A

Project Report On

# Gesture Based Application Control System

**B.Tech.-Sem VII**


**Prepared by**

Shiv H. Hansoti (IT-034)

Bhargav S. Joshi (IT-044)



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**FACULTY OF TECHNOLOGY,**

**DHARMSINH DESAI UNIVERSITY**

**COLLEGE ROAD, NADIAD - 387001**


**October , 2018**

**A**

**Project Report On**

# Gesture Based Application Control System

**B.Tech.-Sem VII**

**In Partial fulfillment of requirements for**

Bachelor of Technology

in

Information Technology


**Submitted By:**

SHIV HANSOTI (IT-034)

BHARGAV JOSHI(IT-044)


**Under the Guidance of**

Prof. N. P .Desai


DEPARTMENT OF INFORMATION TECHNOLOGY

FACULTY OF TECHNOLOGY,

DHARMSINH DESAI UNIVERSITY

COLLEGE ROAD, NADIAD - 387001

# CANDIDATE'S DECLARATION

We declare that pre-final semester report entitled " Gesture Based Application Control System " is our own work conducted under the supervision of the guide Prof. N. P. Desai .

We further declare that to the best of our knowledge, the report for B.Tech. VII semester does not contain part of the work which has been submitted either in this or any other university without proper citation.

Candidate's Signature:

Candidate's Name: BHARGAV JOSHI
Student ID: 15ITUOF057

Candidate's Signature:

Candidate's Name: SHIV HANSOTI
Student ID: 15ITUBS020

Submitted To: Prof. N. P. Desai

Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Gujarat.

DHARMSINH DESAI UNIVERSITY

NADIAD-387001, GUJARAT

# CERTIFICATE

This is to certify that the project carried out in the subject of Software Design Project, entitled **"Gesture Based Application Control System"** and recorded in this report is a bonafide report of work of

**1) BHARGAV JOSHI Roll No.: IT-044 ID No.: 15TUOF057**
**2) SHIV HANSOTI Roll No.: IT-034 ID No.: 15ITUBS020**

of Department of Information Technology, Semester VII. They were involved in Project work during academic year 2017-2018.

Prof. N. P. Desai
(Project Guide)
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

Prof. (Dr.) V. D. Desai
Head , Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

# ACKNOWLEDGEMENT

This project being the first exposure to real life problems in our field, undertaken by us was an unforgettable and educative experience. We take this opportunity to thank all those who have generously helped us to give a proper shape to our work.

Our sincerest appreciation must be extended to Dharamsinh Desai University, Nadiad. We also want to thank the qualified Faculties as they have been very helpful for the cause.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from Prof. N. P. Desai of Department of Information Technology who helped us in successfully completing our project work. Also, I would like to extend our sincere esteems to all staff in laboratory for their timely support.

Yours Sincerely,

**BHARGAV JOSHI ( IT-044 )**

**SHIV HANSOTI ( IT-034 )**

# TABLE OF CONTENTS

# ABSTRACT

There are many researches going on about detecting user gestures and controlling or performing actions on soft wares. It would be useful if we can access some basic functionalities by performing simple hand gestures.

Once system for detecting hand and then extracting gestures is developed, it can be directly integrated in many devices such as laptops, projectors, smart televisions, drones etc.

This application provides such functionalities in laptop using its front camera. For controlling currently running application user performs some gesture for specific action. By this user can control or give some basic actions to soft wares like vlc player, Chrome, Groove Music, Car racing game. User do not need to touch Keyboard or touch-pad. Just by detecting gestures via front camera, system will perform appropriate action for appropriate detected gesture.

When the system has enough accuracy, probability of detecting wrong gesture is least and so there is not Any possibility of wrong action taken.

# LIST OF FIGURES

**Figure** **Page No.**

# LIST OF TABLES

**Table** **Page No.**

# CHAPTER:1 INTRODUCTION

## 1.1 PROJECT DETAILS

This is a system which detects gestures of user and perform actions in currently running application. It can be used with softwares such as Chrome, VlC player, Groove Music, Car Racing Game. Any softwares in which we can use basic functionalities of the software by pressing keyboard keys, can be integrated with this system.

It provides basic functionalities such as volume up/down, video play/pause and forward/backward in vlc player. Functionalities such as Zoom in/out, scroll up/down left/right, opening new tab in Chrome. Playing game, which is controlled by arrow keys, volume up/down, and play/pause in Groove music. Applications can also be closed by providing gesture

## 1.2 PURPOSE

The purpose of this project is to eliminate the need of physical contact between software and user. It makes touch less user interface interface.It also eliminate the need of any kind of remote control because it is completely based on gestures given by users. It can replace complex keyboard key press sequence with simple gestures.

## 1.3 SCOPE

This Application can be divided in three modules.

1) Image Contouring and Extraction
   Contour with specific area and color range is extracted from the image.

2) Processes Detection:
   Currently running process is determined for its usage.

3) Actions:
   Based on the Extracted Contours, process and gestures the required or specified actions are performed in the application.

## 1.4 OBJECTIVE

The objective of this project is to successfully extract appropriate gesture from actions of user and control applications based on that. Gestures must be extracted currently and actions on the software must be performed very quickly. It must be able to process user gesture in minimum amount of time.

## 1.5 TECHNOLOGY AND LITERATURE OVERVIEW

### 1.5.1 About OpenCV

It's an open source library written in c++ and its primary interface is in c++. It provides implementation of the various image processing algorithms. There are bindings in Python,java and Matlab/octave.The api for these interface can be found in the online documentation.

OpenCV's application area includes facial recognition,gesture recognition, 2D and 3D feature toolkits, motion understanding, object identification ,motion tracking , augmented reality etc. To support some of the above areas it also includes the statistical machine learning library that contains decision tree learning, gradiantt boosting trees, k-nearest neighbor algorithm ,Naïve Bayes classifier, support vector machines, Deep neural networks etc.

OpenCV runs on following desktop os such as Windows, Linux, MacOS etc. and mobile os such as Android, ios, Blackberry etc.

### 1.5.2 About pyautogui

PyAutoGUI is a cross-platform GUI automation Python for human beings.Used to programmatically control the mouse and keyboards. The API is designed to be as simple as possible. It runs on all major os such as Windows, MacOS, Linux. PyAutoGUI can simulate moving the mouse, clicking the mouse, dragging with the mouse, pressing keys, pressing and holding keys, and pressing keyboard hotkey combinations.

Program could get out of control (even though it is following your instructions) and need to be stopped. This can be difficult to do if the mouse is moving around on its own, preventing you from clicking on the program's window to close it down.

As a safety feature, a fail-safe feature is enabled by default. When pyautogui.FAILSAFE = True PyAutoGUI functions will raise a pyautogui.FailSafeException if the mouse cursor is in the upper left corner of the screen. If you lose control and need to stop the current PyAutoGUI function, keep moving the mouse cursor up and to the left. To disable this feature, set FAILSAFE to False

# CHAPTER:2 PROJECT MANAGEMENT

## 2.1 FEASIBILITY STUDY

The feasibility of software can be tested in 4 dimensions:

**Technology-**

Is the project technically feasible? As in our case, we have a few existing examples and references of the same, so the software is technically feasible.

**Finance-**

Is is financially feasible? Does it have too much cost of development? As in our case, the need is of a simple colour camera. In most of the devices it is already integrated and even it is of very low cost if needed externally. So,this project is financially feasible.

**Time-**

Will it take too much time to complete? As in our case, we have planned each phase and it seems to be in control and within time and so no extra time cost shall be added.

### 2.1.1. Technical Feasibility

All the tools required for this project are easily available and are available for free with respect to development purposes. The user will simply require their PCs or mobile phones alongwith an active Internet Connection.

#### 2.1.2. Time Schedule Feasibility

This project is estimated to complete within 10-12 weeks which is under the allocated time.

### 2.1.3. Operational Feasibility

Reliability: This application runs flawless under appropriate light conditions. It would have problem in detecting gesture if there is not enpogh light for detecting object or too much direct light.

Portability: This application would run on any system which have python installed.Hence, portability is assured.

Usability: Easy and correct extraction of gesture and touch less interface increases usability of the software.

### 2.1.4. Implementation Feasibility

It can be easily implemented keeping in mind the feasibility and requirements of the project along with some limitations using OpenCV and PyAutoGUI.

## 2.2 PROJECT PLANNING

### 2.2.1. Project Development Approach And Justification

The project idea was clicked by the requirement of application which can controls other applicatons without direct contact of desktop.

The solution was to create an application which contols other application by User gestures. Gestures can only be detected using front camera of the laptop.So there is not any need of remote control or any physical contact with hardware device.

Image processing library OpenCV is used for gesture extraction and for action PyAutoGui is used.For project development Iterative Waterfall Model is used.

Iterative Waterfall Method:

The Iterative water fall model approach overcomes the problems associated with the waterfall model approach.

If any difficulty or problem encounter in any phase may require going back to the previous phase and performing the required modifications and proceeds sequentially.

This backtracking allows modifying any corrections or modifications required in the previous phase.

As illustrated in Fig 2.1, this model divides the cycle into the phases mentioned below:

1. Feasibility Study.
2. Requirements analysis and specification.
3. Design.
4. Coding and Unit Testing.
5. Integration and System Testing.
6. Maintenance.

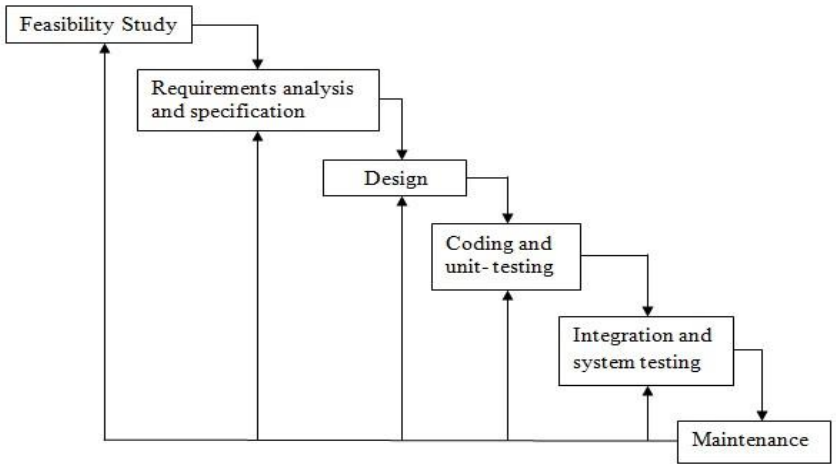



Fig. 2.1 Iterative Waterfall Model

### 2.2.2. Project Plan

The plan composed to develop a system which can control other application based on gestures performed by user. Making user interface touch less and much less complex.

Here we have decomposed the system into modules. Also the internals of the individual modules are designed in greater details.

Coding and Unit Testing phase is required to translate the software design into source code. Also during this phase each module is unit tested to determine the correct working of all the individual modules.

Integration and System Testing phase consists of the integration of the modules in a planned manner. Here during each integration step, we have tested the partially integrated system.

Finally, when all the modules were successfully integrated and tested, system testing was carried out successfully.

### 2.2.3. Milestones And Deliverables

Management needs information. As software is intangible, this information can only be provided as documents that describe that state of the application being developed.

Without this information, it is impossible to judge progress and cost estimates and schedules cannot be updated.

When planning a project series of milestones are established.

Milestones:

- Weekly Reporting of the milestones as done orally on the specified dates.
- Changing the RGB image in HSV image and masking the contour of specific object.
- Finding Contour of maximum area and take that as Gesture performing object.
- By calculating difference between positions of center of selected contour extracting the performed gesture.
- Detection of currently running software.
- Performing actions on currently running software based on extracted gesture.
- Extract the disturbance generated by the background objects and selecting the correct object.

Deliverables:

- In our case system can only be delivered when It is developed. It can not be delivered in modular way because each module here serves different purpose and even with one module missing system can not even work partially.

### 2.2.4. Roles and Responsibilities

| **Name** | **Roles** | | | | |
|---|---|---|---|---|---|
| | **Analysis** | **Designing** | **Coding** | **Testing** | **Documenta-tion** |
| Bhargav Joshi | Yes | Yes | Yes | Yes | Yes |
| Shiv Hansoti | Yes | Yes | Yes | Yes | Yes |

Table 2.1 Roles and Responsibilities

### 2.2.5. Group Dependencies

All the tasks for each of Analysis, Designing, Coding, Testing and Documentation is equally distributed beforehand and there was no dependencies between the project developers.

## 2.3 PROJECT SCHEDULING

Scheduling the project tasks is an important project planning activity. It involves deciding which tasks should be taken up and when. In order to schedule the project activities; a software project manager needs to do the following:

1. Identify all the tasks needed to complete the project.
2. Break down large tasks into small activities.
3. Determine dependencies amongst different activities.
4. Establish most likely estimates for the time durations necessary to complete the activities.
5. Allocate resources to activities.
6. Plan the starting and ending dates for various activities.

# CHAPTER:3 SYSTEM REQUIREMENTS STUDY

## 3.1 Study of Current System

Present application control systems are based on hardware based remote controls, there is no such System exist which uses Image Processing for performing actions in other applications.

## 3.2 Problems and Weaknesses of Current System

As high resolution camera of laptop is used the time for processing the gestures is more.

This system can be used for single application at a time not multiple application. Eg:- If system is used for chrome then it cannot be used for VLC or any other application.

System underperforms when enough light is not available for camera to detect gestures accurately.

## 3.3 User Characteristics

It is personal desktop based application so it has not any authentication functionality. End users uses it directly without any parental control.

## 3.4 Hardware and Software Requirements

- Python installed environment.
- Camera

## 3.5 Constraints

### 3.5.1 Hardware Limitations

Device must have camera for capturing user gestures. Device must have enough ram and processor configuration for simultaneously capturing and processing images and performing actions on other software.

#### 3.5.2 Software Limitations

Device must have python installed for running this application. The actions which user want to perform on a particular software must have keyboard equivalent key press sequence for that action. Ex. For play and push keyboard key is ‘SPACE’ and for Zoom in Chrome keyboard key sequence is ‘CTRL’ and ’+’.

#### 3.5.3 Lighting Requirements

There must be normal lighting condition for easy detection of object.

#### 3.5.4 Reliability Requirements

The system must extract the gesture and perform action in minimum acceptable time.

#### 3.5.5 Higher order Language Requirement

System is based on PYTHON and uses some libraries implemented in higher level languages.

## 3.6 Assumptions and Dependencies

- Hardware is compatible for high processing. Camera is installed.
- PYTHON is installed.
  The object in normal lighting conditions provides gestures.

# CHAPTER:4 SYSTEM ANALYSIS

## 4.1 REQUIREMENTS OF NEW SYSTEM

### 4.1.1. USER REQUIREMENTS:

**R.1: PRECISE GESTURE EXTRACTION.**

**Description**: when user performs gesture it must be extracted precisely and it must not conflict with any other gesture.

**State:** Camera is open.
**Input:** User's hand movement.
**Output:** Extracted Predefined Gesture.

**R.2: PROCESS DETECTION**

**Description**: Same gesture have different action in different Application, so the currently running application process must be detected for taking appropriate action.

**R.3: ACTION PERFORM**

**Description**: Based on extracted gesture and currently running application process, appropriate actions in that application is performed.

### 4.1.2. SYSTEM REQUIREMENTS:

**R.1: FINE TUNING**

**Description:** All the noises and background disturbance is obviated or affects the system at minimal level.

**R.2: HARDWARE COMPATIBILITY**

**Description:** For simultaneously processing image from camera and performing actions based on extracted gesture enough primary memory and Multi-core processor is required.

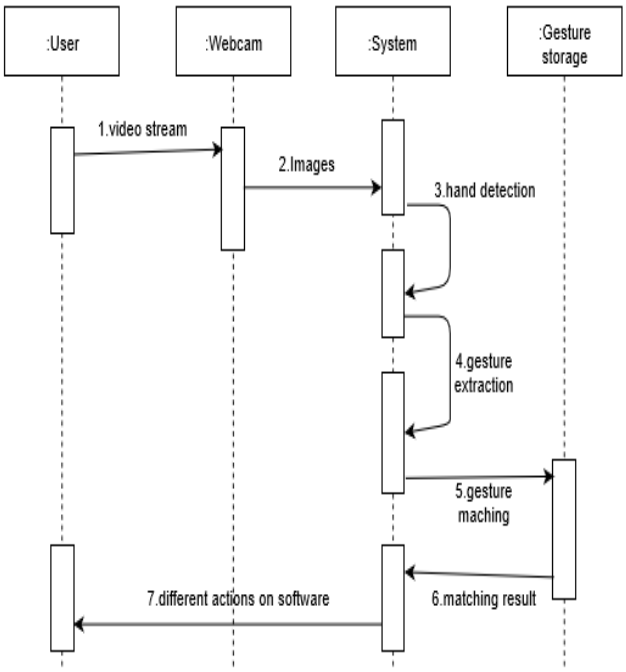
## 4.2 FEATURES OF NEW SYSTEM

This is the core concept of Advanced Gesture Technology which can be used in various ways.

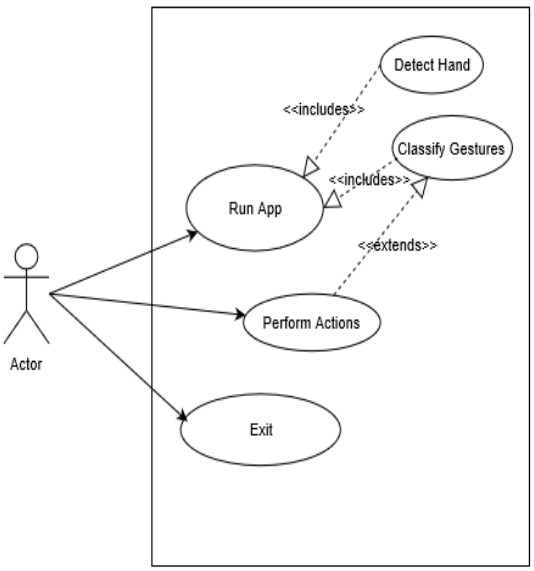
This application will provide features such as:

- Real Time Gesture Extraction and Contours.
- Process Detection.
- Application Control.
- No physical or Wired Hindrance.

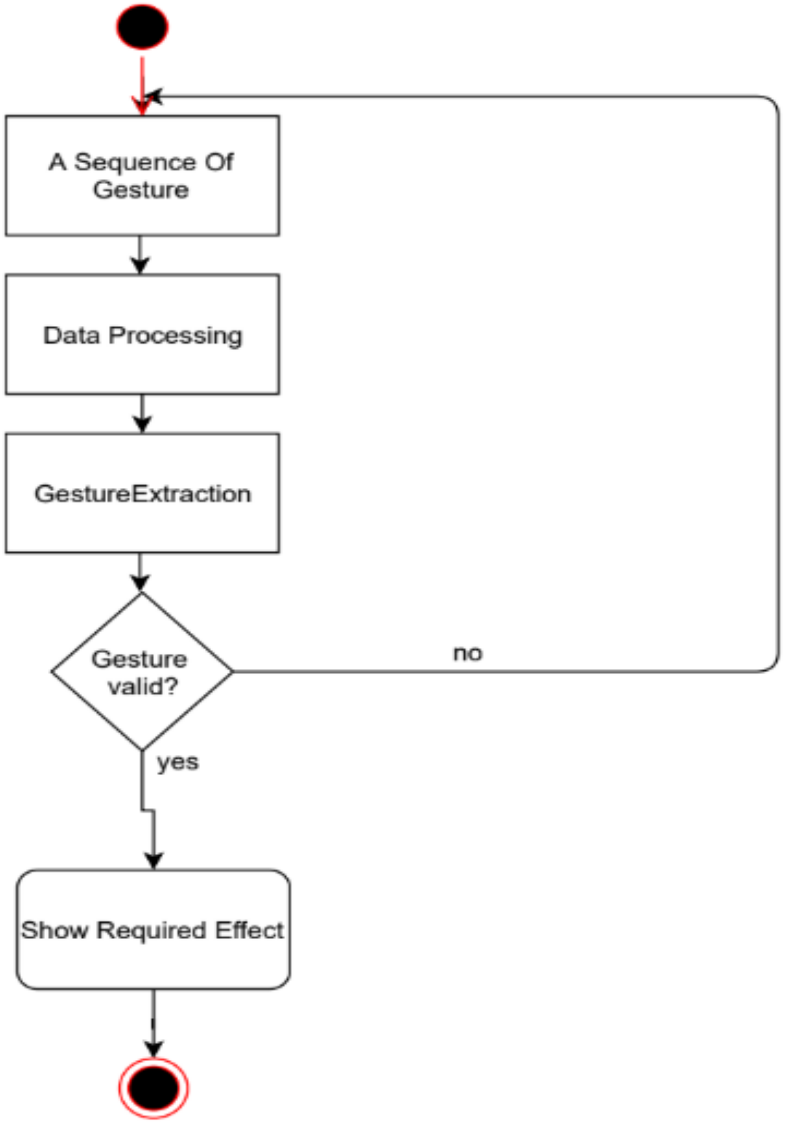
## 4.3 SEQUENCE DIAGRAM



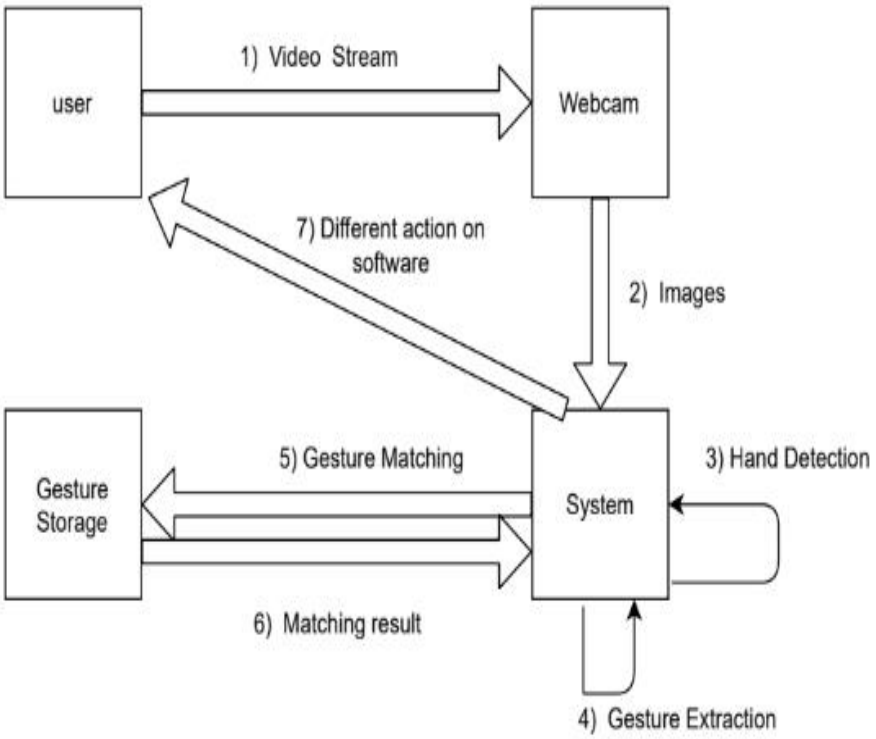
## 4.4 USE-CASE DIAGRAM



## 4.5 FLOW CHART



## 4.6 COLLOBRATION DIAGRAM



# CHAPTER: 5
# IMPLEMENTATION PLANNING

## 5.1 IMPLEMENTATION ENVIRONMENT

This Application gives idea the of Gesture Recognization Technique, Which is used in Smart televisions, Army Drones, Advance Gadgets.

An Experienced and authorized user can use this system for controlling Systems using Hand Gestures. User could interact with devices by simple Hand Gestures.

This Application avails the user with a amicable environment by

- Moving reference finger.
- Usage of camera for gesture recognition.
- Validations to avoid wrong inputs in the Application.

## 5.2 PROGRAM SPECIFICATION

This Application is built in Python with the help of Anaconda framework the data here is mostly dynamic and to provide support for gesture recognition we have used front camera of laptop.

**5.3 SOURCE CODE**

## DRIVER.PY

```
import cv2
import numpy as np
from collections import
deque
from processd import
current_process
import pyautogui as gui
import time



buff=128
oldx=0
oldy=0
currx=0
curry=0
counter=0
counter1=0
call_counter=0;
flag=0
last_motion=""
flag_do_gesture=0
current_gesture = []

yellow_lower =
np.array([7, 96, 85])
yellow_upper =
np.array([255, 255,
255])

blue_lower =
np.array([110,50,50])
blue_upper =
np.array([130,255,255]
)
cap =
cv2.VideoCapture(0)
def perform(gesture):
current=current_process()
if(current=="chrome"):
if(gesture=="N"):
        gui.hotkey('DOWN')
        gui.hotkey('DOWN')
        gui.hotkey('DOWN')
        gui.hotkey('DOWN')
        gui.hotkey('DOWN')

    if(gesture=="S"):
        gui.hotkey('UP')
        gui.hotkey('UP')
        gui.hotkey('UP')
        gui.hotkey('UP')
        gui.hotkey('UP')

    if(gesture=="E"):
        gui.hotkey('LEFT')
        gui.hotkey('LEFT')
        gui.hotkey('LEFT')

    if(gesture=="W"):
        gui.hotkey('RIGHT')
        gui.hotkey('RIGHT')
        gui.hotkey('RIGHT')

  if(gesture=="NW"):
      gui.hotkey('ctrl','t')

    if(gesture=="SW"):
        gui.hotkey('ctrl','-')

    if(gesture=="NE"):
      gui.hotkey('ctrl','+')

    if(gesture=="SE"):
        gui.hotkey('alt','f4')
```

```
if(current=="vlc"):
if(gesture=="S"):
gui.hotkey('DOWN')

if(gesture=="N"):
gui.hotkey('UP')

if(gesture=="E"):
gui.hotkey('RIGHT')

if(gesture=="W"):
gui.hotkey('LEFT')

if(gesture=="SW"):
gui.hotkey('space')

if(gesture=="SE"):
gui.hotkey('alt','f4')


if(current=="music"):
if(gesture=="S"):
gui.hotkey('volumedown')

if(gesture=="N"):
gui.hotkey('volumeup')

if(gesture=="SW"):
gui.hotkey('space')

if(gesture=="SE"):
gui.hotkey('alt','f4')
while(1):
_, frame = cap.read()
img=frame
img = cv2.flip(img, 1)
img = cv2.resize(img, (480, 360))
hsv = cv2.cvtColor(img,
cv2.COLOR_BGR2HSV)
```

```
mask = cv2.inRange(hsv, blue_lower, blue_upper)

   blur = cv2.medianBlur(mask, 15)
   blur = cv2.GaussianBlur(blur , (5,5), 0)
   temp,thresh = cv2.threshold(blur,0,255,
     cv2.THRESH_BINARY+cv2.THRESH_OTSU)

   _,thresh = cv2.threshold(blur,0,255,
      cv2.THRESH_BINARY+cv2.THRESH_OTSU)
   cv2.imshow("Thresh", thresh)

   _, contours, _ = cv2.findContours(thresh.copy(),
       cv2.RETR_TREE,
cv2.CHAIN_APPROX_NONE)

   print(len(contours))

   if(len(contours)==0):
       line_pts = deque(maxlen = buff)
       print("Empty Image")

       processed_gesture_hand1 = tuple
         (process_gesture(current_gesture))

       if flag_do_gesture == 0:

          if processed_gesture_hand1 != ():

do_gesture_action(processed_gesture_hand1)
          flag_do_gesture = 1
       print(processed_gesture_hand1)
       created_gesture = []
       flag = True
```

```
current_gesture.append(last_motion)


        elif diffx > 20 and diffy > 20:
                last_motion="SE"
                perform(last_motion);

current_gesture.append(last_motion)

       elif diffx < -20 and diffy > 20:

elif diffx > 20 and diffy < -20:
                last_motion="NE"
                perform(last_motion);

current_gesture.append(last_motion)

      elif diffx < -20 and diffy < -20:
                last_motion="NW"
                perform(last_motion);

current_gesture.append(last_motion)

          flag=False
    cv2.imshow("Detected: ", img)
    k = cv2.waitKey(5) & 0xFF
    if k == 27:
        break

cv2.destroyAllWindows()
```

## PROCESS.PY

```
import psutil


VLC = "vlc.exe"
CHROME = "chrome.exe"
GAME = "Highway Racer.exe"

def current_process():
    for proc in psutil.process_iter():
        if proc.name()==VLC:
            #print("vlc")
            return "vlc"

        if proc.name()==CHROME:
            #print("chrome")
            return "chrome"

        if proc.name()==GAME:
            #print("chrome")
            return "game"



current_process()
```

# CHAPTER: 6 TESTING

## 6.1 TESTING PLAN

The testing technique that is going to be used in the project is black box testing.

In black box testing the expected inputs to the system are applied and only the outputs are checked.

## 6.2 TESTING STRATEGY

The development process repeats this testing sub-process a number of times for the following phases.

a) Unit Testing.
b) Integration Testing

Unit Testing tests a unit of code (module or program) after coding of that unit is completed.

In application Unit Testing of three modules was done.

1) Gesture Extraction (Does the system responds correctly when any gesture is drawn)
2) Process Identification (All the running processes are identified)
3) Action Module (Actions performed virtually Based on given conditions)

Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct.

Firstly, the process is identified by the application and user performs gesture. And based in that gestures, actions are performed. Hence by integrating Process Identification, Gesture Extraction and Action Module the application works correctly.

Testing is carried out in such a hierarchical manner to ensure that each component is correct and the assembly/combination of components is correct.

Merely testing a whole system at the end would most likely throw up errors in components that would be very costly to trace and fix.

## 6.3 TEST CASES

| Test Case Description | Input | Expected Output | Resultant Output | Pass / Fail |
|---|---|---|---|---|
| Gesture Recognition | Move reference finger In east, west, north or south direction | Gesture Recognized | As Expected | Pass |
| Forward video in VLC player | Move reference finger in east direction | Video forwards by 5 seconds | As Expected | Pass |
| Rewind video in VLC player | Move reference finger in west direction | Video rewinds by 5 seconds | As Expected | Pass |
| Volume Up In VLC player | Move reference finger in North direction | Volume increases initially by 10% | As Expected | Pass |
| Volume Down In VLC player | Move reference finger in South direction | Volume decreases initially by 10% | As Expected | Pass |
| Close VLC media Player | Move reference finger in South East direction | VLC media player gets closed | As Expected | Pass |
| Pause/Play video | Move reference finger in South West direction | Video gets Resume/paused in VLC player | As Expected | Pass |

| Test Case Description | Input | Expected Output | Resultant Output | Pass / Fail |
|---|---|---|---|---|
| Move Car in Forward Direction | Move reference finger in North direction | Moves car Ahead | As Expected | Pass |
| Stop the car | Move reference finger in South direction | Stops the moving car | As Expected | Pass |
| Move car left | Move reference finger in west direction | Car moves left | As Expected | Pass |
| Moves car right | Move reference finger in east direction | Car moves right | As Expected | Pass |

| | | | | |
|---|---|---|---|---|
| Move page Up in Google Chrome | Move reference finger in North direction | Scrolls page in upward direction | As Expected | Pass |
| Move page Down in Google Chrome | Move reference finger in South direction | Scrolls page in Downward direction | As Expected | Pass |
| Zoom in the page for better observation | Move reference finger in North East direction | Zoom in the page | As Expected | Pass |
| Zoom out the page for better observation | Move reference finger in South West direction | Zoom out the page | As Expected | Pass |

Test cases are presented such that possible paths of control flow through the software item are traced. Hence more defects than black-box testing are likely to be found.

The disadvantages are that exhaustive path testing is infeasible and the logic might not conform to specification.

Instrumentation techniques can be used to determine the structural system coverage in white box testing.

For this purpose, tools or compilers that can insert test probes into the programs can be used.

# CHAPTER: 7

# USER MANUAL

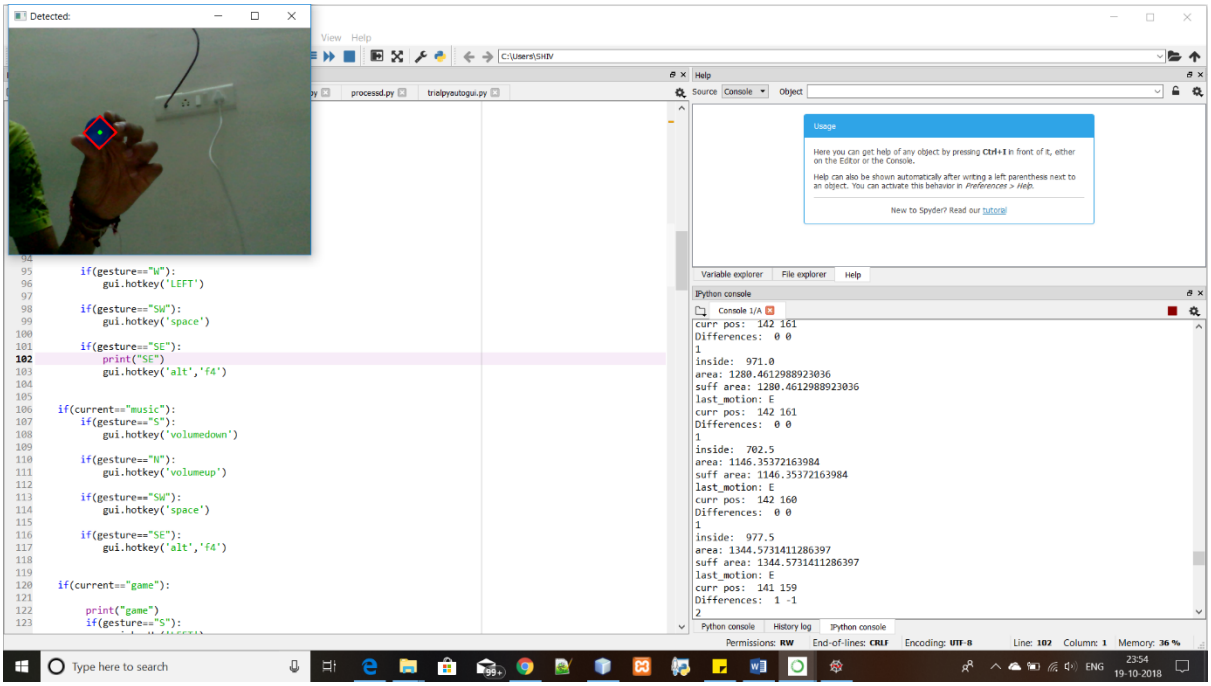



Fig. 7.1 Camera And Gesture Detection

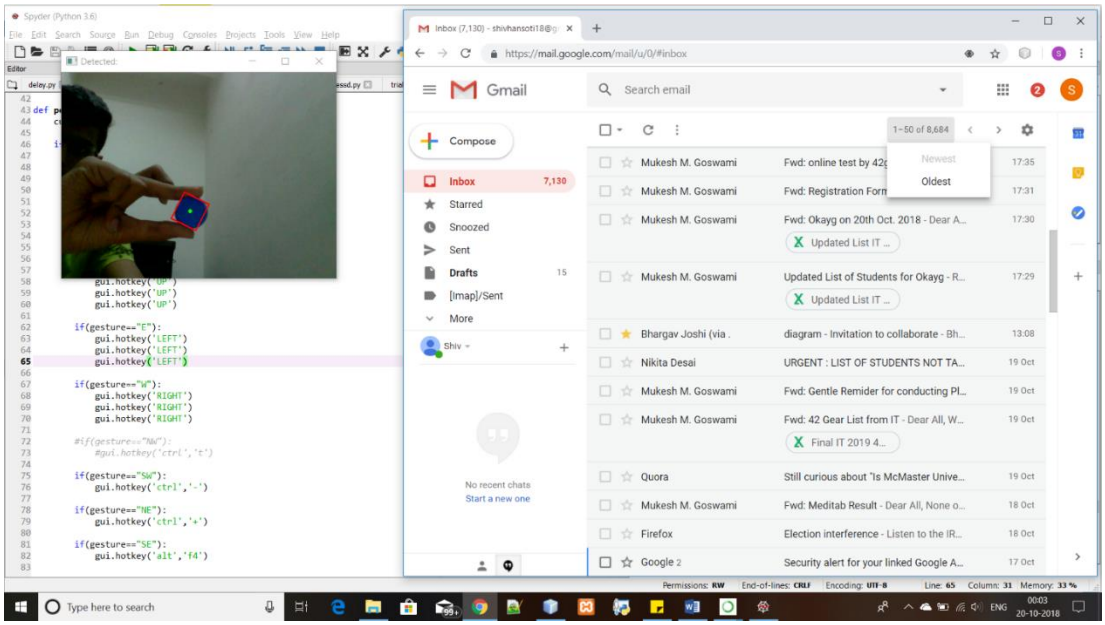


Fig. 7.2.1 Before Scroll down in Chrome

**Perform SOUTH gesture for scrolling down and NORTH gesture for scrolling up.**

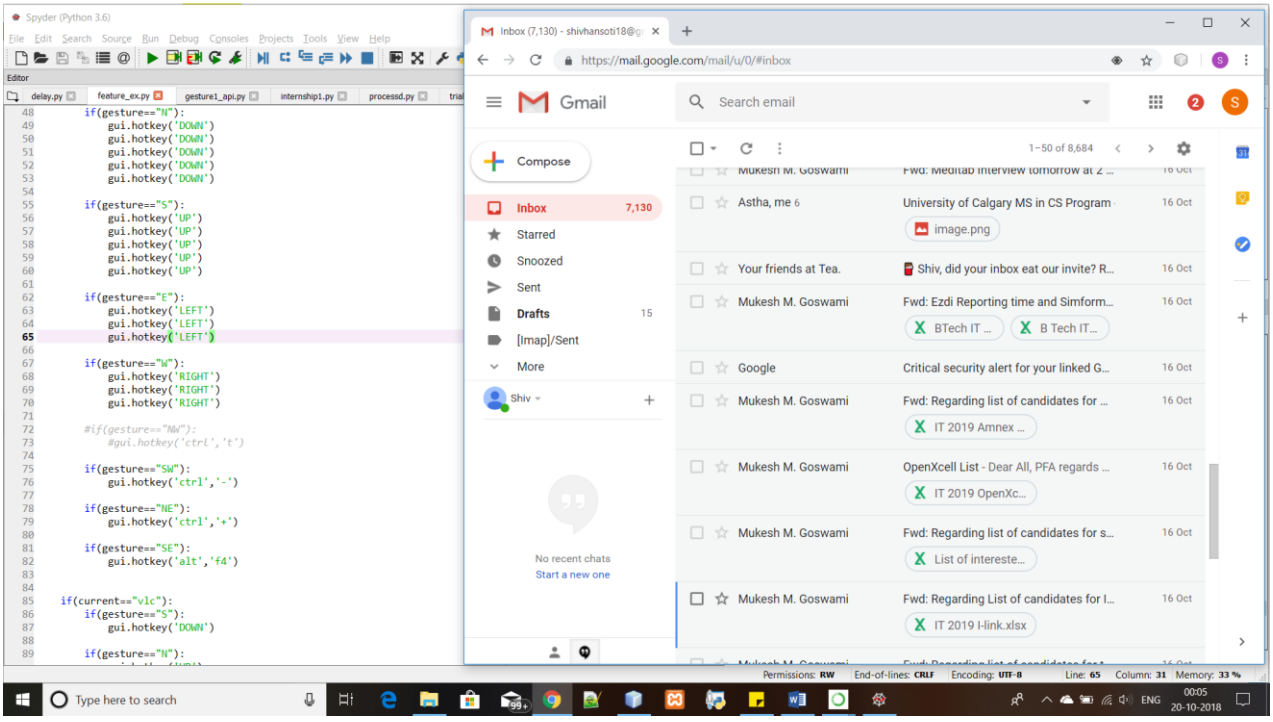


Fig. 7.2.2 After Scroll down in Chrome

**Perform NORTH-EAST gesture for zooming page**

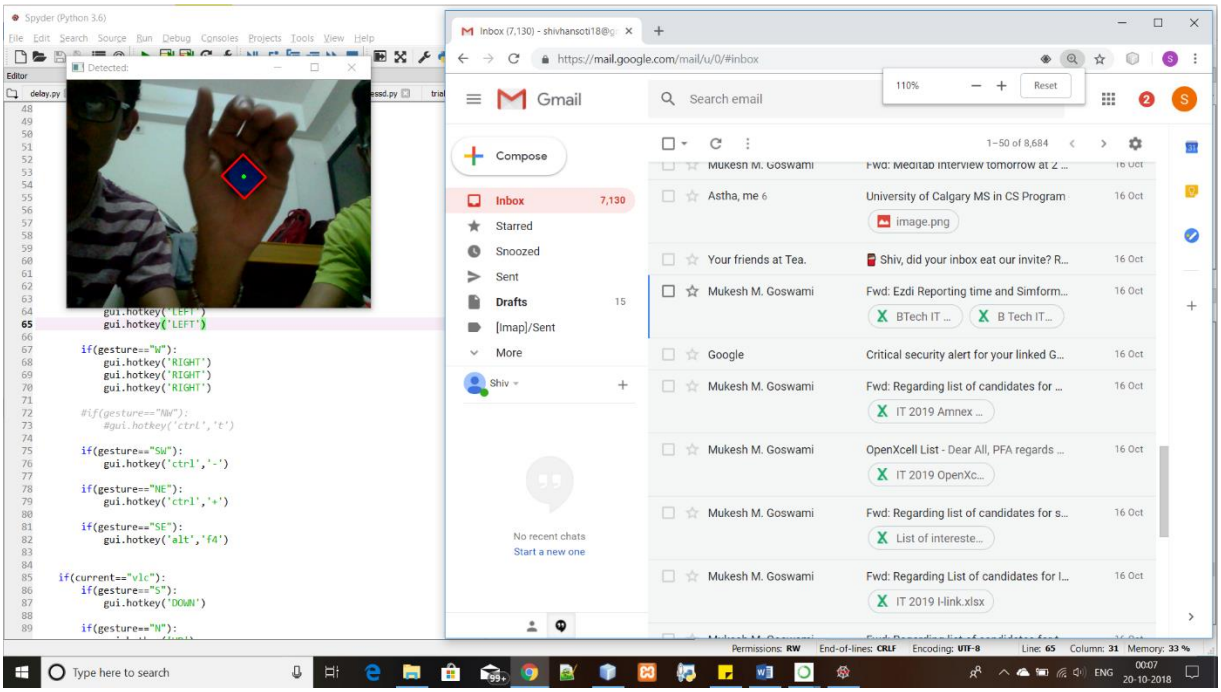


Fig. 7.3.1 Zoom in Chrome Page

**Perform SOUTH-WEST gesture for zooming out page**

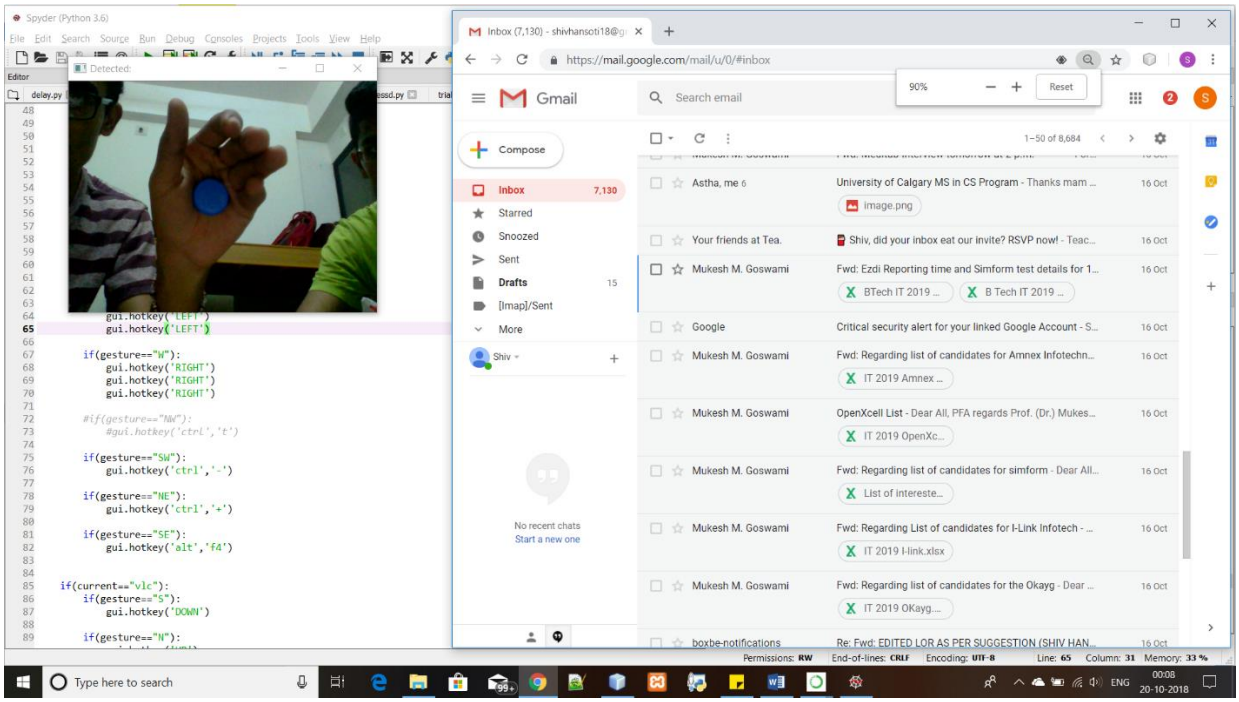


Fig. 7.3.2 Zoom out from Chrome Page

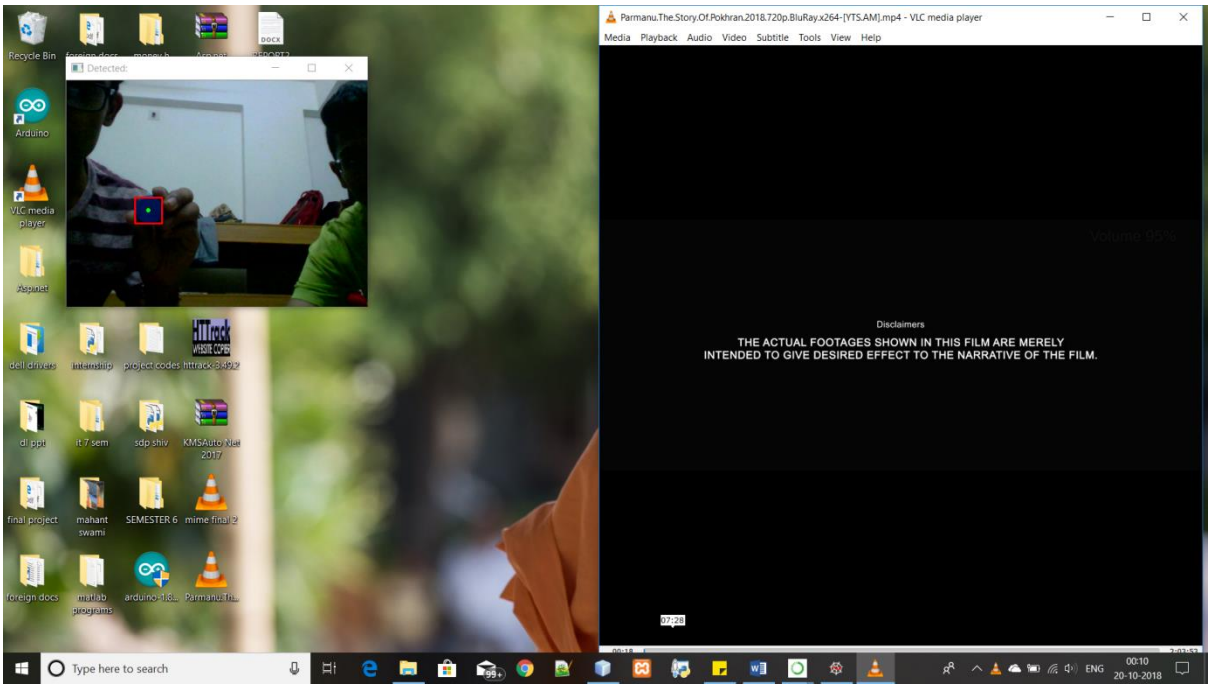



Fig. 7.4 Simple VLC Open

**Perform EAST gesture for forwarding video**

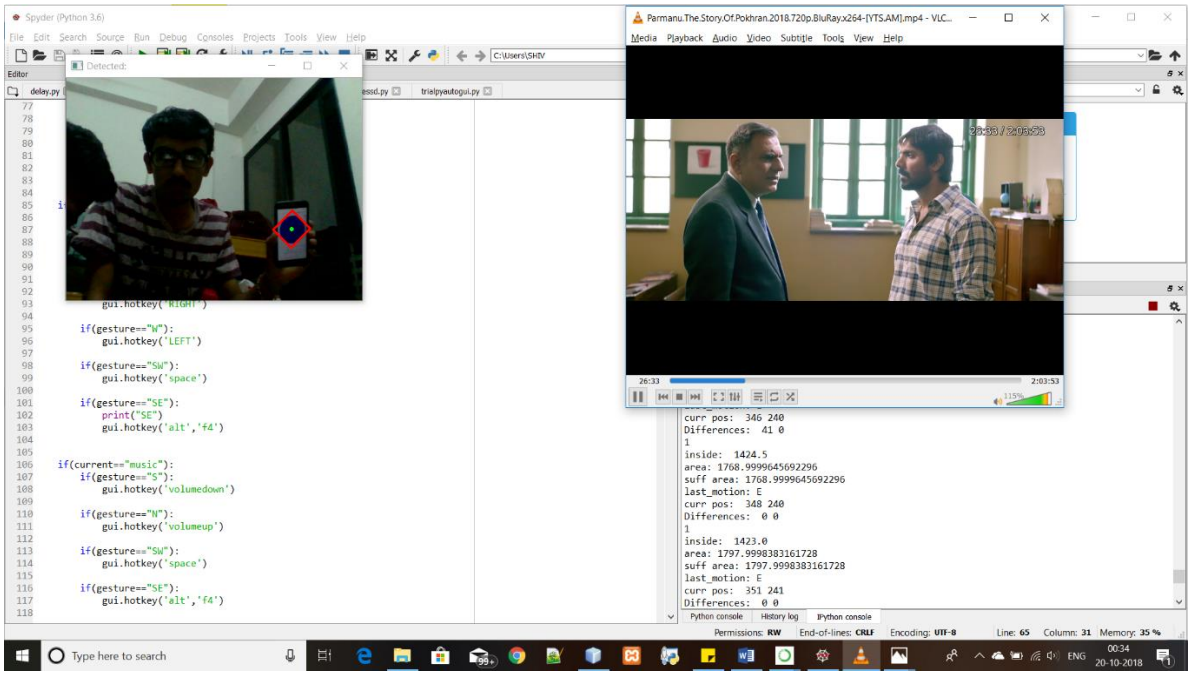


Fig. 7.4.1 Forward Video in VLC

**Perform WEST gesture for reversing video**

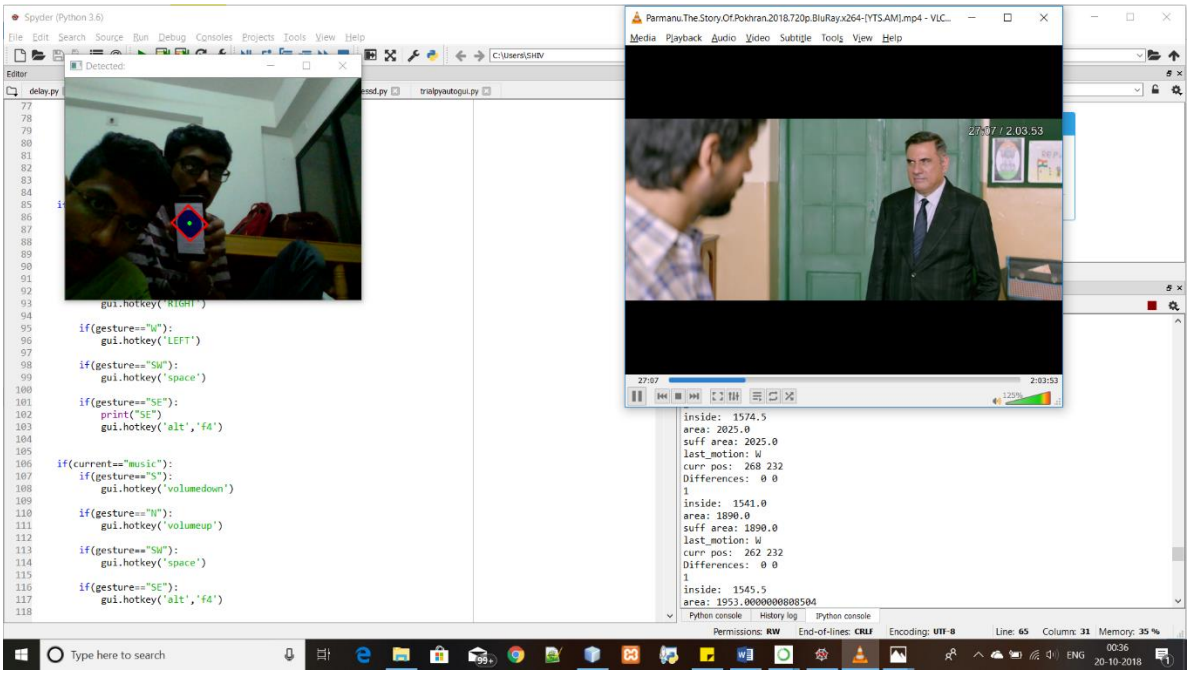


Fig. 7.4.2 Forward Video in VLC

**Perform SOUTH-WEST gesture for pause/play video**

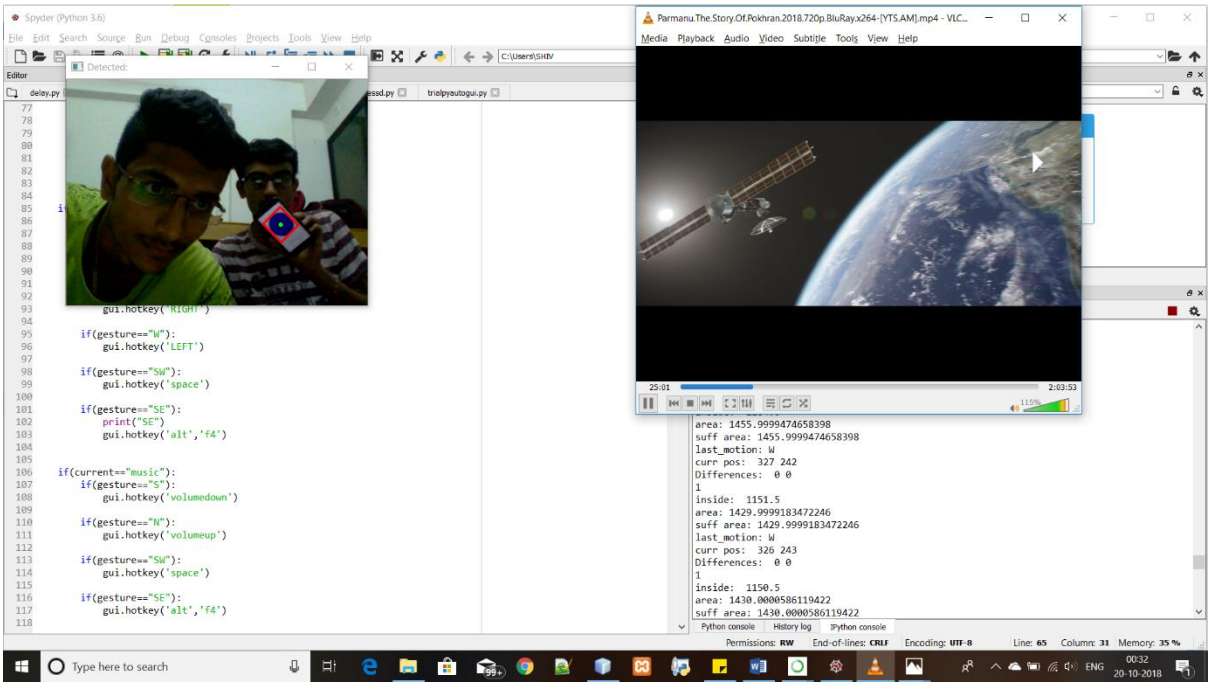


Fig. 7.4.3 Play Video in VLC

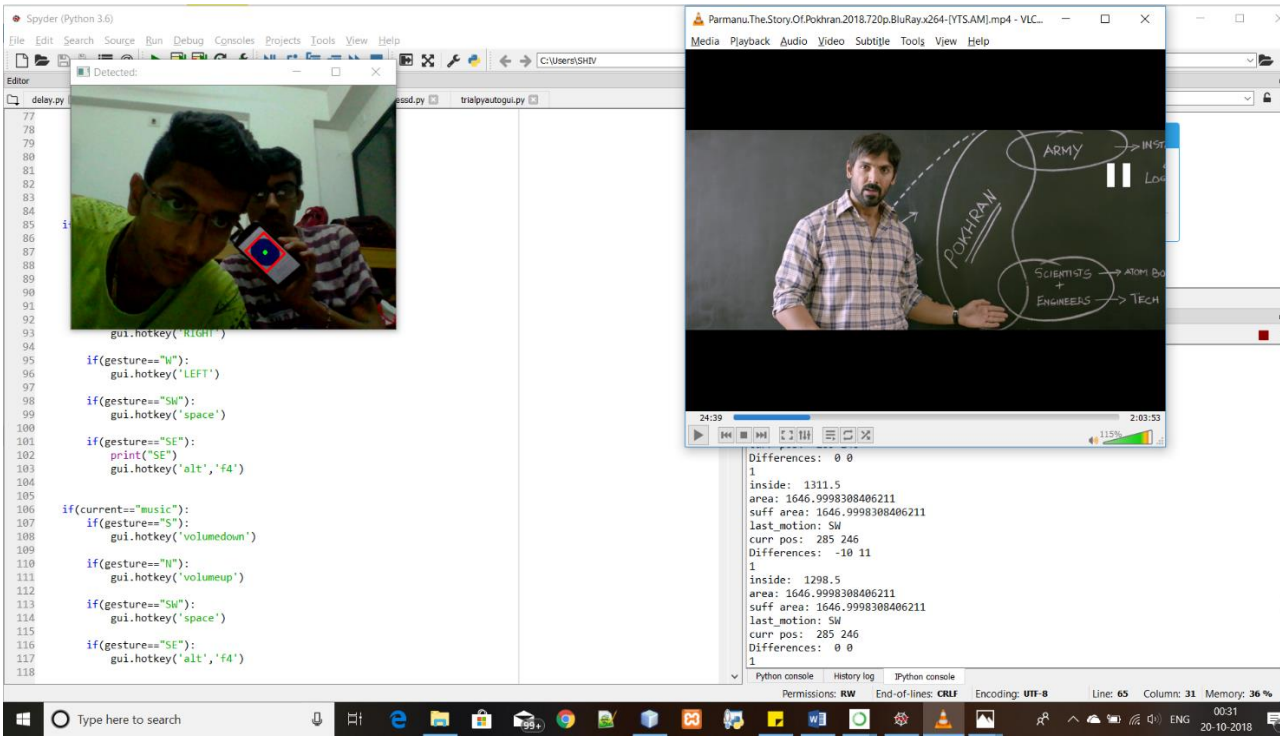


Fig. 7.4.3 Pause Video in VLC

**Perform SOUTH gesture for decreasing volume**

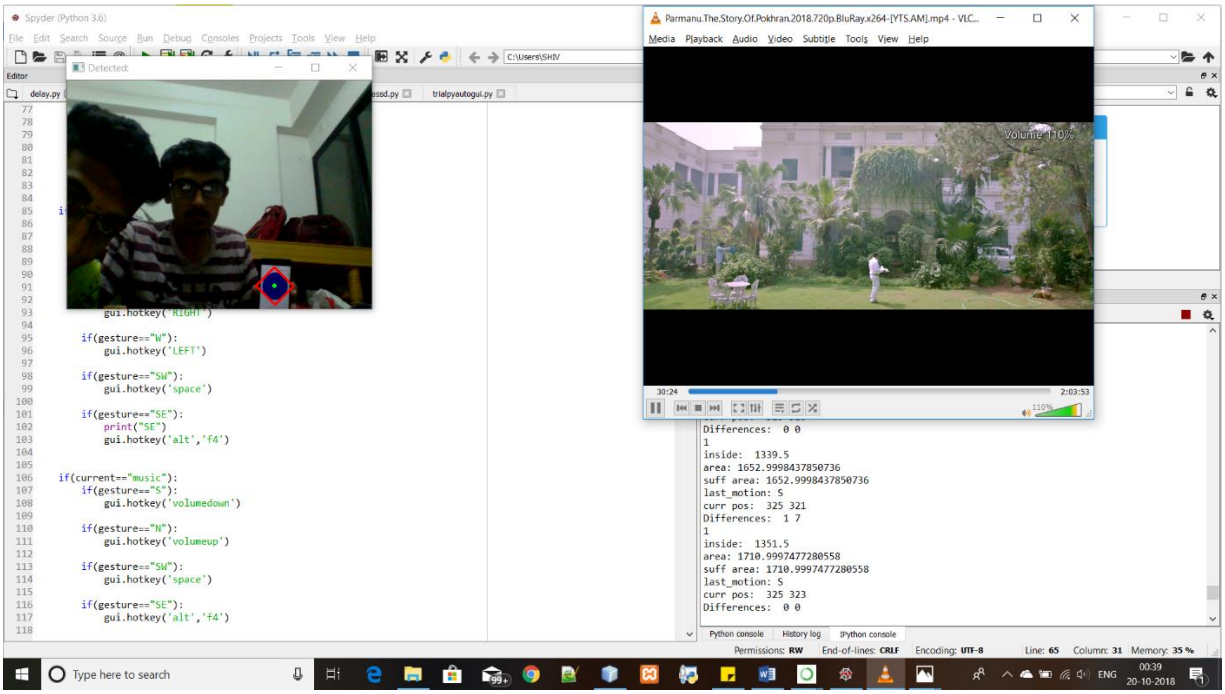


Fig. 7.4.4 Volume Down in VLC

**Perform NORHT gesture for increasing volume**

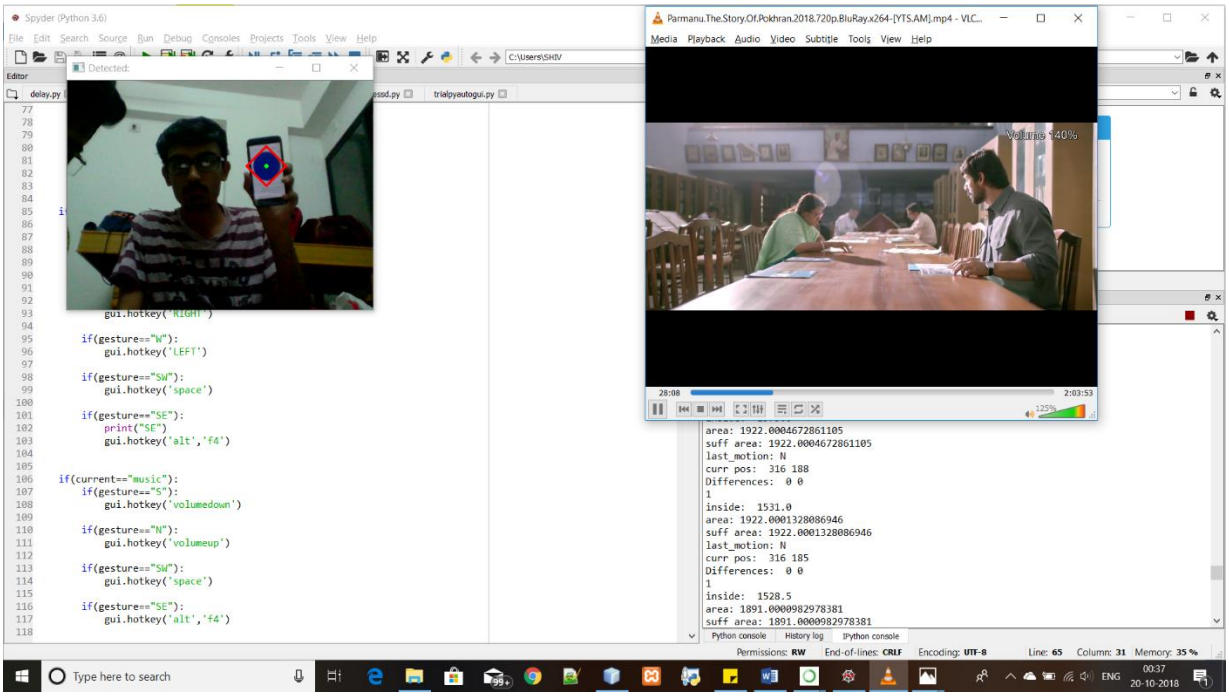


Fig. 7.4.5 Volume Up in VLC

**Perform WEST gesture for moving car left**

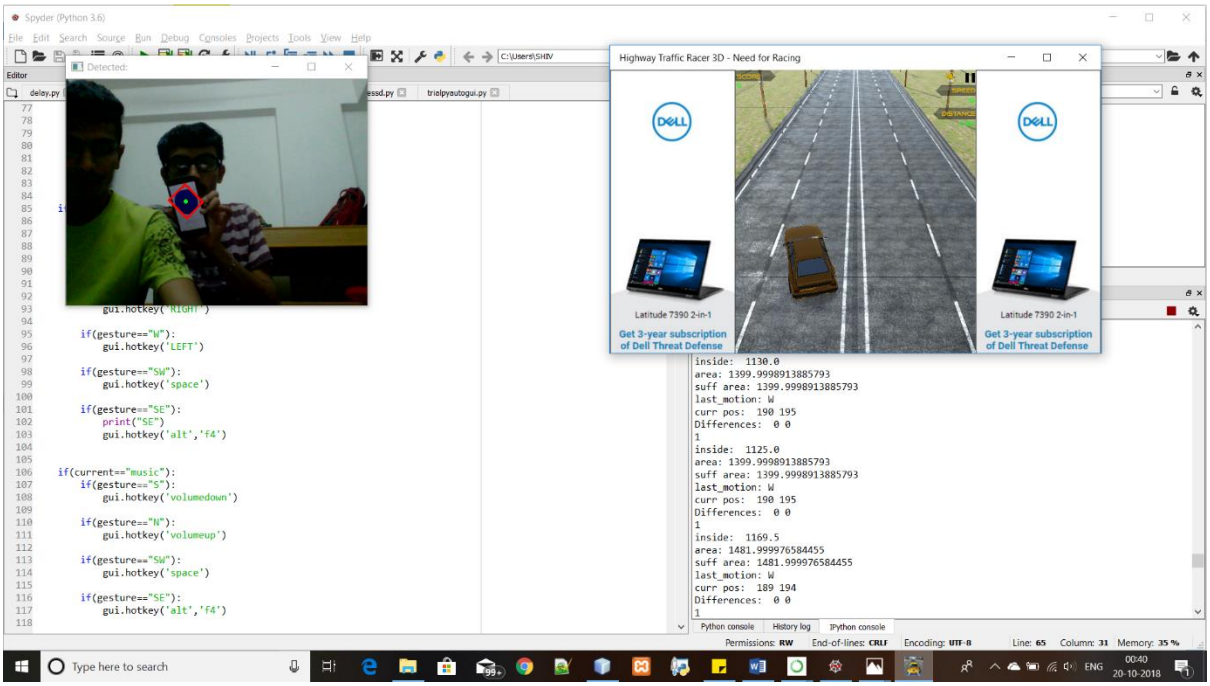



Fig. 7.4.6 Left Turn in GAME

**Perform EAST gesture for moving car right**

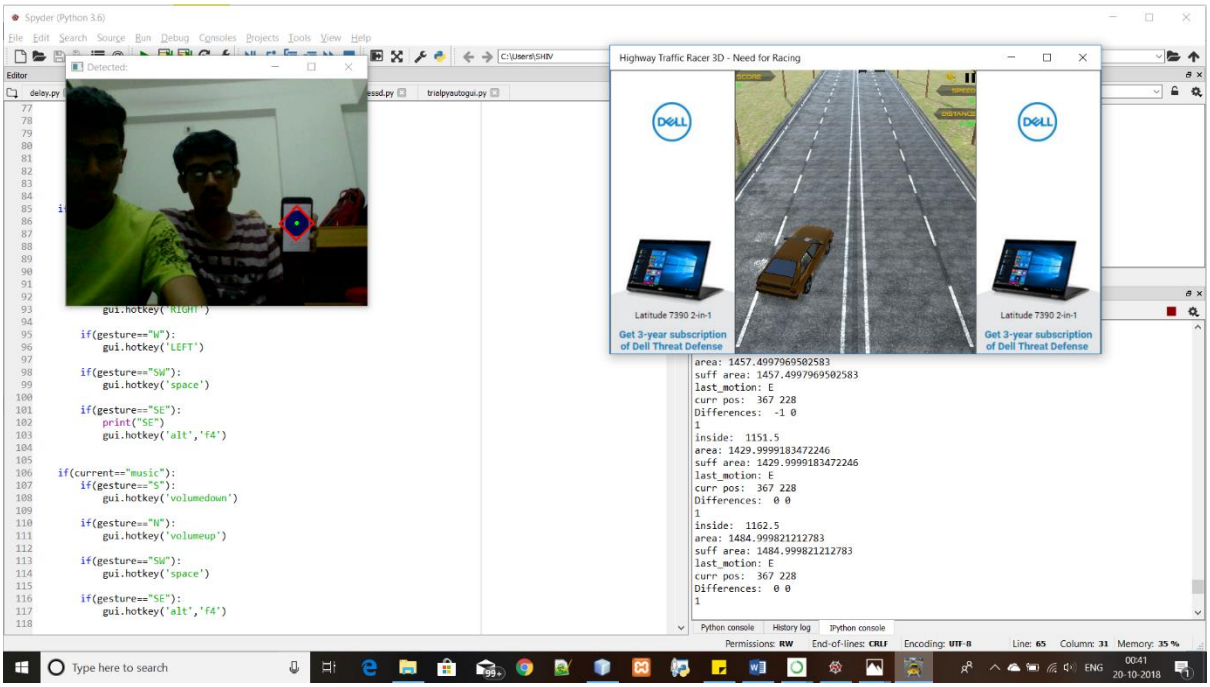



Fig. 7.4.7 Right Turn in GAME

**Perform NORTH gesture for moving car ahead**

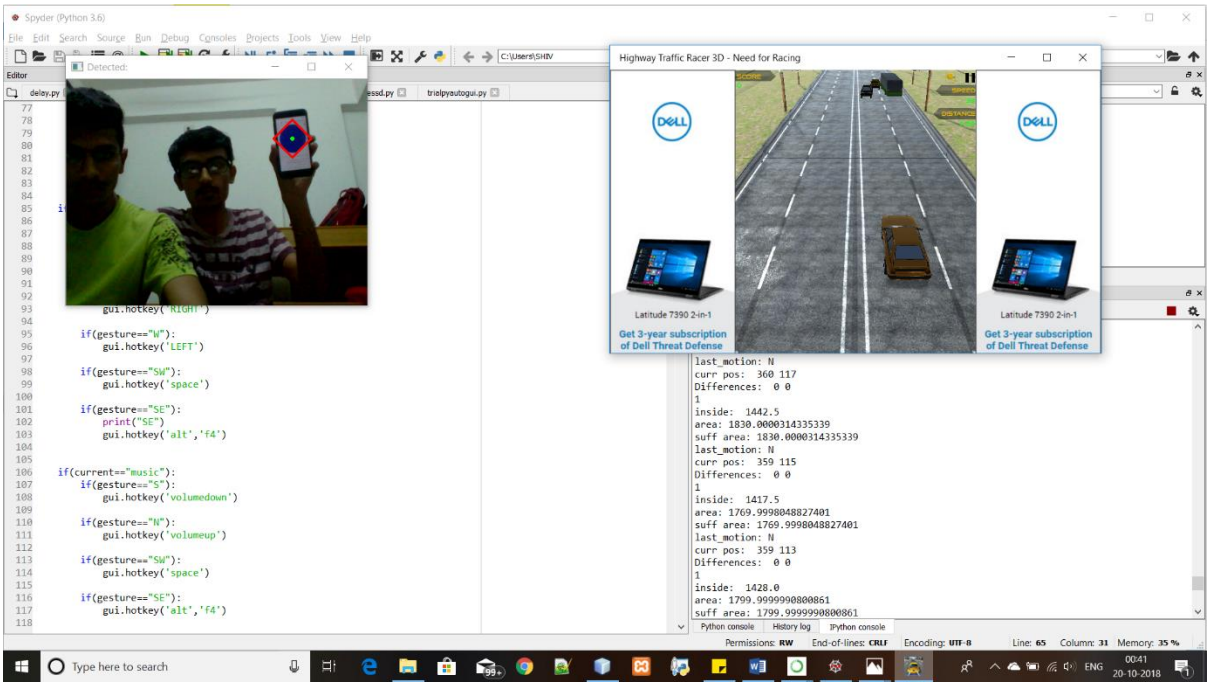



Fig. 7.4.8 Speed Up in GAME

**Perform SOUTH gesture for applying break**

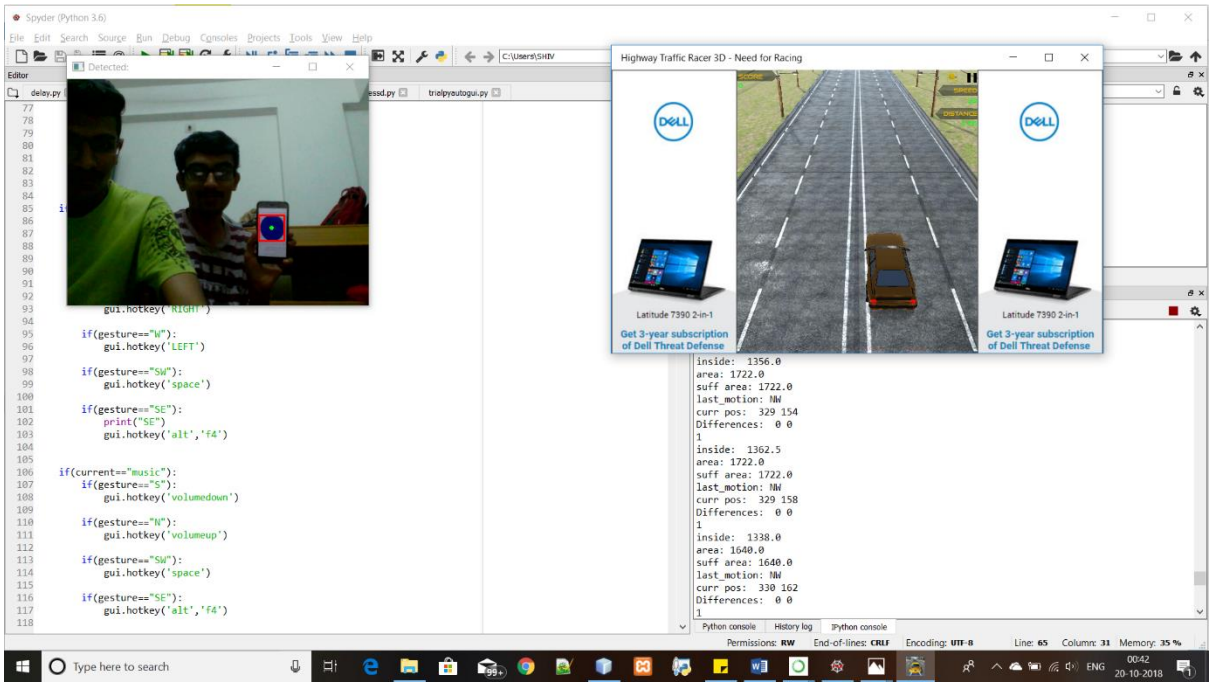



Fig. 7.4.9 Brake in GAME

# CHAPTER: 8 LIMITATIONS AND FUTURE ENHANCEMENTS

## 8.1 LIMITATIONS

- The application must require camera which must be kept on.
- If two same reference objects are detected the performance of application is affected.
- If there is no light, then Image Contouring becomes difficult.
- Range for providing Gesture is short and limited.

## 8.2 FUTURE ENHANCEMENTS

- If this same concept is provided in Drone with IOT devices like Raspberry pi then a gesture controlled Drone can be built.
- It could be used in Enhancement of Smart Gadgets like Smart wrist Bands.
- Also an application of Gesture Controlled Robotics, in which Deep Learning Algorithms are used to Train Robots by observing Human activities for performing activities.

# CHAPTER: 9 CONCLUSIONS AND DISCUSSIONS

## 9.1 CONCLUSIONS AND FUTURE ENHANCEMENTS

It can be concluded that by using the resources available such as Spyder Framework of Anaconda IDE of Python platform, Open CV, Hardware such as camera one can implement Advance Gesture Recognition system for various purposes.

In our case, we choose Gesture Recognition, as in recent times many gadgets and devices are built with Gesture Recognition Technology in which Image of the user is processed and actions are performed based on their gestures.

This application not only reduces overheads of remotes and buttons but also proves to be cost effective.

This application is implemented on small platform but it's a core concept of Image processing and Image Contouring which is also used in Driverless car's for making car to take turn. Only difference is of implementation of different actions.

## 9.2 DISCUSSION

### 9.2.1. Self Analysis Of Project

**Viabilities**

As is said about the project viability that it states the outcome of a project to be prudent and profitable, comparing with its associated cost, time, quality and manpower requirements.

Undertaking the Self Analysis of our Project Viabilities, we can say that the project seems prudent and profitable once this concept is extended and implemented bases on requirement.

This we can confidently say because of the fact that there is not much of investment in terms of money i.e. Cost and also the time needed to develop such an application is not much with manpower requirement of 2-3 people for the development and 1-2 persons for maintenance.

## 9.2.2. Problem Encountered and Possible Solutions

The major problems faced in the development of the application was how we could use same Gesture and action for different applications.

What if the background was same as the color used for the reference finger.

Except this, no other problems were faced during the development of the currently developed application.

### 9.2.3. Summary of Project Work

To summarize the project work:

- Our project is “” which is a application based on Gesture Recognition and can effectively deployed run on any device having a support of python and camera.
- The project is based over Python and OpenCV.
- The project is developed using the Anaconda software.
- The project starts with camera ready for receiving input.
- A user need to open an application and then use the reference finger wrapped with tape(blue colour object) for performing gestures.
- If the gesture performed by the user is valid then action is performed else an error message is printed.
- Using these technique user can interact with the applications like VLC, Chrome, any racing games etc.
- To wrap up, user’s gesture is recognized by image contouring and based on that contours the actions are performed by the system.

.

# CHAPTER: 10
# BIBLIOGRAPHY

## 10.1 WEBSITES REFERRED


- python.org
- opencv.org
- w3schools.com
- OpenCV with Python for Image and Video Analysis by Sentdex.