GESTURE RECOGNITION AND COMMUNICATION

Thesis submitted to
Visvesvaraya National Institute of Technology - Nagpur
In partial fulfillment of requirement for the Award of degree of

Bachelor of Technology (Computer Science and Engineering)

By

Amanchi Girish Kumar
BT13CSE007

Boddupally Venkatesh
BT13CSE019

Chereddy Rajanikanth Reddy
BT13CSE024

Gnyan Prakash
BT13CSE036

Kakne Prateek
BT13CSE048

Mervej Raj
BT13CSE053

Pradyanand Milind Pohare
BT13CSE065

Sai Kiran Rathod
BT13CSE074

Under the guidance of
DR. POONAM SHARMA

Department of Computer Science and Engineering
Visvesvaraya National Institute of Technology
Nagpur 440 010 (India)
May 2017

CERTIFICATE

The thesis titled "**GESTURE RECOGNITION AND COMMUNICATION**" submitted by

Amanchi Girish Kumar BT13CSE007	**Boddupally Venkatesh** BT13CSE019
Chereddy Rajanikanth Reddy BT13CSE024	**Gnyan Prakash** BT13CSE036
Kakne Prateek BT13CSE048	**Mervej Raj** BT13CSE053
Pradyanand Milind Pohare BT13CSE065	**Sai Kiran Rathod** BT13CSE074

For the award of degree of Bachelor of Technology in "Computer Science and Engineering", has been carried out under my supervision at the Department of Computer Science and Engineering of Visvesvaraya National Institute of Technology, Nagpur. The work is comprehensive, complete and fit for evaluation.

DR. POONAM SHARMA
Assistant Professor,
Department of Computer Science
and Engineering,
VNIT - Nagpur.

DR. U.A.DESHPANDE
Head of Department,
Department of Computer Science
and Engineering,
VNIT - Nagpur.

Declaration

I, hereby declare that the thesis titled “**GESTURE RECOGNITION AND COMMUNICATION”** submitted herein has been carried out by us in the Department of Computer Science and Engineering of Visvesvaraya National Institute of Technology, Nagpur. This written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

AMANCHI GIRISH	**BT13CSE007**	______________________
BODDUPALLY VENKATESH	**BT13CSE019**	______________________
RAJINIKANTH CHEREDDY	**BT13CSE024**	______________________
SULUGANTI GNYAN PRAKASH	**BT13CSE036**	______________________
KAKNE PRATEEK	**BT13CSE048**	______________________
MERVEJ RAJ	**BT13CSE053**	______________________
PRADNYANAND	**BT13CSE065**	______________________
RATHOD SAI KIRAN	**BT13CSE074**	______________________

ACKNOWLEDGEMENT

We would sincerely like to thank our project guide Dr. Poonam Sharma, Professor, Department of Computer Science and Engineering for her immense help and valuable guidance. We are extremely grateful for her insights, resourcefulness, and timely support and for his alacrity to help us whenever we needed help. It was her constant motivation that helped us complete this journey. We have gained immense knowledge under her patronage and are obliged to her for providing not only her valuable guidance, but also the necessary facilities provided to complete the project.

We take this opportunity to thank our Head of Department Dr.U.A.Deshpande for his support and for making labs available for our project experimentation. The resources provided by VNIT, Nagpur were an asset and made sure we met all our needs.

We would like to thank all the administrative and technical staff of the department for who have advised and helped us in their respective roles. We would like to thank all those who have directly or indirectly contributed to the completion of the project. Their role and support was of vital importance.

Lastly, we would like to thank the lab assistants and non-technical staff for their support and making sure we have access to the lab whenever required.

ABSTRACT

Communication between deaf-dumb people with hearing especially with their care-takers is a challenging task.About 350 million people in the world,according to a survey,are either deaf or dumb or both which is to be quite a large number to be ignored.Vocally challenged people use gestures to convey any message to his/her care-taker.But what if the latter is not with him/her ?There has to be a support system to solve this issue. This system should be very easily available, can be carried from place to place easily. So, our project, Gesture Recognition and Communication tool solves this issue by capturing gestures of that particular person and translates it into a predefined message which will be sent as a push-notification or a text message to the care-takers mobile.This system consists of a personal computer with a web-camera which captures the gestures of that person.This computer takes gesture as the input ,processes it and converts it into the corresponding text message which will be sent to the server.From server this message is directly sent as a notification or text message to the application present in the care-takers mobile.The care-taker and that particular deaf-dumb have a mutual understanding as to what gesture means what. These predefined gesture-messages are flexible and can be changed from time-to-time and person-to-person.The care-taker looks at the message received and acts accordingly. So basically this system needs a personal computer, a server and a mobile.

Table of Contents

Chapter 1

Introduction

1.1 Idea behind the project

The main idea behind this project is to provide an aid to people who are deaf and dumb. According to a survey , it is known that nearly 5% of the world's population are either deaf or dumb. They find it difficult to communicate with others especially their care takers.

When his/her care-taker is not with him , he cannot convey any message to the care-taker. To solve this issue, we thought of building a support sytem which does this task easily.

The system should not be difficult to understand. Anyone should understand the system easily. A lay-man should be able to convey his message to his care-taker without any knowledge on how it actually works.

So the person would just show any predefined sign to this system and the system takes this sign as input , processes it and finds out what the sign actually means (signs are priorily stored in data base) and sends this message to the caretaker.

Sign language is a language through which communication is possible without the means of sound. This can involve simultaneously combining hand shapes, orientation and movement of the hands, arms or body and facial expressions to fluidly express a speaker's thoughts .It's a visual language. It is sixth most spoken language in the world.

Sign language plays the major role for deaf and dumb people to communicate with others especially with their care-takers. It is a boon for them in the area of communication. So, when that person wants something from his care-taker and the latter is not present there with him , there should be some easy way of communication.

So, the care-taker and that person will define few signs priorily as to which sign means what.

For example:

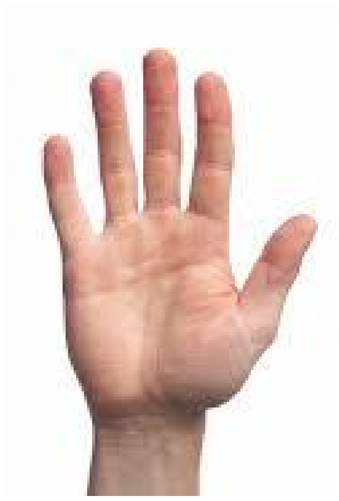

"thirsty"

Fig 1.1

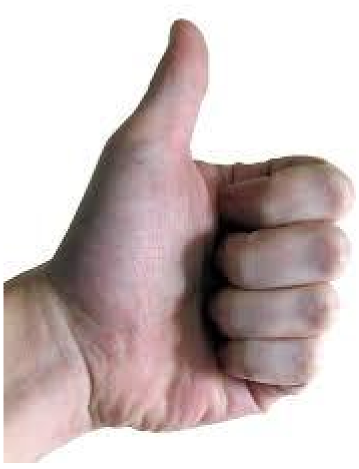

"hungry"

Fig 1.2

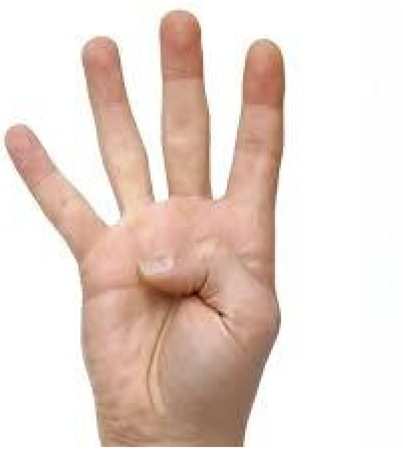
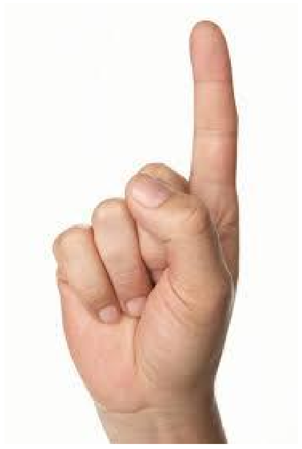
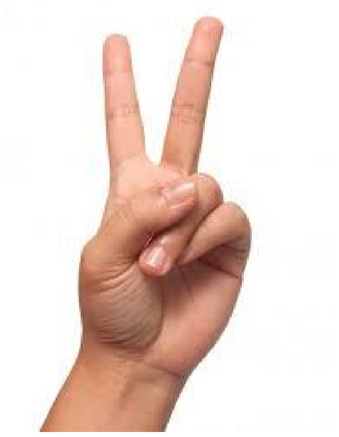

"tablets"

Fig 1.3



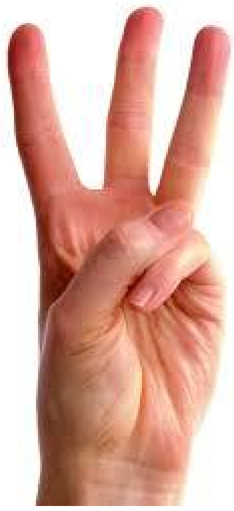
“need a walk”

Fig 1.4



“bring eggs”

Fig 1.5



“emergency”

Fig 1.6

This project aims at delivering a solution to problem mentioned above by developing sign language translator. The objectives of the solution are to-

(1) Develop a wearable sign language translator which is easy to handle and carry anywhere.

(2) Provide medium of communication between hearing individual and deaf, through mobile application.

In today’s world, due to easily available laptops and android-based mobiles with a good internet connectivity, their uniqueness and ease of usage, gesture recognition and communication is one of the most reliable and cost efficient systems by which life of deaf people can be made easier.

1.2 Basic overview of the system

The system comprises of three major parts:-

(1) Image capturing unit.
(2) Image to message conversion unit.
(3) Message display unit.

Image capturing unit has a personal computer , either desktop or laptop. It should have a web camera installed to capture the gesture. The camera captures the gesture thrown by the person and sends it to the message conversion unit.’

Message conversion unit basically has the entire software installed which processes the gesture and finds out what message is defined to that gesture and sends this output to the message display unit.

Message display unit takes the message and displays as a push notification or a text message to the care-taker.

Image capturing unit and message display unit form the hardware part. Message conversion unit forms the software part. We also need an intermediate server which takes message from the conversion unit and hands it over to the message display unit(mobile application).

So , briefly we need to have a personal computer , a server and a mobile.

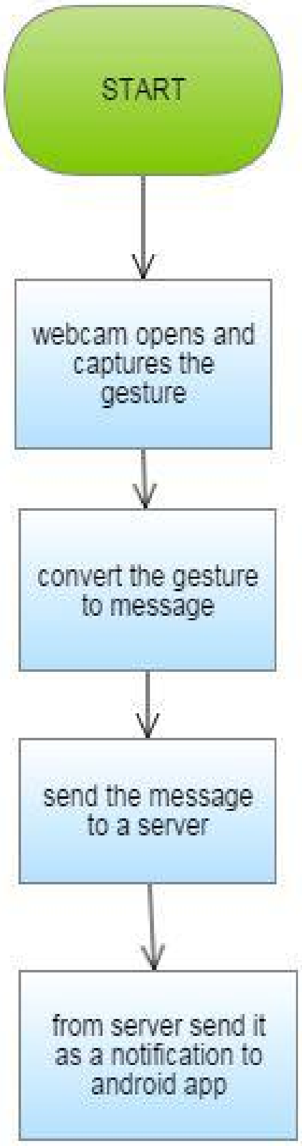

START
webcam opens and captures the gesture
convert the gesture to message
send the message to a server
from server send it as a notification to android app

Fig 1.7:Flow chart

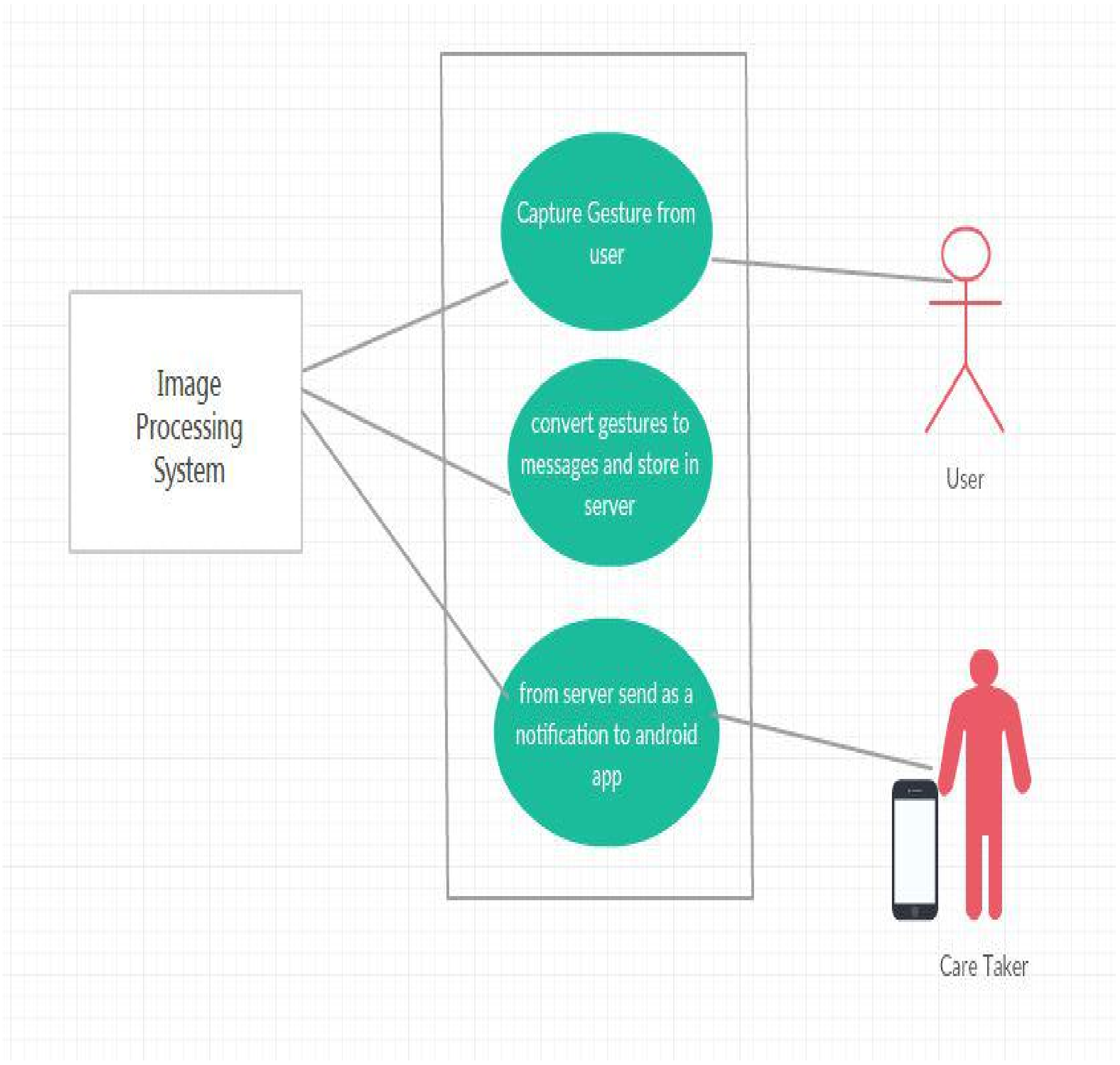


Image Processing System
Capture Gesture from user
convert gestures to messages and store in server
from server send as a notification to android app
User
Care Taker

Fig 1.8 : Use-case diagram

Chapter 2

Literature Survey

2.1 Introduction

Gesture recognition is a communication method between deaf-dumb person and his/her care-taker, using gestures. It serves as the kind of "emergency language" for the dumb. Gesture recognition uses units of different forms to interpret the words. These units comprise the following two basic things:

1. Hand shape.
2. Background (white is preferred).

2.2 Translation Technology- A tool to overcome communication barrier

For communication between care-taker and deaf-dumb, some medium must be there. They individual needs an efficient means of expressing and interpreting information in order to communicate and sign language has proven effective in providing that. . Gesture recognition and notification tool will be a boon for deaf and dumb.This tool will help them to notify messages to their care-takers if the latter are not with the former.

The primary condition is that this tool should be easily understood and easily carryable.

2.3 Previous Research work relevant to the Project

In the recent years, there has been remarkable research on the gesture recognition.

Vision based:-

Computer vision is an interdisciplinary field that deals with how computers can be made for gaining high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do.

Computer vision tasks include acquiring, processing, analyzing and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information, *e.g.*, in the forms of decisions. Understanding in this context means the transformation of visual images (the input of the retina) into descriptions of the world that can interface with other thought processes and elicit appropriate action. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory.

Image processing based:-

In imaging science, **image processing** is processing of images using mathematical operations by using any form of signal processing for which the input is an image, a series of images, or a video, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve isolating the individual color planes of an image and treating them as two-dimensional signal and applying standard signal-processing techniques to them. Images are also processed as three-dimensional signals with the third-dimension being time or the z-axis.

Image processing usually refers to digital image processing, but optical and analog image processing also are possible. This article is about general techniques that apply to all of them. The *acquisition* of images (producing the input image in the first place) is referred to as imaging.

2.4 Similar projects

A number of projects have been done in the field of sign language detection and translation. Many pc games such as “Genpla” use this technology.

Fig 2.1: Genpla game which uses gesture recognition

Sixth sense technology:

Even the famous sixth sense technogy is based on gesture recognition. SixthSense is a gesture-based wearable computer system developed at MIT Media Lab. It comprises a headworn or neck-worn pendant that contains both a data projector and camera. Headworn versions were built at MIT Media Lab in 1997 (by Mann) that combined cameras and illumination systems for interactive photographic art, and also included gesture recognition (e.g. finger-tracking using colored tape on the fingers).

The SixthSense prototype is comprised of a pocket projector, a mirror and a camera. The hardware components are coupled in a pendant like mobile wearable device. Both the projector and the camera are connected to the mobile computing device in the user's pocket. The projector projects visual information enabling surfaces, walls and physical objects around us to be used as interfaces; while the camera recognizes and tracks user's hand gestures and physical objects using computer-vision based techniques. The software program processes the video stream data captured by the camera and tracks the locations of the colored markers (visual tracking fiducials) at the tip of the user's fingers using simple computer-vision techniques. The movements and arrangements of these fiducials are interpreted into gestures that act as interaction instructions for the projected application interfaces. The maximum number of tracked fingers is only constrained by the number of unique fiducials, thus SixthSense also supports multi-touch and multi-user interaction.

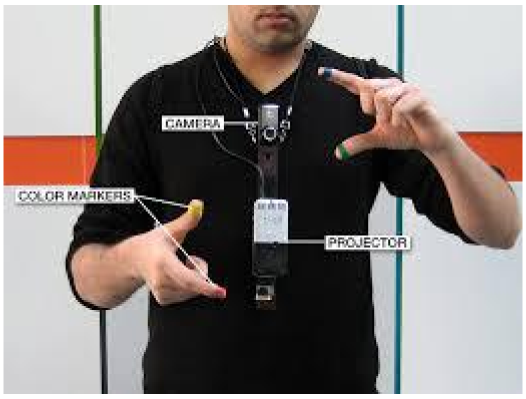


Fig 2.2: Sixth sense technology

Fig 2.3: Clicking pictures using gestures

Other real life projects include:

2.4.1 MyVoice

MyVoice prototype is developed by students of University of Houston Texas which translates spoken words in sign language and sign language into spoken words.
The MyVoice prototype concept is based on a handheld device with a built-in microphone, speaker, soundboard, video camera, and screen.

This device processes sign language motions and can translate them using specific algorithms into an electronic voice.

The device can be placed on a hard surface, where it reads a user's sign language movements. After processing motion, sign language is translated into space through electronic device.

2.4.2 EnableTalk glove

In 2003 Microsoft Imagine Cup, Quadsquad team of Ukraine set out to develop EnableTalk gloves that translate sign language into speech in real time. The basic design steps are based on:
The gloves themselves are fitted with an accelerometer, gyroscope, compass, and 15 flex sensors along the fingers, thumb and palm that determine the position of the glove in space.

Data from the sensors is relayed to a controller on the back of the glove that is powered by a rechargeable battery. The battery can be recharged via USB, while a small solar panel also helps extend the intervals between charging.

Data from the glove is then transmitted via Bluetooth to a mobile device that translates the signs into speech using the Microsoft Speech API and Bing API.

2.4.3 Google Gestures

The concept of Google Gesture is a way to enable signed conversation with those who don't use sign language. It is a service that translates sign language into speech.

It analyses the position and muscle activity both in hand and forearm of signer who wears two electronic wristbands.

The band sends obtained information to google gesture application which translates sign to speech in real time.

Chapter-3

Overview of Hardware

In our project of communication using gestures , we need a personal computer (laptop having a good web camera or a desktop having web camera installed to it) . A mobile operating on android operating system which should have enough space to get our app installed in it.

<u>Personal Computer</u>:-

- To capture the real time hand gesture .
- The processing of this gesture (conversion to message) is done in this computer where code is written already.
- The personal computer should have a good camera and internet connectivity.
- Camera, to capture the hand gesture for creating a response.
- Internet connectivity (wired or wireless) to send the response to a server.
-

    - PC: Windows 7, Windows 8, Windows 10 with (IE 11*, Chrome 43+, Firefox 38+, and Edge)

Fig 3.1: Laptop having a web cam

Mobile:-

- A mobile that operates on android operating system and enough space to install the application.
- The response that was sent to server is again sent to mobile application.
- The care-taker possesses to the mobile whom the response is notified through that application.
    - Android devices with version 4.4+.

12:05 MON, MARCH 17
Internet up
12:05 PM
HOME

Mar 17, 2014 12:05:31 PM: No issues detec..

Fig 3.2:Push notification in a mobile

Internet Connectivity:-

- Both pc and mobile should have an internet connectivity.
- Internet in personal computer connects that system to the server.
- Internet in mobile connects the mobile to the server.

Chapter 4

Working principle

4.1 Introduction

The basic idea of the project is to make a simple application which recognizes a set of predetermined hand gestures using the combination of inputs from image capturing devices to detect the corresponding input gesture. The detected gestures are then processed to notify the Base station i.e a receiver.

4.2 Working of the system

Hand gesture recognition application's explanation starts with knowing finger and hand position as shown in figure 1.1 in chapter

As explained earlier, this application system comprises of

(1) Gesture Detection Unit

(2) Digital Output Unit

4.2.1 Gesture Detection Unit

The detection of an input gesture requires a stable background and a clear input. This has been achieved using an image capturing device(in our case a camera) . The input thus obtained is fed to the application which is pre-processed and then compared metadata of earlier determined gestures. The application logic helps detect input gesture.

4.2.2 Digital Output Unit

Based on the input gesture detected by the Gesture detection unit, corresponding notification for the gesture is sent to the base station receiver application(in our case hand-held mobile device application) to notify the concerned entity. This is done by establishing a secure connection between the gesture detection application server and the output unit using token based approach. This is further explained in more detail in later sections.

Chapter 5

Software Implementation

5.1 OpenCV

5.1.1 Introduction

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer applications. Officially launched in 1999, the OpenCV project was initially an Intel Research initiative to advance CPU-intensive applications. OpenCV supports a wide variety of programming languages such as C++, Python, Java, etc., and is available on different operating platforms like Windows, Linux, OS X, Android and iOS.

- **OpenCV-Python** - OpenCV-Python is the Python API for OpenCV, combining the best of the both of OpenCV C++ API and the Python language. Python is a general purpose programming language that became very popular very quickly, mainly because of its simplicity and code readability though it is slower compared to languages like C/C++. Python can be easily extended with C/C++, which allows us to write computationally intensive code in C/C++ and create Python wrappers that are generally used as Python modules. This gives us two advantages: first, the code is as fast as the original C/C++ code (since it is the actual C++ code working in background) and second, it easier to code in Python than C/C++. OpenCV-Python is a Python wrapper for the original OpenCV C++ implementation.

- **OpenCV-NumPy** - NumPy is the fundamental package for scientific computing with Python . It is a highly sprcified library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays

.

5.1.2 Installation

- **OpenCV** - The following steps were followed for the installation of OpenCV.

  - In order to install OpenCV the following packages must be downloaded and installed to their locations -

    1. Python-2.7.x.
    2. Numpy.

  - After installation of the dependencies Python IDLE is opened and “import numpy“ is entered to make sure Numpy is working fine.

  - Latest OpenCV release is downloaded from sourceforge site and extracted.

  - In **opencv/build/python/2.7** folder, **cv2.pyd** is copied to **C:/Python27/lib/site-packages**.

  - In Python IDLE the following codes are typed in Python terminal.

```
>>> import cv2
>>> print cv2.__version__
```

If the results are printed out without any errors then OpenCV-Python is installed successfully.

- **Numpy** – The best way to install Numpy is to by using an pre-built package for the specific operating system.

5.1.3 Features

Some of the features of OpenCV that are used in this project are –

- **Basic Image Operations** – The various operations that are related to image operations are –

  - ➢ Access pixel values and modify them – After loading an color image we can access a pixel value by its row and column coordinates. For BGR image, it returns an array of Blue, Green, Red values and for grayscale image, just corresponding intensity is returned. An example for accessing and modifying pixel of an image –

```
# accessing RED value
>>> img.item(10,10,2

# modifying RED value
 >>> img.itemset((10,10,2),100)
>>> img.item(10,10,2)
```

  - ➢ Access image properties – The basic properties of an image include number of rows, columns and channels, type of image data, number of pixels etc.

Shape of image is accessed by img.shape. It returns a tuple of number of rows, columns and channels (if image is color):

```
>>> print img.shape
(342, 548, 3)
```

Total number of pixels is accessed by img.size:

```
>>> print img.size
562248
```

Image datatype is obtained by `img.dtype`:

```
>>> print img.dtype
uint8
```

- Setting Region of Image (ROI) –When we have to select certain region of the image to play with for example to detect an eye in images, first face detection is done all over the image and when face is obtained, we select the face region alone and search for eyes inside it instead of searching whole image.

```
>>> eye = img[280:340, 330:390]
>>> img[273:333, 100:160] = eye
```

- Splitting and Merging images - Sometimes it may happen that we may need to work separately on B,G,R channels of image. Then we need to split the BGR images to single planes or some another time,we may need to join these individual channels to BGR image. We can do these by simply

```
>>> b,g,r = cv2.split(img)
>>> img = cv2.merge((b,g,r))
```

Almost all the operations that are described above are mainly related to Numpy rather than just OpenCV. Thus, a good knowledge of Numpy along with openCV is required to write better optimized code with OpenCV.

- **Image Filtering** –The functions and classes described in this section are used to perform various linear or non-linear filtering operations on 2D images. The trick of image filtering is that we have a 2D filter matrix and the 2D image. Then for every pixel of the image, we take the sum of products. Each product is the color value of the current pixel or a neighbor of it, with the corresponding value of the filter matrix. The center of the filter matrix has to be multiplied with the current pixel, the other elements of the filter matrix with corresponding neighbor pixels.

  - **Bilateral Filter** – Bilateral Filter can reduce unwanted noise very well while keeping edges fairly sharp. However, it is very slow compared to most filters. The function code to apply bilateral code to an image is –
    cv2.bilateralFilter(src, d, sigmaColor, sigmaSpace[, dst[, borderType]]) → dst
    Here let us see what each parameter describes –

    - src – Source is a 8-bit or floating-point, 1-channel or 3-channel image.
    - dst – Destination of the image of the same size and type as src .
    - d – Diameter of each pixel in the neighborhood that is used during filtering. If it is non-positive then it is computed from sigmaSpace .
    - sigmaColor – Filter the given sigma in the color space. A larger value of the parameter usually means that farther colors within the pixel neighborhood will be mixed together, resulting in larger areas of semi-equal color.
    - sigmaSpace – Filter the given sigma in the coordinate space. A larger value of the parameter usually means that farther pixels will influence each other as long as their colors are close enough. When d>0 , it specifies the neighborhood size regardless of sigmaSpace . Otherwise d is proportional to sigmaSpace .

    Sigma values: For simplicity, we can set the 2 sigma values to be the same. If they are small (< 10), the filter will not have much effect, whereas if they are large (> 150), they will have a very strong effect, making the image look “cartoonish”.

    Filter size: Large filters (d > 5) are very slow, so it is recommended to use d=5 for real-time applications and perhaps d=9 for offline applications that need heavy noise filtering.

- **Contours -** Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition. There are two functions that are associated with contours –
  - cv2.findContours() – This function is used to retrieve contours from the binary image using the algorithm. The contours are a useful tool for shape analysis and object detection and recognition. Syntax of this function is –

    ```
    cv2.findContours(image,mode,method);
    ```

    Here the parameter describes –
    - image – it is the source image, an 8-bit single-channel image. Here non-zero pixels are treated as 1's while zero pixels remain 0's and so the image is treated as binary.
    - mode – it is an integer that indicates the contour retrieval mode.
    - method – it is an integer that indicates contour approximation method.

  - cv2.drawContours() –This function is used to draw the contours. It can also be used to draw any shape provided you have its boundary points. The syntax of this function is –

    ```
    cv2.drawContours(image,contours,contourIdx);
    ```

    Here the parameter describes –

    - image - it is the destination image.
    - contours - it is the contours which should be passed as a Python list.
    - contourIdx – it is the parameter indicating a contour to draw. If it is negative, all the contours are drawn.

- **Changing Colorspaces** – In this section we convert images from one color-space to another, like BGR ↔ Gray, BGR ↔ HSV etc. In addition to that, we will create an application which extracts a colored object in a video. Two functions that are used in for this feature are –

  - **cv2.cvtColor()** - This function converts an input image from one color space to another. The default color format in OpenCV is referred to as RGB but it is actually BGR. So the first byte in a standard (24-bit) color image will be an 8-bit Blue component, the second byte will be Green and the third byte will be Red. The syntax of this function is –
    cvtColor(src, dest, code);

    Here the parameter describes –

    - img1 – it is the input image that is 8-bit unsigned, 16-bit unsigned or single-precision floating-point.

    - Img2 – this is output image which is of the same size and depth as source image.

    - code – this is an integer that represents the color space conversion code.

  - **cv2.inRange()** - Checks if array elements lie between the elements of two other arrays. The function checks the range as follows:

    For every element of a single-channel input array:

    $\mathrm{dst}(I) = \mathrm{lowerb}(I)_0 \leq \mathrm{src}(I)_0 \leq \mathrm{upperb}(I)_0$

    For two-channel arrays:

$$dst(I)=lowerb(I)_0 \le src(I)_0 \le upperb(I)_0 \land lowerb(I)_1 \le src(I)_1 \le upperb(I)_1$$

When the lower or upper or both boundary parameters are scalars then the indexes (I) at lowerb and upperb in the above formulas are omitted.

The syntax for the function is –

```
cvtinRange(src, dest, flags);
```

Here the parameter describes –

- src – it is the input floating-point real or complex array.
- dest – it is the output array whose size and type depend on the flags.
- flags – it is an integer that defines the operation flags.

- **Convex Hull** – This functions is used to Find the convex hull of a point set i.e. it finds the convex hull of a 2D point set using the Sklansky's algorithm that has O(N logN) complexity in the current implementation. The syntax of the convex hull function is –

```
convenHull(input array, output array, bool);
```

Here the parameter describes –

- input array - Input array is a 2D point set usually stored in std::vector or Matrices.
- output array – This is the output convex hull. It is either an integer vector of indices or vector of points. In the first case, the hull elements are 0-based indices of the convex hull points in the original array (since the set of convex hull points is a subset of the original point set) and in the second case the hull elements are the convex hull points themselves.
- bool – It is the Boolean orientation flag i.e. if it is true the output convex hull is oriented clockwise. Otherwise it is oriented counter-clockwise. Here generally the

assumed coordinate system has its X axis pointing to the right and its Y axis pointing upwards.

Here is an example of how the convex hull works –

Initial image –

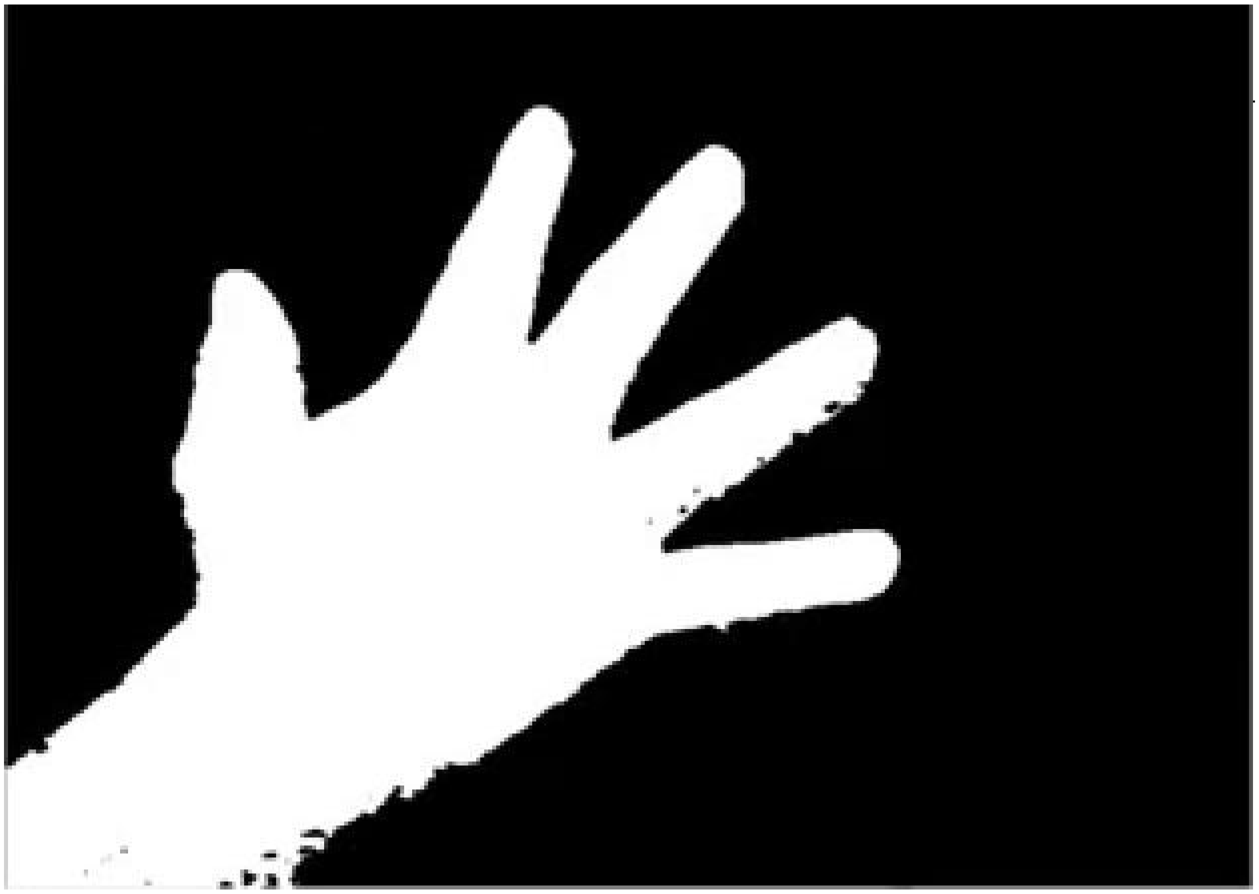


Fig 5.1 – Image before Convexhull function

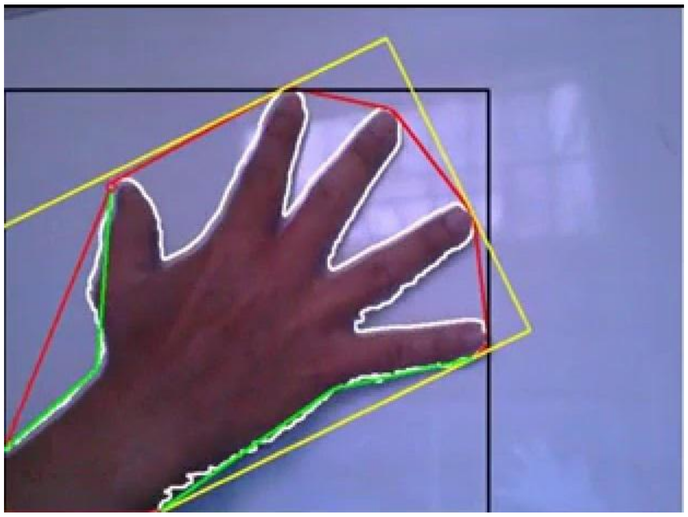


Fig 5.2 – Image after Convexhull function

5.2 Py Game Module

5.2.1 Introduction

Pygame Open Source python programming language library for making multimedia applications like games built on top of the excellent SDL library. Pygame is highly portable and runs almost on platforms and operating systems. Python and SDL work on multiple platforms. Pygame is fairly low-level when it comes to writing games but the great thing about this is there is nothing inside pygame to get in our way. Some of the features of Pygame module are-

- Does not require OpenGL - Pygame uses either opengl, directx, windib, X11, linux frame buffer and thus doesn't require OpenGL as OpenGL is often broken on linux as well as on windows.

- Multi core CPUs can be used easily – Pygame also supports in multicore CPUs as they are the basic requirement of every computer system now-a-days.

- Operational in almost all the operating system – Pygame is fully functional in almost all the operating system even the less popular ones.

- Does not require a GUI to use all functions - Pygame can be without a monitor – ths can be done if we want to use it just to process images, get joystick input or play sounds

.

- Small amount of code – Pygame has a relatively small amount of code as it uses optimized C and assembly code for core functions. As C code is often 10-20 times faster than python code and assembly code can easily be 100x or more times faster than python code.

5.2.2 Installation

Pygame requires Python. The best way to install Pygame is by pip tool which is available in almost all the recent version of python. And we then run the command with the --user flag to install into the home directory, rather than globally.

```
python3 -m pip install pygame --user
```

5.3 Python

5.3.1 Introduction

Python is a powerful programming language that has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing and its interpreted nature, make it an ideal language for scripting and rapid application development. Python features a dynamic type system and automatic memory management and supports multiple programming paradigms. Python also has a standard library that is very extensive, offering a wide range of facilities. The Python library contains built-in modules (written in C) that provide access to system functionality which would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

5.3.2 Installation

Python can be downloaded from the official python page https://www.python.org/download/. There are various software builds of python that are available for download. It is up to us which version to download depending on our requirement. Once python is downloaded installation is done as normal executable package.

- Adding Python to System Path Variable -

If the Python 3.4.1 version is used then we do not need to follow this process as the new update integrates this process in the installation phase and so we no longer need to manually add the System Path Variable. But if we want to add a second set of variables for Python, we can still follow the procedure but replace “27” with “34.” The stpeps are as follows –

1. In the start menu and typing in “environment” and the option called “Edit the system environment variables” is selected.

2. When the “System Properties” window appears, we click on “Environment Variables…”

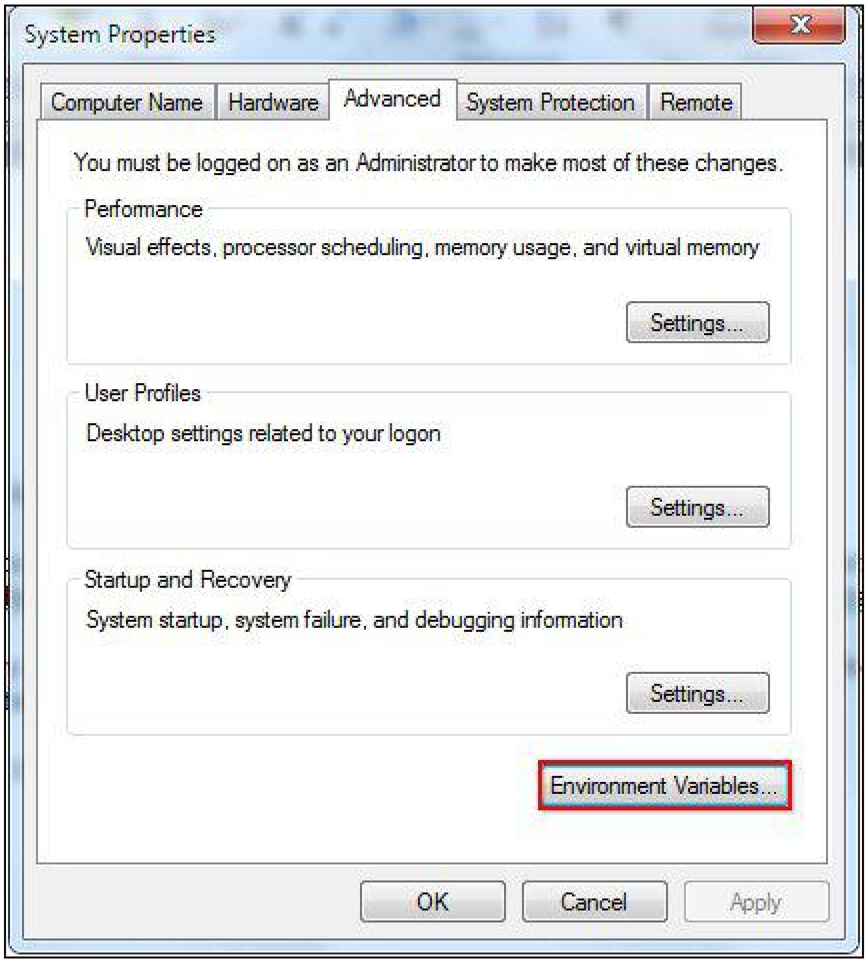
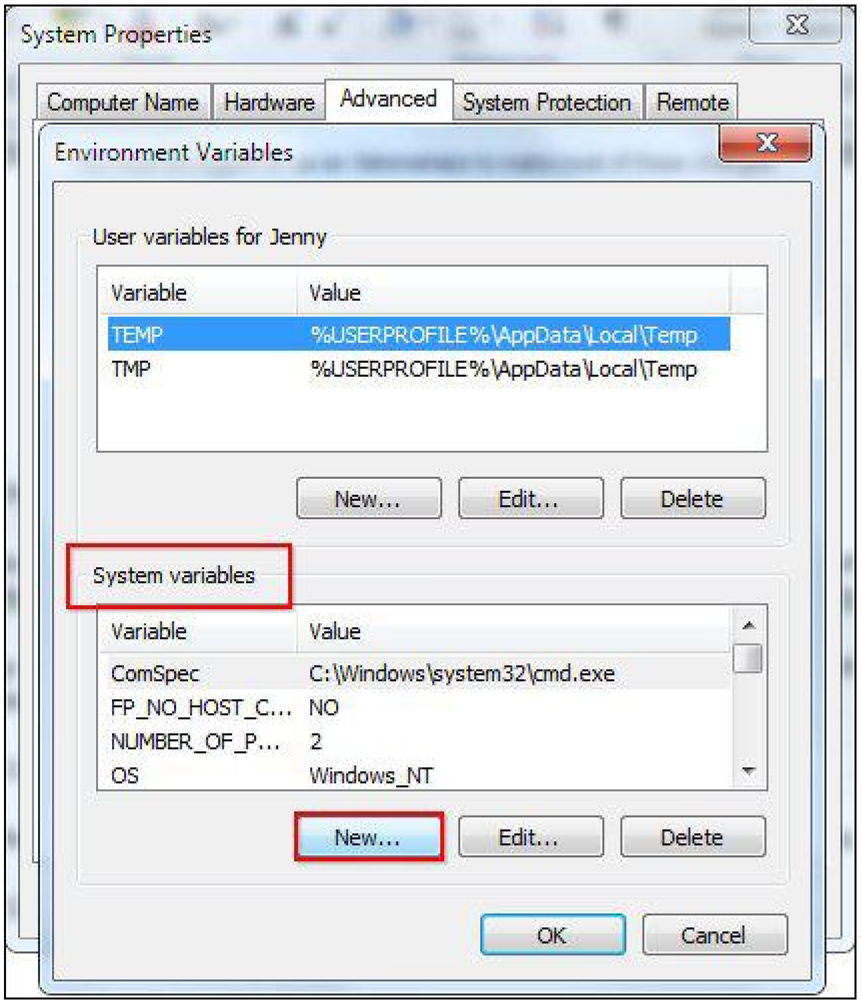

System Properties
Computer Name
Hardware
Advanced
System Protection
Remote
You must be logged on as an Administrator to make most of these changes.
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...
User Profiles
Desktop settings related to your logon
Settings...
Startup and Recovery
System startup, system failure, and debugging information
Settings...
Environment Variables...
OK
Cancel
Apply

Fig 5.3 - Image of System Properties dialog box

3. Once the “Environment Variables” window opens. We will notice that it controls all the “System Variables” rather than just this associated with our user. Now we click on “New…” to create a new variable for Python.


System Properties

Computer Name | Hardware | Advanced | System Protection | Remote

Environment Variables

User variables for Jenny

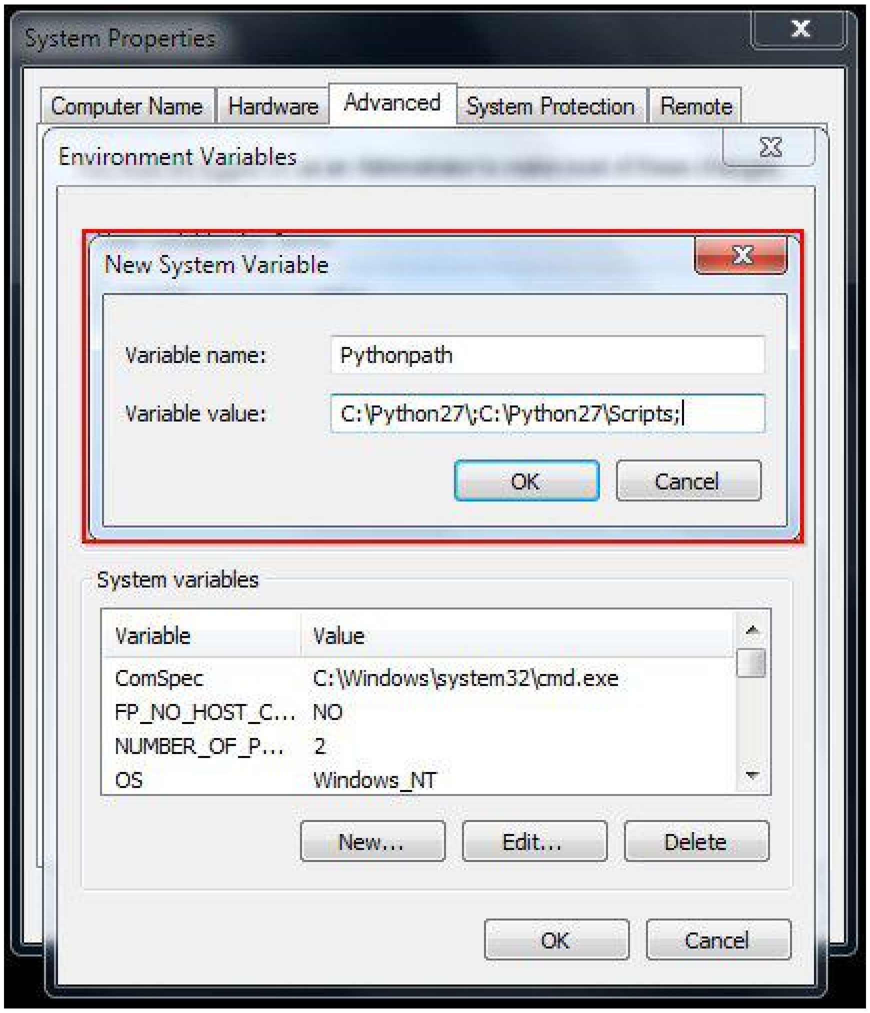
Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	2
OS	Windows_NT

Fig 5.4 - Image of Environment Variables dialog box

4. Now we simply enter a name for our Path and the code shown below. The string that we will need to enter is: “C:\Python27\;C:\Python27\Scripts;”



Variable name: Pythonpath

Variable value: C:\Python27\;C:\Python27\Scripts;

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	2
OS	Windows_NT

Fig 5.5 - Image of New System Variable dialog box

5. We then press "OK," then "OK," then "OK," then the red "X" to accept all changes and exit the "System Properties" window.

5.4 Android Application

An Android app is a software application running on the Android platform. Because Android platform is built for mobile devices, a typical Android app is designed for a smartphone or a tablet PC running on the Android OS. Android apps are written in the Java programming language also it uses Java core libraries. They are first compiled to Dalvik executables to run on the Dalvik virtual machine, which is a virtual machine specially designed for mobile devices.

5.4.1 Why Android Application?

We have to display the result which we get from openCv. So to display result we choose to develop an android application so that the result will come in form of push notification. To develop Android application there are many platform we can use like Android Studio,Ionic platform and Eclipse IDE. Here we choose Android Studio. because it is a open source software, its reduces cost of development and it has rich development environment. We can directly use OpenCv library in Android studio so it will be easy to link OpenCv to Android Application.

5.4.2 Introduction of Android Studio

Android Studio IDE from Google allows developers to build apps running on desktop, mobile and other emerging technologies running on Android also Android Studio is an integrated development environment (IDE) from Google that provides developers with tools needed to build applications for the Android OS platform.

5.4.3 Installation of Android Studio

Installation for Windows is simple.First of all download the android exe file from android studio website(i.e https://developer.android.com/studio/index.html) then just launch the .exe you downloaded and then click through the setup guide.

5.4.4 Creating Android Project

Android Studio makes it easy to create Android apps for various form factors, such as phone, tablet and Google Glass. The New Project wizard lets you choose the form factors for your app and populates the project structure with everything you need to get started.

Step 1: Start and configure the project

Step 2: Select form factors and API level

Step 3: Add an activity

Step 4: Configure the created activity

5.4.5 Implementation

To build an Android application first of all we have to create its User Interface means layout. A layout defines the visual structure for a user interface, such as the UI for an activity. After that we have to build a java package which will help the layout to perform its functionality.

5.4.6 Building The Android App Gradle

In Android Studio, Gradle is a custom build tool used to build android packages (apk files) by managing dependencies and providing custom build logic. We are creating Android Push Notification using Firebase Console so we have to add some plugins and dependencies in android gradle. First of all we add dependencies in project level gradle file i.e classpath ‘com.google.gms:google-services:3.0.0’. Now we have to add Firebase plugins to app so we added apply plugin: ‘com.google.gms.google-services'. For cloud messaging and notification we need Firebase libraries so we added compile ‘com.google.firebase:firebase-messaging:9.2.1' to build.gradle. We also added decencies volley volley i.e compile 'com.android.volley:volley:1.0.0' to send token from file server to application server.

5.4.7 Building AndroidManifest

The manifest file provides essential information about your app to the Android system, which the system must have before it can run any of the app's code.To add all the services of the firebase we build the AndroidManifest.xml. We added two services to AndroidManifest, first one is by name “.FcmInstanceIdService” which is for to get registered token, which has action by name android:name=“com.google.firbase.INSTANCE_ID_EVENT”.

Second service is “.FcmMessagingService” which is used to receive messages and notification from firebase, which has action as android:name=“com.google.firbase.MESSAGING_EVENT”.Also to send Fcm token we have give internet permission so we added the internet permission to AndroidManifest i.e <uses-permission android:name="android.permission.INTERNET"/>.

5.4.8 Creating Android Layout

A layout defines the visual structure for a user interface, such as the UI for an activity or app widget. You can declare a layout in two ways:

- Declare UI elements in XML - Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses.

- Instantiate layout elements at runtime - Application can create View and ViewGroup objects (and manipulate their properties) programmatically.

We here using first way i.e Declare UI elements in XML .For android application to show its result to user we have created two layout one is activity_main.xml and other is activity_notification.xml. When user will open the Android application then user will land on home page i.e activity_main.xml. On this layout there is button, when user will press this button then it will take user to another page i.e activity_notification.xml.
The result which came from OpenCv will be displayed on activity_notification.xml layout. In activity_notification.xml layout we have a interface TextView. A TextView displays text to the user and optionally allows them to edit it. A TextView is a complete text editor but the basic class is configured to not allow editing.In both layout we used RelativeLayout which is a type of View. The basic building block for user interface is a View object which is created from the View class and occupies a rectangular area on the screen. RelativeLayout is a view group that displays child views in relative positions.

5.4.9 Creating Java Classes

Android Studio can create the following new classes and types:

- Java classes
- Enumeration and singleton classes
- Interface and annotation types

With the Create New Class dialog and file templates, Android Studio helps you to quickly create new classes. In activity_main.xml file we have a button, so when we click the button to make it responding we create java file i.e MainActivity.java. We used here FindViewById, which will search button in the activity_main layout by button's id. To make button responding for any click by user we used here OnClickListner. OnClickListner makes it listen to the user's click. After that we assign that OnClickListener to that button using button.setOnClickListener(new View.OnClickListener().

When the user clicks the button, the onClick function of the assigned OnClickListener is called. When user clicks button we have to get the recent token so we use SharedPreferences. We get the recent token from SharedPreferences and then stored it to variable by name token. Now we have to create a string request so for that we have to first specify the app server url "http://10.0.2.2/fcmtest/fcm_insert.php". After that we created stringRequest so that we can add a string request to queue.

We have created FcmInstanceIdService.java class to receive the registered token, which extends using FirebaseInstanceIdService. To get recent registered token we need to override method by name onTokenRefresh(). In onTokenRefresh() method we use SharedPreferences to get shared token and then we saved it to the string by name recent_token. We have create MySingleton class for volley which is used to send token from file server to application server i.e to localhost.

In order to receive message and notification from firebase we created a java class by name FcmMessagingService.java to extends firebase messaging services. Our application received notification or message data in method called onMessageReceived(). Now we have to create our own notification so we created an intent variable, we set a flag for intent variable to clear all the stalk top.

Next we created a PendingIntent. A PendingIntent is a token that you give to a foreign application, which allows the foreign application to use your application's permissions so that you can execute a predefined piece of code or PendingIntent a description of an Intent and target action to perform with it. Then we created NotificationCompat.Builder which allows easier control over all the flags, as well as help constructing the typical notification layouts. We used NotificationManager which will tell the user that something has happened in the background.

5.4.10 Running The Android Application

To run the application you can use the Android Emulator. Android Emulator simulates device and displays result on your development computer. It lets you develop, and test Android apps without using a hardware device. The emulator supports Android phone, tablet, Android

Wear, and Android TV devices. Android Emulator comes with predefined device types so you can get started quickly, and you can create your own device definitions and emulator skins.

Before you run your app on an emulator, we need to create Android Virtual Device (AVD) definition. An AVD defines specifies the characteristics of an Android phone, tablet, Android Wear device that you want to simulate in the Android Emulator.

Steps To Create AVD:

1. Launch the Android Virtual Device Manager by selecting Tools > Android > AVD Manager.

2. In the Your Virtual Devices screen, click Create Virtual Device.

3. In the Select Hardware screen, select a phone device, such as Pixel, and then click Next.

4. In the System Image screen, click Download for one of the recommended system images. Agree to the terms to complete the download.

5. After the download is complete, select the system image from the list and click Next.

6. On the next screen, leave all the configuration settings as they are and click Finish.

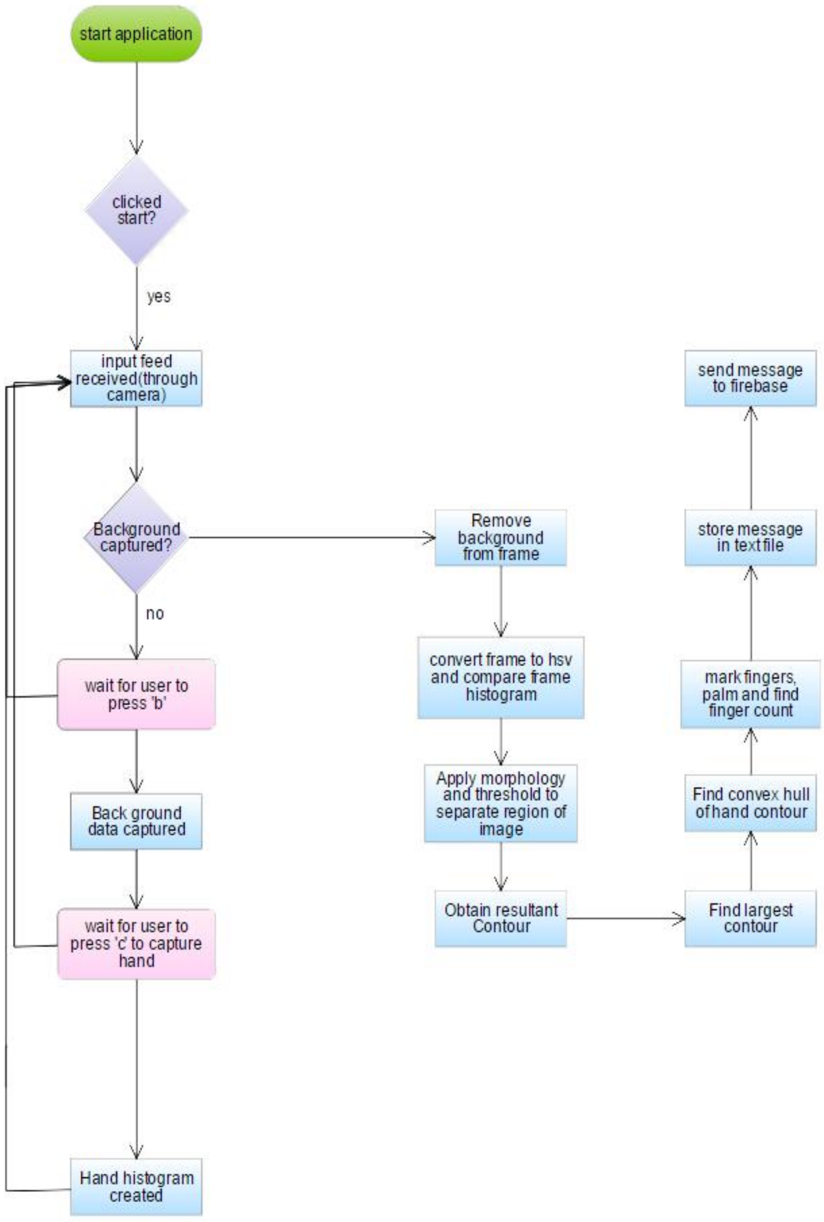
7. Back in the Your Virtual Devices screen, select the device you just created and click Launch this AVD in the emulator.

Steps To Start Application

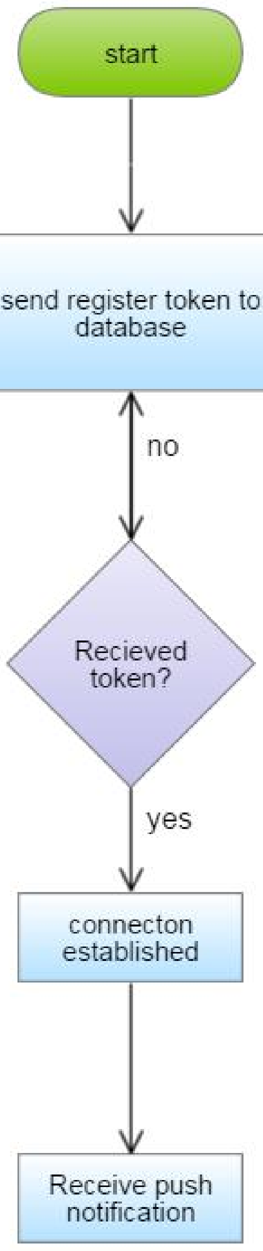
1. Once the emulator is booted up, click the app module in the Project window and then select Run > Run

2. In the Select Deployment Target window, select the emulator and click OK.

5.5 Flow Chart


start application
clicked start?
yes
input feed received(through camera)
Background captured?
no
wait for user to press 'b'
Back ground data captured
wait for user to press 'c' to capture hand
Hand histogram created
Remove background from frame
convert frame to hsv and compare frame histogram
Apply morphology and threshold to separate region of image
Obtain resultant Contour
Find largest contour
Find convex hull of hand contour
mark fingers, palm and find finger count
store message in text file
send message to firebase

CAPTURE AND DETECTION UNIT FLOWCHART



MESSAGE COMMUNICATION UNIT

5.6 Working principle

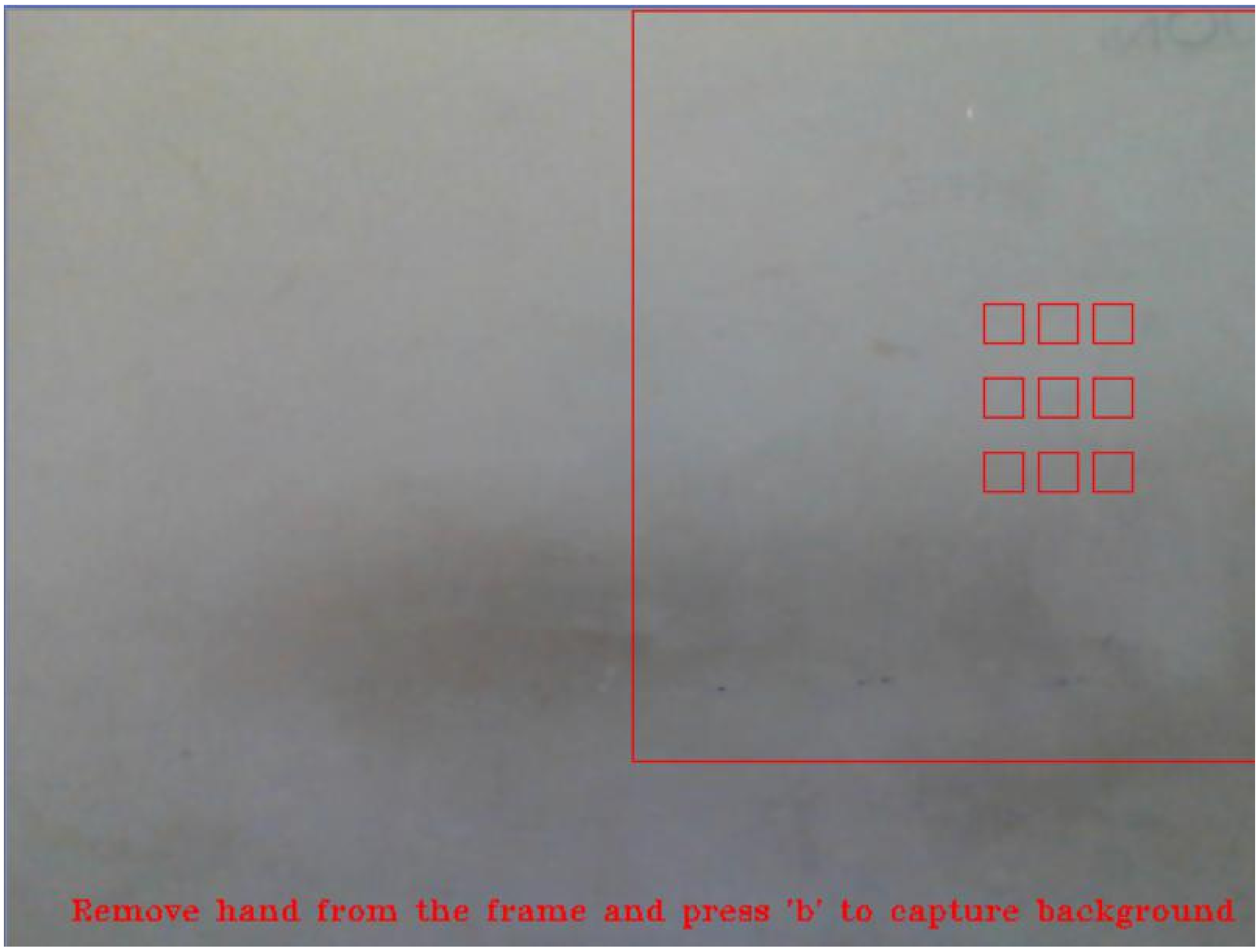
The application runs using simple techniques of video capturing, background extraction and push notifications. Running 'GestureRecognition.py' presents a welcome screen. On start it opens window that will act as the output frame. The camera input is currently being displayed on the screen. The input is captured using

```
input_camera = cv2.VideoCapture(0)

ret, frame = input_camera.read()
```

'frame' is a Numpy array. For better image quality and smoothing, bilateral filtering has been used because of its excellent edge-preserving and noise reducing qualities.

```
frame=cv2.bilateralFilter(frame,5,50,100)
```

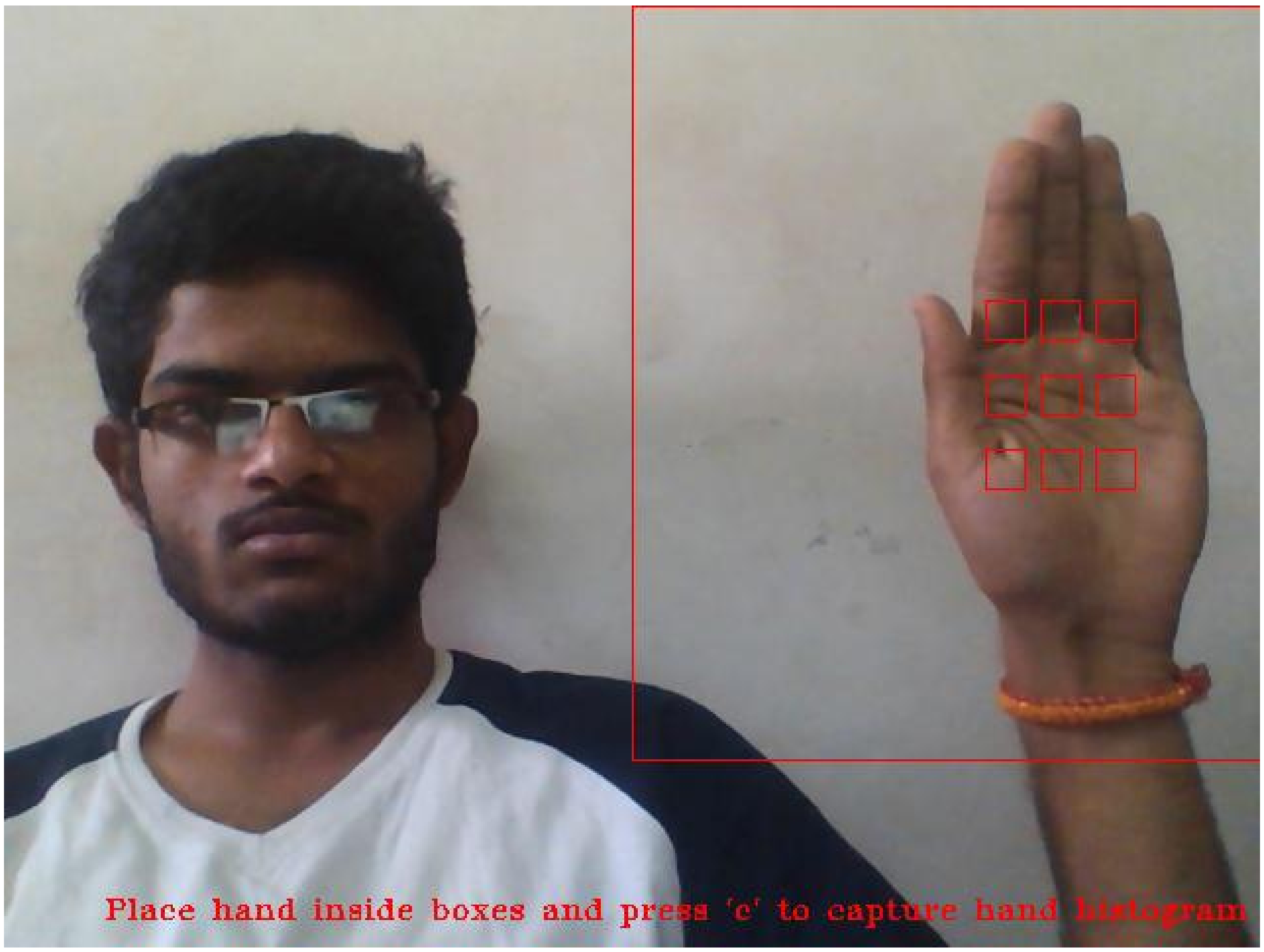

Remove hand from the frame and press 'b' to capture background

After filtering the frame background will be captured on pressing a key 'b'. This helps detect false stationary objects which are around same histogram range as the users hand. Precaution should be taken that while capturing the background users hand should not be present in the rectangular box that has been provided at the right corner of the frame which will be later used for capturing actual hand gesture. However actual hand gestures are obtained by subtracting background frame from these frames. The background model is obtained using the line.

```
bg_model = cv2.BackgroundSubtractorMOG2(0,10)
```

To capture different shades of the hand once the background model is created we use a set of 9 small boxes. The user should place his hand and cover all these 9 boxes without any shadows or air gaps for proper recognition. The user presses 'c' to capture hand histogram. It contains a histogram of set of arrays of shades. It is done on invocation of the user defined function hand_capture().

```
hand_capture(frame_in,box_x,box_y)
```


Place hand inside boxes and press 'c' to capture hand histogram

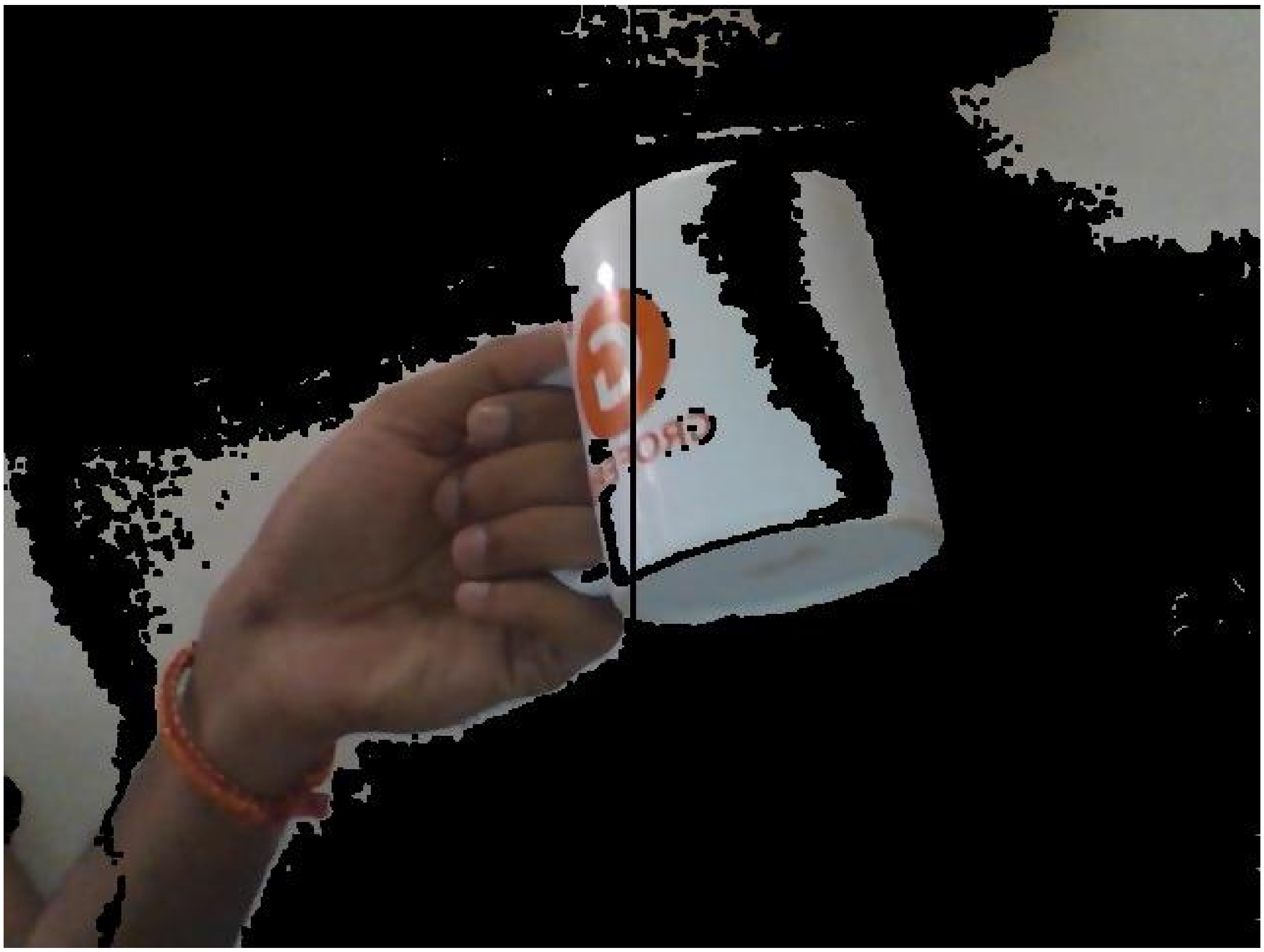
Every time user presses 'c', this function performs some processing like:

```
Function hand_capture():
    Convert image frame to HSV color space
    Extract pixel values from the boxes in HSV color region space.
    Return a hand histogram created using calcHist from openCV.
```

Once the background and hand histogram have been obtained, we continuously capture the camera feed for the users hand movements.

Background is removed from the capture images taking the background model we created and masking it. The mask is applied after to the captured frames. This will result in a frame containing only hand gestures. 'frame' contains a frame without background after applying the lines.

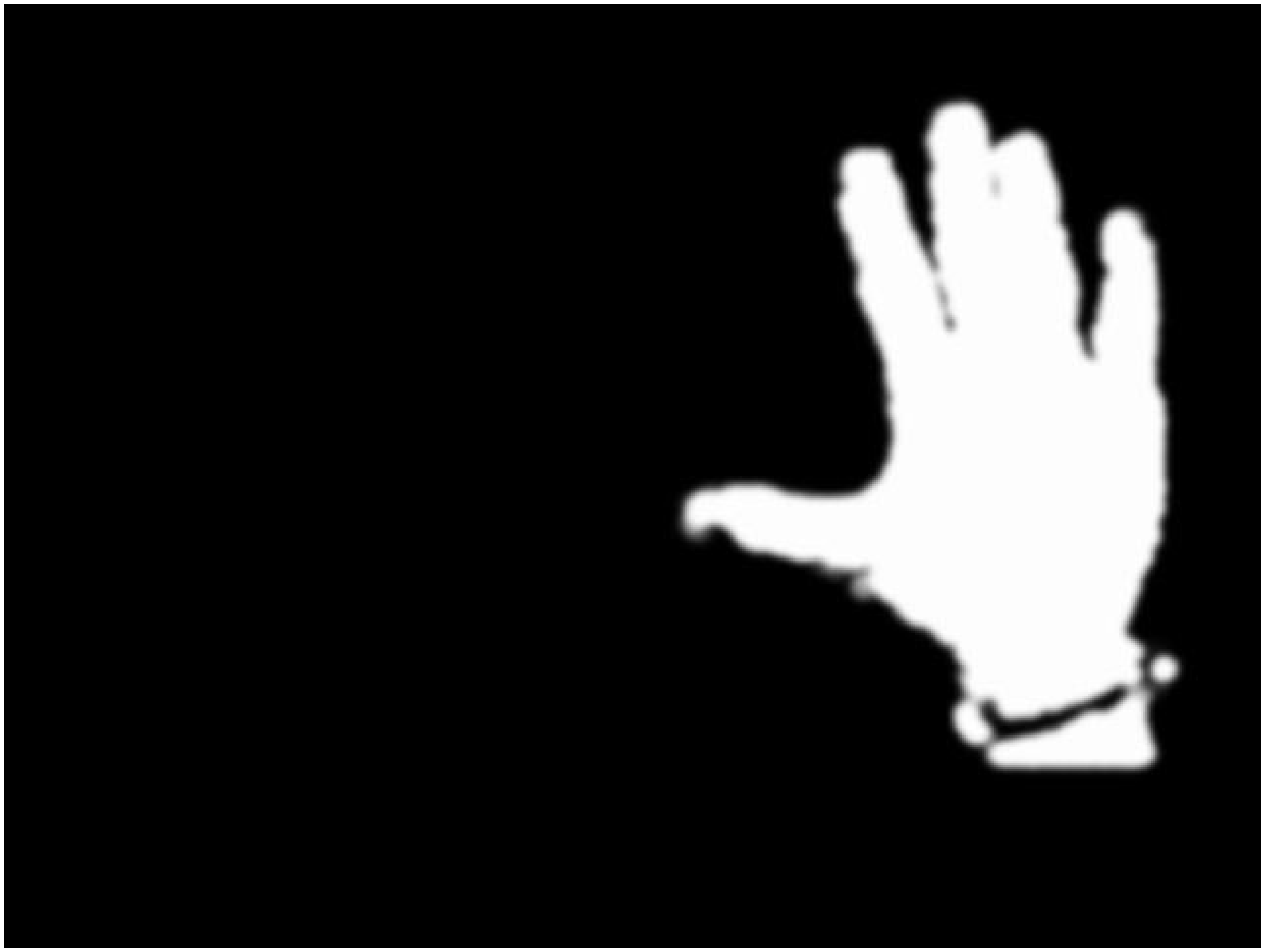
```
fg_mask=bg_model.apply(frame)
kernel = np.ones((3,3),np.uint8)
fg_mask=cv2.erode(fg_mask,kernel,iterations = 1)
frame=cv2.bitwise_and(frame,frame,mask=fg_mask)
```



In this 'frame' we try to find the hand using histogram which contains the shades of hand.

To the 'frame' we do the following transitions.

- $\rightarrow$ Blur the input feed frame for noise reduction
- $\rightarrow$ Convert frame to HSV color space
- $\rightarrow$ Separate Hand part from the rest by using only rectangle area.
- $\rightarrow$ Back projection is calculated using the histogram of hand generated.
- $\rightarrow$ Morphology and smoothing techniques to this back projection.
- $\rightarrow$ Binary image is generated from this projection. This is done using the threshold which creates the mask and separates hand from the rest



We next find the contours of the image generated. The image that has been generated may contain some errors and noise or false detections no matter how efficient the back propagation algorithm is or how perfect the histogram obtained is but because of image disturbances, there can be errors. So instead of just finding the contours, we also apply methods for removing these. We first find the contours of the image and verify if it matches profile of hand.

```
Contours,hierarchy = cv2.findContours(contour_frame,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
```

The 'contours' array consists of contours from the frame using the function from cv2 library. We use the 'hand_contour_find' which is user defined to find largest contour in the image. The largest contour is based on the area using inbuilt function

cv2.contourArea(cont)

and it is assumed that there is no large false detection. The function returns false value in case a blank frame is provided. I.e. in the back projection, detection of any useful data. This is in order to prevent blank frames from stopping execution of other functions that use the detected contours. Only a return state of 'True' would let the program execute on current frame or else the program crashes. The largest contour found is also returned by the function.

'detect_fingers' is the user defined utility function used to implement multiple tasks.

- Going counter clockwise direction, eliminate convex hull points very near to each other. Usually we obtain 10-15 points for each finger but doing the above, we only get 1 point per finger. OpenCV library defined function convexHull() method already does this but we use the user defined function in order to get exact implementation as required for the project.
- All the points too far or near the center of the hand obtained from the 'detect_hand_center'. This gets a single hull point for each finger raised.
- Only 5 fingers at maximum are obtained. We have after this center of the palm, palm radius, and the number of fingers in convex hull. By fingers, we now mean the exact co-ordinates of the points of fingers we selected.

Sine we have palm radius, palm center and finger points, we can make the required model work on these.

The number of fingers obtained from each frame are stored in a list 'ans_list'. The process is repeated on for 30 frames and each time fingers count is entered stored in 'ans_list'. This is done in order to delete false detections or misdetections. From the list the finger count with maximum frequency is taken as the final finger count. This is based on the assumption that the algorithms detect the correct number of the fingers more often than not.

Depending on the input gesture message from the user message is sent accordingly to be written as input to the text file 'data.txt' in format as given below

'header : message'

Finger Count	Header	Message
1	Alert	Message 1
2	Emergency	Message 2
3	Header1	Message 3
4	Hungry	Need food

We have to start the wamp server. Then with the credentials connect to database using php script. Read data.txt file which is used to store the gesture messages through php script. Using curl request send the header and message to base station. Curl allows transfer of data across wide variety of protocols. Before doing with a CURL request, we need to instantiate an instance of curl by calling the function curl_init().

The php script is invocated on the server side using the command. This script reads the text file containing the header:message pair.

```
webbrowser.open('http://localhost/fcmtest/send.php')
```

A server key is generated using the Firebase Cloud Messaging infrastructure. This key is used to establish a secure cURL session connection and authorisation of the user to the firebase server. The firebase server is a third party between application and receiver base station.

The curl session is established using various parameters.

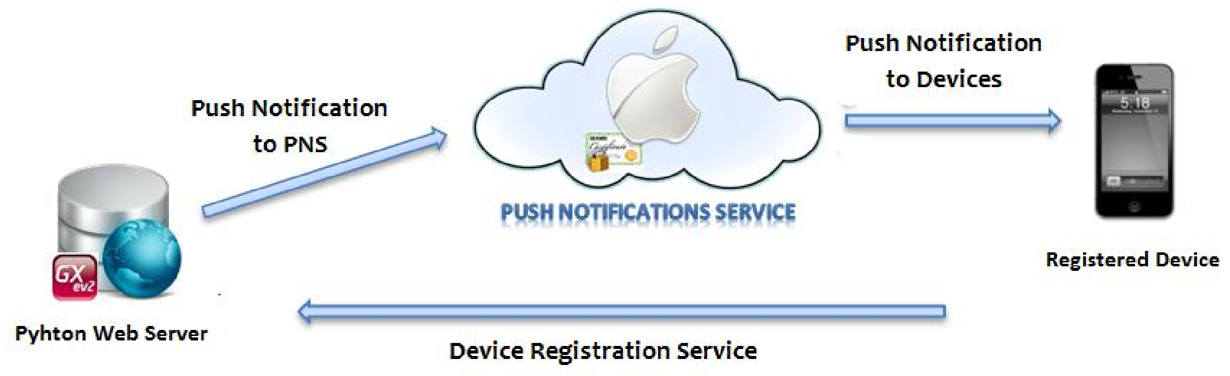
→ CURLOPT_RETURNTRANSFER : Returns the response as a string instead of on screen output. : True

→ CURLOPT_CONNECTTIMEOUT: Number of seconds spent to obtain connection.

→ CURLOPT_TIMEOUT: Number of seconds allowed to cURL to execute

→ CURLOPT_USERAGENT: User agent string for usage of request.
→ CURLOPT_URL: URL to send request. : $path_to_fcm
→ CURLOPT_POST: Send request as post. : True
→ CURLOPT_POSTFIELDS: Array of data to post as a request : $payload

Flow Diagram of the model



Push Notifications allow receiving alerts in the registered Smart Devices from our remote Python Seven when the Smart Device applications are not running. This mechanism describes a style of communication where the request for a given transaction is initiated by the publisher i.e. the python web server. The basic idea is that after the Server knows which its clients are it is able to send the given dedicated message. The main components of the push notification service are –

1. Smart Device Application, which will be the "client-side" (receiver) of the Push Notification service.

2. Web Application, which will be the "server-side" (sender) of the Push Notifications.

3. Push Notification Service, an external resource that provides Push Notification delivery.

This is an outline of the push notification used for this project is describe below –

1. The Android Device is registered with the for push service with the server and a registered ID is given to the Device. This Device ID is known as the Token.

2. Now the android app uploads the Android token received to the server to receive the notification.

3. Now we enable the mobile app to handle push notifications using the Android Studio SDK.

4. Now in the python code of the server whenever an image and is processed it uploads the corresponding text to the processed image to the server which in turn triggers to send push notifications when the certain event occurs.

5. Now the server checks for the token that is registered with it and when it finds the device, it sends the notification to the token device.

Chapter 6

Results and Conclusions

Requirements

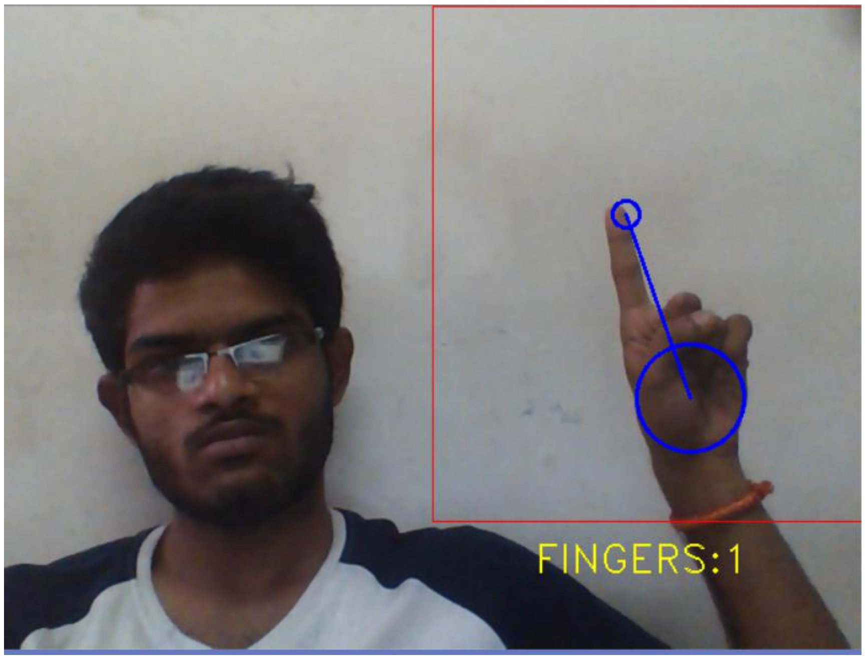
The requirement of the current project is that the application should be able to capture the input feed and detect the sign given by the user correctly. This should be done with minimum noise and fault detection. All the factors considered, the device should be quick to detect and transmit the message to the receiver station who can then take proper action for the request of the sender.

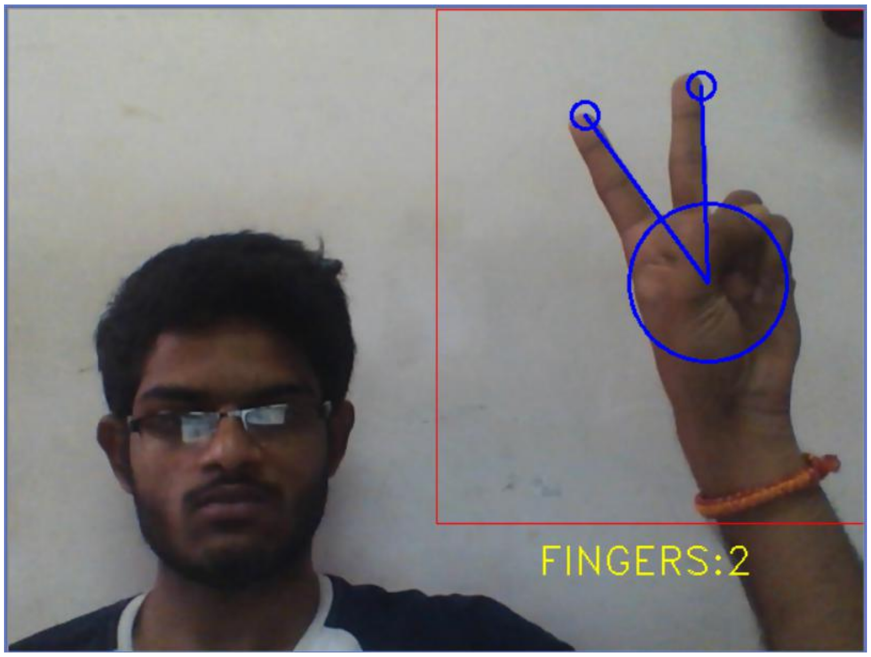
Results

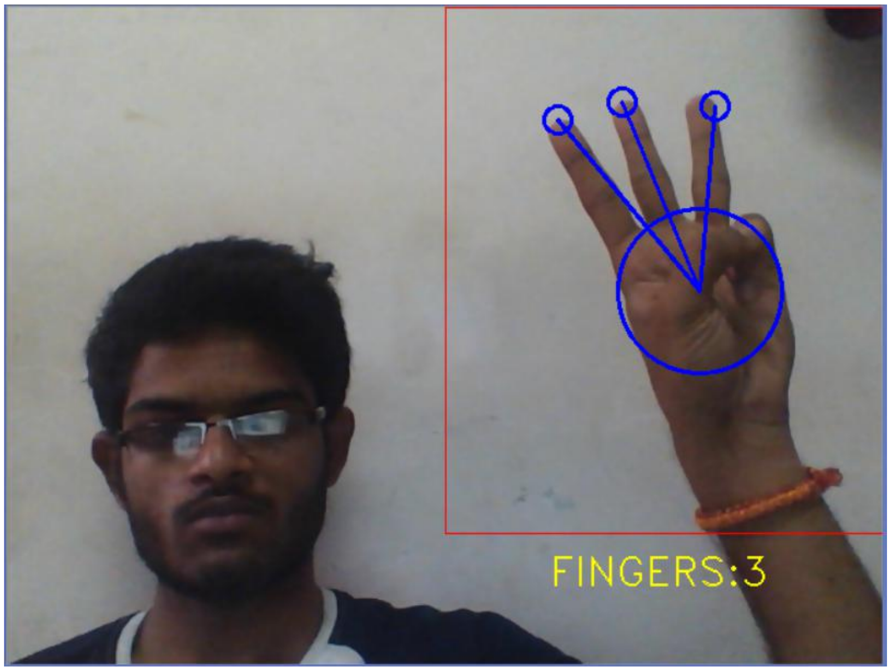
The results obtained by the finger detection, gesture recognition and message communication tool were satisfactory and accurate. These were obtained by the use of good quality hardware and efficient software and infrastructure available that was best for the project.

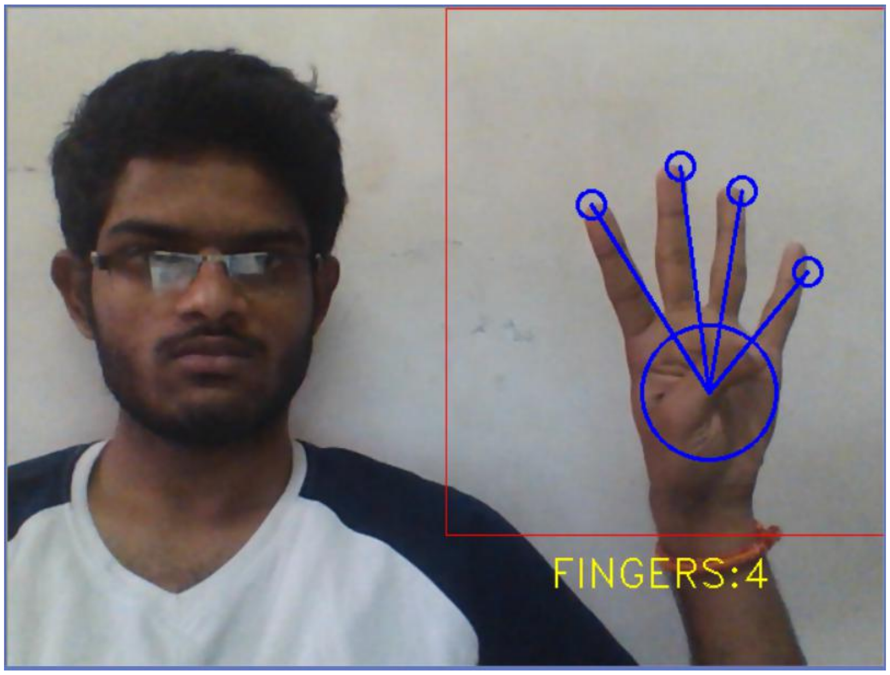
Some of the results obtained from the tool have been presented below.

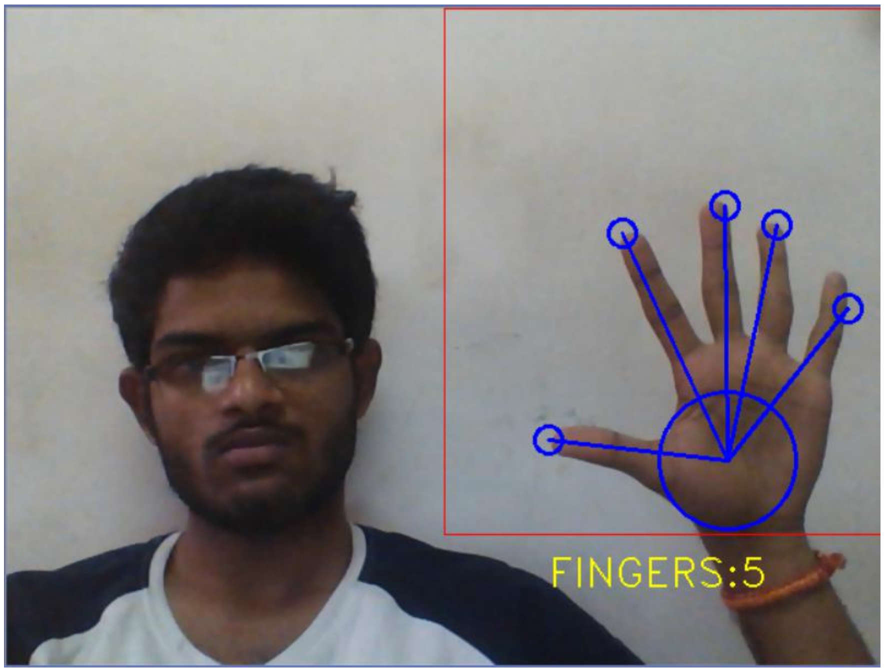
SCREENSHOTS

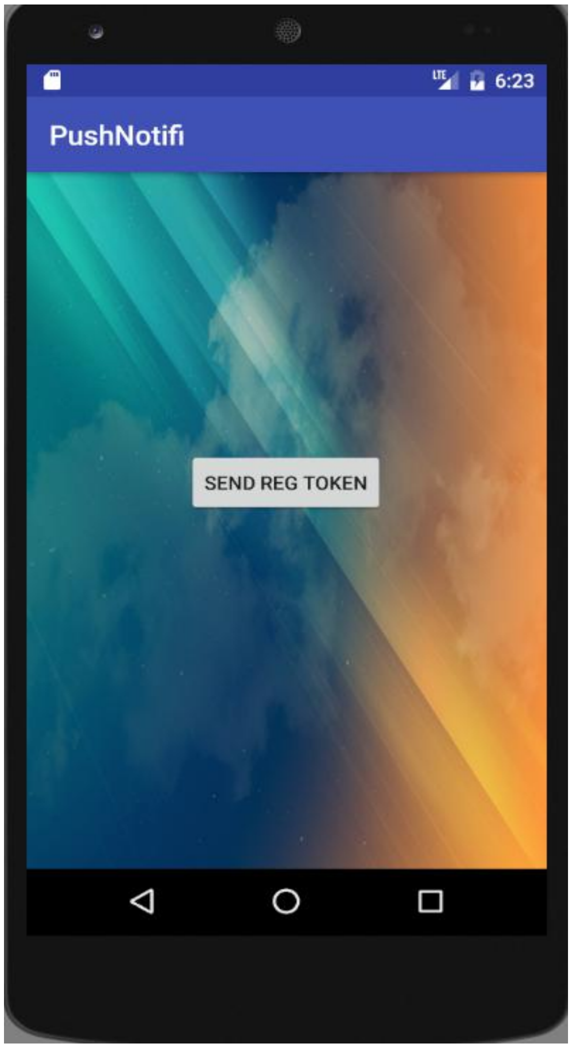


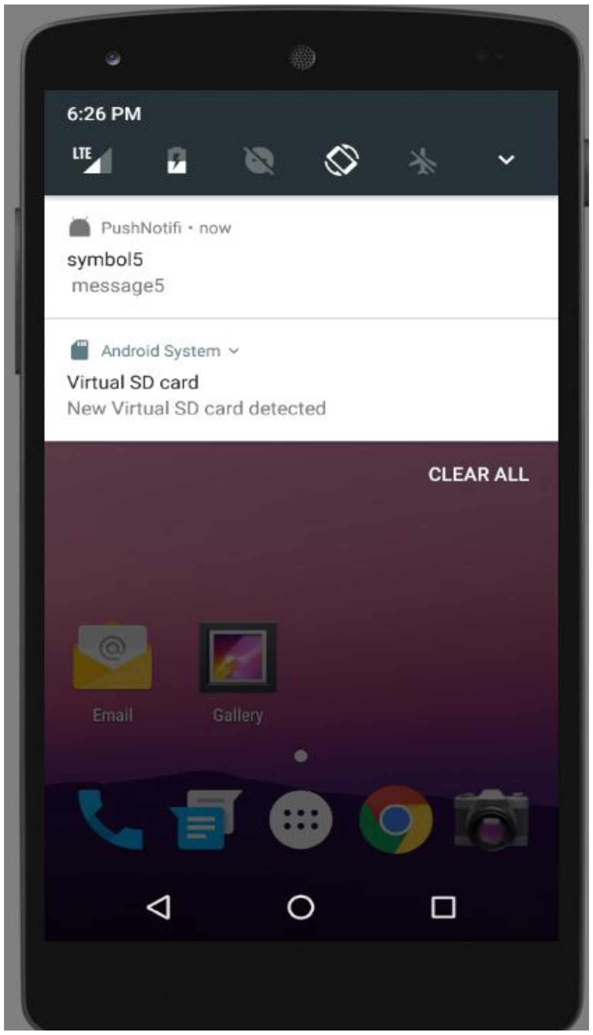

FINGERS:2


FINGERS:3


FINGERS:4


FINGERS:5





PushNotifi • now
symbol5
message5

Android System
Virtual SD card
New Virtual SD card detected

In the modern world where technology is being used by us for accomplishing various human tasks, it is important that technology be also used for helping the differently abled and help them get their needs taken care of. The "GESTURE RECOGNITION AND COMMUNICATION TOOL" addresses this issue and helps in the communication between hearing impaired and dumb with the use of some of the advanced technologies available.

BIBLIOGRAPHY

[1] OpenCV Courses https://www.udemy.com/master-computer-vision -with-opencv-in-python/

[2] https://www.youtube.com/watch?v=ujOTNg17LjI&list=PLQVvvaa0QuDdL kP8MrXLe_rKuf6r80KO.

[3] Contours Tutorial http://docs.opencv.org/trunk/d4/d73/tutorial_ py_contours_begin.html.

[4] Contours http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/ py_imgproc/py_contours/py_contours_hierarchy/py_contours_hierarchy.html

[5] Android tutorials point https://www.tutorialspoint.com/android/

[6] Pygame tutorials https://pythonprogramming.net/pygame-python-3-part-1-intro/

[7] Push Notificationhttps://www.youtube.com/watch?v=uuiuVDb2bug

[8] PHP Scripting https://www.w3schools.com/php/

[9] Filtering and Smoothening http://docs.opencv.org/2.4/modules/imgproc/doc/ filtering.html

[10] Opencv video tutorials https://www.youtube.com/watch?v=Z78zbnLlPUA&list= PLQVvvaa0QuDdttJXlLtAJxJetJcqmqlQq.

[11] Filtering and smoothening images http://docs.opencv.org/2.4/modules/imgproc/ doc/filtering.html

[12] Numpy usage http://cs231n.github.io/python-numpy-tutorial/

[13] OpenCV documentation: http://docs.opencv.org/2.4.8/

[14] FCM http://www.androidhive.info/2012/10/android-push-notifications -using-google-cloud-messaging-gcm-php-and-mysql/