



# 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.

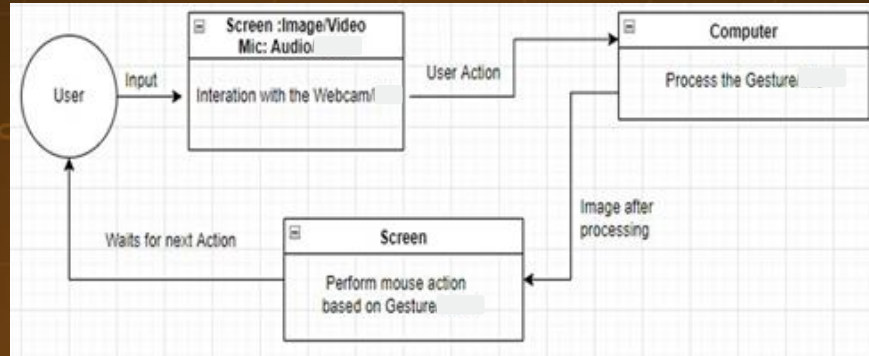


# DATA FLOW DIAGRAM

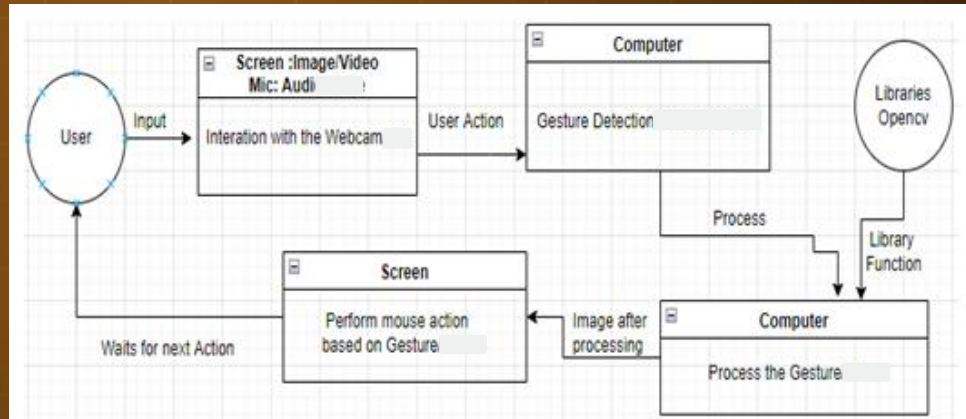
## DFD Level 0



## DFD Level 1




## DFD Level 2



YouTube

Ted ed on ai



and since then we've only increased the odds against us.

**The most important century in human history**

Save to Liner

TED-Ed 19.6M subscribers

Subscribed

14K

Share

Share summary

20°C Mostly cloudy

EXPLORER

OPEN EDITORS 1 unsaved

- Welcome
- check.py C:\User...
- check3.py
- mushroom.jpg
- check5.py
- check6.py
- mp\_try1.py
- check4.py
- numpytry.py

OPENCV2

- check3.py
- check4.py
- check5.py
- check6.py
- kidroom.png
- mp\_try1.py
- mushroom.jpg
- mushroom.png
- numpytry.py

```
check5.py > WirelessSoundControl
1 import cv2
2 import mediapipe as mp
3 import math
4 from ctypes import cast, POINTER
5 from ctypes import CLSCTX_ALL
6 from pycaw.pycaw import AudioUtilities, IAudioEndpoint
7 import numpy as np
8
9 def WirelessSoundControl():
10     cap = cv2.VideoCapture(0)
11     mpHands = mp.solutions.hands
12     hands = mpHands.Hands()
13     mpDraw = mp.solutions.drawing_utils
14     devices = AudioUtilities.GetSpeakers()
15     interface = devices.Activate(
16         IAudioEndpointVolume._iid, CLSCTX_ALL, None)
17     volume = cast(interface, POINTER(IAudioEndpointV
18
19     while True:
20         success, img = cap.read()
21         imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
22         results = hands.process(imgRGB)
23         #print(results.multi_hand_landmarks)
24
25         if results.multi_hand_landmarks:
26             for handLms in results.multi_hand_landmark
27                 lmList = []
```

PROBLEMS OUTPUT TERMINAL

python

```
-10.1316781948948612 316
-5.56911948948612 331
-7.477387237625763 325
-7.743997614543185 324
-6.716685037591461 327
-5.564141276901509 331
-31.48347124625647 245
-69.05754599363831 119
PS C:\Users\Anam\OpenCV2> python
```

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

# 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.

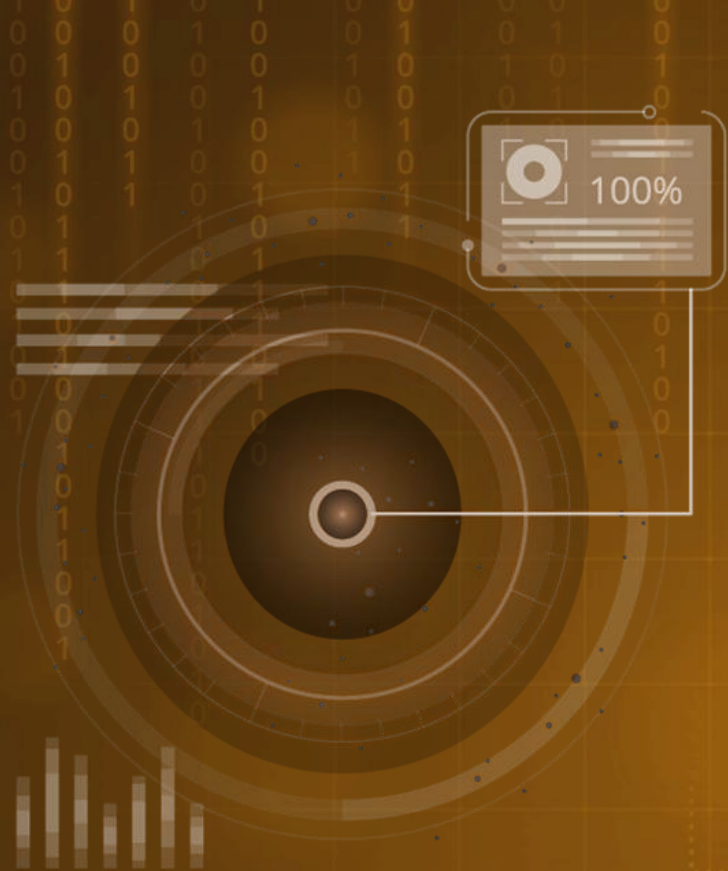
# LIMITATIONS

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 Smoothing 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



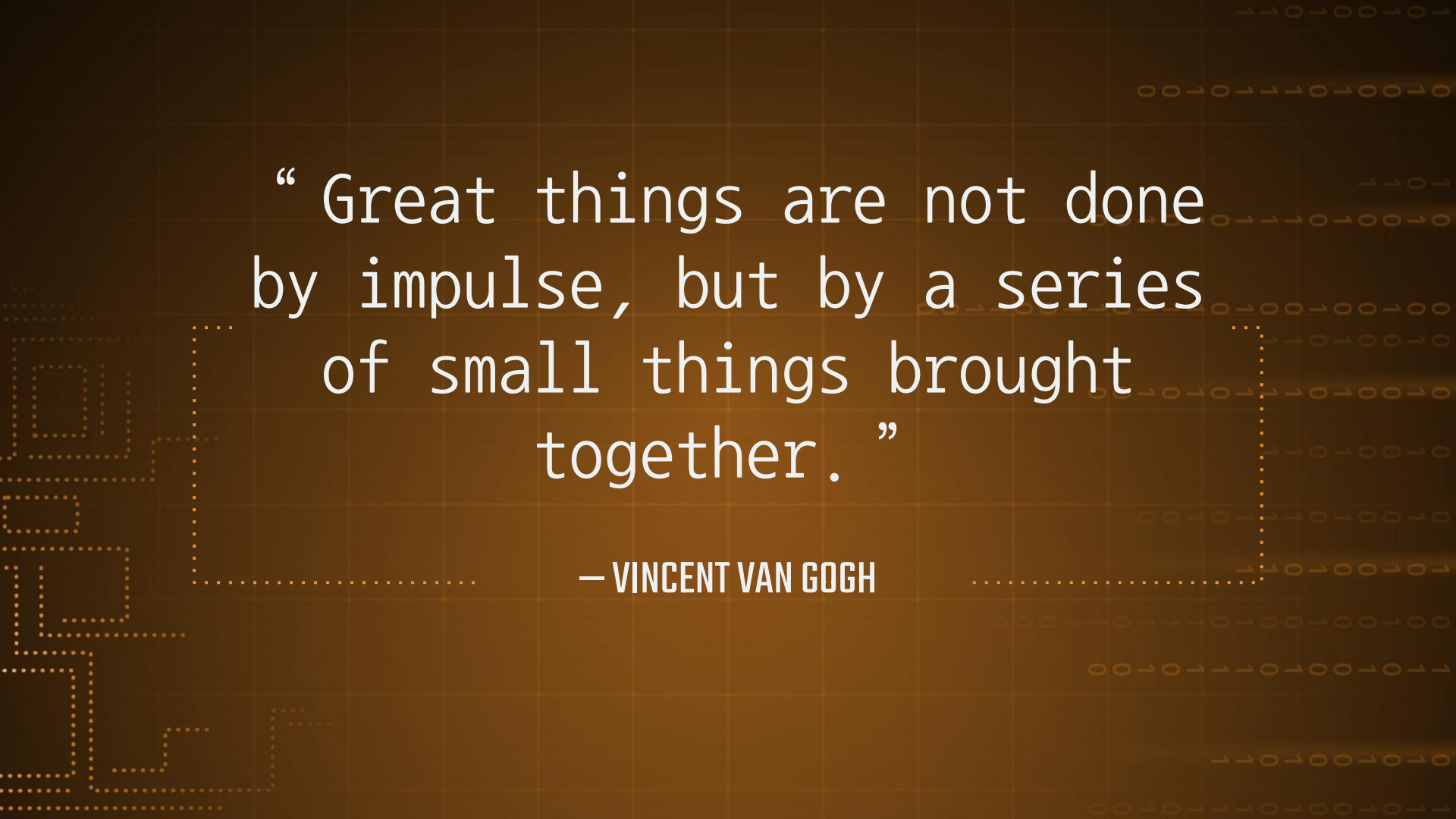
## Hand Gesture and Voice Assistants

Authors: B. Latha, Sri  
Sowndarya, Swethamalya,  
Ashish Raghuwanshi



## Hand Gesture Controller (Virtual Mouse) and Voice Assistant using OpenCV, ML, Python

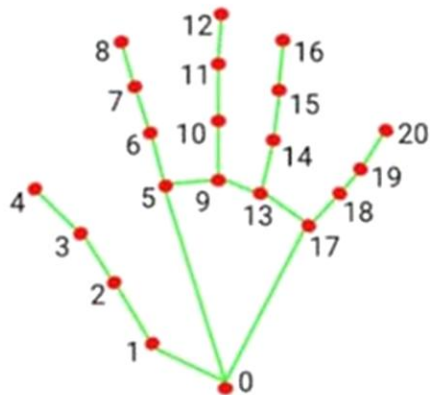
Authors: Dr. Pratibha V.  
Waje1, Ms.Shipranjali K.  
Gangurde, Ms. Snehal S.  
Sonawane, Ms. Pallavi S.  
Avhad, Mr.Shubham S. Raut

The background is a solid dark brown color with a subtle grid pattern. Overlaid on this are several dotted lines in a lighter brown shade. On the left side, there are vertical and horizontal dotted lines forming a stepped, staircase-like pattern. On the right side, there are horizontal dotted lines, some of which are interrupted by small circles, resembling a digital or binary sequence.

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

— VINCENT VAN GOGH

# Hand Tracking



- |                       |                       |
|-----------------------|-----------------------|
| 0. WRIST              | 11. MIDDLE_FINGER_DIP |
| 1. THUMB_CMC          | 12. MIDDLE_FINGER_TIP |
| 2. THUMB_MCP          | 13. RING_FINGER_MCP   |
| 3. THUMB_IP           | 14. RING_FINGER_PIP   |
| 4. THUMB_TIP          | 15. RING_FINGER_DIP   |
| 5. INDEX_FINGER_MCP   | 16. RING_FINGER_TIP   |
| 6. INDEX_FINGER_PIP   | 17. PINKY_MCP         |
| 7. INDEX_FINGER_DIP   | 18. PINKY_PIP         |
| 8. INDEX_FINGER_TIP   | 19. PINKY_DIP         |
| 9. MIDDLE_FINGER_MCP  | 20. PINKY_TIP         |
| 10. MIDDLE_FINGER_PIP |                       |

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.



## OpenCV

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

## PyCaw

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



## MediaPipe

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