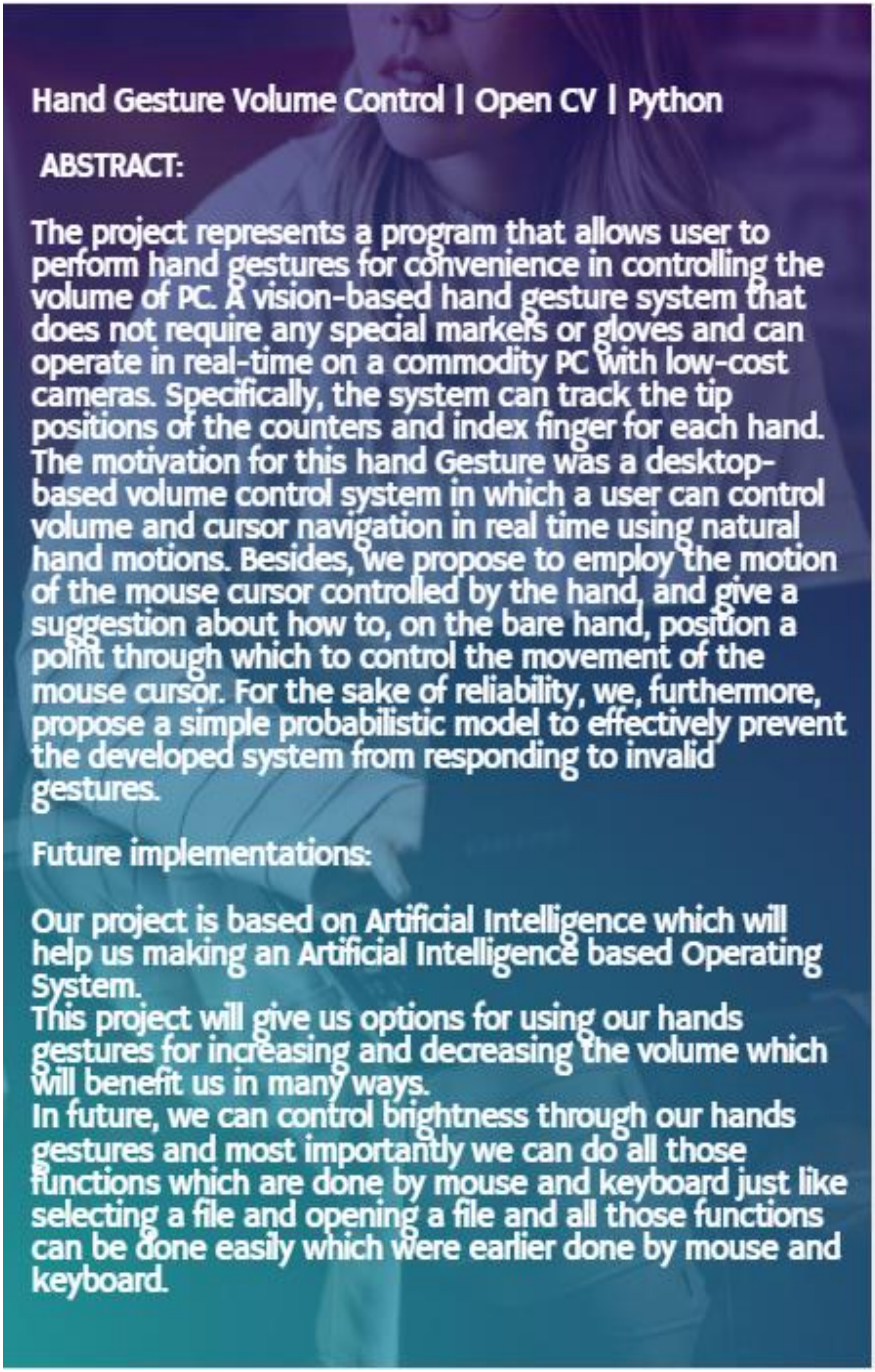
2022

Volume Controller using Python

ALIMA KHAN

PYTHON TRADE



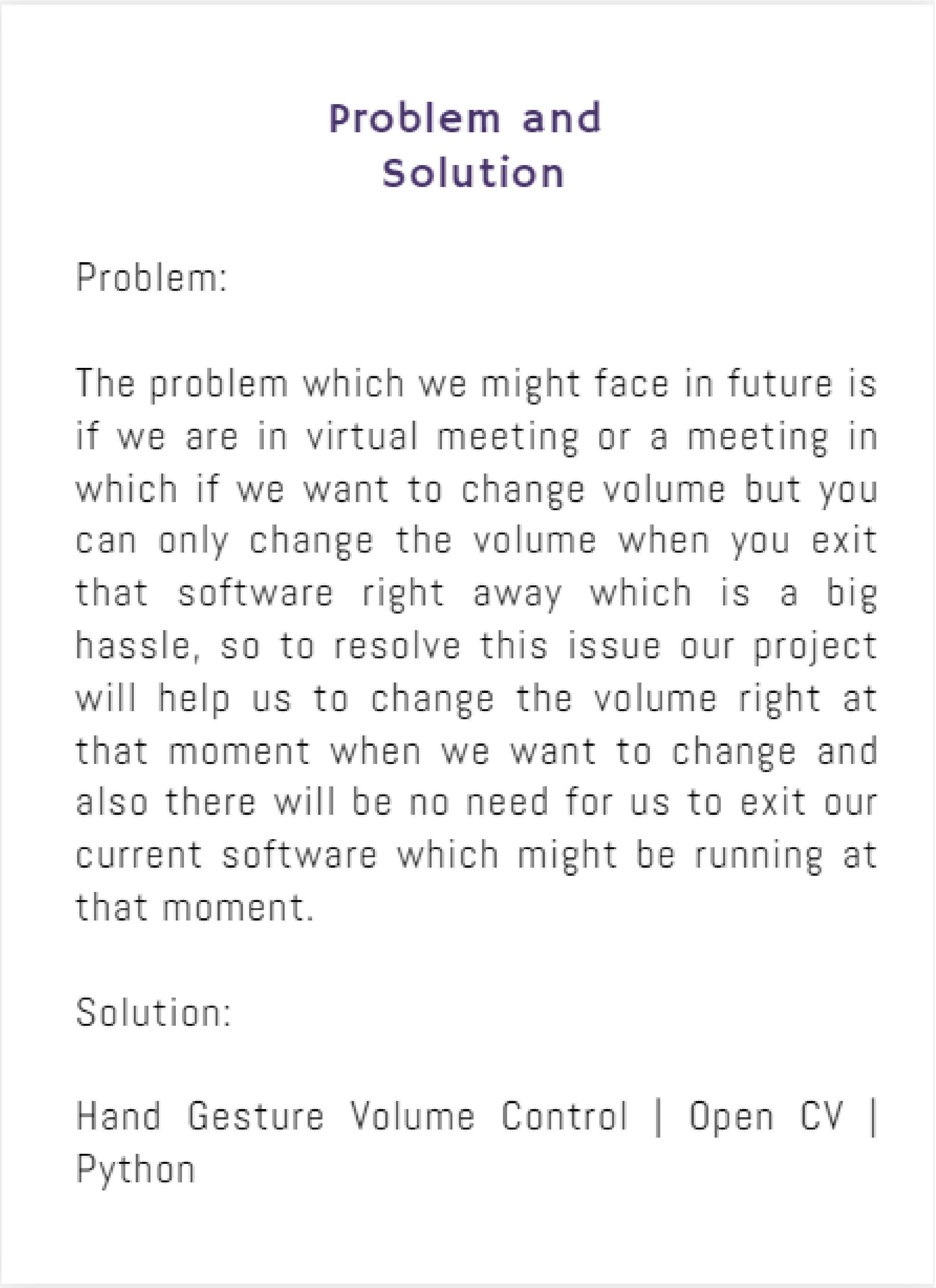
Hand Gesture Volume Control | Open CV | Python

ABSTRACT:

The project represents a program that allows user to perform hand gestures for convenience in controlling the volume of PC. A vision-based hand gesture system that does not require any special markers or gloves and can operate in real-time on a commodity PC with low-cost cameras. Specifically, the system can track the tip positions of the counters and index finger for each hand. The motivation for this hand Gesture was a desktop-based volume control system in which a user can control volume and cursor navigation in real time using natural hand motions. Besides, we propose to employ the motion of the mouse cursor controlled by the hand, and give a suggestion about how to, on the bare hand, position a point through which to control the movement of the mouse cursor. For the sake of reliability, we, furthermore, propose a simple probabilistic model to effectively prevent the developed system from responding to invalid gestures.

Future implementations:

Our project is based on Artificial Intelligence which will help us making an Artificial Intelligence based Operating System.
This project will give us options for using our hands gestures for increasing and decreasing the volume which will benefit us in many ways.
In future, we can control brightness through our hands gestures and most importantly we can do all those functions which are done by mouse and keyboard just like selecting a file and opening a file and all those functions can be done easily which were earlier done by mouse and keyboard.



Problem and Solution

Problem:

The problem which we might face in future is if we are in virtual meeting or a meeting in which if we want to change volume but you can only change the volume when you exit that software right away which is a big hassle, so to resolve this issue our project will help us to change the volume right at that moment when we want to change and also there will be no need for us to exit our current software which might be running at that moment.

Solution:

Hand Gesture Volume Control | Open CV | Python

INTRODUCTION:

The Volume Control With Hand Detection OpenCV Python was developed using Python OpenCV. In this Python OpenCV Project With Source Code. We build a Volume Controller with OpenCV to change the volume of our computer.
We first look into hand tracking and then we will use the hand landmarks to find gesture of our hand to change the volume.
This project is module-based which makes the hand tracking very easy.
Building a Volume Controller with OpenCV can be accomplished in just 3 simple steps:

- Step 1. Detect Hand landmarks
- Step 2. Calculate the distance between thumb tip and index finger tip.
- Step 3. Map the distance of thumb tip and index finger tip with volume range. For my case, distance between thumb tip and index finger tip was within the range of 15 – 220 and the volume range was from -63.5 – 0.0.

Tools and Technologies:

PPYCHARM:
PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains (formerly known as IntelliJ)
LANGUAGE:
Python:
Python is a popular programming language. Python can be used on a server to create web applications.
TECHNOLOGIES/MODULES:
CV2:- OpenCV has a function to read video, which is cv2. VideoCapture(). We can access our webcam using pass 0 in the function parameter.

Mediapipe:- MediaPipe offers open source cross-platform, customizable ML solutions for live and streaming media.
From math import hypot:- Python hypot() is an inbuilt function that calculates the hypotenuse of a right triangle.
From ctypes import cast, POINTER:- It provides C compatible data types.

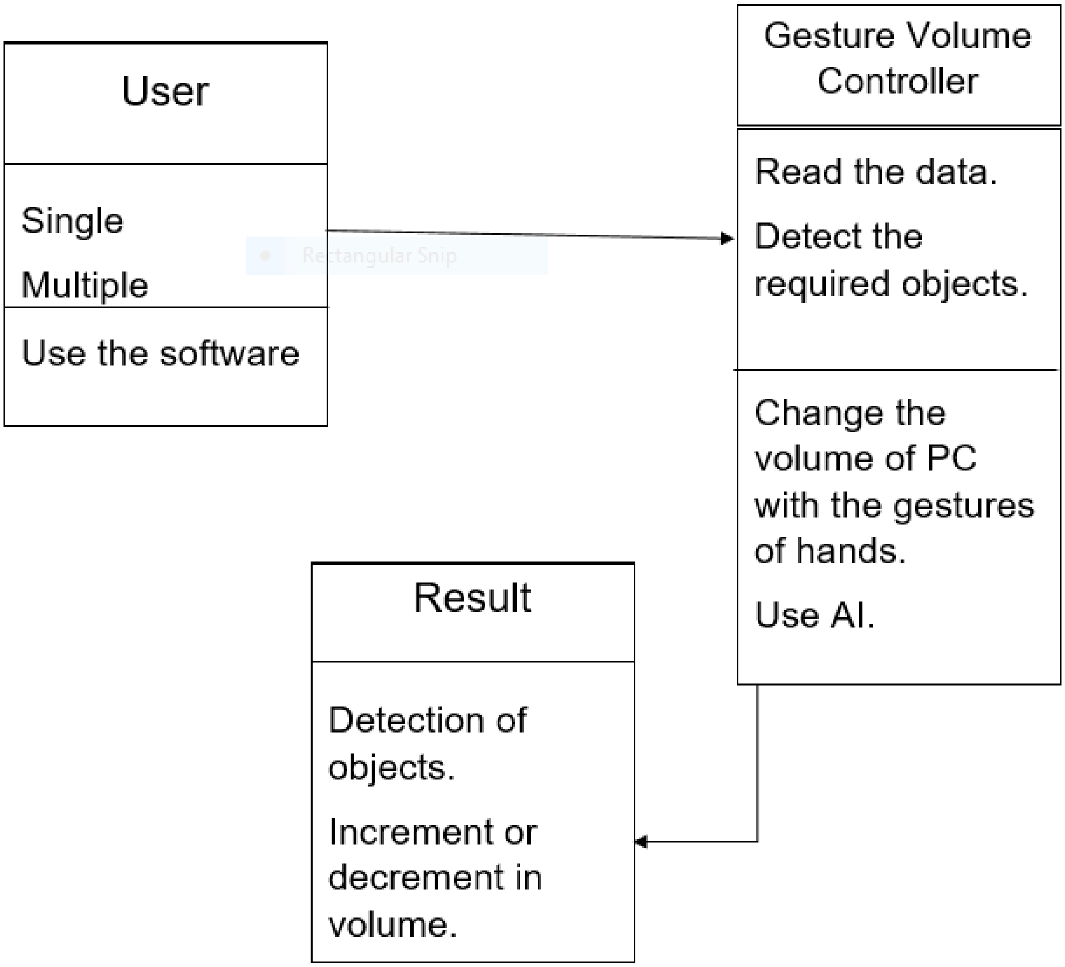
From comtypes import CLSCTX_ALL:- Automation objects are objects that expose a programmable interface that can be used by another program or environment.

From pycaw.pycaw import AudioUtilities, IAudioEndpointVolume:- It is used to change the volume of our PC.
import numpy as np:- Numpy is a module used to interact with arrays in Python.

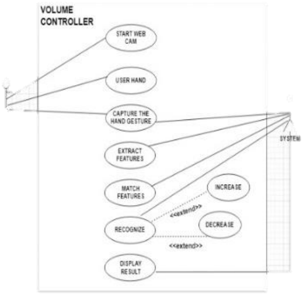
Code

```
import cv2
import mediapipe as mp
from math import hypot
from ctypes import cast, POINTER
from comtypes import CLSCTX_ALL
from pycaw.pycaw import AudioUtilities, IAudioEndpointVolume
import numpy as np

cap = cv2.VideoCapture(0)

mpHands = mp.solutions.hands
hands = mpHands.Hands()
mpDraw = mp.solutions.drawing_utils

devices = AudioUtilities.GetSpeakers()
interface = devices.Activate(IAudioEndpointVolume._iid_, CLSCTX_ALL, None)
volume = cast(interface, POINTER(IAudioEndpointVolume))
volMin, volMax = volume.GetVolumeRange()[:2]

while True:
    success, img = cap.read()
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    results = hands.process(imgRGB)

    lmList = []
    if results.multi_hand_landmarks:
        for handlandmark in results.multi_hand_landmarks:
            for id, lm in enumerate(handlandmark.landmark):
                h, w, _ = img.shape
                cx, cy = int(lm.x * w), int(lm.y * h)
                lmList.append([id, cx, cy])
            mpDraw.draw_landmarks(img, handlandmark, mpHands.HAND_CONNECTIONS)

    if lmList != []:
        x1, y1 = lmList[4][1], lmList[4][2]
        x2, y2 = lmList[8][1], lmList[8][2]
        cv2.circle(img, (x1, y1), 4, (255, 0, 0), cv2.FILLED)
        cv2.circle(img, (x2, y2), 4, (255, 0, 0), cv2.FILLED)
        cv2.line(img, (x1, y1), (x2, y2), (255, 0, 0), 3)
        length = hypot(x2 - x1, y2 - y1)
        vol = np.interp(length, [15, 220], [volMin, volMax])
        print(vol, length)
        volume.SetMasterVolumeLevel(vol, None)

    cv2.imshow('Image', img)
    if cv2.waitKey(1) & 0xff == ord('q'):
        break
```

Class Diagram:-


User
Single
Multiple
Use the software
Rectangular Snip
Gesture Volume Controller
Read the data.
Detect the required objects.
Change the volume of PC with the gestures of hands.
Use AI.
Result
Detection of objects.
Increment or decrement in volume.

Use-case Diagram:-


VOLUME CONTROLLER
START WEB CAM
USER HAND
CAPTURE THE HAND GESTURE
EXTRACT FEATURES
MATCH FEATURES
RECOGNIZE
INCREASE
DECREASE
<<extend>>
<<extend>>
DISPLAY RESULT
SYSTEM