

PROGRAMA ACADÉMICO

Departamento de ciencias básicas

UNIDAD DE ESTUDIO

robótica

INTEGRANTES

Kevyn Estiven Marín Nivia

Manuel Orlando Reyes

DOCENTE

Edwin Nikolay Prieto Parrado

Trabajo

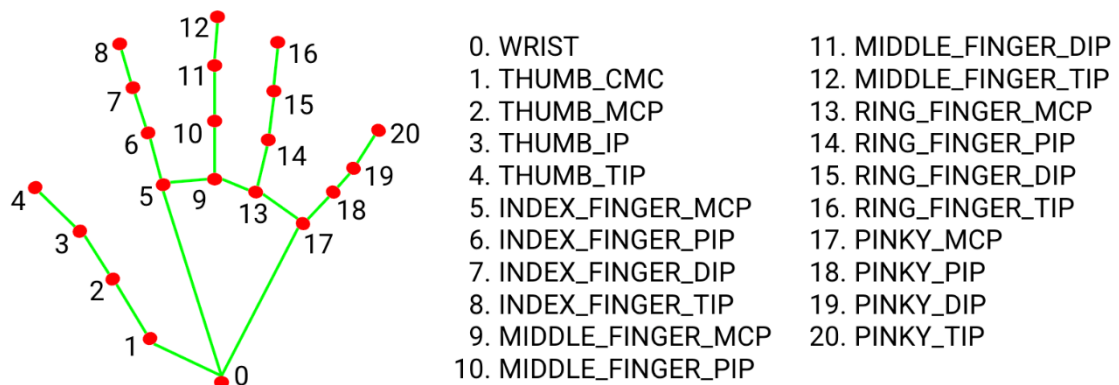
Obtención de información de ángulos entre articulaciones con Mediapipe

¿Cómo extraer la información numérica del movimiento articular del brazo?

Modelo de referencia de mano

Después de la detección de la palma de la mano en toda la imagen, nuestro modelo posterior de puntos de referencia de la mano realiza una localización precisa del punto clave de 21 coordenadas 3D de los nudillos de la mano dentro de las regiones de la mano detectadas a través de la regresión, es decir, la predicción directa de coordenadas. El modelo aprende una representación de pose de mano interna consistente y es robusto incluso para manos parcialmente visibles y autooclusiones.

Para obtener datos reales sobre el terreno, hemos anotado manualmente ~30 000 imágenes del mundo real con 21 coordenadas 3D, como se muestra a continuación (tomamos el valor Z del mapa de profundidad de la imagen, si existe por coordenada correspondiente). Para cubrir mejor las posibles poses de la mano y proporcionar una supervisión adicional sobre la naturaleza de la geometría de la mano, también representamos un modelo de mano sintética de alta calidad sobre varios fondos y lo asignamos a las coordenadas 3D correspondientes.



Por lo anterior cada distancia se puede calcular ya que cada punto tiene una coordenada específica. De ser necesario obtener información numérica se puede suministrar de las 21 coordenadas existentes.

Código de detección de manos

```
import cv2

import mediapipe as mp

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

# For webcam input:

cap = cv2.VideoCapture(0)

with mp_hands.Hands(
    model_complexity=0,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as hands:
    while cap.isOpened():
        success, image = cap.read()

        if not success:
            print("Ignoring empty camera frame.")

            # If loading a video, use 'break' instead of 'continue'.

            continue
```

```
# To improve performance, optionally mark the image as not writeable to
# pass by reference.

image.flags.writeable = False

image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

results = hands.process(image)


# Draw the hand annotations on the image.

image.flags.writeable = True

image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)


# Initially set finger count to 0 for each cap

fingerCount = 0


if results.multi_hand_landmarks:

    for hand_landmarks in results.multi_hand_landmarks:

        # Get hand index to check label (left or right)

        handIndex = results.multi_hand_landmarks.index(hand_landmarks)

        handLabel = results.multi_handedness[handIndex].classification[0].label


        # Set variable to keep landmarks positions (x and y)

        handLandmarks = []
```

```

# Fill list with x and y positions of each landmark

for landmarks in hand_landmarks.landmark:

    handLandmarks.append([landmarks.x, landmarks.y])


# Test conditions for each finger: Count is increased if finger is
#   considered raised.

# Thumb: TIP x position must be greater or lower than IP x position,
#   depending on hand label.

if handLabel == "Left" and handLandmarks[4][0] > handLandmarks[3][0]:

    fingerCount = fingerCount+1

elif handLabel == "Right" and handLandmarks[4][0] < handLandmarks[3][0]:

    fingerCount = fingerCount+1


# Other fingers: TIP y position must be lower than PIP y position,
#   as image origin is in the upper left corner.

if handLandmarks[8][1] < handLandmarks[6][1]:    #Index finger

    fingerCount = fingerCount+1

if handLandmarks[12][1] < handLandmarks[10][1]:    #Middle finger

    fingerCount = fingerCount+1

if handLandmarks[16][1] < handLandmarks[14][1]:    #Ring finger

    fingerCount = fingerCount+1

if handLandmarks[20][1] < handLandmarks[18][1]:    #Pinky

```

```
fingerCount = fingerCount+1
```

```
# Draw hand landmarks
```

```
mp_drawing.draw_landmarks(  
    image,  
    hand_landmarks,  
    mp_hands.HAND_CONNECTIONS,  
    mp_drawing_styles.get_default_hand_landmarks_style(),  
    mp_drawing_styles.get_default_hand_connections_style())
```

```
# Display finger count
```

```
cv2.putText(image, str(fingerCount), (50, 450), cv2.FONT_HERSHEY_SIMPLEX, 3,  
(255, 0, 0), 10)
```

```
# Display image
```

```
cv2.imshow('MediaPipe Hands', image)
```

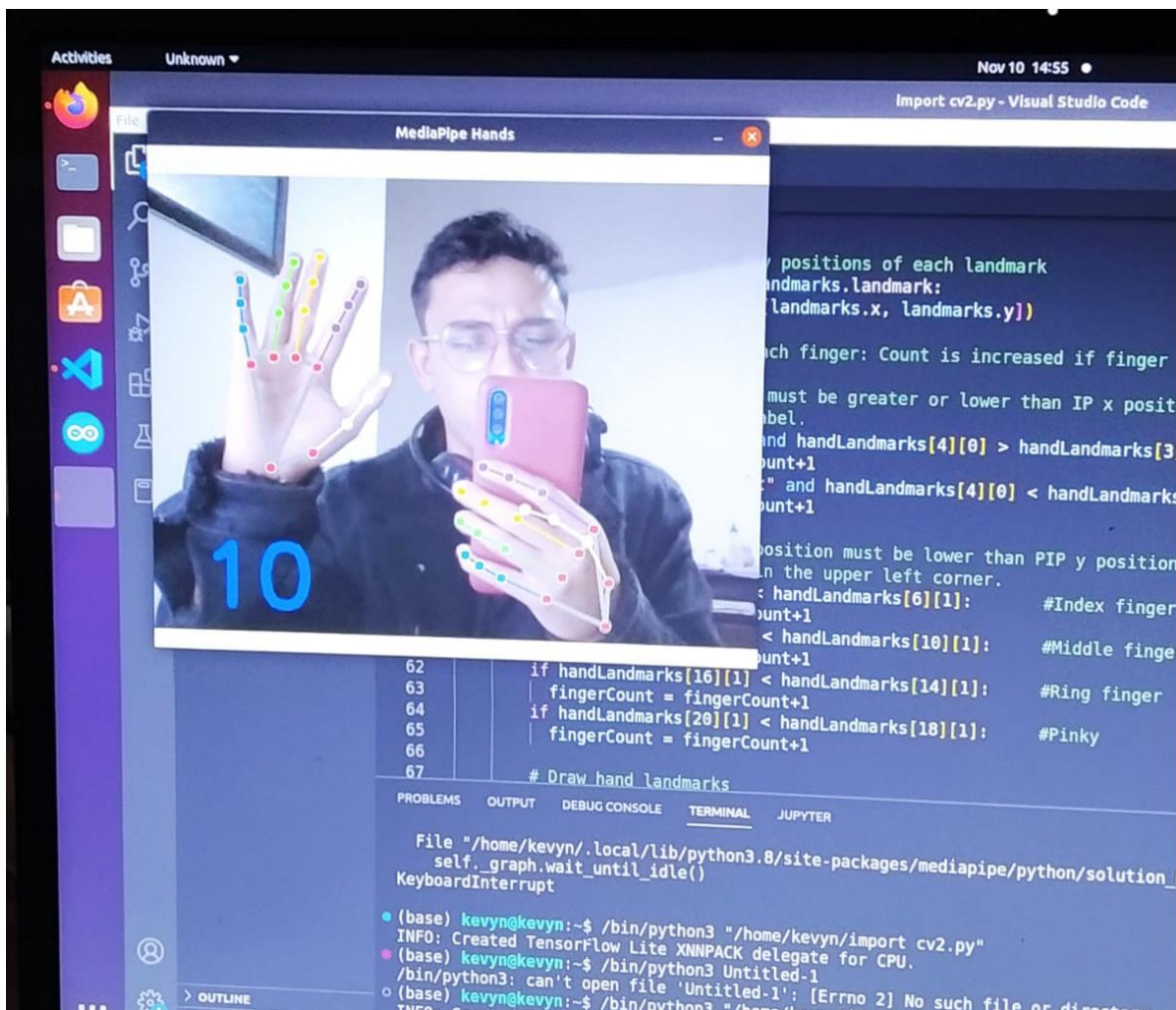
```
if cv2.waitKey(5) & 0xFF == 27:
```

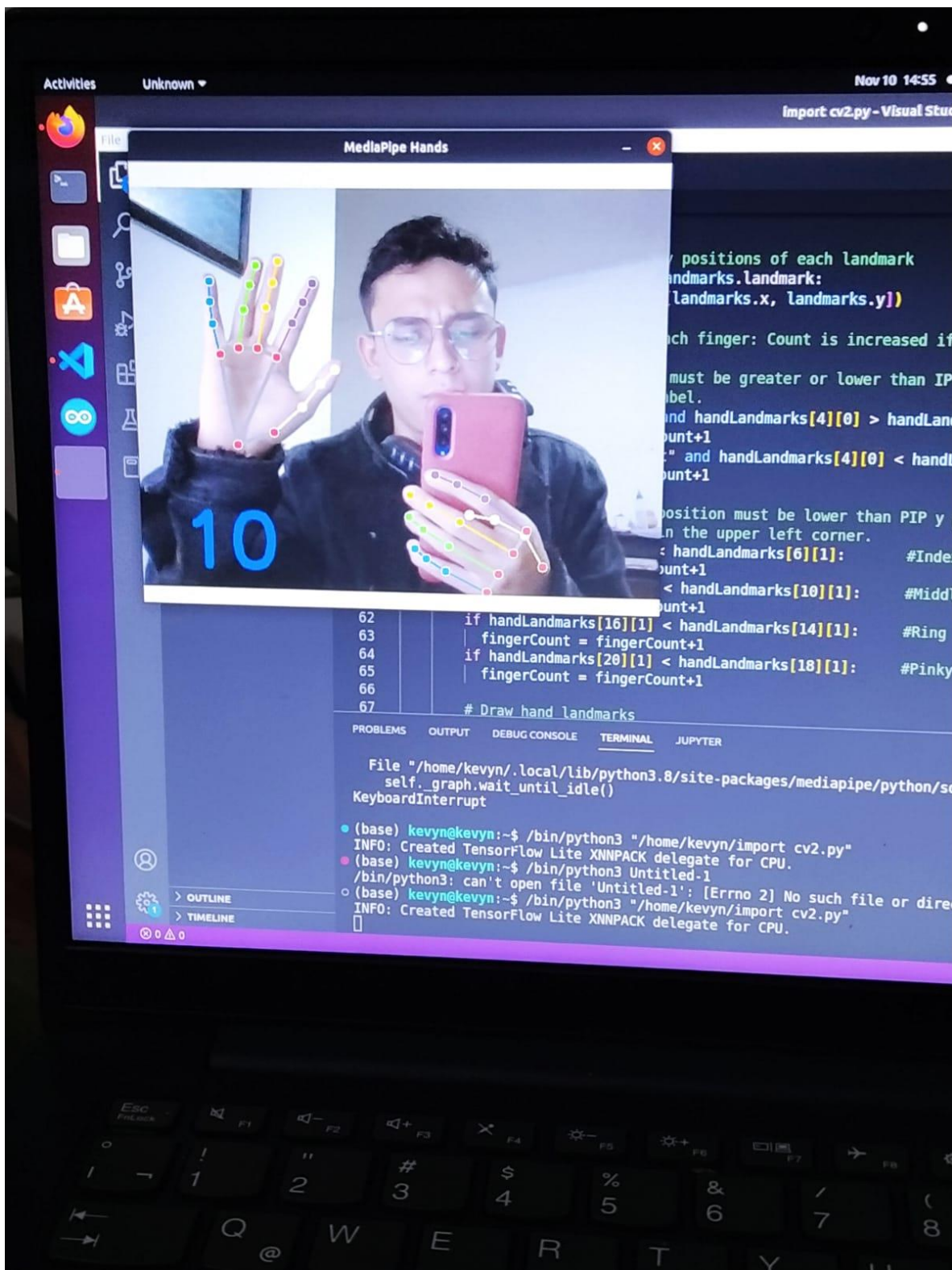
```
    break
```

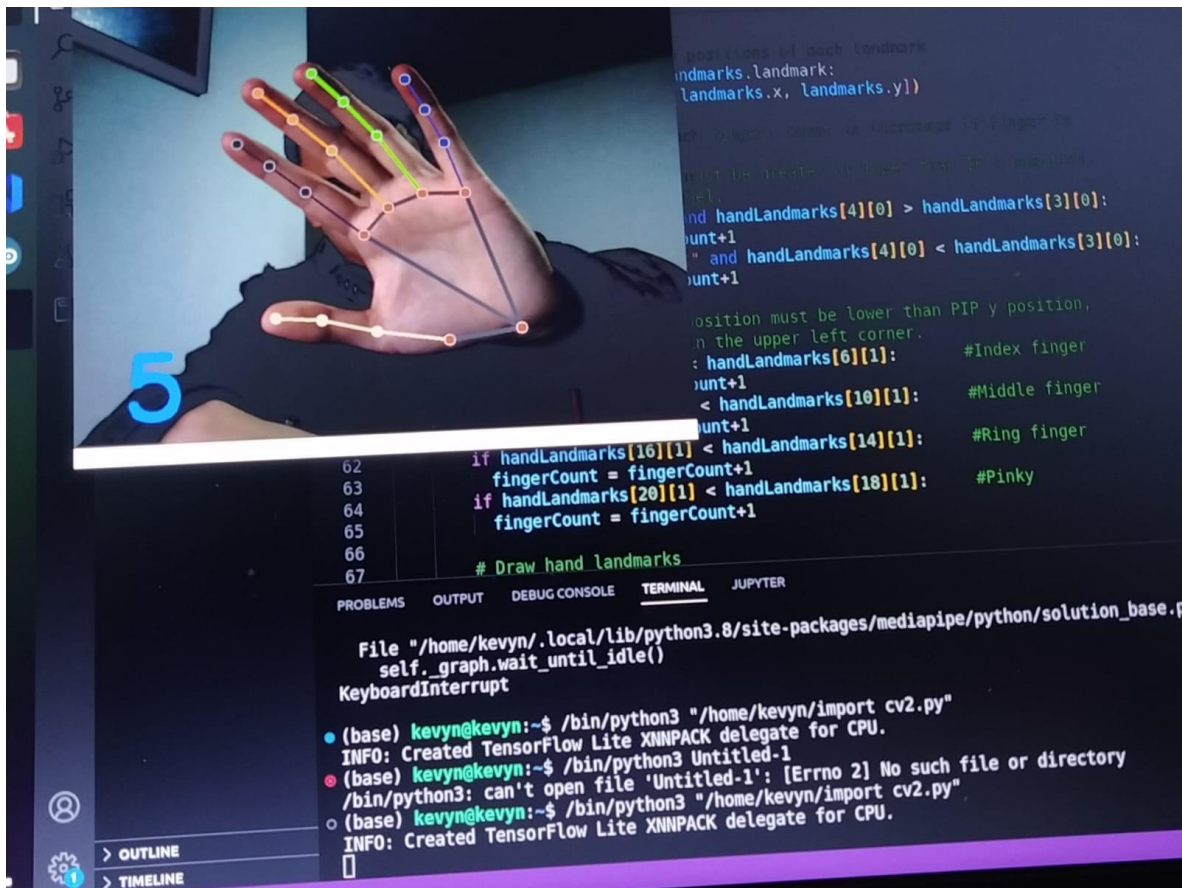
```
cap.release()
```

Criterio de aceptación

Ejemplo real de movimiento articular con la librería mediapipe (u openpose) desde su PC.







Trabajos similares empleando movimiento por medio de cámara y transmitiendo la información para el movimiento de un servomotor

<https://github.com/rizkydermawan1992/handtracking/blob/Main/pwm/skematik%20.jpg>

