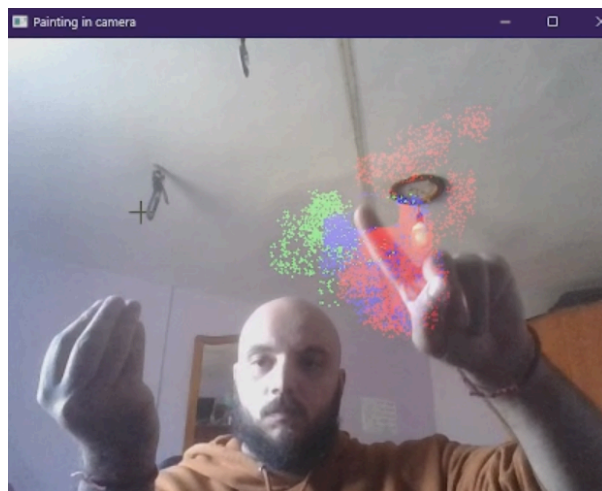
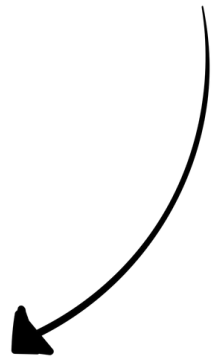
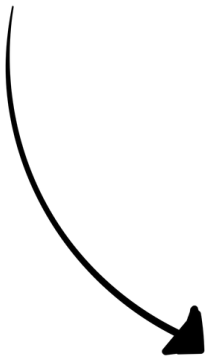
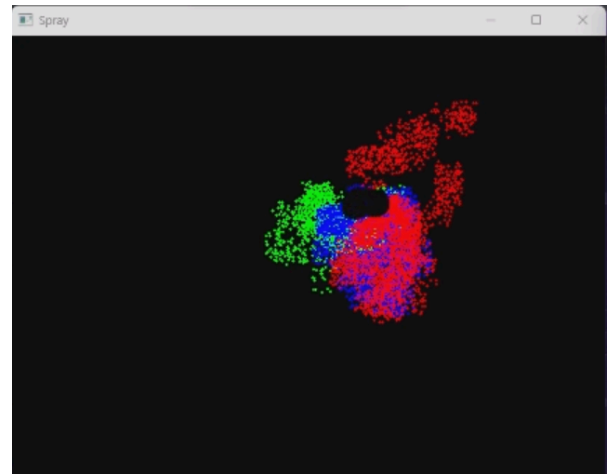
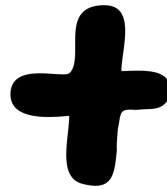
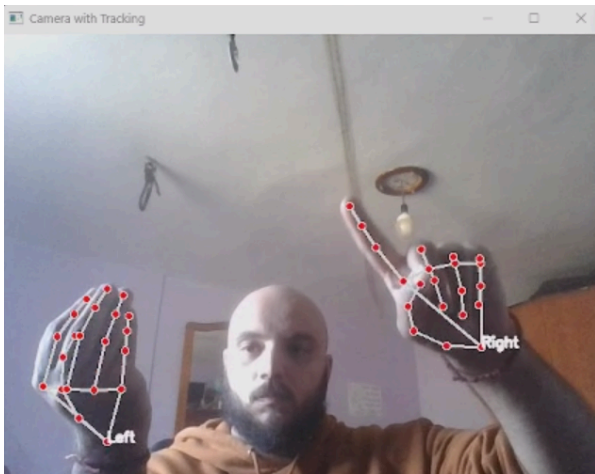


Memoria de Proyecto Final

Visión por Computador

Reconocimiento de gestos utilizando MediaPipe sin entrenamiento de modelo



Nauzet Déniz Perdomo
Alejandro Rodríguez Moreno

1. Motivación/argumentación del trabajo.....	3
2. Objetivo de la propuesta.....	3
3. Descripción técnica del trabajo realizado.....	3
4. Fuentes y tecnologías utilizadas.....	5
5. Conclusiones y propuestas de ampliación.....	5
6. Indicación de herramientas/tecnologías con las que les hubiera gustado contar.....	5
7. Créditos materiales no originales del grupo.....	6

1. Motivación/argumentación del trabajo

Durante la explicación del proyecto final para la asignatura de Visión por Computador se mostraron varias demostraciones de MediaPipe, siendo una de ellas una que reconocía las manos, y mediante un modelo, daba un índice de fiabilidad que indicaba qué gesto se utilizaba.

Esto despertó nuestra curiosidad sobre hasta qué punto sería posible reproducir resultados parecidos en el reconocimiento de gestos, pero sin utilizar un modelo. Es decir, hasta qué punto podíamos imitar el uso de un modelo, utilizando aritmética básica mediante los landmarks que proporciona el reconocimiento de manos de MediaPipe.

A su vez, por distinguirnos un poco, y no mantener la misma demostración que indicaba qué gesto se estaba utilizando, nos decidimos por convertir la cámara en un lienzo, permitiendo con el dedo índice de la mano derecha indicar dónde se pinta en la pantalla y con la mano izquierda mediante distintos gestos indicar si se pinta de verde (Puño cerrado), azul (Signo V), rojo (Pulgar hacia arriba), se borra (Juntar todos los dedos) o cambiar entre modo pincel y spray (Los 3 dedos centrales hacia arriba).

Con motivos de permitir ver el proceso interno del proyecto, se decidió por tener tres salidas distintas:

1. La primera muestra la detección de las manos, con las landmarks para comprobar el tracking.
2. La segunda muestra el lienzo sobre el que se pinta, que se crea de las mismas dimensiones que la cámara.
3. La tercera sería la salida real y final del proyecto, que muestra la entrada de la cámara añadiendo, solamente, lo pintado en el lienzo.

2. Objetivo de la propuesta

El objetivo del proyecto final es el reconocimiento, a través de una cámara, de las manos, utilizando la librería MediaPipe y, sin utilizar un modelo, reconocer gestos y catalogarlos para interactuar con el vídeo recibido por la cámara como si fuera un lienzo, los gestos permitiendo cambiar entre diferentes colores, o borrar lo pintado.

3. Descripción técnica del trabajo realizado

El proyecto tuvo tres etapas de desarrollo, siendo estas las siguientes:

1. En primer lugar, la implementación de la librería MediaPipe para permitir el reconocimiento de las manos básico. Esta implementación utiliza las distintas funciones y utilidades que permite MediaPipe.

Para la detección se utiliza la función `process()` del módulo `hands`, que devuelve un diccionario con los resultados. Esto nos permite saber si hay más de una mano, si la mano detectada es izquierda o derecha, y por cada mano devuelve 21 landmarks, permitiendo así tener un mapeado de estas.

2. A continuación, se desarrolló la detección de los distintos gestos, tomando puntos de las landmarks y realizando distintos cálculos con ellos.

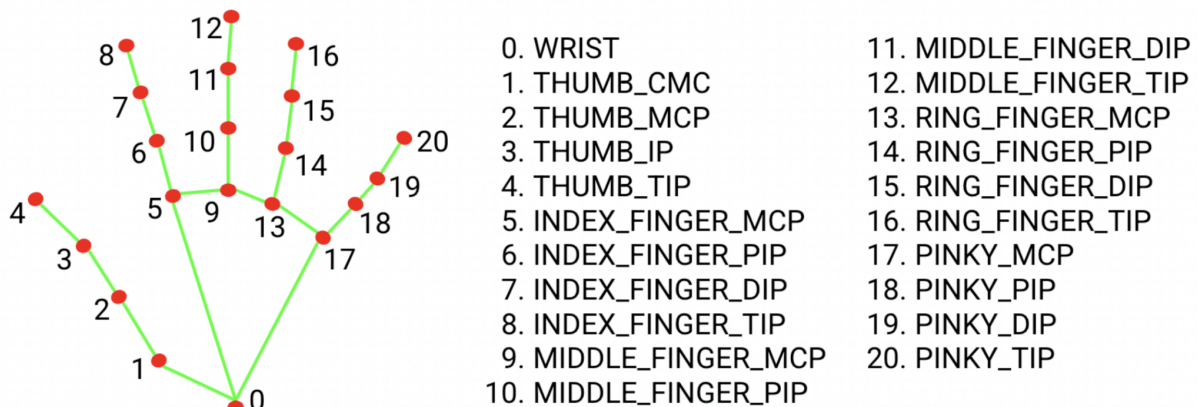


Figura 1. [Landmarks obtenidos \(MediaPipe\)](#)

Por ejemplo, para el cálculo del gesto “Pulgar hacia arriba” se tuvo en cuenta que el punto 4 estuviera por encima del punto 3 en el eje Y, y a su vez que no estuvieran muy alejados en el eje X, dando a entender que el pulgar estaba hacia arriba. También se tuvo que tener en cuenta que hubiera una distancia mínima entre el punto 4 y el 8, para no dar errores con otros gestos.

Para el gesto “Puño cerrado” se calculó la distancia entre el punto 4 y el 20 fuera menor que un escalar arbitrario. A su vez, esta comprobación se usará para el siguiente gesto, ya que si pulgar y meñique están juntos pero el índice (8) y el corazón (12) están en alto, será el “Gesto de paz (V)”, y si a esto se le suma también el anular (16), tendríamos otro gesto que hemos implementado con los 3 dedos centrales hacia arriba.

Por último, el gesto en el que se juntan todos los dedos de una mano y se apunta hacia arriba (en inglés, Pinch) se calcula teniendo en cuenta que la punta de todos los dedos (4, 8, 12, 16, 20) estén cerca.

Para el correcto reconocimiento del gesto que se quiere usar, se tiene en cuenta que el usuario mantenga al menos 2 segundos fijo el gesto, para así entender que realmente quiere hacer ese cambio.

3. Por último, una vez se reconocían todos los gestos y se diferenciaba entre mano derecha e izquierda, se creó un marco vacío, de las mismas dimensiones que la cámara. En este marco, se pintaría o borraría donde el landmark 8 de la mano derecha (la punta del dedo índice) apunte. Este efecto, varía de color o borra dependiendo del gesto realizado en la mano izquierda (“Pulgar hacia arriba” modo color rojo; “Puño cerrado” modo color verde; “Gesto de paz (V)” modo color azul;

“Pinch” modo borrar; siendo el color verde el predeterminado). También, se puede cambiar entre un modo spray que pinta una serie de píxeles aleatorios en el radio de acción, o el modo pincel que pinta completamente el radio donde se aplica, estos modos se van intercalando con el gesto de los 3 dedos hacia arriba. Este marco se suma a la entrada de cámara para crear la ilusión de que se pinta sobre la propia cámara.

4. Fuentes y tecnologías utilizadas

Se ha utilizado el idioma de programación Python para la realización de este proyecto. También, se han utilizado las librerías math, numpy, time, OpenCV y mediapipe a la vez que su [documentación](#).

5. Conclusiones y propuestas de ampliación

El proyecto funciona según las expectativas, tiene algún error de detección de las propias manos mediante MediaPipe, que por ejemplo a veces te detecta la mano izquierda como derecha haciendo que pintes cuando intentas hacer un gesto, esto se puede deber a la iluminación, a la distancia de la mano o al propio movimiento de la mano, que al estar moviéndose durante un instante puede ser detectada como la otra mano. También existe algún error con los gestos, al haber gestos parecidos se puede llegar a detectar uno que no se tiene intención de hacer, pero en general una vez que se realiza el gesto y teniendo el margen de 2 segundos para fijar el gesto, se logra entender siempre lo que se quiere interpretar.

Para mejorar el proyecto se nos ocurren varias ideas, tales como añadir más gestos que permitieran cambiar a más funcionalidades, y que estas funcionalidades fueran por ejemplo pintar con más de un dedo, borrar todo el dibujo que se ha hecho, intentar hacer una función de relleno que permita si el usuario logra hacer una figura cerrada pintar todo el interior de la figura cerrada, en general añadir funciones que permitan dibujar y pintar más libremente.

En conclusión, estamos bastantes contentos con los resultados, puesto que no creíamos que el reconocimiento de los gestos pudiera llegar a ser tan funcional como es, sin haber entrenado un modelo.

6. Indicación de herramientas/tecnologías con las que les hubiera gustado contar

Una de las tecnologías que hubiera estado curiosa, hubiera sido extender el proyecto al campo de la Realidad Virtual/Realidad Aumentada. Dado que se basa en gestos y pintar, aplicarlo en una tercera dimensión, permitiendo solo con las manos pintar en un ámbito 3D podría haber sido bastante interesante.

Obviamente, es una aplicación totalmente distinta a lo ya realizado, pero de resto no creemos que haya una tecnología que podría haber ayudado en el desarrollo del proyecto.

7. Créditos materiales no originales del grupo

Como dicho anteriormente, para la realización de este proyecto se utilizaron las librerías: MediaPipe, a su vez como [guías de su uso](#) y [documentación](#).

A su vez, también se utilizaron la librería OpenCV-python (cv2) para facilitar el trabajo con vídeo e imágenes, y las librerías time, math y numpy que fueron necesarias para alguna tarea específica.