Translation of Sign Language Finger-Spelling to Text using Image Processing

Krishna Modi
B. Tech. (comp.)
Mukesh Patel School of Technology and Management Engineering
JVPD Scheme Bhaktivedanta Marg, Vile Parle (W), Mumbai-400 056

Amrita More
B. Tech. (comp.)
Mukesh Patel School of Technology and Management Engineering
JVPD Scheme Bhaktivedanta Marg, Vile Parle (W), Mumbai-400 056

ABSTRACT

It is difficult for most of us to imagine, but many who are deaf-mute rely on sign language as their primary means of communication. They, in essence, hear and talk through their hands. Sign languages are visual languages. They are natural languages which are used by many deaf-mute people all over the world. In sign language the hands convey most of the information. Hence, vision-based automatic sign language recognition systems have to extract relevant hand features from real life image sequences to allow correct and stable gesture classification. In our proposed system, we intend to recognize some very basic elements of sign language and to translate them to text. Firstly, the video shall be captured frame-by-frame, the captured video will be processed and the appropriate image will be extracted, this retrieved image will be further processed using BLOB analysis and will be sent to the statistical database; here the captured image shall compared with the one saved in the database and the matched image will be used to determine the performed alphabet sign in the language. Here, we will be implementing only American Sign Language Finger-spellings, and we will construct words and sentences with them.

General Terms

Sign language translation, Gesture recognition system.

Keywords

Sign language translation, Gesture recognition system, American Sign Language, finger-spellings.

1. INTRODUCTION

As known to many, Sign languages are the only medium through which most of the educated deaf-mutes communicate today. There are different sign languages according to countries. Most popular and widely used amongst them is the American Sign Language.

Technology has always been the best and fastest medium for communication in recent years. There has already been a great deal of work done both in the US and abroad in the area of text-to-sign language conversion. The area of sign language to text (or audio) is less mature, although there have been some recent breakthroughs incorporating data gloves for positional extraction. We are attempting to bridge cultural barriers, with technology as a medium. We intend to make technology available also for the deaf-mutes to enable better communication for them with people not knowing sign language.

Our aim is to devise a method that can enable translating Sign language finger-spellings to text in English and enable forming words and sentences using alphabets. We intend to put forth a better approach for sign language translation which would be a simpler approach and enable easy implementation.

Therefore, we present a new approach for recognition of finger-spellings using just a normal computer webcam and our proposed method. We have divided the complete approach into simple steps. Before looking at the proposed approach, we will first look at the related works done in this field.

2. RELATED WORKS

Before trying to find a better way for implementing Sign Language finger-spelling translation, we have to study the existing systems. We found few papers explaining different approaches to tackle this problem. First of them, titled 'Hand Gesture Recognition for Human-Machine Interaction' [1], achieves a 90% recognition average rate implementing recognition of 26 different hand postures. It uses a complex method to recognize the hand postures. This mechanism is displayed in the Figure 1 below.

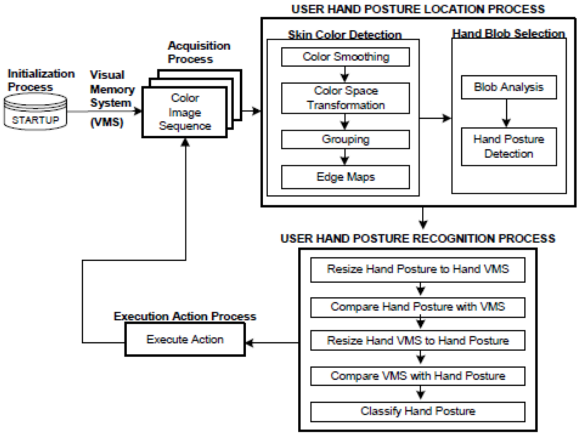

USER HAND POSTURE LOCATION PROCESS
Skin Color Detection
Hand Blob Selection
Color Smoothing
Color Space Transformation
Grouping
Edge Maps
Blob Analysis
Hand Posture Detection
Acquisition Process
Initialization Process
STARTUP
Visual Memory System (VMS)
Color Image Sequence
USER HAND POSTURE RECOGNITION PROCESS
Resize Hand Posture to Hand VMS
Compare Hand Posture with VMS
Resize Hand VMS to Hand Posture
Compare VMS with Hand Posture
Classify Hand Posture
Execution Action Process
Execute Action

Fig 1: Overview of implementation in paper Hand Gesture Recognition for Human-Machine Interaction [1]

It uses the color image and performs several transformations in its color space to detect the skin color pixels. Then, it detects the BLOBs from them and selects the appropriate BLOB containing the hand posture. Then there is a series of steps being executed to process the hand posture BLOB and classify it further. This paper uses Hausdorff distance algorithm for comparing the processed input image with the database and calculating the distance between them.

The second approach that we studied was from the paper titled 'An Image Processing Technique for the Translation of ASL

Finger-Spelling to Digital Audio or Text' [2]. This paper used a different approach with Adaptive Statistical Database which evolves and adapts the changes as the input comes along. The Figure 2 shows the implementation procedure of this paper.

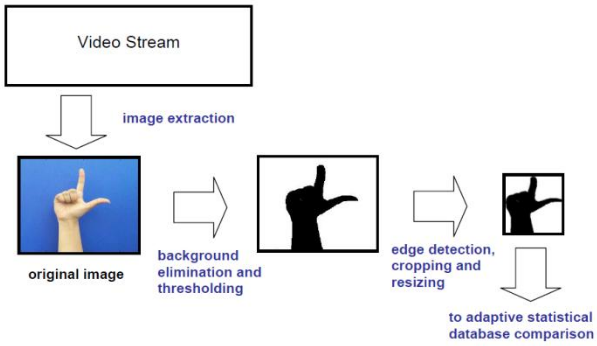

Video Stream
image extraction
original image
background elimination and thresholding
edge detection, cropping and resizing
to adaptive statistical database comparison

Fig 2: Overview of implementation in paper 'An Image Processing Technique for the Translation of ASL Finger-Spelling to Digital Audio or Text' [2]

It extracts the image from the video stream, eliminates its background using RGB filtering and thresholding. Then, it performs edge detection on the binary image and crops the hand posture part from it. Then, it resizes the image to a standard size and then uses error matrix to calculate error among all the images in the database and the processed captured image. Our proposed method is inspired from this paper, but it uses a more step-wise mannered and simpler approach which makes the implementation even simpler and easier.

The third paper which we studied used a special range camera to detect the hand gestures. Though we do not intend to use any such special camera device with our approach, we studied this paper to understand various other ways of implementing hand gesture detection for sign language translation. This paper, titled 'Real time Hand Gesture Recognition using a Range Camera' [3], used highly complex mechanism with many constraints and limitations that we do not intend to observe in our implementation method. So, we have not considered any part of this paper suitable for our study.

With basic idea about various methods of implementing hand-posture detection, we proceed further to our proposed method of implementing hand-posture detection to detect sign language finger-spellings and translate them successfully with best results.

3. PROPOSED METHOD

3.1 Overview

To proceed with our proposed method of implementation, we need to follow few basic steps. The Figure 3 shows overview of the steps to be carried out for the implementation of our proposed technique.

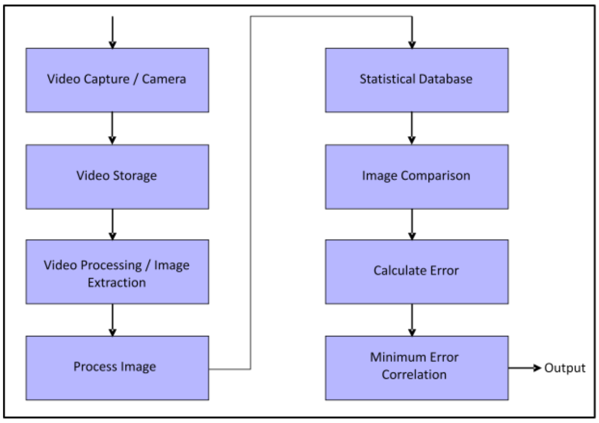


Fig 3: Overview of steps for implementing proposed method

The implementation starts with capturing video using an internal or external webcam. We capture the video and extract the current image frame from that video after every 4 seconds. This 4 second delay is the time frame for the user to be able to change the finger-spelling gesture. This extracted image is to be processed to extract the feature content from it. This processed image is then compared with the statistical database of all the finger-spellings. The processed image is then compared to every image in the database and the error between them is calculated. The image with minimum error or let us say the image that matches the input image the most is considered. The corresponding alphabet to the matched image is considered and is displayed to the user for the finger-spelling performed. The main part in these steps is processing the image. Before seeing the processing to be done on the extracted image, let us have a look on all the constraints and limitations we have for this method.

3.2 Constraints

There are certain constraints to abide by which we have considered in order to ensure efficient results.

The background of every image that is captured should be of a distinctively different color (preferably black) which can be easily differentiated

The finger-spelling should not contain any kind of movement. Hence, we do not support finger-spellings of alphabets J and Z as they contain movement of hand in their finger-spellings

The image should consist of only the hand without any accessories or any other body part being visible

With the knowledge about these constraints, we can proceed further to the section of processing the input image for feature extraction.

3.3 Processing the Input Image

The input image needs to be processed to extract the required features from it which can be compared with the database to obtain results. The Figure 4 shows an overview of the steps to be performed in processing the input image.

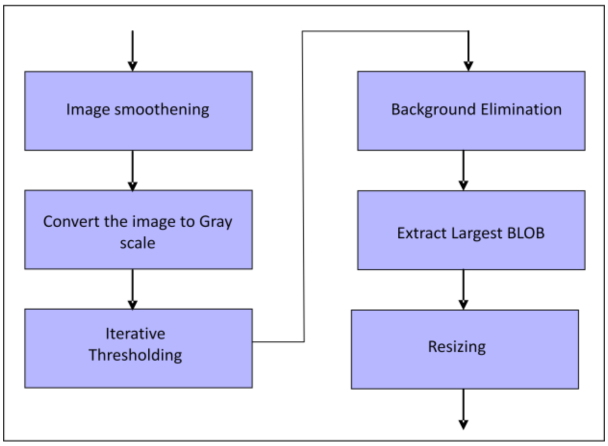


Fig 4: Overview of steps in processing input image

Firstly, according to the proposed method the image need to be smoothened by applying the mean filter over the complete image. This will blur the image and balance the level of colors to certain extent in the image. Then convert this RGB image to grayscale. To do this, for every pixel we multiply the R component to 0.2125, G component to 0.7154 and B component to 0.0721, and then replace the corresponding pixel in the grayscale image with the sum of these products. This has to be done for every pixel in the input image.

Now, this image needs to be converted to binary format. We use iterative threshold [4] for converting it to binary. We need to eliminate the background of the image and consider just the hand gesture. For this, we use the threshold value as 40 which will eliminate the black background and give us the finger-spelling. The pixel values above 40 will be white and the rest will be black. We are concerned about the white area in this image. We have successfully eliminated the background.

Now, we have to extract the largest BLOB from this binary image. A BLOB is any closed object in an image. We suppose that the largest BLOB in the image is the finger-spelling we aim to obtain from the captured image.

After acquiring the largest BLOB from the image, we need to resize the image to a standard dimension. We have used the standard dimension as 300x300 pixels. The images in the database are also of the same standard dimension. We use bilinear interpolation algorithm for resizing the image as it gives best results with least distortion due to non-integral zoom factor. The images stored in the database undergo this same processing.

Note that, the database contains images of sign language finger-spelling by right hand. So it can directly be used for comparing with the database but the left handed images need to be mirrored so that they become compatible for comparing with the database. This makes it easy to make our proposed system applicable for right-handed as well as left-handed people.

This processed image contains the feature that we intend to extract from the supplied input image. After the processing, we compare this processed image with the images in the database and try to get the most matching image with its corresponding alphabet.

3.4 Image Comparison and Error Calculation

There are three images saved for every finger-spelling in the database. The average value of the three samples is used to match with every finger-spelling.

We use a very simple method to compare the processed image with the database images. We compare these images pixel by pixel. Every pixel of the processed image is compared with the database image. As the images are binary images, if the pixel value is same, then the pixels are matching, or else they are assumed to be opposite. We calculate the total number of pixels matching in the two images. We finally calculate the total number of pixels matching and divide it by the total number of pixels in the image and multiple it by 100 to get the percentage of pixel matching in the two images. The equation is given below.

$$Match\ \% = \frac{Total\ number\ of\ pixel\ matching}{Total\ number\ of\ pixels\ in\ the\ image} * 100$$

As every finger-spelling has three samples in the database, we take the average of the three sample's comparison values. We compare the processed input image with all finger-spelling samples in the database, and then we try to get the maximum of all these values. The database image with maximum percentage of matching pixel is considered. If this maximum percentage pixel matching value is above 75%, we consider the alphabet corresponding to that image as the resultant alphabet for the gesture performed by the user.

4. EXPERIMENTAL RESULTS

To implement the proposed method, we have used the Aforge.NET library for implementing the image processing tasks [5]. We have implemented the paper in Visual Studio 2010 using C#. On implementation, we have studied the feasibility of the proposed method and have obtained the statistical results. The Figure 5 is the collection of the finger-spellings with their corresponding English translated alphabets.

The Figure 6 is a collection of the images stored in the database. The database contains three different samples of each of these finger-spellings. The feature extracted from the captured image looks as shown in Figure 6. These images are then compared to the database image.

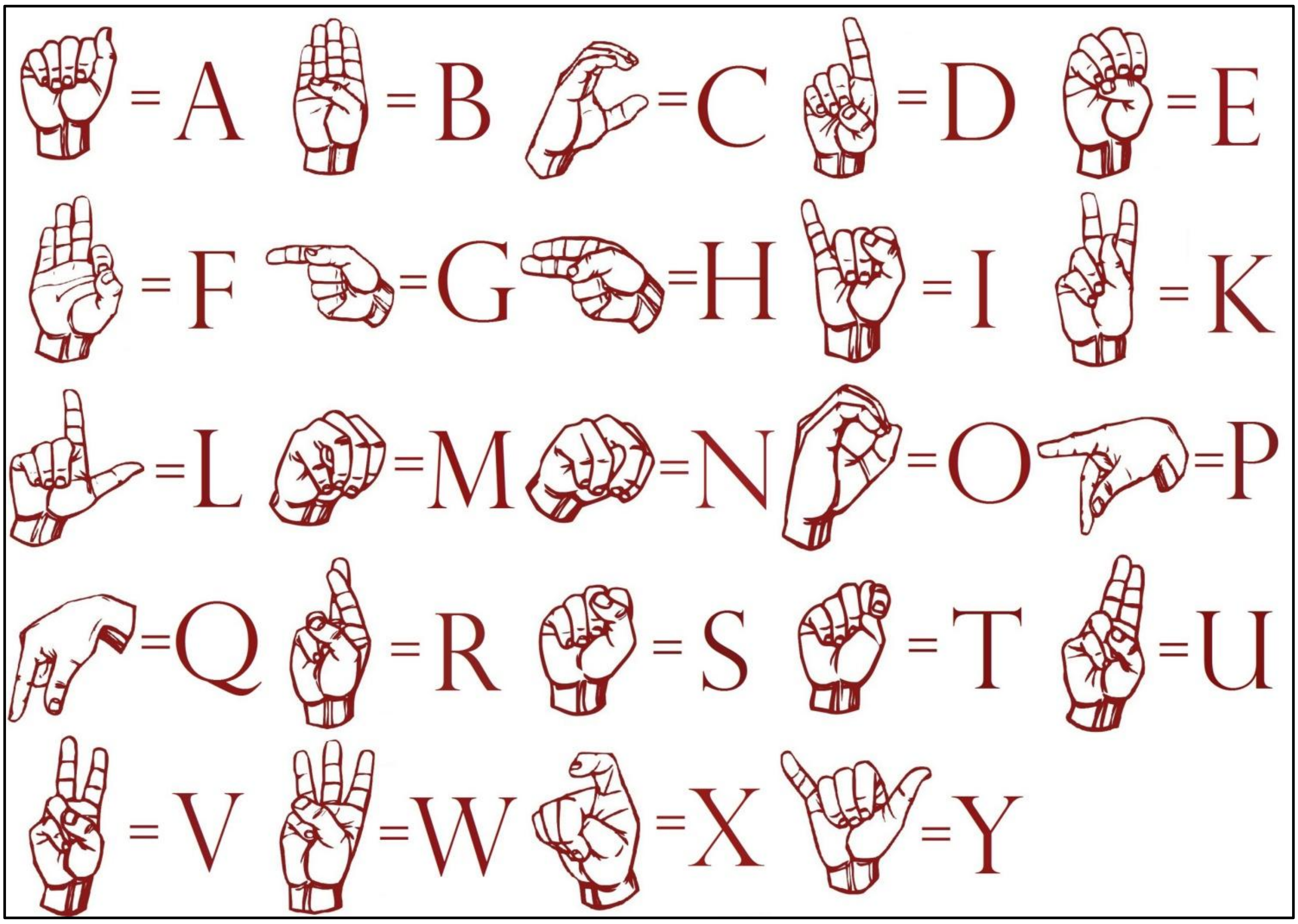


Fig 5: Collection of the American Sign Language Finger-spellings with their corresponding alphabets

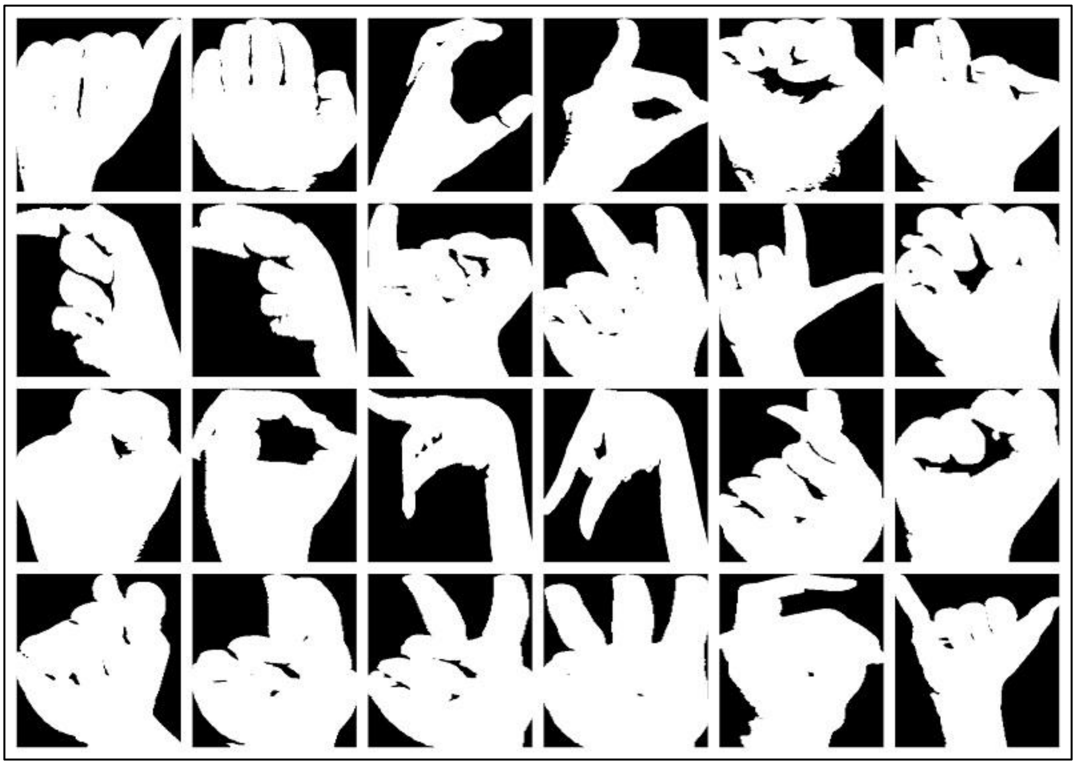
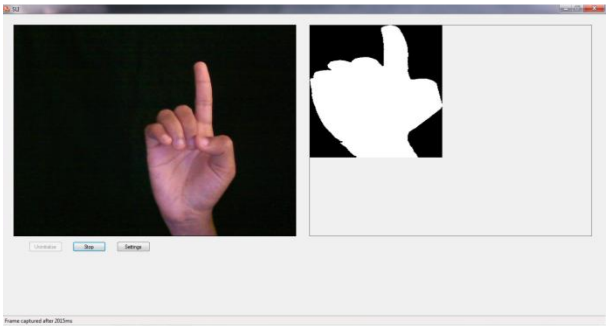


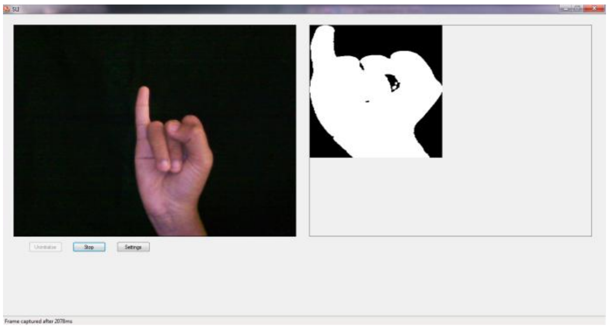
Fig 6: Collection of the images stored in the database

We have developed a project in Visual studio using C# and have implemented the proposed approach to study its feasibility, output results and other statistics to measure its performance. We have taken a black background and performed gestures over it. The Figures 7 and 8 are screenshots showing the processed image after capturing the finger-spelling gesture.



Frame captured after 2015ms

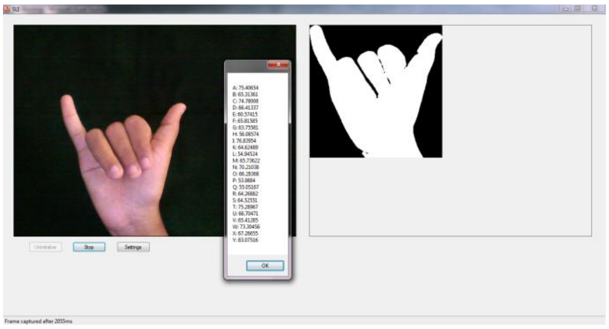
Fig 7: Processed image output for finger-spelling of 'F'



Frame captured after 2078ms

Fig 8: Processed image output for finger-spelling of 'I'

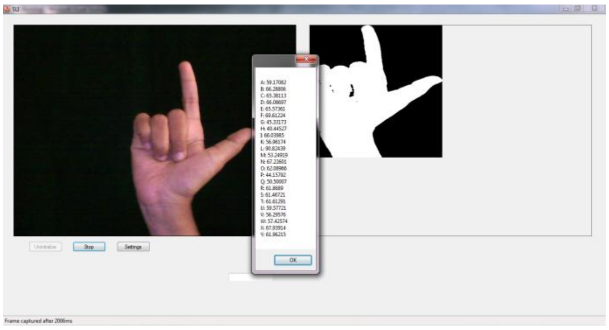
The Figures 9, 10 and 11 are the screenshots showing the results on comparison of the captured image with the database. In case of Figure 9, the finger-spelling performed is of alphabet 'Y'. The results show that the processed captured image matches the database image for finger-spelling 'Y' by 83.075% which is the maximum as compared to other database images. Hence that image can be considered to be perfectly matching and 'Y' as the matched corresponding alphabet. Similarly, Figure 10 shows results for alphabet L and Figure 11 shows results for alphabet 'B' on comparison with database images.



```
A: 75.40654
B: 65.31361
C: 74.78008
D: 66.41337
E: 60.57415
F: 65.81585
G: 63.75581
H: 56.08574
I: 76.83954
K: 64.62489
L: 54.94524
M: 65.73622
N: 70.21038
O: 66.28368
P: 53.9884
Q: 55.05167
R: 64.26882
S: 64.52551
T: 75.28967
U: 66.70471
V: 65.41285
W: 73.30456
X: 67.28655
Y: 83.07516
```

Frame captured after 2055ms

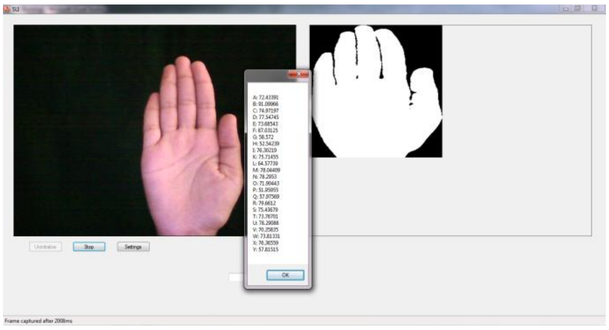
Fig 9: Comparison results for finger-spelling of 'Y'



```
A: 59.17082
B: 66.28806
C: 65.38113
D: 66.08697
E: 65.57361
F: 69.61224
G: 45.33173
H: 40.44527
I: 66.03985
K: 56.98174
L: 90.82439
M: 53.24919
N: 67.22601
O: 62.08966
P: 44.15792
Q: 50.50007
R: 61.8689
S: 61.46721
T: 61.61291
U: 59.57721
V: 56.29576
W: 57.42574
X: 67.93914
Y: 61.96215
```

Frame captured after 2006ms

Fig 10: Comparison results for finger-spelling of 'L'

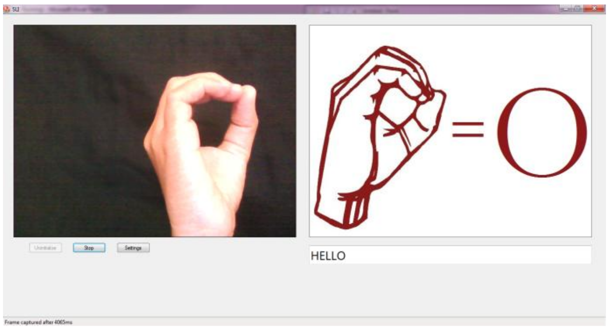


```
A: 72.43391
B: 91.09966
C: 74.97197
D: 77.54745
E: 73.68543
F: 87.03125
G: 58.572
H: 52.54239
I: 76.30219
K: 75.71455
L: 64.57739
M: 78.04409
N: 78.2953
O: 71.90443
P: 51.95955
Q: 57.97569
R: 79.6612
S: 75.43679
T: 73.76701
U: 78.29088
V: 70.35835
W: 73.81331
X: 76.36559
Y: 57.81515
```

Frame captured after 2008ms

Fig 11: Comparison results for finger-spelling of 'B'

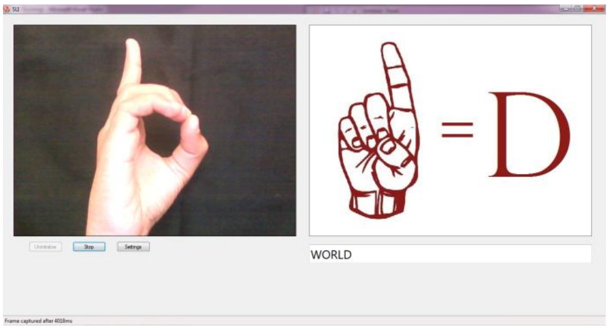
As it can be seen from the screenshots, there is a clear comparison obtained for each finger-spelling with all other database images. We can see that the performed gesture matches to a greater extent with the expected image from the database and gives us desired output. We have tried performing every finger-spelling and have recorded their results. On an average, we obtained that there is 96% accuracy in the results obtained from the project implementing our proposed approach for translation of sign language finger-spellings. As compared to other existing complicated methods, our proposed approach gives better results using a more simplified approach. Hence, due to higher probability of correct output, we can use these finger-spellings to construct words and sentences. The screenshots in Figures 12, 13 and 14 show examples of words we have constructed using finger-spellings without any errors. Figure 12 shows word formation using finger-spellings for 'HELLO'. Figure 13 and 14 show the same for the words 'WORLD' and 'YELLOW' respectively. It can be seen that these words are formed without any error during finger-spelling detection.



HELLO

Frame captured after 4065ms

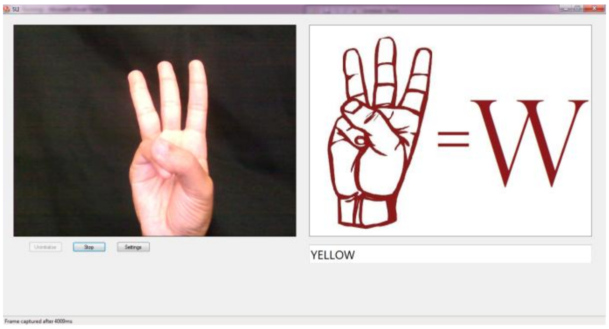
Fig 12: 'HELLO' Word formation using finger-spellings



WORLD

Frame captured after 4018ms

Fig 13: 'WORLD' Word formation using finger-spellings



YELLOW

Frame captured after 4009ms

Fig 14: 'YELLOW' Word formation using finger-spellings

It is observed that it takes at most 50 milliseconds to perform the processing and comparison of the image with the database before giving us the output on an average system. This small delay of 50 milliseconds is negligible and unnoticeable. Hence, we have obtained least possible processing time for obtained the best results using our proposed method.

Further, we can use words to construct sentences. This would help the deaf-mutes to communicate with us in English even if we do not know the Sign Language.

5. CONCLUSION

With the proposed method, we found that the probability of obtaining desired output is around 0.96 which is sufficient and can be enough to make it suitable to be used on a larger scale for the intended purpose. As compared to other existing systems, our approach uses a simpler mechanism to detect the finger-spelling and compare it with the database. It can be easily implemented using the AForge.net library as observed in our experiment.

We can, therefore conclude that our proposed approach for sign language translation is better and give best results with least possible processing overhead. The proposed system does not require the use of any gloves or extra camera to capture complete images. It simply uses the computer webcam with capability of capture sharp images for our purpose.

We expect that more research will be performed in this field so that the proposed system can be extended to support more gestures that make use of motion and maintain its performance results.

There are great number of applications for this system that can help improve human life in different walks of life. It can be easily manipulated to support various uses other than sign language finger-spelling translation.

6. ACKNOWLEDGMENTS

We thank our HOD, Prof. Dhirendra Mishra for cultivating an interest in the field of image processing in us. We are grateful to him for suggesting such a challenging and innovating field for research. We are also thankful to our mentor, Prof. Pravin Shrinath for guiding us though out the research work and helping us understanding the subject. We are grateful to him for helping us in our literature survey and to study the existing techniques. We sincerely thank him for his supervision and constant motivation to help us write this paper in a more versatile manner.

7. REFERENCES

[1] Elena Sánchez-Nielsen, Luis Antón-Canalís and Mario Hernández-Tejera, Journal of WSCG, Vol.12, No.1-3, ISSN 1213-6972. Hand Gesture Recognition for Human-Machine Interaction.

[2] Chance M. Glenn, Divya Mandloi, Kanthi Sarella, and Muhammed Lonon, An Image Processing Technique for the Translation of ASL Finger-Spelling to Digital Audio or Text. The Laboratory for Advanced Communications Technology / CASCI ECTET Department / CAST Rochester Institute of Technology Rochester, New York 14623.

[3] Zhi Li, Ray Jarvis. Real time Hand Gesture Recognition using a Range Camera. Monash University, Wellington Road Clayton, Victoria AUSTRALIA.

[4] Gonzalez/Woods, Digital Image Processing, Ch.10 page:599

[5] http://www.aforge.net/, Used for AForge.net library and framework details.