Hand Gesture Recognition for Sign Language: A New Hybrid Approach

J.Rekha¹, J.Bhattacharya² and S.Majumder²

¹,²Surface Robotics Laboratory, Central Mechanical Engineering Research Institute, Durgapur-713209, West Bengal, India.

Abstract - This paper proposes two new approaches of hand gesture recognition which will recognize sign language gestures in a real time environment. A hybrid feature descriptor, which combines the advantages of SURF & Hu Moment Invariant methods, is used as a combined feature set to achieve a good recognition rate along with a low time complexity. To further increase the recognition rate and make the recognition system resilient to viewpoint variations, the concept of derived features from the available feature set is introduced. K-Nearest Neighbour (KNN) and Support Vector Machine (SVM) are used for hybrid classification of single signed letter. In addition, finger spelled word recognition using Hidden Markov Model (HMM) for a lexicon based approach is also proposed in this paper. The performance analysis of the proposed approaches is presented along with the experimental results. Comparative study of these methods with other popular techniques shows that the real time efficiency and robustness are better.

Keywords: Finger Spelled Word Recognition, Hu Moment Invariant, Hidden Markov Model (HMM), Sign Language, Speeded Up Robust Features (SURF), Support Vector Machine (SVM).

1 Introduction

Sign language is used as a communication medium among deaf & dumb people to convey the message with each other. A person who can talk and hear properly (normal person) cannot communicate with deaf & dumb person unless he/she is familiar with sign language. Same case is applicable when a deaf & dumb person wants to communicate with a normal person or blind person. In order to bridge the gap in communication among deaf & dumb community and normal community, Video Relay Service (VRS) is being used nowadays. In VRS a manual interpreter translates the hand signs to voice and vice versa to help communication at both ends. A lot of research work has been carried out to automate the process of sign language interpretation with the help of image processing and pattern recognition techniques.

The approaches can be broadly classified into “Data-Glove based” and “Vision-based” [1]. Tracking bare hand and recognizing the hand gestures using low level features such as color, shape, or depth information [2] generally require uniform background, invariable illumination, a single person in the camera view, and/or a single large centred hand in the camera view. A lot of researchers initially used morphological operations [3] to detect hand from image frames. The main drawback of this method lies in its huge computational complexity which is further handled with the concept of integral image. The use of integral image for hand detection in viola-Jones [4] method reduces computational complexity and shows satisfactory performance only in a controlled environment. To detect hand in a clutted background, N.Petersen & D.Stricker used color information and histogram distribution model [5]. Some Local orientation histogram technique is [6] also used for static gesture recognition. These algorithms perform well in a controlled lighting condition, but fails in case of illumination changes, scaling and rotation. To resist illumination changes, Elastic graphs [2] are applied to represent different hand gestures in Triesch’s work with local jets of Gabor Filters. Mathias and Turk used Adaboost for wearable computing. It is insensitive to camera movement and user variance. Their hand tracking is promising, but segmentation is not reliable. Chan & Ranganath used Fourier descriptors of binary hand blobs as feature vector to Radial Basis Function (RBF) classifier for pose classification and combined HMM classifiers for gesture classification [7]. Even though their system achieves good performance, it is not robust against multi variations during hand movement. To overcome the problem of multi variations like rotation, scaling, translation some popular techniques like SIFT [8], Haar-like features [9] with Adaboost classifiers [10], Active learning [11] and appearance based approaches [2] are used. However, all these algorithms suffer from the problem of time complexity. To increase the accuracy of the hand gesture recognition system, combined feature selection approach [16] is adopted. Automatic translation of cued speech gesture recognition with supervised classification [12] [13] using a multi-layer perceptron is used with the help of 2D & 3D information. Several features like motion cue, colour [14], global and local motion components, position, orientation [15], and velocity [16] are used along with Hidden Markov Model (HMM) classification. But this approach requires more number of dataset samples.

In this paper, two new approaches for real time hand gesture recognition which can identify different hand postures in a robust and faster way are introduced. American Sign Language (ASL) alphabet signs are used for recognition process.

Video dataset of ASL alphabets is taken with three different background and environmental conditions. Considering the trade-off between recognition rate and processing time, proposed approach uses optimized feature set with combined output from hybrid classifier architecture i.e., KNN & SVM classifiers for single letter recognition. A
lexicon based HMM for finger spelled word recognition is also proposed in this paper.

This paper is organized in the following manner: Section 2 explains the proposed approaches. Section 3 explains finger-spelled word recognition approach. Experimental results and discussions are shown in section 4. Section 5 summarizes the paper with conclusion.

2 Hand Gesture Recognition System

Hand gesture recognition system consists of the following steps (a) Pre-processing and hand segmentation, (b) Hand detection and tracking, (c) Hand posture recognition and (d) Hand gesture classification as shown in figure 1.

![Figure 1. Block diagram of Hand Gesture Recognition System](image)

This section describes about individual letter recognition approach, consisting of hand segmentation (section 2.1), extraction of invariant feature descriptor (section 2.2), recognition of letters (section 2.3) and derived feature approach for improving accuracy (section 2.4).

2.1 Hand Segmentation

Skin color segmentation is performed using k-means clustering [19] method. RGB frames I(m,n,p), (where m, n and p are number of rows, number of columns and number of color planes) are converted into 3 1-dimensional feature vector X of single column and m*n rows. Then I = { X(1), X(2), .... X(N)} (N=3) is obtained. Through experimental observation classification of colors using 3 clusters (K) under Euclidean distance measure provides better performance. For every pixel in the input, k-means returns an index corresponding to a cluster. Skin pixel region is identified from the different color regions using a thresholding method in RGB color space where the threshold value is selected experimentally. Repeat the cluster for 3 times to avoid local minima. Figure 2 shows example results of the segmentation algorithm.

![Figure 2 (a) Original frame, (b) cluster-labelled image and (c) Skin region segmented image](image)

2.2 Hand Detection using Invariant Feature Descriptors

After obtaining skin segmented RGB image, it is converted into gray scale. The converted gray scale image is normalized. Invariant features are extracted using Scale Invariant Feature Transform SIFT [8] method. The basic idea is to extract the invariant key point which represents/identifies hand from the segmented image. For this purpose of hand detection, SIFT features are first extracted from a set of reference images and stored in a database. An image frame is matched by individually comparing each feature from the image frame to this previous database and finding candidate matching features based on Euclidean distance of their feature vectors.

Four major steps are followed to find out invariant key points: scale-space extrema detection, key point localization, orientation assignment and defining key point descriptor. Detecting locations that are invariant to scale change of the image can be accomplished by searching for stable features across all possible scales, using a continuous function of scale known as scale space. The scale space of an image is defined as a function $L(x,y,\sigma)$ that is produced from the convolution of a variable-scale Gaussian, $G(x,y,\sigma)$. With an segmented input image $Segl(x,y)$,

$$L(x,y,\sigma) = G(x,y,\sigma) * Segl(x,y)$$  \hspace{1cm} (1)

Where $*$ is the convolution operation in x and y, and

$$G = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$  \hspace{1cm} (2)

To efficiently detect stable key point locations in scale space, the difference of Gaussian function can be computed from the difference of two nearby scales separated by a constant multiplicative factor $k$.

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * Segl(x,y)$$  \hspace{1cm} (3)

$$= L(x,y,k\sigma) - L(x,y,\sigma)$$

The initial image is incrementally convolved with Gaussians (with = 1.2 ) to produce images separated by a constant factor $k$ in scale space. Each octave of scale space is divided (i.e., doubling of $\sigma$) into an integer number, $s$. ($s=1$ at start), of intervals, so $k = 2^s$. In the stack of blurred images, $s$+3 images are produced for each octave, so that final extrema detection covers a complete octave. Adjacent image scales are subtracted to produce the difference-of-Gaussian images. Once a complete octave has been processed, resample the Gaussian image that has twice the initial value of $\sigma$ (it will be 2 images from the top of the stack) by taking every second pixel in each row and column. The accuracy of sampling relative to $\sigma$ is no different than for the start of the previous octave, while computation is greatly reduced.

In order to detect the local maxima and minima of $D(x,y,\sigma)$, each sample point is compared to its eight neighbours in the current image and nine neighbours in the scale above and below(see [8] for detailed description about selection of sample points). It is selected only if it is larger than all of these neighbours or smaller than all of them. Once a key point candidate has been found by comparing a pixel to
its neighbours, the next step is to perform a detailed fit to the nearby data for location, scale, and ratio of principal curvatures (refer section 4 in [8] for details). The principal curvatures can be computed from a 2x2 Hessian matrix, H, computed at the location and scale of the key point,

\[
H = \begin{bmatrix}
D_{xx} & D_{xy} \\
D_{xy} & D_{yy}
\end{bmatrix}
\] (4)

The derivatives are estimated by taking differences of neighbouring sample points (refer [8] for details). The Eigen values of H are proportional to the principal curvatures of D. Key point location is obtained. For each image sample, \(L(x,y)\), the gradient magnitude, \(m(x,y)\), and orientation \(\theta(x,y)\) (refer section 5 in [8]), is pre-computed using pixel differences, to assign orientation to these key points. An orientation histogram is formed from the gradient orientations of sample points within a region around the key point. The orientation histogram has 36 bins covering the 360 degree range of orientations. Peaks in the orientation histogram correspond to dominant directions of local gradients. A key point descriptor is created by summarizing the contents over 8x8 sub-regions, from a 16x16 sample array. At last we obtain 4 x 4 array of histogram in 8 orientation bins, 4x4x8=128 element feature vector for each key point.

![Figure 3: Extracted key points from reference images](image)

This 128x1 feature vector is called invariant feature descriptor. The database contains both positive (contains hand) and negative images (non hand image). SIFT key point descriptors are calculated for all the sample images and classified as hand (+1) and non-hand (-1). For the classification purpose Adaboost classifier [4][20] is used. It is a weak learning algorithm designed to select the prominent feature which best separates the positive and negative examples. For each feature, the weak learner (classifiers) determines the optimal threshold classification function, such that the minimum number of samples is misclassified. 55 weak classifiers are chosen for the purpose of error minimization as shown in figure 4. A weak classifier \(h_x(x)\) thus consists of a feature \(f_i\), a threshold \(p_j\), and a parity \(p_j\) indicating the direction of the inequality sign.

\[
h_j(x) = \begin{cases}
1 & \text{if } p_j f_i(x) < p_j \\
0 & \text{otherwise}
\end{cases}
\] (5)

Here x represents the sample images in the training database. A sliding window of size 120x120, with unit values is moved over the segmented image. Sum of product of window and image area is calculated. If the sum is greater than a threshold value then SIFT feature vectors are calculated for that sub-window pixels. If not, then the sub-window is not taken for processing.

![Calculated SIFT features are compared with the database feature vectors through Adaboost classifier.](image)

Calculated SIFT features are compared with the database feature vectors through Adaboost classifier. If the features are matched, the classifier outputs +1 and sub-window co-ordinates are stored. If the classifier output is -1, then the sub-window is left. By taking average of the matched sub-window co-ordinates, hand is detected in the segmented image as shown in figure 4.

### 2.3 Recognition of Letters

The bounding box of the detected hand in each frame is obtained from the previous section. To recognize the posture of detected hand, a combined feature extraction methodology using Speeded Up Robust Features (SURF) [22] and Hu Moment Invariant features [24] is incorporated. Bounding box, BBIm(x,y) is taken as test image. Features are calculated and compared with the database features. Minimum Euclidean distance between the feature vectors recognizes particular hand posture/letter.

#### 2.3.1 SURF Features

Given an image BBIm(x,y), integral image ii(x,y) is calculated using,

\[
ii(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} BBIm(x', y')
\] (6)

To find out the interest points from the integral image, Fast Hessian Detector [22] is used. Given a point X=(x,y) in image ii(x,y), the Hessian matrix H(X,\sigma) in X at scale \(\sigma\) is defined as

\[
H(X, \sigma) = \begin{bmatrix}
L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\
L_{yx}(X, \sigma) & L_{yy}(X, \sigma)
\end{bmatrix}
\] (7)

Where \(L_{xx}(X, \sigma)\) is the convolution of the Gaussian second order derivative \(\frac{\partial^2}{\partial x^2} g(\sigma)\) with the image ii in point X, and similarly for \(L_{xy}(X, \sigma)\) and \(L_{yx}(X, \sigma)\). To localize interest points in the image and over scales, a non-maximum suppression in a 3x3x3 neighbourhood is applied (see [22] for details). The maxima of the determinant of the Hessian matrix are then interpolated in scale and image space with the Browns method [22]. In order to be invariant to rotation, Haar-wavelet responses in x and y direction, within radius 6s around interest point is calculated. For the extraction of the descriptor, the first step consists of constructing a square region centred on the interest point. The region is split up regularly into smaller 4 x 4 square sub-regions. This keeps important spatial information in. For each sub-region, a few
simple features at 5x5 regularly spaced sample points are computed. $dx$ the Haar wavelet response in horizontal direction and $dy$ the Haar wavelet response in vertical direction (filter size 2s). The wavelet responses $dx$ and $dy$ are summed up over each sub-region and form a first set of entries to the feature vector. Absolute values of the responses $|dx|$ and $|dy|$ provide polarity information. Each sub-region has a four-dimensional descriptor vector $v$, $v = (\sum dx, \sum dy, \Sigma|dx|, \Sigma|dy|)$. This results in a descriptor vector for all 4x4 sub-regions of length 64.

Two-dimensional moments of detected hand image $BBIm(x, y)$ of size m×m is given as,

$$m_{pq} = \sum_{x=0}^{x=m-1} \sum_{y=0}^{y=m-1} \frac{(x)^p(y)^qBBIm(x,y)}{2}$$

The moments $BBIm(x,y)$ translated by an amount (a,b), are defined as,

$$\mu_{pq} = \sum_{x} \sum_{y} (x-a)^p(y-b)^qBBIm(x,y)$$

Thus the central moments $m_{pq}$ or $\mu_{pq}$ can be computed from

on substituting $a=x$ and $b=y$ as $\bar{x} = \frac{m_{10}}{m_{00}}$ and $\bar{y} = \frac{m_{01}}{m_{00}}$.

$$\mu_{pq} = \sum_{x} \sum_{y} (x-\bar{x})^p(y-\bar{y})^qBBIm(x,y)$$

When a scaling normalization is applied the central moments change as, $\eta_{pq} = \frac{\mu_{pq}}{\eta_{00}}$, $\gamma = \frac{p+q}{2} + 1$. Hu defines seven values, computed by normalizing central moments through order three, that are invariant to object scale, position and orientation. Now a 1x64 feature vector from surf and 1x7 feature vector from moment invariant are obtained for all the reference (posture/ letter) images and stored in a database.

### 2.3.3 Classification using K-Nearest Neighbour

Combined feature vectors of database images are stored in a database. These feature vectors are classified using KNN classifier. Feature vectors of detected hand image from subsequent frame are compared with the stored feature vectors by means of a Euclidean distance measure in KNN. “D”, a dataset matrix of dimension N×P containing P samples $s'$, where each sample $s'$ contains N (64+7 in this case) features $s' = [s'_1,...,s'_N]$ is computed. A vector $o$ with length P of output values $o = [o'_1,...,o'_N]$ accompanies this matrix, listing the output value $o'_i$ for each sample $s'_i$. To classify the detected hand image ($BBIm(x,y)$) feature vector $q$, the output values of the $M$ nearest neighbours in vector $r = [r_1,...,r_M]$ is stored by repeating the following loop $M$ times:

The next sample $s'_i$ in the data set is considered, where $i$ is the current iteration within the domain $[1,...,P]$.

1. If $q$ is not set or $q < d(q,s'_i)$, then $q \leftarrow d(q,s'_i)$, $t \leftarrow o'_i$.
2. Loop is continued till the end of the dataset (i.e $i = P$).
3. Then $q$ and $t$ are stored in vector $c$ and vector $r$ respectively.

The arithmetic mean output across $r$ is calculated using $\bar{r} = \frac{1}{M} \sum_{i=1}^{M} r_i$. Then $\bar{r}$ as the output value for the test sample BBIm(x,y). For example, consider M=8, $r = [r_1,...,r_8]$, then $\bar{r}_q$ is calculated and mapped to single value $r_0$ using the condition, $r_0 = \{ f \bar{r}_q \geq b_0 then letter r_0 where b_0 is a vector of length 13 having weights for each ASL letters. This procedure is carried out for the sequence of detected hand from video frames. For each frame classifier recognizes a single letter as output.

### 2.3.4 Classification using Support Vector Machines

Simultaneously the feature vectors of the dataset are given to SVM classifier [23] for training. The basic principle of SVM is to find an optimal separating hyperplane (OSH) which can separate different classes in a feature space, that is, the distances between these classes should be the furthest. To perform the classification between two classes, a nonlinear SVM classifier is applied by mapping the input data $(x,y)$ into a higher dimensional feature space using a non-linear operator $\phi(x)$, where $x_i \in R^d$ and $y_i \in \{1,-1\}$. The OSH can be computed as a decision surface:

$$f(x) = sign(\sum \alpha_i y_i K(x_i,x) + b)$$

Where sign () is the sign function and $K(x_i,x) = \phi(x_i)^T\phi(x)$ is the predefined kernel function. In this approach the radial basis function (RBF) is used and it is defined as:

$$K(x_i,x) = e^{-\frac{|x_i-x|^2}{\sigma^2}}, \sigma > 0$$

where $\sigma$ is the Gaussian width. The coefficients $\alpha_i$ and $b$ in (11) can be determined by the quadratic problem. This procedure is carried out for the sequence of detected hand from video frames. For each frame classifier recognizes a single letter as output. The results given by both the classifiers
are taken as a combined feature vector for gesture classification.

2.4 Improving accuracy through derived features

To further increase the recognition rate and speed of processing, prominent features are derived from the available data set of features using forward selection algorithm. To improve the performance of classifier on a dataset, it is possible to evaluate each feature’s deviation. The deviation is computed per feature \( x_j \) in the set of \( N \) features \( x = [x_1, \ldots x_N] \) by calculating the sum of all differences between the calculated resultant feature vector \( y \) when feature \( x_j \) is left out and the actual resultant feature vector \( o' \) of sample \( s \) in the dataset \( D = N \times P \), containing \( P \) samples, where each sample \( s \) contains \( N \) features \( s = [s_1, \ldots s_N] \). For clarity, we’ll define a new feature set \( y \) that excludes \( x_j \), such that \( y = [x_1, \ldots x_{j-1}, x_{j+1}, \ldots x_N] \)

The algorithm run as follows on feature \( x_j \), with feature set \( y \) that excludes \( x_j \):

1. **Feature deviation**
   
   \[ d_j = |\text{resultant feature vector} - o'| \] is stored.

   Note that step 2 ensures that the feature deviation is always increments positively. The forward selection computes the “best features” of the data set, i.e. features whose feature deviation is minimal and prominent for best recognition rate. The resultant feature vector along with its deviation parameter is appended to the already available surf and moment invariant feature database for single letter recognition using hybrid classifier.

3 Gesture Classification-Finger Spelled Word Recognition

A simple Hidden Markov Model HMM [19] [20] model as shown in figure 9, is designed for the continuous recognition of finger spelled words.

To recognize continuously signed words the classifier is combined with a lexicon of known words, modelled as a Hidden Markov Model (HMM). The single letter classifier outputs a feature vector over letters and non-letters. The goal is: (a) to pick up the dominant image frame classifier output to avoid repetition of same letter over multiple frames and (b) to clearly distinguish letters which are very similar in representation. A HMM is build for each word in the lexicon as a chain of alternating letter and non-letter (\( \alpha \)) states. The probability of a word \( w \) given the sequence of observed descriptors \( z_1 \ldots k \) is defined as,

\[
P(w|z_1 \ldots k) \propto \max_{s_1 \ldots k} P(s_1) \prod_{t=1}^{k} p(z_t|s_t) \prod_{t=2}^{k} P(s_t|s_{t-1})
\]

Where \( s_t \) denotes the (unobserved) state at time \( t \) (one of the nodes in figure 7). Viterbi algorithm [15] is used for maximization of state transition sequence. A given sequence is assigned a path which maximizes \( P(w|z_1 \ldots k) \). Start and end states are properly defined for the entire test data words. For example, word “LATHA” can be recognized by parsing the lexicon structure, which identifies the maximum path value for each letter transition in HMM Model.

4 Experimental Results

The accuracy and performance of the proposed approaches are further verified using an experimental dataset consisting of single ASL letters and continuous word (fingerspelled). All the experiments are carried out using Matlab R2009a, with Intel core i3 processor (CPU 2.27GHz) on a 64 bit windows platform.

4.1 Dataset

A dataset of more than 50 videos are taken as test samples. Three types of test sample videos (see figure 8) are considered for experimental purpose for all the methods and procedures explained in this paper namely clear background (C1), slightly cluttered background (C2), and slightly cluttered background with changing lighting conditions (C3). Some of the videos are captured in home environment without any special lighting using a consumer quality web camera. The resolution of the video is 320x240. Frame rate considered for processing is 5.

4.1.1 For Hand Detection

The database contains 250 positive images containing hand and 400 negative images not containing hand (see figure 9). Resolution of all the images is 320x240. They are normalised, smoothed before processing and converted into gray scale. SIFT feature descriptors are calculated for all the key points obtained for all the image samples in the database.

Figure 7: HMM representation of the lexicon. Circle denotes states with corresponding letters (“\( \alpha \)” = non-letter). Hollow and filled square indicates start/end states. Person names such as “LATHA”("left") and “RAMY” is shown as HMM model.

Figure 8: Dataset frames from C1 (above) and C2 (below)
4.1.2 For Letter Recognition

The training dataset for letter recognition contains 20 images per hand sign (see figure 10), so totally 15x20=300 images. Totally we considered 4 sets of 300 image dataset. Hence 1200 images are taken for training database. Those images are subjected to illumination changes, scaling, mirroring, blurring, rotation, view-point variant and translated. Resolution is 120x190 for all the images.

4.2 Single Letter Recognition- Analysis and Comparison

This section presents the performance and efficiency of the proposed method by comparing it with other existing methods. Three different sets of videos with clear background (C1), slightly cluttered background (C2) and slightly cluttered background with varying lighting conditions (C3) are considered as shown in figure 8. The video length is 30 seconds.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Avg Processing Time (seconds)</th>
<th>No.of Correct Letters (out of 15)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>60</td>
<td>6</td>
</tr>
<tr>
<td>SURF</td>
<td>27</td>
<td>11</td>
</tr>
<tr>
<td>Hu moment invariant</td>
<td>28</td>
<td>7</td>
</tr>
<tr>
<td>SURF-Moment</td>
<td>38</td>
<td>11</td>
</tr>
<tr>
<td>SURF-Moment-derived</td>
<td>44</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 1: Comparison of processing time and no. of recognized letters

SIFT for hand posture recognition alone takes little more processing time for an image because of the stack of image pyramids formed by the algorithm. But it is robust against rotation, translation, scaling and illumination changes. For each key point 1x128 feature vector is obtained. SIFT method with KNN classifier shows 68% and 60% recognition rate with C1 and C2 set respectively with processing time of more than 60s. SURF features with KNN classifier has shown good real time performance with recognition rate of 84.6% for C1 set. It is 3 times faster than SIFT, because it uses integral image for image convolutions and fast Hessian detector. But it is not robust against rotation and illumination changes. Recognition rate and speed of processing of SURF method are found to be promising. But shows poor performance with the C3 set where the illumination change occurs. Hu moment invariant feature based recognition has shown poor performance in case of C2, C3 sets whereas for C1 it showed satisfactory results. Out of 15 signs it recognized 7 letters correctly with less processing time of 28s.

<table>
<thead>
<tr>
<th>Methods</th>
<th>C1 Set</th>
<th>C2 Set</th>
<th>C3 Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>68</td>
<td>60</td>
<td>59.2</td>
</tr>
<tr>
<td>SURF</td>
<td>84.6</td>
<td>84</td>
<td>72.3</td>
</tr>
<tr>
<td>Hu moment invariant</td>
<td>58</td>
<td>54</td>
<td>49.3</td>
</tr>
<tr>
<td>SURF-Moment</td>
<td>86.2</td>
<td>85.8</td>
<td>85.2</td>
</tr>
<tr>
<td>SURF-Moment-derived features</td>
<td>89.9</td>
<td>88</td>
<td>88</td>
</tr>
</tbody>
</table>

Table 2: Comparison of Recognition Rates (%).

The proposed method with SURF & Hu moment invariant features has shown consistent performance in case of all the sets. To eliminate the illumination and rotation variation problem faced by SURF, proposed approach combines moment invariant features with it. This approach shows better real time performance when compared to others with processing time 38s. To further increase the recognition rate, the concept of derived features is used. These features when combined with previous approach provided better efficiency. The results obtained are given in table 2.

4.3 Word Recognition

To test the proposed model for recognition of continuously spelled words (see section 2.4), the full test set of 10 words are used. The non-letter class is treated identically to the letter classes, with non-letter samples sampled randomly from the training videos. Word recognition accuracy of 96% is achieved. Words like “LATHA”, “RAMY”, “KUMAR”, “VANITHA”, “COFFEE” have been taken for testing purpose.

5 Conclusion and Future Work

It is observed from the experimental results that SURF & moment invariant features, is robust against multiple
variations like rotation, scale, lighting and view-point and provides good real time performance. Use of derived features from available feature set along with SURF & moment invariant features further makes the approach highly robust against multiple variations and shows consistent real time performance with improved processing speed. Both approaches make use of hybrid classifier architecture i.e KNN & SVM. The trade off between accuracy and speed of processing is maintained by the methods. In future, research work will be focused on automatic Indian sign language (ISL) interpretation as text or voice. As ISL uses both hands for signing it involves both local and global hand movements thus the concept of gesture spotting, inter-hand occlusion will be investigated deeply in near future.

6 References


