International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

SIGN LANGUAGE AND GESTURE RECOGNITION FOR DEAF AND DUMB PEOPLE

Nagma Neha[1], Prof. Aditi Ravichandra[2]

[1]*UG Student, Computer Science, Atria Institute of Technology, Bengaluru, India*
[2]*Assistant Professor, Computer Science, Atria Institute of Technology, Bengaluru, India*

---***---

Abstract - Human-computer interaction is very much essential part of most people's daily life. The human and computer interaction mode is through keyboard, mouse, joystick, and wireless input devices. Interaction between people and computers have made it easier for people to operate the computer and improve work efficiency these days. But what for those who actually cannot be a part of the computer technology in these days.

Firstly Sign Language is a natural language which deaf community uses for communication. Sign Language (SL) is a subset of gestures or signs made with fingers, hands, arms, eyes, and head, face etc.in simple words Sign language is a means of communication among the deaf and dumb community. It emerges and evolves naturally within hearing impaired community.

Key words: Sign Language Recognition, Natural Language, Segmentation, Feature Extraction, Digital Image Processing, Data Input, Orientation Binning etc..

1. INTRODUCTION

The sign language differs from people to people, and from region to region. The people can make interactions with the normal people and with the people of their community as well. Each gesture in the SL has its own logic, meaning and understanding to it. Learning SL is understanding the meaning of the gestures. The gesture recognition terms are heavily encountered in human daily computer interactions. Gestures are the motion of the body or the physical action form by the human in order to convey some meaningful information and to communicate. The gesture recognition is the process by which gestures are made by the human and it's made like its known to the system or the other human. Every country has its own name for the SL like American Sign Language (ASL), British Sign Language (BSL), Turkish Sign Language (TSL), and In India we call it as Indian Sign Language (ISL). Using computer vision or machine eye, there can be great emphasis on using hand gesture as a major tool of new input technique in broad range applications. With the developed and virtual realization in the environment, current user-machine interaction tools and methods like mouse, joystick, keyboard and electronic pen are not sufficient enough, whereas the hand gesture has the natural ability to represents ideas and actions very easily.

This type of natural interaction is more convenient in our environment. Just if we ignore the world of computers and technology for a while and start interacting among human beings, we can simply realize that we are utilizing a wide range of gestures in our daily personal communications. It is also shown that people are not more conscious when they are talking on telephone and are not able to see each other as like face to face communication. The gestures are available greatly among Indian cultures and easiest way to start communication. The significant use of gestures in our daily life as a mode of interaction motivates the use of gestural interface and employs them in wide range of application through computer vision.

Aim for this research is to develop an algorithm that will translate the ISL into English, so that even Deaf and Dumb people can learn English. The accuracy of this system is as high as 97.5% and can help dumb and deaf people and can also reduce illiteracy percentage in INDIA.

2. RELATED WORK

In the paper [1] in below References, the sign and hand gestures are captured and Processed with the help of mat lab and the converted into speech & text form. The feature extraction of values of the images is evaluated based on 7Hu (7 moments) invariant moment technique and the classification techniques are applied using K-Nearest Neighbor (KNN) is compared with The PNN (Probabilistic Neural Network) for the accuracy Rate. The performance of proposed classifier KNN is decided Based on various parameters Parameters can be calculate by Formula's, using 7hu moments technique's for feature Extraction will be done, the 7Hu moments are a vector of Algebraic invariants that added regular moment. KNN (k-nearest neighbor's classification) algorithm will Classifies the objects on basis of majority votes of its Neighbors. The object is assigned to the class by Most Common among its nearest neighbor's k (k is a small positive integer).Using (KNN) classification we get approximately 82% accuracy rate. The limitations of this system is the results of speech is audio output and the gestures or signs is only on the left side of the upper half of the captured image. And Capturing of image carefully and with high quality webcam is needed.

In the paper [2] in below references, hand gesture recognition can be done by wearing gloves this proposed system can work for real-time translation of Taiwanese sign language. The different hand gesture can be identified by using the 10 flex sensors and inertial sensors which is embedded into the glove, it includes three major parameters which are;

1. The posture of fingers
2. Orientation of the palm
3. Motion of the hand

As defined in Taiwanese Sign Language and it can recognized without ambiguity. The finger flexion postures can be collected from the flex sensors, the palm orientation acquired by G-sensor, also the motion trajectory acquired by gyroscope are used as the input signals of the proposed system. The input signals will be gathered and validate or checked periodically to see if it is a valid sign language gesture or not. The flex sensor are attached with a fixed resister, to calculate the voltage between the flex sensor (RF) and the fixed resistor (RB) using below formula

$$VO = VI\ RF\ /RF + RB$$

Depending upon the values of RF &RB the angle of bent in hand figure will be obtained. The orientations of palm like up, down, right, left etc. Will be identified by using 3D data sequence along x-axis y-axis and z-axis. Once all these process done the sampled signal can last longer than a predefined clock cycles, and it is regarded as a valid hand sign language gesture and will be sent to smart phones via Bluetooth for gesture identification and to speech translation. The accuracy for gesture recognition in this proposed system is up to 94%.

In the paper [3] in below references, a vision based interface system is developed for controlling and performing various computer functions with the aim of human computer interaction. The human computer interaction aims at easy way of interaction with the system. Image segmentation and feature extraction can be done by using the vision base technology

In the paper [4] in below references, a vision based sign language recognition system using LABVIEW for automatic sign language has been presented. The goal of this project is to develop a new vision based technology for recognizing and translating continuous sign language to text. Although the deaf, hard of hearing and hearing signers can communicate without problems among themselves ,there is a serious challenge for deaf community trying to integrate into educational ,social and work environments. Here convex hull and convexity defects algorithms are programmed in Open CV

3. IMPLEMENTATION

The Implementation includes the following steps:

1. Image Segmentation
2. Orientation Detection
3. Features Extraction
4. Classification of bits Generation
5. Hand Gesture interpretation

The Flow Chart of the algorithm is:

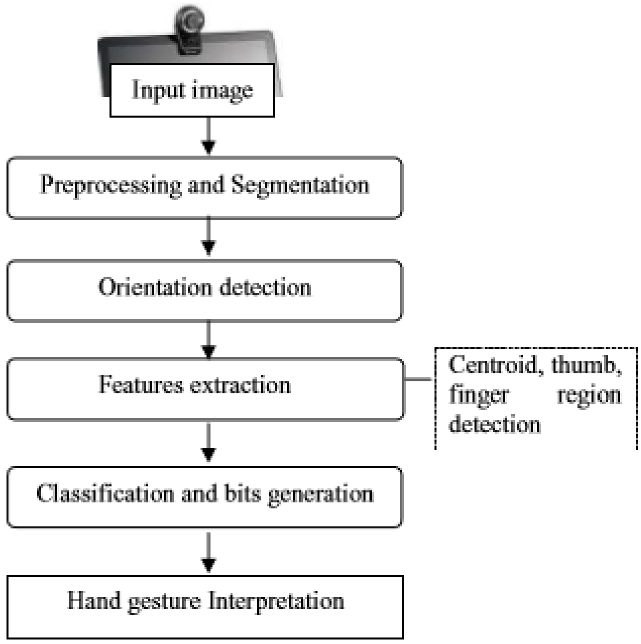


Figure-1: Flowchart of the Algorithm

3.1 Image Segmentation:

Image processing is very much necessary for image enhancement in this algorithm for achieving good results. Firstly in this algorithm, the input sequence of RGB images are converted in to YCbCr images because, as we all know the RGB color space is more sensitive and more complicated to different light conditions in the nature so we need to convert the RGB image in to YCbCr. Image segmentation is literally performed to locate the hand object in the image. The K-means clustering algorithm is the first technique which is used to convert the image into K clusters. Here the K-mean technology computes and converts the distance of each cluster in order to minimize the sum of distances from each object to its cluster and the sum cannot be decreased further. The result of K-means clustering is a set of clusters that are well separated from other clusters and compacted in their own cluster. In this approach images are in uniform. Where Plain background consist of only one hand object, so we have to use two clusters, for representing of the hand object and background of the image Cluster '1' which represents the hand, and has all pixel values set as 1 Cluster '2' which represents its background has 0 intensity pixels.
To reduce the background noise we have to remove all small insignificant smudges or connected components from the image which is already connected because it has fewer than P pixel. We apply filling of holes on binary image. Then after hand segmentation we have to calculate the boundary contours in order to locate the hand region in the image which is very important and which is performed by scanning the image from top to bottom and left to right. Then the first white pixel which is encountered is the set as left side of the

hand. Then start scanning from right to left in top to bottom manner and the first white pixel which is found is set as right side of the hand. In this way we have got the vertical bounds of hand in image .Later on within these vertical bounds we will perform a horizontal scan from left to right and top to bottom. The first white pixel encountered is fixed as top-most point of the hand. The hand extends from the bottom-most part of the image, so no scanning is needed to locate the end of the hand. That's how the first step goes and the below figure shows the Input Image, Cluster Image, enhanced Image, and Localized Hand Object.

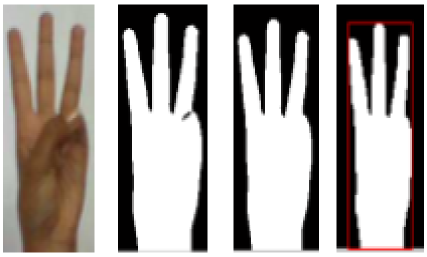


Figure-2: Input image, Cluster image, Enhanced image, and Localized hand object

3.2 Orientation Detection:

For orientation detection of hand in image we have to employ dual approach because this step is very much essential to give perfect result and the success of this approach is completely depend on this Step. Where as in this step firstly we identify whether the hand is vertical or horizontal. Then when we compute the ratio of length to width of bounding box we just keep an assumption that if the hand is vertical and then length of the bounding box is greater than the width which is found of bounding box and their ratio would be greater than 1.

What if hand is horizontal?

If the hand is horizontal then width of bounding box is greater than the length of bounding box and their ratio would be lesser than 1.

Secondly to continue the process we trace the boundary matrices or edges of hand in binary image. For horizontal hand whenever we get X boundary is equal to 1 along with the increasing value of Y boundary for some time span we classify it as the horizontal hand and if we get Y-boundary is equal to maximum of size of image with increasing value of X-boundary, it is set as vertical hand.

To reduce the uncertainties discussed above and to get proper orientation detection these two methods of orientation detection should give the same results.

The Uncertainties are, it may be possible that if any vertical image touch the left bottom corner of the image then it will show 'x=1' and 'y=max' which in turns gives an error or unexpected result or in rare condition it can also be possible that ratio of length to width for vertical image is less than 1. Here just to avoid these kinds of errors we need to consider and compare the orientation results of both methods.

In this way we categorize the two categories of hand patterns, horizontal and vertical

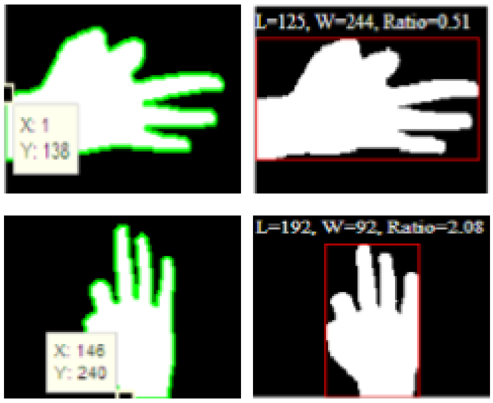


Figure-3: Horizontal image and Vertical image

3.3 Features Extraction:

Various roles to be followed to feature extraction:

3.3.1) Centroid: Centroid is also called center of mass and it divide the hand into two halves at its geometric center only if the image is uniformly distributed. The centroid is here is calculated using image moment, which is the weighted average of pixel's intensities of the image. to calculate here In this step, Firstly we have to calculate the centroid for partitioning the hand into two halves, one which represents the finger portion and other which represents non finger region and the centroid is calculated by first calculating the image moment using this formula(i)

$$M_{ij} = \sum_{x}\sum_{y} x^i y^j I(x, y) \quad \text{(i)}$$

Where M_{ij} is image moment, I(x, y) is the intensity at coordinate (x, y).

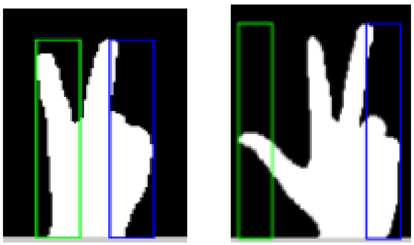
$$\{\bar{x}\ \bar{y}\} = \{M_{10}/M_{00}, M_{01}/M_{00}\} \quad \text{(ii)}$$

By using equation (ii), we compute the coordinates of centroid. x', y' are the coordinates of centroid and M_{00} is the area for binary image.

3.3.2) Thumb detection: The thumb detection step is performed in order to detect the presence or absence of thumb in hand gesture and which plays a vital role and changes the actual results too. In humans hand thumb is considered as a significant shape feature to classify various hand gestures in this approach. We know that thumb can either be reside at right most side of all finger of the hand or at left most side of the hand in general. To detect the presence of thumb in hand, we proceed with the previously calculated bounding box and consider the left side and right side of this bounding box.

By taking 30 pixels width from each side of the bounding box we crop this bounding box into two region,

One which is represented by green boundary is left box (a) And the other one is right box (b) represented by blue boundaries in the Figure-4 shown below.



(a) (b)

Figure-4: Thumb detection

Above Figure-4 shows the partition of bounding box in two boxes represented by green and blue box. It detect thumb at the left hand side and in the green box, in which percentage of white pixels is counted less than 7 % of total white pixels, moving forward after having these two boxes with us we count the total number of white pixels presents in binary image, which actually represent the hand object.

Then moving ahead we count number of white pixels present in each box that is green (a) and blue box (b) on the figure-4. If there is less than 7% of total white pixels exist in any of the right box or left box, we have to consider that thumb is present in that box only.

If both boxes having more than 7% percent of total white pixels in the image, which straightly means the thumb is not present in any of the box.

If both boxes having less than 7% of total number of white pixels exist in image then which means thumb is not present in any of the box, because thumb is only one per hand and it cannot be detected at both side of the bounding box for the same hand gesture. This method is applicable to both categories of hand the results will be highly influenced by variation in orientation as you can see in the figure-4.

3.3.3) Finger region detection: Moving forward we Give tip of the finger as peak. For getting the total number of finger raised in hand gesture we need to process only finger region of the hand that we have got in previous step by computing centroid and to proceed this task we have to trace the entire boundary matrices of hand.

Vertical hand image and horizontal hand image have been processed in different manner. For Vertical hand image, we only consider the "y" coordinates of the boundary matrices. When we get the values of y coordinates the boundaries starts increasing after the sharp decrement in the y-boundaries value. We consider this indication as tip of the finger and we fix it as a peak value.

Similarly in Horizontal hand image, we consider the "x" coordinate of the boundary matrices. This time only the x coordinates of the boundary matrices is traced. When we get the x coordinate of boundaries it starts decreasing after the continuous increment, and we mark this point as a tip of the finger in horizontal hand and set it as peak value.

In this way we found the tip of all raised and folded fingers in the image, but still we need to classify significant peaks and insignificant peaks among them, because both gives big difference.

For this we need to proceed to the next step to calculate the Euclidean distance.

3.3.4) Euclidean distance: By proceeding to the next step, After marking the detected peaks or tip of the fingers in the hand in the precious step then we must find out the highest peak in the hand image because that is necessary. To find the highest peak we calculate the distance between all tip of the fingers that is (detected peaks) and centroid by using Euclidean distance formula that is mentioned below

$$E.D\,(a, b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \qquad \text{(iii)}$$

In the formula (iii) 'a' represents all the boundary points and 'b' represents all the reference points that is taken as centroid. And on the basis of this distance formula we can easily find the length of each raised or folded finger taking centroid as a reference point, and this is done in order to extract the exact number of finger raised in the image. There are chances that there might be some peaks detected which actually do not represent the tip of the raised fingers. These peaks are considered as insignificant peaks as shown in figure-5 below.

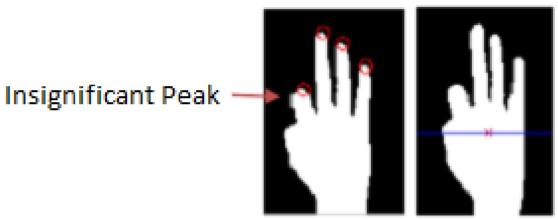


Figure-5: Detected peaks and Centroid of segmented hand

We can also get rid of these kinds of insignificant peaks by only computing the maximum peak. That is by putting the threshold at 75% of the maximum peak value, we can also choose only those significant peaks whose values are more than this threshold value. As these peaks are represented as the raised finger in hand gesture. The Other peaks that are detected but do not intersect or fall above this threshold line would be treated as insignificant peak or folded fingers it may reduce the loss in result and avoid failure and saves the time as well.

3.4 Classification of bits Generation:

The classifications of various hand gestures is based on the features calculated in part II.

The five bit binary sequence is thus generated very uniquely and carefully to recognize and utilize. These recognized hand gesture for supporting human computer interaction what we were trying to do. Peak & Centroid plots are shown in Figure below.

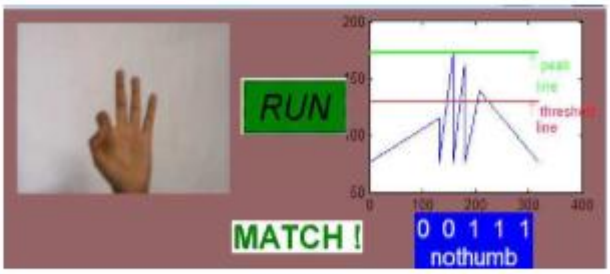


Figure-6: Hand gesture with bits code and distance plot

In the figure-6 it is clearly showing that the significant peaks we identified in previous step is encoded as '1' and insignificant peaks is encoded as '0'.

Based on the intersection status of various finger tips to threshold line.

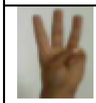
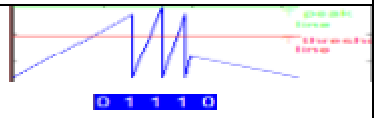
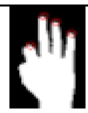
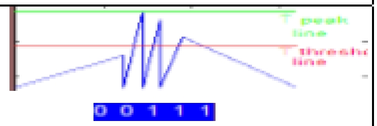
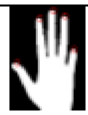
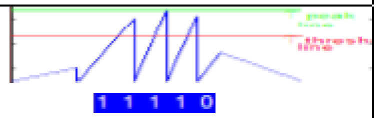
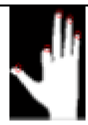
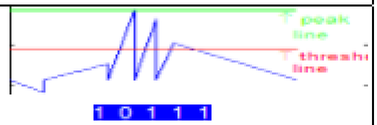
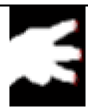
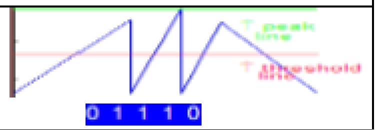
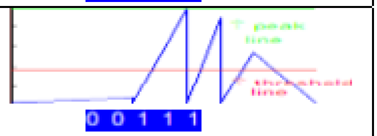
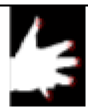
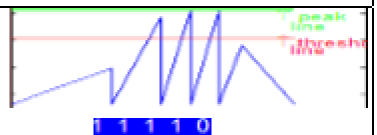
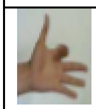
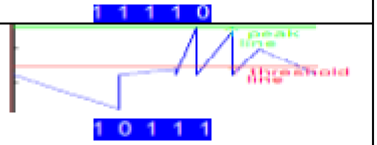
You can see the leftmost bit in the 5 bit binary sequence is reserved for status of thumb in hand image. If thumb is present, leftmost bit will be 1 otherwise 0 as you can see.

4. EXPERIMENTAL RESULTS:

We have tried and applied the above discussed algorithm and with this algorithm we have tested 450 Imaged with 45 different patterns. We have used these effective shapes based features and have encoded bit sequence and now we can recognize and classify 45 different hand gesture patterns. Here On the basis of generated binary bit sequences we can also assign allot of different-different task which can support human computer interaction. The Table-1 provide us with the experimental data result which shows some of the input gestures along with their peaks-centroid plot and resulted corresponding bits. Under this same category of orientation with vertical or horizontal these generated bit sequence will always be unique. However it may be possible that two hand images which are having same hand shape patterns but it belong to different categories of orientation here. Which have generate the same coded bits.

This type of similarity is solved internally in the code on the basis of their orientation category so that these two same encoded bits which belong to different class of orientation can be assigned to perform different key press events. For example generated bit sequence of gesture 'j' and gesture '3' are same, but they are very different in their orientation category. As these hand gestures belongs to two different orientations, vertical and horizontal respectively. And Table-2 shows the result of 450 images tested through this algorithm. Out of which 450 images tested here it has correctly identified only 423 images and falsely identified the remaining 27 Images, And gives the success rate of 94% approximately with average computation time of 2seconds for recognizing single image in the image sequence. The algorithm is based on simple shape based feature calculation which provides us with the comfort of implementation.

Table -1
Experimental Data Results

Gesture	Peaks	Coded bits
		 0 1 1 1 0
		 0 0 1 1 1
		 1 1 1 1 0
		 1 0 1 1 1
		 0 1 1 1 0
		 0 0 1 1 1
		 1 1 1 1 0
		 1 0 1 1 1

In Table-1, all gestures and their corresponding finger tips (peaks) are plotted using centroid as a reference point for hand gesture recognition as discussed in the above algorithm. After finding this If we divide the hand into two regions using centroid, then it will be partitioned the hand region into the finger region and non-finger region, so for detecting the number of fingers in hand image we will use This peak-centroid plot considering the finger region as a significant portion of hand gesture. From other half of the hand which is non finger region, we take only thing which matters is the presence or absence of thumb in the hand and for the detection of thumb we have applied different approach on the basis of assumption. So we have only considered the finger region of the hand to plot the graph for detected peaks. Green line in the graph is plotted to show the highest fingertip or maximum distance of peak from the centroid and red line represents the threshold line which is plotted at the 75% of the maximum peak value or distance. This threshold line plays a vital role in classifying the detected peaks into significant peaks and insignificant peaks. Which is used to denote the raised finger and folded finger respectively.

These detected peaks are then encoded into binary bit sequence of 0 and 1 accordingly.

Figure-7: Sample hand getures

Figure 6 shows all the hand gesture with their corresponding key press events.

Moving forward the below table (Table2) shows the Hand Gesture Recognition Results

Table -2:
Hand Gesture Recognition Result

Gesture	Input Image	Successful Cases	Recognition Rate	Elapsed Time
1	10	10	100	1.72s
2	10	10	100	1.56s
3	10	10	100	1.74s
4	10	09	90	2.90s
5	10	10	100	1.57s
6	10	10	100	1.41s
7	10	10	100	2.45s
8	10	10	100	1.43s
9	10	10	100	1.65s
10	10	09	90	1.76s
11	10	10	100	2.56s
12	10	10	100	1.84s
13	10	10	100	2.41s
14	10	10	100	2.46s
15	10	09	90	1.50s
16	10	09	90	1.87s
17	10	10	100	1.72s
18	10	09	90	1.45s
19	10	10	100	2.74s
20	10	10	100	1.67s
21	10	09	90	1.53s
22	10	09	90	2.64s
23	10	10	100	1.82s
24	10	10	100	1.54s
25	10	10	100	1.70s
26	10	10	100	1.43s
27	10	10	100	1.63s
28	10	09	90	1.66s
29	10	10	100	1.79s
30	10	09	90	1.51s
31	10	10	100	2.46s
32	10	10	100	2.52s
33	10	10	100	2.64s
34	10	09	90	2.52s
35	10	10	100	2.54s
36	10	09	90	2.61s
37	10	09	90	1.45s
38	10	10	100	1.65s
39	10	10	100	1.98s
40	10	09	90	1.23s
41	10	09	90	1.45s
42	10	09	90	1.43s
43	10	10	100	1.62s
44	10	09	90	2.71s
45	10	09	90	1.54s
All	**450**	**423**	**94.0%**	**1.96s**

5. HUMAN COMPUTER INTERACTION:

The section here introduces the application to the hand gesture recognition system. In our system, for producing the interaction between gesture recognition software and MS Office/notepad we have used the Abstract Windows Toolkit. It is used to control the mouse and keyboard remotely. For providing the human computer interaction through gesture recognition system we have generated some of the key press events as shown in Table-3 Below. The sample hand gestures get converted into corresponding key press events which we have assigned.

Table -3:
Comparison table for gestures and their corresponding key press events

(a)

(b)

(c)

(d)

6. CONCLUSION:

We Have proposed this system of shape based approach for hand gesture recognition with several steps including smudges elimination orientation detection, thumb detection, finger counts etc. Visually Impaired people can make use of hand gestures for writing text on electronic document like MS Office, notepad etc. The strength of this approach includes its simplicity, ease of implementation, and it does not required any significant amount of training or post processing. It provide us with the higher recognition rate with minimum computation time. The weakness of this method is that we define certain parameters and threshold values experimentally since it does not follow any systematic approach for gesture recognition, and maximum parameters taken in this approach are based on assumption made after testing number of images. The success rate is 94%. The computation time is just fraction of seconds. The proposed algorithm is simple, user friendly and independent of user characteristics.

REFERENCES:

1) Umang Patel & Aarti G. Ambekar "Moment based sign language recognition for Indian Languages "2017 Third International Conference on Computing, Communication, Control and Automation (ICCUBEA)
2) Lih-Jen kau, Wan-Lin Su, Pei-Ju Yu, Sin-Jhan Wei "A Realtime Portable Sign Language Translation System" Department of Electronic Engineering, National Taipei University of Technology No.1, Sec. 3, Chung-Hsiao E. Rd., Taipei 10608, Taiwan, R.O.C.
3) Obtaining hand gesture parameters using Image Processing by Alisha Pradhan and B.B.V.L. Deepak ,2015 International Conference on Smart Technology and Management (ICSTM).
4) Vision based sign language translation device (conference paper: February 2013 by Yellapu Madhuri and Anburajan Mariamichael).
5) https://doi.org/10.1109/coginf.2006.365596
6) https://en.wikipedia.org/wiki/Sign_language