Asian Journal of Convergence in Technology *Volume IV, Issue I*
Issn No.:2350-1146, I.F-5.11

Real-time Static Numeric Devnagari Sign Language Translator

Jayshree Pansare[1*, 2]
[1] *M. E. S. College of Engineering, Pune, India*
[2] *DAVV, India*
jayshree.pansare23@gmail.com[1*]

Abstract— **Recognition of Numeric Devnagari Sign Language (DSL) using Hand Gesture Recognition System (HGRS) has become an essential tool for dump and deaf people to interact with commoners. The development of the proposed system Real-time Numeric Devnagari Sign Language Translator (*RTNDSLT*) is the significant objective of our research work. The system architecture of *RTNDSLT* system is mainly comprised of five phases arranged in layered fashion from top to bottom. Vision-based, real-time and static *RTNDSLT* system works in cluttered background with mixed lighting conditions. This system focuses on histogram recognition technique, centroid recognition technique, and fingertip recognition technique. RTDSLT is based on algorithms such as *Histogram Recognition algorithm*, *Centroid Recognition algorithm*, *Peak-and-Valley Detection* algorithm, and *Peak-Point Detection* algorithm along with *Sample Image Database* algorithm. *RTNDSLT* system achieves a detection rate of 94.13% by applying the approach of Two Hands and Single Camera along with fingertip recognition technique.**

Key Words: Devnagari Sign Language (DSL), Histogram, Centroid, Fingertip Recognition Technique, Two Hands Single Camera approach, Cluttered Background, Devnagari Numbers

I. INTRODUCTION

Real-time Numeric Devnagari Sign Language Translator (RTNDSLT) is used to detect 11 DSL numbers from 0 to 10 by applying Two Hands and Single Camera approach in mixed lighting conditions and background clutter. RTNDSLT is constructed specifically for dump and deaf users. Convex hull technique is used in HGRS incorporates to recognize 24 American Sign Language (ASL) alphabets and attains a precision of 97.1% [1]. The centroid of Binary linked object (Blob) approach applied in real-time HGRS for recognizing 26 ASL alphabets in complex background achieves identification rate of 90.19% [2]. Fingertip recognition technique is used in HGRS to recognize ASL numerals. This technique works in real-time with an accuracy of 95% [3]. In HGRS, Dynamic Time Warping (DTW) and DCT (Discrete Cosine Transform) algorithms are used to detect 10 Arabic Sign Language (ArSL) numerals with success rate of 92.3% [4]. HGRS uses histogram matching tactics to recognize DSL numbers from 0 to 9 in static background. Vision-based approach integrated in system recognizes hand gestures achieves recognition rate of 93.1% [5]. Histogram comparison in HGRS is applied to detect DSL numbers from 0 to 9. The system achieves 81.1% recognition rate with 0.5 seconds recognition time [6]. Static HGRS detects 10 DSL numerals using Fourier Transform, DCT, Centroid, and Edge Orientation Histogram. It achieves recognition rate of 85% at a distance of 15 inches [7]. However, there is wide spread scope of development of system based on Two Hands and Single Camera approach towards dump and deaf people to interact with general people.

Consequently, our research work is based on the development of Real-time Numeric Devnagari Sign Language Translator (RTNDSLT) system. The abstract model and detailed system architecture of RTNDSLT are discussed in Section 2. The working of RTNDSLT system is presented in Section 3. Experimental results of various recognition techniques such as centroid, histogram and fingertip are compared in Section 4. Eventually, concluding remarks are specified in Section 5.

II. REAL-TIME NUMERIC DEVNAGARI SIGN LANGUAGE TRANSLATOR (RTNDSLT)

This system is useful for detection of Devnagari Sign Language Numerals into Devnagari Numerals and its translation in speech. The abstract model for RTNDSLT system is depicted in Fig. 1. It has been taken into account that the processing time for recognition must be low for real-time execution. Simultaneously, the system detects hand in cluttered background for robustness. The system architecture of RTNDSLT system is mainly comprised of five phases arranged in layered fashion from top to bottom and is depicted in Fig. 2. These phases include image capturing, pre-image processing, hand tracking, image post-processing and speech conversion. The layered phases along with their functioning are described in subsequent sections.

A. Image Capturing

In this phase, we create camera object namely; Vid (in Fig. 3) for capturing images of frame size [320x240] pixels with brightness, frame rate as depicted in Fig. 3. Every first or third frame is extracted from input image frame buffer and RGB image is produced as an Output-1.

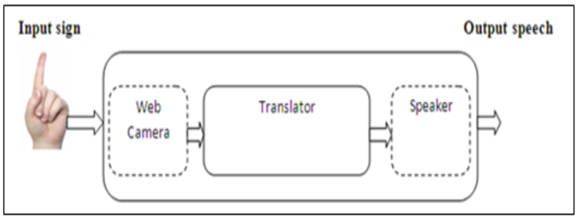

Figure 1 Abstract Model of Real-time Numeric Devnagari Sign Language Translator

B. *Pre-Image Processing*

The basic function of pre-image processing phase is to detect body parts from the human image. Here, we are performing two main sub-tasks namely; skin segmentation and black and white conversion as represented in Fig 4. We have discussed them as follows:

1) *Skin Segmentation*

Using the function rgb2ycbcr, input RGB image is converted into YCbCr color format. In YCbCr image, chrominance pixel ranges are defined for skin pixel detection and luminance Y is not used in skin segmentation. If YCbCr image pixels lie within the defined ranges of chrominance, then the pixels are considered to be the skin color pixels thereby may be assumed as body parts in image. The remaining pixels are assigned to be black color value for background subtraction.

2) *Black and White Conversion*

The output of skin segmentation phase is the binary image in YCbCr color format. Further, these images are converted back into RGB color format. The non-skin pixel values are already set to black and skin pixels are assigned as the white pixel values.

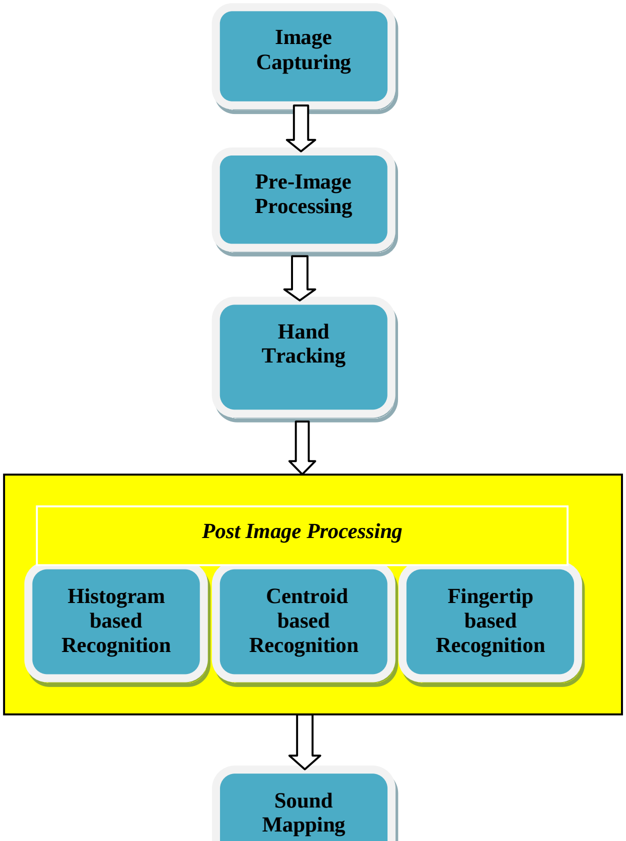

Figure 2 System Architecture of RTNDSLT

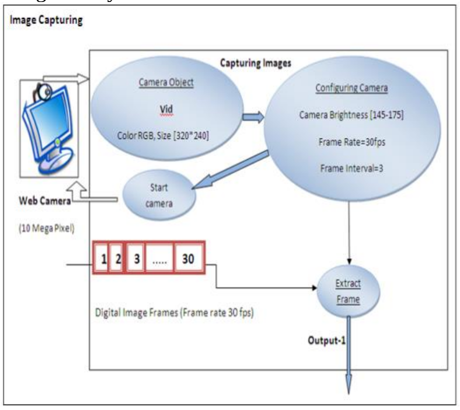

Figure 3 Image Capturing using Web Camera

C. *Hand Tracking*

This phase is the most important phase of RTNDSLT architecture. Morphological operation, hand detection and tracking are performed in this phase. The input to this phase is black and white image which may contain noise. The morphological operation and blob extraction are the sub-phases as presented in Fig 5.

1) *Morphological Operation*

These operations include smoothening, dilation of image. Some of the pixels may be lost in black and white image. The noisy input image is always rough and cannot be used for any meaningful decision. To recover the lost pixels and to smoothen the image, some logical operations are applied. The dilatation process is executed by introducing 4 pixels in width and 75 pixels in height in each line. At this stage, the holes filling process is applied with four pixels connectivity in the image. Thus, the image obtained is a new embedded version of image i.e. smooth and noise free.

2) *Binary linked object (Blob) Extraction*

Blob analysis and extraction are required for detection of hand object in human image.

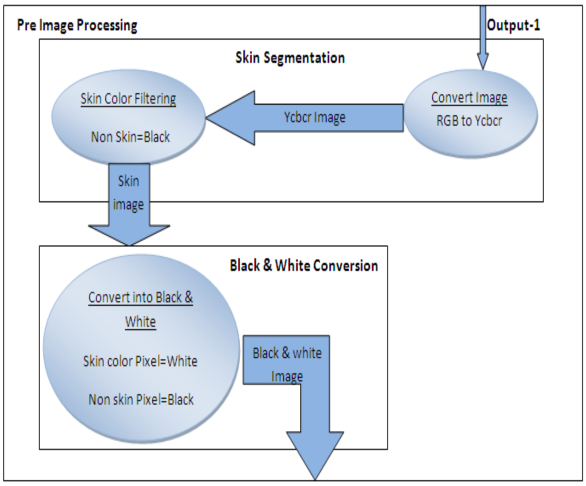

Figure 4 Pre-Image Processing of Human Image

D. Post-Image Processing

The objective of recognition of hand gesture is to provide low execution time and higher recognition rate. This phase may be accomplished by using one of the three recognition methods namely; Histogram Based Recognition, Centroid (Center of Mass) Based Recognition and Fingertip Recognition. Each recognition method is further divided into two common sub-phases; firstly feature extraction and another is numeric hand gesture recognition. We explain each recognition method along with its sub-phases as follows:

1) Histogram Based Recognition Method

Histogram represents graphical plots for the number of pixels for each color value. The extracted blob is a binary (i.e. black and white) image. Hence, it contains only two tons of which first is represented by black pixel and other by white pixels as shown in Fig. 6.

In this phase, we construct sample image datasets for each numeric hand sign and then extract image histogram of each sample image dataset. The sample image histogram values are stored in histogram dataset for each hand sample dataset. This phase may be further divided into two sub-phases as depicted in Fig. 7. The sub-phases are namely; histogram feature extraction and histogram numeric hand sign recognition and are discussed below:

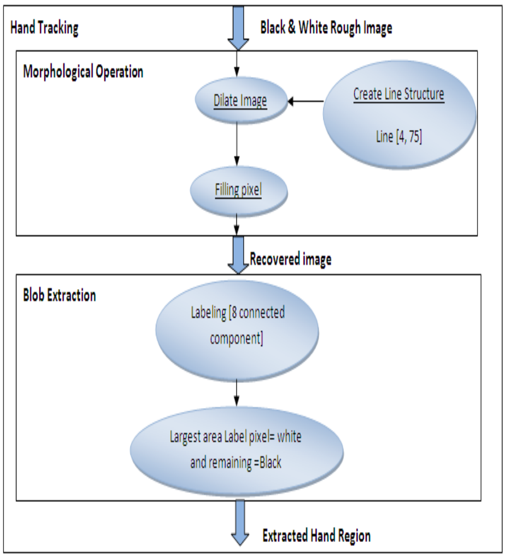

Figure 5 Hand Tracking Process

- *Histogram Feature Extraction*

The input image of size [320*240] pixels is converted into image of size [160*160] pixels i.e. compressing the image thereby forced to speed up the RTNDSLT system. At this stage, we also extract histogram feature of the resized image.

- *Histogram Numeric Hand Sign Recognition*

The Euclidean distance dataset is constructed by calculating distance between input image histogram value and sample image histogram dataset values. A function min is used to select a minimum distance from the resultant Euclidean distance dataset.

2) Centroid Based Recognition Method

In the context of binary image, the centroid is equivalent to the center of mass. Here, centroid can be replaced by the center of mass. Centroid of a binary image is shown in Fig. 8. The following mathematical equations are used for calculating centroid (COM) of binary image I:

[Ix Iy] = Find Two Dimension Vectors (I)

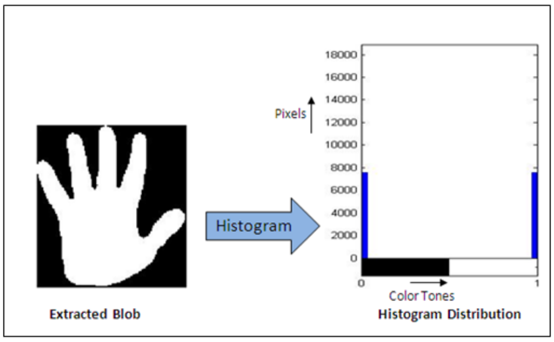

Figure 6 Binary (Black and White) Image Histogram

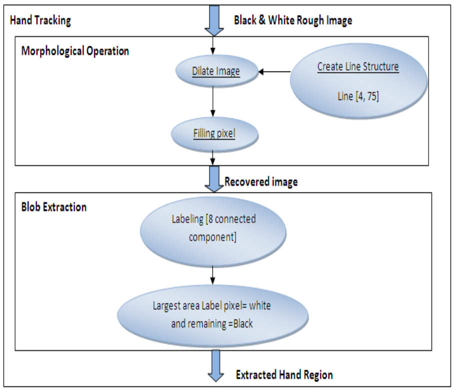

Figure 7 Histogram Based Recognition

$$\text{Mean (Ix)} = \frac{\sum_1^n \text{Ix}}{\text{n}} \qquad \dots (1)$$

where, n is number of white pixels.

$$\text{Mean (Iy)} = \frac{\sum_1^n \text{Iy}}{\text{n}} \qquad \dots (2)$$

$$\text{COM} = [\text{Mean (Ix), Mean (Iy)}] \qquad \dots (3)$$

This phase may be further divided into two sub-phases namely; centroid feature extraction and centroid numeric hand sign recognition as represented in Fig. 9.

- *Centroid Feature Extraction*

The image resize function is discussed in the previous Histogram Based Recognition method is same in this phase. Here, we try to extract the centroid feature from resized image for further processing.

- *Centroid Numeric Hand Sign Recognition*

The centroid value of input image is compared with centroid values of sample dataset with the use of Euclidean distance and further these distances are stored in distance dataset. A function min is used to select a minimum distance from the resultant distance dataset, and then the minimum distance value is compared with defined centroid threshold T as follows:

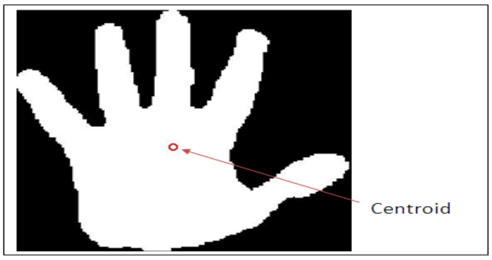

Figure 8 Binary (Black and White) Image Centroid

3) *Fingertip Recognition Method*

This method used to locate open finger peak points in detected hand region. In this phase, we extract hand contour from extracted hand region where a hand contour is series of edge points of hand object. We need to determine fingertips in hand contour at this stage. Fingertips can be defined as points at end of a smooth hand contour as shown in Fig. 10. This phase may be further divided into two sub-phases namely; fingertip feature extraction and fingertip numeric hand sign recognition as depicted in Fig. 11 and are discussed as follows:

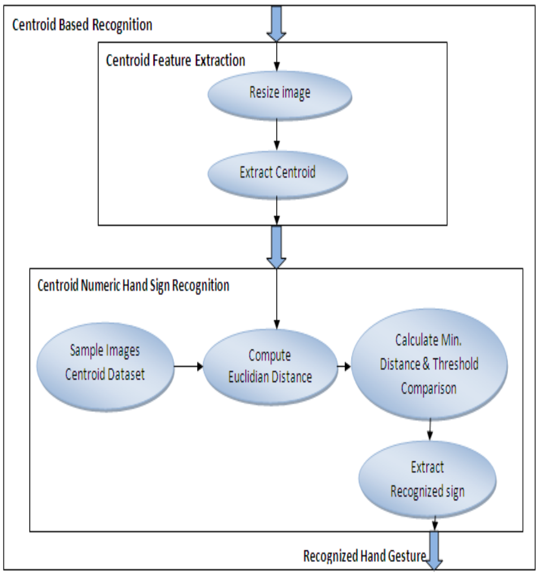

Figure 9 Centroid Based Recognition Method

- *Fingertip Feature Extraction*

At this stage, we create contour set of filtered hand image which contains boundary (edge) points. Thus, we have developed two feature sets of an image; a convex hull set and a contour set.

- *Fingertip Numeric Hand Sign Recognition*

This sub-phase is initiated with the peak and valley point detection algorithm. Contour points set acts as input for peak and valley detection algorithm. Using this algorithm, we can find each curve point from contour set namely; peak valley points. Then, the distance calculation function is applied to compute Euclidean distance between all peak valley points and convex hull set. Only those peak valley points are considered as peaks which are under the ten pixel distance from the convex hull points. The Count Total function produces a total number of peaks of hand contour.

C. Sound Mapping

Sound mapping generates sound in respect to recognized numeric hand gesture for each digit in RTNDSLT system (i.e. from 0 to 10).

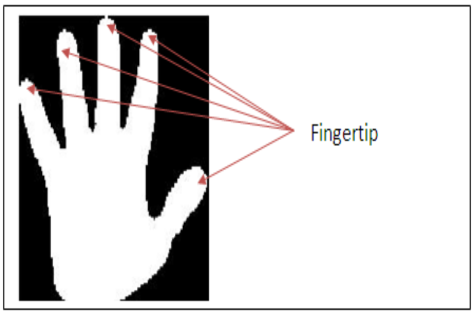

Figure 10 Fingertips in Hand Image

It is observed that lower sampling rates are beneficial for small data size and easier storage and transport. According to Nyquist-Shannon theorem, sampling rates higher than about 50 kHz to 60 kHz cannot supply more usable information for human listeners. The Audio Engineering Society recommends 48 kHz sample rate for most applications. On the other hand, it suggests 44.1 kHz for Compact Disc and other consumer uses. In our experimentation, we have used the audio sampling rate as 11025 Hz in our recording function. It may be observed that sampling rate of 11025 Hz is one quarter of the sampling rate of audio CDs. It is used for lower-quality PCM, MPEG audio and for audio analysis of subwoofer band passes.

Figure (a) Figure (b) Figure(c)

In this phase, the input consists of numeric digit of the recognized hand sign. Here, we load the wave library and then send the audio signals to the speaker for generating sound with respect to numeric digit using play function. The sound mapping is depicted in Fig 12.

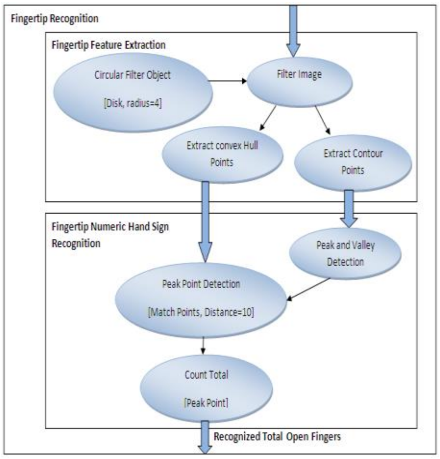

Figure 11 Fingertip Recognition

III. WORKING OF RTNDSLT SYSTEM

We discuss the working of RTNDSLT system with the help of experimentation in brief. Also, the prerequisite requirements, skin color segmentation, morphological image processing, hand blob detection and hand sign recognition have been explained as the important part of our system in subsequent sections.

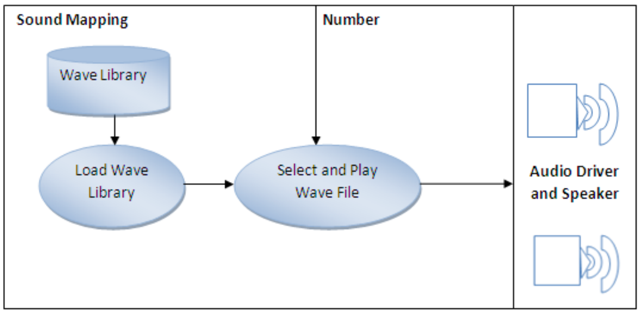

Figure 12 Sound Mapping Process

A. Pre-requisite Requirements

In our experimentation, there exist some hardware and software requirements. The hardware requirements include processor 3.06 GHz, RAM 1 GB, Web camera 10 Megapixel, web camera frame rate 30 fps (frames per second). The web camera and light source must be located in front of the user. In

Figure 13(a) Input as RGB Image for RTNDSLT system, 13 (b) Converted YCbCr Image of Input Image, 13 (c) Skin Segmented Image of YCbCr Image

our experiments, odd frame (i.e. only one or three) grab interval is applied to incoming frame buffer thereby creating live vision environment scenario. If image size is large then RTNDSLT system takes more time in image processing whereas it is small then noise cannot be removed from image in further phases. Therefore, we have fixed our input image of size as 320*240 pixels in height.

B. Skin Color Segmentation

Skin color segmentation uses color transformation for skin pixel classification. Many color spaces exist for skin segmentation such as RGB, HSV, YIQ and YCbCr. Out of these color spaces, it has been observed that YCbCr color space is effective in skin segmentation [3, 11]. YCbCr color space is used in our experiment as the skin-color is more centralized in the YCbCr color space. In YCbCr color, It is constructed as a weighted sum of the RGB values as shown in Eq. (4). Cr and Cb are calculated by subtracting luminance from red and blue components as shown in Eq. (5) and Eq. (6) respectively.

$$Y = 0.299\,R + 0.587\,G + 0.114\,B \quad \ldots (4)$$
$$Cb = R - Y \quad \ldots (5)$$
$$Cr = B - Y \quad \ldots (6)$$

For skin segmentation process, the RGB image as shown in Fig. 13 (a) is input. By setting upper and lower bound thresholds, it becomes possible to distinguish the skin color from non-skin color regions. It also subtracts background from the input image. In our experimental environment, we have defined the threshold values from 133 to 173 for Cr and from 77 to 127 for Cb. Skin segmented image is converted back into RGB as depicted in Fig. 13 (b) and further this RGB image is transformed into binary image as shown in Fig. 13 (c).

C. Morphological Image Processing

Morphological operations are used such as dilation and erosion as depicted in Fig. 14, white and grey pixels contain zero and non-zero values respectively. A morphological operation on a binary image creates a new binary image in which the pixel contains a non-zero value only if the test is successful at particular location in input image.

The dilation of an image (f) by a structuring element (s) produces a new binary image (g) with ones in all locations (x, y). The dilation operation is denoted by $\oplus$ symbol and is used in Eq. (7) to compute g. If all locations possess ones, then the structuring element s hits or intersects the input image f. This action may be shown mathematically by Eq. (8).

$$g = f \oplus s \quad \ldots (7)$$

$$g(x, y) = \left.\begin{array}{ll} 1 & \text{if s hits f} \\ 0 & \text{otherwise} \end{array}\right\} \quad \ldots (8)$$

Dilation process includes boundaries of regions. Since skin segmented image contains noise in the form of black pixels, therefore the image will be dilated with lines. In our experiment, we have assumed the line size as 4*75 pixels. Thus, the input image is partially recovered by line dilation as shown in Fig.15.

After line dilation, it may be possible that some noise may still exist in this partially recovered image. And hence, noise region holes are filled with connected four components pixel connectivity to recover complete image. As a result, the image gets fully recovered as shown in Fig. 16.

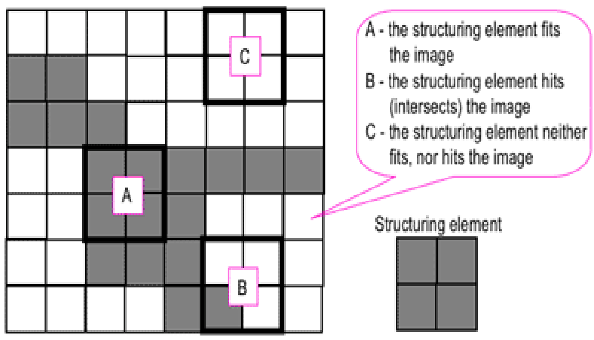

Figure 14 Structuring Element

Figure 15 Partial Recovered Image after Line Dilation

Figure 16 Full Recovered Image after Holes Filling Process

D. Hand Blob Detection Process

Blob detection is an application in object recognition and/ or object tracking. Blob detection is usually performed after color detection and noise reduction to find the required object from the image finally. A lot of important blobs that are not important may be present in the input binary image. At the same time, we need the details of the blob pixels such as centre, corners and coordinates.

Component connectivity can be defined by touching pixels as adjacent pixels along the vertical axis, horizontal axis and diagonally adjacent pixels. 4-connectivity is comprised of adjacent pixel (i.e. left, right, up and down) excluding diagonal pixel. 8-connectivity includes adjacent pixel with diagonal pixel (i.e. left, right, up, down and diagonals) as presented in Fig.17.

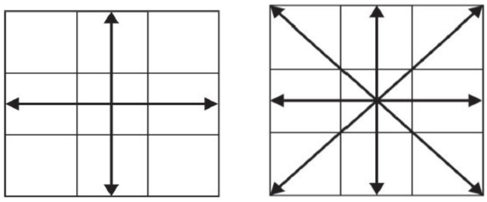

Figure 17 (a) 4 - Connectivity Figure 17 (b) 8 -Connectivity

Connected component labelling partitions image graph can be 4-connected or 8-connected as shown in Fig. 18. In case of 4-connected component connectivity, two labels will be present in image graph otherwise in case of 8-connected component connectivity, a single label will exist in image graph. The input image contains hand as foreground body part in each image, since the user shows hand signs in front of the web camera. Thus, hand is a biggest region in noise recovered image as shown in the previous section. Biggest-Blob algorithm is used to etract largest Blob from image. The input to this algorithm is a noise recovered image and output is extracted hand.

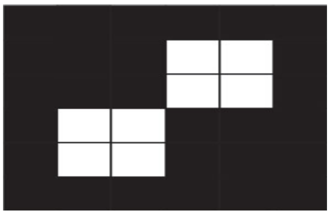

Figure 18 One or Two Labels in Image

After the execution of Biggest-Blob algorithm, the biggest region of hand gets detected. Single execution of an algorithm causes to track only one hand. Let us consider it as X. In case of two hands detection with the input image Y, X is subtracted to return the biggest region image. Thus, the subtracted image (Y-X) will be passed as input to the Biggest-Blob algorithm for detection of another hand. We have shown one hand and two detected hands in Fig 19 (a) and Fig. 19 (b). In Fig. 19 (b), two hands appear in the image, some noise gets introduced at the edges. Even in one hand case, some noise may be seen at the corner of the hand image. The filter with circular disk of radius 4 pixel may be used to smooth the hand image and the curve at the boundary sharp. The resultant smooth images with one hand is depicted in Fig. 20 (a) and two hands is represented in Fig. 20 (b).

E. Hand Sign Recognition

Hand signs are treated as hand gestures whose recognition is very complicated task since the hand is a rigid object. Hand gesture recognition interprets human gestures using mathematical algorithms. Hand gestures are originated from hand motion or state. A more specific domain of hand sign language is numeric hand sign language.

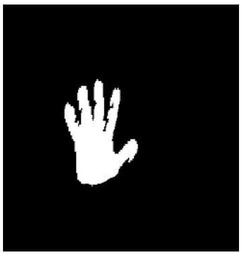


Figure 19 (a) Figure 19 (b)

Figure 19 (a) One Hand Detection using Biggest-blob Algorithm,

Figure 19 (b) Two Hands Detection using Biggest-blob Algorithm

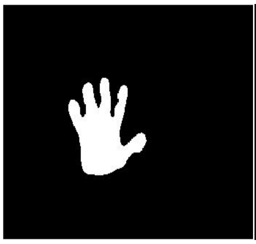


Figure 20 (a) Figure 20 (b)

Figure 20 (a) Smooth One Hand after Circular Filter Process,
Figure 20 (b) Smooth Two Hands after Circular Filter Process

For better recognition results, recognition methods must be satisfied shape feature properties such as rotation invariant, scale invariant, noise resistance and statically independent. We have developed three different recognition methods namely: Histogram Based Recognition, Centroid Based Recognition and Fingertip Recognition. Knowledge dataset will be required in Histogram Based Recognition and Centroid Based Recognition. We have constructed knowledge dataset using sample image database and feature extraction. The open fingers will be detected by Fingertip Recognition. Sample image database, Histogram Based Recognition, Centroid Based Recognition and Fingertip Recognition are briefly discussed in subsequent sections.

1) *Sample Image Database*

The sample image database is a collection of noise free images for each sign. Initially, our detection is concerned with five hand signs i.e. from 1 to 5 using right hand only. We have developed Sample-Image-Database algorithm for capturing and storing images in image database. This algorithm is divided into three modules viz. skin segmentation, morphological image processing and hand blob detection as discussed in previous sections. Some image samples for recognition of each hand sign ranging from one to five are shown in Fig. 21 (a) to Fig. 21 (e) respectively.

Knowledge dataset is a collection of information of each image sample, which is used in recognition of an input hand image. This information forms a feature of an image. System intelligence is dependent on knowledge dataset. In numeric hand sign recognition, shape feature is used in knowledge dataset. At this stage, we have developed histogram feature knowledge dataset and centroid feature knowledge dataset. We have developed MATLAB program for developing histogram (Refer Appendix I) and centroid knowledge dataset file of one numeric hand sign.

2) *Histogram Based Recognition*

The histogram is a graph, which represents the number of pixels in an image at each different color value found in an image. For an 8-bit grayscale image, histogram is graphically concerned with 256 color distribution of pixels. The input to Histogram Based Recognition is smooth black and white hand image and therefore the input image histogram consist of only two color distribution i.e. for black pixels and white pixels. We need to discuss the concept of Euclidean distance (E_d) as it plays a significant role in the recognition methods of our interest.

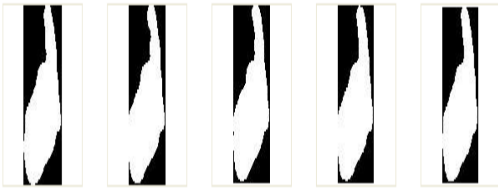

Figure 21(a) One Hand Sign Image Samples

Figure 21 (b) Two Hand Sign Image Samples

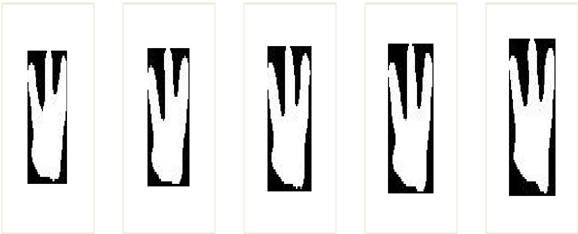

Figure 21 (c) Three Hand Sign Image Samples

Figure 21 (d) Four Hand Sign Image Samples

Figure 21 (e) Five Hand Sign Image Samples

In two-dimensional geometry, we can compute this E_d (A, B) between two points A and B having the co-ordinates (Ax, Ay) and (Bx, By) respectively, using Eq. (9).

$$E\_d\ (A,B) = \sqrt{(Ax - Bx)^2 + (Ay - By)^2} \qquad \dots (9)$$

While using Histogram-Recognition algorithm for recognition of five numeric hand signs, our input is a smooth (filtered) one hand image with size 320*240 pixels. In the first step, hand region is cropped from input image and then cropped hand image is resized with 160 pixels in height and 160 pixels in width (same as sample image resized before feature extraction).

In computer vision, when the distance increases between camera and object, then object becomes smaller in size and vice versa. Hence, the concept of resizing satisfies size invariant property of shape. We present Histogram-Recognition algorithm is presented below:

3) *Centroid Based Recognition*

Centroid is average of all pixel points present in image. The center of mass is the arithmetic mean of all points weighted by the local density or specific weight. If a physical object has uniform density, then its center of mass is the same as the centroid of its shape. In black and white image, a region is represented by white pixels, the white pixels contain uniform density. Thus, the center of mass and centroid can be replaced in context of black and white image.

We have also developed Centroid-Recognition algorithm for recognition of numeric hand gesture. The primary steps discussed in Histogram-Recognition algorithm such as load dataset files, crop hand region and resize image are same in this algorithm also. In this algorithm, we have calculated centroid distance between smooth (filtered) input hand image centroid and centroid knowledge datasets using E_d discussed above.

4) *Fingertip Recognition*

Fingertip Recognition detects open finger or fingers in filtered hand image. We may execute this technique on one hand or two hands (both right and left hand). Each extracted filtered (smooth) hand image is passed separately as input in fingertip recognition. Some of the essential image features and algorithms are used in Fingertip Recognition. These image features are image contour, image convex hull and binary image centroid whereas algorithms used are Peak and Valley Detection and Peak Point Detection. Now, we focus on image contour finding, convex hull and the aforesaid algorithm in subsequent sections.

a) Contour Finding Method

Use of this technique leads to lower computation cost along with the contour information which we require in decision making for fingertip detection. When contour search is finished, it indicates that a hand contour is found from smooth hand image. A hand region and a hand contour may be plotted on RGB input image as shown in Fig. 22. A rectangle is denoted by hand region, it is boundary box of extracted hand. A contour is denoted by boundary edge of extracted hand. Hand boundary box size is the criteria to decide whether valid input or not. In our experiment, the hand boundary box size is defined as 30 pixels in width and 60 pixels in height. If any input hand image violates the boundary box criterion, then it will not be processed further. Only valid images are accounted for further processing.

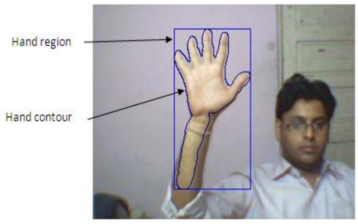

Figure 22 Hand Regions and Contour of Filtered Hand Image

b) *Image Convex Hull*

We calculate the convex hull of the fore-arm contour in order to find the desired information. Usually, the arm part is a smooth contour and contains important information but it is not sufficient. The hand may contain more convex contours with the desired information. We need to explain the concept of convex polygon which will be further used in convex hull. A convex polygon is a polygon which does not contain any concave part. Fig.23 (a) shows a concave part of a polygon by an arrow.

For example, in Fig.23 (b), there are ten points in convex polygon. The hexagon in the Fig.23 (b) is the convex hull including the set of ten points. The six points are composed the hexagon are called "hull points".

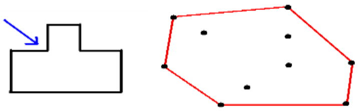

Figure 23(a) Not a Convex Polygon

Figure 23(b) A Convex Hull with 10-points

A function convhull successfully generate the convex hull of the hand region as shown in Fig. 24. By observing the contour and its convex hull, we notice that some areas are contained in convex hull but not in the contour, are called the convexity defects. In the case of fore-arm contour, we find out that the convexity defects are often being around the palm.

c) *Peak-and-Valley Detection Algorithm*

Hand Peak and valley are curve points denoting peaks or valley points in hand contour. A contour plot is required in peak and valley detection for traversing edge boundary of an object. A hand contour set consists of all the edge boundary points. Peak and Valley Detection algorithm assumes contour points set as input and further more it works on the pair of three points.

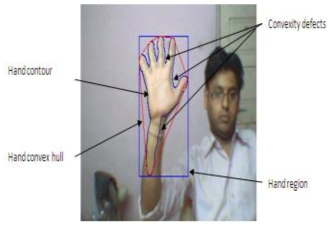

Figure 24 Hands Convex Hull and Convexity Defects

Using this algorithm, we can create pair of three points in such a way that if first point is at k^{th} location say X1, then next points will be located at $(k+16)^{th}$ location i.e. X, $(k+32)^{th}$ location say X2 in contour set. Two lines L1 and L2 are formed between the points X1 and X; and X and X2 respectively. Also, the angle between two lines L1 and L2 can be found out using Eq. 10:

$$\theta = \cos^{-1}\left(\frac{uv}{||u||\ ||v||}\right) \qquad \ldots (10)$$

Where, u = vector difference between X and X1.

v= vector difference between X2 and X.

In our algorithm, angle between the lines (θ) is computed and further compared with the threshold. If $\theta \leq$ 90 degrees, the middle point $(K+16)^{th}$ is marked as desired point considering it as either peak or valley. Now, X1 is shifted at location $(k+33)^{th}$ in contour set. If $\theta >$ 90 then we may traverse each point in contour set. The process of traversing is stopped until last point in contour set is visited.

For identification of peaks and valleys in hand contour, an algorithm is designed namely: Peak and Valley Detection. Hand contour points set passes as input in this algorithm. In Peak and Valley Detection algorithm, contour points are traversed linearly with use of pair of three points, angle of a point is computed and compare with threshold (i.e. $\theta \leq 90°$) and detected peak and valley points set is return as output. A contour plot of any image (Cp) is shown in Fig. 25(a). Peaks and valleys may be detected in Cp using Peak and Valley Detection algorithm as shown in Fig. 25(b). We discuss this algorithm with its execution run below:

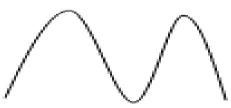

Figure 25 (a) Cp for Any Image

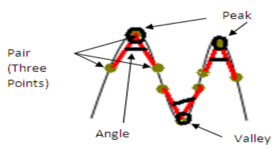

Figure 25 (b) Peaks and Valleys in Cp

After the execution of Peak and valley Detection algorithm, the identified peak and valley points in hand contour are denoted by star (*) symbol as depicted in Fig 26. Out of them, some peaks or valleys may be valid or noisy. The existence of curve at the bottom of hand contour shows the noise peak or valley as shown in Fig.27. Let us consider H as the hand region boundary box as shown by Eq. (11). To prevent from noise points at bottom, the biggest hand region image is cropped with size S, given by Eq. (12).

$$H = [x, y, w, h] \qquad \dots (11)$$

$$S = [0, 0, (x + w), (y + h - 10)] \qquad \dots (12)$$

where, (x, y) represents the location of H in two dimensional matrix of image, w (width) and h(height).

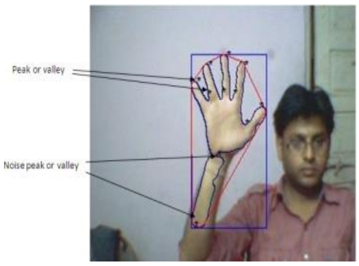

Figure 26 Detected Peak and Valley Points in Hand Contour

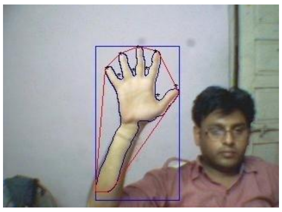

Figure 27 Prevented from Noisy Peaks or Valleys at Bottom of Hand Contour

Now, cropped input image is passed as input for convex hull and contour finding processing. The bottom may be eliminated from hand region thereby eliminating the noise. Outcome of RTNDSLT system have been shown for each hand sign from zero to ten in Fig. 28.

IV. EXPERIMENTAL RESULTS

We have developed RTNDSLT system using Histogram Recognition, Centroid Recognition and Fingertip Recognition. A comparison of these techniques has been performed on the basis of properties of shape features namely; rotation invariant, scale invariant, noise resistance and statistical independence. It is observed that Histogram Recognition and Centroid Recognition techniques are only scale invariant but not rotation invariant, noise resistant and statistical independent. On the other hand, Fingertip Recognition technique is rotation invariant, scale invariant, noise resistant and also statistical independent as shown in Table 1. On the basis of random input sample size, we have performed experiments using the aforesaid recognition techniques under the homogeneous environment as depicted in Table 2. The real time recognition rates associated with the number of recognized numeric hand signs show the performance of each technique. A comparative study of each recognition technique based on recognition rate is depicted in Table 2.

V. CONCLUSION

Using Histogram Recognition, we have observed the recognition rate as 38.42% with the execution rate nine frames per second whereas the recognition rate as 52.85% in Centroid Recognition, with the same execution. It is most important to note that recognition rate is 94.13% using Fingertip Recognition technique in our RTNDSLT system. Thus, the techniques Histogram Recognition and Centroid Recognition are not suitable in recognition of numeric hand signs even if these execute faster as compared to Fingertip Recognition technique. The RTNDSLT system produces output in form of text as well as audio of recognized numeric hand gesture. However, there is scope to increase recognition success rate and enhance efficiency for two digit numbers in RTNDSLT system.

TABLE 1

COMPARATIVE STUDY OF RECOGNITION TECHNIQUES BASED ON SHAPE FEATURE PROPERTIES

REFERENCES

[1] J. R. Pansare and M. Ingle, “Real-time static hand gesture recognition for American Sign Language (ASL) in complex background”, *Journal of Signal and Information Processing*, vol. 3, no. pp. 364-367, 2012. , ISSN: 2159-4465, doi: 10.4236/jsip.2012.33047.

[2] P. SubhaRajam, Dr. G. Balakrishnan “Real Time Indian Sign Language Recognition System to aid Deaf-dumb People”, Communication Technology (ICCT), *IEEE 13th International Conference on Signal Processing & Analysis, 2011*, pp. 737-742, doi:10.1109/ICCT.2011.6157974.

[3] Baoyun Zhang, Ruwei Yun “Robust Gesture Recognition Based on Distance Distribution Feature and Skin-Color Segmentation”, *Audio Language and Image Processing (ICALIP), 2010 International Conference on Signal Processing & Analysis*, pp. 886- 891, doi: 10.1109/ICALIP.2010.5685201.

[4] Gang-Zeng Mao, Yi-Leh Wu, Maw-KaeHor, Cheng-Yuan Tang “Real-Time Hand Detection and Tracking against Complex Background”, Intelligent Information Hiding and Multimedia Signal Processing 2009 IIH-MSP '09, Fifth International Conference on Signal Processing & Analysis, pp. 905- 908, doi: 10.1109/IIH-MSP.2009.133.

[5] Yikai Fang, Kongqiao Wang, Jian Cheng and andHanqing Lu “A Real-Time Hand Gesture

[6] Syed AkhlaqHussain Shah, Ali Ahmed, IftekharMahmood and KhurramKhurshid, “Hand gesture Based User Interface for Computer Using Camera and Projector”, *Signal and Image Processing Applications (ICSIPA), IEEE International Conference on Signal Processing & Analysis, 2011*, pp. 168-173, doi: 10.1109/ICSIPA.2011.6144111.

[7] Dong Wang, “GIUC: A Gesture Interface for Ubiquitous Computing”, *Communications and Mobile Computing 2009, CMC ’09, WRI International Conference on ; Computing & Processing*, vol. 3, pp. 170-174, doi: 10.1109/CMC.2009.324.

[8] Miaolong Yuan, FarzamFarbiz, Corey Mason Manders and Tang Ka Yin, “Robust Hand Tracking Using a Simple Color Classification Technique”, International Journal of Virtual Reality, 2009, vol. 8, no. 2.

[9] FahadUllah, “American Sign Language Recognition System for Hearing Impaired People Using Cartesian Genetic Programming”, *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on Signal Processing & Analysis*, pp. 96-99, doi: 10.1109/ICARA.2011.6144863.

Recognition Techniques	**Shape Features**			
	Rotation Invariant	**Scale Invariant**	**Noise Resistance**	**Statistical Independent**
Histogram Recognition	No	Yes	No	No
Centroid Recognition	No	Yes	No	No
Fingertip Recognition	Yes	Yes	Yes	Yes

TABLE 2 COMPARATIVE STUDY OF RECOGNITION TECHNIQUES BASED ON SHAPE FEATURE PROPERTIES

Recognition Methods/ Numeric Sign	**Histogram Recognition**			**Centroid Recognition**			**Fingertip Recognition**		
Comparative Measure	**n**	**M**	**r**	**n**	**m**	**r**	**N**	**m**	**R**
One **(1)**	89	47	52.8	95	76	80	226	216	95.57
Two **(2)**	96	28	29.16	96	30	31.25	227	217	95.59
Three **(3)**	95	34	35.78	95	47	49.47	221	215	97.28
Four **(4)**	96	14	14.58	95	16	16.84	211	200	94.78
Five **(5)**	95	58	61.05	92	81	88.04	202	198	98.01
Six **(6)**	-	-	-	-	-	-	221	200	90.49
Seven **(7)**	-	-	-	-	-	-	205	190	92.68
Eight **(8)**	-	-	-	-	-	-	215	200	93.02
Nine **(9)**	-	-	-	-	-	-	196	180	91.83
Ten **(10)**	-	-	-	-	-	-	198	186	93.93
Zero **(0)**	-	-	-	-	-	-	247	228	92.30
Total	**471**	**181**	**38.42**	**473**	**250**	**52.85**	**2369**	**2230**	**94.13**

Where n = Input Sample Size, m = No. of Recognized Samples, r= Recognition rate (%).

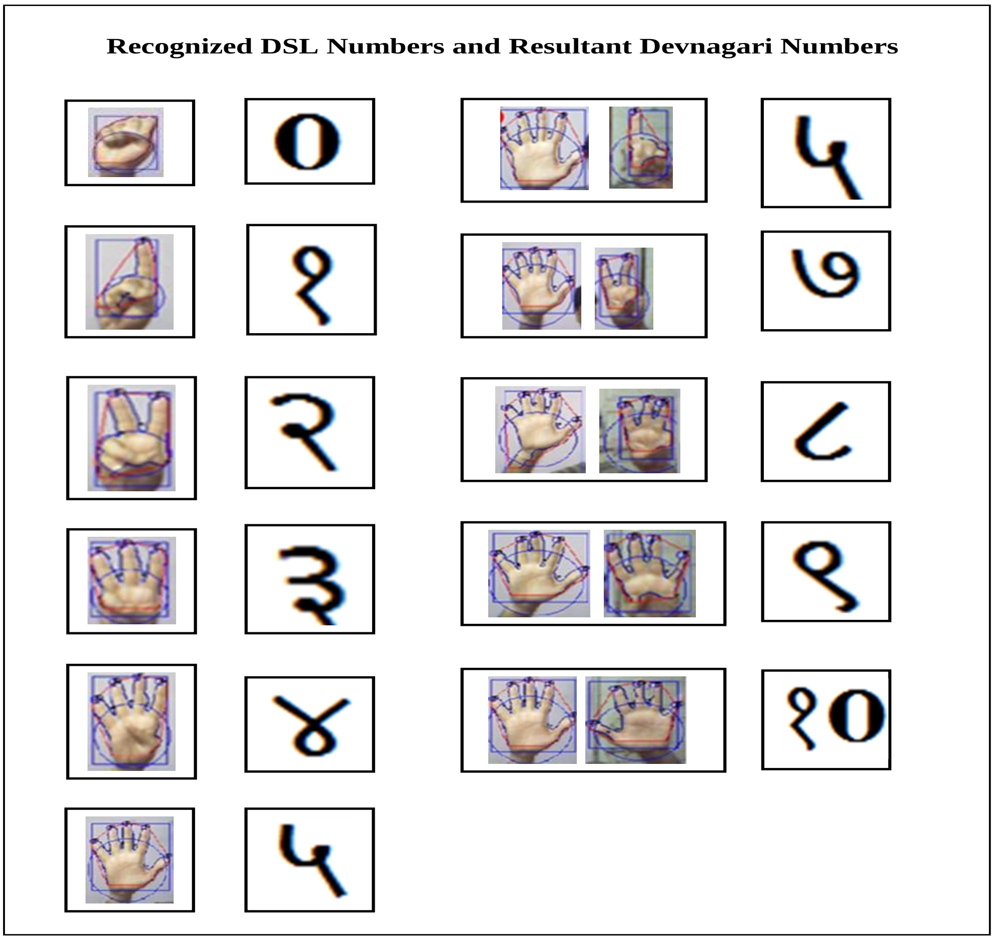

Figure 28 Recognized Numeric Hand Signs for 0 to 10 DSL Numbers

[10] ZhengmaoZou, PrashanPremaratne, Ravi Monaragala, NalinBandara, MalinPremaratne "Dynamic Hand Gesture Recognition System using Moment Invariants", *5th International Conference on Signal Processing & Analysis, 2010*, pp. 108-113, doi: 10.1109/ICIAFS.2010.5715644.

[11] Shuying Zhao, Wenjun Tan, Chengdong Wu, Chunjiang Liu, Shiguang Wen, "A Novel Interactive Method of Virtual Reality System Based on Hand Gesture Recognition", *Control and Decision Conference, 2009. CCDC '09 topic on Computing & Processing*, pp. 5879-5882, doi: 10.1109/CCDC.2009.5195252.

[12] Kwok-Wai Wong, Kin-Man Lam and Wan-Chi Siu "An Efficient Color Compensation Scheme For Skin Color Segmentation", *International Symposium on Signal Processing & Analysis*, vol. 2, pp. II-676 - II-679, doi: 10.1109/ISCAS.2003.1206064.

[13] J. R. Pansare and M. Ingle, "Comprehensive Performance Study of Existing Techniques in Hand Gesture Recognition System for Sign Languages," Int. J. Comput. Sci. Inf. Technol., vol. 7, no. 3, pp. 1343–1347, 2016.

[14] J. R. Pansare and M. Ingle, "A Real-time Devnagari Sign Language Recognizer (α-DSLR) for Devnagari Script", in Springer Proceedings of Worlds Conference on Smart Trends in Systems, Security and Sustainability (WS4 2017), 2017, London, United Kingdom. (In print)

[15] J. R. Pansare and M. Ingle, "2D Hand Gesture for Numeric Devnagari Sign Language Analyzer Based on Two Cameras", *A. Basu et. al. (Eds): IHCI 2016, Lecture Notes in Computer Science (LNCS), Springer*, vol. 10127, pp. 148-160, 2017, ISSN: 0302-9743, ISBN: 978-3-319-52502-0, doi: 10.1007/978-3-319-52503-7.

[16] J. R. Pansare and M. Ingle, "Vision-based Approach for American Sign Language using Edge Orientation Histogram", in IEEE Proceedings of 2016 International Conference on Image, Vision and Computing (ICIVC 2016), University of Portsmouth, United Kingdom, pp. 90-94, 2016, ISSN: 978-1-5090-3754-4/16, doi: 10.1109/ICIVC.2016.7571278.

[17] J. R. Pansare and M. Ingle, "Real-time Static Hand Gesture Recognition for American Sign Language", in Proceedings of the ACM Symposium on Women in Research 2016 (ACM-WIR 2016), Sri Aurobindo Institute of Technology, Indore, India, 2016.

[18] J. R. Pansare and M. Ingle, "Comprehensive Performance Study of Existing Techniques in Hand Gesture Recognition System for Sign Languages", *International Journal of Computer Science and Information Technology*, vol. 7, no. 3, pp. 1343–1347, 2016, ISSN: 0975-9646.

[19] J. R. Pansare, *et. al*., "Gestulle: A System to Recognize Dynamic Hand Gestures using Hidden Markov Model to Control Windows Applications", *International Journal of Computer Applications*, vol. 62, no.17, pp. 19-24, 2013, ISSN: 975-8887.

[20] J. R. Pansare, *et. al.*, "Real-time Static Devnagari Sign Language Translation using Histogram", *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 2, no. 4, pp. 1-5, 2013, ISSN: 2278-1323.

[21] J. R. Pansare, *et. al.*, "Real Time Static Hand Gesture Recognition System in Complex Background that uses Number system of Indian Sign Language", *International Journal of Advanced Research in Computer Science*, vol. 2, no. 3, pp. 1086–1090, 2013, ISSN: 2278-1323.

[22] J. R. Pansare, *et. al.*, "Real-time Static Hand Gesture Recognition System using HCI for Recognition of Numbers", *International Journal of Advanced Research in Computer* Science, vol. 4, no.4, pp. 258-262, 2013, ISSN: 0976-5697.