Optical Flow Hand Tracking and Active Contour Hand Shape Features for Continuous Sign Language Recognition with Artificial Neural Networks

P.V.V.Kishore and M.V.D.Prasad
K. L. University, E.C.E.
pvkishore@kluniversity.in, mvd_ece@kluniversity.in

Abstract

To extract hand tracks and hand shape features from continuous sign language videos for gesture classification using backpropagation neural network. Horn Schunck optical flow (HSOF) extracts tracking features and Active Contours (AC) extract shape features. A feature matrix characterizes the signs in continuous sign videos. A neural network object with backpropagation training algorithm classifies the signs into various words sequences in digital format. Digital word sequences are translated into text with matching and the sutting text is voice translated using windows application programmable interface (Win-API). Ten signers, each doing sentences having 30 words long tests the performance of the algorithm by computing word matching score (WMS). The WMS is varying between 88 and 91 percent when executed on different cross platforms on various processors such as Windows8 with Intel i3, Windows8.1 with intel i3 and windows10 with intel i3 running MATLAB13(a).

Keywords: Sign Language Recognition, Horn Schunck Optical Flow, Hand Tracking, Active Contour Models, Shape Extraction, Artificial Neural Networks.

1. Introduction

Learning skills of a hearing hindered person are seriously hampered because of the missing hearing sense. From here, a mute person has to depend largely on visual sense and any learning and communication aids will help them learn faster and communicate better. Usually human interpreter trained in sign language understanding acts as a bridge between the normal people (with hearing sense) and mute people (without or with low hearing sense).

The difficulties faced by deaf and elderly community when moving alone and wanted to mingle publicly at government offices, schools, shopping malls and hospitals are indescribable. Having a human interpreter accompanying the deaf person always in a country like India is impractical due to few trained sign language interpreters.

The problem statement at a broader level faced by researchers in sign language recognition is expressed in the Figure 1.

Empowerment of the hearing impaired can be achieved using a mobile based application that can understand sign language and translate it to speech and conversely. The solution proposed is to develop a machine interpreter that can help deaf people to voice themselves at any location in the presence of people with hearing sense.

This machine translation of sign language acts an interpreter between these two groups of humans i.e., with and without hearing sense, intendeds to replace the human interpreter with machine interpreter. Sign language recognition is a major research area that encompasses video image analysis, shape extraction, feature optimization and pattern classification working in tandem to convert video signs into text and voice messages.
Figure 1. Translation of Sign Language into Speech and Vice Versa as a Communication Medium between the Deaf and Normal People

Previous research in the area show various methods applied to achieve this objective and to a certain extent achieved by most of the researchers. The entire sign language recognition follows three major methods, tracking sensors [1], glove based sensors [2] and visual sensors [3, 4]. The most widely used and most difficult one is sign language recognition using visual sensors and video cameras.

The camera based SLR has two operating modes. The operating techniques are static (image based) and dynamic (video based). In video based the researchers focused on discrete videos with an average running time of 2 to 4 seconds, churning out a frame rate of 30fps. A few researchers implemented the same for continuous signs, where the video lasts for around 3-4 minutes at 30fps putting 7,200 frames for processing.

Early sign language recognition concentrated on articulatory units called phonemes [5]. Rationally sign language is understood as a set of dialectal analysis of hands tracking, hands shapes, hands locations, sign articulation, head orientations and facial expressions.

Research on real time American Sign Language recognition shows a wearable computer based video camera [3] using Hidden Makov Model (HMM) recognizes continuous American Signs with good precision. Four HMM states try to capture the signs of American Sign Language producing good recognition rates. But the stability of the method is compromised when the signer changes in the video sequence making it signer dependent.

To make the system more robust hand shapes and hand trajectories are extracted to recognize static and dynamic hand signs from Indian sign language [6]. The authors propose a video object abstraction model for segmentation and tracking by dividing the video objects into constituent planes and hand is considered as an object. The proposed system can classify and recognize static, dynamic gestures along with sentences with superior consistency.

A real time model for Malaysian sign language translation with a color segmentation module has managed a recognition rate of around 90%[7]-[8]. Hands and face segmentation with color and motion cues from the one second content based sign video representations of signers.

A signer adaptation model that combines maximum of a posteriori and iterative vector field smoothing [9] which reduces the amount of data to represent a video sign. This method has achieved good recognition rate of around 89% for continuous signs. Review shows a variety of sign language recognition systems with statistical approaches [10], example based approaches [11], finite state transducers [12] showing higher recognition rates close to 90%.
Continuous sign recognition from videos calls for solving some major issues related to recognition. Two such issues that are being addressed in this work are hand tracking and hand shape analysis which constitute around 70% to SLR. Remaining 30% is related to movement emptiness, facial tracking and sign corpus identification.

The authors of this paper has done a considerable amount of work previously related to static gesture classification for Indian Sign Language with static and dynamic sign videos[13-14]. For sign segmentation of video frames the authors used wavelet based image fusion of canny edge operator and morphological differential gradient. Elliptical Fourier descriptors are used to model hand shapes and head portion. They have tested for 80 static signs with neural network classifier [15] and fuzzy inference engine [16] respectively. The percentage of recognition achieved is 87% when neural network classifier is used and 90% when fuzzy inference engine is used as classifier.

Yikai Fang et. al., [17] proposed a real time adaptive hand gesture segmentation and tracking using motion and color cues with 2596 frames recorded for 6 Gestures. They reported 98% recognition rate for simple backgrounds and around 89% recognition for cluttered video backgrounds. Each frame was processed for around 90 to 110 milliseconds.

G. Fang et. al., [18] has developed signer independent continuous sign language recognition HMM, Self-Organizing Maps and Recurrent Neural Networks. They used 208 signs form Chinese sign language and reported a recognition rate of around 92%.

Xiying Wang [19] incorporated tracking of deformable human hand for recognizing gestures in sign language. They showed that by incorporating tracking into a sign language recognition system will improve the performance of the classifier [20].

This research proposes a sign language recognition system with hand tracking and shape analysis that builds the feature vector for the classifier. Here we use artificial neural network as a classifier which is trained with error backpropagation algorithm.

2. Tracking with Optical Flow – Revisit

All Optical flow algorithms maps the spatial displacements in consecutive frames into velocity vectors with imposed constraints from illumination and displacements. Horn schunck [21] optical flow (HSOF) is considered for tracking hands in continuous sign videos under simple backgrounds. The optical flow constraint equation at each pixel location in two consecutive video frames $f^n(x, y, t)$, $f^{n-1}(x, y, t)$ with illumination smoothness constraint which says the hands intensity does not change with displacement,

$$f(x + dx, y + dy, t + dt) = f(x, y, t)$$

Where $(x, y)$ are spatial locations of objects in the frames and $t$ is the time variation between frames. By expanding the left side of eq’n(1) using Tylor’s series

$$f(x + dx, y + dy, t + dt) = f(x, y, t) + \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy + \frac{\partial f}{\partial t} dt + \hat{H} (\bullet)$$

$$f(x, y, t) + \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy + \frac{\partial f}{\partial t} dt + \hat{H} (\bullet) = 0$$

$$\frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy + \frac{\partial f}{\partial t} dt + \hat{H} (\bullet) = 0$$

The final optical flow constraints equation can be obtained by dividing eq’n(3) by $dt$ to get eq’n(4)
\[
\frac{\partial f}{\partial x} \frac{dx}{dt} + \frac{\partial f}{\partial y} \frac{dy}{dt} + \frac{\partial f}{\partial t} = 0
\]  
\hspace{1cm} (4)

The terms \(dx/dt\) and \(dy/dt\) are represented with velocity vectors \(u^x\) and \(v^y\) along spatial \(x\) and \(y\) directions. Eq’n (4) has two unknowns with only one eq’n to be solved. This eq’n is modeled by Horn and Schunck on the flow field \(\Gamma\) of optical velocities on bounded image domain \(\lambda\) by minimizing the following optical flow constraints equation modeled as

\[
\Gamma \rightarrow E(\Gamma / f) = \frac{1}{2} \int_\lambda \left( \langle \nabla f, \Gamma \rangle + \frac{\partial f}{\partial t} \right)^2 \, dx \, dy + \frac{k}{2} \int_\lambda \left( \| \nabla u \|^2 + \| \nabla v \|^2 \right) \, dx \, dy
\]  
\hspace{1cm} (5)

Where \(u\) and \(v\) are the coordinate functions of \(\Gamma\) and \(k\) is a positive constant to weigh the relative contribution of the two terms of the functional. Eq’n (5) can be restructured as Euler-Lagrange equations as

\[
f_{x}(f_{x}u + f_{y}v + f_{t}) - k\nabla^2 u = 0
\]
\[
f_{x}(f_{x}u + f_{y}v + f_{t}) - k\nabla^2 v = 0
\]  
\hspace{1cm} (6)

On the boundary \(\partial \lambda\) of \(\lambda\) with the Neumann boundary conditions as shown

\[
\frac{\partial u}{\partial n} = 0, \quad \frac{\partial v}{\partial n} = 0
\]  
\hspace{1cm} (7)

Where \(n\) is the normal to the boundary \(\partial \lambda\) and \(\partial/\partial n\) is the differentiation operator in the direction of the normal. If \(\lambda\) is discretized by superimposing a unit-spacing grid \(G\) having grid points indexed by the integers \(\{1, 2, \ldots, N\}\), numbering from top-to-down and left-to-right. For all grid points \(G\) the indices \(i \in \{1, 2, \ldots, N\}\), we have the Euler-Lagrange equations in (6) as a discrete approximation

\[
f_{x}u_{i} + f_{y}v_{i} + f_{t} - k \sum_{j \in N_{i}} (u_{j} - u_{i}) = 0
\]
\[
f_{x}u_{i} + f_{y}v_{i} + f_{t} - k \sum_{j \in N_{i}} (v_{j} - v_{i}) = 0
\]  
\hspace{1cm} (8)

Where \((u_{i}, v_{i})\) is \(i^{th}\) grid optical velocity vectors denoted by \(I_{x}, I_{y}\) and \(I_{t}\). Let \(N_{i}\) be the indexed set of neighborhood pixels around the \(i^{th}\) grid. Here we choose 8-neighbours for \(i^{th}\) grid point denoted by \((N_{i}) = 8, \eta_{i} = \text{pos}(N_{i})\), the following linear equations for \(i^{th}\) grid point is

\[
(f_{x}^2 + k\eta_{i})u_{i} + f_{x}f_{y}v_{i} - k \sum_{j \in N_{i}} u_{j} = -f_{x}f_{t}
\]
\[
f_{x}f_{y}u_{i} + (f_{y}^2 + k\eta_{i})v_{i} - k \sum_{j \in N_{i}} v_{j} = -f_{y}f_{t}
\]  
\hspace{1cm} (9)
For an vector \( \mathbf{x} \in \mathbb{R}^{2N} \) with coordinates \( x_{2i-1} = u_i, x_{2i} = v_i \) and an arbitrary vector \( \mathbf{b} \in \mathbb{R}^{2N} \) with coordinates \( b_{2i-1} = -f_{x_i}u_i, b_{2i} = -f_{y_i}v_i \) for all \( i = 1 \) to \( N \), the linear system is modelled as

\[
Ax = b
\]  
(10)

Where \( A \in \mathbb{R}^{2N \times 2N} \) matrix having elements, \( A_{2i-1,2i-1} = f_{x_i}^2, A_{2i-1,2i} = f_{x_i}^2 + k\eta_i, A_{2i,2i-1} = A_{2i,2i-1} = f_{x_i}f_{x_i}, \) and \( A_{2i-1,2j-1} = A_{2i,2j} = -K \forall i, j \in \{1 \ldots \ldots N\} \) such that \( j \in N_i \), and all other elements being zero.

The matrix \( A \) is a symmetric and positive definite matrix. Assuming matrix \( A \) as nonsingular matrix equation 10 is solved pointwise and block wise using efficient convergent methods in numerical linear algebra [22, 23]. Jacobi iterations for a \( 2 \times 2 \) block division matrix are used to estimate velocity vectors iteratively in Horn and Schunk optical flow using

\[
u^{K+1}_i = \frac{f_{y_i}^2 + K\eta_i}{\eta_i(f_{x_i}^2 + f_{y_i}^2) + K\eta_i^2} \sum_{j \in N_i} u_j
\]

\[-\frac{f_{x_i}f_{y_i}}{\eta_i(f_{x_i}^2 + f_{y_i}^2) + K\eta_i^2} \sum_{j \in N_i} v_j
\]

\[
u^{K+1}_j = \frac{f_{x_i}^2 + K\eta_i}{\eta_i(f_{x_i}^2 + f_{y_i}^2) + K\eta_i^2} \sum_{j \in N_i} v_j
\]

\[-\frac{f_{x_i}f_{y_i}}{\eta_i(f_{x_i}^2 + f_{y_i}^2) + K\eta_i^2} \sum_{j \in N_i} u_j
\]

(11)

\[
u^{K+1}_j = \frac{f_{x_i}^2 + K\eta_i}{\eta_i(f_{x_i}^2 + f_{y_i}^2) + K\eta_i^2} \sum_{j \in N_i} v_j
\]

\[-\frac{f_{x_i}f_{y_i}}{\eta_i(f_{x_i}^2 + f_{y_i}^2) + K\eta_i^2} \sum_{j \in N_i} u_j
\]

\[-\frac{f_{x_i}f_{y_i}}{\eta_i(f_{x_i}^2 + f_{y_i}^2) + K\eta_i^2} \sum_{j \in N_i} v_j
\]

(12)

With the Jacobi method, the above velocity vector equations in x and y directions are updated for all the pixels in the video frame. The final velocity vectors are collected in \( (u_i, v_i) , \forall i \in \{1 \ldots 2N\} \) using the significant differences between video frames.

3. Shape Segmentation with Level Sets – Revisit

Active contours are a class of model based image segmentation algorithms based on total variational methods [24, 25]. Variational methods define a solution space for the problem and builds a mathematical model that becomes linear during optimization process. First models were introduced by Terzopoulos [26, 27]. An initial smooth contour that deforms itself actively towards object edges in the image resulting in a solution space consisting of a boundaries of objects in the image.

Two initial conditions while defining the snake active contour model are, the solution space image should be very much similar to original image and they should also exhibit spatial smoothness. For certain class of images this works extremely well. But as the
problem domain increases the snakes model gives unstable solutions for small changes in pixel values. The stability of the active contours is increased by using the concepts of level sets [28], which can handle object deformities automatically.

Most of the active contours end their growth based on image gradients. Chan and Vese (CV Model) [29] model uses level sets and the growth of the curve is controlled by Mumford-Shah distance [30]. CV Model for level sets does not necessarily consider gradient for stopping the curve evolution.

The active contours are elastic models of continuous, flexible curve that is imposed upon and matched to the image by varying the elastic parameters. The fundamental idea is to make the curve or snake to fit tightly to the boundary of a particular object in the image.

The design of evolution equation is such that the snake can easily embrace the object of importance, to be able to develop a similarity. The first snake model was proposed by Kass [31]. The minimization energy function in order to achieve equilibrium is

$$E_{Snake} = \int_0^1 E_{Snake}(v(s))ds = \int_0^1 E_{internal}(v(s))ds + E_{image}(v(s))ds + E_{CoN}(v(s))ds$$  \hspace{1cm} (13)

where the location of the snake on the image is represented parametrically by a planer curve

$$\chi(s) = (x(s), y(s))$$  \hspace{1cm} (14)

and $E_{internal}$ represents the internal energy of the curve due to bending and $E_{image}$ represents the image forces that push the snake towards the desired object. $E_{CoN}$ is the constraint that helps keep the snake movements smooth in all directions. The internal energy model was defined as

$$E_{internal} = \frac{\alpha(s)\chi'(s)^2 + \beta(s)\chi''(s)^2}{2}$$  \hspace{1cm} (15)

Where $\chi'(s)$ First derivative of $\chi(s)$ which tracks changing curve length and $\alpha(s)$ maintains the degree of contraction in all directions. Similarly, $\chi''(s)$ is Second order derivative of $\chi(s)$ with respect to $s$ representing changes in snake curvature and $\beta(s)$ normalizes curvature movements in the direction of the normal along the snake boundary. The model of image energy is defined as

$$E_{image} = \|\nabla f(x, y)\|^2$$  \hspace{1cm} (16)

This model is further refined by Chan-Vese [29] which finds a contour $\hat{I} : s \rightarrow \mathbb{R}^2$, that approximates the object regions in image $f(x, y)$ into a single real gray value $\hat{I}_{internal}$ to boundary of the contour $\hat{I}$ and $\hat{I}_{external}$ to exterior of the boundary $\hat{I}$. Energy function in CV model is represented with linear Mumford-Shah [30] model which approximates a 2D function $f(x, y)$ by a piece wise smooth function giving rise to distance minimization problem defined as

$$E_{chan-ve}(\hat{I}, \hat{I}_{internal}(E), \hat{I}_{external}(E)) = \min_{\theta \in \{\hat{I}_{internal}(E), \hat{I}_{external}(E)\}} \int_{\theta} (I_{xy} - \hat{I}_{xy})^2 h(\hat{I}_{xy}) + \int_{\theta} (I_{xy} - \hat{I}_{xy})^2 (1 - h(\hat{I}_{xy}))dxdy + \int_{\theta} \|\nabla h(\hat{I}_{xy})\|dxdy$$  \hspace{1cm} (17)
The last term in the eq. 17 indicates arc length which guarantee evenness of $\tilde{t}$. The first term has two integrals. The first integral function pushes the contour $\tilde{t}$ towards the image $f(x, y)$ while the second integral function ensures the differentiability on the contour $\tilde{t}$. $\chi_2$ and $\chi_1$ are the regularization parameters which define the percentage of smoothness required for a particular set of pixels.

Sethian and Osher [32] represented boundaries of $\tilde{t}(x, y)$ implicitly and a set of partial differential equations model their propagation around the edges in the image. Initial level set function $\phi(x)$ is the boundary trace. The interface boundary in the level set model is parametrized by a zero level set function $\phi(x) = 0$ where, $\phi: \mathbb{R}^2 \rightarrow \mathbb{R}$. $\tilde{t}$ is defined for all values of $x$.

$\tilde{t} = \{ \phi(x) = 0, x \in \mathbb{R}^2 \}$

(18)

The sign of $\phi(x)$ controls the pixel $x$ as it is inside the contour $\tilde{t}$ or external to it. The sets $\{x, \phi(x) \leq 0\}$ and $\{x, \phi(x) > 0\}$. The curvature $\kappa$ wheels the level set towards the image objects and the curve smoothness is from the outward normal $\vec{n}$ in terms of parameter $\phi$ as

$$\kappa = \nabla \left[ \frac{\nabla \phi}{|\nabla \phi|} \right] \quad \text{and} \quad \vec{n} = \nabla \phi \prod \frac{1}{|\nabla \phi|}$$

(19)

Here the curve $\tilde{t}$ evolution is a time dependent process and the time dependent level set function is represented as $\phi: \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$, $\tilde{t}(t) = \{ \phi(x, t) = 0, x \in \mathbb{R}^2 \}$. One way to solve is to approximate spatial derivatives of motion and update the position of the curve over time.

4. Continuous Sign Language Recognizer – Proposed Model

The focus is on hands tracking and hand shapes as features which constitute around 70% of characterization for any sign language. For hand tracking horn schunck optical flow algorithm is used and shape features are extracted in each frame with active contour level set model. The tracking features are set of velocity vectors extracted from each moving hand in the frames. Hand shapes from each frame are represented with hand outliners extracted with shape numbers. Two features from each frame are concatenated to represent signs in each frame. The entire process of the proposed continuous sign language recognizer is presented as a flow chart in Figure 2. From Figure 2, the video
frames for this work represent a sign with a set of frames. The set of frames can be identified using velocity vectors computed using HSOF algorithm. A start of sign (SOS) set frames is recognized when velocity vectors between frames is maximum. A no sign frame $f_{\text{NoSign}}$ will have velocity vectors $u_{\text{NoSign}}^{(n)}$ and $v_{\text{NoSign}}^{(n)}$ from HSOF algorithm. Similarly the consecutive sign frame $f_{\text{Sign}}^i$ is having velocity vectors $u_{\text{Sign}}^{(n)}$ and $v_{\text{Sign}}^{(n)}$.

The following formulation decides the SOS

$$S_{\text{SOS}} = \begin{cases} u_{\text{Sign}}^{(n)} - u_{\text{NoSign}}^{(n)} < T \\ u_{\text{Sign}}^{(n)} - u_{\text{NoSign}}^{(n)} \geq T \end{cases}$$

(20)

Where $T$ is the velocity threshold. For less hand movements between frames $T$ is very small, whereas larger values of $T$ are produced between sign frames and no sign frames. Similar model is used to decide on end of sign (EOS). In the sequence of frames the first SOS is extracted and the next low threshold difference is marked as EOS. Once EOS is marked, the remaining frames will have almost zero threshold as there will not be any
hand or head movements detected by HSOF. The next SOF will be the max threshold value from producing maximum velocity differences. Figure 3 shows a video sequence indicating the SOS by green border and EOS by red border frames. Close observations of the frames reveal the idea of selecting SOS and EOS. For display in figure only important frames are used.

A feature matrix is extracted between one SOS and EOS. This feature matrix becomes input to the classifier. Training to the Artificial Neural Network classifier is provided with this feature matrix. Error backpropagation algorithm is used as a training algorithm. The error is calculated with gradient descent algorithm. The following ANN model is used for combined feature vector classification as shown in Figure 4. The details of weights, biases and learning rate values are randomly initialized and re iteratively updated with a convex cost function such as gradient descent algorithm.

![Neural Network Model](image)

**Figure 4. Neural Network Model used for Sign Classification**

5. **Results and Discussion**

For testing the above process on the videos of continuous sign language, a camera setup is constructed. To ensure uniform lighting, two 23 luminance bulbs are erected at an
angle of 45 degrees from the signer. A HD Sony camcorder at a distance of 22 meters ensures excellent HD videos of continuous sign language. For experimentation for this model a subset of Indian Sign Language is used. The setup is shown in Figure 5.

A total of 10 test subjects were used. Each performing the same set of sentences for Indian Sign Language. A continuous sentence comprising 58 words is chosen for our experimentation. The sentence used is “Hai, Good Morning, My Name is Kishore, I am Student of K.L.University, Studying final year Undergraduate Engineering, From Department of Electronics and Communications engineering, the college is located at a lush green surroundings, with estimated area of around 50 acres, we are doing this sign language as a part of our final year project, thank you”. Each video recording of the above sequence of sentences is averaged around 100 sec. This is because, the signers are not regular users of sign language.

Sony camcorder outputs videos with a 30fps HD sequences with a frame size of 640×480. For a 100 sec sequence we are looking at 3000 frames for a sentence of 58 words. For 10 different signers the figure is 30000 frames. Different signers are selected to make the system signer independent. Of these 5 samples are used for training and 5 will be testing samples. From any 5 samples feature vector is built combining tracking position vectors from HSOF and shape outliner’s vectors from AC models, which will train artificial neural network.

Horn Schunck optical flow is the tracker for hands in the video frames. The algorithm implements on two frames at a time and computes the velocity vectors in x and y directions. From these velocity vectors, we compute the position of the hand in the frame. Multiple positions are extracted due to the some ambiguity in the hand positions. This is due to light intensity variations in the frames during capture. Hence the average position vector is obtained for each hand in the frames. The tracking using Horn Schunck Optical Flow (HSOF) on frames of Figure 3 is shown in Figure 6.
Figure 6. Video Frames Showing Results of HSOF Tracking Signer’s Hand and Head. The Green Arrows Resembling Velocity Vectors Along x and y Directions. Here the Signer is Saying “HAI”.

The same HSOF algorithm can do the job of segmentation of moving objects in a video. The results of HSOF segmentation for a few frames are shown in Figure 7.

Figure 7. HSOF based Sign Segmentation. (a) Frame 52 (b) Frame 52 Segmentation Output from HSOF (c) Frame 75 (d) Segmentation Result of (c) with HSOF.

From Figure 7(b) and (d) the HSOF segmented hands fails to produce exact contours to represent shape features. Hence in this HSOF algorithm will only track hands. The final tracks on the continuous video sequence is given in Figure 8. Intermediate frames are displayed in Figure 8 out of a total frame count of 2998 frames.
Figure 8. Tracks of Hands in a Continuous Sign Video Sequence used for Experimentation.

Figure 9. Hand Tracks of (a) Right Hand of Signer (b) Left Hand of Signer
Right hand bounding box is in red color and for left hand green is used. ‘X’ and ‘Y’ are position vectors with respect to head of the signer. Head position is marked manually in the first frame. Figure 9 displays tracks in three dimensional space. A feature vector is generated by careful labelling to both hands of the signer. The vector varied from signer to signer as their hand speeds changed during sign acquisition. Hence the tracking feature space is normalized by generating intermediate lost values for fast signers and by removing repeated position vectors for slow signers. Finally for a particular signer with 58 words we have a $1996 \times 58 \times 2$ matrix. Further normalization by temporal averaging, the tracking feature matrix is represented with $1996 \times 58$, giving 1996 tacks for 58 words sequences. By using only tracking information it is impossible to determine the classification problem.

Hence shape information of hands should accompany the tracks to the input of the classifier. Hand shape extraction is accomplished with active contour model. The contour is placed in an area close to the head of the signer. This enables the AC segmentation algorithm to trigger only when there is significant movement near the torso of the signer. Figure 10 shows the AC segmentation process on a video frame of the signer. The contour is placed close to the torso of the signer to keep the number of iterations for segmentation to a minimum.

![Active Contour Segmentation on a Video Frame](image)

The segmentation is near perfect and no further processing is required. The head and hand shape boundary numbers are considered as feature vector per frame. A set of frames are presented in Figure 11 for the sentence described above. Total number for frames in this particular video are around 2998. Only 699 frames have useful information that can be considered as feature vector design. These 699 frames are selected based on frame differencing model used in equation (20). Here threshold is set based on the velocity difference value. Large velocity value changes in frames are retained and those with lesser velocity gradient are discarded.

The contours are extracted from the region boundaries of the segments. Hand and head segment contours are manually labelled as Head and hand contours. The extracted contours for a few frames are shown in Figure 12. The boundary of the contours are given unique combination of numbers to uniquely identify a particular shape in the video.
frame. These labelled shape numbers are fused with tracking features for the same frame. A final feature matrix is an amalgamation of two important characteristics for machine understanding of sign language from video sequences.

Figure 11. Segmentation Outputs of a Few Important Frames form the Video Sequence. The Frames are Layered in Horizontal Format.

Figure 12. Extracted Shape Contours for Few Frames
The feature vector for both training and testing is built based on velocity vector gradients. Lesser gradients marks Star of sign (SOS) and End of Sign (EOS) frames. All the middle frames will have a feature vector. In this work we have 58 words and hence we have 58 feature vectors. Each feature vector is represented by variable number of samples with both tracking and shape numbers. For a complete 58 word sentence we have 699 frame video.

A complete tracking feature matrix is created from 6 position vectors obtained from tracking i.e., right hand position \((x, y)\), left hand position and Fixed head position. Similarly hands and head are represented with 52 shape numbers per frame constitute shape feature matrix. Concatenating the two produces a \(58 \times 699\) feature matrix that trains the backpropagation neural network. The entire process of feature vector design is mapped diagrammatically as shown in Figure 13.

![Figure 13. Feature Vector Design Process](image)

The target matrix consists of 58 words in the sentence in the order of sequence described previously. To improve the efficiency of the training program, four more samples are added to the one derived feature matrix earlier. The training input vector is \(290 \times 699\) whereas the target is \(58 \times 699\). Both the matrices are supplied as input to the neural network with 290 inputs and 58 targets. Gradient descent error is transmitted to update the weights after every iteration as described in Section 4. Log sigmoid activation function is used in all the layers. The model of neural network object is created in MATLAB and is shown in Figure 14.

A good number of hidden neurons will speed up the training process and 350 hidden neurons are chosen for the input vector of 290 elements. There is no cap on the hidden neurons expect for the fact that there are algorithms and formulae to decide on the number of hidden neurons. In this simulation hidden neurons are chosen a little above input neurons.
In backpropagation training the networks weights and biases are continuously updated with the mean square error to minimize network performance. The mean square error tolerance is fixed at 0.01 for training the samples. The learning rate and momentum factor were chosen as 0.241 and 0.5. The training graph between mean square error and epochs for the neural network object is shown in Figure 15.

![Neural Network Object for Continuous Sign Language Recognizer](image)

**Figure 14. Neural Network Object for Continuous Sign Language Recognizer**

The performance of the proposed system is computed with word matching score given by

\[
\omega_{\text{wms}} = \frac{\text{Word Matching}}{\text{Total Words}} \times 100
\]

(20)

For individual words in the sentence, word matching score is computed with 5 samples for training and remaining 5 samples and the 5 already trained ones are used for testing the trained network. Table 1 gives values of WMS for the proposed method against the three other methods. The experimentation was done 5 times. The WMS values in the Table 1 are averaged values over 5 times. Each time training is accomplished with same training set for all models of sign language systems in table.

Table 2 gives word matching scores of differently trained artificial neural networks. The number of signer samples for training are increased to 6, 7 and 8 to check whether the accuracy of recognition improves with increased training data. The neural networks are powerful tools to solve this type of problems only is sufficiently large training samples are employed. We could not train 9 samples because of out of memory exception in Matlab.
Table 1. Word Matching Scores for Individual Words in the Sentence used in this Work

<table>
<thead>
<tr>
<th>Words</th>
<th>Word Matching Score-Proposed, HSOF+AC+ANN</th>
<th>Sobel+DCT+ANN [33]</th>
<th>Sobel+Hough Transform+ANN [34]</th>
<th>Active Contour Model for seg &amp; tracking+ANN [14]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hai</td>
<td>90</td>
<td>50</td>
<td>60</td>
<td>90</td>
</tr>
<tr>
<td>good</td>
<td>100</td>
<td>70</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>morning</td>
<td>100</td>
<td>70</td>
<td>70</td>
<td>100</td>
</tr>
<tr>
<td>my</td>
<td>80</td>
<td>20</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>name</td>
<td>80</td>
<td>60</td>
<td>60</td>
<td>70</td>
</tr>
<tr>
<td>is</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Kishore</td>
<td>70</td>
<td>30</td>
<td>40</td>
<td>80</td>
</tr>
<tr>
<td>I</td>
<td>70</td>
<td>10</td>
<td>20</td>
<td>60</td>
</tr>
<tr>
<td>am</td>
<td>60</td>
<td>10</td>
<td>20</td>
<td>70</td>
</tr>
<tr>
<td>a</td>
<td>100</td>
<td>60</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>student</td>
<td>90</td>
<td>60</td>
<td>60</td>
<td>80</td>
</tr>
<tr>
<td>of</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>K</td>
<td>100</td>
<td>50</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>L</td>
<td>100</td>
<td>80</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>University</td>
<td>80</td>
<td>40</td>
<td>50</td>
<td>70</td>
</tr>
<tr>
<td>Studying</td>
<td>90</td>
<td>60</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>final</td>
<td>80</td>
<td>20</td>
<td>50</td>
<td>80</td>
</tr>
<tr>
<td>year</td>
<td>90</td>
<td>80</td>
<td>70</td>
<td>90</td>
</tr>
<tr>
<td>Undergraduate</td>
<td>70</td>
<td>20</td>
<td>10</td>
<td>60</td>
</tr>
<tr>
<td>Engineering</td>
<td>80</td>
<td>20</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>From</td>
<td>100</td>
<td>40</td>
<td>50</td>
<td>90</td>
</tr>
<tr>
<td>Department</td>
<td>70</td>
<td>20</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>of</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Electronics</td>
<td>90</td>
<td>50</td>
<td>40</td>
<td>70</td>
</tr>
<tr>
<td>and</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Communications</td>
<td>100</td>
<td>80</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>engineering</td>
<td>80</td>
<td>20</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>the</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>college</td>
<td>80</td>
<td>20</td>
<td>30</td>
<td>70</td>
</tr>
<tr>
<td>is</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>located</td>
<td>80</td>
<td>30</td>
<td>50</td>
<td>80</td>
</tr>
<tr>
<td>at</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>a</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>lush</td>
<td>80</td>
<td>30</td>
<td>40</td>
<td>70</td>
</tr>
<tr>
<td>green</td>
<td>80</td>
<td>10</td>
<td>20</td>
<td>60</td>
</tr>
<tr>
<td>surroundings</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>with</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>estimated</td>
<td>70</td>
<td>20</td>
<td>30</td>
<td>70</td>
</tr>
</tbody>
</table>
Table 2. Continuous Sign Language Recognizer Word Classification Rate with Proposed Model

<table>
<thead>
<tr>
<th>Number of samples for training</th>
<th>Number of epochs for training</th>
<th>Number of unknown samples for testing</th>
<th>Number of words Correctly Recognized</th>
<th>Word Matching Score calculated from eq. 29</th>
</tr>
</thead>
<tbody>
<tr>
<td>290</td>
<td>695</td>
<td>290</td>
<td>478/580</td>
<td>82.4137%</td>
</tr>
<tr>
<td>348</td>
<td>772</td>
<td>232</td>
<td>510/580</td>
<td>87.931%</td>
</tr>
<tr>
<td>406</td>
<td>899</td>
<td>174</td>
<td>532/580</td>
<td>91.7241%</td>
</tr>
<tr>
<td>464</td>
<td>934</td>
<td>116</td>
<td>556/580</td>
<td>95.862%</td>
</tr>
<tr>
<td>1508</td>
<td>3300</td>
<td>812</td>
<td>2076</td>
<td>89.482%</td>
</tr>
</tbody>
</table>

The average word matching score is 89.482% for the entire classification process which is on par with other researchers for American Sign Language [35] and Chinese Sign Language in [18]. The results indicate that by employing multiple features in continuous sign language recognition process, the word classification rates can be improved drastically when compared to single feature SLR models. Finally the recognized text words input the voice application programmable interface available in windows OS to produce voice from text.

6. Conclusion

This work gives a multi feature model for recognizing continuous gestures of Indian sign language. Videos of continuous signs are captured for 58 words forming meaningful sentences. Horn Schunck optical flow algorithm extracts tracking features of both hands
providing position vectors of hands in each frame. Active Contour model on each frame extracts hand shapes features along with head portion. The combined feature matrix having tracking and shape features train the backpropagation neural network. The classified signs are mapped to text from the target matrix of ANN and converting those text inputs to voice commands with windows text-to-speech application programmable interface. Validating the proposed model by computing the word matching score for each word recognized by the neural network. The word matching score over multiple instances of training and testing of the neural network resulted in around 90%. This work can be extended to include other characteristics of continuous sign language.

References


Authors

P.V.V.Kishore (MIEEE’07), He received his Ph.D degree in electronics and communications Engineering from Andhra University College of engineering in 2013. He received M.Tech from Cochin University of science and technology in the year 2003. He received B.Tech degree in electronics and communications engineering from JNTU, Hyd. in 2000. He is currently full professor and Image, signal and speech processing Head at K.L.University, ECE Department. His research interests are digital signal and image processing, Artificial Intelligence and human object interactions. He is currently a member of IEEE. He has published 50 research papers in various National and International journals and conferences including IEEE, Springer and Elsevier.
M.V.D.Prasad. He received M.Tech from Andhra University in the year 2009, registered Ph.D. at K.L.University the year 2013. Working as an Assistant professor in the KL University since 2012. His Research areas are image processing, video processing and speech processing. He has published 10 research papers in Various National and International journals and conferences including IEEE, Springer and Elsevier.