# PROJECT REPORT

# ON

# VOLUME CONTROL

## DEPARTMENT OF COMPUTER APPLICATION(MCA)


**Graphic Era Hill University**

**http://www.gehu.ac.in**

**Society Area Road, Clement Town**

**Dehradun, Uttarakhand 248002**

**Submitted to:**
Assistant Professor
Deepti Ma'am

**Submitted by:**
Saurabh Suman
MCA 'c'
Semester: 2
University Roll no: 2001135

# BONAFIDE CERTIFICATE

This is to certify that this project report entitled Volume Control submitted to Graphic Era Hill University, Dehradun is a bonafied record of work done by **Saurabh Suman** bearing **University Roll Number - 2001135** who carried out the project under my supervision.

**SIGNATURE(HOD)**

**Mr. Naveen Garg**

**HEAD OF THE DEPARTMENT**

**SIGNATURE**

**Deepti Ma'am**

**SUPERVISOR**

**Assistant Professor**

**Graphic Era Hill University**

**Society Area Road, Clement Town**

**Dehradun, Uttarakhand 248002**

# ABSTRACT

The project is a first step for an application using computer vision for hand recognition. While showing the palm in front of the camera its reads the palm and read the finger position and calculate the distance between them and convert it according to volume.

Gesture Recognition is the most favored and practicable solution to improve human interaction with the computer and has been widely accepted in recent years by the gaming devices like Xbox etc and other devices like laptop, smartphone and many more.

Volume Control is a project that can be useful in many devices in upcoming devices just for giving instruction to that device just by showing the fingers in front of its camera.

TABLE OF CONTENTS

# CHAPTER 1

# PROJECT INTRODUCTION

## 1.1 OBJECTIVE

The purpose of "Volume Control " is to check the distance between the two finger which is feed already in the program. It calculates the distance between the finger and convert it to the volume. A computer, unlike the human eye, processes images in two dimensions. Furthermore, the object's size, direction in space, attitude, and position in the environment are all factors to consider.

## 1.2 INTRODUCTION

This project is about the Volume Control. With the help of this project, just by showing the palm in front of camera, we can read the palm we read the specific finger which is feed in the program to calculate and give the accurate output according to it. For this we need a different library to make it work. All libraries have a different functionality.

## 1.3 PURPOSE AND NEED

Biometric technologies make use of various physical and behavioral characteristics of human such as fingerprint, hand gesture, face etc. These features can be processed and can be used for security purpose.
In the upcoming time we are moving toward the gesture control in the digitalize era so this project is a first step toward learning the advance version of this we are already seeing the multiple things with the gesture control like car, tv, and many more and finger counter will act as an medium between the human and device as to instruct the device.

## 1.4 SOFTWARE REQUIREMENTS

The technologies, software used in this project are mentioned below:-

1. OpenCV library: OpenCV is the huge open-source library for computer vision; we are using this library in order to process images and videos to identify objects, faces, or even the handwriting of a human.

2. NumPy: This library I am using in this project for a very basic purpose to get the current time. This module of the python provides many functions related to the Time. It also provides functionality other than representing time, like waiting during code execution and measuring the efficiency of your code.

3. PyCaw: Pycaw is Python Core Audio Windows Library. It works with python2 and python3. Its help to read the volume of the system and helps to set the volume according to our need.

4. PyCharm: PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

3. Media Pipe

4. Python 3.7

# CHAPTER 2

# SYSTEM DESIGN

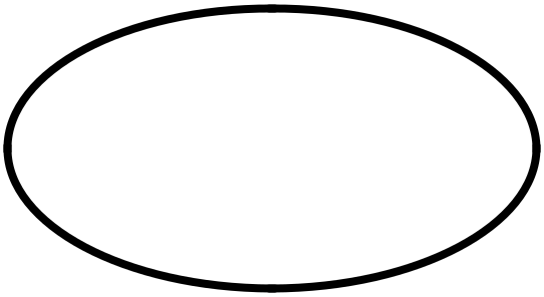
## 2.1 WORKFLOW DIAGRAM


## 2.2 DATA FLOW DIAGRAM

A representation of a system at different levels of details with graphic nexus of symbols representing data flows, data stores, procedures and data end points like source and destinations is termed as Data Flow Diagram.

## 2.3 DESIGN NOTATIONS

**Process -** A procedure or process does operations and give the output on the supplied arguments. The pure Functions are considered as low-level process that does not have side effects. A process data flow component is represented as an ellipse.



**Data Flows -**The nexus between one process to another or one sub identity to mother is represented by the with the intermediate value or the label on it

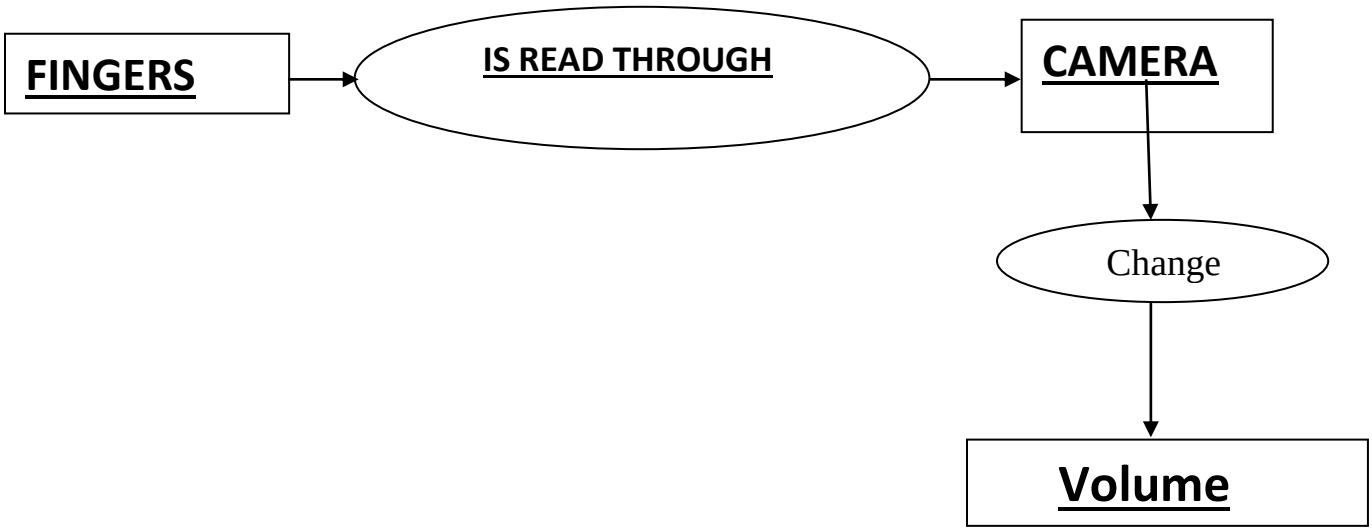
**Actors-**The element that drives the data flow by taking the inputs and thereby computing the out is termed as the actor.

**Data Store-** Sometimes data is required to be accessed later in the data flow that is done by data store component of DFD.

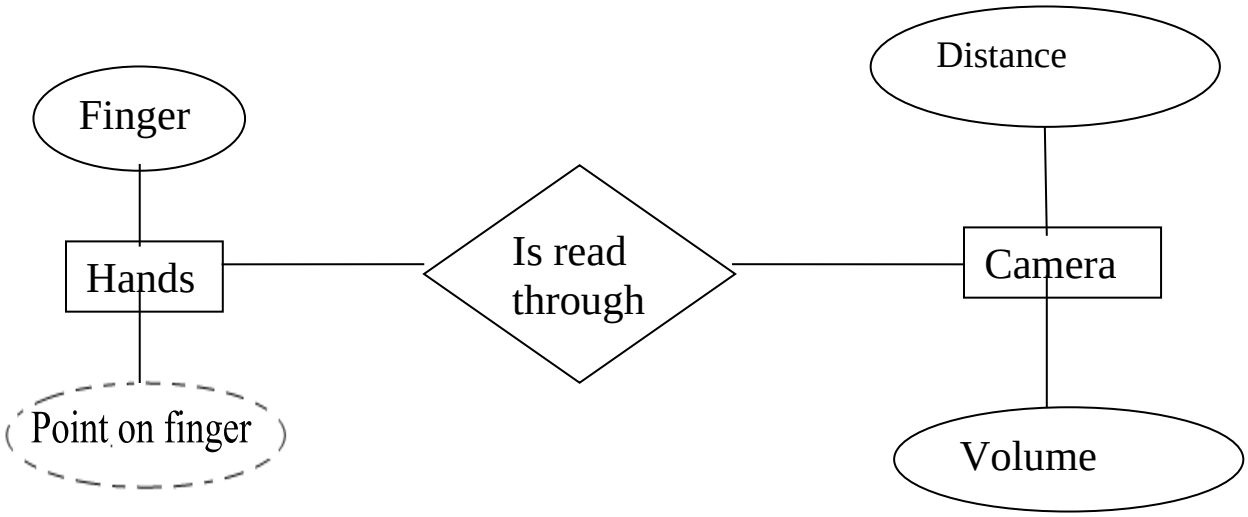
**External Entity-** Any external entity which can access the flow in DFD like a librarian, is called as External Entity component. It is represented as a rectangle.

## 2.3 DETAILED DESIGN

**ZERO LEVEL DFD**



## ER DIAGRAM


Distance
Finger
Is read through
Hands
Camera
Point on finger
Volume

# CHAPTER 3

# IMPLEMENTATION

## 3.1 PROJECT FILES

We have to have these files/libraries:-

1. Open CV2
2. Media pipe
3. Numpy
4. Math
5. pycaw
6. Python

## 3.2 VOLUME CONTROL USING WEBCAM

**CODE1:**

```
import cv2
import mediapipe as mp
import time


class handDetector():
    def _init_(self, mode=False, maxHands=2, detectionCon=0.5, trackCon=0.5):
        self.mode = mode
        self.maxHands = maxHands
        self.detectionCon = detectionCon
        self.trackCon = trackCon

        self.mpHands = mp.solutions.hands
        self.hands = self.mpHands.Hands(self.mode, self.maxHands,
                                        self.detectionCon, self.trackCon)
        self.mpDraw = mp.solutions.drawing_utils

    def findHands(self, img, draw=True):
```

```
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        self.results = self.hands.process(imgRGB)
        # print(results.multi_hand_landmarks)

        if self.results.multi_hand_landmarks:
            for handLms in self.results.multi_hand_landmarks:
                if draw:
                    self.mpDraw.draw_landmarks(img, handLms,
                                               self.mpHands.HAND_CONNECTIONS)
        return img

    def findPosition(self, img, handNo=0, draw=True):

        lmList = []
        if self.results.multi_hand_landmarks:
            myHand = self.results.multi_hand_landmarks[handNo]
            for id, lm in enumerate(myHand.landmark):
                # print(id, lm)
                h, w, c = img.shape
                cx, cy = int(lm.x * w), int(lm.y * h)
                # print(id, cx, cy)
                lmList.append([id, cx, cy])
                if draw:
                    cv2.circle(img, (cx, cy), 15, (255, 0, 255), cv2.FILLED)

        return lmList


def main():
    pTime = 0
    cTime = 0
    cap = cv2.VideoCapture(1)
    detector = handDetector()
    while True:
        success, img = cap.read()
```

```
        img = detector.findHands(img)
        lmList = detector.findPosition(img)
        if len(lmList) != 0:
            print(lmList[4])

        cTime = time.time()
        fps = 1 / (cTime - pTime)
        pTime = cTime

        cv2.putText(img, str(int(fps)), (10, 70), cv2.FONT_HERSHEY_PLAIN, 3,
                    (255, 0, 255), 3)

        cv2.imshow("Image", img)
        cv2.waitKey(1)


if _name_ == "_main_":
    main()
```

**CODE 2:**

```
import cv2
import time
import numpy as np
import HmodTrec as hmt
import math

from ctypes import cast, POINTER
from comtypes import CLSCTX_ALL
from pycaw.pycaw import AudioUtilities, IAudioEndpointVolume

width_cam, height_cam = 640, 480

cap = cv2.VideoCapture(0)
```

```
cap.set(width_cam, 1)
cap.set(height_cam, 2)
past_time = 0

detector = hmt.Hand_Detection(detectionCon=0.8)

device = AudioUtilities.GetSpeakers()
interface = device.Activate(
    IAudioEndpointVolume._iid_, CLSCTX_ALL, None)
volume = cast(interface, POINTER(IAudioEndpointVolume))
# volume.GetMute()
# volume.GetMasterVolumeLevel()
print(volume.GetVolumeRange())
volRange = volume.GetVolumeRange()
volume.SetMasterVolumeLevel(-0.5, None)
minVol = volRange[0]
maxVol = volRange[1]
vol = 0
volBar = 400

while True:
    success, img = cap.read()
    img = detector.search_hands(img)
    Hand_List = detector.find_position_of_hands(img, draw=False)
    if len(Hand_List) != 0:
        # print(lmList[4],lmList[8])

        x1, y1 = Hand_List[4][1], Hand_List[4][2]
        x2, y2 = Hand_List[8][1], Hand_List[8][2]
        cx, cy = (x1 + x2) // 2, (y1 + y2) // 2

        cv2.circle(img, (x1, y1), 10, (255, 0, 0), cv2.FILLED)
        cv2.circle(img, (x2, y2), 10, (255, 0, 0), cv2.FILLED)
        cv2.line(img, (x1, y1), (x2, y2), (255, 0, 0), 3)
        cv2.circle(img, (cx, cy), 10, (255, 0, 0), cv2.FILLED)
```

```
    length = math.hypot(x2 - x2, y1 - y2)
    # print(length)

    # Hand Range min-50, max-300
    # Volume Range min - 65, max - 0

    vol = np.interp(length, [50, 220], [minVol, maxVol])
    volBar = np.interp(length, [50, 300], [400, 150])
    print(vol)
    volume.SetMasterVolumeLevel(vol, None)

    if length < 50:
        cv2.circle(img, (cx, cy), 10, (0, 0, 0), cv2.FILLED)

cv2.rectangle(img, (50, 150), (85, 400), (0, 255, 0), 3)
cv2.rectangle(img, (50, int(volBar)), (85, 400), (0, 255, 0), cv2.FILLED)

current_time = time.time()
fps = 1 / (current_time - past_time)
past_time = current_time

cv2.putText(img, f'fps:{int(fps)}', (40, 50), cv2.FONT_HERSHEY_COMPLEX,
            1, (200, 34, 0), 3)

cv2.imshow("Img", img)
cv2.waitKey(1)
```

# CHAPTER 4

# TESTING

A set of activities carried out to check the functionality or stability is termed as testing. These activities are so planned and performed systematically that it leaves no scope for rework or bugs. General characteristics of these strategies are:

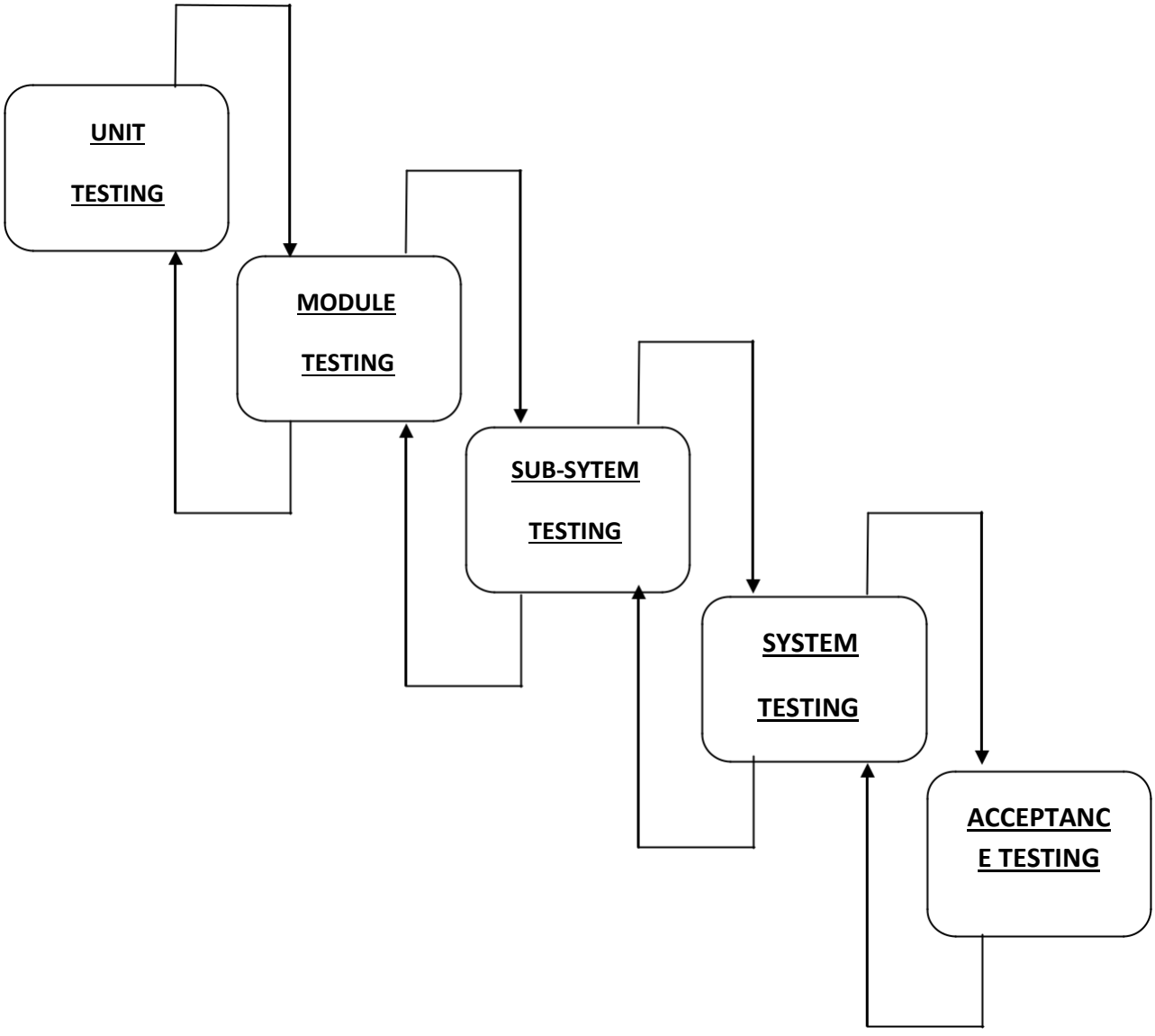
1 Testing begins at the module level and works outward".

2 disparate testing techniques are appropriate at disparate points in time.

3 Debugging and testing are altogether disparate procedures.

4 The developer of the software conducts testing and if the project is big then there is a testing team.

The System testing involved is the most widely used testing procedure consisting of five stages as shown in the figure. In general, the sequence of testing activities is component testing, integration testing, and then user testing.



## 4.1 Unit Testing

The unit test focused on the internal processing logic. All statements in a module have been exercised at least once. The interface module was tested to ensure that information properly flowed into and out of the program unit under test.

## 4.2 Functional Testing

Once the system is completed developed and integrated it is checked and evaluated for its functionality as whole for specific demands and requirements. This type of testing falls under the category of Blackbox testing and does not require the knowledge of in depth working and protocol off the system.

## 4.3 Structural Testing

In contrary to Functional testing Structural testing checks for the functionality of the different modules of the whole system and how well they are in link with another module. This type of testing requires full knowledge of the behavior, protocol, and the working of the system as a whole and module wise. The system base coding and programming knowledge is also a requirement to perform this testing. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs.

# CHAPTER 5

# MAINTAINANCE

Maintenance is the term that is used to refer to modifications that are made to software system after its release. System maintenance is an ongoing activity which covers a wide variety of activities including removing program and design errors, updating documentation and test data and updating user support Maintenance can be broadly classified into following three classes:

## 5.1 Corrective maintenance

This is used to remove errors in the program, which occurs when the product is delivered as well as during maintenance. Thus, in corrective maintenance the product is modified to solve the discovered errors after the software product is being delivered to customer.

## 5.2 Adaptive maintenance

Adaptive maintenance is generally not requested by client, but it is imposed by the outside environment. It may include following organizational changes:

- Change in the object
- Change in algorithms for faster performance
- Change in frames like instead of live detecting we need video frames
- Change in system controls and security needs etc.

## 5.3 Perfective maintenance

It means changing the software to improve some of its qualities like add new functions improve computer efficiency, make it easier to use. This type of maintenance is used to respond to users' additional needs may be due to the changes within or outside of the organization. These changes include:

- Changes in software
- Economic and competitive conditions
- Changes in models

System evaluation is the process of checking the performance of a complete system to acknowledge how it is likely to perform in live market conditions. It measures the performance of the system that whether it may compete or not.

# CHAPTER 6

# CONCLUSION

## 6.1 CONCLUSION

“It was a wonderful and learning experience for me while working on this project. I enjoyed every bit of work I had put into this project.”
“The project took me through various phases of different libraries and gave me a better understanding of python and its libraries. These libraries play an important part to complete this project. This project taught me how we can use multiple libraries and integrate them and make them work. This project makes me research to know the libraries we need and how we can use them.