

DETECCIÓN Y SEGUIMIENTO DE PUNTOS DE REFERENCIAS EN MANOS CON MEDIAPIPE

Antes de empezar a explicar el código hablemos un poco sobre MediaPipe, esta librería fue creada por Google para el reconocimiento y seguimiento de objetos en imágenes y videos, una de las aplicaciones más importantes de esta librería es la detección de puntos de referencia en las manos y seguimiento de estas mismas, sin duda una herramienta de gran ayuda para proyectos en personas con discapacidad auditiva y del habla.

Para este proyecto identificaremos los puntos de referencia y seguimiento de manos en tiempo real, con el fin de conocer el funcionamiento básico de esta herramienta y tener los conocimientos para crear distintos proyectos en relación con esto a futuro.

Estaremos utilizando dos librerías, la primera OpenCV, que será la encargada del video en tiempo real, y por último MediaPipe, que será la encargada de lo relacionado con las manos. Importamos estas dos librerías y estaremos listos para comenzar.

```
import cv2
import mediapipe as mp
```

Inicializamos el módulo `.solutions` de la librería Mediapipe, con la función `.hands`, lo que hacemos en esta inicialización es mandar a traer las funciones necesarias para el código, la función `.hands` es la que nos ayudara en las detecciones y seguimientos de las manos.

```
mp_hands = mp.solutions.hands
```

Ahora configuraremos los parámetros para la detección, para esto usaremos la función `.Hands()`, no hay que confundir esta función con la función de la inicialización porque no son lo mismo, `.hands` solo inicializa y `.Hands()` configura los parámetros.

Los parámetros que usaremos para `.Hands()` son:

- `static_image_mode`: este parámetro booleano se encarga de diferenciar entre una imagen y un video, si es True (valor que tiene por defecto), se tomara como que estaremos usando una imagen para la detección y en caso de ser False, se tomara como que usaremos un video con movimiento.

- `max_num_hands`: este parámetro tiene dos valores posibles, 1 y 2; si usamos 1 detectará solo una mano, caso contrario de ser 2 se detectarán dos manos.
- `min_detection_confidence` y `min_tracking_confidence`: estos dos parámetros flotantes son los encargados de ajustar el porcentaje de confianza para la detección y seguimiento respectivamente.
- `model_complexity`: aquí indicamos la complejidad del modelo, si es 0 será más rápido, pero menos preciso y si es 2 será más preciso, pero también más lento.

```
hands = mp_hands.Hands(static_image_mode=True,
                        max_num_hands=2, min_detection_confidence=0.5,
                        min_tracking_confidence=0.5, model_complexity=0)
```

Una vez hecho esto, vamos a inicializar la función para dibujar los puntos de referencia, esto mediante `.drawing_utils`.

```
mp_drawing = mp.solutions.drawing_utils
```

Teniendo inicializado tanto el detector de manos y la función de dibujo, vamos a inicializar la captura de video mediante la función `.videoCapture()`, y así poder empezar a trabajar en el algoritmo completo.

```
Cap = cv2.VideoCapture(0)
```

Pasaremos a crear nuestro ciclo while en el cual haremos las configuraciones necesarias para el funcionamiento correcto de la librería MediaPipe, primero leeremos nuestros frames a través de la función `.read()`. Teniendo esto, haremos un efecto espejo con la función `.flip()`, esto para que la cámara este de acuerdo a nuestros movimientos y no de forma contraria, esta función es en dado caso de que la cámara que estamos usando lo necesite. En dado caso de usar esta función `.flip()`, le daremos como parámetros la variable a la cual aplicaremos el

efecto espejo y como segundo parámetro el flipCode que es un número entero correspondiente al efecto que necesitamos, los distintos valores que puede tener son:

- 0: invierte la imagen horizontalmente.
- 1: invierte la imagen verticalmente.
- -1: invierte tanto verticalmente como horizontalmente.

Por último, configuraremos el cambio de escala de color de BGR a RGB, esto porque las herramientas de MediaPipe funcionan en escala RGB, este cambio lo haremos mediante la función `.cvtColor()`, los parámetros que usaremos serán:

- La variable a la que le aplicaremos el cambio de escala de color.
- El código de cambio, que en este caso será `cv2.COLOR_BGR2RGB`.

La parte del código con las configuraciones realizadas se escribiría de la siguiente manera:

```
while True:
    Ret, Frames = Cap.read()
    Frames = cv2.flip(Frames, 1)
    Frames_rgb = cv2.cvtColor(Frames, cv2.COLOR_BGR2RGB)
```

Ahora, almacenaremos los datos de los frames en una variable haciendo uso de la función `.process()`, y como parámetros la variable que contiene el cambio de BGR A RGB.

Esta función se encarga de almacenar toda la información de los frames para luego devolverla y usarla de forma adecuada en el código.

```
Results = hands.process(Frames_rgb)
```

Crearemos la configuración correspondiente para los círculos y líneas que se mostrarán en las manos al ser detectadas, para ambas configuraciones usaremos la misma función la cual es `.DrawingSpec()`. Primero, empezaremos con la configuración de los círculos, a la cual le daremos tres parámetros:

- El color con el que se mostraran los círculos (escala BGR).
- El grosor de los círculos.
- El radio de los círculos.

Y para las líneas usaremos solo dos:

- El color de las líneas (escala BGR).
- El grosor de las líneas.

```
Circles_color=mp_drawing.DrawingSpec(color=(255,0,0),thickness=4, circle_radius=2)
Lines_color=mp_drawing.DrawingSpec(color=(0,0,255),thickness=3)
```

Ahora, pasaremos a crear un condicional `if`, en este condicional verificaremos que nuestra variable `Results`, que es la que contiene los datos de detección, en conjunto con la función `.multi_hands_landmarks`, que contiene las coordenadas tridimensionales de la mano, tenga información que devolvernos. Una vez teniendo este condicional y teniendo datos que entregar, pasaremos a crear un ciclo `for` con el cual dibujaremos los puntos de referencia en las manos, utilizaremos una variable con el nombre `Landmarks` la cual estará iterando en `Results.multi_hands_landmarks`.

Dentro de ese ciclo `for` usaremos la variable `mp_drawing`, que contiene las herramientas de dibujo previamente inicializadas, junto con la función `.draw_landmarks()`, a esta función le daremos los siguientes parámetros:

- La variable en la cual dibujaremos los puntos de referencia.
- La variable con la lista puntos de referencia detectados, en este caso será `Landmarks`.
- Usaremos `mp_hands.HAND_CONNECTIONS`, que es la lista de conexiones entre los puntos de referencia, estas serán las líneas mostradas en la mano.
- La variable con la configuración de color y tamaño de los círculos previamente configurada.
- La variable con la configuración de color y tamaño de las líneas previamente configurada.

Teniendo todo esto, las líneas de código explicadas se verían de la siguiente manera:

[illegible]

Con esto terminado, podremos mostrar la video captura con los frames y las manos con sus puntos de referencia y líneas de conexión, para esto usaremos la función `.show()`, y como parámetros el nombre que tendrá nuestra ventana y la variable que mostraremos.

```
cv2.imshow("LANDMARKS",Frames)
```

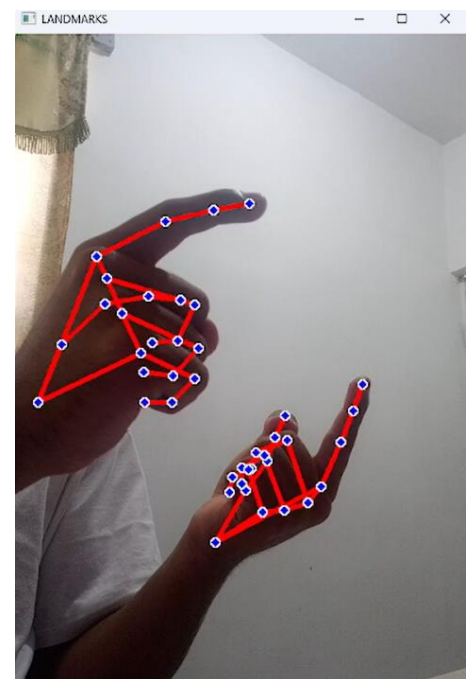
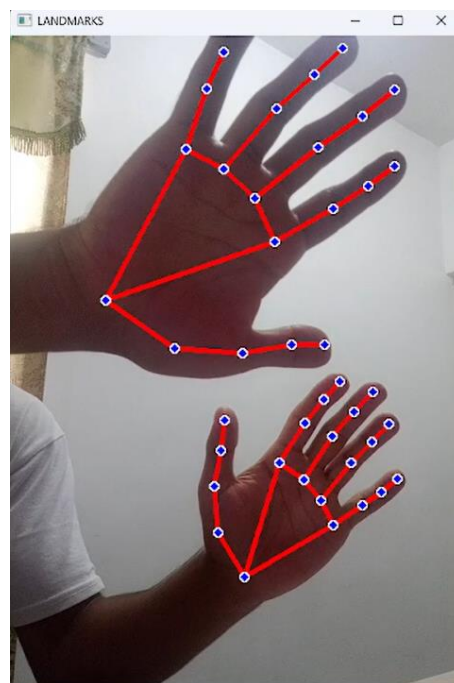
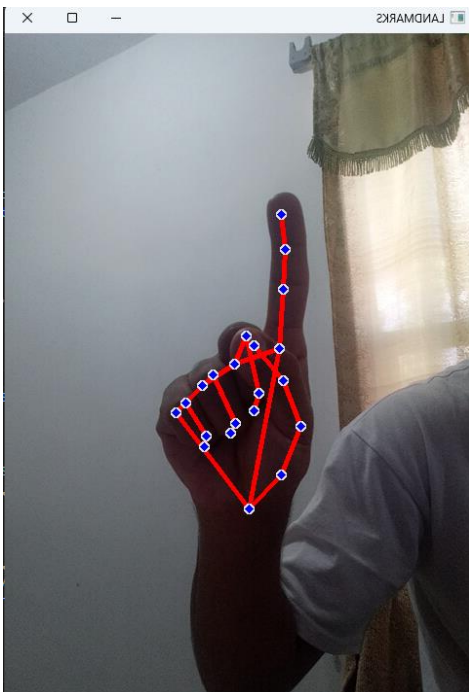
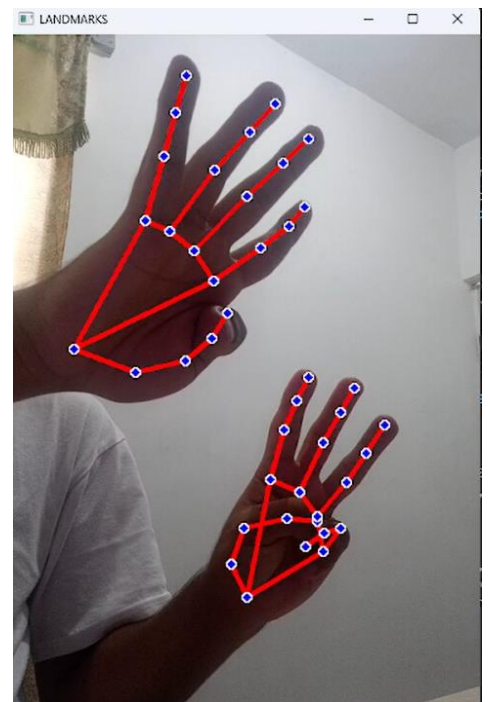
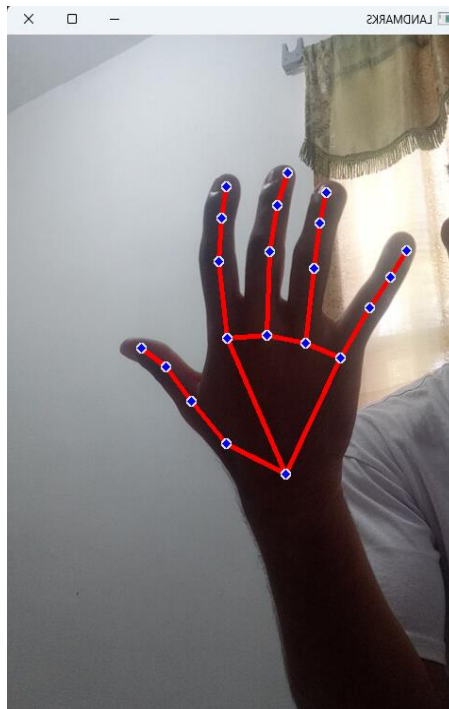
Indicamos que con la letra “q” o “Q”, saldremos de nuestro programa esto mediante la función `.waitKey(1)` y un condicional `if`.

```
t = cv2.waitKey(1)
if t == ord('q') or t == ord("Q"):
    break
```

Ya para terminar, borraremos nuestra captura de video con la función `.release()`, y también nuestra ventana con `.destroyAllWindows()`.

```
Cap.release()
cv2.destroyAllWindows()
```

RESULTADOS



Si llegaste hasta esta parte te agradezco el tiempo que me regalaste y espero te haya ayudado con tus dudas o inquietudes de aprender más, esto solo es un pequeño fragmento de explicación si quieres seguir aprendiendo más puedes visitar las siguientes páginas que te ayudaran a seguir aprendiendo y mejorando en el tema de la visión artificial y Python.

- 1.- <https://omes-va.com/>
- 2.- <https://www.pythonpool.com/>
- 3.- <https://hetpro-store.com/TUTORIALES/>
- 4.- <https://learnopencv.com/>

Lo principal es la sabiduría; adquiere sabiduría, Y con todo lo que obtengas adquiere inteligencia. Proverbios 4:7