A Project-I Report

On

Smart Screen Monitoring using ML(SSM)

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTHAPURAMU

In Partial Fulfillment of the Requirements for the Award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

Submitted By

M.Abdul Gani Baig – 19691A0501
S.Ameer Suhail – 19691A0506
R.Arjun Reddy – 19691A0510
R.Hemanth Kumar – 19691A0547

Under the Guidance of

Dr. Mahboob Basha Professor

Department of Computer Science & Engineering

MADANAPALLE INSTITUTE OF TECHNOLGY & SCIENCE

(UGC – AUTONOMOUS)

(Affiliated to JNTUA, Ananthapuramu)

Accredited by NBA, Approved by AICTE, New Delhi)

AN ISO 9001:2008 Certified Institution

P. B. No: 14, Angallu, Madanapalle – 517325

2019-2023

✆ : 08571 - 280255, 280590
MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
An Engineering College Sponsored by :
RATAKONDA RANGA REDDY EDUCATIONAL ACADEMY
Approved by AICTE, New Delhi and Affiliated to JNTUA, Anantapur
P.B. No. 14, Angallu, MADANAPALLE - 517 325. Chittoor Dist. A.P.
www.mits.edu

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the project work entitled **"Smart Screen Monitoring using AI(SSM)"** is a bonafide work carried out by

M.Abdul Gani Baig – 19691A0501
S.Ameer Suhail – 19691A0506
R.Arjun Reddy – 19691A0510
R.Hemanth Kumar – 19691A0547

Submitted in partial fulfillment of the requirements for the award of degree **Bachelor of Technology** in the stream of **Computer Science & Engineering** in **Madanapalle Institute of Technology and Science, Madanapalle,** affiliated to **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** during the academic year 2022-2023

Guide
Dr. Mahboob Basha
Professor,
Department of CSE

Head of the Department
Dr. R. Kalpana
Professor and Head,
Department of CSE

Submitted for the University examination held on:

Internal Examiner
Date:

External Examiner
Date:

ACKNOWLEDGEMENT

We sincerely thank the **MANAGEMENT** of **Madanapalle Institute of Technology and Science** for providing excellent infrastructure and lab facilities that helped me to complete this project.

We sincerely thank **Dr. C. Yuvaraj, M.E., Ph.D**., **Principal** for guiding and providing facilities for the successful completion of our project at **Madanapalle Institute of Technology and Science,** Madanapalle.

We express our deep sense of gratitude to **Dr. R. Kalpana, M. Tech., Ph.D., Professor and Head of the Department of CSE** for her continuous support in making necessary arrangements for the successful completion of the Project.

We express my deep sense gratitude to **Dr. K P Manikandan, Ph.D** , **Project Coordinator and Mrs. M Bommy ,M.E., Project Co Coordinator** for their valuable guidance and encouragement that helped us to complete this project.

We express our deep gratitude to my guide **Dr. Mahboob Basha, PhD.**, **Professor, Department of CSE** for his guidance and encouragement that helped us to complete this project.

We also wish to place on record my gratefulness to other **Faculty of CSE Department** and also to our friends and our parents for their help and cooperation during our project work.

DECLARATION

We hereby declare that the results embodied in this project **" Smart Screen Monitoring using AI(SSM)"** by us under the guidance of **Dr. Mahboob Basha**, **Professor, Dept. of CSE** in partial fulfillment of the award of **Bachelor of Technology** in **Computer Science & Engineering** from **Jawaharlal Nehru Technological University Anantapur, Ananthapur** and we have not submitted the same to any other University/institute for award of any other degree.

Date :
Place :

PROJECT ASSOCIATES:

M.ABDUL GANI BAIG - (19691A0501)
S.AMEER SUHAIL - (19691A0506)
R.ARJUN REDDY - (19691A0510)
R.HEMANTH KUMAR - (19691A0547)

I certify that above statement made by the students is correct to the best of my knowledge.

Date : **Guide**

TABLE OF CONTENTS

ABSTRACT

In this project, we are developing an application which controls the media player which plays and pauses the video by detecting the hand gestures using Mediapipe which is controls video through hand gestures using Mediapipe which internally uses machine learning concepts, here the image frame set which is given as input to the application is dealt by Mediapipe library which helps in finding angles and distance between the fixed points of hand through which we can identify different positions of hand. After identifying patten in accordance to its position respective command is executed.

with the help of PyAutoGUI the very next feature is automation of keyboard using PyAutoGUI which helps to control the media player. Along with these, the web camera will also detect the users hand gestures which can be used for performing various events like increasing or decreasing the volume, changing to next video or previous video, etc. Currently we propose to build prototype for exploring the use of marking menus in gesture-based interaction for controlling the Media player

LIST OF TABLES/FIGURES

CHAPTER-1
INTRODUCTION

1.1 MOTIVATION

While watching a video at some distance if we want to control the media player, we have to go again towards the system to control it. Well, we got a solution to this problem. A media player that gets controlled from some distance using hand gestures that are captured through camera. This enhanced media player can help in minimizing human efforts. In future, this technique can be used to control systems using HCI like pdf reader, power point etc.

- Get better experience of using media player
- We have tried to achieve this goal by automating it to a wide extent.
- We are doing this by hand gestures for controlling varied features of the media player.

1.2 PROBLEM DEFINITION

With the development of ubiquitous computing, current user interaction approaches with keyboard, mouse and pen are not sufficient. Due to the limitation of these devices the usable command set is also limited. Direct use of hands can be used as an input device for providing natural interaction. In this project, we are developing a python program to control media. The web camera will detect the users hand gestures which can be used for performing various events like increasing or decreasing the volume, changing to next video or previous video, etc.

1.3 OBJECTIVE OF THE PROJECT

- It helps the user to interact with the media player without touching the screen.
- This system also provides the feature of controlling other functions of media player such as volume up, volume down, forward and backward using hand gestures.
- The media player pause the video as soon as the user face is not detected.
- The hand gestures should be captured accurately and actions associated to them should performed perfectly.

1.4 LIMITATIONS OF PROJECT

- The user’s hands must be empty and should have all fingers that is user should not be handicapped so as to operate this application
- The camera must have enough standards to recognize the quality of finger’s frames

1.5 ORGANIZATION OF DOCUMENTATION

1.5.1 Feasibility Study

The performance of the project is evaluated, as well as the probability that the system will be beneficial to the organization. The feasibility study's major goal is to determine the technical, operational, and financial feasibility of adding new modules and troubleshooting existing systems. If you give any system unlimited resources and infinite hours, it will work. There are aspects in the feasibility study portion of the preliminary investigation:

Technical Feasibility

Operation Feasibility

Financial Feasibility

Technical Feasibility:

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Is the suggested equipment technically capable of storing the data necessary to operate the new system?
- Will the proposed system provide an adequate response to inquiries, regardless of the number or location of users?

Operation Feasibility:

The software used in the project's operational feasibility comprises user-friendliness, dependability, security, portability, availability, and maintainability.

Economical Feasibility:

Analyze the project's expenses and revenue to see if it's sensible and feasible to execute.

CHAPTER-2
LITERATURE SURVEY

2.1 INTRODUCTION

With the development in Computer Vision and Human Machine Interaction the Computer holds most important role in our daily life. Human Computer Interaction can provides 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 humans 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

2.2 EXISTING SYSTEM

Many various systems have been developed that are being controlled by gesture. These systems consist of games, sign language recognition, all these systems can be controlled by facial gestures, hand gestures can also control mouse. A system was developed in 2012 that recognizes seven various hand gestures consists of various gestures such as up, and down, right, and left, cross and round. Three various modules were built in this system to recognize various hand gestures. Signals by MEMS 3-axes accelerometers were been given as input to the system. The gesture of the hand in three perpendicular directions is been detected by 3 accelerometers and been transmitted to the system by Bluetooth. Segmentation algorithm has been applied and finally the gestures are recognized and compares with the gesturers already been saved in the system. People get daily information about news weather etc with the use of the internet. To get these above information people have to use mouse and keyboard which can be prevented by this system. An article was been presented in 2011 by Ginu Thomas, A Review of Various Hand Gesture Recognition Techniques in which he compared the results achieved by several hand gesture recognitions techniques present. The various techniques used are edges method, and pixel by pixel comparison, orientation histogram. A database has been used that store various static hand gestures inputs. These inputs were the subset of ASL i.e. American sign languages. Filtering of the input image has been done to remove the

noise present in the input image and then segmentation was done to the input image to analyze it.

The input image was then converted into feature vector and then it was compared with the stored, trained set of hand gestures. A system developed by Anupam Agrawal in 2010 had various used hand gestures to operate the VLC media player application. The K nearest neighbour algorithm has been used to recognize the various hand gestures. A VLC media player system that has been controlled by various hand gestures consists of play, and pause, Full screen, and stop, increase volume, and decrease volume features. Lucas Kanade Pyramidical Optical Flow algorithm has been used to detect hand gestures from the input video. This algorithm present in the system detects moving points in the input received by the input video. After this K -MEAN has been used to locate the centre of the hand. Using this centre point also known as centroid of the hand, hand is been recognized. The above mentioned system used a database that consists of various hand gestures and then input image was been compared with this saved image and accordingly the purposed output command was been performed by VLC media player

In 2015, [1] 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.

In 2014, [2] Swapnil D. Badgujar, “Hand Gesture Recognition System” proposed the system which recognize the unknown input gestures by using hand tracking and extraction method. This system is applied to recognize the single gesture. There is assumption of stationary background so that system will have smaller search region for tracking. This system only control mouse with the finger using it on web cam.

In 2014, [3] Viraj Shinde, Tushar Bacchav, Jitendra Pawar and Mangesh Sanap developed “Hand Gesture Recognition System Using Camera”. They focus on

using pointing behaviors for a natural interface to classify the dynamic hand gesture, they developed a simple and fast motion history image-based method. This paper presents low complexity algorithm and gestures recognition complexity and more suitable for controlling real time computer system. It is applicable only for the application Of power point presentation.

In 2014,[4] N. Krishna Chaitanya and R. Janardhan Rao presents "Controlling of windows media player application using hand gesture recognition", this system uses various hand gestures as input to operate the windows media player application. This system uses single hand gestures and its directional motion which defines a particular gesture for the above-mentioned application. In this system decision tree has been used for classification. This system only supports windows media player application and not any others.

In 2012, [5] Ram Rajesh J., Sudharshan R., Nagarjunan D. and Aarthi R., "Remotely controlled PowerPoint presentation navigation using hand gestures" developed the system in which slides of power point presentation are controlled without using any marker and gloves. In this system the developer used the segmentation algorithm for hand detection. After detecting hand calculation is for active figures. If the fingers are not stretched properly while making a gesture then application did not work properly.

In 2012, [6] Ruize Xu, Shengli Zhou and Wen J. Li, "MEMS Accelerometer Based Nonspecific-User Hand Gesture Recognition", had built a system which can recognize various hand gestures such as up, and down

2.3 DISADVANTAGES OF EXISTING SYSTEM

- It fails in accurate recognition of gestures
- Heavy Architecture
- More Computational power
- Low-level Image feature

2.4 PROPOSED SYSTEM

In this proposed system we are implementing two features first one is controlling video through hand gestures using Mediapipe which internally uses machine learning concepts, here the image frame set which is given as input to the application is dealt by Mediapipe library which helps in finding angles and distance between the fixed points of hand through which we can identify different positions of hand.After identifying patten in accordance to its position respective command is executed with the help of PyAutoGUI .the very next feature is automation of keyboard using PyAutoGUI which helps to control the media player

2.5 ADVANTAGES OF PROPOSED SYSTEM

- Easy to use
- Better Results
- predicts the appropriate and accurate controls through gestures**.**

CHAPTER-3
ANALYSIS

3.1 INTRODUCTION

Tom Mitchell states machine learning as “A computer program is said to learn from experience and from some tasks and some performance on, as measured by, improves with experience”. Machine Learning is combination of correlations and relationships, most machine learning algorithms in existence are concerned with finding and/or exploitingrelationship between datasets. Once Machine Learning Algorithms can pinpoint on certain correlations, the model can either use these relationships to predict future observations or generalize the data to reveal interesting patterns. In Machine Learning there are various typesof algorithms such as Regression, Linear Regression, Logistic Regression, Naive Bayes Classifier, Bayes theorem, KNN (K-Nearest Neighbor Classifier), Decision Tress, Entropy, ID3, SVM (Support Vector Machines), K-means Algorithm, Random Forest and etc.,

Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data. This hand tracking solution utilizes an ML pipeline consisting of several models working together:

- A palm detector model (called BlazePalm) that operates on the full image and returns an oriented hand bounding box.
- A hand landmark model that operates on the cropped image region defined by the palm detector and returns high fidelity 3D hand keypoints.
- A gesture recognizer that classifies the previously computed keypoint configuration into a discrete set of gestures.

This architecture is similar to that employed by recently published face mesh ML pipeline and that others have used for pose estimation. Providing the accurately cropped palm image to the hand landmark model drastically reduces the need for data augmentation (e.g., rotations, translation and scale) and instead allows the network to

dedicate most of its capacity towards coordinate prediction accuracy. MediaPipe—an

open-source cross platform framework for building pipelines to process perceptual data of different modalities, such as video and audio. This approach provides high-fidelity hand and finger tracking by employing machine learning (ML) to infer 21 3D keypoints of a hand from just a single frame. Whereas current state-of-the-art approaches rely primarily on powerful desktop environments for inference, our method achieves real-time performance on a mobile phone, and even scales to multiple hands.

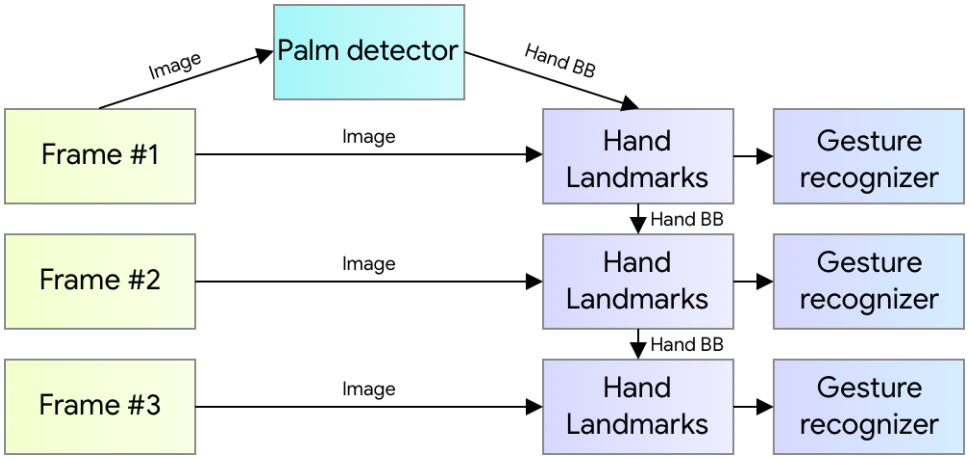

Fig 3.1 Hand Perception pipeline Overview

3.2 FUNCTIONAL REQUIREMENTS

A Functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements (also known quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

3.3 NON-FUNCTIONAL REQUIREMENTS

Non-functional Requirements refer to the constraints or restrictions on the system. They may relate to emergent system properties such as reliability, response time and store occupancy or the selection of language, platform, implementation techniques and tools. The non-functional requirements can be built on the basis of needs of the user, budget constraints,organization policies and etc.

Performance requirement: The user must have all the fingers so as to operate the application.

Platform constraints: The main target is to control media player through automating the keyboard using PyautoGUI

Accuracy and Precision: Requirements are accuracy and precision of the data

Modifiability: Requirements about the effort required to make changes in the software. Often,the measurement is personnel effort (person- months).

Reliability: Requirements about how often the software fails. The definition of a failure mustbe clear. Also, don't confuse reliability with availability which is quite a different kind of requirement. Be sure to specify the consequences of software failure, how to protect from failure, a strategy for error Prediction and a strategy for correction.

Security: One or more requirements about protection of your system and the code.

Usability: Requirements about how difficult it will be to learn and operate the system. Therequirements are often expressed in learning time or similar metrics

Accessibility: Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible. In our project people can have access to system files so as to access the media. User interface is simple and efficient andeasy to use**.**

Maintainability: In software engineering, maintainability is the ease with which a software product can be modified in order to include new functionalities can be added in the project based on the user requirements just by adding the appropriate files to existing project using
.net and programming languages. Since the programming is very simple, it is easier to find and correct the defects and to make the changes in the project.

Scalability: System is capable of handling increase total throughput under an increased loadwhen resources (typically hardware) are added. System can work normally under situations such as low bandwidth and large number of users.

Portability: Portability is one of the key concepts of high-level programming. Portability isthe software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another. Project can be executed underdifferent operation conditions provided it meet its minimum configurations. Only system files and dependent assemblies would have to be configured in such case.

3.4 HARDWARE REQUIREMENTS

- System : Pentium 4, Intel Core i3, i5, i7 and 2 GHz Minimum
- RAM : 512Mb or above
- Hard Disk : 10 GB or above
- Input Device : Keyboard and Mouse
- Output Device : Monitor or PC

3.5 SOFTWARE REQUIREMENTS

- Operating System : Windows 7, 10 or Higher Versions
- Platform : Pycharm

- ❖ Modules : PyautoGUI, Mediapipe, OpenCV
- ❖ Back End : Python and Files
- ❖ Programming Lang. : Python

3.6 SOFTWARE DESCRIPTION

3.6.1 PYTHON

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects. Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's developers strive to avoid premature optimization and reject patches to non- critical parts of CPython that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extensionmodules written in languages such as C, or use PyPy, a just-in-time compiler. Cython is alsoavailable, which translates a Python script into C and makes directC-level API calls into the Python interpreter. An important goal of Python's developers is keeping it fun to use. Python'sdesign offers some support for functional programming in the Lisp tradition. It has filter, map,and reduce functions, list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

3.6.2 BENEFITS OF PYTHON

- Presence of Third-Party Modules
- Extensive Support Libraries
- Open Source and Community Development
- Learning Ease and Support Available
- User-friendly Data Structures
- Productivity and Speed
- Highly Extensible and Easily Readable Language**.**

3.6.3 MEDIAPIPE

With MediaPipe, the perception pipeline can be built as a directed graph of modular components, called Calculators. Mediapipe comes with an extendable set of Calculators to solve tasks like model inference, media processing algorithms, and data transformations across a wide variety of devices and platforms. Individual calculators like cropping, rendering and neural network computations can be performed exclusively on the GPU. For example, we employ TFLite GPU inference on most modern phones.

The graph consists of two subgraphs—one for hand detection and one for hand keypoints (i.e., landmark) computation. One key optimization MediaPipe provides is that the palm detector is only run as necessary (fairly infrequently), saving significant computation time.

We achieve this by inferring the hand location in the subsequent video frames from the computed hand key points in the current frame, eliminating the need to run the palm detector over each frame. For robustness, the hand tracker model outputs an additional scalar capturing the confidence that a hand is present and reasonably aligned in the input crop. Only when the confidence falls below a certain threshold is the hand detection

model reapplied to the whole frame. The purely synthetic data poorly generalizes to the domain. To overcome this problem, we utilize a mixed training schema. A high-level model training diagram is presented in the following figure.

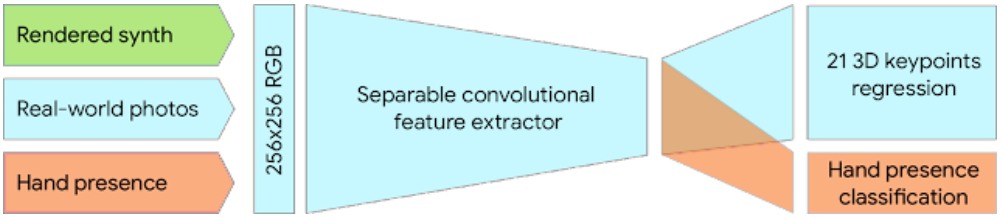

Fig 3.6.3 Mixed training schema

Mixed training schema for hand tracking network. Cropped real-world photos and rendered synthetic images are used as input to predict 21 3D key points.

The table below summarizes regression accuracy depending on the nature of the training data. Using both synthetic and real-world data results in a significant performance boost.

Dataset	**Mean regression error normalized by palm size**
Only real-world	16.1 %
Only rendered synthetic	25.7 %
Mixed real-world + synthetic	**13.4 %**

After the palm detection over the whole image our subsequent hand landmark model performs precise keypoint localization of 21 3D hand-knuckle coordinates inside the detected hand regions via regression, that is direct coordinate prediction. The model learns a consistent internal hand pose representation and is robust even to partially visible hands and self-occlusions.

To obtain ground truth data, it manually annotated ~30K real-world images with 21 3D coordinates, as shown below (we take Z-value from image depth map, if it exists per

corresponding coordinate). To better cover the possible hand poses and provide additional supervision on the nature of hand geometry, we also render a high-quality synthetic hand model over various backgrounds and map it to the corresponding 3D coordinates.

3.6.4 PYAUTOGUI

We humans get bored when we do the same work multiple times. And we always look for the way that can overcome it. So a solution has come before us: we can create something that can do this kind of task with a single click.
Automation plays a huge role in our lives, and it allows us to do one more task during the process.
Python provides many useful and advanced libraries that make a Python programmer's life easier. The pyautogui library is one of the extensive collections of the useful methods. In this tutorial, we will learn about the pyautogui library and implement into the code by using its features. So, without further delay, let's briefly introduce the pyautogui Library.

Python pyautogui library is an automation library that allows mouse and keyboard control. Or we can say that it facilitates us to automate the movement of the mouse and keyboard to establish the interaction with the other application using the Python script. It provides many features, and a few are given below.

- We can move the mouse and click in the other applications' window.
- We can send the keystrokes to the other applications. For example - filling out the form, typing the search query to browser, etc.
- We can also take snapshots and give an image.
- It allows us to locate a window of the application, and move, maximize, minimize, resizes, or close it.
- Display alert and message boxes.

CHAPTER-4

DESIGN

4.1 INTRODUCTION

The Design goals consist of various design which we have implemented in our system disease prediction using machine learning. This system has built with various designs such asdata flow diagram, sequence diagram, class diagram, use case diagram, component diagram,activity diagram, state chart diagram, deployment diagram. After doing these various diagrams and based on these diagrams we have done our project.

We have designed our system in such a way that whenever user wants to control the media player at some distance, the user has to run the application, and any user can use the system without registering in the system. Here are the things that this system can perform.

a. Choosing Media

b. Controlling through hand gestures

Entering Symptoms: Once user successfully runs the program then he/she has to select a media file from the local pc.

Controlling through hand gestures: After opening the media the user has to give input controls so as to control the media.

4.2 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. It shows how data enters and leaves the system, what changes the information, and where data is stored. The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

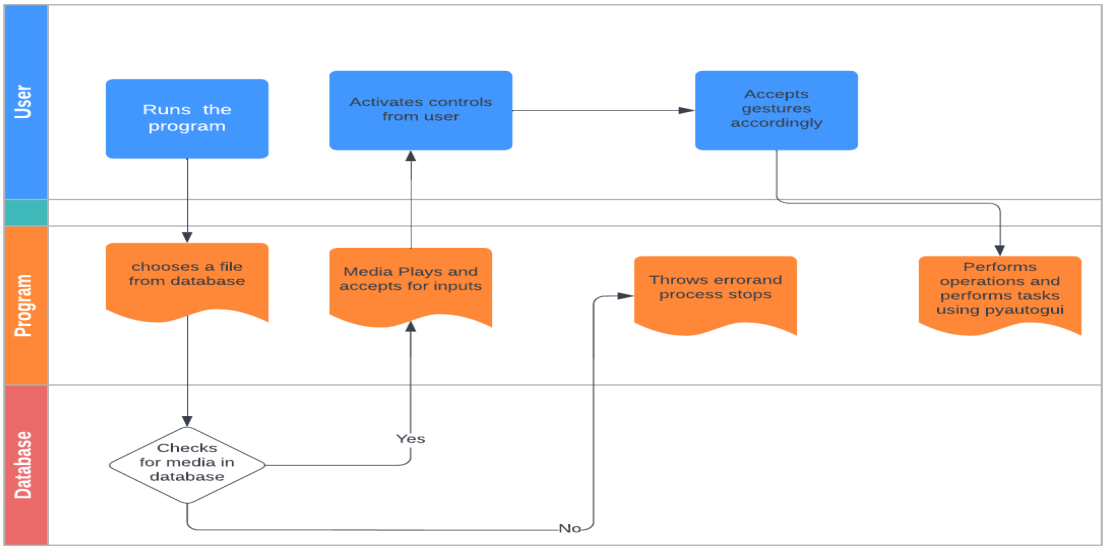

Fig 4.2 Data Flow Diagram

4.3 UML DIAGRAMS

4.3.1 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

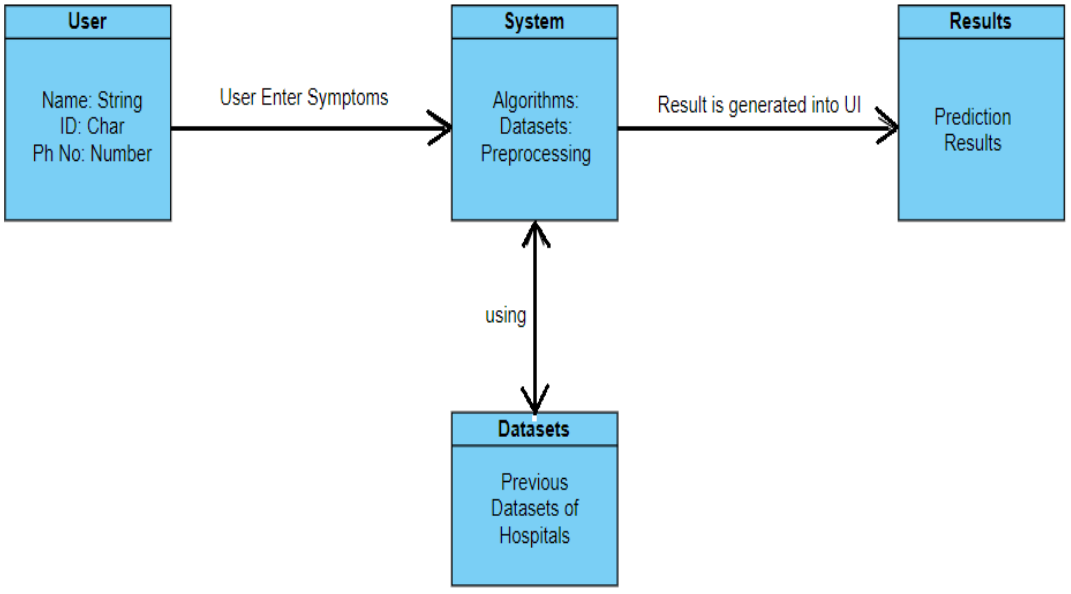

Fig 4.3.1 Class Diagram

4.3.2 SEQUENCE DIAGRAM

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

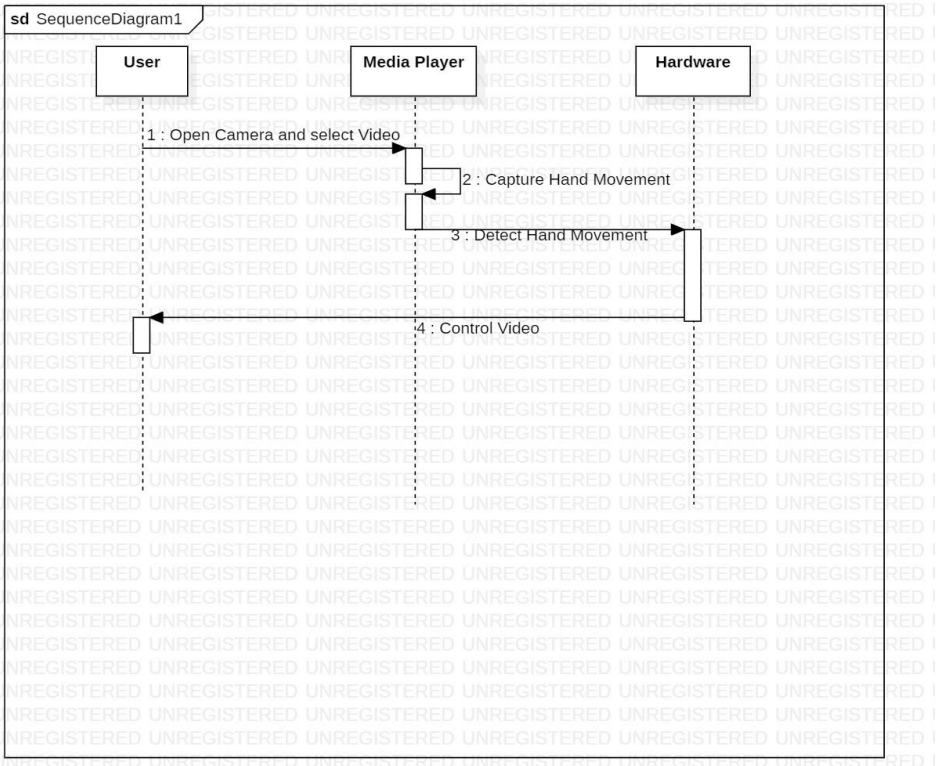

Fig 4.3.2 Sequence Diagram

4.3.3 USE CASE DIAGRAM

The Use Case diagram of the project smart health prediction system using machine learning consist of all the various aspects a normal use case diagram requires. This use case diagram shows how from starting the model flows from one step to another, like he enter into the system then enters all the information's and all other general information along with the symptoms that goes into the system, compares with the prediction model and if true is predictsthe appropriate results otherwise it shows the details where the user if gone wrong while entering the information's and it also shows the appropriate precautionary measure for the user to follow. Here the use case diagram of all the entities are linked to each other where theuser gets started with the system as shown in the Fig 4.3.3.

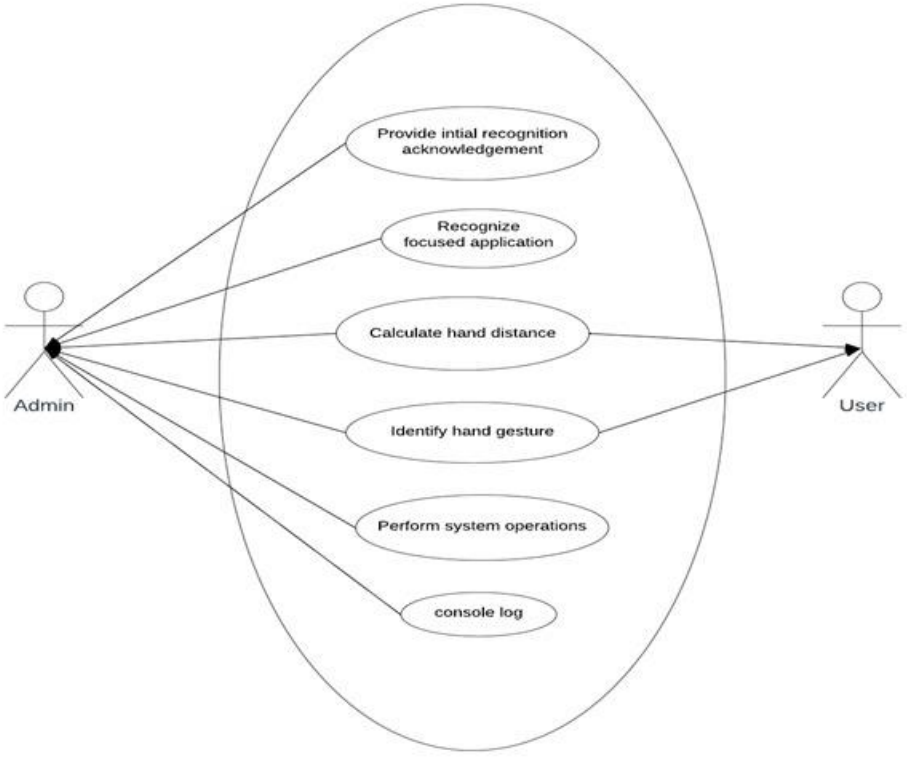

Fig 4.3.3 Use Case Diagram

4.3.4 COLLABORATION DIAGRAM

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

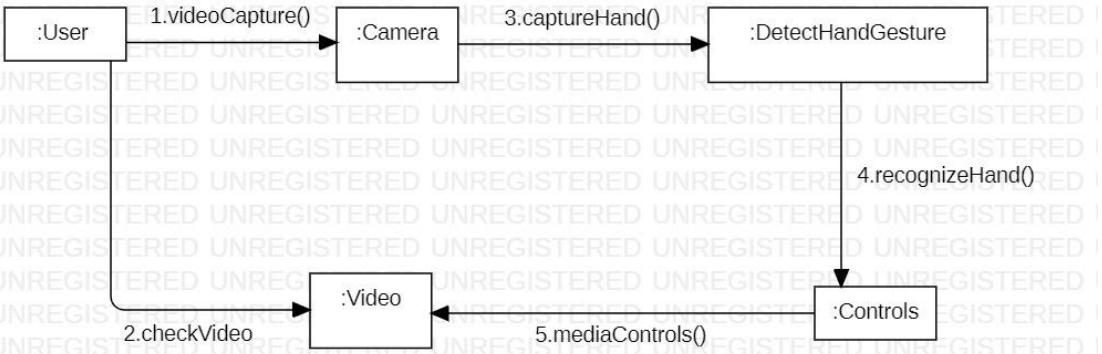

Fig 4.3.4 Collaboration Diagram

4.3.5 DEPLOYMENT DIAGRAM

The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships.It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths Both the deployment diagram and the component diagram are closely interrelated to each other as they focus on software and hardware components. The component diagram represents the components of a system, whereas the deployment diagram describes how they are actually deployed on the hardware

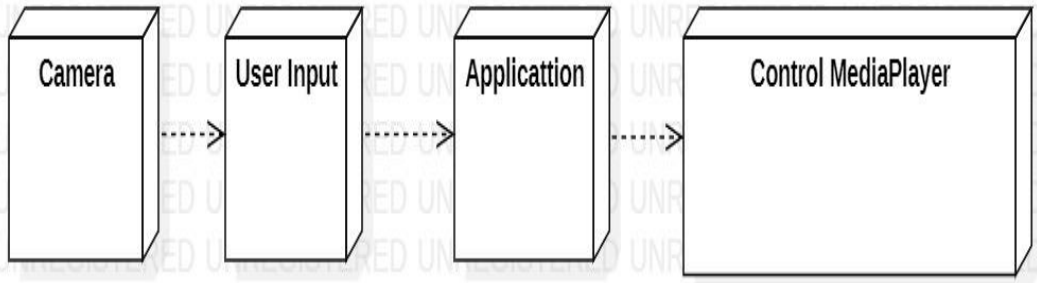

Fig 4.3.5 Deployment Diagram

4.4 CONCLUSION

Based on all the diagrams we can design the required functionalities and the flowof data that is to be maintained between each of them. By doing all this we can maintain the application without any bugs and errors. All the diagrams that are developed show us the functionalities of the model and the application developed on top of it.

CHAPTER-5

IMPLEMENTATION

AND

RESULTS

5.1 INTRODUCTION

The project Smart Screen Monitoring System using PyautoGUI and Media pipe(which internally uses machine learning concepts) is developed to automate media player. The Project “Smart Screen Monitoring System” using machine learning is implemented using python completely. Here firstthe user needs to select a input video after running the program.After choosing the video starts playing through VLC media player and if user wants to control the media player he has to start the application by activating the hand controls by appropriate gesture .After the palm detection over the whole image the subsequent hand landmark model performs precise key point localization of 21 3D hand-knuckle coordinates inside the detected hand regions via regression, that is direct coordinate prediction. The media pipe learns a consistent internal hand pose representation and is robust even to partially visible hands and self-occlusions. To better cover the possible hand poses and provide additional supervision on the nature of hand geometry, we also render a high-quality synthetic hand model over various backgrounds and map it to the corresponding 3D coordinates .Here we use cosine theorem to find angles between each point and distance between two points formula to calculate distance between two points .After giving respective commands the user would be satisfied with the predicted result.

5.2 METHOD OF IMPLEMENTATION

PART-1(MEDIA CONTROLE)

```
import subprocess
import pyautogui
import time
import os
class Controlmedia:
vlcp = "C:/Program Files/VideoLAN/VLC/vlc.exe"
def __init__(self,videoloc):
```

```
    self.videoloc = videoloc
  def Checkvlc(self):
    if (os.path.exists(self.vlcp)):
        print("vlc mediaplayer exist")
    else:
        print("vlc mediaplayer does not exist")
def Checkvideo(self):
    lst=self.videoloc
    lst=lst.split("\\")
    name=lst[len(lst)-1]
    if (os.path.exists(self.videoloc)):
        print("video exist and name is ",name)
    else:
        print("video does not exist or path is invalid
  for ",name)
def openvideo(self):
    p = subprocess.Popen([self.vlcp, self.videoloc])
@staticmethod
def volup():
    pyautogui.press("volumeup")
@staticmethod
def voldown():
    pyautogui.press("volumedown")
@staticmethod
def backward():
    pyautogui.press("left")
@staticmethod
def forward():
    pyautogui.press("right")
@staticmethod
def pause():
```

```
        pyautogui.press(" ")
    @staticmethod
    def play():
        pyautogui.press(" ")
if __name__ == "__main__":
    p1                                                  =
     Controlmedia("D:\\movies\\marvel\\MARVEL\\2
     .captain marvel.mkv")
    p1.Checkvlc()
    time.sleep(3)
    p1.Checkvideo()
    time.sleep(3)
    p1.openvideo()
    time.sleep(3)
    p1.volup()
    time.sleep(3)
    p1.voldown()
    time.sleep(3)
    p1.backward()
    time.sleep(3)
    p1.forward()
    time.sleep(3)
    p1.pause()
    time.sleep(3)
    p1.play()
```

PART-2(MAIN.PY)

```
import cv2 as cv
from matplotlib import pyplot as plt
import mediapipe as mp
from      mediaController.mediaController      import
```

```
    Controlmedia
from        Recognitions.handDetection.handDection
    import HandDetection
from tkinter import filedialog
import time
import threading
if __name__ == '__main__':
    file = filedialog.askopenfile(initialdir="../")
    path = file.name.replace('/','\\')

    print(path)
    mediaController = Controlmedia(path)
    t1                                                  =
     threading.Thread(target=mediaController.openvi
     deo)
    t2                                                  =
     threading.Thread(target=HandDetection().captur
     eHand)
    t2.start()
    time.sleep(1)
    t1.start()
    # mediaController.openvideo()
    # HandDetection().captureHand()
```

PART3(FACEDETECTION)

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
class DetectFace:
    face_rect = None
    def __init__(self, frame) -> None:
```

```
        '''using frontal alt tree to recognize face because
        it is more sharp in finding faces when compared
        with others'''
        harr_cascade                                    =
        cv.CascadeClassifier('public/Recognitions/HaarC
        ascadeFiles/haarcascade_frontalface_default.xml'
        )
        gray_frame                                      =
        cv.cvtColor(frame,cv.COLOR_BGR2GRAY)
        self.face_rect                                  =
        harr_cascade.detectMultiScale(gray_frame,scale
        Factor=1.1,minNeighbors=4)
    '''it will draw rectangle around the faces'''
    def drawFaces(self):
        for (x,y,w,h) in self.face_rect:

        cv.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),thic
        kness=2)
    def isThereFace(self):
        return (True if len(self.face_rect)>0 else False)
    def countFaces(self):
        return len(self.face_rect)
    def areaFaces(self):
        return [face[2]*face[3] for face in self.face_rect]
if __name__ == '__main__':
    cap = cv.VideoCapture(0)
    while True:
        isTrue,frame = cap.read()

        FacesDetected = DetectFace(frame)
        FacesDetected.drawFaces()
```

```
        cv.imshow("face",frame)
        if cv.waitKey(10) & 255 == ord('q'):
            break
    cap.release()
    cv.destroyAllWindows()
    print("form main")
```

PART4(HANDDETECTION)

```
import cv2 as cv
import mediapipe as mp
from matplotlib import pyplot as plt
import numpy as np
import math
from mediaController.mediaController import
    Controlmedia
class HandDetection:
    def __init__(self):
        self.mp_drawingUtils =
    mp.solutions.drawing_utils
        self.mp_hands = mp.solutions.hands
        self.video_status = True
        self.cap = cv.VideoCapture(0)
        self.x_res = 640
        self.y_res = 480
        self.COLOR_FOR_CONTROLLS_STATUS =
    (255,0,102)
        self.COLOR_FOR_PAUSE_AND_PLAY =
    (0,0,0)
        self.COLOR_FOR_VOLUME_CONTROLS =
    (8,171,177)
```

```
 self.COLOR_FOR_FARWARD_AND_BACK
 WARD_CONTROLS = (67,3,177)


 self.POSITION_FOR_CONTROLLS_STATUS
 = (self.x_res-320,self.y_res-50)
   self.POSITION_FOR_PAUSE_AND_PLAY  =
(50,50)
   self.POSITION_FOR_VOLUME_CONTROLS
 = (self.x_res-350,50)

 self.POSITION_FOR_FARWARD_AND_BAC
 KWARD_CONTROLS = (self.x_res-200,90)
   self.activateControlsHelper = 3

@staticmethod
def forward():
   Controlmedia.forward()
@staticmethod
def backward():
   Controlmedia.backward()
@staticmethod
def pause():
   Controlmedia.pause()
@staticmethod
def play():
   Controlmedia.play()
@staticmethod
def volumeUp():
   Controlmedia.volup()
@staticmethod
```

```
def volumeDown():
    Controlmedia.voldown()
def checkForControlsPause(self,angles):
    return (all([True if angle >=150 else False for angle in angles[1:3]]) and all([True if angle <=70 else False for angle in angles[3:5]]))
def checkForControlsActive(self,angles):
    return (all([True if angle >=150 else False for angle in angles[1:4]]) and all([True if angle <=70 else False for angle in angles[4:5]]))
def checkForPause(self,angles,result): #return true if need to pause any video else play
    pointsOfThumb = self.getRequiredPoints([4,2,0],result)
    a = self.angleBetweenThreePoints(pointsOfThumb[0],pointsOfThumb[1],pointsOfThumb[2])

    return (all([True if angle <=50 else False for angle in angles[1:]]) and a<160)
def checkForPlay(self,angles,result):
    return (all([True if angle >=100 else False for angle in angles[1:]]))
def checkForVolume(self,angles): # returns true if firstTwo fingers are opened and last three fingers are closed
    lastThreeClosed = all([True if angle <=70 else False for angle in angles[2:]])
    firstTwoOpened = all([True if angle >=50 else False for angle in angles[:2]])
```

```
    return lastThreeClosed and firstTwoOpened

def videoSkipControl(self,angles,result):
    pointsOfThumb =
 self.getRequiredPoints([4,2,0],result)
    a =
 self.angleBetweenThreePoints(pointsOfThumb[0
 ],pointsOfThumb[1],pointsOfThumb[2])

    return (all([True if angle <=50 else False for
 angle in angles[1:]]) and a>=160)

def
 distanceBetweenTwoPoints(self,point1,point2):
    return math.sqrt(math.pow(point1[0]-
 point2[0],2)+math.pow(point1[1]-point2[1],2))

def whichHand(self,result):
    if
 result.multi_handedness[0].classification[0].label
 .upper()=="LEFT":
        return "LEFT"
    else:
        return "RIGHT"

def
 angleBetweenThreePoints(self,point1,point2,poi
 nt3):

 a=self.distanceBetweenTwoPoints(point1,point2)
```

```
 b=self.distanceBetweenTwoPoints(point2,point3)

 c=self.distanceBetweenTwoPoints(point3,point1)

   try:
      cal = ((a*a+b*b-c*c)/(2*a*b))
      return math.degrees(math.acos(cal))
   except:
      return 90.0

def getRequiredPoints(self,joints,result):
   return
 [(int(result.multi_hand_landmarks[0].landmark[j
 oint].x*self.x_res),int(result.multi_hand_landmar
 ks[0].landmark[joint].y*self.y_res)) for joint in
 joints]
def findAnglesForEveryFinger(self,result):
   finger1 = self.getRequiredPoints([2,3,4],result)
   finger2 = self.getRequiredPoints([5,6,8],result)
   finger3                                          =
 self.getRequiredPoints([9,10,12],result)
   finger4                                          =
 self.getRequiredPoints([13,14,16],result)
   finger5                                          =
 self.getRequiredPoints([17,18,20],result)
   fAngle1                                          =
 self.angleBetweenThreePoints(finger1[0],finger1
 [1],finger1[2])
   fAngle2                                          =
 self.angleBetweenThreePoints(finger2[0],finger2
 [1],finger2[2])
```

```
  fAngle3 =
self.angleBetweenThreePoints(finger3[0],finger3
[1],finger3[2])
  fAngle4 =
self.angleBetweenThreePoints(finger4[0],finger4
[1],finger4[2])
  fAngle5 =
self.angleBetweenThreePoints(finger5[0],finger5
[1],finger5[2])
  return
[fAngle1,fAngle2,fAngle3,fAngle4,fAngle5]
def RecognizeHand(self,frame):
  with
self.mp_hands.Hands(min_detection_confidence
=0.8,min_tracking_confidence=0.5,max_num_ha
nds=1) as hands:
    image_rgb =
cv.cvtColor(frame,cv.COLOR_BGR2RGB)

    results = hands.process(image_rgb)

    # image_bgr =
cv.cvtColor(image_rgb,cv.COLOR_RGB2BGR)

    if results.multi_hand_landmarks:
      for hand in results.multi_hand_landmarks:

self.mp_drawingUtils.draw_landmarks(frame,ha
nd,self.mp_hands.HAND_CONNECTIONS,self.
mp_drawingUtils.DrawingSpec(color=(121,22,7
6),thickness=2,circle_radius=4),self.mp_drawing
```

```
 Utils.DrawingSpec(color=(250,44,250),thickness
 =2,circle_radius=4))
            return results
def captureHand(self):
    video_status = "play"
    last_distance = 9999
    curr_volume = 100
    pauseCount = 0
    playCount = 0
    while True:
        _,frame = self.cap.read()
        frame = cv.flip(frame,1)
        self.x_res = frame.shape[1]
        self.y_res = frame.shape[0]
        result = self.RecognizeHand(frame)
        if(self.activateControlsHelper == 3):
            cv.putText(frame,"controlls                are
 active",self.POSITION_FOR_CONTROLLS_ST
 ATUS,cv.FONT_HERSHEY_SIMPLEX,1,self.
 COLOR_FOR_CONTROLLS_STATUS,2)
        elif(self.activateControlsHelper == 0):
            cv.putText(frame,"controlls                are
 stoped",self.POSITION_FOR_CONTROLLS_S
 TATUS,cv.FONT_HERSHEY_SIMPLEX,1,self.
 COLOR_FOR_CONTROLLS_STATUS,2)

        if result is not None:
            angles                                    =
 self.findAnglesForEveryFinger(result)
            if(self.checkForControlsPause(angles)):
                self.activateControlsHelper = 0
```

```
        if(self.checkForControlsActive(angles)):
            self.activateControlsHelper = 3

        if(self.activateControlsHelper == 3):
            if(self.checkForVolume(angles)):
                points = self.getRequiredPoints([4,8],result)
                distance = self.distanceBetweenTwoPoints(points[0],points[1])
                if(distance>last_distance):
                    # curr_volume+=1
                    HandDetection.volumeUp()
                    cv.putText(frame,f'volume : increasing...',self.POSITION_FOR_VOLUME_CONTROLS,cv.FONT_HERSHEY_SIMPLEX,1,self.COLOR_FOR_VOLUME_CONTROLS,2)
                elif(distance<last_distance):
                    # curr_volume-=1
                    HandDetection.volumeDown()
                    cv.putText(frame,f'volume : decreasing..',self.POSITION_FOR_VOLUME_CONTROLS,cv.FONT_HERSHEY_SIMPLEX,1,self.COLOR_FOR_VOLUME_CONTROLS,2)
                last_distance=distance
            if(self.checkForPause(angles,result)):
                if(video_status=='pause' and pauseCount==1):
                    HandDetection.pause()
```

```
                pauseCount+=1
                playCount=0
                video_status = "pause"
            if(self.checkForPlay(angles,result)):
                if(video_status=='play' and playCount
== 1):
                    HandDetection.play()
                playCount+=1
                pauseCount=0
                video_status = 'play'
            if(self.videoSkipControl(angles,result)):
                if(self.whichHand(result)=='LEFT'):

cv.putText(frame,"forwarding...",self.POSITION
_FOR_FARWARD_AND_BACKWARD_CON
TROLS,cv.FONT_HERSHEY_SIMPLEX,1,self.
COLOR_FOR_FARWARD_AND_BACKWAR
D_CONTROLS,2)
                    HandDetection.forward()
                else:

cv.putText(frame,"backward...",self.POSITION_
FOR_FARWARD_AND_BACKWARD_CONT
ROLS,cv.FONT_HERSHEY_SIMPLEX,1,self.C
OLOR_FOR_FARWARD_AND_BACKWARD
_CONTROLS,2)
                    HandDetection.backward()

cv.putText(frame,video_status,self.POSITION_F
OR_PAUSE_AND_PLAY,cv.FONT_HERSHE
Y_SIMPLEX,1,self.COLOR_FOR_PAUSE_AN
```

```
    D_PLAY,2)


        cv.imshow("frame",frame)

        if cv.waitKey(10) & 255 == ord('q'):
            break
      self.cap.release()
      cv.destroyAllWindows()
if __name__ == '__main__':
   HandDetection().captureHand()
```

PART5(MAIN.PY)

```
import cv2 as cv
from matplotlib import pyplot as plt
import mediapipe as mp
from mediaController.mediaController import
    Controlmedia
from Recognitions.handDetection.handDection
    import HandDetection
from tkinter import filedialog
import time
import threading
if __name__ == '__main__':
   file = filedialog.askopenfile(initialdir="../")
   path = file.name.replace('/','\\')

   print(path)
   mediaController = Controlmedia(path)
   t1 =
    threading.Thread(target=mediaController.openvi
```

```
deo)
t2                                                  =
threading.Thread(target=HandDetection().captur
eHand)
        t2.start()
       time.sleep(1)
       t1.start()
        # mediaController.openvideo()
        # HandDetection().captureHand()
```

5.3 OUTPUT SCREENS AND RESULT ANALYSIS

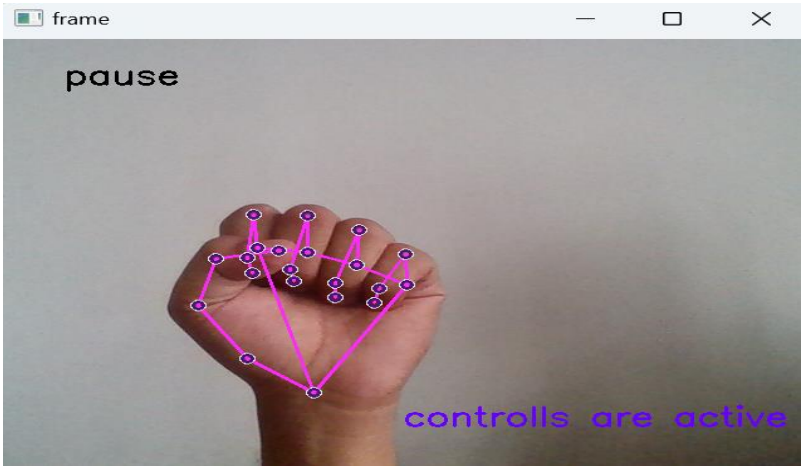

Fig:5.3.1:Pause Operation

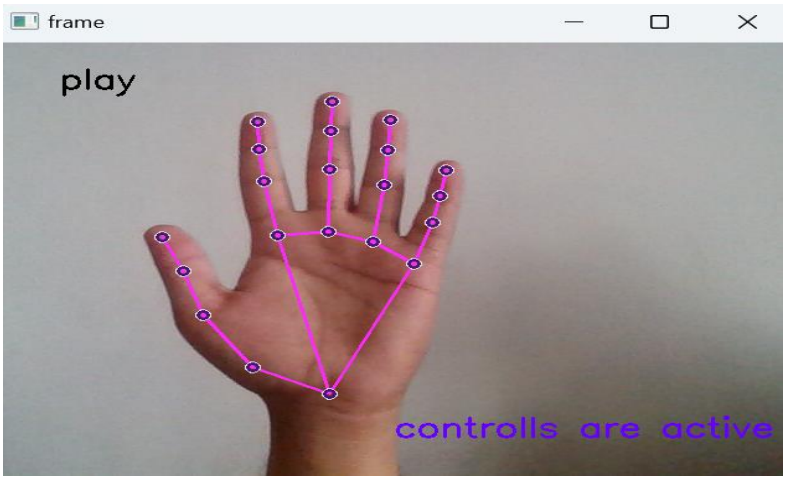

Fig:5.3.2:Play Operation

Fig:5.3.3:play Forward Operation

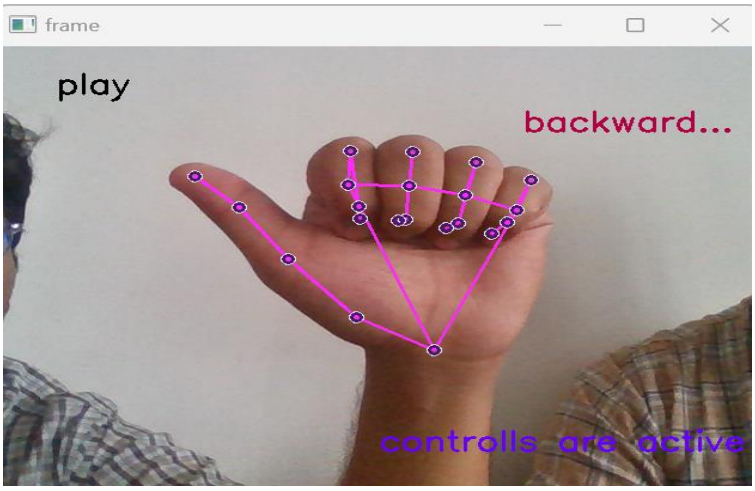

Fig:5.3.4:Play Backward Operation

Fig:5.3.5:Volume Increasing

Fig:5.3.6:Volume Decreasing

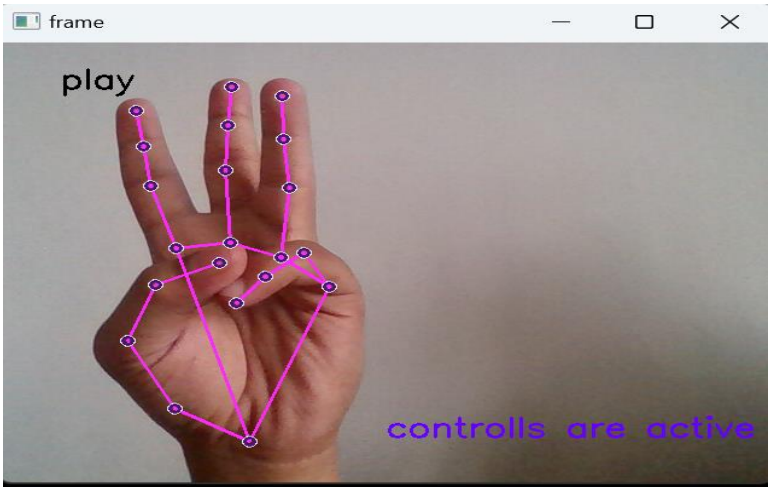

Fig:5.3.7:Activating Controls

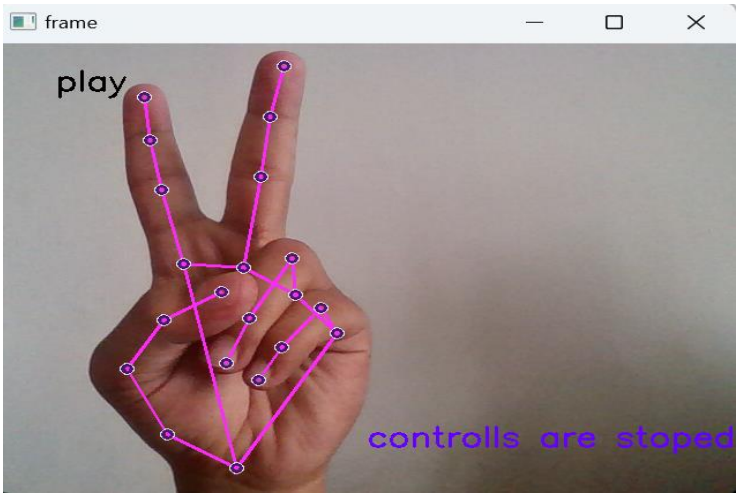

Fig:5.3.8:Deactivating Controls

CHAPTER-6

TESTING

AND

VALIDATION

6.1 INTRODUCTION

INTRODUCTION TO TESTING

Testing is a process, which reveals the errors in the program. It is the major quality measure employed during software development process. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to performor not. To make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development.

UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logicis functioning properly, and that program inputs produce valid outputs. All decisionbranches and internal code flow should be validated. It is the testing of individual softwareunits 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 isinvasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration.

INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine ifthey run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the componentswere individually satisfaction, as shown by successfully unit testing, the combination ofcomponents is correct and consistent. Integration testing is specifically aimed at exposingthe problems that arise from the combination of components.

SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integratedsystem to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

6.2 VALIDATION TESTING

An engineering validation test (EVT) is performed on first engineering prototypes, to ensure that the basic unit performs to design goals and specifications. It is important in identifying design problems and solving them as early in the design cycle as possible,is the key to keeping projects on time and within budget. Too often, product design and performance problems are not detected until late in the product development cycle — when the product is ready to be shipped. The adage holds true: It costs a penny to make a change in engineering, a dime in production and a dollar after a product is in the field.

Recognition rate of different gestures :

Gesture	Recognition Rate (%)
Play - Pause	95%
Volume -Increase	90%
Forword - Backword	95%

6.3 DESIGN OF TEST CASES AND SCENARIOS

Module Under Test	Main.py mediaController.py faceDetection.py handDetection.py
Description	User needs to select the input as a video and can check the Accuracy
Input	Video and hand gestures
Output	If user start to show the gesture through camera of laptop and according to the input the action will be performed if the user is in front of camera within 10-20 mts distance range.
Remarks	Test Successful.

Table 6.1 Test Case for control

6.4 CONCLUSION

We set out to create a system which can control the media player using media pipe on the basis of hand gesture commands given to it. Such a system can automate the media player and can provide assistance to drivers by different gestures for each specific cause. Another strategy for hand sign detection is presented in the proposed method. The hand area is identified from the background by the background subtraction technique. At that point, the palm and fingers are divided. On an average we achieved accuracy of 94%. Such a systemcan be largely reliable to do the job. Creating this system, we can also extend this model for complex high-end application using eye gestures. Our system also has an easy-to-use interface.

CHAPTER-7

CONCLUSION

7.1 CONCLUSION

In this project we aim to help the user get better experience of using advance media player and is very much usefulin everyone's day to day life. We are doing this by using hand gestures recognition and face detection for controlling features of the media player such as playing the video and pausing when the user is not looking at the screen and controlling functions as volume up and volume down, playing next and previous video. This can be successfully used in the real world applications. Depend upon the gesture given by the user we can control media player using various operations through media pipe. The Media Pipe graph uses a hand landmark tracking subgraph from the hand landmark module, and renders using a dedicated hand renderer subgraph . The hand landmark tracking subgraph internally uses a hand landmark subgraph from the same module and a palm detection subgraph from the palm detection module.

7.2 FUTURE ENHANCEMENT

- Develop an Application.
- More interactive user interface.
- Control Using Eye.
- Facial Recognize.
- Can be done as Mobile Application.

REFERENCES

[1] Chong Wang, Sabiha Pathan, Neha Rokade at Uma Annamalai “Hand Gesture Recognition System For Multimedia Applications”, International Research Journal of Engineering and Technology (IRJET) Volume: 03 Issue: 04 | Apr-2016, Apr2016.

[2] Swapnil D. Badgujar , Amrapali Waghmare , Reshma Ganjewar , Dr. Abhijit Banubakode “A Media Player which operates depending on Human Emotions” International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 5, May 2015.

[3] Viraj Shinde, Tushar Bacchav, “Emotion Detection Using Facial Expression”, Journ. International Journal Of Advanced ResearchIn Computer Science & Software Engineering, Vol. 4, PP-465467, April 2016.

[4] N. Krishna Chaitanya and R. Janardhan Rao, “Speech Recognition using FIR Wiener Filter”, International Journal of Application or Innovation in Engineering & management (IJAIEM),pp.204-20,2013.[6] Madhu N: Note on measures for spectral flatness. Electron. Lett 2009,45(23):1195–1196 Khoa PC: Noise robust voice activity detection. Master’s thesis, NangYang Technological University, 2012

[5] In 2012, Ram Rajesh J., Sudharshan R., Nagarjunan D. and Aarthi R., “Remotely controlled PowerPoint presentation navigation using hand gestures” developed the system Viola and Jones, "Rapid object detection using a boosted cascade of simple features", Computer Vision and Pattern Recognition,

[6] In 2012, Ruize Xu, Shengli Zhou and Wen J. Li, “MEMS Accelerometer Based Non specific-User Hand Gesture Recognition”, had build a system which can recognize various hand gestures such as up, and down