands and head very frequently in sign language videos. Most sign language recognition systems insist that the signer should face the camera directly to avoid occlusions of hands largely. This problem is solved using our proposed level set method. It is shown in fig. 7. We initialized contour for only right hand of the signer in the first frame.

With the left hand coming in the path of right hand as can be observed from the original sequence in figure 7(a), it's difficult to segment and track the shape of right hand. But the results in figure 7(b) show the segmentation and tracking of right hand only with occlusions from left hand. This breakthrough will help in designing sign language systems with utmost robustness.

In the next experiment, the supremacy of our proposed technique is shown, where the video sequence contains fast moving objects in contrast to hand and head movements. The video sequence is shot on an Indian road and in the natural environment. Fig. 8 shows the original sequence in column (a) along with the results of CV model [50] in column (b) and our method in column (c).

Table. I give the results of testing the proposed network. For simple backgrounds the neural network is classifying signs with a recognition rate of 97.23%.

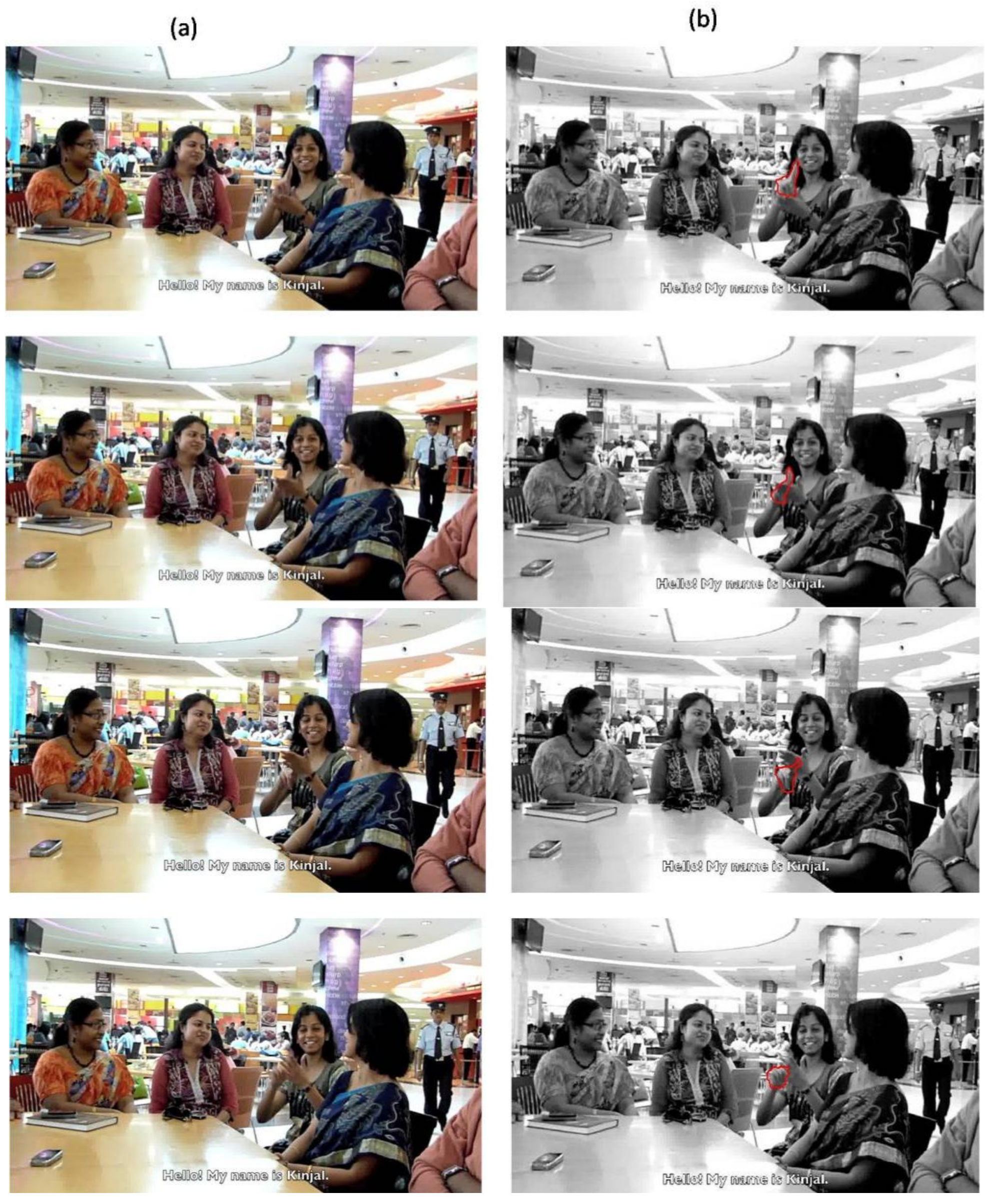
(a)
(b)
Hello! My name is Kinjal.
Hello! My name is Kinjal.
Hello! My name is Kinjal.
Hello! My name is Kinjal.
Hello! My name is Kinjal.
Hello! My name is Kinjal.
Hello! My name is Kinjal.
Hello! My name is Kinjal.

Figure 7. Showing the influence of prior shape knowledge. Here only the occluded right hand is segmented and tracked. Column (a) shows original image sequence and column (b) the segmentation and tracking result.

Observation of third row exposes the disadvantage associated with CV method without prior shape term in energy level set for tracking. In this video (see Fig. 8) the background object suddenly appears in the frame to which the method in [50] technique provides much resistance and the final segmentation and tracking result include the suddenly appeared object. But providing prior shape information along with object colour, texture and boundary edge map proves the strength of our method. Also, we get the unwanted tracks in the form of leaves of trees in the background for left hand of the signer in row three for the method used in [50], which is not an issue with our method.

Using the sequences in fig.7 and fig.8, which represents a more real time scenario for testing our proposed system, we created a data base of around 110 signs. These videos are taken under different background clutters like in fig. 7 and fig. 8. We have chosen two samples of each sign for training the network and two signs of similar gesture for testing the network. The proposed network is shown in fig. 8 containing 220 input neurons, 181 hidden neurons and 110 output neurons. The video sequences, used here, contain continuous gestures with cluttered backgrounds, such as offices, on road, restaurants, and videos from [33].

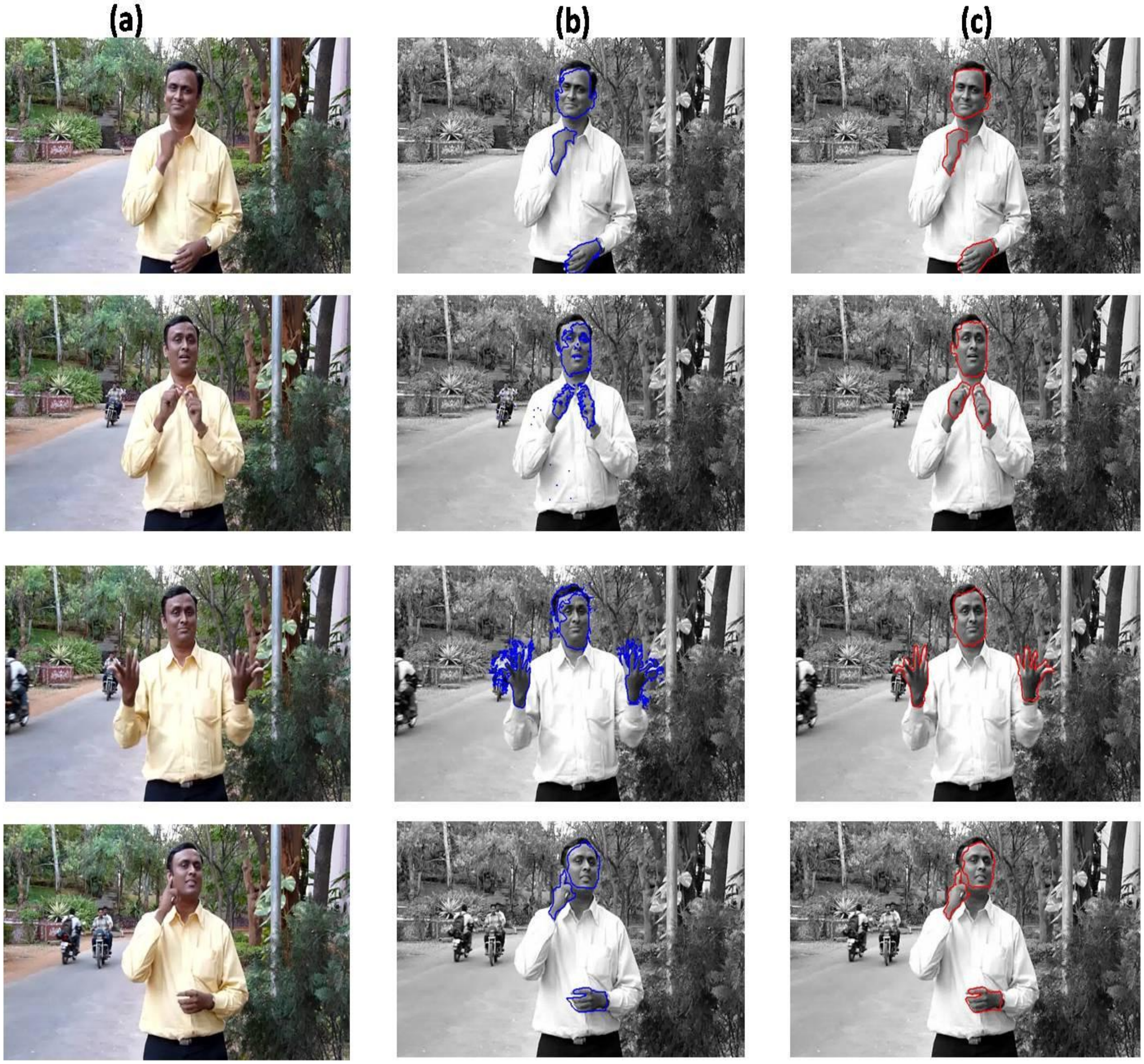

Figure 8. Frames of a sign video sequence on a Indian road and under natural environment. Column (a) is original sequence of frames 39, 54,79 and 99. Column (b) method used in [50] and Column (c) our proposed tracking method.

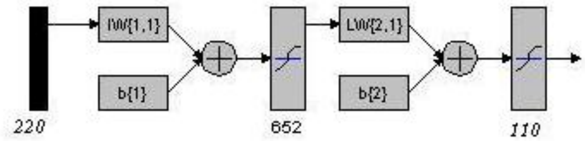

Figure 9. Neural Netwok Architecture for cluttered backgrounds with two samples per gesture.

The neural network is trained with a feature matrix extracted from sign videos with complex backgrounds. The training graph for the network in fig.9 is shown in fig.10. We observe that even though the input has two sets of signs for each gesture, it took more number of iterations to train due to occlusions and cluttered background.

Table. II gives the results of testing the proposed network. For complex and variable backgrounds, the neural network is classifying signs with a recognition rate, of which is slightly less than the recognition rate for simple backgrounds i.e. 95.68%.

In the third and final experiments, the proposed neural network is trained using a combination of two signs from simple backgrounds and two signs from complex backgrounds for each sign. A total of 200 signs are trained four from each gesture. The neural network employed for this task consists of 200 input neurons, 652 hidden neurons and 50 output neurons to classify 50 signs and convert them in to audio and text. The network architecture for this job is shown in fig. 10.

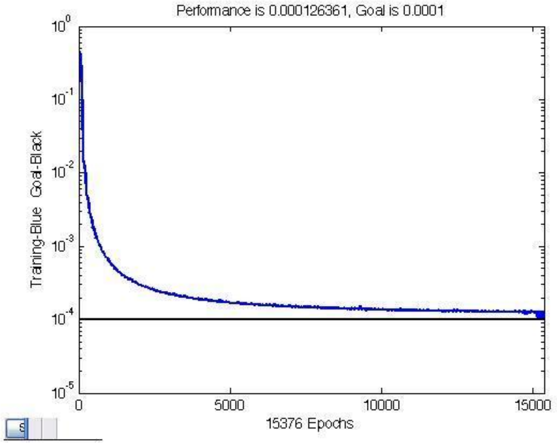

Figure 10. Neural Netwok Training graph for cluttered backgrounds with two samples per gesture.

TABLE II. RESULTS OF TESTING, USING GESTURES WITH CLUTTERED BACKGROUNDS

Number of input neurons : 220 Number of hidden neurons: 652 Number of output neurons: 110 Activation Function: sigmoid Learning rate: 0.05 Momentum Factor: 0.9 Error Tolerance: 0.0001 Number of training samples used: **220** Number of testing samples: **220**			
Data	Total Number of Samples	Correctly Recognized samples	Performance Rate of the Neural Network (%)
Training With 2 Sample/Gesture			
Training	220	220	100%
Testing	220	201	91.5%
Total	440	421	95.68%

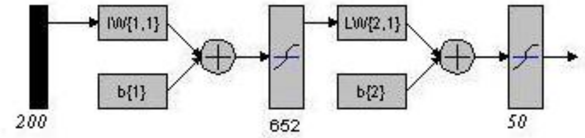

Figure 11. Neural Netwok Architecture for cluttered backgrounds with two samples per gesture along with two samples from simple backgrounds.

The neural network is trained with a feature matrix, combining the feature matrix from simple backgrounds and feature matrix from complex backgrounds. The training graph produced for this feature matrix is shown in fig. 12.

Table. III gives the results of testing the proposed network. For complex backgrounds and simple backgrounds combined the resultant recognition rate, and has slightly decreased due to shape occlusions for longer periods of time. This can be avoided by using regularly updating the shape module after losing shape information during longer occlusion period. For all our experiments, if the occlusion period is more than 10 frames, we have to reinitialize the contour. The recognition rate for our third experiment is 85%.

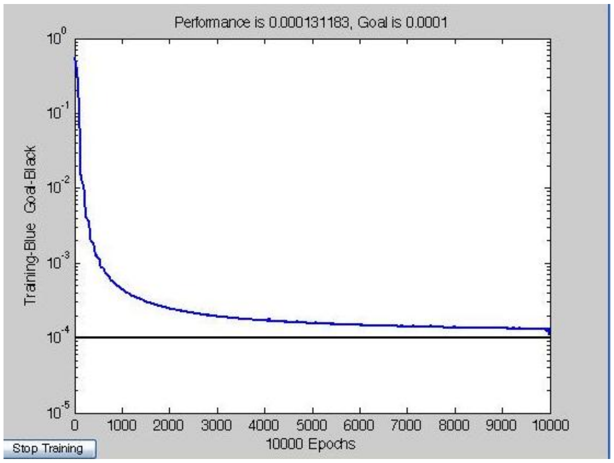

Figure 12. Neural Netwok Training graph for cluttered as well as simple backgrounds with each two samples per gesture.

TABLE III. RESULTS OF TESTING USING GESTURES WITH SIMPLE CLUTTERED BACKGROUNDS

Number of input neurons : 200 Number of hidden neurons: 652 Number of output neurons: 50 Activation Function: sigmoid Learning rate: 0.05 Momentum Factor: 0.9 Error Tolerance: 0.0001 Number of training samples used: **200** Number of testing samples: **500**			
Data	Total Number of Samples	Correctly Recognized samples	Performance Rate of the Neural Network (%)
Training With 4 Sample/Gesture			
Training	200	190	95%
Testing	200	150	75%
Total	400	340	85%

VII. CONCLUSIONS

This paper carries us slightly closer to building sign language recognition system that performs well under natural backgrounds. The proposed method combines effectively the colour, texture, boundary and prior shape information to produce an effective video segmentation and tracking of sign language videos under various harsh environments such as cluttered backgrounds, poor lighting, fast moving objects and occlusions. The colour and texture information is extracted statistically by creating a feature vector and classifying each pixel in the frame to object and background pixel. Boundary information is provided by divergence operator along with the curvature of the object under consideration. Including shape information from the previous frame it is done a whole lot of difference to the level set minimization to segment correctly and track effectively the occulted hand from other hand and also head some times. We have effectively demonstrated by experimentation of the proposed method by applying it to video sequences under various conditions. Nevertheless, there are challenges which are to be addressed to apply this method to continuous sign language recognition systems to carry out the recognition in real time. The recognition rate of the overall sign language recognition system is 93% which is a good score for our proposed real time system.

The segmentation experiments are run on an Intel core i3 2.5GHz processor with 3GB RAM using MATLAB software. The average running time was 3 frames per second with around 40 level set iterations per frame. The resolution of the video is kept to minimum along with colour information which in this paper was restricted to only one colour plane. This is to reduce the computation time which otherwise increases for higher resolutions and full colour processing. The speed of the algorithm can be increased by using fast numerical schemes for determining the level sets as given in [51].

REFERENCES

[1] Imagawa I et. al, 'Recognition of local features for camera-based sign language recognition system', Proc. ICPR'00, pp849-853, 2000.

[2] P.V.V.Kishore, P.Rajesh Kumar, E.Kiran Kumar & S.R.C.KIshore . "Video Audio Interface for Recognizing Gestures of Indian Sign Language" *International Journal of Image Processing (IJIP), CSC Journals,* Vol. 5, No.4, pp479-503, Sep 2011

[3] Yang M, Ahuja N, 'Recognizing hand gesture using motion trajectories',CVPR'99, pp892-897, 1999.

[4] W. Stokoe, D. Casterline, and C. Croneberg. A Dictionary of American Sign Language on Linguistic Principles. *Gallaudet College Press, Washington D.C.*, USA, 1965.

[5] U. R. Wrobel. Referenz in eb¨ardensprachen: Raum und Person. Forschungsberichte des Instituts f¨ur Phonetik und Sprachliche Kommunikation der Universit ¨at M¨unchen, 37:25–50, 2001.

[6] C. Vogler and D. Metaxas. A framework for recognizing the simultaneous aspects of american sign language. Computer Vision & Image Understanding, 81(3):358–384, March 2001.

[7] Ravikiran J, Kavi Mahesh, Suhas Mahishi, Dheeraj R, Sudheender S, and Nitin V Pujari," Finger Detection for Sign Language Recognition", Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2009), vol. I, Hong Kong, March 2009.

[8] J. Mackie ,B. McCane, "Finger Detection with Decision Trees", University of Otago, Department of Computer Science.

[9] Claudia N¨olker, Helge Ritter, "Detection of Fingertips in Human Hand Movement Sequences", Gesture and Sign Language in Human-Computer Interaction, Proceedings of the International Gesture Workshop, Published by Springer, 1997, pp. 209-218.

[10] E. Stergiopoulou, N. Papamarkos, "Hand gesture recognition using a neural network shape fitting technique", Engineering Applications of Artificial Intelligence (ELSEVIER), vol. 22,pp.1141–1158, 2009, doi: 10.1016/j.engappai.2009.03.008.

[11] Mehdi, S.A.; Khan, Y.N. "Sign language recognition using sensor gloves**"** Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02. Volume 5, pp18-22, Nov. 2002.

[12] W.Kadous. Recognition of Australian Sign language using instrumented glove. Master Thesis. University of New South Wales. October 1995.

[13] Tan Tian Swee, A.K. Ariff, Sh-Hussain. Salleh, Siew Kean Seng, and Leong Seng Huat, "*Wireless Data Gloves Malay Sign Language Recognition System"* Proceedings of the international conference on Information and computing Systems, 2007, ICICS'07,pp56-59.

[14] Collins, R.T., Liu, Y., Leordeanu, M. (2005) "*Online selection of discriminative tracking features"*. IEEE Trans. Pattern Anal. Mach. Intell. Vol.27 No.10 , pp1631–1643.

[15] Ginmo Chung and Luminita Vese. (2003) "Image segmentation using a multilayer level set approach.Technical Report" 03-53, UCLA.

[16] R. Malladi, J.A. Sethian, and B.C. Vemuri. (1995) *"Shape modeling with front propagation: A level set approach*". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.17 No. 2,pp158-175.

[17] D. Terzopoulos and K. Fleischer (1988) "*Deformable models"*. The Visual Computer, Vol.4 No.6, pp306-331.

[18] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer.(1987) "*Elastically deformable models*" In Comp.Graphics Proc., pages 205-214. ACM Press/ ACM SIGGRAPH.

[19] Osher, S., Sethian, J. (1988) Fronts propagating with curvature-dependant speed: algorithms based on Hammilton-Jacobi formulations.J. Comput. Phys. Vol.79 No.1, pp12–49.

[20] L. Brkthes, P. Menezes, E Lerasle and J. Hayet "Face tracking and hand gesture recognition for human-robot interaction" Pmcwdingr of the 2004 IEEE InternatIona1 Confennee on RobotIco 6 Automation New Orleans. April 2004.

[21] Mansouri.A.R, Region tracking via level set pdes without motion computation. IEEE Trans. Pattern Anal. Mach. Intell. 2002, Vol.24, no.7, 947–961.

[22] Jehan-Besson, S, Barlaud, M.,Aubert,G.: Detection and tracking of moving objects using a new level set based method. Proceedings of IEEE International Conference on Pattern Recognition, Barcelona, Spain, 3–8 September, 2000, pp. 7112–7117.

[23] Newby, G. (1993). Gesture recognition using statistical similarity. In Proceedings of virtual reality and persons with disabilities.

[24] Birk, H., Moeslund, T. B., & Madsen, C. B. (1997). Real-time recognition of hand alphabet gestures using principal component analysis. In Proceedings of the 10th Scandinavian conference on image analysis.

[25] Rangarajan, K., & Shah, M. (1991). Establishing motion correspondence. CVGIP: Image Understanding, 54, 56–73.

[26] Dorner, B., & Hagen, E. (1994). Towards an American sign language interface. Artificial Intelligence Review, 8(2–3), 235–253.

[27] Fels, S. (1994). Glove-TalkII: mapping hand gestures to speech using neural networks—an approach to building adaptive interfaces. PhD thesis, Computer Science Department, University of Toronto.

[28] Fukushima, K. (1989). Analysis of the process of visual pattern recognition by the neocognitron. Neural Networks, 2, 413–420.

[29] Haykin, S. (1999). Neural networks: a comprehensive foundation. Prentice Hall.

[30] Charniak, E. (1993). Statistical language learning. Cambridge: MIT Press.

[31] Starner, T. (1995). Visual recognition of American sign language using hidden Markov models. Master's thesis, Massachusetts Institute of Technology.

[32] Seymore, K., McCallum, A., & Rosenfeld, R. (1999). Learning hidden Markov model structure for information extraction. AAAI 99 workshop on machine learning for information extraction.

[33] www.deafsigns.org , Indian sign Language Dictionary.

[34] M.Kass, A Witkin, D Terzopoulos ,"Snakes:active Contour Models", *Int. J. of Computer Vision,* 1987,pp 321-331.

[35] R. Malladi, J.A. Sethian, and B.C. Vemuri. *"Shape modeling with front propagation: A level set approach*". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.17 No. 2, 1995, pp158-175.

[36] Tuceryan, M., Jain, A, Texture analysis. In: Chen, C.H., Pau, L.F., Wang, P.S.P. (eds) The Handbook of Pattern Recognition and Computer Vision, 2nd edn. chap. 2.1, 207–248.World Scientific Publishing, Singapore.

[37] Allili, M.S., Ziou, D.,(2006), *"Automatic colour-texture image segmentation by using active contours"*. In: Proceedings of 1st IEEE International Workshop on Intelligent Computing in Pattern Analysis/Synthesis, Xi'an, China, 26–27, LNCS 4153, pp. 495–504.

[38] S. Allam, M. Adel, P. Refregier, "Fast algorithm for texture discrimination by use of a separable orthonormal decomposition of the co-occurrence matrix," Applied Optics, vol.36, 1997, pp.8313–8321.

[39] R. M. Haralick, K. Shangmugam, I. Dinstein, (1973) *"Textural Feature for Image Classification,"* IEEE Trans on Systems, Man, Cybernetics, 3(6), pp.610—621.

[40] McLachlan, G., Peel, D, Finite Mixture Models, Wiley Series in Probability and Statistics (2000).

[41] A Dempster, N. Laird, and D. Rubin. “Maximum likelihood from incomplete data via the EMalgorithm,” *Journal of the Royal Statistical Society*, vol. 39, 1977, 1-38.

[42] Allili, M.S., Ziou, D.: Automatic colour-texture image segmentation by using active contours. In: Proceedings of 1st IEEE International Workshop on Intelligent Computing in Pattern Analysis/Synthesis, Xi’an, China, 26–27, August 2006, LNCS 4153, pp. 495–504.

[43] Cremers, D., Soatto, S.: A pseudo-distance for shape priors in level set segmentation. In: Proceedings of 2nd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision, Nice, France, 13–16 October 2003, pp169–176.

[44] K.Seshadrinnathan and A.Bovik, “Motioin Tuned Spatio-Temporal quality assesment of quality videos” IEEE Trans on Image Processing, vol.19, No.2,pp335-350,Feb.2010.

[45] Braik, M., Sheta, A., "A New Approach for Potentially Breast Cancer Detection Using Extracted Features and Artificial Neural Networks", Journal of Intelligent Computing Vol. 2, No. 2, June 2011, pp.54-71.

[46] Honjun Lu., Rudy S., huan lui., 1996, Effective data mining Using Neural Networks, IEEE Transactions On Knowledge and Data Engineering, Vol. 8, No. 6, , pp. 957-961.

[47] Radha, V., Nallammal., N, October 19-21, 2011, "Neural Network Based Face Recognition Using RBFN Classifier" In Proceedings of the World Congress on Engineering and Computer Science, Vol. I, San Francisco, USA.

[48] Darwis, Y. and, Sheta, A., 2008, " Minutiae Extraction for Fingerprint Recognition” In Proceedings of the Fifth IEEE International Multi-Conference on System, Signal and Devices (SSD'08), Amman, Jordan.

[49] Michael, N., 2005 Artificial Intelligence: A Guide to Intelligent Systems, Pearson Education Limited.

[50] T. Chan and L. Vese(2001) “*Active contours without edges*”. IEEE Transations on Image Processing,Vol. 10 No.2,pp266-277.

[51] Weickert, J., K ¨ uhne, G.: Fast methods for implicit active contour models. In: Osher, S., Paragios, N.: (eds) Geometric Level Set Methods in Imaging, Vision and Graphics, chap. 3, pp. 44–57. Springer, Heidelberg.

AUTHORS PROFILE

P.V.V.Kishore (SMIEEE’07) received his M.Tech degree in electronics from Cochin University of science and technology in the year 2003, and currently pursuing PhD at Andhra University College of engineering in Department of ECE from 2008. He is working as research scholar at the Andhra university ECE department. He received B.Tech degree in electronics and communications engineering from JNTU, Hyd. in 2000. His research interests are digital signal and image processing, computational intelligence, human computer interaction, human object interactions. He is currently a student member of IEEE.

Dr. P.Rajesh Kumar (MIEEE’00, FIETE’09) received his Ph.D degree from Andhra University College of Engineering for his thesis on Radar Signal Processing in 2007. He is currently working as associate professor at Dept. of ECE, Andhra University College of engineering, Visakhapatnam, Andhra Pradesh. He is also Assistant Principal of Andhra University college of Engineering, Visakhapatnam, Andhra Pradesh. He as produced numerous research papers in national and international journals and conferences. He has guided various research projects. His research interests are digital signal and image processing, computational intelligence, human computer interaction, radar signal processing.