SIGN LANGUAGE CONVERTER

Project Report submitted in partial fulfillment for the degree

of

Bachelor of Technology

in

Electronics and Communication Engineering

Under the Supervision of

Dr. Neeru Sharma

By

Aashish Kumar - 131017

Rishabh Sharma -131019

Manish Kumar - 131036

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

May-2017

Jaypee University of Information technology, Waknaghat

Solan – 173234, Himachal Pradesh

CERTIFICATE

This is to certify that project report entitled "**SIGN LANGUAGE CONVERTER**", submitted by **Aashish Kumar(131017), Rishabh Sharma(131019), Manish Kumar(131036)** in partial-fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to **Jaypee University of Information Technology**, Waknaghat, Solan(H.P.) has been carried out under my guidance and supervision. This work has not been submitted partially or fully to any University or Institute for the award of the award of completing degree

Date:

Supervisor's Name: Dr. Neeru Sharma

Signature:

Student Signature:
Aashish Kumar (131017)
Rishabh Sharma (131019)
Manish Kumar (131036)

ACKNOWLEDGEMENT

We would like to thanks to many people, who have given valuable time and supported us during the project. Our sincere thanks to **Dr. Neeru Sharma** our project guide, for her exemplary guidance, monitoring and constant encouragement through the course of this project work .

She has taken pains to go through the project and make necessary correction as and whenever needed. We are also grateful to Mr. Kamlesh (ECE Project Lab assistant) for his practical help and guidance. We would like to thanks to all ECE faculty members without whom progress of this project would have been distant reality. We also extent our heartfelt thanks to our family and well-wishers.

Date:

Name of the Students:

Aashish Kumar (131017)

Rishabh Sharma (131019)

Manish Kumar (131036)

CONTENT

LIST OF FIGURES

ABSTRACT

Sign language is a way of expression that is being utilized as the means of communication mainly by deaf and dumb people. This language is based on visual aids and hand gestures that signify letters and words. Sign language converter has always been a challenging subject to work upon which is related to the research as well to the academics. The main aim of SLC is to devise a technique which can identify details of deaf as well as dumbs expression and utilize that information in expressing one's feelings as well as thought process. SLC as an efficient tool plays an important role as a replacement for speech, increasing an individual's ability to put his thoughts into word and comfortability among different group of people. In our project, we have discussed the various steps for input, recognition and analyzation of the hand movements or their position and changing them effectively into written letters and words by the means of comparing the two set of images . First, set is of the images that are given in real time by the means of camera of our laptop .Second, set comprises of the images that are stored in the database of our laptop that have been taken from the same laptop camera earlier in ideal conditions (such as desirable light effects, correct distance between the camera and hand gesture etc.)

In this report we have taken 'Sign Language' from different users and applied different algorithm, after doing so we tried to extract information from them. We matched the input image to our stored database image and able to recognize the message that they conveyed.

CHAPTER 1

INTRODUCTION

1.1 Introduction:

Gestures are movements of hand or other body part, which are able to express something in a communication system and effectively connects between different ends in a communication process. Gestures are based on movements of various body parts that have crucial information. Gesture is used for expressing and conveying messages and connect with the people. Static gestures are crucial because they possess accurate postures. Activities that are associated with gesture movement are different, that are dependent on the positions of different signs in the body, we can consider it to be a hand, an arm, a head or a face expressions. This report is prepared keeping in mind the hand gestures. The hand gestures are sub divided into dynamic gestures and static gestures. The interpretation of gestures is quite a complicated field to work upon, because there are concepts related to many-to-one mappings which can result high complexity and low efficiency in gesture recognition. The main disadvantage with gestures is that they are completely ambiguous in nature as well as incompletely specified. Indian hand gesture recognition is one area which we have actively researched in our project. It gives us flexibility to communicate effectively with others and if we suppose the user is lacking technical knowledge about the system, they will still be able to utilize the system with ease without posing any difficulties. Gestures are used to convey and express the significance of a particular statement told by people. Our gestures normally come naturally with our speech when we are expressing our feelings to others which ease the process of communication. It permits deaf and dumb people to express their thoughts and feelings even without the usage of verbal speech of communication. In, this modern age of development software's for sign language recognition are quite indispensable and are seeking great attention from scholars and researchers. These software establish an understandable level of communication between common masses as well as mute people, it is also helpful for deaf people as it provides them the skills to interact with each other in a professional and quick manner by simply by their hand movements. The indian sign language (ISL) consist 26 different set of hand symbols each one points to an alphabetic letter. This was and has been the language of more than 100,000 deaf and dumb people throughout the India.

Sign Language Converter is receiving popularity as it supports and enable communication with a human and mute people. When the user gives the gesture in real time, to the system it is made in such a manner that it is capable of recognizing and interpreting it effectively and successfully. This is termed as “Gesture Recognition”. The goal of our project is constructing an efficient system that is able to classify any particular hand gesture and extract the feature required to interpret corresponding sign. This entire method of gesture recognition is based on the ‘Indian Sign Language’ characters .

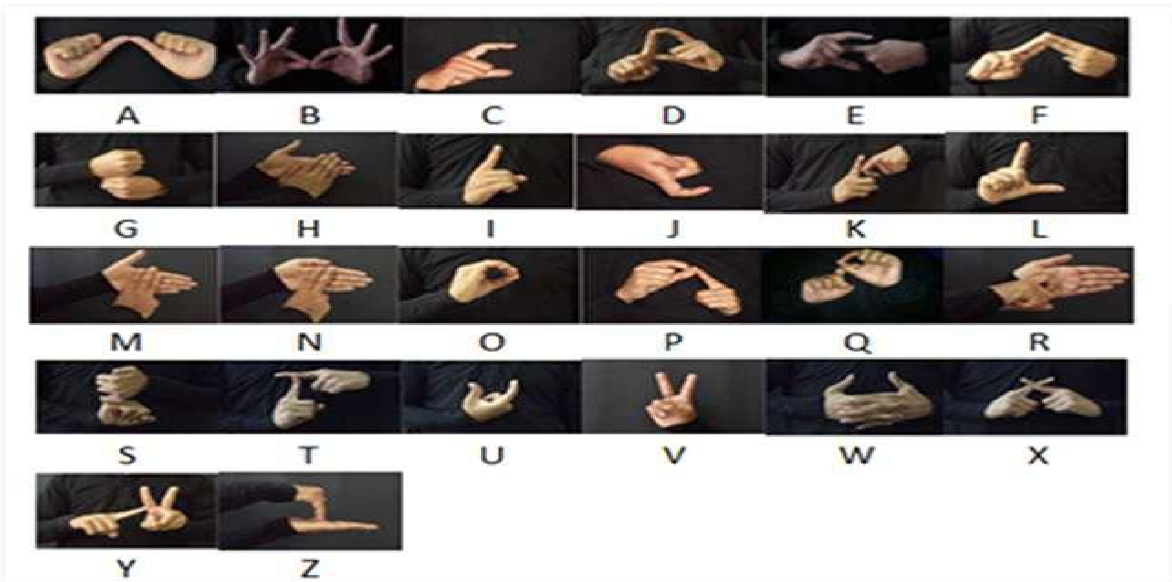

Figure 1.1: Indian sign language

1.2 Related work:

Developing efficient gesture recognition is a tedious task and is done step by step by different people and different community over a past decade, for this purpose they categorized the recognition system mainly in three parts these are shown below:

Glove based analysis [1]: In this, it uses sensors either mechanical or optical or both which are attached to a glove that transduce finger flexions into electrical signals for determining the hand posture. Movement of fingers at different angle offers different resistance that is sensed by a sensor and electrical signal gets generated. These signal further passed through transducers to get the desired output. Vision based [2]: It is based on the way we perceive information about surrounding. In vision based analysis user gives input gesture either using mouth or hand movements but we have done our report on vision based recognition system, although this system has lots of challenging factors that we will discuss later. Analysis of drawing gesture [3]: This is third category of recognition system and is highly used for vehicle tracking or in similar areas. Its analysis can also lead to recognition of written text.

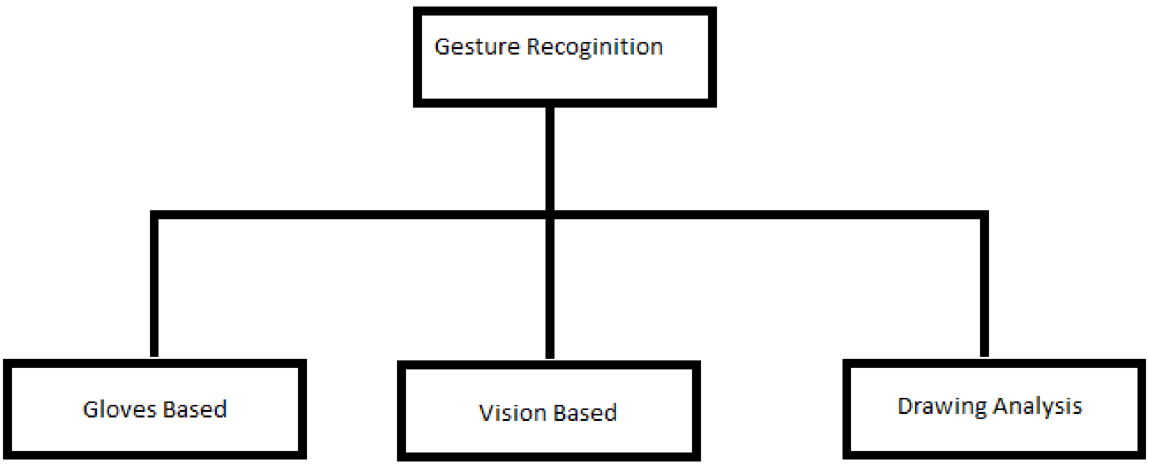

Figure 1.2: Gesture recognition types

Key issues of hand gesture recognition system and their challenges are discussed in [1. Now a day Various recent methods of gestures recognition system are presented and compared based on research results obtained and database used along with their advantages and disadvantages. [2] presents a system using Wavelet transform and neural networks (NN) for recognizing static gestures of alphabets.

Certainly, the use of glove based techniques will simplify the process of extracting the features by offering an exact position and orientation of the hand which in turn will facilitate hand tracking [3], in this report; we considered the vision based techniques which required a natural interaction with the computer without cost to be mentioned [4]. After extracting the features, a suitable recognition modeling technique such as Neural Network (NN), Hidden Markov Models (HMMs) [5] is selected to identify the fingers and recognize the gesture. Recently a lot of surveys are done to demonstrates the details of gesture recognition systems and their applications, these are available in the following references [6-9].

[10] Recognize static posture of American Sign Language using neural networks algorithm. The input image are converted into HSV color model, resized into 80x64 and some image preprocessing operations are applied to segment the hand [10]from a uniform background [10], features are extracted using histogram technique and Hough algorithm. Feed forward Neural Networks with three layers are used for gesture classification. 8 samples are used for each 26 characters in sign language, for each gesture, 5 samples are used for training and 3samples for testing, the system achieved 92.78% recognition rate using MATLAB language[10]. Alex pentland [5] applied scaled normalization for gesture recognition based on brightness factor matching. The input image wich is segmented using thresholding technique where the background is black. Any segmented image is normalized (trimmed), and the center mass [5] of the image are determined, so that the coordinates are shifted to match the centroid of the hand object at the origin of the X and Y axis [5]. Since this method depends on the center mass of the object, the generated images have different sizes [5], for this reason a scaled normalization operation are applied to overcome this problem which maintain image dimensions and the time as well [5], where each block of the four blocks are scaling with a factor that is different from other block's factors. Two methods are used for extraction the features; firstly by using the edge mages, and secondly by using normalized features where only the brightness values of pixels are calculated and other black pixels are neglected to reduce the length of the feature vector [5]

In [11] the system i variation to environment lighting changes which produces erroneous segmentation of the hand region. HMM tools are perfect for recognition dynamic gestures [5] but it is computational consuming.

Other system limitation as listed in [12] where the gestures are made with the right hand only, the arm must be vertical, the palm is facing the camera, and background is plane and uniform. In System limitations restrict the application such as; gestures are made with the right hand only, the arm must be vertical, the palm is facing the camera, background is uniform. In [13] the system could recognize numbers only form 0 to 9. While the system proposed in [16] for controlling a robot, can counts number of active fingers only without regard to which particular fingers are active with a fixed set of commends [16].

In real world people want to use application on the go. So once application is designed, there shouldn't be any difficulty in operating. Keeping in mind of user's requirement research community has found another way to increase accuracy of the system by making the system skin invariant and background invariant, however these adjustment introduces more coding space and time complexity.

1.3 Organization of the report:

The report has been organized as follows:

Chapter 2: In this chapter we have shown our working models, how we have approached toward the project and the basic steps that we have implemented for the recognition system.

Chapter 3: This steps involved in implementing the proposed work are discussed in this chapter. Role of camera, preprocessing of images, database management, implementation of SIFT algorithm and implementation of matching algorithm are explained in detail.

Chapter 4: This chapter is dedicated to results. We took hand gesture as input from different users and processed through our system. If recognition goes beyond to 85% then gesture is interpreted and letter corresponding to that is displayed. Different examples where gestures have been correctly identified have been included here.

Chapter 5: Proposed work is concluded in this chapter. Various problems encountered during implementing the proposed work have been highlighted here. Also how this work can be taken forward to implement more such applications have been discussed.

CHAPTER 2

SYSTEM MODELING

In modern era computers are highly used by every individual extensively. Developing and making sophisticated human computer interface is the one of the main focus area for the scholars and in attempt of that they have developed algorithm, different image processing and feature extraction techniques to make the computer to understand our facial expression, speech, hand gesture etc. Once these gestures are recognized we can implement it accordingly. But, recognition is not as easy as it sounds. This process has lots of hurdles some of them are crossed and some are still milestone for researchers. After getting the input image through the means of camera we are using that is, web-cam of the laptop. We can classify gesture recognition system mainly [4] into three steps which are:

1) Hand Tracking and Segmentation.
2) Feature Extraction.
3) Classification and Recognition.

These steps have been shown in figure: 2.1.

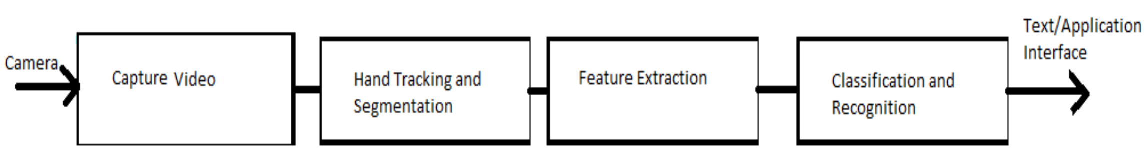

Figure 2.1: Block diagram of recognition System

After capturing the hand gesture, the area of our interest that is the gesture is recognized, from this feature vector has to be structured. The feature vector comprise of the "Indian Sign Language" standardized database which comprise of .JPG extension formats of already present in database together with some real-time image which has been captured by us at the same instant in real time environment conditions. All the defined key points of database image are stored in an array. All the shades of grey whose levels are higher than the zero are assigned as key points and then we generate key points array as shown in Figure 3.2. After key point generation, matching is done. This matching is not made for each pixel value instead of reducing complexity, dimensional reduction is done. [5]Data after matching is recorded as the ultimate feature vector. While deciding descriptor retaining the key points with the ratio of the vector angle to second closest neighbour is greater.

figure 2.2: System's processing steps

The implementation of our hand gesture recognition system provided us with 70% precision in recognizing the input gesture from the database. The recognition overall percentage slowly started to fall to 0%. In letter "M" "N" and "R" as they have almost identical shape (Figure 1.1). The method that we have used is quite complicated when we consider with respect to the overall coding that needs to be done. We took input data either by samples (training method) or in real time by the use of web-cam (testing method). Our database is broad and flexible as it covers almost all combination of hand gesture and different skin colours.

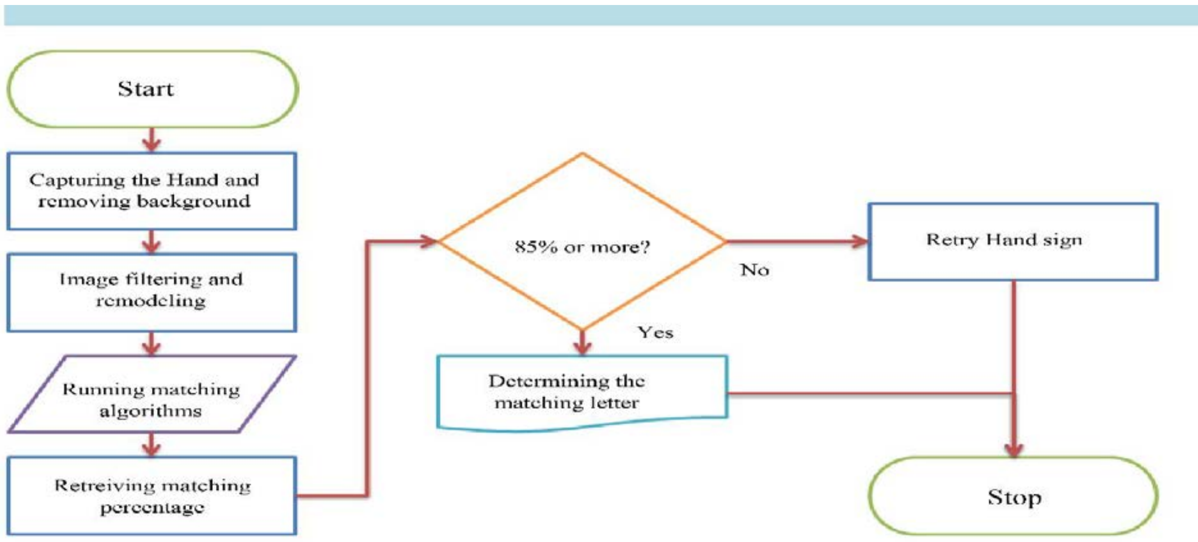

figure 2.3: Flow chart of the system

2.1 MATLAB Overview:

MATLAB stands for *matrix laboratory* and is highly used for complex technical computation. It's typical uses include:

- Mathematical computation
- Algorithm implementation
- Designing, Simulation, and Prototyping
- Data analysis, Exploration, and Visualization
- Engineering graphics
- Developing and deploying application in MATLAB.

MATLAB is an effective tool with essential data element with an array which does not need dimensioning. MATLAB perform processing on data considering them as theyare stored in n-dimensional matrix vector. This permits dealing with many complex data and it's computation. MATLAB can also convert its scripts to different language like C/C++, FORTRAN etc. The reason, why we have decided to use MATLAB for the development of this project is because of its computational efficiency and support to mathematical modeling of different algorithm.

MATLAB consist toolbox like FDA, NNA, GUIDE etc. to reduce coding requirements. It can also auto generate codes if required for further implementation.

Now a days, users of MATLAB have increased manifolds as compared to last decades, due to the reason that it provides effective tool for visualization and conceptualization of different mathematical problems. In 2016, MATLAB had around three million users across industry and academia. These users hell from different backgrounds of economics, applied mathematics, applied science, engineering, and graphic designers.

Graphics and Graphical User Interface Programming:

MATLAB supports different features of graphical user interface. It consists of 'Graphical User Interface Progress Environment' for graphics designing. It also possesses the special feature of graph-plotting. For example, we can use the function plot to make a graph. Following code gives basic ideas of plot function:

```
Clear all;

Close all;

p = 0:0.001:2*pi;

plot (sinc(p))
```

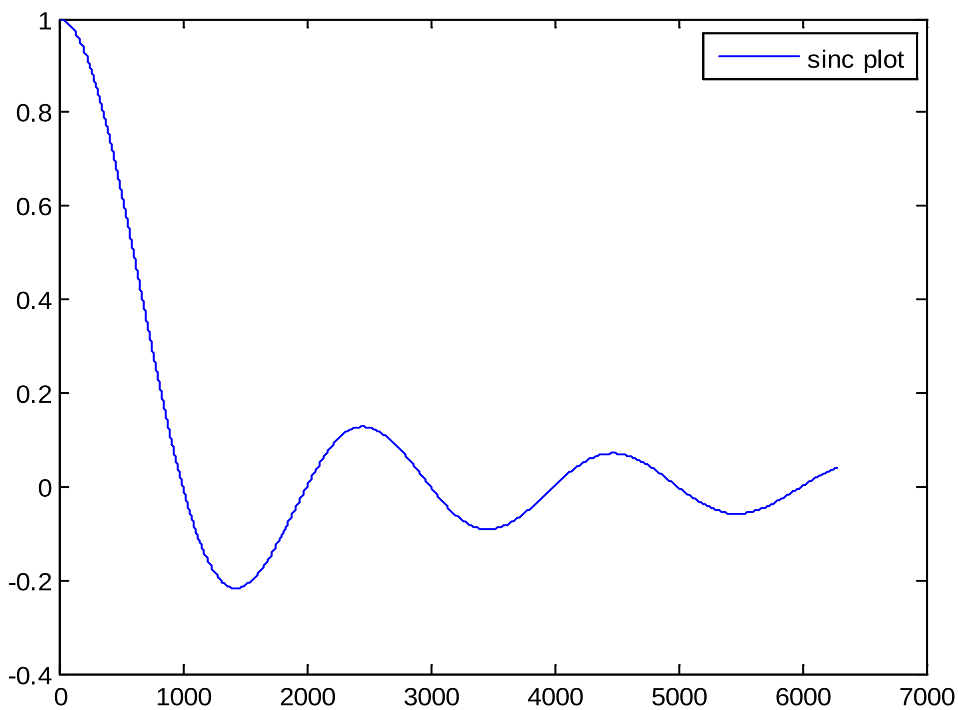

Figure 2.4**:** Sinc function

MATLAB also supports 3-D plots for better realization of curves, contour, and surfaces.

```
[p,q]=meshgrid(-1:0.01:1);
 r=q*exp(-x.^2-y.^2);
 surf(p,q,r);
```

This code produces a surface 3D plot which is plotted below:

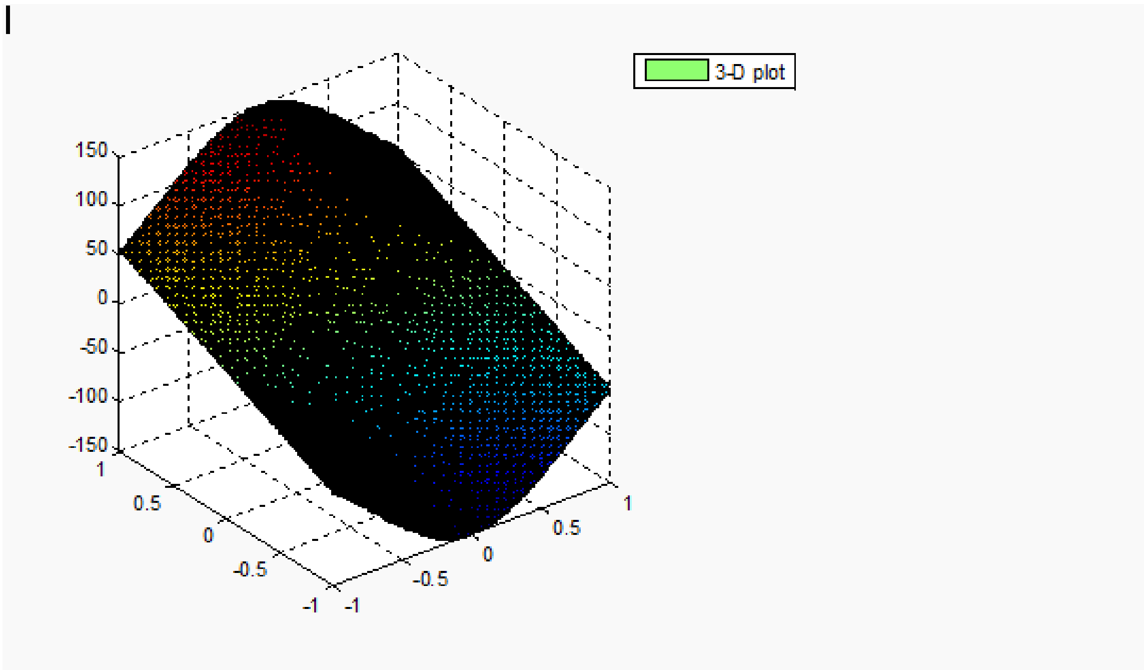

Figure 2.5: Surface 3D plot

In MATLAB, graphical user interfaces can be programmed with the GUI design environment (GUIDE) tool.

Interface with Other Languages:

MATLAB supports various languages such as C/C++, JAVA, FORTAN. Functions and subroutine written in these languages can be called in MATLAB which make it more versatile. Its calling is done by making a wrapper class which allows MATLAB data type to pass through as well as return its value. Executable object files after the compilation are termed and saved with .MEX extension file.

CHAPTER 3

IMPLEMENTATION OF WORK

Development or design of recognition system is complex but it can be done by dividing our proposed work in small groups and subgroups. Once these subparts are done they are pipelined to the system.

3.1 Camera:

Here, by camera we mean images that are taken through the web-cam. It is quite essential to state here that the hardware we are using, mainly the camera, is not a high-quality camera as its resolution is quite low so sometime it may show some degree of error probability .Hence, even after dealing with some of the difficulties minor errors are can't be avoided. These errors are not due to insufficiency of the code but due to lack of professional camera.

3.1.1 Input Image via Camera:

This is the initial and foremost step in Image processing. Here we have used web-cam for capturing the image. Next step is taking input of hand gesture, but be cautious while recording the input into the web-cam. Make sure you are not far from the camera, your hand is straight and motionless while facing the camera in the middle.

Once these steps are done, different algorithm techniques are used for preprocessing. The foremost step is cropping and resizing the image. The next step is filtering noise from the image and is done by getting rid of limited size connected component from images, and identifying region of interest whose pixel value are less than the selected threshold. As, we have low quality webcam which can naturally introduce noise into the image therefore we don't require any specification for any connectivity. We have set the default connectivity to eight through SIFT (Scale Invariant Feature Transform) algorithm. Reducing the level of noise to the extent that it is completely removed noise in the image is most crucial and one of the most complex techniques of pre-handling. Noise reduction can somehow be done using [6] 2-D median filters. This filters works in 3x3 neighbourhood. It we take the median of neighbouring pixel of center point (interest point) and replace it with the median value. This filters works to reduce salt and pepper noise and introduce equal intensity to the centre point. 2-D median filter is also known as smoothing filter.

3.2 Pre-processing:

At this stage, to bring input image in the appropriate from certain modifications are done. For this we have trimmed, cropped, and resized the image. Total mass that is the number of pixels in a given image ,[7]Centroid center of mass, Orientation which is the angle of major axis, Asymmetry, [8]Kurtosis and Fourier transform along with its series are some of the important characteristics for the recognition of the particular image that has been taken into consideration for 2-D images.

3.3 Creating Database:

Creating Image database is another important work in this paper. From, the past decades, database management tool have evolved with the development of other types of database management tools.

Our stored database contains about 50 images showing gesture of the 'Indian Sign Language', all images are stored in the .JPG format. As we know, having a large database points to more efficiency and accuracy in results and hence, it is more prone to increase in the recognition percentage. But it has some limitation, while we work on the software part we must take into account overall size of the program and its complexity, complexity may be time as well as space complexity. So, to make a suitable combination to meet both efficiency and reduce the overall complexity, we have changed our database in accordance with fulfillment of the desirable and efficient results, and thus we have been able to reduce the overall complexity of the code.

3.4 Feature Extraction:

In this step, it is essential to recognize only the area of interest that is hand gesture containing the important information as it is the essential part we call as descriptors. We use descriptors to erase all other unwanted characteristics of the images captured through the camera. [9]Structure stabilization is vital for measuring the distance, which is the distance between the camera and the hand.

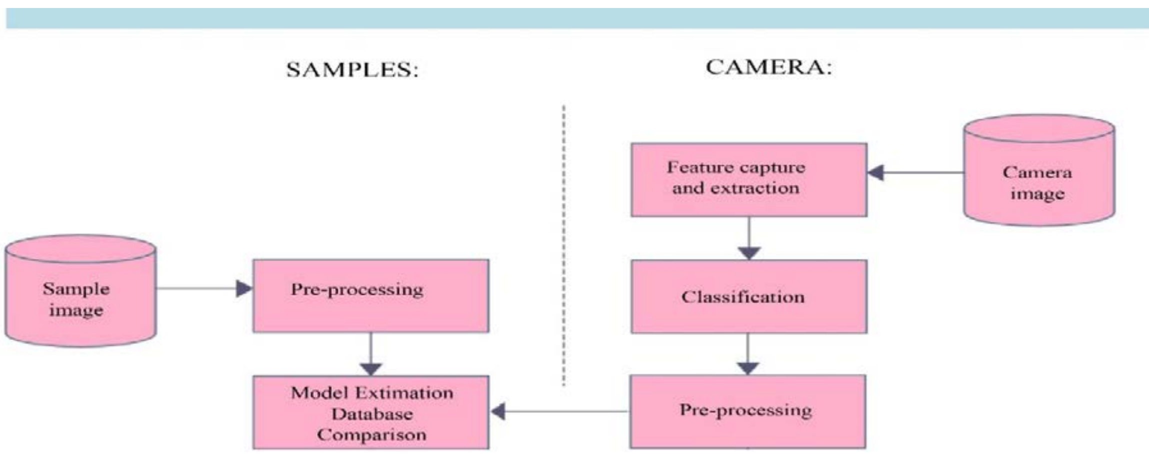

Figure 3.1: Block diagram of proposed work

After doing processing on input image (implementing filter and matching algorithm) we retrieve how much image is matching to our database. If matching goes beyond 85%, matching is accepted else the gesture is supposed to be vague or complicated to interpret (figure-3.1).

3.5 SIFT Algorithm:

The algorithm that we have implemented in our project is LOWE'S SIFT algorithm. SIFT algorithm is considered as an invariance algorithm because of its invariant nature as a result of which it is widely used for real time applications. The scale invariance is the main intention of selecting this algorithm. SIFT was developed by Lowe and is basically a histogram of gradients. This algorithm focuses on key points at each and every possible grey level location.

The acos function that of the key vectors are arranged serially and consequently, the foremost 128 of such value are utilized as the feature vector. Here we take an input image which is compared with all its key points with respect to the database images key vector points, the key points are noted and matched. [10]The image in the database with maximum number of key points matched successfully is subsequently identifies as that alphabet. The algorithm implemented is shown in figure 3.4.

When we consider image, the major factor that comes to play is that of variance especially when the image is made to appear in a large screen. The size of the image in the particular frame may differ in real-time environment. Actually, we have basically [11] five types of invariance which can be observed particularly in images are as follows:-

1.) Rotation invariance.
2.) Affine transformation.
3.) Scale invariance.
4.) Illumination invariance.
5.) Perspective invariance.

The foremost step in developing an efficient hand based gesture recognition system is working mainly on reduction of the scale invariance, illumination invariance as well as invariance due to rotation which we have worked upon. The SIFT algorithm that we have implemented helps in effectively managing such types of invariance. The method of implementation is illustrated under figure 3.4.

Firstly, we will do feature extraction which is done by identifying the key points. The location and scale variance are removed at that instant through edge reduction techniques. Sub pixel reduction is implemented through the means of down-sampling of the interest area under observation. Under edge elimination technique, various edges is taken into account based on the [12] 'SOBEL EDGE DETECTION' method. The essential images are obtained through the image gradients that over a period of time undergo sampling as (16*16 array of locations in scale space) and after that an array of histograms is build. The figure 3.2 shows the image gradients and the key-points obtained for a particular image.

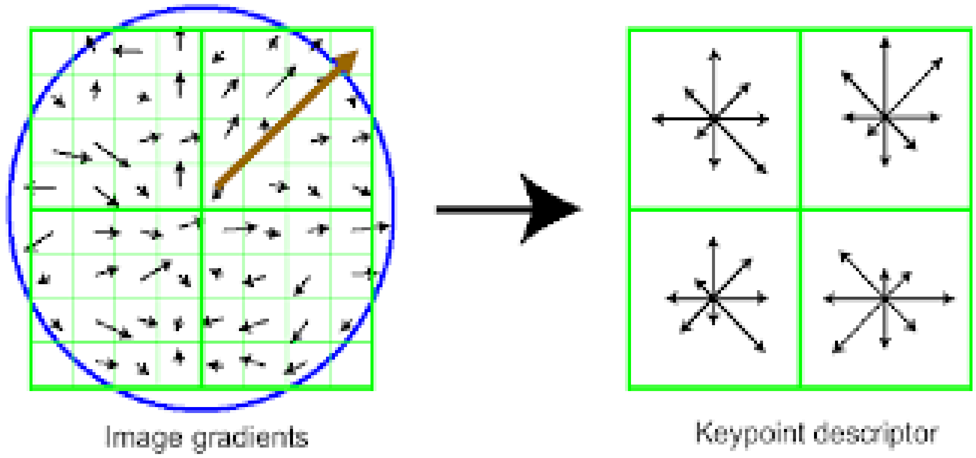

Figure 3.2: Image gradient and key-point descriptor

Difference of Gaussian (DOG method) is implemented to obtain this scale invariance in entire scale space. The figure 3.3 shows the exact working of the DOG method. This scale works based on concept of octave by octave.

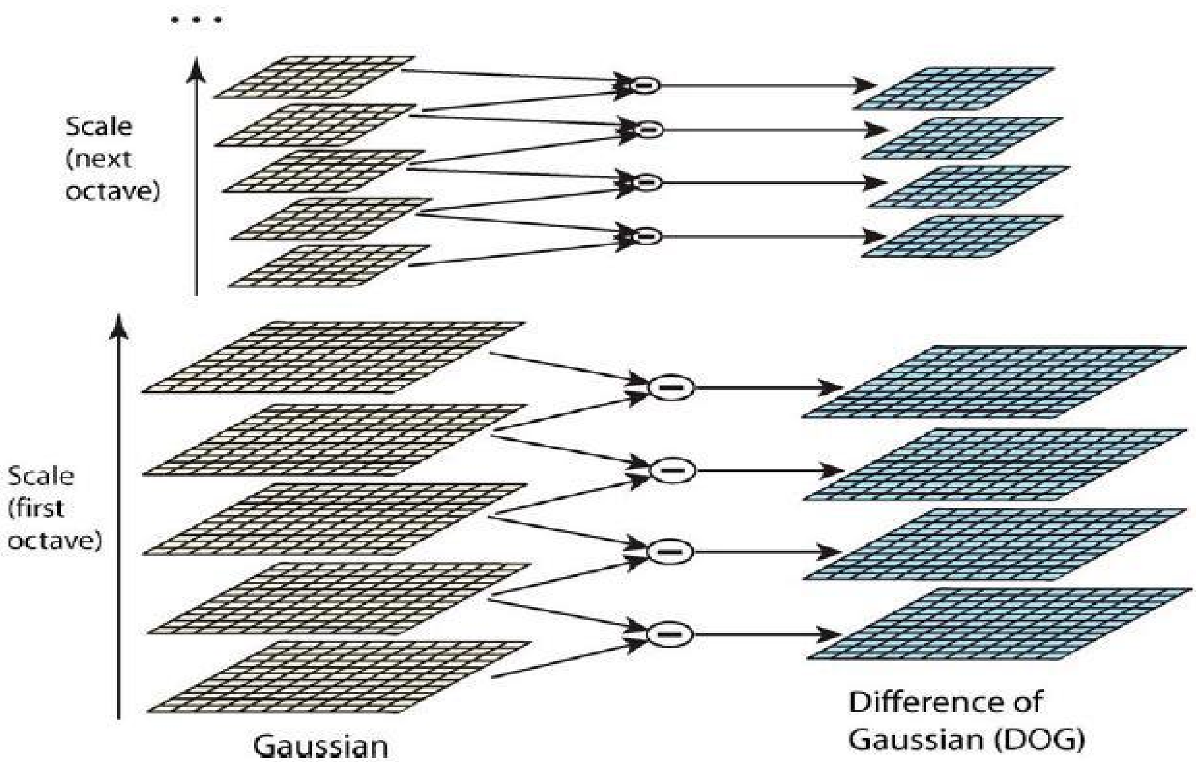

Figure 3.3: Difference of Gaussian

Every scale is taken in term of an octave and the change in between the next octave is calculated as a Gaussian function. Since such type of characteristics are obtained through the calculation of image difference, if such features are continuously present alongside the difference of Gaussians we consider it to be scale invariance and thus it is successfully retained. It provides an important factor for the overall efficiency of our system.

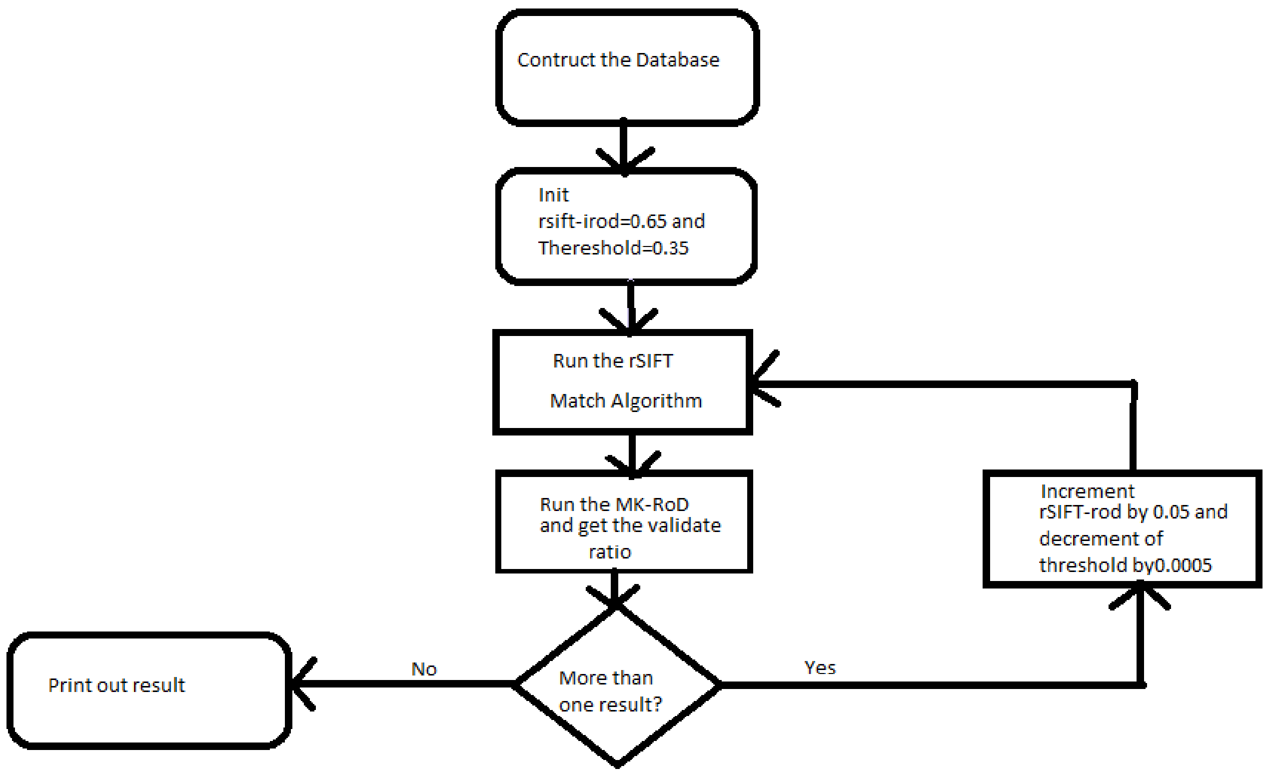

Figure 3.4: Flow chart of SIFT algorithm

3.6 Matching Algorithm:

The main aim is to identify a gesture based intelligent system on the input of information which is basically signal based. Thus, what we need is to focus on the basic features of Sign Language Gestures and look for an efficient classification tool. Each gesture varies in time as well as in space. The, speed can differ significantly between the signing of different images. Even if we consider only a single person perform the same gesture at different times, the overall position and speed of hand gesture can be different. Moving towards building a reliable database and thus getting the final input picture, we still need to make a comparison of the input picture with all pictures that are stored in our database so as to sort out the most closest match to that of the input image. The best possible results are obtained as well as received by means of appropriate matching algorithms.

Matching [13] through three different methods are as follows:

1.) 2-D correlation

2.) Edge detection.

3.) Data Histogram.

The matched descriptors of input image are stored in the database and ultimately the image with maximum matching is finally selected.

3.6.1 2-D Correlation Coefficient Matching:

[10**]**The correlation coefficient is expressed as an integer stating the similarity between two images based on their strength of the pixels. The equation which represents the correlation coefficient is

$$r = \frac{\sum_m \sum_n (Amn - \bar{A})(Bmn - \bar{B})}{\sqrt{\left(\sum_m \sum_n (Amn - \bar{A})^2\right)\left(\sum_m \sum_n (Bmn - \bar{B})^2\right)}}$$

where *A* = mean2(*A*) and *B* = mean2 (2)

A and *B* depicts the pictures we have used for matching, while *m* and n represent the pixel position in the image respectively. The important thing to mention is that the size of the two images must be identical; otherwise the result would be inaccurate. This algorithm calculates for every pixel position in both the images, it also calculates the intensity value and compares it to the mean intensity of the complete image. Ultimately, the closer this coefficient is to the value 1, the closer are the two images under observation. It also breaks down an image into various blocks before calculating the correlation coefficient which reduces the time complexity. Although, if the correlation coefficient is equal for 2 blocks of images, this doesn't necessarily mean that images are identical. When we make use of small sized images in the database we can reduce errors to a small extend. This method shows recognition up to 70% with minor errors in case of quite similar gestures. To further improve the accuracy, we opt for second method of matching algorithm.

3.6.2 Matching via Edge Detection:

Edge detection is a method or tool for finding out the edges of objects in the images. It's working mechanism is based on identifying certain breaks in the illumination of the image. We can use edge detection for image separation and data abstraction which can be linked to image processing. By applying this process we were able to raise matching accuracy from 70% to 80% images pixel by pixel. The main aim of this function is to identify edges intensity within the image, this process is quite an accurate method of application. Though two images may have same number of white pixels, but they are made to differ in terms of edges, and hence are at a certain level of dissimilarity. This method is used for comparison of one image with a large number of images in the database is very accurate. There are several edge recognition algorithms such as Sobel, Canny and Fuzzy logic methods etc. but we have used Sobel edge detector in this project. Selection of a suitable algorithm for your function is important as much as specifying the image's threshold, which is an important parameter in identifying its sensitivity as shown in figure 3.6. We have implemented the Sobel method in the SIFT function. This is a major improvement that has been present in the system. It has increased recognition by 3% - 5%. As the edge detection is certainly affected by image noise, it is important to filter out the noise to avoid false recognition.

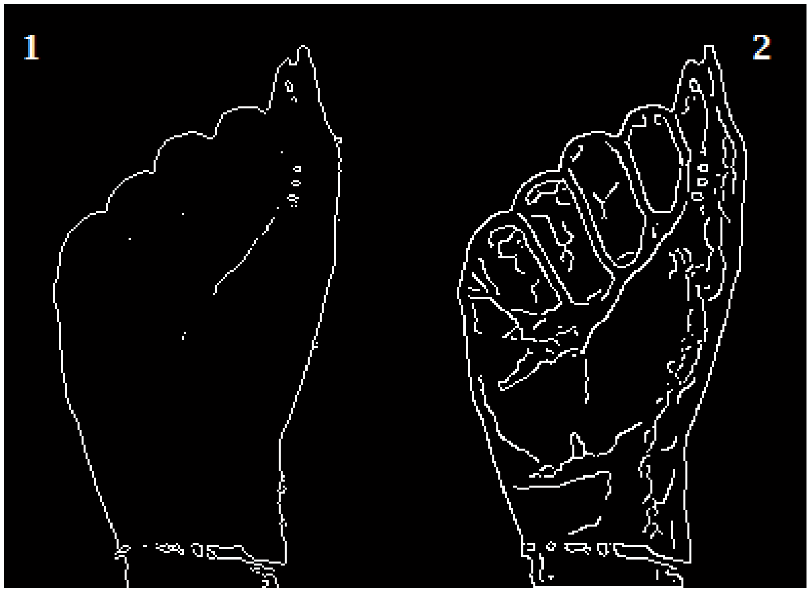

Figure 3.5: Noise Reduction

So we use the Gaussian filter the main advantage of using such filters is such that they remove any noise and rough parts of the image which can lead to false results.

An important step called suppression is applied to narrow the edge and eliminate pixels that are not suitable to be a part of it. Thus, it can contribute to overturn all the gradient values to 0 excluding the local maximal so as to compare the strength of the recent pixel and the adjacent pixel in the both directions. Suppose, if the intensity of the current pixel is the large compared to the other pixels in the same direction the value will be preserved. Otherwise, the value will be replaced. The only disadvantage of executing this step is that there would still be quite some errors due to color disparity. In order to get rid of such situation, it is important to filter out the edge pixel with the minimal gradient value and reserve the edge with the maximal gradient value. Therefore, two threshold values are designated to elucidate pixels, first is upper threshold value and the second is the lower threshold value. The two threshold values are not simple to calculate. They are calculated by the complex method “by trial and error” method. In this unlimited values are applied until the exact one could be determined. This one is quite a complex procedure and increases space as well as time complexity. Thus, we can’t apply every concept for improving the overall efficiency in results as it will in turn result in more complex and difficult systems to implement.

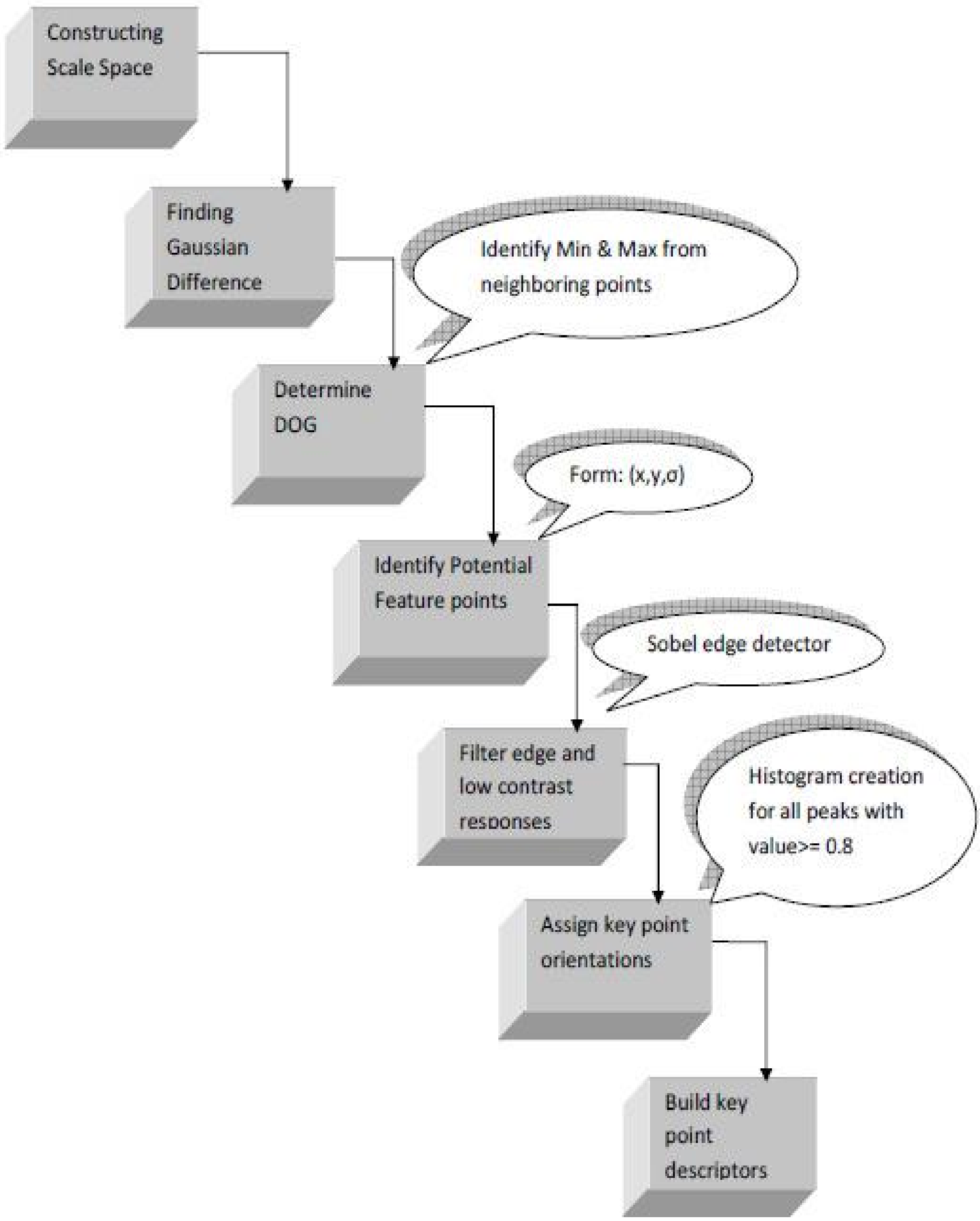

Figure 3.6: Waterfall representation of our System design

CHAPTER 4

RESULT AND CONCLUSION

Result:

The accomplishment of the entire algorithm was achieved in MATLAB. The standardized 'Indian Sign Language' dataset was utilized on the basis of 26 alphabets of English letter with different lighting conditions and different orientation. The implementation gave 70% accuracy in identifying the test sample for our dataset. The images for training known as the sample images that are stored in database are shown in the Figure 4.1.

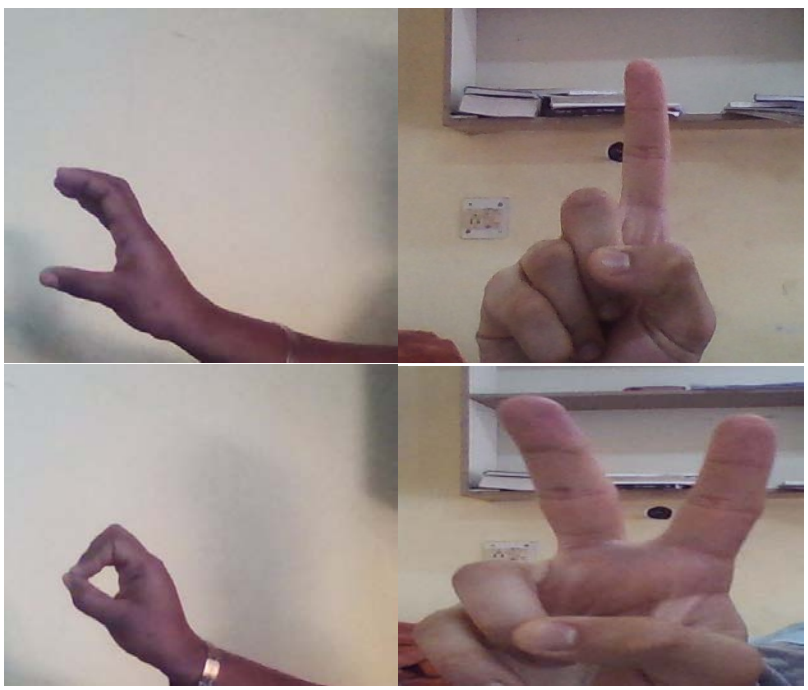

Figure 4.1: Database images

The dataset numbered 1 which is matching the images from within the database itself was first and foremost practical experiment conducted. For example, the character "C" was given as an input, the most relevant images which are most similar within dataset-1. The Figure 4.2 depicts the MATLAB command windows i.e. basically matching process and recognition.

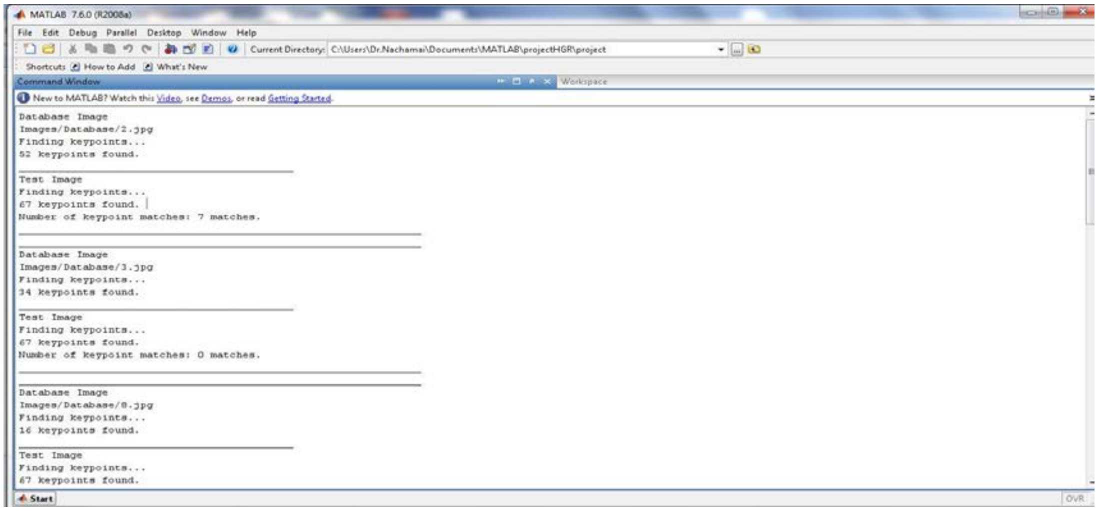

Figure 4.2: MATLAB simulation window

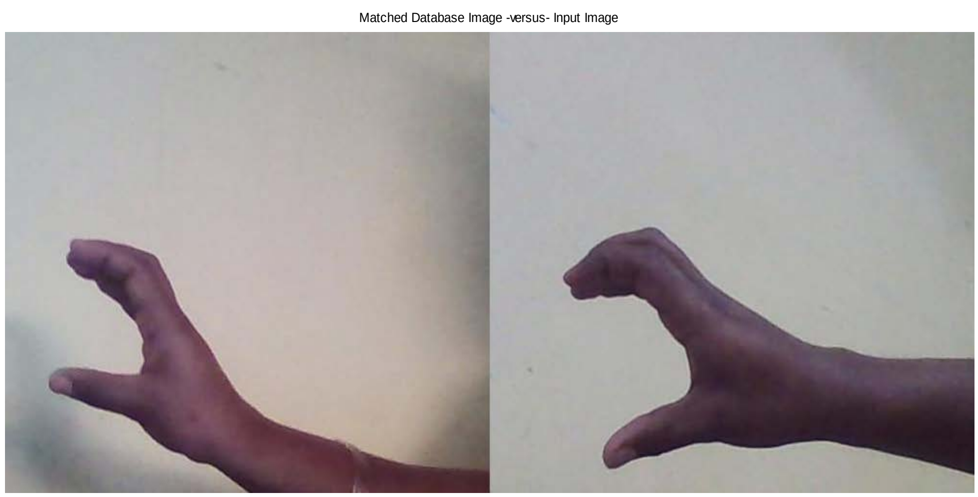

Figure 4.3: Match database image verses input image

From the Figure 4.3 it is clear that even if the images do not match completely it terms of exact color intensity and finger orientation the algorithm is still able to find maximum key point matches or we can say that it has focused on the descriptors i.e. the points on the edges where the most crucial information and data lies. It is evident from the images the thumb finger placement does not match exactly. We have been able to find out the right alphabet recognition. The images were taken with the help of webcam. The Figure 4.3 shows the test sample taken from an user for the same letter “C” and matched successfully.

Further, the extended experiment results with different gestures are depicted as follows.

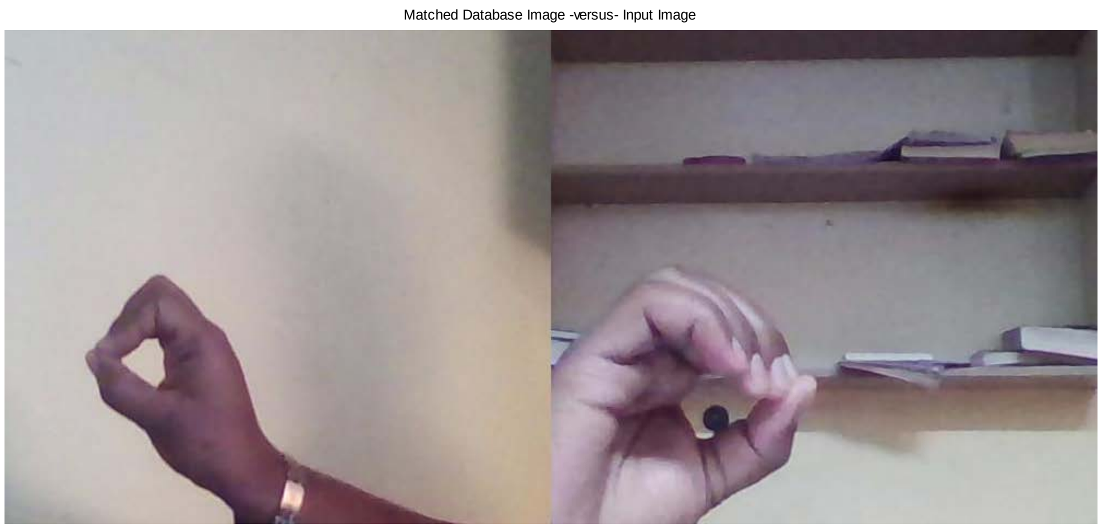

Figure 4.4: Database image match for letter “O”.

The Figure 4.4 depicts that the algorithm implemented was able to identify the letter “O”, here we have used the test sample which is a left hand image while the training sample was actually an image of right hand. So the algorithm is efficient enough in identifying the sign successfully with even two different hands with completely different axis and orientation

Conclusion:

In attempts of making robust and reliable system we have looked upon almost all the aspect that can influence the results and efficiency. Starting from pre-processing to implementation of SIFT algorithm, works were challenging and time consuming, but with all the odds we have shown accuracy up to 70% which is average accuracy. The system has shown good response for stored database of ‘Indian Sign Language’. Specialty of this system is that after certain modification that we have included in our project it can work efficiently on other sign language too.

Our system has also qualitatively qualified for real time processing, which makes it more versatile and unique to other system.

CHAPTER 5

CHALLENGES AND FUTURE WORK

Sign language which is mostly used by deaf and dumb people to communicate with people around them is totally different from spoken language. The fundamental of Sign language is based on two major properties which are spatial properties and iconicity properties. Various hand parameters that may be based on shape, movement, orientation of hands as well as facial expression which include mouth movements together constitute a significant part of Sign language. In this project, we have dealt with hand gestures and it's parameter, other body part gesture are not included in the paper. While doing preprocessing we come up with lots of technical issue like blurring of images due to movement of hand in front of camera, background brightness, user's distance with camera etc. We also faced time complexity which is quite large in the project that needs to be re-considered. In this project we have primarily worked upon letters not on words, because while processing with word time complexity went high that is a milestone yet to be achieved.

[14]In future, facial gesture recognition could be used in vehicles to alert drivers who are about to fall asleep. It can be used for better computer interaction with human.

CODE:

For capturing Image & pre processing:

```
clc
clear all
close all

vid = videoinput('winvideo',1);
set(vid, 'ReturnedColorSpace', 'RGB');
preview(vid);
pause(5);
he = getsnapshot(vid);


closepreview(vid); % im = getsnapshot(vid); alternative use

he=he(100:650,300:850,:);
he=imresize(he,[180,280]);
imwrite(he,'C:\Users\Aashish Barnwal\Desktop\h_sample.jpg');




cform = makecform('srgb2lab');
lab_he = applycform(he,cform);
ab = double(lab_he(:,:,2:3));
nrows = size(ab,1);
ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,2);

nColors = 3;
% repeat the clustering 3 times to avoid local minima
[cluster_idx, cluster_center] = kmeans(ab,nColors,'distance','sqEuclidean', ...
                                'Replicates',3);
pixel_labels = reshape(cluster_idx,nrows,ncols);
segmented_images = cell(1,3);
rgb_label = repmat(pixel_labels,[1 1 3]);

for k = 1:nColors
   color = he;
   color(rgb_label ~= k) = 0;
   segmented_images{k} = color;
end
```

```
fi=imsubtract(he,segmented_images{2});

imwrite(fi,'Images/Inputs/sample/h_sample.jpg');
input='Images/Inputs/sample/h_sample.jpg';
results=hgr(input);
```

Algorithm Implementation:

above code calls hgr function for further processing:

```
function results = hgr(input);

load theHGRDatabase

distRatio=0.67; %Distance Ratio for the SIFT Match methods. % default distRatio =
0.65
threshold=0.035; % default threshold=0.035
distRatioIncrement=0.05; % default distRatioIncrement=0.05
thresholdDecrement=0.005; % default thresholdDecrement=0.005

% 'Selecteds' indicate the selected Database Images.
% If you add 1.jpg to the Database folder, you have to change the first
% number as 1. For our case, you have to make the first 0 of 'Selected'as 1.
% Also note that, this array stores the candidate database images.
% At the end of the algorithm only 1 selected (matched) image is left
% inside this array
Selecteds=[0 2 0 4 0 0 0 0 9 0 0 12 0 0 15 0 0 0 0 20 0 0 0 0 0 0];

% 'mask' denotes the maximum character number (The maximum charachter
% number for ISL (Indian Sign Language) is set as 26.
mask=26;

% 'check' specifies the number of checks done so far.
% it is set as 1 for the initial check/test.
check=1;
% Since there isn't any process, the initial value of the matchFound is 0.
matchFound=0;

% StringArray is used for outputing the equivalent ASCII character
StringArray=Selecteds;
StringArray=StringArray+64;
% ---------------------------------------------------------------
% Step2) Run the main algorithm
while(sum(mask)>1)
   % Investigate the function 'formResults' for the details of the
   % algorithm
   results=formResults(input,distRatio,threshold,Selecteds);
```

```
    if(mask<=2)
        % Select the best candidate
        Selecteds=findMax(results(:,7),1); %Note that 7th field of results hold the validity
ratio of the validly matched keypoints
    else
        % Select the 3 best candidates
        % Note: This section could be changed by replacing the 'depth' of 3 as
        % the numberOfBestCandidates by setting a separate ValidityRatio threshold after
the first iteration.
        % For demonstration purposes it is left as 3.
        Selecteds=findMax(results(:,7),3);
    end

    % Inform that the n'th check is done and increment the check for the
    % next possible iteration
    disp('---------------');
    fprintf('Check %d Done.\n',check);
    disp('---------------');
    check=check+1;

    % Increment the distRatio for the next iteration for the selected
    % candidate database images in order to find more matched keypoints
    distRatio=distRatio+distRatioIncrement;
    if(distRatio>=0.9)
        distRatio=0.9;
    end

    % Decrement the threshold value in order to find more valid keypoints.
    threshold=threshold-thresholdDecrement;
    if(threshold<=0.01)
        threshold=0.01;
    end
    % Store the Selecteds inside 'mask' and enforce it to be under 1 in
    % order to exit the loop if only 1 selected item left. In the case of
    % having 2 candidates the loop will continue.
    mask=(Selecteds)./(Selecteds+1);
end
fprintf('End of tests...\n');
%    ----------------------------------------------------

for i=1:26
    if (Selecteds(i)~=0)
        matchFound=1;
        inputImage=imread(input);
        outputImage=imread(dataBase(Selecteds(i),:));
        imshow(outputImage);
        title('Matched Database Image -versus- Input Image');
```

```
        fprintf('Match Found: %c char.\n',StringArray(i));
    end
end

if(matchFound==0)
fprintf('No match found...\n');
end
```

```
function results=formResults(input, distRatio,threshold,Selecteds);
% For
load theHGRDatabase;

for i=1:size(Selecteds,2)
    if(i==Selecteds(i))
        % Match the images and get the matched keypoints' data
        [match1,match2,cx1,cy1,cx2,cy2,num]  = match(dataBase(Selecteds(i),:), input,
distRatio);

        % If the number of Matched Keypoints are greater than 2, start the
        % algorithm
        if num>2
            %Calculate the distances of the matched keypoints to the center of the
            %keypoints
            for j=1:num
                distance1(j)=sqrt((match1(j,2)-cx1).^2+(match1(j,1)-cy1).^2);
                distance2(j)=sqrt((match2(j,2)-cx2).^2+(match2(j,1)-cy2).^2);
            end

            % Sum the distances and calculate the Distance Ratio Array.
            distanceSum1=sum(distance1);
            distanceSum2=sum(distance2);

            if(distanceSum1==0)
                distanceSum1=1;
            end

            if(distanceSum2==0)
                distanceSum2=1;
            end

            for j=1:num
                distanceRatio1(j)=distance1(j)./distanceSum1;
                distanceRatio2(j)=distance2(j)./distanceSum2;
            end

            % This operation is done in order to determine the similar

            % The absolute of the difference of the points which are below
```

```
            % the given threshold are treated as valid matched keypoint.
            distanceMask=abs(distanceRatio1-distanceRatio2)<threshold;

            %Calculate the total valid points by summing the distanceMask
            distanceMaskSum=sum(distanceMask);
        else
            % If the number of matched keypoints are not greater than 2
            % than the number of valid points are directly 0.
            distanceMaskSum=0;
        end
        % Store the results
        results(i,1)=cx1; % X position of the database image's center point
        results(i,2)=cy1; % Y position of the database image's center point
        results(i,3)=cx2; % X position of the input(query) image's center point
        results(i,4)=cy2; % Y position of the input(query) image's center point
        results(i,5)=num; % Number of matched keypoints
        results(i,6)=distanceMaskSum; % Number of valid matched keypoints

        % Calculate the validity ratio of the keypoints simply by dividing
        % the valid matched keypoints by the matched keypoints.
        % Store the ratio inside the results array.
        if(num==0);
            validRatio=0;
        else
            validRatio=distanceMaskSum/num;
        end
        results(i,7)=validRatio;
        results(i,8)=i;

% Reset the parameters for the next iteration
        distanceRatio1=0;
        distanceRatio2=0;
        distanceMask=0;
        distanceMaskSum=0;

    else
        results(i,:)=0;
    end
end
```

finding descriptor and matched keypoints:

```
function [match1,match2,cx1,cy1,cx2,cy2,num]  = match(image1, image2, distRatio);
%close all;
match1=0;
match2=0;
cy1=0;
cx1=0;
```

```
cy2=0;
cx2=0;
% Find SIFT keypoints for Image-1
disp('Processing Image-1 (Database)');
disp(image1);
[im1, des1, loc1] = sift(image1);


disp('---');
disp('Processing Image-2 (Input/Query)');
[im2, des2, loc2] = sift(image2);



% For efficiency in Matlab, it is cheaper to compute dot products between
%  unit vectors rather than Euclidean distances.  Note that the ratio of
%  angles (acos of dot products of unit vectors) is a close approximation
%  to the ratio of Euclidean distances for small angles.
%
% distRatio: Only keep matches in which the ratio of vector angles from the
%   nearest to second nearest neighbor is less than distRatio.
distRatio = distRatio;
%distRatio = 1;

% For each descriptor in the first image, select its match to second image.
des2t = des2';                          % Precompute matrix transpose
for i = 1 : size(des1,1)
   dotprods = des1(i,:) * des2t;        % Computes vector of dot products
   [vals,indx] = sort(acos(dotprods));  % Take inverse cosine and sort results

   % Check if nearest neighbor has angle less than distRatio times 2nd.
   if  (vals(1) < distRatio * vals(2))
        myMatch(i) = indx(1);
   else
      myMatch(i) = 0;
   end
end



j=1;
for i = 1: size(des1,1)
 if (myMatch(i) > 0)
    %plot(loc1(i,2),loc1(i,1),'ro');
    match1(j,1)=loc1(i,1);
    match1(j,2)=loc1(i,2);
    match2(j,1)=loc2(myMatch(i),1);
    match2(j,2)=loc2(myMatch(i),2);
```

```
    j=j+1;
  end
end


% Create a new image showing the two images side by side.
im3 = appendimages(im1,im2);

% Calculate the center points
num = sum(myMatch > 0); %matched points

if (num > 1)
    s1=sum(match1);
    s2=sum(match2);
    cy1=s1(1)/num;
    cx1=s1(2)/num; %Center point for the database image

    cy2=s2(1)/num;
    cx2=s2(2)/num; %Center point for the input image
end

fprintf('Found %d matches.\n', num);
disp('-----------------------------------');
```

SIFT algorithm

```
function [image, descriptors, locs] = sift(imageFile)

% Load image
image = imread(imageFile);


 if isrgb1(image)
    image = rgb2gray(image);
 end

[rows, cols] = size(image);

% Convert into PGM imagefile, readable by "keypoints" executable
f = fopen('tmp.pgm', 'w');
if f == -1
    error('Could not create file tmp.pgm.');
end
fprintf(f, 'P5\n%d\n%d\n255\n', cols, rows);
fwrite(f, image', 'uint8');
fclose(f);
```

```
% Open tmp.key and check its header
g = fopen('tmp.key', 'r');
if g == -1
   error('Could not open file tmp.key.');
end
[header, count] = fscanf(g, '%d %d', [1 2]);
if count ~= 2
   error('Invalid keypoint file beginning.');
end
num = header(1);
len = header(2);
if len ~= 128
   error('Keypoint descriptor length invalid (should be 128).');
end

% Creates the two output matrices (use known size for efficiency)
locs = double(zeros(num, 4));
descriptors = double(zeros(num, 128));

% Parse tmp.key
for i = 1:num
   [vector, count] = fscanf(g, '%f %f %f %f', [1 4]); %row col scale ori
   if count ~= 4
      error('Invalid keypoint file format');
   end
   locs(i, :) = vector(1, :);

   [descrip, count] = fscanf(g, '%d', [1 len]);
   if (count ~= 128)
      error('Invalid keypoint file value.');
   end
   % Normalize each input vector to unit length
   descrip = descrip / sqrt(sum(descrip.^2));
   descriptors(i, :) = descrip(1, :);
end
fclose(g);
```

APPENDIX

Clc: clear workspace and closes all the figure.

Clear all : clears data stored in variables and command window .

videoinput: this function is used for taking video as input.

pause: this function is used to pause the sytem for certain seconds.

getsnapshot: As we are taking images as input show we take snapshot from video frame as input.

Imresize: is used for resizing image in certain window of mxn dimensions.

Imwrite: is used for writing image.

Makecform, **applycform**: is used to create color transformation structure.

Reshape: is used for reshaping image

Kmeans: is a partionining method for dividing n numbers of mutually exclusive clusters

Imsubtract: is used for subtracting image from one image to another pixel wise, for thus their size must agree.

Sum: sum is used to take summation of mxn dimensional data and put them in one row i.e 1xn dimensional.

Sqrt: it throw square root of given data.

Abs: abs stand for absolute, it gives absolute value of data.

Sort: this is used for sorting data set.

Acos: acos is equivalent to finding angle between two vectors.

Rgb2gray: used for convertion RGB image to Gray image.

Fopen: fopen is used to open a file and get access of read write or both.

Fscanf: is used to read data from text file this reading can be done in different fashion.

REFRENCES:-

[1] Noor A. Ibraheem&Rafiqul Z. Khan, (2012) “Vision Based Gesture Recognition Using Neural Networks Approaches: A Review”, International Journal of human Computer Interaction (IJHCI), Vol. 3, No. 1.

[2] H. Dardas & Mohammad Alhaj, (year) “Hand Gesture Interaction with a 3D Virtual Environment”, The Research Bulletin of Jordan ACM, Vol. 2, No. 3, pp 86-94.

[3] Sanjay Meena,(2011), “A Study on Hand Gesture Recognition Technique”, National Institute Of Technology, India(thesis-M.Tech).

[4] Hand Gesture Recognition Using Appearance Features by Yanwen Chong, Jianfeng Huang, Shaoming Pan.

[5] Had Starner & Alex Pentland ,(1995), Real-Time American Sign Language Recognition from Video Using Hidden Markov Models, (IEEE), pp. 265-271.

[6] D. Kumarage, S. Fernando, P. Fernando, D. Madushanka& R. Samarasinghe, (2011), “Real-time Sign Language Gesture Recognition Using Still-Image Comparison & Motion Recognition”, 6th International Conference on Industrial and Information Systems(ICIIS), Sri Lanka.

[7] D. Kumarage, S. Fernando, P. Fernando, D. Madushanka& R. Samarasinghe, (2011), “Real-time Sign Language Gesture Recognition Using Still-Image Comparison & Motion Recognition”, 6th International Conference on Industrial and Information Systems(ICIIS), Sri Lanka.

[8] Rauta ray_Agrawal_2012_Real_time_hand_gesture_recognition.

[9] A Research Study of Hand Gesture Recognition Technologies and Applications by Carl A. Pickering, Keith J. Burnham, Michael J. Richardson

[10] A Research Study of Hand Gesture Recognition Technologies and Applications by Carl A. Pickering, Keith J. Burnham, Michael J. Richardson

[11] Real-time Motion-based Gesture Recognition by Mark Bayazit, Alex Couture-Beil, Greg Mori

[12] Hand gesture recognition using a real-time tracking metho by Feng-Sheng Chen, Chih-Ming Fu, Chung-Lin Huang

[13] Vision-Based Gesture Recognition: A Review by Ying Wu and Thomas S. Huang

[14] Hand Gesture Recognition System Swapnil D. Badgujar , Gourab Talukdar , Omkar Gondhalekar, Mrs. S.Y. Kulkarni

[15] A Study on Hand Gesture Recognition Technique, Delhi University

[16] Hand Gesture Recognition based on Digital Image Processing using MATLAB By Tahir Khan under supervision of Dr. Amir Hassan Pathan Faculty of Engineering, Sciences and Technology, IQRA University

[17] A novel real time hand detection based on skin-color by Yuh-Rau Wang; Wei-Hung Lin; Ling Yang

[18] A real time hand gesture recognition system using motion history image by Chen-Chiung Hsieh; Dung-Hua Liou; David Lee

[19] Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering by L. Bretzner; I. Laptev; T. Lindeberg

[20] An HMM-based threshold model approach for gesture recognition by Hyeon-Kyu Lee; J.H. Kim

[21] Parametric hidden Markov models for gesture recognition By A.D. Wilson; A.F. Bobick

[22] Recognition of Indian Sign Language using feature fusion by Subhash Chand Agrawal; Anand Singh Jalal; Charul Bhatnagar

Submission Info

SUBMISSION ID	806306218
SUBMISSION DATE	28-Apr-2017 10:57
SUBMISSION COUNT	1
FILE NAME	Project_Report_Aashish_...
FILE SIZE	524.6K
CHARACTER COUNT	30644
WORD COUNT	5940
PAGE COUNT	36

ORIGINALITY

OVERALL	13%
INTERNET	11%
PUBLICATIONS	11%
STUDENT PAPERS	6%

GRADEMARK

LAST GRADED	N/A
COMMENTS	0
QUICKMARKS	