

SMART SPEAKING GLOVE FOR SPEECH IMPAIRED PEOPLE

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering
by

NINDRA GURU SAI PRAKASH (38110372)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC**

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL – 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **NINDRA GURU SAI PRAKASH (38110372)**, who carried out the project entitled "**SMART SPEAKING GLOVE FOR SPEECH IMPAIRED PEOPLE**" under my supervision from November 2021 to April 2022.

InternalGuide
Dr. K. MohanaPrasad

Head of the Department
Dr. S. VIGNESHWARI, M.E., Ph.D., / Dr. L. LAKSHMANAN, M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I **NINDRA GURU SAI PRAKASH (38110372)**, hereby declare that the Project Report entitled "**SMART SPEAKING GLOVE FOR SPEECH IMPAIRED PEOPLE**" done by me under the guidance of Dr. K. MohanaPrasad at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.SASIKALA M.E., Ph.D, Dean, School of Computing ,Dr.S.VIGNESHWARI M.E., Ph.D. and Dr.L.LAKSHMANAN M.E., Ph.D. , Heads of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. K. MohanaPrasad** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

This aims to present a real time system for hand gesture recognition on the basis of detection of some meaningful shape based features like orientation, centre of mass, status of fingers in terms of raised or folded fingers of hand and their respective location in image. This hand gesture recognition system provides us natural, innovative, user friendly way of interaction with the computer which is more familiar to us. Gesture recognition has a wide area of application including human machine interaction, sign language, game technology robotics etc. Opencv and efficient feature extraction technique is presented for hand gesture recognition. AdaBoost based hand-pose detectors are trained with a reduced Haar-like feature set to make the detector robust. The corresponding context-free grammar based proposed method gives effective real time performance with great accuracy and robustness for more than four hand gestures. Rectangles are creating some problem due to that we have also implement the alternate representation method for same gestures i.e. fingertip detection using convex hull algorithm. This method deals with the dynamic aspects of gesture.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	x
1.	CHAPTER 1:INTRODUCTION	1
	1.1 GESTURES	1
	1.2 OPEN CV	2
	1.3 IMAGE PROCESSING	3
	1.4 OBJECTIVE	3
2.	CHAPTER 2:LITERATURE SURVEY	4
3.	CHAPTER 3: METHODOLOGY	7
	3.1 DESIGN	7
	3.2 CONSTRAINTS	8
	3.3 SYATEM ARCHITECTURE	9
	3.4 METHODOLOGIES	10
	3.5 MODULES	10
	3.6 MODULE DESCRIPTION:	10
	3.6.1Frames Capture	10
	3.6.2 Blur Frame	11
	3.6.3 Frame Segmentation	12
	3.6.4 Draw Contours	12
	3.6.5 Find Convex Hull and Convexity Defects	13

	EXISTING SYSTEM	14
	DRAWBACKS IN EXISTING SYSTEM	14
	PROPOSED SYSTEM	14
	ADVANTAGES IN PROPOSED SYSTEM	15
	REQUIREMENTS	15
	HARWARE REQUIREMENTS	15
	SOFTWARE REQUIREMENTS	16
	SOFTWARE SPECIFICATION	16
4.	CHAPTER 4:RESULTS AND DISCUSSION	17
	SAMPLE IMAGES CONTAINED IN THE DATABASE	17
	RESULTS	18
5.	CHAPTER 5:CONCLUSION AND FUTURE ENHANCEMENT	26
	CONCLUSION	26
	FUTURE ENHANCEMENT	26
	APPLICATIONS	26
	REFERENCES	29
	APPENDICES	30
	A. SOURCE CODE	30
	B. SCREENSHOTS	34

LIST OF FIGURES

S.NO	FIGURE	PG.NO
1.	Frame Capture	11
2.	Blurr Image	12
3.	Frame Segmentation	12
4.	Contours Image	13
5.	Convex Hull and Convexity Defects	14
6.	Few of the images in the database	17
7.	Candidate Image in the database	18
8.	Rotated and scaled query image	19
9.	Rotated and scaled candidate image	19
10.	Distance Transform of the query image	19
11.	Distance Transform of the candidate image	19
12.	A similar case as Case 1 (A)	20
13.	Image Capture	20
14.	The rotated and scaled image of the query image	21
15.	The rotated and scaled image of the candidate image	21
16.	Distance Transform of the query image	21
17.	Distance Transform of the candidate image	21

18.	Image Capture	22
19.	Candidate image in the database	22
20.	Rotated and scaled image of the query image	23
21.	Rotated and scaled image of the candidate image	23
22.	Distance Transform image of the query image	23
23.	Distance Transform image of the candidate image	23
24.	Frame Capture	24
25.	Candidate image in the database	24
26.	Rotated and scaled query image	25
27.	Rotated and scaled candidate image	25
28.	Distance transform of the query image	25
29.	Distance Transform of the candidate image	25

LIST OF ABBREVIATION

S.NO	ABBREVIATION	EXPANSION
1.	opencv	open source computer vision

CHAPTER 1

INTRODUCTION

For the past decade, highly skilled interpreters were used to make communication easy between deaf and hearing people. The main objective is to build correct and effective translation between spoken language and sign language. Because these kinds of interpreters are expensive, frequent use of interpreters are not sustainable. Pre-notification is to be required when someone like to use it. The key point is that to support the millions of deaf people, a significant number of well-trained interpreters will not be available. To bridge this gap a Machine Translator (MT) system which translates to sign language from spoken language and vice versa. For an example, it translates from English to American Sign Language (ASL) and Indian Sign Language (ISL). Visualization of sign language is possible by MT. This system is used for numerous applications, like telephony accessibility for deaf communities and sign language education. Through this the Deaf can easily access a variety of public and information services. When you use gesture, you are entering into a whole history of human communication, because there is no language that exists entirely without gesture. People can't communicate without gesture. It's so connected to intention that there is a phrase "empty gesture," used to mean an action or movement that is without genuine feeling.

GESTURES

Gestures are one of the first forms of communication when people of different languages meet, and no one knows in which language they should express their feelings or needs. In gesture reorganization the receiver receives the gestures made by users. Gestures are used to clarify and support your words. Gestures are highly visible, and it effectively paints your ideas. In recent days many gesture recognition systems are used to communicate between the Deaf and hearing communities. Different kind of gestures are facial expressions, hand movements, eye movements etc. Expressive, meaningful body motions are considered as Gestures. In gesture, various physical

movements of the body, such as hands, face, arms, and fingers, are involved. Gestures are important to interact with the environment and convey meaningful information. Gesture recognition is utilized as a part of numerous applications.

Gesture recognition has been a very interesting problem in Computer Vision community for a long time. This is particularly due to the fact that segmentation of foreground object from a cluttered background is a challenging problem in real-time. The most obvious reason is because of the semantic gap involved when a human looks at an image and a computer looking at the same image. Humans can easily figure out what's in an image but for a computer, images are just 3 -dimensional matrices. It is because of this, computer vision problems remains a challenge.

OPEN CV

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source BSD license.

OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation. Wrappers in other languages such as C#, Perl, Ch,Haskell, and Ruby have been developed to encourage adoption by a wider audience.

Since version 3.4, OpenCV.js is a JavaScript binding for selected subset of OpenCV functions for the web platform.

All of the new developments and algorithms in OpenCV are now developed in the C++ interface.

OpenCV runs on the following desktop operating systems: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD. OpenCV runs on the following mobile operating systems: Android, iOS, Maemo, BlackBerry 10. The user can get official releases from SourceForge or take the latest sources from GitHub. OpenCV uses CMake.

IMAGE PROCESSING

In computer science, **digital image processing** is the use of a digital computer to *process* digital images through an *algorithm*. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems. The generation and development of digital image processing are mainly affected by three factors: first, the development of computers; second, the development of mathematics (especially the creation and improvement of discrete mathematics theory); third, the demand for a wide range of applications in environment, agriculture, military, industry and medical science has increased.

OBJECTIVE

This aims to present a real time system for hand gesture recognition on the basis of detection of some meaningful shape based features like orientation, centre of mass, status of fingers in terms of raised or folded fingers of hand and their respective location in image. This hand gesture recognition system provides us natural, innovative, user friendly way of interaction with the computer which is more familiar to us.

CHAPTER 2

LITERATURE SURVEY

[1] This research work presents a prototype system that helps to recognize hand gesture to normal people in order to communicate more effectively with the special people. Aforesaid research work focuses on the problem of gesture recognition in real time that sign language used by the community of deaf people. The problem addressed is based on Digital Image Processing using Color Segmentation, Skin Detection, Image Segmentation, Image Filtering, and Template Matching techniques. This system recognizes gestures of ASL (American Sign Language) including the alphabet and a subset of its words.

[2] The proposed system is a cost-effective and possible to minimize the distance between hearing and speech impaired people with normal people. Proposed system captures the hand signs and compare with existing database and then accordingly converted into text. This system is using an Image Processing algorithm which processes the detection and extraction of the input hand gesture from the image stream. In this system we are using functions like skin color based thresholding, contour detection for detection of hands and identification of important points on the hand respectively. The distance between these contour points from the centroid of the hand becomes our feature vector against which we will train our neural network.

[3]The concept of hand gesture recognition has been widely used in communication, artificial intelligence and robotics. It is a staple method of interaction especially for the deaf and the blind. Techniques for recognizing hand gestures are in great demand. Many algorithms have been discovered for this purpose, each of them having their own advantages and disadvantages. In this paper we present a novel algorithm for hand recognition using image processing and explore its application in security based systems.

[4]Hand Gesture recognition and Human Computer Interaction is an open research problem as the main purpose behind gesture recognition research is to identify a particular human gesture by computer. Hand gesture recognition based man-machine interface is being currently developed and researched upon both by industry and academia. In this paper, we develop a technique for recognizing the gestures made by user using MATLAB and implement this technique to control the computer remotely without any keyboard or mouse. We present a real-time hand tracking technique by using a white cloth glove with a red color marker placed at fingertip. In this paper, a hand gesture recognition method based on color marker detection is presented as we have used a red color marker that is mounted on the fingertip of right hand to track the user's hand. We proposed a technique for natural, finger-tip based interaction with computer.

[5]The primary goal of Gesture Recognition is to create a system which can recognize specific human gestures and can be used to convey information or for device control. This paper aims at development of a Hand Gesture Recognition (HGR) system captured from a webcam in real time. The system consists of three stages: image acquisition, feature extraction, and recognition. In the first stage input image of hand gestures are acquired by digital camera in approximate frame rate. In second stage a rotation, scaling and orientation invariant feature extraction method has been introduced to extract the feature of the input image. Finally, edge detection is used and K-Curvature algorithm is applied to recognize the hand gestures.

[6]Mobile and wearable devices are continuously optimized towards a small outline. At the same time the number of functions in these devices continuous to increase. While this Development is clearly beneficial for the ubiquity of mobile and wearable systems, e.g. for using the systems during daily activities, it hampers interaction. MEMS sensors play a major role in the μ IMU due to their low-cost and miniaturized size. We use MEMS sensors to measure the 3D accelerations and 3D angular rates. A Micro Control Unit (MCU) performs coordinate transformations and filtering calculations. In this paper, we are measuring the actions performed on sign languages in to an equal acceleration

values. The acceleration values are measured in 3 axis, using a 3-axis accelerometer. Every action generates a unique set of acceleration values in all the axis. The action and the corresponding acceleration values are placed in to a look up table along with appropriate voice commands. Once the glove is placed in the hands, whenever an action for sign language is performed, the acceleration values are obtained and are checked with the look up table. If the acceleration values match with any set of the look up table, then the corresponding action is identified and the voice channel is selected. Here we are using a 3-axis accelerometer. The 3 channel analog output is fed to the micro controller's ADC channels. The Processed output is then output via the Ports available on the controller. The output is fed to the Voice chip, where the pre-recorded voices are stored. By activating the corresponding channel the voice will be played in the speaker.

[7] This paper presents an MEMS accelerometer mostly based on gesture recognition algorithm and its applications. The hardware module consists of a triaxial mems accelerometer, microcontroller, and zigbee wireless transmission module for sensing and collecting accelerations of handwriting and hand gesture trajectories. Users will use this hardware module to write down digits, alphabets in digital kind by making four hand gestures. The accelerations of hand motions measured by the accelerometer are transmitted wirelessly to a personal computer for trajectory recognition. The trajectory algorithm composed of information assortment collection, signal preprocessing for reconstructing the trajectories to attenuate the cumulative errors caused by drift of sensors. So, by changing the position of MEMS (micro electro mechanical systems) we can able to show the alphabetical characters and numerical within the PC.

CHAPTER 3

METHODOLOGY

Design is a meaningful engineering representation of a thing that is to be built. This involves:

Contours: Contours are sequences of points defining a line/curve in an image. Contour matching can be used to classify image objects.

The property possessed by the curving of a line or surface roundness - the property possessed by a line or surface that is curved and not angular straightness - freedom from crooks or curves or bends or angles crookedness - having or distinguished by crooks or curves or bends or angles

Database: Contains the images of various hand gestures

Moments: Image moments are useful to describe objects after segmentation. Simple properties of the image, which are found via image moments, include area (or total intensity), its centroid and information about its orientation.

Ratio of the two distance transformed images of the same size = (No of pixels whose difference is zero or less than a certain threshold) / (Total number of pixels in the distance transformed image)

DESIGN

Step 1: User takes a picture of the hand to be tested either through the cell phone camera or from the Internet.

Step 2: The image is converted into gray scale and smoothed using a Gaussian kernel.

Step 3: Convert the gray scale image into a binary image. Set a threshold so that the pixels that are above a certain intensity are set to white and those below are set to black.

Step 4: Find contours, then remove noise and smooth the edges to smooth big contours and melt numerous small contours.

Step 5: The largest contour is selected as a target.

Step 6: The angles of inclination of the contours and also the location of the center of the contour with respect to the center of the image are obtained through the bounding box information around the contour.

Step 7: The hand contours inside the bounding boxes are extracted and rotated in such a way that the bounding boxes are made upright (inclination angle is 0) so that matching becomes easy.

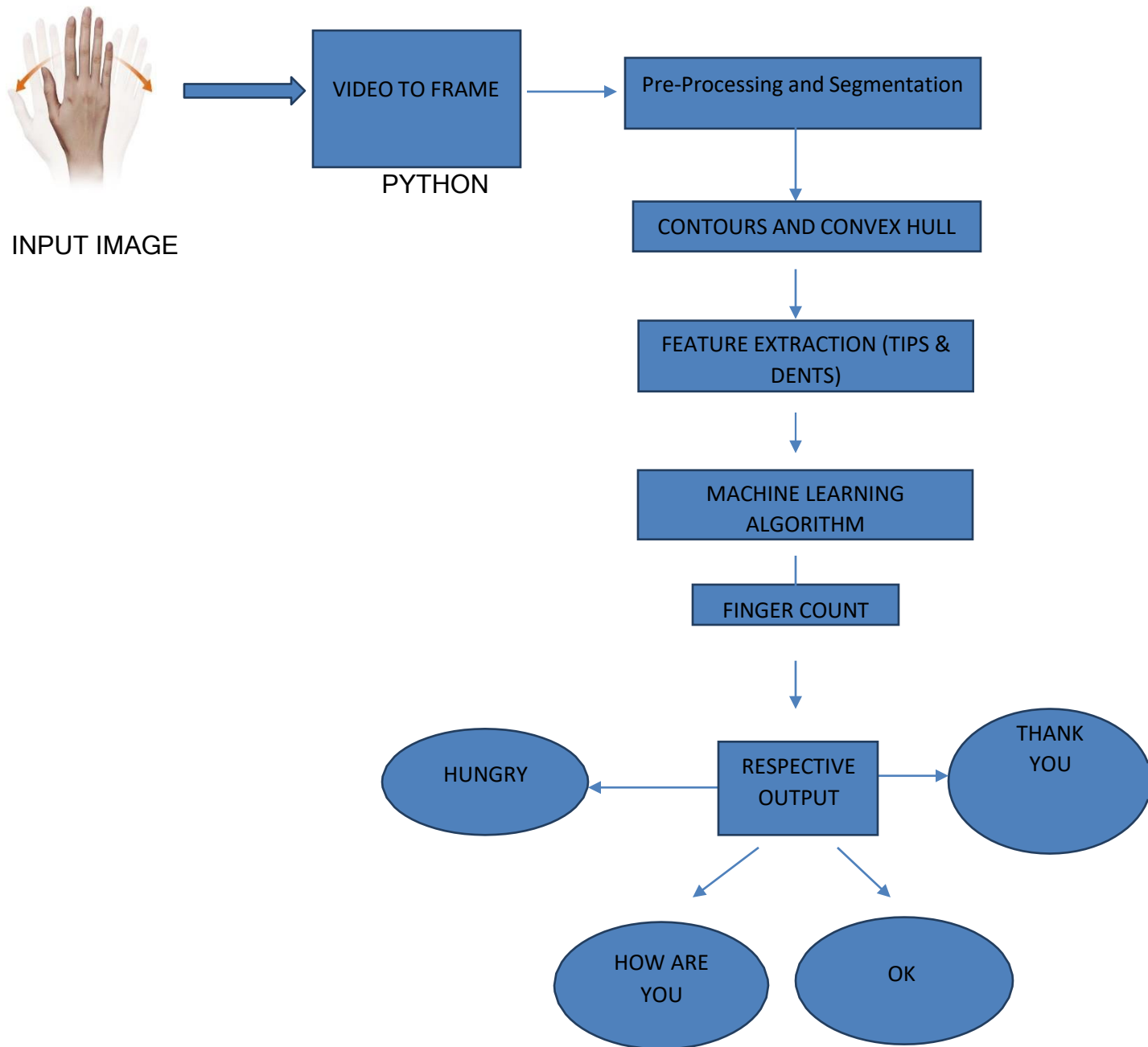
Step 8: Both the images are scaled so that their widths are set to the greater of the two widths and their heights are set to the greater of the two heights. This is done so that the images are the same size.

Step 9: The distance transform of both the query image and the candidate images are computed and the best match is returned.

CONSTRAINTS:

1. The picture of the hand must be taken against a dark background
2. The program recognizes a limited number of gestures as long as there are gestures similar to them in the database.
3. We must have each gesture with at least four orientations at 90 degrees each to return the best match.

SYSTEM ARCHITECTURE



The modules listed below will help in achieving the required result. We have explored Background Subtraction, Thresholding, Segmentation, Contour Extraction, Convex Hull and convexity defects. We have followed the methodology to quickly recognize hand gestures.

METHODOLOGIES

Following modules involves

MODULES:

1. Frames Capture
2. Blur Frame
3. Frame Segmentation
4. Draw Contours
5. Find Convex Hull and Convexity Defects

MODULE DESCRIPTION:

Frames Capture

The input data can be a frame or a sequence of video frames, taken by a Raspberry Pi camera module pointed toward user's hand. The frames captured with simple background and stable light. Region of Interest (ROI) is the hand region, so it captured the images of the hand and converts them to gray scale in order to find the ROI i.e. the portion of the image that is further interested for image processing. Our ROI is the the hand region, so we capture the images of the hand and convert them to grayscale.

GRAY SCALE: We convert an image from RGB to grayscale and then to binary in order to find the ROI i.e. the portion of the image we are further interested for image processing. By doing this our decision becomes binary: "yes the pixel is of interest" or "no the pixel is not of interest".



FIG 1.Frame Capture

Blur Frame

In image processing, a Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail. The processing of blur frame shown in "Figure 2" starts with noise reduction using Gaussian Blurring on the original frame. Blur frame is necessary process for frame enhancement and for getting good results. Blurring is used for smoothing frames and reducing noise and details from the frame. With blurring, smooth transformation from one color to another and reduction of the edge contents are satisfied. Mathematically, applying a Gaussian blur to an image is the same as convolving the image with a Gaussian function. This is also known as a two-dimensional Weierstrass transform.

I've used Gaussian Blurring on the original image. We blur the image for smoothing and to reduce noise and details from the image. We are not interested in the details of the image but in the shape of the object to track.

By blurring, we create smooth transition from one color to another and reduce the edge content. We use thresholding for image segmentation, to create binary images from grayscale images.



FIG 2.Blurr Image

Frame Segmentation

The task of segmenting and grouping pixels that are tracked is simplified by the high quality of the footage captured for most stop motion animations. Additionally, scenes shot with a moving camera tend to be the exception, so background subtraction is a natural choice for segmenting the action. If a clean background plate is not available, median filtering in the time domain can usually generate one. It is observed a pixel location over the entire sequence, sorting the intensity values (as many as there are frames). By choosing the median, the background can be reconstituted one pixel at a time.



FIG 3.Frame Segmentation

Draw Contours

The basic idea in active contour models or snakes is to evolve a curve, subject to constraints from a given image in order to detect objects in that image. For instance,

starting with a curve around the object to be detected, the curve moves toward its interior normal and has to stop on the boundary of the object.

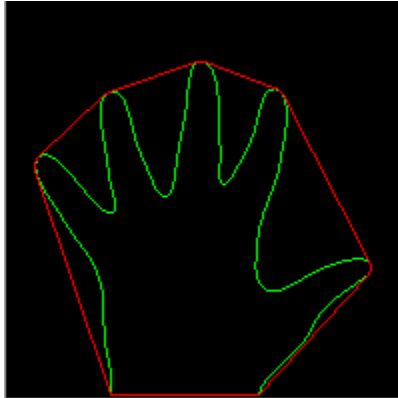


FIG 4.Contours Image

Find Convex Hull and Convexity Defects

In this project, the convex points are considered to be the tips of the fingers. Therefore, it finds convexity defects, which is the deepest point of deviation on the contour. By this, it can find the number of fingers extended and then it can perform different functions according to the number of fingers extended. Convex Hull used to find the fingertips, is the convex set enclosing the hand region. The red dots are the defect points appeared in every valley. Depending upon the number of defect points, the number of fingers unfolded is calculated.

In Convex Hull algorithm first step is segmentation of the hand image that contains the hand to be located. In order to make this process it is possible to use shapes, but they can be changed greatly in interval that hand moves naturally. So, we select skin-color to get characteristic of hand. The skin-color is a distinctive cue of hands and it is invariant to scale and rotation [8]. In the next step we use the estimated hand state to extract several hand features to define a deterministic process of finger recognition. After the hand is segmented from the background, a counter is extracted. The counter vector contains the series of coordinates of edges of hand. Then the processing of counter vector gives the location of the finger tip.

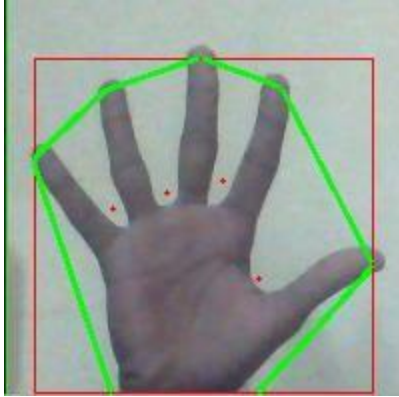


FIG 5. Convex Hull and Convexity Defects

EXISTING SYSTEM

In existing system, mems based hand movements, head movements, leg movements are been recognized. In this sensor has to be placed on the people body which causes radiation damage to the peoples. Our literature survey states all base papers states gesture are recognition using matlab and mems sensor. MEMS are termed as micro electro mechanical system. Also in existing system we use accelerometer sensor. An accelerometer is a device that measures the physical acceleration.

The approach which is commonly used for acquiring information necessary for gesture recognition system.

1. Methods which use Data gloves: mechanical or optical sensors are attached to a glove that transduces finger flexions into electrical signals for determining the hand posture. This approach forces the user to carry a load of cables which are connected to the computer and hinders the ease and naturalness of the user interaction.

3.7.1 Drawbacks in Existing system

In the existing system sensor has to be placed on the people body which causes radiation damage to the people.

PROPOSED SYSTEM

The essential aim of hand gesture recognition system is to establish a communication between human and computerized systems for the sake of control. The recognized gestures are used to establish a smooth communication channel between human &

computers and also a deaf people find it easy to communicate with normal human beings in sharing their thoughts.

The research effort centralizes on the efforts of implementing an application that employs computer vision algorithms and gesture recognition techniques which in turn results in developing a low cost interface device for interacting with objects in virtual environment using hand gestures. In this project, we use python to analyze the hand gesture open CV based machine learning and efficient feature extraction technique.

3.8.1 Advantages in Proposed system

The recognized gestures are used to establish a smooth communication channel between human & computers and also a deaf people find it easy to communicate with normal human beings in sharing their thoughts.

REQUIREMENTS

These are the requirements for doing the project. Without using these tools and software's we can't do the project. So we have two requirements to do the project. They are

- 1.HardwareRequirements.
- 2.SoftwareRequirements.

HARWARE REQUIREMENTS

All the physical equipment's i.e. input devices, processor, and output device & inter connecting processor of the computer s called as hardware.

- Hard Disk minimum of 100 GB.
- RAM minimum of 2 GB.
- Dual Core and up ,15" Monitor.
- Integrated webcam or external webcam (15 -20 fps).

SOFTWARE REQUIREMENTS

A set of instructions or program required to make hardware platform suitable for desired task is known as software. Software can also be defined as the utility programs that are required to drive hardware of computer.

- Operating system- Microsoft Windows 7 SP 1 or above
- Python

SOFTWARE SPECIFICATION

The OpenCV library is basically used in HCI, robotics, biometrics, image processing and other areas where visualization is important and includes an implementation of Haar classifier detection and training. To train our own classifiers, two set of images are required. One set which referred as the negative images that contains an image or scene which does not contain the interested object. In this case the hand gesture feature that look exactly like the positive one, except those containing same hand gesture. The other set of images are positive images where they look exactly like the object which we want to detect. It is also important that they should be different in lighting and background. In our implementation, five hand postures are tested.

CHAPTER 4

RESULTS AND DISCUSSION

SAMPLE IMAGES CONTAINED IN THE DATABASE:

In our experiment, we will be identifying limited number of gestures, which is shown as below. The query image even with slight orientation (+ or – 30 degrees) will be able to match with one of the image contained in the database. Images with different orientations are present within the database. In order to maintain performance, we have a limited number of gesture images in our database. New gesture images can be added to the database without any pre-processing.

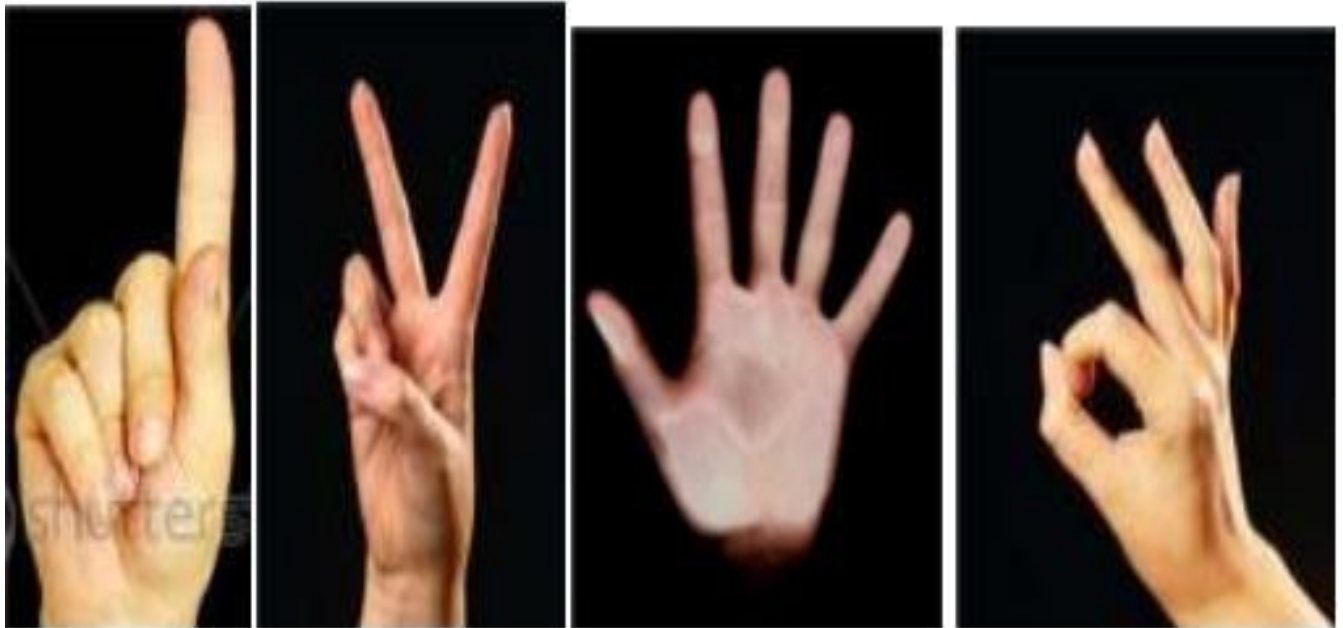
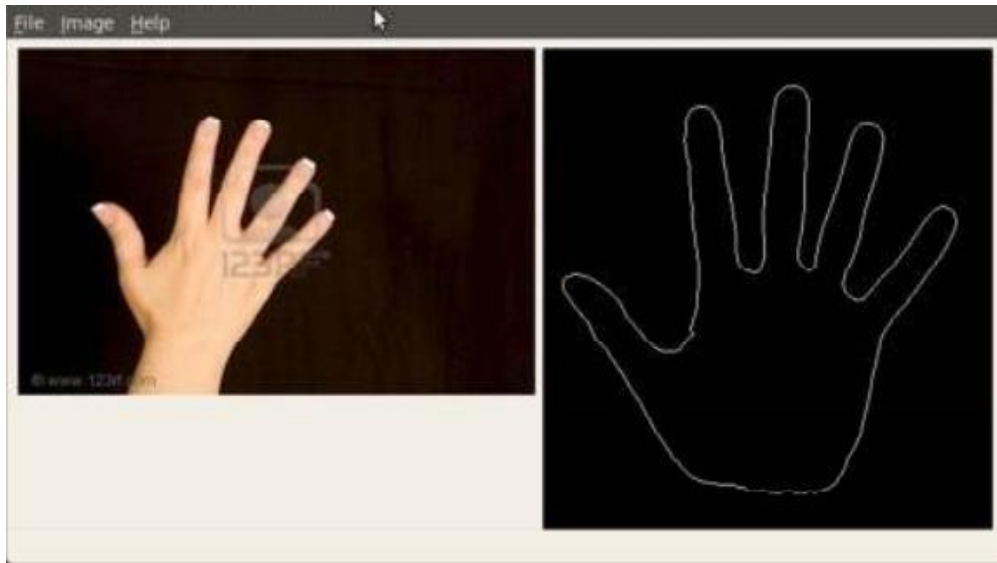


FIG 6. Few of the images in the database.

RESULTS:

Case 1:(A)



In the above case, we can see the query image on the left hand side and the matched candidate image from the database on the right.



FIG 7.Candidate Image in the database

Both the images are converted to binary images and the contours are computed. Using the bounding box information obtained from the contours, we get the angle of inclination of the contours and also, the center, height, and width of the bounding box. The results obtained after rotating and scaling the query and candidate images are:



FIG 8. Rotated and scaled query image



FIG 9. Rotated and scaled candidate image

candidate image

Now, the distance transforms of both the query image and the candidate images are computed as follows:



FIG 10. Distance Transform of the query candidate image



FIG 11. Distance Transform of the image

Now, the difference of the two image images is computed and the ratio of the match is found by the number of pixels whose difference between the two corresponding pixels is zero or below a certain threshold divided by the total number of pixels in one of the image. If the ratio is above 65% then the candidate image is determined as a match and returned.

(B)

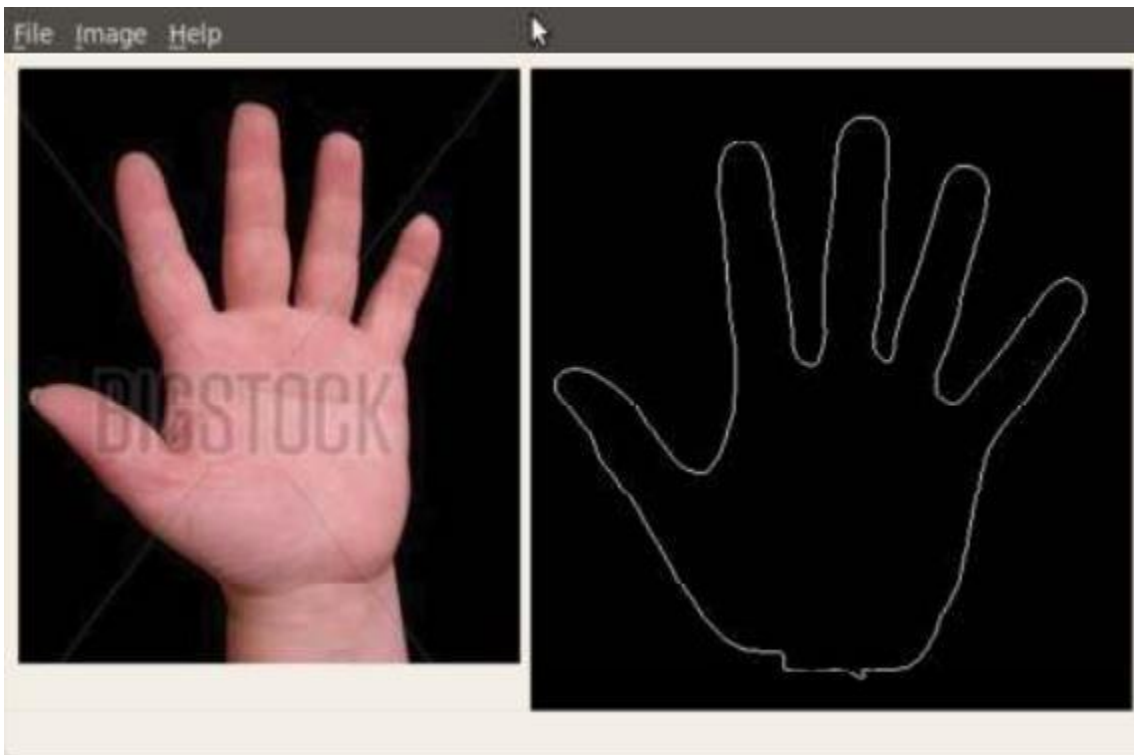


FIG 12.A similar case as Case 1 (A)

Case 2:



FIG 13.Image Capture

In the above case, the hand gesture on the left hand side is slightly tilted and still the right gesture from the database is returned.



FIG 14. The rotated and scaled image of the query image



FIG 15. The rotated and scaled image of the candidate image

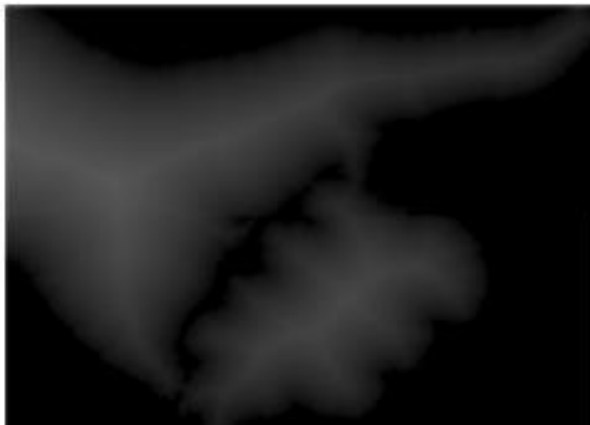


FIG 16. Distance Transform of the query image



FIG 17. Distance Transform of the candidate image

CASE 3:

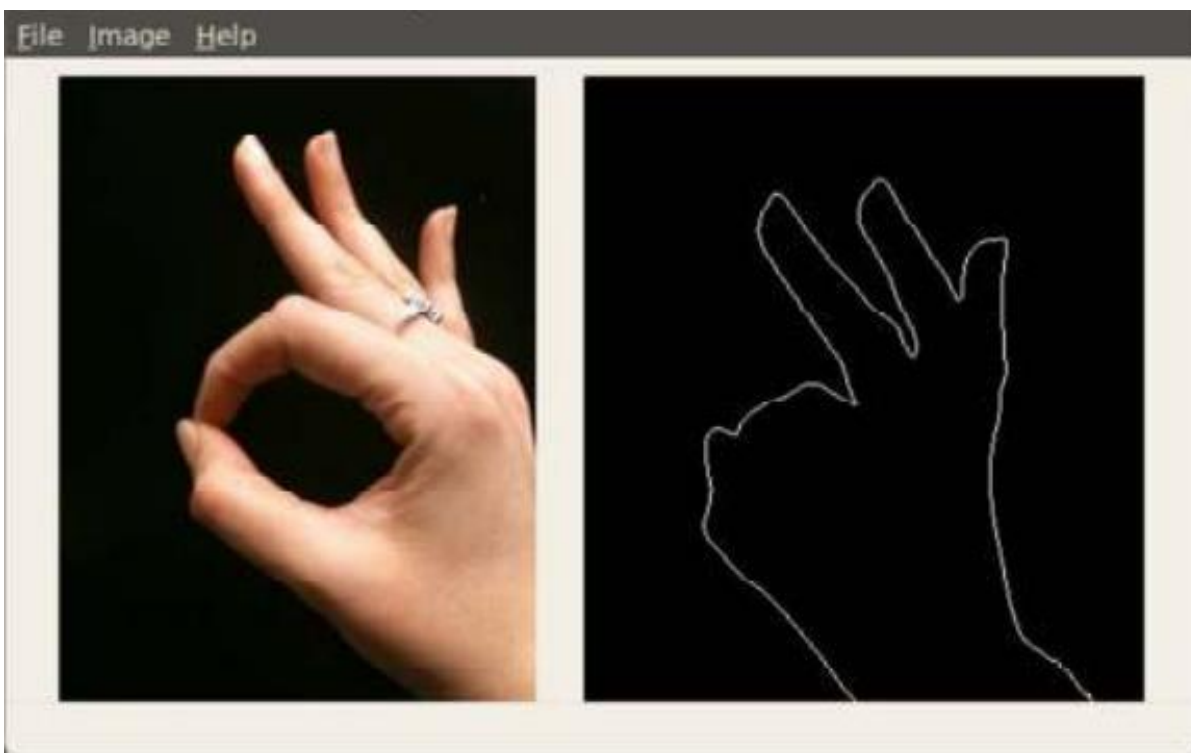


FIG 18. Image Capture



FIG 19. Candidate image in the database



FIG 20. Rotated and scaled image of the query image



FIG 21. Rotated and scaled image of the candidate image



FIG 22. Distance Transform image of the query image



FIG 23. Distance Transform image of the candidate image

Case 4:

In the below case, the program returns the right gesture image, even though the database doesn't contain the left hand gesture as in the query image because, the candidate image passes the ratio test.

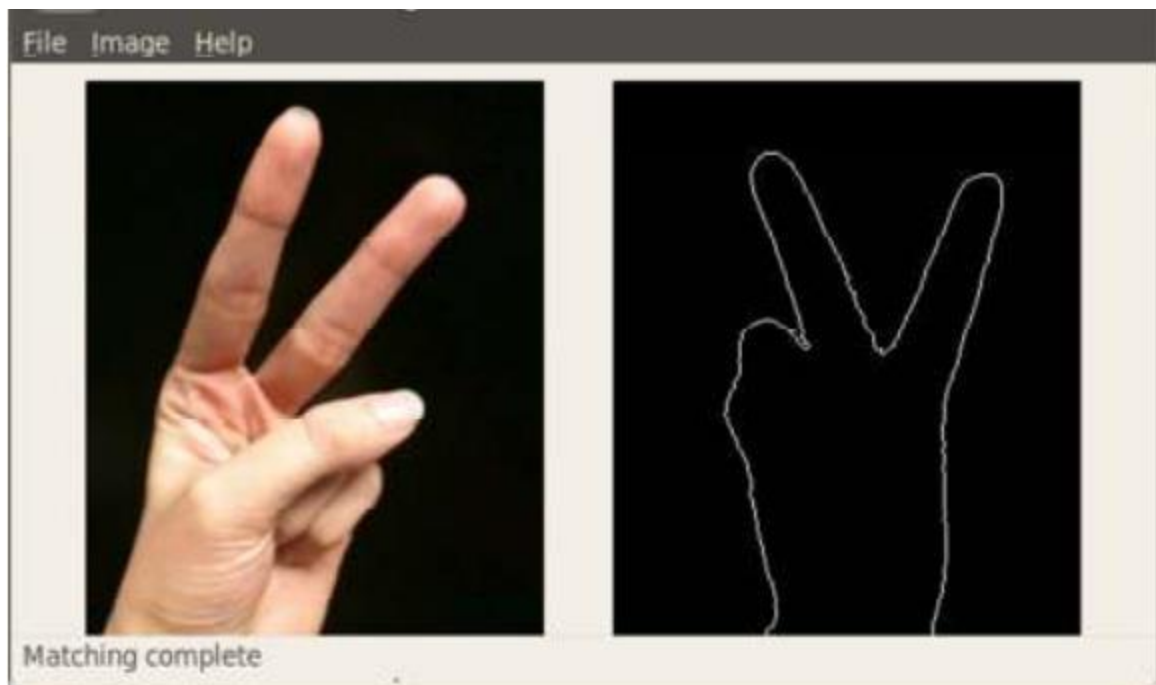


FIG 24.Frame Capture



FIG 25.Candidate image in the database



FIG 26. Rotated and scaled query image



FIG 27. Rotated and scaled candidate image



FIG 28. Distance Transform of the query candidate image

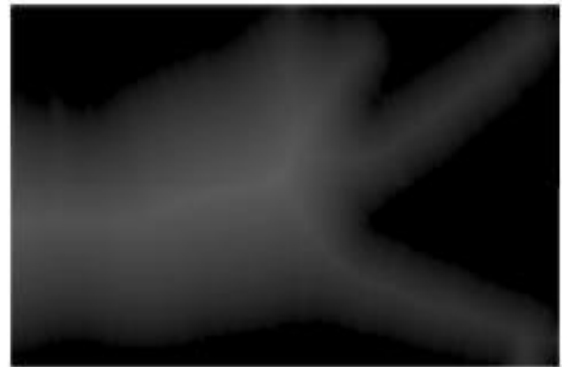


FIG 29. Distance Transform of the image

Case 5:

If a similar gesture as the query image is not present in the database then a nomatch is returned.

If a similar gesture as the query image is not present in the database then a nomatch is returned.

Case 6:

Cases of False Positives. The right gesture is returned but not logical.

If a similar gesture as the query image is not present in the database then a nomatch is returned.

CHAPTER 5

CONCLUSION AND FUTURE WORK

CONCLUSION

In this system, different effective techniques for man-machine interaction are observed. A few systems for preprocessing of input image are presented. This project also introduced gesture detection in the hand which increases the freedom of usability. These methods are going to use in different applications in future.

FUTURE ENHANCEMENT

This hand gesture recognition system is very vast in day-today technical solutions. Its main applications are

1. For controlling the traffic signals as per the wish of traffic controller to reduce the problem of traffic jam at peak hours.
2. To control the mouse so that physically handicapped people can use it and also it mouse will not be inefficient to use while travelling.

APPLICATIONS:

- Hand gesture controlled robot for physically challenged.
- Hand gesture controlled doors and vehicles.
- Hand gesture controlled keyboard and mouse to interact with computer.
- Gesture controlled appliances like air conditioner.
- Switching Channels, Without A TV Remote
- Driving To Safety
- To enable interaction of very young children with computers
- Development of forensic identification techniques
- For the hearing-impaired aids are developed
- Virtual environments manipulation and navigation
- Sign language reorganization
- Distance learning/tele-teaching assistance;

- Continuously watching stress level of medical patients
 - Automated Homes
 - Lie detection
 - Video conference interaction
 - Observing vehicle drivers' readiness/languor levels, and so on Gestures are deficient and equivocal, so there is one-to-numerous calibrating from gestures to concepts and the other way around.
-
- **Medical Applications** – Advanced robotics systems with gesture recognition can be placed in hospitals or homes to recognize and treat life threatening conditions like heart attacks or strokes.
 - **Alternative computer interfaces** – Gesture recognition, along with voice recognition, facial recognition, lip movement recognition and eye tracking combined can be used to create something called a perceptual user interface (PUI), a completely different way to interact with computer systems which will improve usability and creativity by leaps and bounds.
 - **Entertainment applications** – Most videogames today are played either on game consoles, arcade units or PCs, and all require a combination of input devices. Gesture recognition can be used to truly immerse a players in the game world like never before.
 - **Automation systems** – In homes, offices, transport vehicles and more, gesture recognition can be incorporated to greatly increase usability and reduce the resources necessary to create primary or secondary input systems like remote controls, car entertainment systems with buttons or similar.
 - **An easier life for the disabled** – One of the biggest challenges faced today is providing separate and equally non cumbersome services to the differently abled and handicapped. While there are special provisions around the world, there's still huge room for improvement to bring all lives on equal footing. Gesture recognition technology can eliminate a lot of manual labor and make life much easier for those who aren't as fortunate as most of us are.

FUTURE ENHANCEMENT

1. Without any dark background hand gesture will be detected from the image.
2. More sophisticated ways of Gesture recognition from various other human actions will take place instead of just hand gestures.
3. Voice recognition system will be coupled with gesture recognition system which will then completely remove the requirement of hardware like Key-board and Mouse.

REFERENCES

- [1]. Hand gesture recognition system by Prof. Praveen D. Hasalkar¹, Rohit S. Chougule², Vrushabh B. Madake³, Vishal S. Magdum⁴ ,2015.
- [2]. Vision Based Hand Gesture Recognition for Human Computer Interaction by Radhika Bhatt, Nikita Fernandes ,ArchanaDhage, 2013.
- [3]. Implentation of Hand Gesture Recognition Technique for HCI Using OpenCVNayanaP ,SanjeevKubakaddi 2014
- [4]. REAL TIME HAND GESTURE RECOGNITIONSYSTEM FOR DYNAMIC APPLICATIONS. Siddharth S. Rautaray¹, Anupam Agrawal²
- [5]. Gesture recognition -A Review by miss kawadeSonam P¹, Prof V.S. Ubale.
- [6] W. Z. Chen, "Real-time palm tracking and hand gesture estimation based on fore-arm contour," M. S. Thesis, Dept. Inform. Eng.,National Taiwan University of Science and Technology, 2011.
- [7] Kim, S., S. Takahashi and J. Tanaka, 2010. New interface using palm and fingertip without marker for ubiquitous environment. Proceedings of the 9th International Conference on Computer and Information Science, Aug. 18-20, IEEE Xplore Press, Yamagata, pp: 819-824. DOI: 10.1109/ICIS.2010.110
- [8] MeenakshiPanwar, "Hand Gesture Recognition based on Shape Parameters".
- [9] Thresholding<http://www.cse.unr.edu/~bebis/CS791E/Notes/Thresholding.pdf>

APPENDICES

A. SOURCE CODE

```
import cv2
import numpy as np
import math
cap = cv2.VideoCapture(0)
while (1):
    try: # an error comes if it does not find anything in window as it cannot find contour of
max area
    # therefore this try error statement
    ret, frame = cap.read()
    frame = cv2.flip(frame, 1)
    kernel = np.ones((3, 3), np.uint8)
    # define region of interest
    roi = frame[100:300, 100:300]
    cv2.rectangle(frame, (100, 100), (300, 300), (0, 255, 0), 0)
    hsv = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)
    # define range of skin color in HSV
    lower_skin = np.array([0, 20, 70], dtype=np.uint8)
    upper_skin = np.array([20, 255, 255], dtype=np.uint8)
    # extract skin colour image
    mask = cv2.inRange(hsv, lower_skin, upper_skin)
    # extrapolate the hand to fill dark spots within
    mask = cv2.dilate(mask, kernel, iterations=4)
    # blur the image
    mask = cv2.GaussianBlur(mask, (5, 5), 100)
    # find contours
    _, contours, hierarchy = cv2.findContours(mask, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
    # find contour of max area(hand)
```

```

cnt = max(contours, key=lambda x: cv2.contourArea(x))
# approx the contour a little
epsilon = 0.0005 * cv2.arcLength(cnt, True)
approx = cv2.approxPolyDP(cnt, epsilon, True)
# make convex hull around hand
hull = cv2.convexHull(cnt)
# define area of hull and area of hand
areahull = cv2.contourArea(hull)
areacnt = cv2.contourArea(cnt)
# find the percentage of area not covered by hand in convex hull
arearatio = ((areahull - areacnt) / areacnt) * 100
# find the defects in convex hull with respect to hand
hull = cv2.convexHull(approx, returnPoints=False)
defects = cv2.convexityDefects(approx, hull)
# l = no. of defects
l = 0
# code for finding no. of defects due to fingers
for i in range(defects.shape[0]):
    s, e, f, d = defects[i, 0]
    start = tuple(approx[s][0])
    end = tuple(approx[e][0])
    far = tuple(approx[f][0])
    pt = (100, 180)
    # find length of all sides of triangle
    a = math.sqrt((end[0] - start[0]) ** 2 + (end[1] - start[1]) ** 2)
    b = math.sqrt((far[0] - start[0]) ** 2 + (far[1] - start[1]) ** 2)
    c = math.sqrt((end[0] - far[0]) ** 2 + (end[1] - far[1]) ** 2)
    s = (a + b + c) / 2
    ar = math.sqrt(s * (s - a) * (s - b) * (s - c))
    # distance between point and convex hull

```

```

d = (2 * ar) / a
# apply cosine rule here
angle = math.acos((b ** 2 + c ** 2 - a ** 2) / (2 * b * c)) * 57
# ignore angles > 90 and ignore points very close to convex hull(they generally come
due to noise)
if angle <= 90 and d > 30:
    l += 1
    cv2.circle(roi, far, 3, [255, 0, 0], -1)
    # draw lines around hand
    cv2.line(roi, start, end, [0, 255, 0], 2)
    l += 1
    # print corresponding gestures which are in their ranges
    font = cv2.FONT_HERSHEY_SIMPLEX
    if l == 1:
        if areacnt < 2000:
            cv2.putText(frame, 'Put hand in the box', (0, 50), font, 2, (0, 0, 255), 3, cv2.LINE_AA)
        else:
            if arearatio < 12:
                cv2.putText(frame, '0', (0, 50), font, 2, (0, 0, 255), 3, cv2.LINE_AA)
            elif arearatio < 17.5:
                cv2.putText(frame, 'Best of luck', (0, 50), font, 2, (0, 0, 255), 3, cv2.LINE_AA)
            else:
                cv2.putText(frame, 'HUNGRY', (0, 50), font, 2, (0, 0, 255), 3, cv2.LINE_AA)
        elif l == 2:
            cv2.putText(frame, 'HELP', (0, 50), font, 2, (0, 0, 255), 3, cv2.LINE_AA)
        elif l == 3:
            if arearatio < 27:
                cv2.putText(frame, 'HOW ARE YOU', (0, 50), font, 2, (0, 0, 255), 3, cv2.LINE_AA)
            else:
                cv2.putText(frame, 'ok', (0, 50), font, 2, (0, 0, 255), 3, cv2.LINE_AA)
        elif l == 4:

```

```

cv2.putText(frame, 'THANK YOU', (0, 50), font, 2, (0, 0, 255), 3, cv2.LINE_AA)
elif l == 5:
cv2.putText(frame, 'HI', (0, 50), font, 2, (0, 0, 255), 3, cv2.LINE_AA)
elif l == 6:
cv2.putText(frame, 'reposition', (0, 50), font, 2, (0, 0, 255), 3, cv2.LINE_AA)
else:
cv2.putText(frame, 'reposition', (10, 50), font, 2, (0, 0, 255), 3, cv2.LINE_AA)
# show the windows
cv2.imshow('mask', mask)
cv2.imshow('frame', frame)
except:
pass
k = cv2.waitKey(5) & 0xFF
if k == 27:
break
cv2.destroyAllWindows()
cap.release()

```

B. SCREENSHOTS

