Universidad Autónoma de Tamaulipas
Facultad de Ingeniería Tampico

ASIGNATURA
Automatización y Robótica
9no. Semestre – Grupo "J"
2025 -3

TRABAJO
MediaPipe

UNIDAD
2 – Uso de Interfaces Naturales (NUI) para la Automatización y la Robótica

Docente: Dr. García Ruiz Alejandro H.

Integrantes:
Aldama Trinidad Alfonso Rene
Clemente Villegas José Adán
Cristóbal Francisco Jesús Marcelino
Domínguez Reyes Pavel Noel

Índice

Repositorio(s)

Unidad 2	Repositorio
Python	https://github.com/adanclv/AyR_2025_3.git

Conteo y suma de dedos

Descripción

Implementación de un sistema de visión por computadora que detecte las manos de una persona en tiempo real, identificando los dedos levantados y mostrar en pantalla:

- El número de dedos levantados si se detecta una sola mano.
- La suma de los dedos levantados si se detectan dos manos.

Desarrollo

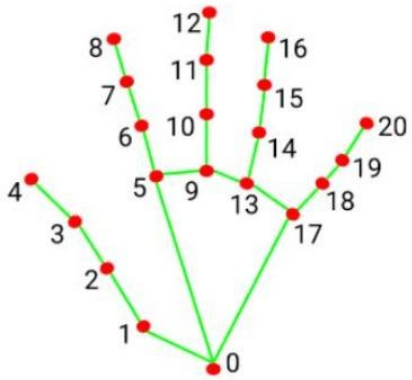
Código

```
import cv2
import mediapipe as mp

mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils

def contar_dedos(hand_landmarks, handedness):
    dedos = []

    if handedness == "Right":
        if hand_landmarks.landmark[4].x < hand_landmarks.landmark[3].x:
            dedos.append(1)
        else:
            dedos.append(0)
    else:  # Left
        if hand_landmarks.landmark[4].x > hand_landmarks.landmark[3].x:
            dedos.append(1)
        else:
            dedos.append(0)

    for tip in [8, 12, 16, 20]:
        if hand_landmarks.landmark[tip].y < hand_landmarks.landmark[tip - 2].y:
            dedos.append(1)
        else:
            dedos.append(0)

    return sum(dedos)

cap = cv2.VideoCapture(1)

with mp_hands.Hands(max_num_hands=2, min_detection_confidence=0.75) as manos:
    while True:
        ret, frame = cap.read()
        if not ret:
            break
```

```
        frame = cv2.flip(frame, 1)
        rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        results = manos.process(rgb)

        if results.multi_hand_landmarks:
            numeros = []
            for hand_landmarks, hand_handedness in
zip(results.multi_hand_landmarks,
results.multi_handedness):
                label = hand_handedness.classification[0].label  # "Left" o
"Right"
                dedos = contar_dedos(hand_landmarks, label)
                numeros.append(dedos)

                mp_drawing.draw_landmarks(frame, hand_landmarks,
mp_hands.HAND_CONNECTIONS)

                        if len(numeros) == 2:
                suma = numeros[0] + numeros[1]
                cv2.putText(frame, f"{numeros[0]} + {numeros[1]} = {suma}",
                            (50, 100), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0),
3)

            elif len(numeros) == 1:
                cv2.putText(frame, f"Numero: {numeros[0]}",
                            (50, 100), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0),
3)

        cv2.imshow("Suma con manos", frame)
        if cv2.waitKey(1) & 0xFF == 27:
            break

cap.release()
cv2.destroyAllWindows()
```

Se utiliza el modelo de MediaPipe Hands que detecta 21 landmarks de cada mano. Permite identificar articulaciones y la posición de los dedos.

Hand Land Marks

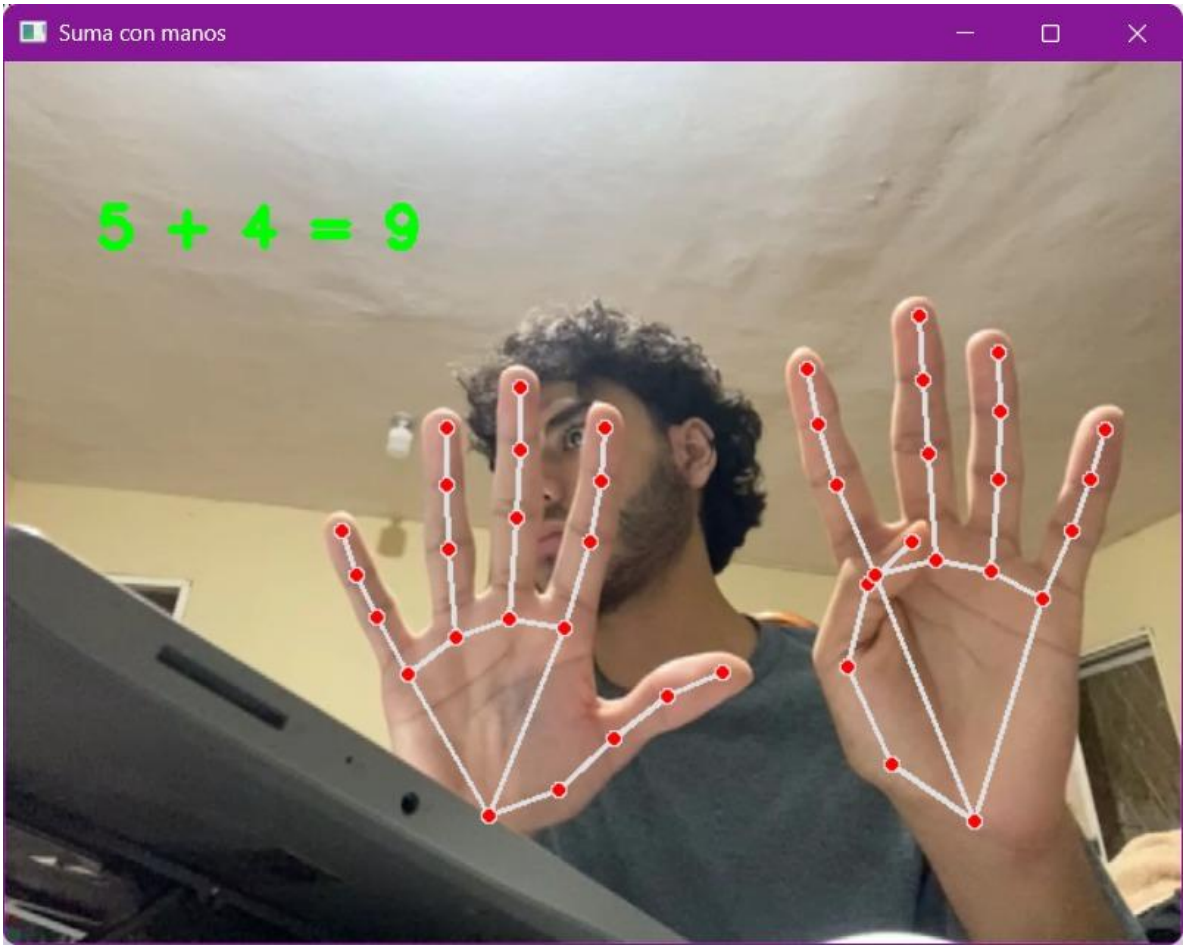


0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP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

La función `contar_dedos(hand_landmarks, handedness)` recibe los puntos de la mano y determina cuántos dedos están levantados según el eje correspondiente:

- El pulgar se evalúa comparando la coordenada X del punto de la punta (landmark[4]) con la base (landmark[3]), ya que el movimiento principal es lateral.
- Los demás dedos (índice, medio, anular y meñique) se evalúan con la coordenada Y de la punta (landmark[tip]) respecto a la articulación anterior (landmark[mcp]), pues se levantan en el eje vertical.

Resultado



Conclusión

El ejercicio demuestra cómo combinar detección de landmarks con lógica condicional para interpretar gestos humanos, mostrando a MediaPipe como es una herramienta poderosa que ofrece detección rápida y precisa.