A

Mini Project

On

FACE AND HAND GESTURE RECOGNITION SYSTEM FOR CONTROLLING VLC MEDIA PLAYER

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

K. GOHATHI	(197R1A05L7)
K. SAKETH REDDY	(197R1A05L6)
K. PRADEEP	(197R1A05M0)

Under the Guidance of

M.MADHUSUDHAN

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2019-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **“FACE AND HAND GESTURE RECOGNITION SYSTEM FOR CONTROLLING VLC MEDIA PLAYER”** being submitted by **K. GOHATHI (197R1A05L7), K. SAKETH (197R1A05L6) & K. PRADEEP (197R1A05M0)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2022-23.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

M.Madhusudhan
(Associate Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on ___________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **M.Madhusudhan,** Associate Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. Punyaban Patel, Ms. Shilpa, Dr.M . Subha Mastan Rao & J. Narasimharao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion ofthis project.

K.GOHATHI	**(197R1A05L7)**
K.SAKETH REDDY	**(197R1A05L6)**
K.PRADEEP	**(197R1A05M0)**

ABSTRACT

In this project, we are developing an enhanced media player which plays and pauses the video by detecting the users face looking at screen or not and also the field of computer vision-based hand gesture interfaces for Human-Computer Interaction (HCI). System will continuously monitor whether the user is looking at the screen or not using a web camera. If it detects then the video will play without any interruption. Along with these, the web camera will also detect the users hand gestures which can be used for performing various events like play, pause, forward and backward, etc. If the system could not detect user's face then the video will be stopped immediately. Currently we propose to build prototype for exploring the use of marking menus in gesture-based interaction for controlling the Media player.

LIST OFFIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OFCONTENTS

TABLE OFCONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

This project is titled “VLC Media Controlling Using Hand Gesture”. With the development in Computer Vision and Human Machine Interaction the Computer holds most important role in our daily life. Human Computer Interaction can provide several advantages with the introducing the different natural forms of device free communication. Gesture recognition is one of the several types of them to interact with the human’s gestures are the natural form of action which we often used in our day to day life. But in computer application to interact humans with machine the interaction with devices like keyboard, mouse etc. must be requires. As the various hand gestures are frequently used by humans so the aim of this project is to reduce external hardware interaction which is required for computer application, and hence this causes system more reliable for use with ease.

1.2 PROJECT PURPOSE

Hand gesture recognition in aspect to human machine interface is being developed vastly in recent days. Because of the disturbance of lighting and background being not plain, many visual hand gesture recognition systems operate or show successful results only in restricted background. To recognize the various hand gestures, we will build a non-complex and with greater speed motion history image related system. In our system, we mainly focus on applying pointing behavior for the human machine interface. Now days, the gesture recognition has been a new developmental and experimental thing for most of the human related electronics. This system allows people to operate electronic products more conveniently. In our system, a gesture recognition method is to be build which will be an interface between human machine interaction i.e. HMI. In our system we propose some non-complex algorithm and hand gestures to decrease the hand gesture recognition complexity and would be easier and simpler to control real-time computer systems.

1.3 PROJECT FEATURES

The main features of this project is to recognize the hands gesture, pattern and position of hand along with other parts of the body and perform the action as defined in the code and algorithm to perform like playing the video, pausing it, forward and backward, volume up and down

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

A general statement of this project is to understand the hand gesture with body pattern to perform various action with a particular video running through VLC media

.

2.2 EXISTING SYSTEM

In 2015, Chong Wang, “Super pixel-Based Hand Gesture Recognition with Kinect Depth Camera” proposed the system which uses kinect depth camera. It is based on a compact representation in the form of super pixels, which efficiently capture the shape, texture and depth features of the gestures. Since this system uses kinect depth camera, the cost of system is more.

2.2.1 DISADVANTAGES OF EXISTINGSYSTEM

Following are the disadvantages of existing system:

- Hand recognition has some technical issues based on image dataset loading. Delay in finding accuracy of face.
- Huge storage requirements.
- Potential privacy issues

2.3 PROPOSED SYSTEM

The design for the proposed device is to stumble on hand gestures and control the media participant without touching the keyboard. OpenCV is used to get admission to the video from a webcam. PyAutogui is used to configure the media player with the python code. This works properly with VLC media players and as a result, its miles counseled to use VLC. Get the video the usage of the cv2 video capture feature. Then follow blur and try to put off the noise from the historical past; the usage of cv2 gaussian blur. Dilate and erode are used to filter out the history noise. It hit upon a variety of fingers with the usage of angle ninety. Pyautogui is used to combine gestures with the keyboard. Data acquisition: Done by using an in-constructed webcam at the computer. Segmentation: there are kinds of strategies used. Skin detection version for detection of hand place. Skin detection model for detection of hand region. - Approximate median technique for the subtraction of history. Recognition phase: choice tree turned into used as a class tool. Windows interplay: deliver the suitable command to the windows participant in keeping with the regarded gesture

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Users cannot miss any part of the video.
- The video stops as user changes their view from the video thereby no need of users to keep on dragging back to the point from where they missed.
- You can also forward and backward the video if required.
- It saves time and electricity.
- It gives accurate result

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis:

- EconomicFeasibility
- TechnicalFeasibility
- SocialFeasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication that the system is economically possible for development.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible

2.5 HARDWARE & SOFTWAREREQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : Intel Dual Core I5 and above
- Hard disk : 8GB and above
- RAM : 8GB and above
- Input devices : Keyboard, mouse.

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system : Windows 8 and above
- Languages : Python, Html, CSS
- Tools : Python IDEL3.8 version, Anaconda - Jupyter, Spyder

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

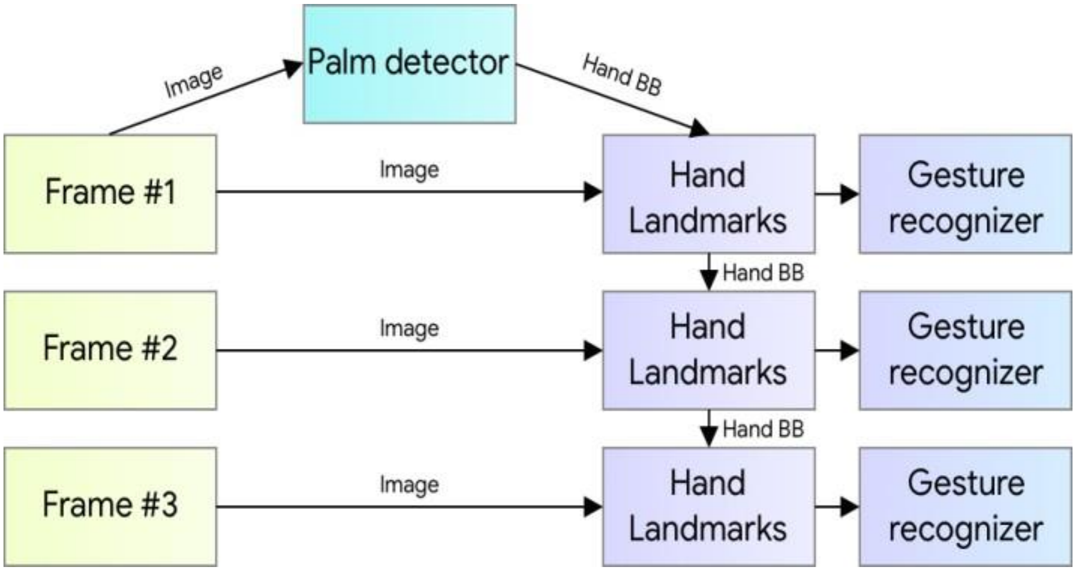


Figure 3.1: Project Architecture of VLC Media Controlling Using Hand Gesture

3.2 DESCRIPTION

Following are the ways through which gesture controlled applications were created in the past:

A. Image Detection and Processing

Image Detection and Processing: OpenCV (Open Source Computer Vision), is one of the most widely used tools for computer vision and image detection and processing tasks. It is used in various applications such as face detection, video capture, motion detection, object disclosure, face mask detection, social distance, and much more. Steps for Image detection and processing which include capturing, resizing, converting, extracting, detecting then finally recognizing.

1. At first, we capture the image in the form of frame from the video. img = cv2.imread(’/image-path, cv2.IMREAD. COLOR)

2. Resize this image frame to 300 X 200.

3.Convert the BGR image into RGB image using the below code line: imgRGB = cv2.cvtColor(img, cv2.COLOR-BGR2RGB)

4. Detecting the boundary points.

5. Extract the Region of Intrest.

6. Recognize the Gesture

B. Gesture Recognition

Gesture Recognition: When the image frame is extracted, The Gesture in the image is recognized according to the calculations based on coordinates of the Media pipe Landmarks. Let us consider the figure 3 which shows the Hand Landmarks defined by Media Pipe over a palm. Here, the ’0th’ landmarks is at the wrist of the palm, Landmarks 1 to 4 are distributed all over the thumb where the ’1st’ landmark is at the base of the thumb and the ’4th’ landmark is at the tip of the thumb; Similarly landmarks 5 to 8 are distributed all over the index finger where ’8th’ landmark is at the tip and ’5th’ is at the base; Landmarks 9 to 12 are distributed all over the middle finger where the ’12th’ landmark is at the tip and ’9th’ is at the base; Landmarks 13 to 16 are distributed all over the ring finger where’13’is at the base and ’16th’ is at the tip, Landmarks 17 to 20 are distributed on the little finger where ’17th’ is at the base and ’20th’ is at the tip. Initially, the landmarks at the tip of the finger have ycoordinate greater than the y-coordinate of the landmark at the base of the finger (eg:The y-coordinate of the 12th landmark is greater than the ycoordinate of the 9th landmark). For the Pause and Play function spacebar is used and the Gesture assigned for that is a closed wrist. Here In this case, when the wrist is closed the tips of the fingers get lowered than the base of the fingers, Thus the landmarks at the tips of the fingers gets lowered than the landmarks at the base of the fingers. So the new y-coordinates of the landmarks at the finger tips become smaller than the y-coordinates of the landmarks at the finger bases. Thus, the closed wrist is detected.

3.3 USE CASE DIAGRAM

In the use case diagram, we have basically one actor who is the user in the trained model.

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

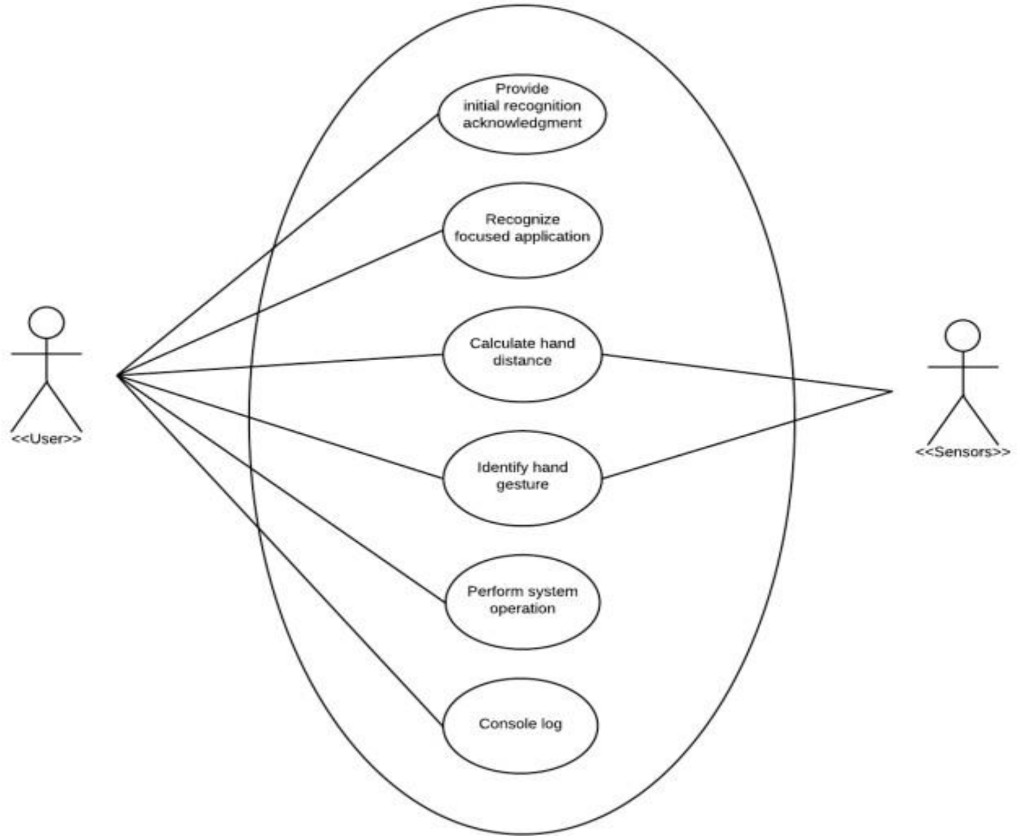


Figure 3.2: Use Case Diagram for Facial Recognition System With Voice Message Enhancement

3.4 CLASS DIAGRAM

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations(or methods), and the relationships among objects.

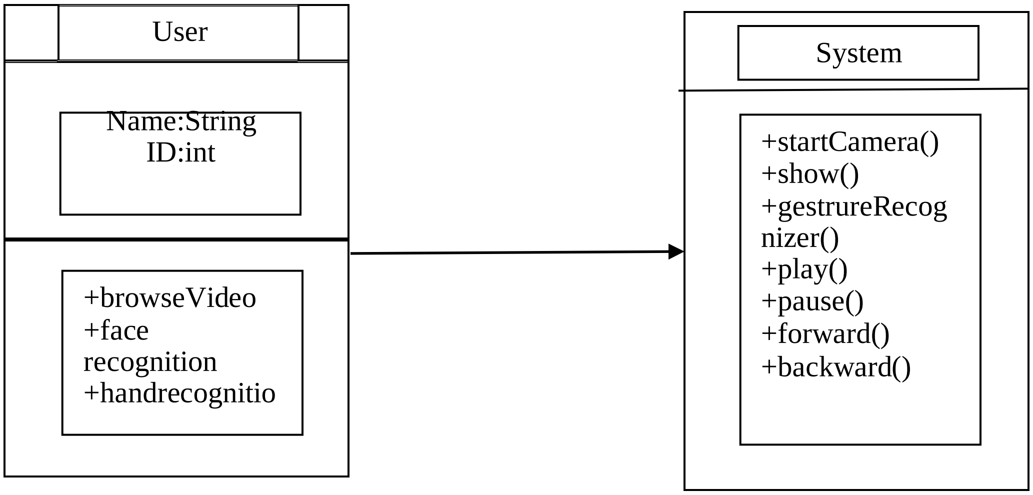


Figure 3.3: Class Diagram for Facial Recognition System With Voice Message Enhancement

3.5 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

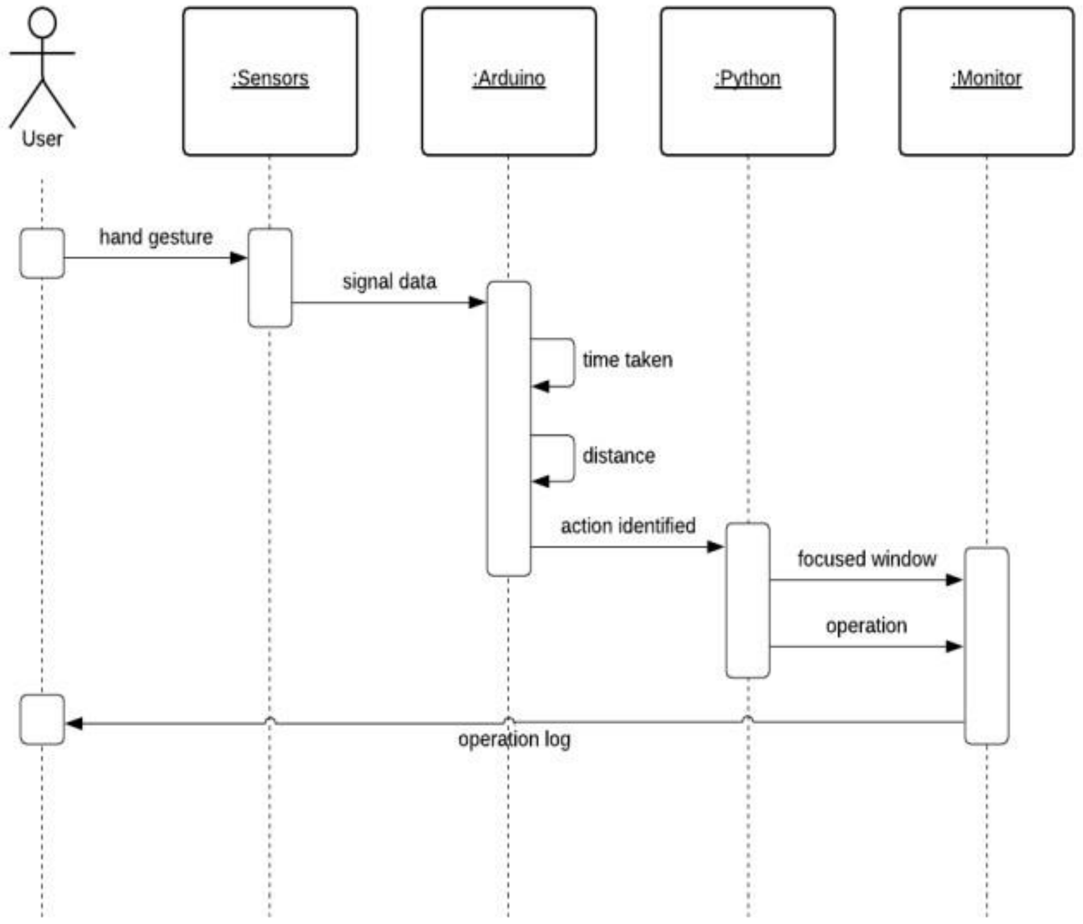


Figure 3.4: Sequence Diagram for Facial Recognition System With Voice Message Enhancement

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data stores.

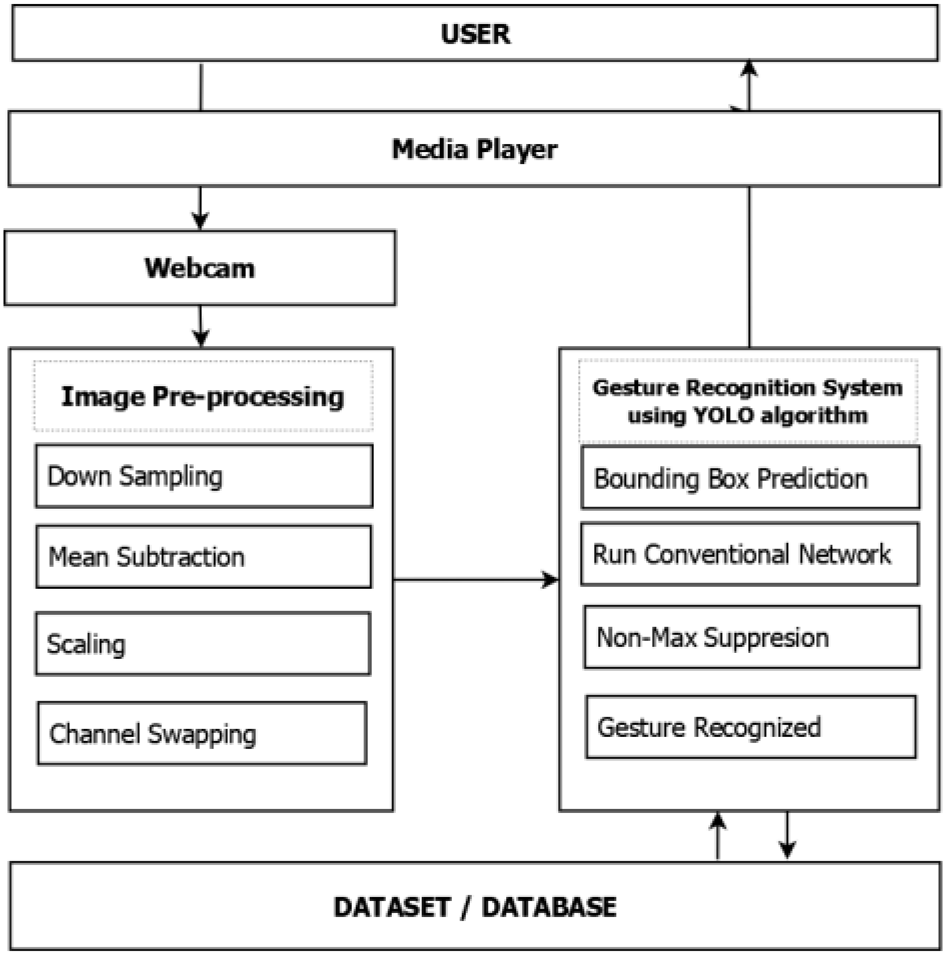


Figure 3.5: Activity Diagram for Facial Recognition System With Voice Message Enhancement

4. IMPLEMENTATION

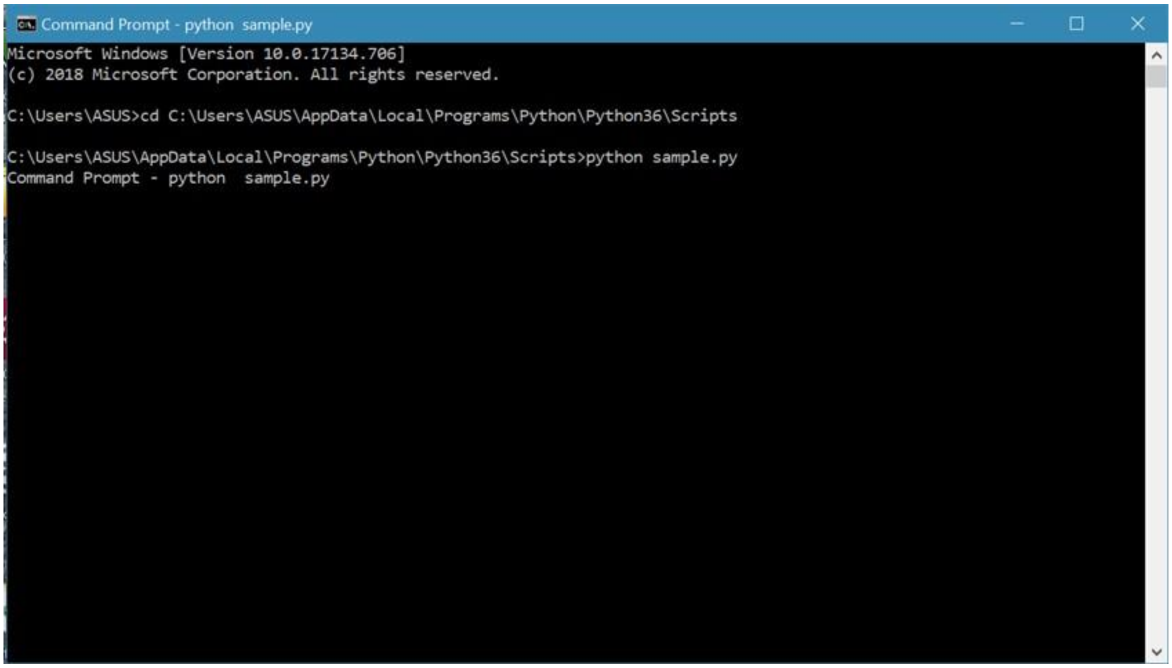
4.1 SAMPLE CODE

```
class Control:
    def___init__(self, media_path):
        self.mp_drawing = mp.solutions.drawing_utils
        self.mp_hands = mp.solutions.hands
        with open('gesture_recognition_model.pkl', 'rb') as f:
            self.model = pickle.load(f)
        self.media_player = vlc.MediaPlayer()
        self.media = vlc.Media(media_path)
        self.media_player.set_media(self.media)
        self.media_player.play()
        time.sleep(2)
        self.value = self.media_player.is_playing()
        self.launch_cam()

# Pause video
    def pause_gesture(self):
        self.media_player.set_pause(1)

# Play video
    def play_gesture(self):
        self.media_player.play()

# Fast forward
    def skip_forwards(self):
        current_time = self.media_player.get_time()
        self.media_player.set_time(current_time + 1000)

# Rewind
    def skip_backwards(self):
        current_time = self.media_player.get_time()
        self.media_player.set_time(current_time - 1000)
```

```
# Increase volume
    def vol_up(self):
        value = self.media_player.audio_get_volume()
        self.media_player.audio_set_volume(value + 1)

# Decrease volume
    def vol_down(self):
        value = self.media_player.audio_get_volume()
        self.media_player.audio_set_volume(value - 1)

    def launch_cam(self):
        # start webcam with opencv
        cap = cv2.VideoCapture(0)
        # Use mediapipe Hands model
        with self.mp_hands.Hands(
                max_num_hands=2,
                min_detection_confidence=0.5,
                min_tracking_confidence=0.5) as hands:
            while cap.isOpened():
                success, image = cap.read()
                if not success:
                    print("Ignoring empty camera frame.")

                    continue

                # Convert to RGB and flip image so that camera output is like mirror.
                image = cv2.cvtColor(cv2.flip(image, 1), cv2.COLOR_BGR2RGB)

                image.flags.writeable = False
                results = hands.process(image)

                # Draw annotations
                image.flags.writeable = True
                image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
                if results.multi_hand_landmarks:
```

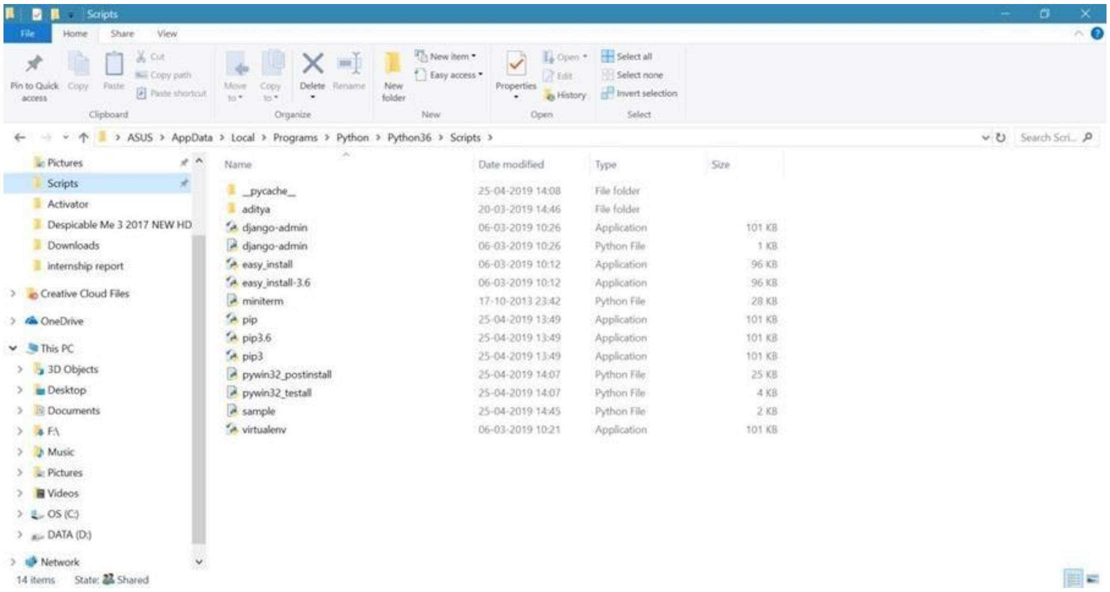
```
            for hand_landmarks in results.multi_hand_landmarks:
                self.mp_drawing.draw_landmarks(
                    image, hand_landmarks, self.mp_hands.HAND_CONNECTIONS)

        try:
            # Get landmarks for hand 1
            hand_1 = results.multi_hand_landmarks[0].landmark
            landmarks_1 = list(
                np.array([[landmark.x, landmark.y, landmark.z, landmark.visibility] for landmark
in hand_1]).flatten())

            # get landmarks for hand 2
            hand_2 = results.multi_hand_landmarks[1].landmark
            landmarks_2 = list(
                np.array([[landmark.x, landmark.y, landmark.z, landmark.visibility] for landmark
in hand_2]).flatten())

            # Concatenate hand 1 and hand 2 data
            data = landmarks_1 + landmarks_2

            # Create pandas dataframe from data
            X = pd.DataFrame([data])

            # Make prediction
            direction_gest = self.model.predict(X)[0]

            # Annotate image with prediction text
            cv2.putText(image, direction_gest.split(' ')[0]
                        , (90, 40), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 2, cv2.LINE_AA)

            # Trigger the different functions based on the predicted gestures
            if direction_gest.split(' ')[0] == "Play":
                self.play_gesture()
            elif direction_gest.split(' ')[0] == "Stop":
                self.pause_gesture()
            elif direction_gest.split(' ')[0] == "Up":
                self.vol_up()
            elif direction_gest.split(' ')[0] == "Down":
                self.vol_down()
            elif direction_gest.split(' ')[0] == "Forwards":
                self.skip_forwards()
            elif direction_gest.split(' ')[0] == "Back":
                self.skip_backwards()

        except:
            pass

        # Show image
        cv2.imshow('Gesture Controller', image)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break

cap.release()
cv2.destroyAllWindows()
```

5. SCREENSHOTS



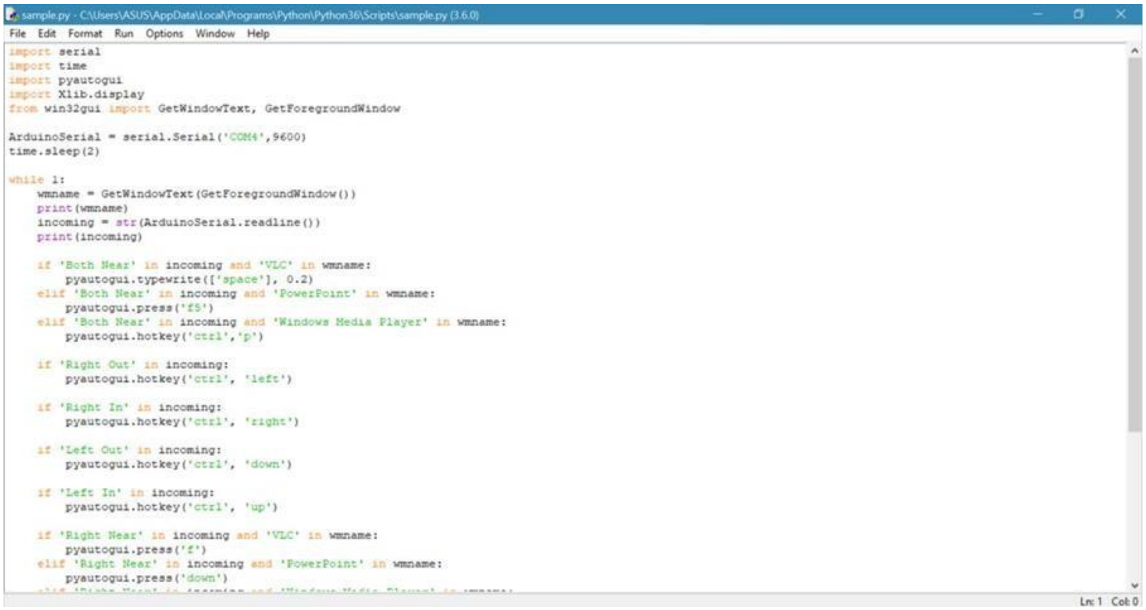
```
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd C:\Users\ASUS\AppData\Local\Programs\Python\Python36\Scripts

C:\Users\ASUS\AppData\Local\Programs\Python\Python36\Scripts>python sample.py
Command Prompt - python  sample.py
```

Screenshot 5.1: Initial Recognition



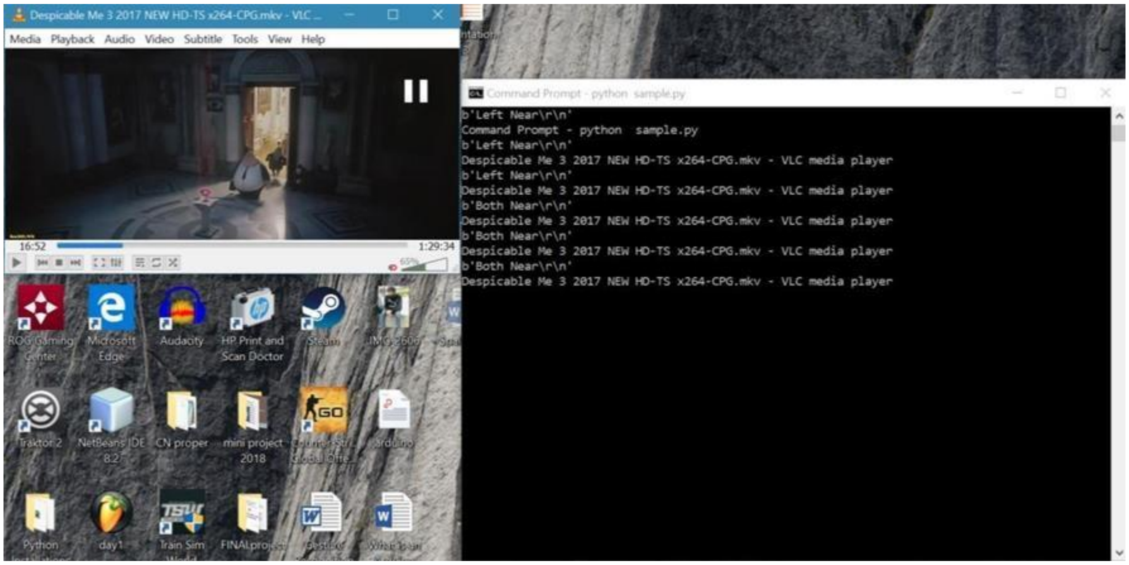
Name	Date modified	Type	Size
__pycache__	25-04-2019 14:08	File folder	
aditya	20-03-2019 14:46	File folder	
django-admin	06-03-2019 10:26	Application	101 KB
django-admin	06-03-2019 10:26	Python File	1 KB
easy_install	06-03-2019 10:12	Application	96 KB
easy_install-3.6	06-03-2019 10:12	Application	96 KB
miniterm	17-10-2013 23:42	Python File	28 KB
pip	25-04-2019 13:49	Application	101 KB
pip3.6	25-04-2019 13:49	Application	101 KB
pip3	25-04-2019 13:49	Application	101 KB
pywin32_postinstall	25-04-2019 14:07	Python File	25 KB
pywin32_testall	25-04-2019 14:07	Python File	4 KB
sample	25-04-2019 14:45	Python File	2 KB
virtualenv	06-03-2019 10:21	Application	101 KB

Screenshot 5.2: Python Main Folder


sample.py - C:\Users\ASUS\AppData\Local\Programs\Python\Python36\Scripts\sample.py (3.6.0)
File Edit Format Run Options Window Help
Ln: 1 Col: 0

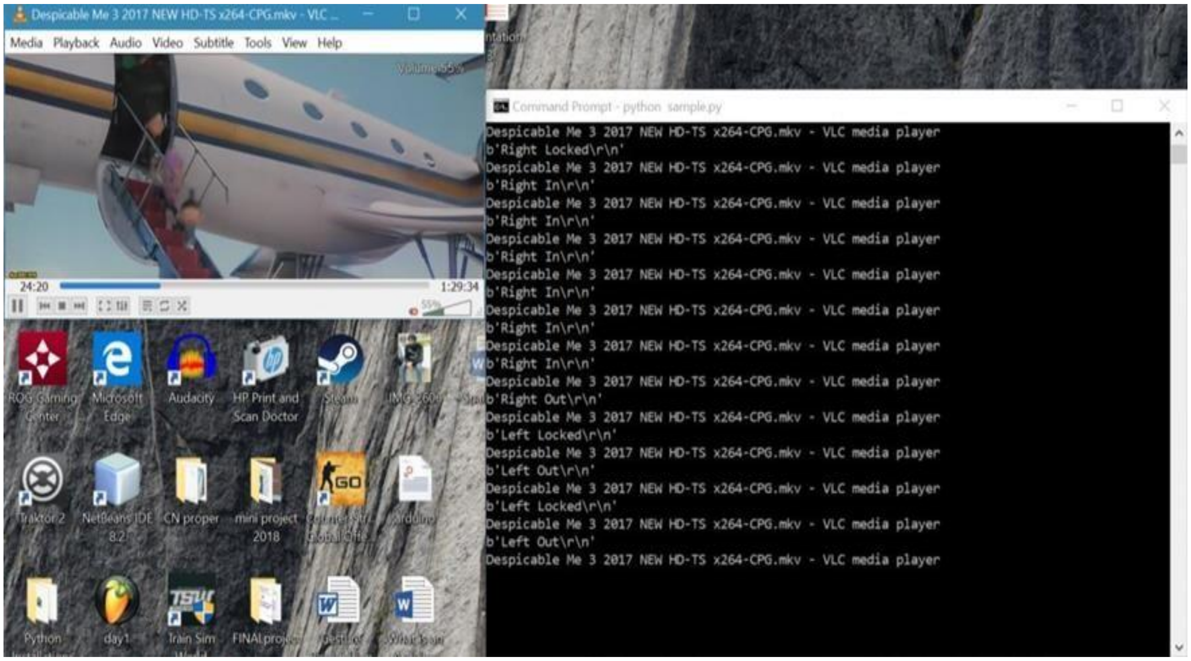
```
import serial
import time
import pyautogui
import Xlib.display
from win32gui import GetWindowText, GetForegroundWindow

ArduinoSerial = serial.Serial('COM4',9600)
time.sleep(2)

while 1:
    wmname = GetWindowText(GetForegroundWindow())
    print(wmname)
    incoming = str(ArduinoSerial.readline())
    print(incoming)

    if 'Both Near' in incoming and 'VLC' in wmname:
        pyautogui.typewrite(['space'], 0.2)
    elif 'Both Near' in incoming and 'PowerPoint' in wmname:
        pyautogui.press('f5')
    elif 'Both Near' in incoming and 'Windows Media Player' in wmname:
        pyautogui.hotkey('ctrl','p')

    if 'Right Out' in incoming:
        pyautogui.hotkey('ctrl', 'left')

    if 'Right In' in incoming:
        pyautogui.hotkey('ctrl', 'right')

    if 'Left Out' in incoming:
        pyautogui.hotkey('ctrl', 'down')

    if 'Left In' in incoming:
        pyautogui.hotkey('ctrl', 'up')

    if 'Right Near' in incoming and 'VLC' in wmname:
        pyautogui.press('f')
    elif 'Right Near' in incoming and 'PowerPoint' in wmname:
        pyautogui.press('down')
```

Screenshot 5.3: Python File



```
b'Left Near\r\n'
Command Prompt - python  sample.py
b'Left Near\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Left Near\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Both Near\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Both Near\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Both Near\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
```

Screenshot 5.4: Pause in VLC Media Player


Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC ...
Media Playback Audio Video Subtitle Tools View Help
24:20
1:29:34
Command Prompt - python sample.py

```
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Right Locked\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Right In\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Right In\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Right In\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Right In\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Right In\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Right In\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Right Out\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Left Locked\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Left Out\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Left Locked\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
b'Left Out\r\n'
Despicable Me 3 2017 NEW HD-TS x264-CPG.mkv - VLC media player
```

Screenshot 5.5: Play in VLC Media Player

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.
Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

6.3 TEST CASES

6.3.1 CLASSIFICATION

Test case ID	Test case name	Purpose	Input	Output
1	Face Recognition	To detect Faces.	The user gives the input in the form of a video using open cv.	An output is showing your face in rectangle box
2	Hand and body pattern Recognition	To detect Hand and Body .	The user walks in a motion towards entry	An output is showing body and hand
3	Perform the action video as per the gesture pattern	To recognize gesture pattern	Body gesture	Show action as output

7. CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

In the current world many resources are available to provide input to any application some require physical touch and some without the use of physical touch (speech, hand touch etc.), the user can manage the system remotely without using the keyboard and mouse. This application provides a novel human computer interface where the user can control the media player (VLC) using hand gestures. The system specific touch to control the VLC player functions. The user will provide a touch as inserted depending on the activity you are interested in. The app provides the flexibility to define a user’s touch of interest with a specific command that makes the app more useful for people with physical disabilities, as they can define touch according to their ability. The system managed to detect the volume down of the Volume Down and detect the action to be performed, so the corresponding action to lower the video volume is active. The program has successfully detected the rewind touch and detected the action to be performed, so the corresponding video rewind action is active

7.2 FUTURE SCOPE

To overcome the drawbacks of the current system, we can modify it for better. While the application is running, if the user brings the hand closer to his/her face, not intending to command the application, it nonetheless recognizes it and accordingly alters the volume or seek controls. To avoid this, we can integrate iris detection to this project to make it run more smoothly. In these times of the Pandemic, where we are cautioned about everything we touch in a public place, this project can be extended to other public service technical systems to avoid direct contact. ATM machines, Ticket Counters, etc can make use of the extended version

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

1. Muhler, Vincent. (2017, Sep 5). Simple Hand Gesture Recognition using OpenCV and JavaScript. Retrieved from https://medium.com/@muehler.v/simple-hand-gesture-recognition-using-opencv-and-javascript-eb3d6ced28a0
2. Circuito Team. (2018, Mar 11). Everything You Need to Know About Arduino Code. Retrieved from https://www.circuito.io/blog/arduino-code/
3. Python Programming Language. (n.d.). Retrieved from https://www.geeksforgeeks.org/python-programming-language/
4. Data-flow diagram. (2019, May 3). Retrieved from https://en.wikipedia.org/wiki/Data-flow_diagram
5. Unit Testing. (n.d.). Retrieved from http://softwaretestingfundamentals.com/unit- testing/

8.2 GITHUB LINK