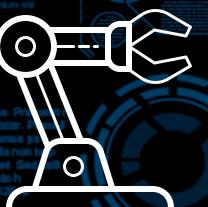


Artificial Intelligence

With Visual Studio Code & NodeRed

Present By ESL



Topic

which we could learning today?

1

OpenCV

2

Mediapipe

- - - - >

Mediapipe Holistic
(guide line)

" Wisdom comes with age, so just live and learn."

By William Landry

Open-cv

OpenCV
(Open Source Computer Vision Library)

- is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.



<https://opencv.org/get-started/>

Face detection

Q1

Install Python

open your command prompt

Q2

```
pip3 install opencv-python
```

```
pip3 install numpy
```



We'll use Visual Studio Code

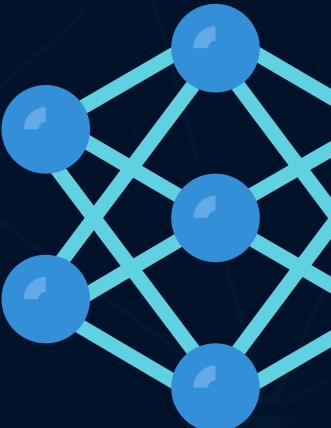


- Create new file your project (.py file)

Q3

Coding

Model (face , eye , etc.)



<https://github.com/opencv/opencv/tree/master/data>

```
1 import cv2
2 import numpy as np
3 cascade = cv2.CascadeClassifier('data/haarcascades/haarcascade_frontalface_default.xml')
4
5 cap = cv2.VideoCapture(0) → Initialize the webcam
6 while True:
7     ret, frame = cap.read() → Read a frame from the webcam
8     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) → Convert the frame to grayscale
9     objects = cascade.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5, minSize=(30, 30))
10    for (x, y, w, h) in objects:
11        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
12
13    cv2.imshow('Object Detection', frame) → Perform object detection
14
15    if cv2.waitKey(1) & 0xFF == ord('q'): → When press key 'q' it will exist
16        break
17
18    cap.release()
19    cv2.destroyAllWindows()
```

Try on your self

to run program

```
PS C:\Users\bluep\Desktop\AI Learning\Object Detecting_test> python face_detection_opencv.py
```

directory path file already in python filename.py

| TITLE | LAST MODIFIED |
|-------------------------------|---------------|
| ch1 colab test.ipynb | May 29 |
| ch1_hello.ipynb | May 29 |
| data | May 29 |
| Object_Detection_yolov5.ipynb | 1:32 am |

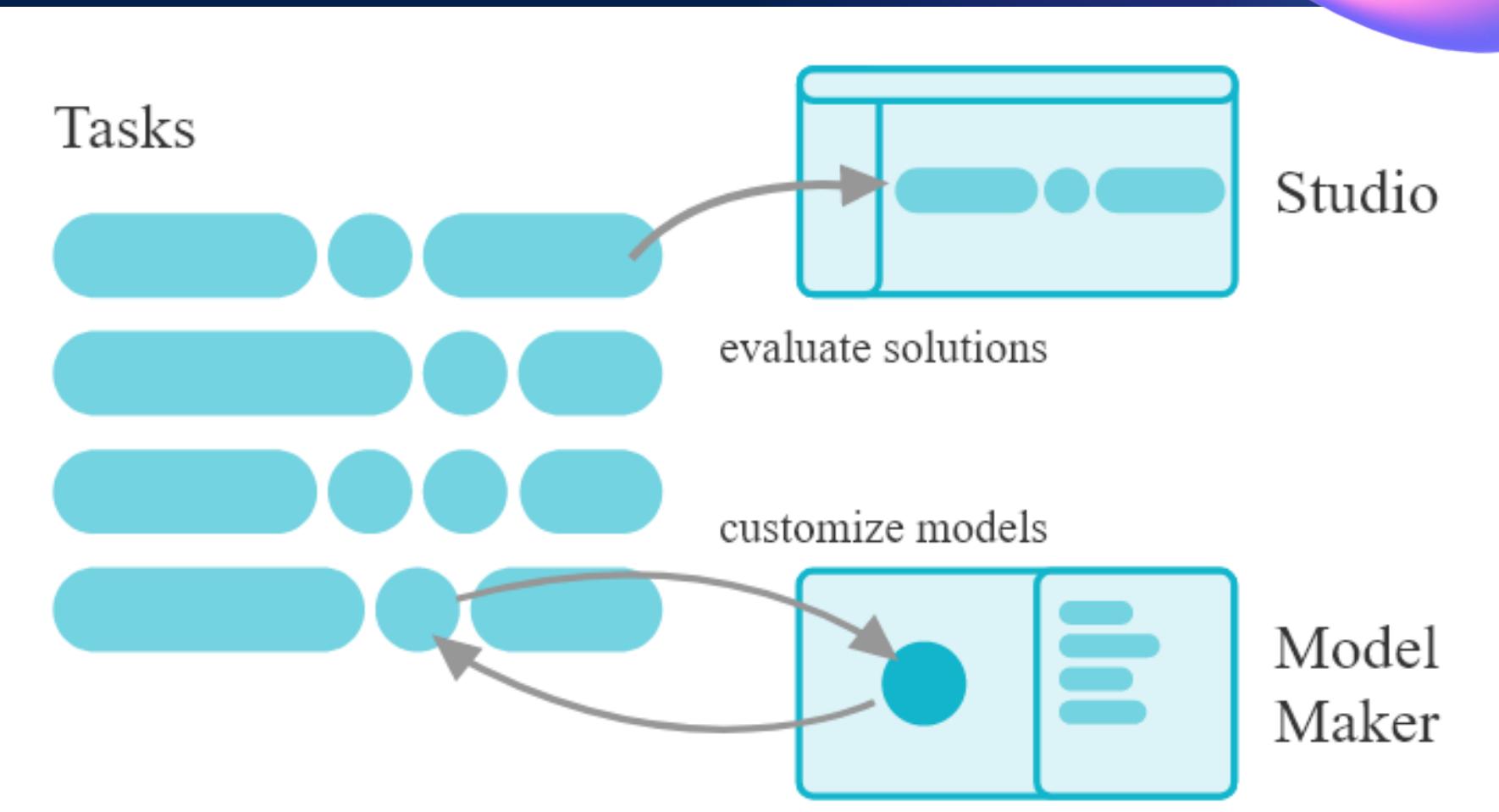
can try on others object example detection on this link : <https://shorturl.at/ejszl>

**DO YOU WANT CONTINUE
DETECTION ANALYZE TYPE >/?<**

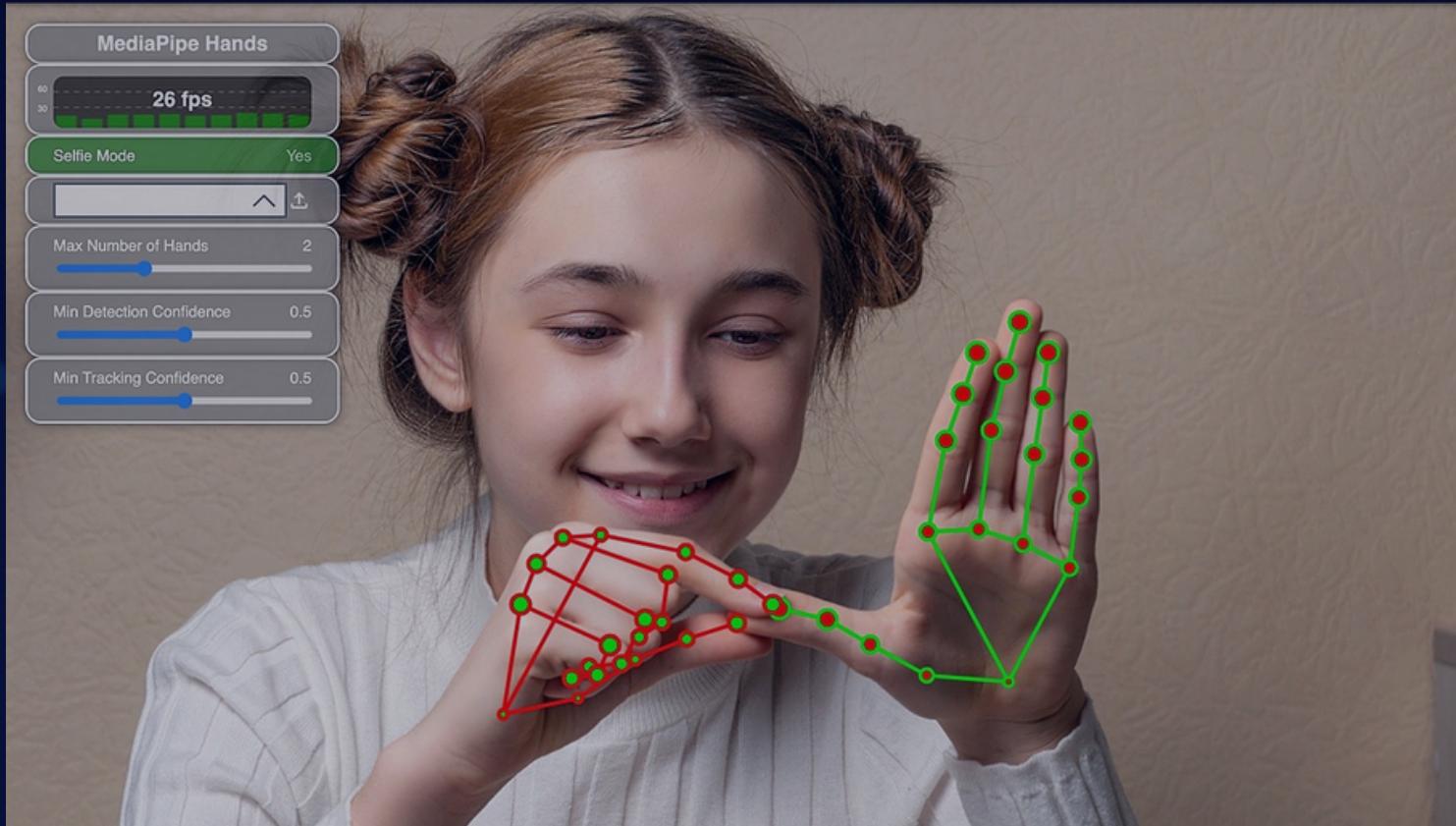
NEXT PAGE >>

Mediapipe is?

- MediaPipe Solutions provides a suite of libraries and tools for you to quickly apply artificial intelligence (AI) and machine learning (ML) techniques in your applications.
- You can plug these solutions into your applications immediately, customize them to your needs, and use them across multiple development platforms. MediaPipe Solutions is part of the MediaPipe open source project, so you can further customize the solutions code to meet your application needs.

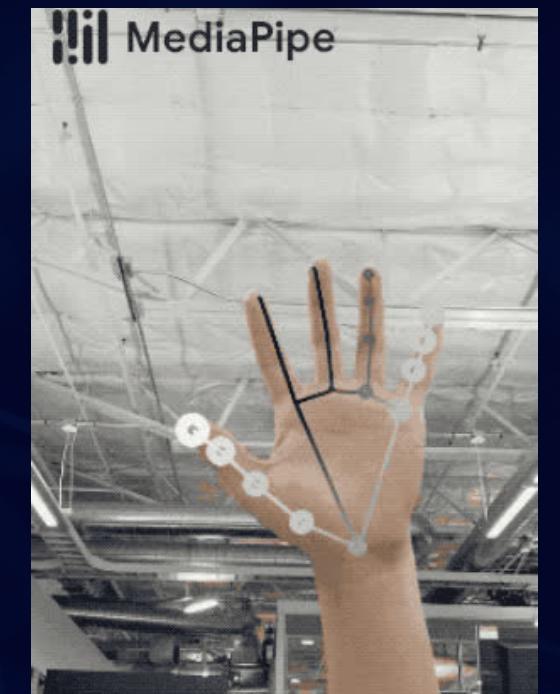


MediaPipe Hands

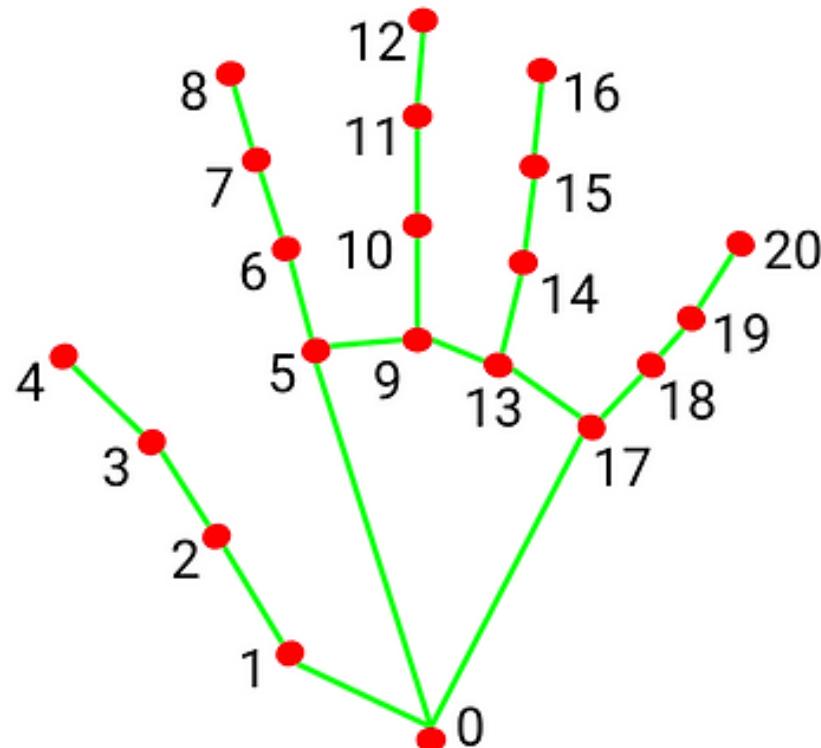


Today we will learn to use MediaPipe, which is a Machine Learning Solutions or a ready-made program from Google that can be used to do Hand Tracking accurately. And as fast as real-time tracking.

Because MediaPipe has a number of solutions available, ranging from gesture detection to facial recognition, today we're going to use MediaPipe Hands that detect hands.



21 point finger Landmarks



- | | |
|-----------------------|-----------------------|
| 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 | |

Hand Tracking starts by sensing the palm with the "Palm Detection Model" and then identifies 21 key hand locations.

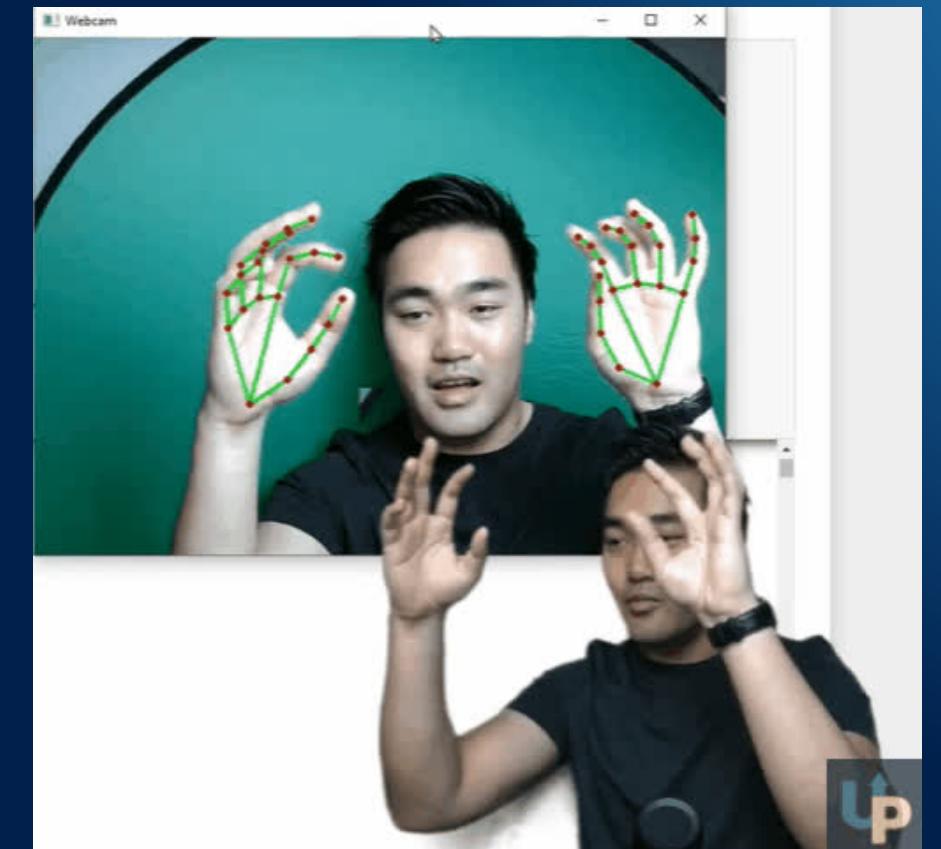
Hand Tracking

open your command prompt

Q1

```
pip3 install mediapipe
```

```
pip3 install opencv-python
```



Q3

Hand Track Coding

Our Vision

```
1 import cv2
2 import mediapipe as mp
3
4 mp_hands = mp.solutions.hands
5 mp_draw = mp.solutions.drawing_utils
6
7 webcam = cv2.VideoCapture(0) —————> Initialize the webcam
8
9 hands = mp_hands.Hands() —————> Initialize the Hand model
10
11 while True:
12     success, image = webcam.read() —————> Read a frame from the webcam
13
14     image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) —————> Convert the frame from BGR to RGB
15
16     results = hands.process(image_rgb) —————> Process the frame to detect
17                                     hand landmarks
18
19     if results.multi_hand_landmarks:
20         for hand_landmarks in results.multi_hand_landmarks:
21             mp_draw.draw_landmarks(image, hand_landmarks, mp_hands.HAND_CONNECTIONS)
22
23     cv2.imshow("Webcam", image)
24     cv2.waitKey(1)
```

Initialize the Hand model

Read a frame from the webcam

Convert the frame from BGR to RGB

Process the frame to detect hand landmarks

Print the detected hand landmarks' positions



Try on your self

Hand Count Finger Send To NodeRed

open your command prompt

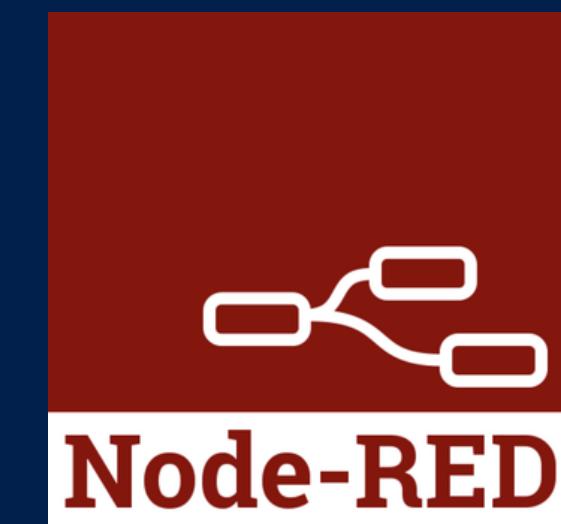
Q1

```
pip3 install mediapipe  
pip3 install opencv-python
```

Q2

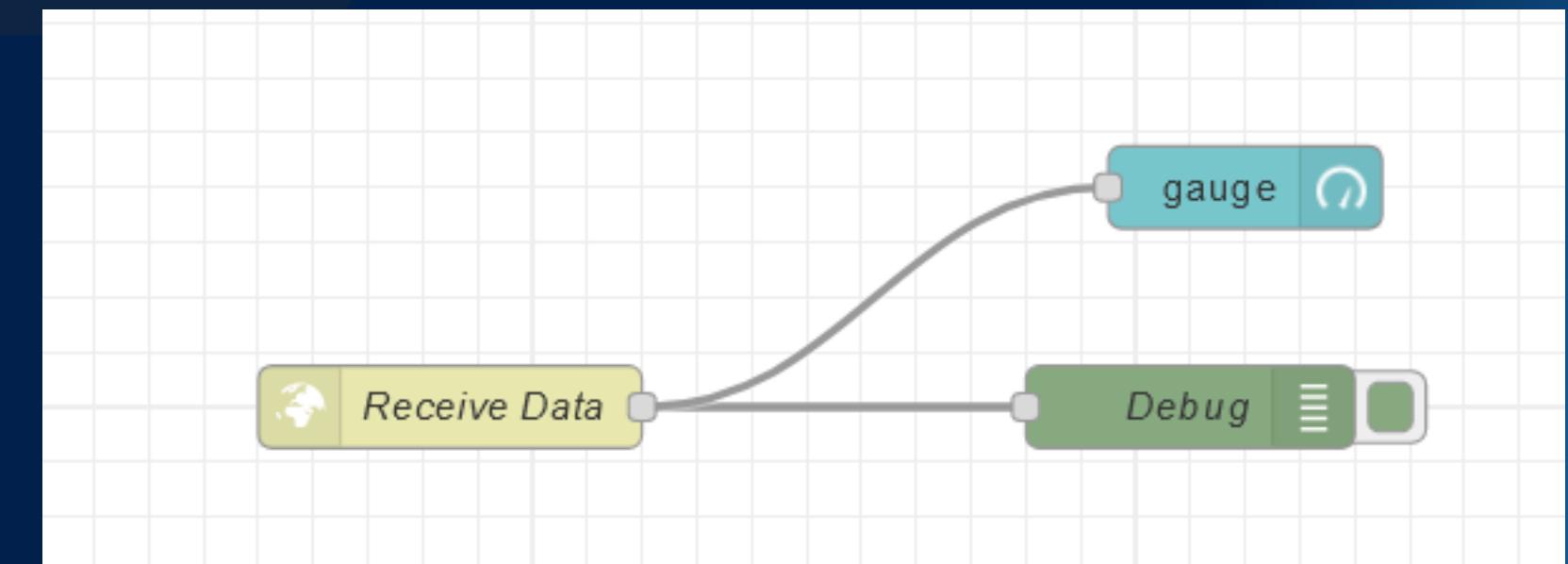
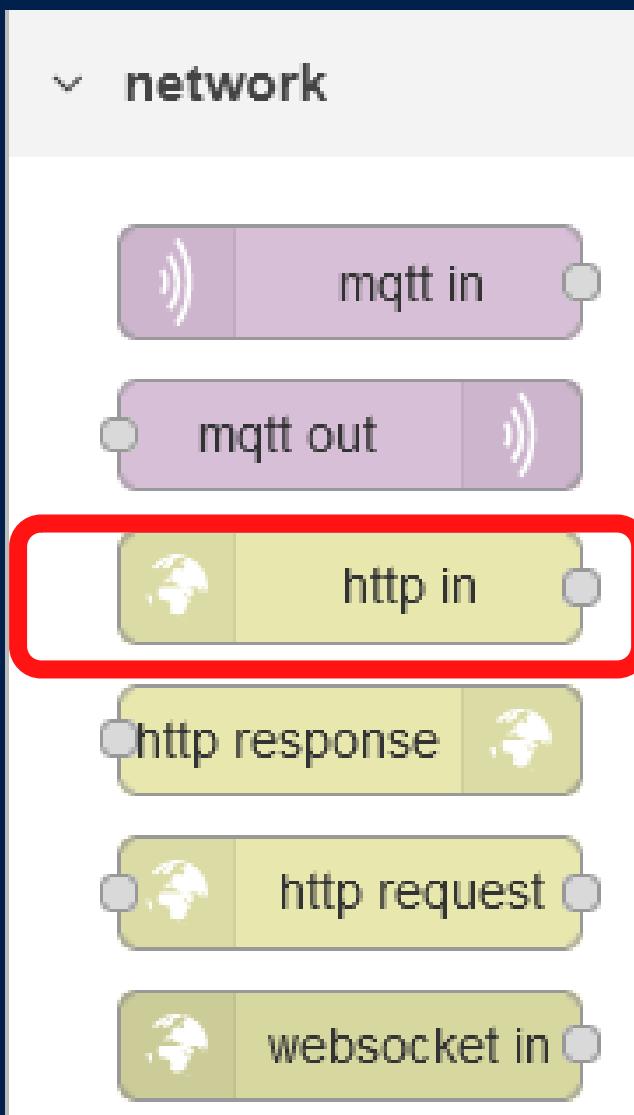
install NodeRed

```
pip3 install requests
```



Q3

Configure padlet in node red



double-click in nodes Receive Data

The configuration dialog for the 'Receive Data' node. It includes the following fields:

- Method: POST
- Accept file uploads?:
- URL: /receive-data **config your URL you need**
- Name: Receive Data

Q3

Hand Count Coding

Our Wisdom

```
1 import cv2
2 import mediapipe as mp
3 import requests
4 url = 'http://127.0.0.1:1880/receive-data' URL
5
6 mp_drawing = mp.solutions.drawing_utils
7 mp_drawing_styles = mp.solutions.drawing_styles
8 mp_hands = mp.solutions.hands
9
10 capture = cv2.VideoCapture(0)
11 with mp_hands.Hands(
12     model_complexity=0,
13     min_detection_confidence=0.5,
14     min_tracking_confidence=0.5) as hands:
15     while capture.isOpened():
16         success, image = capture.read()
17         if not success:
18             print('Ignored empty webcam\'s frame')
19             continue
20         image.flags.writeable = False
21         image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
22         results = hands.process(image)
23
24         image.flags.writeable = True
25         image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
26
27         fingerCount = 0
```

```
29     if results.multi_hand_landmarks:
30         for hand_landmarks in results.multi_hand_landmarks:
31             handIndex = results.multi_hand_landmarks.index(hand_landmarks)
32             handLabel = results.multi_handedness[handIndex].classification[0].label
33
34             handLandmarks = []
35
36             for landmarks in hand_landmarks.landmark:
37                 handLandmarks.append([landmarks.x, landmarks.y])
38
39             if handLabel == "Left" and handLandmarks[4][0] > handLandmarks[3][0]:
40                 fingerCount = fingerCount + 1
41             elif handLabel == "Right" and handLandmarks[4][0] < handLandmarks[3][0]:
42                 fingerCount = fingerCount + 1
43
44             if handLandmarks[8][1] < handLandmarks[6][1]:
45                 fingerCount = fingerCount + 1
46             if handLandmarks[12][1] < handLandmarks[10][1]:
47                 fingerCount = fingerCount + 1
48             if handLandmarks[16][1] < handLandmarks[14][1]:
49                 fingerCount = fingerCount + 1
50             if handLandmarks[20][1] < handLandmarks[18][1]:
51                 fingerCount = fingerCount + 1
52
53             mp_drawing.draw_landmarks(
54                 image,
55                 hand_landmarks,
56                 mp_hands.HAND_CONNECTIONS,
57                 mp_drawing_styles.get_default_hand_landmarks_style(),
58                 mp_drawing_styles.get_default_hand_connections_style()
59             )
60
```

```
61             cv2.putText(image, str(fingerCount), (50,450), cv2.FONT_HERSHEY_COMPLEX_SMALL, 3, (255,0,0), 10)
62             cv2.imshow('FingerCounting Apps',image)
```

Q4

FingerCount Send To NodeRed Coding

youtube : <https://www.youtube.com/watch?app=desktop&v=1iq9FxLxBIY>

```
64     if cv2.waitKey(1) == 27: # Check if the ASCII value of the pressed key is 27 (ESC key)
65         break
66     if cv2.waitKey(1) & 0xFF == ord('c'): # Check if the ASCII value of the pressed key is 99 (C key) // 32 is (backspace key)
67         # send http buffer string to http in Node-red not recall.
68     try:
69         response = requests.post(url, data=str(fingerCount), timeout=5) # Set timeout to 5 seconds
70         if response.status_code == 200:
71             print('Data sent successfully')
72         else:
73             print('Error sending data:', response.text)
74     except requests.Timeout:
75         print('Request timed out. Server did not respond in time.')
76     except requests.RequestException as e:
77         print('An error occurred:', e)
78 capture.release()
```

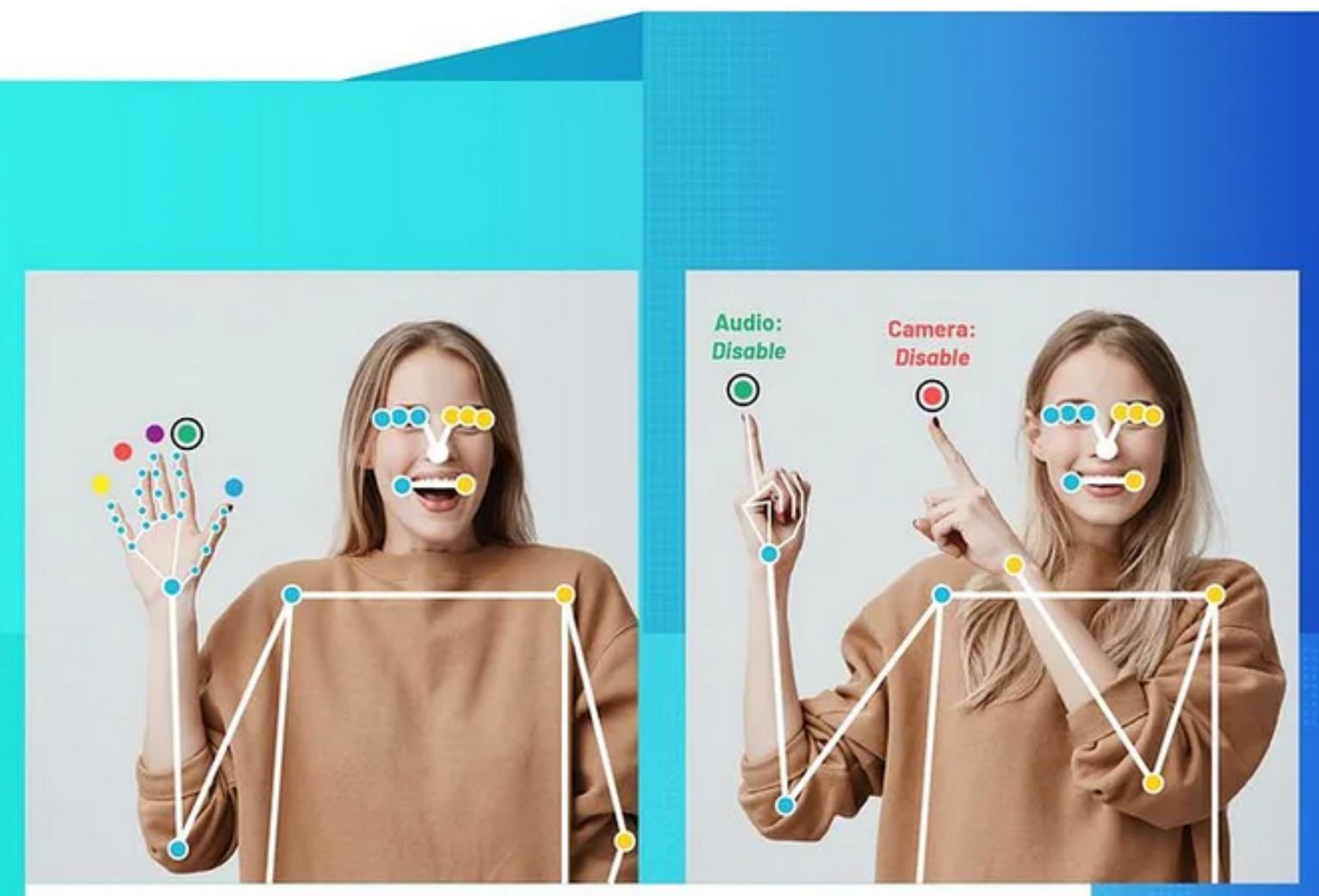
github code :



**[https://github.com/TheedBoyZ/Zero
To_Hero_2023](https://github.com/TheedBoyZ/Zero_To_Hero_2023)**

Intelligence

Google's MediaPipe Holistic Interface



This gesture control and command technique will take us to the next level. And allow us to use new innovations that other devices cannot do

It can be tried on MediaPipe Holistic Interface from Google at:

https://mediapipe.dev/demo/holistic_remote/



Thank You!

End of Session AI

