**Machine Learning with Python**

**Assignment**

Wireless Sound Control

**OMM PRAKASH SAHOO**

This project is about how to build a hand gesture-controlled volume system using Python. In this tutorial, we use the mediapipe and opencv-python libraries to detect the landmarks and gestures of our hand. We then use pycaw to control the volume of our device. We will use a combination of **the thumb** and **index finger** to increase or decrease the volume.

In this tutorial, we will install the following libraries:

1. numpy: to work with arrays.
2. opencv-python: to access camera of device and capture image for further use.
3. mediapipe: to detect hand gestures.
4. pycaw: it gives access to the device's speaker and its main volume.
5. python-math: to find the distance between the thumb and the index finger using hypotenuse.
6. gpib-ctypes, comtypes: we need these because “pycaw” depends on these two libraries.

We then proceed to code by following these steps:

1. Importing the necessary libraries.
2. Detecting, initializing, and configuring the hand gestures.
3. Accessing the speaker using pycaw
4. Finding the volume range of device
5. Capturing an image from our camera
6. Checking if there are multiple hands in our input
7. Creating a for loop to work on each hand’s input
8. Detecting the locus of the thumb and index finger that we will use to manipulate device’s audio
9. Finding the distance between those 2 points using “math.hypotenuse”
10. Converting the distance into volume range in which the device operates
11. Set the volume of the device using pycaw using the volume range

In conclusion, with these simple steps, we can build a hand gesture-controlled volume system using Python. And it can be used in various large scale projects to make then easier.

Each line python code is explained by me (commented out):

# assignment

# OMM PRAKASH SAHOO

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) #Checks for camera

mpHands = mp.solutions.hands #detects hand/finger using mediapipe library

hands = mpHands.Hands() #complete the initialization configuration of hands

#To access speaker through the library pycaw

devices = AudioUtilities.GetSpeakers()

interface = devices.Activate(IAudioEndpointVolume.\_iid\_, CLSCTX\_ALL, None)

volume = cast(interface, POINTER(IAudioEndpointVolume))

volMin,volMax = volume.GetVolumeRange()[:2]

while True: #infinite loop

success,img = cap.read() #If camera works capture an image

Captured\_image = cv2.cvtColor(img,cv2.COLOR\_BGR2RGB) #Convert the captured image to rgb

#Collection of gesture information

results = hands.process(Captured\_image) #completes the image processing.

lmList = [] #it will contain all co-ordinates of all the 21 points on hand

if results.multi\_hand\_landmarks: #if list of all hands detected is not empty --> proceed

#By accessing the list, we can get the information of each hand's corresponding flag bit

for handlandmark in results.multi\_hand\_landmarks:

for id,lm in enumerate(handlandmark.landmark): #adding counter and returning thee x,y and z cordinates of points on hand

h,w,\_ = img.shape #returns the shape of capturedimage

cx,cy = int(lm.x\*w),int(lm.y\*h)

lmList.append([id,cx,cy]) #adding the 21 point co-ordinates in corresponding order to the empty list 'lmList'

if lmList != []: #if lmList is not empty --> proceed

x1,y1 = lmList[4][1],lmList[4][2] #the x and y co-ordinates thumb (point no.4)

x2,y2 = lmList[8][1],lmList[8][2] #the x and y co-ordinates of index finger (point no.8)

#creating circle at the tips of thumb and index finger

cv2.circle(img,(x1,y1),15,(0,255,0),cv2.FILLED)

cv2.circle(img,(x2,y2),15,(0,255,0),cv2.FILLED)

#creating a line b/w tips of index finger and thumb

cv2.line(img,(x1,y1),(x2,y2),(255,0,0),3)

length = hypot(x2-x1,y2-y1) #distance b/w tips using hypotenuse

# from numpy we find our length,by converting hand range in terms of volume range ie b/w -63.5 to 0

vol = np.interp(length,[30,350],[volMin,volMax])

#print(vol,int(length)) #this line is optional to use

volume.SetMasterVolumeLevel(vol, None)#---------------------------------> the final step "Setting the volume"

cv2.imshow('Assignment By OMM PRAKASH SAHOO',img) #Show the image on a separate window named Assignment By OMM PRAKASH SAHOO

if cv2.waitKey(1) & 0xff==ord('q'): #press q to break the loop

break

# after loop breaks

cap.release() #stop cam

cv2.destroyAllWindows() #close window