



HanDraw

Trabajo final. Visión por Computador

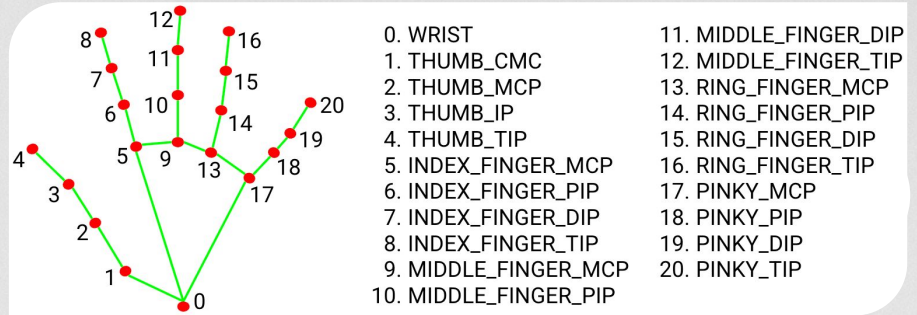
Willy Escovilla Biason
Eduardo Etopa Lechuga

HanDraw

El programa permite dibujar a tiempo real sobre la imagen de video por webcam usando la detección de manos.

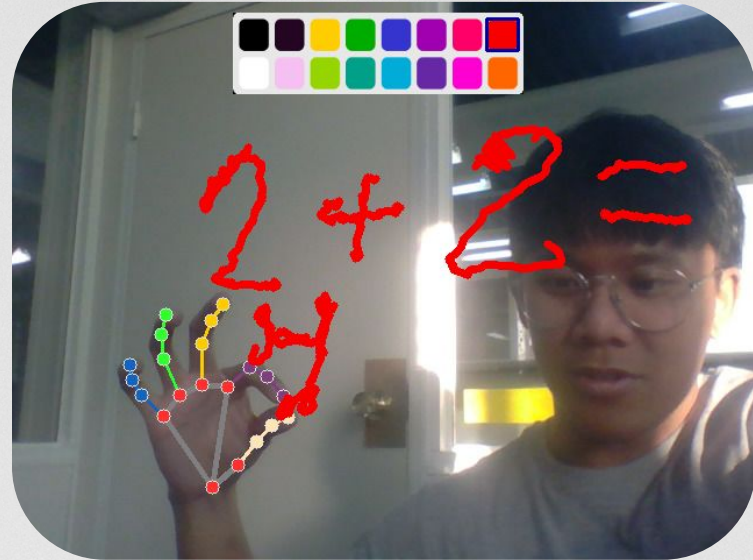
Empleando MediaPipe el programa detecta puntos de referencia en una mano.

El programa detecta que se quiere dibujar cuando detecta una mano que está haciendo el gesto de agarrar un lápiz.



Funcionalidades desarrolladas

- Pintar sobre la imagen.



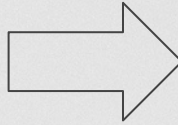
Funcionalidades desarrolladas

- Cambiar color pulsando la 'd'.
Avanza en orden por la paleta.



Funcionalidades desarrolladas

- Deshacer las puntos manteniendo la 'z'.
- Deshacer líneas enteras o trazos pulsando la 'b'.



Código del programa

- Función `add_points_to_list`. Comprueba que se esté realizando el gesto, en tal caso, introduce a la lista de dibujos la posición en donde se quiere dibujar.

```
def add_points_to_list(list_of_draws, hand_landmarks, point_color):
    thumb_x = hand_landmarks.landmark[4].x
    thumb_y = hand_landmarks.landmark[4].y
    thumb_z = hand_landmarks.landmark[4].z

    index_x = hand_landmarks.landmark[8].x
    index_y = hand_landmarks.landmark[8].y
    index_z = hand_landmarks.landmark[8].z

    # Distancia entre la punta del pulgar y la del índice
    distance = math.sqrt((thumb_x - index_x)**2 + (thumb_y - index_y)**2)

    # Umbral de cercanía
    threshold = 0.07

    if distance < threshold:
        midpoint = ((thumb_x + index_x) / 2, (thumb_y + index_y) / 2, (thumb_z + index_z) / 2, point_color)
        list_of_draws.append(midpoint)

    # Añadir una posición sin uso (-1, -1, -1) para cuando no se este dibujando
    elif list_of_draws and list_of_draws[-1][0] != (-1, -1, -1):
        list_of_draws.append((-1, -1, -1, (255, 255, 255)))
```

Código del programa

- Función draw_lines. Recorre la lista de dibujos y une los puntos con líneas para crear el dibujo en la imagen de la webcam.

```
def draw_lines(list_of_draws, image):  
    # Funcion para juntar todos los puntos que se han creado de forma consecutiva para crear el dibujo  
    x, y, z = (-1, -1, -1)  
  
    for position in list_of_draws:  
        prev_x, prev_y, prev_z = x, y, z  
        x, y, z, color_point = position  
  
        # No unir puntos si se paro de dibujar  
        if (x, y, z) == (-1, -1, -1) or (prev_x == -1 and prev_y == -1 and prev_z == -1):  
            continue  
  
        height, width, deep = image.shape  
        midpoint_x, midpoint_y, midpoint_z = int(x * width), int(y * height), int(z * deep)  
        prev_midpoint_x, prev_midpoint_y, prev_midpoint_z = int(prev_x * width), int(prev_y * height), int(prev_z * deep)  
  
        # Evitar dibujar sobre la paleta  
        if 190 < prev_midpoint_x < 460 and 190 < midpoint_x < 460 and 0 < prev_midpoint_y < 90 and 0 < midpoint_y < 90:  
            continue  
  
        cv2.line(image, (midpoint_x, midpoint_y), (prev_midpoint_x, prev_midpoint_y), color_point, 5)
```

Código del programa

- Primero inicializar la clase Hands y luego a cada frame de la webcam aplicarle el detector de manos.
- Si se han encontrado resultados de manos detectadas se realiza todo el proceso de dibujado.

```
# Inicializar clase Hands
with mp_hands.Hands(
    model_complexity=0,
    min_detection_confidence=0.5,
    #max_num_hands=2,
    min_tracking_confidence=0.5) as hands:

    # Lista para almacenar las posiciones en donde se dibuja con su color
    list_of_draws = []

    while cap.isOpened():
        success, image = cap.read()
        if not success:
            print("Ignoring empty camera frame.")
            continue

        # Almacenar los datos del detector de manos sobre el frame
        image.flags.writeable = False
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        results = hands.process(image)
        image.flags.writeable = True
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
```

```
if results.multi_hand_landmarks:
    for hand_landmarks in results.multi_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()
        )
        add_points_to_list(list_of_draws, hand_landmarks, colors_dict[color_name][0])

    draw_lines(list_of_draws, image)
```


Gracias!

