



# WIRELESS SOUND CONTROL

Prepared and Presented by: Anaum Khan

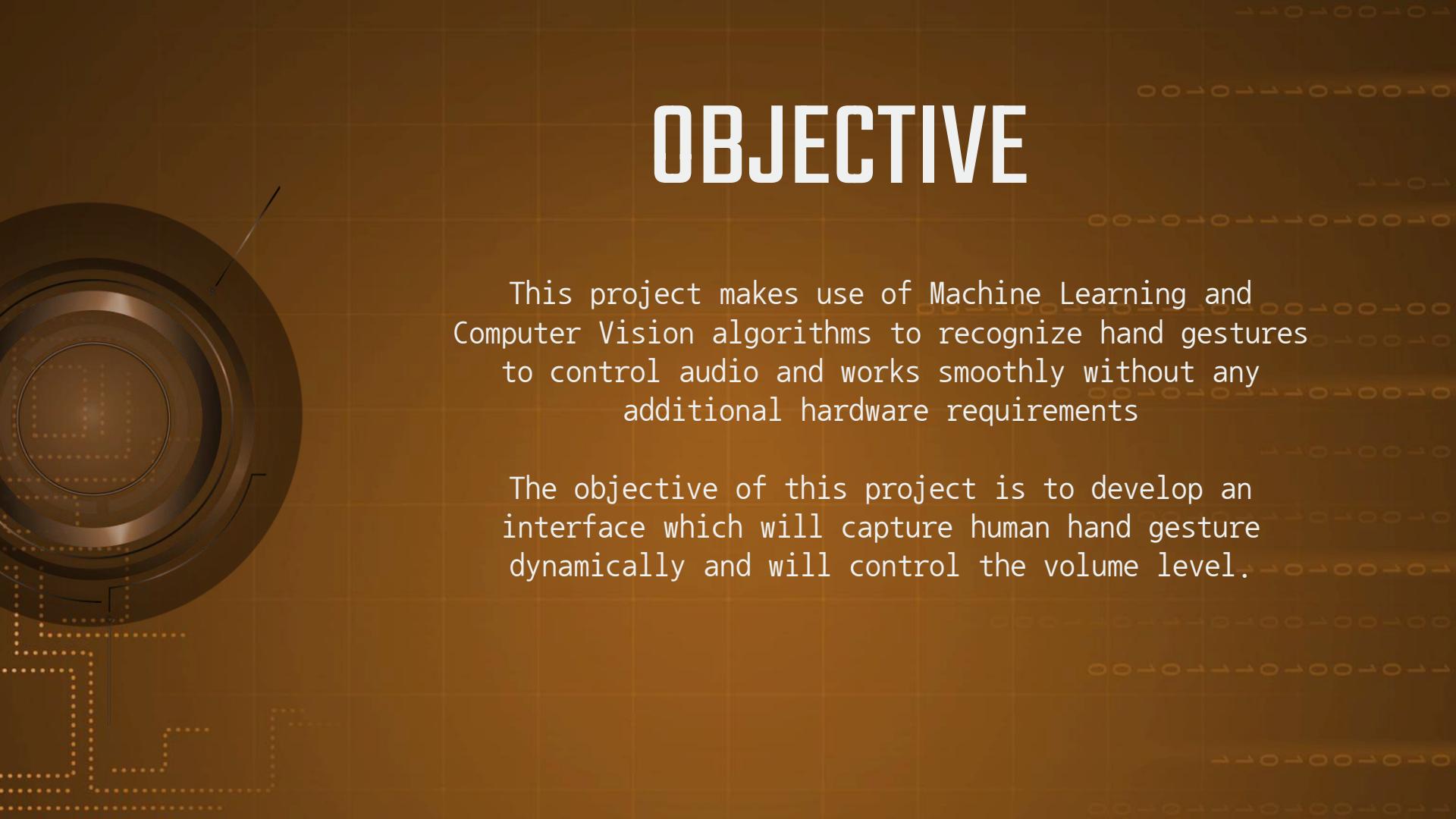
2<sup>nd</sup> year  
Computer Engineering Department

# INTRODUCTION

“ Interaction design isn’t about how interfaces behave, it’s about how people behave, and then adapting technology accordingly. ”

- In Human Computer Interaction (HCI), the finest invention is the mouse. Even though a wireless mouse or the Bluetooth mouse is in demand today, it still lacks in many fields like cost and power.
- Hand gesture mouse control for systems has received a lot of attention in recent years.





# OBJECTIVE

This project makes use of Machine Learning and Computer Vision algorithms to recognize hand gestures to control audio and works smoothly without any additional hardware requirements

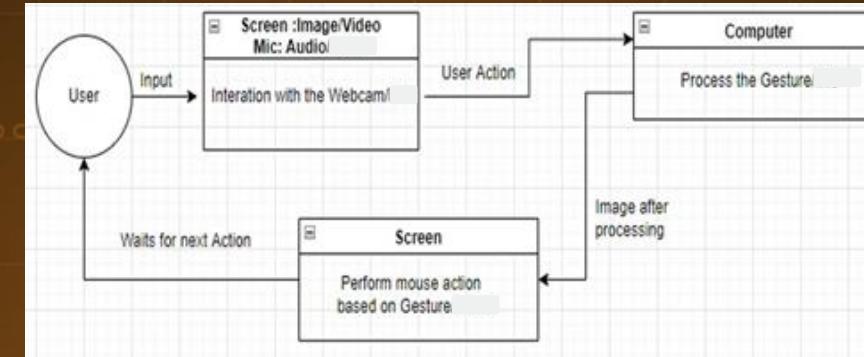
The objective of this project is to develop an interface which will capture human hand gesture dynamically and will control the volume level.

# DATA FLOW DIAGRAM

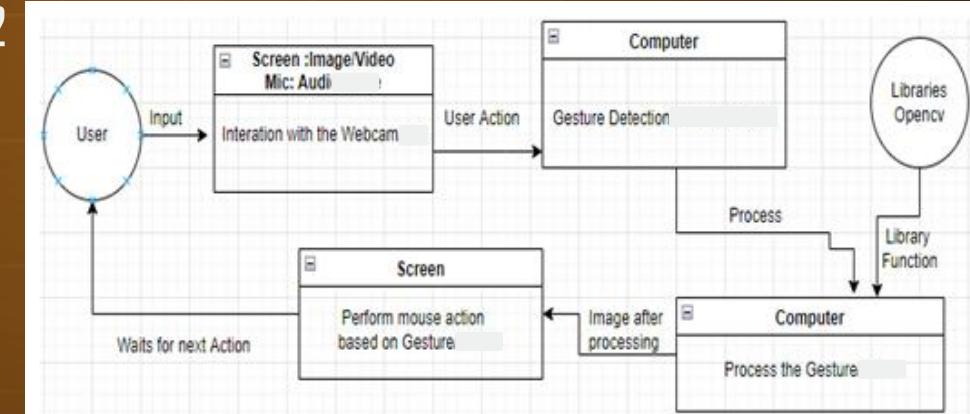
DFD Level 0



DFD Level 1



DFD Level 2



The most important century in human history

Save to Liner

TED-Ed Subscribed 14K Share

Share summary

20°C Mostly cloudy

OPEN EDITORS 1 unsaved

check5.py

check3.py

mushroom.jpg

check6.py

mp\_try1.py

check4.py

numpytry.py

OPENCV02

check3.py

check4.py

check5.py

check6.py

kidroom.png

mp.try1.py

mushroom.jpg

mushroom.png

numpytry.py

```
def WirelessSoundControl():
    cap = cv2.VideoCapture(0)
    mpHands = mp.solutions.hands
    hands = mpHands.Hands()
    mpDraw = mp.solutions.drawing_utils
    devices = Audiounitilities.GetSpeakers()
    interface = devices.Activate(
        IAudioEndpointVolume._iid_, CLSCTX_ALL, None)
    volume = cast(interface, POINTER(IAudioEndpointVolume))

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

        if results.multi_hand_landmarks:
            for handLms in results.multi_hand_landmarks:
                lmList = [
```

PROBLEMS OUTPUT TERMINAL ...

-10.131670194948612 316  
-5.5691194049407 331  
-7.477387237625763 325  
-7.743997614543185 324  
-6.716685037591461 327  
-5.564141276901599 331  
-31.48347124625647 245  
-69.05754599363831 119

PS C:\Users\Anam\OpenCV02> python

LN 19, Col 16 Spaces: 4 UTF-8 CRLF ⚡ Python 3.7.9 64-bit

ENG IN 03-03-2024

# APPLICATIONS

01

In case of an outbreak of communicable disease, it may not safe to use the devices by touching them to avoid situation of spread of the pathogen.

02

The system can be used to control robots and automation systems without the usage of devices.

03

The system provides better interface especially for visually challenged.

# LIMITATIONS



The approach had obvious detection difficulty when the light levels were changed or a complex background was used and required a fixed distance from the camera to the users.

It further requires skin detection techniques and image processing techniques like Background Subtraction and Image Smoothening to be incorporated.

Some other hinderance encountered include:

- Hand detection fails as distance from camera increases
- Dependent upon the quality of the camera

# FUTURE SCOPE

1. We can work to create more gestures thus increasing the functionality of the virtual mouse.
2. Furthermore, the proposed method can be developed alongside voice assistant which is another future scope of Human-Computer Interaction (HCI).



# REFERENCES



[Control Mouse using Hand Gesture and Voice](#)  
Authors: Dr. Jayant Nandwalkar, Mahima Mandal , Amisha Khirari, Tejas Bhalchim



[Hand Gesture and Voice Assistants](#)  
Authors: B. Latha, Sri Sowndarya, Swethamalya, Ashish Raghuvanshi

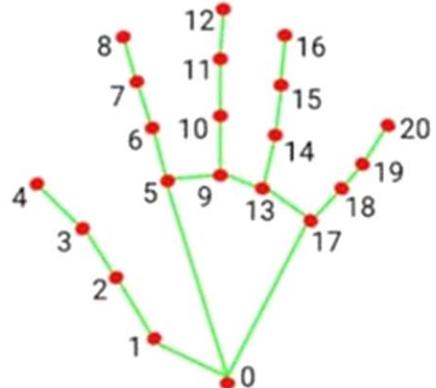


[Hand Gesture Controller \(Virtual Mouse\) and Voice Assistant using OpenCV, ML, Python](#)  
Authors: Dr. Pratibha V. Waje1, Ms. Shipranjali K. Gangurde, Ms. Snehal S. Sonawane, Ms. Pallavi S. Avhad, Mr. Shubham S. Raut

“ Great things are not done  
by impulse, but by a series  
of small things brought  
together. ”

— VINCENT VAN GOGH

# Hand Tracking



- 0. WRIST
- 1. THUMB\_CMC
- 2. THUMB\_MCP
- 3. THUMB\_IP
- 4. THUMB\_TIP
- 5. INDEX\_FINGER\_MCP
- 6. INDEX\_FINGER\_PIP
- 7. INDEX\_FINGER\_DIP
- 8. INDEX\_FINGER\_TIP
- 9. MIDDLE\_FINGER\_MCP
- 10. MIDDLE\_FINGER\_PIP
- 11. MIDDLE\_FINGER\_DIP
- 12. MIDDLE\_FINGER\_TIP
- 13. RING\_FINGER\_MCP
- 14. RING\_FINGER\_PIP
- 15. RING\_FINGER\_DIP
- 16. RING\_FINGER\_TIP
- 17. PINKY\_MCP
- 18. PINKY\_PIP
- 19. PINKY\_DIP
- 20. PINKY\_TIP

Using Hand Landmark Model available in MediaPipe



# IMPLEMENTATIONS

## NumPy

adds support for large, multi-dimensional arrays and matrices, and high-level mathematical functions to operate on these arrays.



## Pycaw

Python Core Audio Windows Library, working for both Python2 and Python3



## OpenCV

Open-Source Computer Vision is a library in which is written in C++ used for Computer Vision

## MediaPipe

A cross-platform open source framework used for building multimodal pipelines in made available by google