



DOCUMENTACIÓN: MEDIAPIPE_OPENC_BEGINNERS BY JACOB TINOCO

Reconocimiento de dedos extendidos usando:
Mediapipe.

Índice

Introducción.....	3
Requisitos Técnicos.....	4
1. Herramientas de Software	4
2. Bibliotecas y Dependencias.....	4
3. Requisitos de Hardware.....	4
4. Clonación del Repositorio	4
5. Ejecución del Proyecto	5
Fundamentos Teóricos	6
1. MediaPipe.....	6
2. Módulo de Detección de Manos (MediaPipe Hands).....	6
3. LandMarks (Puntos de Referencia)	6
4. Extensión de Dedos	7
5. Confianza de Detección.....	7
6. OpenCV	7
7. NumPy	7
Metodología.....	8
Implementación	10
Resultados.....	13
Análisis de Precisión, Limitaciones y Desafíos.....	13
Precisión	13
Limitaciones	13
Desafíos	14
Análisis de los Resultados	15
Conclusión	16

Indice de Figuras

FIGURA 1 LANDMARKS DE MEDIAPIPE, IMAGEN EXTRAÍDA DE MEDIUM.....	6
FIGURA 2 METODOLOGÍA DEL CÓDIGO	9

Introducción.

La **visión por computadora** es un campo de la inteligencia artificial que permite a las máquinas interpretar y entender el mundo visual a través de imágenes o videos. Esta tecnología se utiliza en una amplia variedad de aplicaciones, como reconocimiento facial, detección de objetos, navegación autónoma y, en este caso, la detección de gestos y movimientos de manos.

MediaPipe es un framework de código abierto desarrollado por Google que facilita la implementación de pipelines de percepción multimodal, es decir, sistemas que pueden procesar múltiples tipos de datos (como imágenes, audio y video). MediaPipe ofrece soluciones preentrenadas para tareas como la detección de manos, seguimiento facial, reconocimiento de objetos y más. En este proyecto, se utiliza el módulo **MediaPipe Hands** para detectar y analizar la posición de los dedos en tiempo real.

El objetivo de este informe es explicar cómo se utilizó MediaPipe para desarrollar una aplicación que detecta la extensión de dedos en tiempo real. A través de este proyecto, se busca demostrar cómo la combinación de herramientas como MediaPipe y OpenCV puede facilitar la creación de sistemas de visión por computadora accesibles y eficientes.

MediaPipe Hands utiliza un modelo de aprendizaje automático para detectar **LandMarks** (puntos de referencia) en la mano. Estos LandMarks corresponden a las articulaciones y puntas de los dedos, lo que permite determinar si un dedo está extendido o no. En este proyecto, se implementó un sistema que:

1. Captura video en tiempo real desde una cámara web.
2. Detecta las manos y sus LandMarks utilizando MediaPipe.
3. Calcula la extensión de los dedos basándose en la posición relativa de los LandMarks.
4. Muestra en pantalla el estado de los dedos (extendido o no) y el nivel de confianza de la detección.

Documentación Oficial:

Para más información sobre MediaPipe y su módulo de detección de manos, se puede consultar la documentación oficial:

- [MediaPipe Hands](#)
- [MediaPipe Documentation](#)

Requisitos Técnicos

Para ejecutar este proyecto, es necesario contar con las siguientes herramientas y dependencias instaladas en el sistema:

1. Herramientas de Software

- **Python 3.7 o superior:** El lenguaje de programación utilizado para desarrollar el proyecto.
- **Gestor de paquetes pip:** Para instalar las bibliotecas necesarias.

2. Bibliotecas y Dependencias

Las siguientes bibliotecas deben estar instaladas en el entorno de Python:

- **OpenCV:** Para la captura y procesamiento de imágenes en tiempo real.
- **MediaPipe:** Para la detección de manos y LandMarks.
- **NumPy:** Para operaciones numéricas y de matrices.

Estas dependencias se pueden instalar fácilmente utilizando el archivo requirements.txt incluido en el repositorio. Para instalarlas, ejecuta el siguiente comando en la terminal:

```
pip install -r requirements.txt
```

3. Requisitos de Hardware

- **Cámara web:** Para capturar video en tiempo real.
- **Sistema operativo:** Windows, macOS o Linux.

4. Clonación del Repositorio

Para obtener el código del proyecto, clona el repositorio desde GitHub:

```
git clone <https://github.com/Jacob-Tinoco/Mediapipe_OpenC_Beginners.git>
cd Mediapipe_OpenC_Beginners
git clone <https://github.com/Jacob-Tinoco/Mediapipe_OpenC_Beginners.git>
cd Mediapipe_OpenC_Beginners
```

5. Ejecución del Proyecto

Una vez instaladas las dependencias, puedes ejecutar cualquiera de los scripts proporcionados:

- **Versión 1.0.1:**

```
python fingers_extentionV1.0.1.py
```

- **Versión 1.0.2:**

```
python fingers_extentionV1.0.2.py
```

- **Versión 1.2.0:**

```
python fingers_extentionV1.2.0.py
```

Fundamentos Teóricos

1. MediaPipe

MediaPipe es un framework de código abierto desarrollado por Google que permite crear pipelines de percepción multimodal. Esto significa que puede procesar múltiples tipos de datos, como imágenes, audio y video, para tareas como detección de objetos, seguimiento facial y reconocimiento de gestos. Es altamente eficiente y está diseñado para funcionar en tiempo real.

2. Módulo de Detección de Manos (MediaPipe Hands)

MediaPipe Hands es un módulo específico de MediaPipe que se utiliza para detectar y rastrear manos en imágenes o videos. Utiliza un modelo de aprendizaje automático preentrenado para identificar **LandMarks** (puntos de referencia) en la mano, como las articulaciones y las puntas de los dedos.

3. LandMarks (Puntos de Referencia)

Los LandMarks son puntos clave identificados en la mano que representan las articulaciones y las puntas de los dedos. MediaPipe Hands detecta 21 LandMarks por mano, que incluyen:

- **Pulgar:** 4 LandMarks (desde la base hasta la punta).
- **Índice, Medio, Anular y Meñique:** 4 LandMarks por dedo (desde la base hasta la punta).
- **Muñeca:** 1 LandMarks.

Estos LandMarks permiten analizar la posición y orientación de la mano, así como la extensión de los dedos, en la Figura 1 se muestra el LandMarks de Mediapipe.

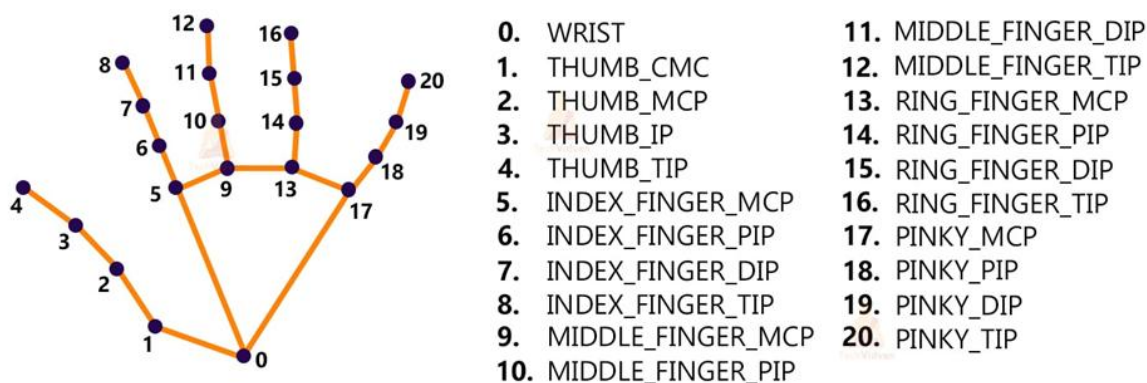


Figura 1 LandMarks de Mediapipe, imagen extraída de [Medium](#).

4. Extensión de Dedos

La extensión de dedos se determina comparando la posición relativa de los LandMarks. Por ejemplo:

- Para los dedos **índice, medio, anular y meñique**, se verifica si la punta del dedo (LandMark superior) está por encima de la articulación (LandMark inferior).
- Para el **pulgar**, se utiliza la coordenada horizontal (eje X) para determinar si está extendido, ya que su movimiento es diferente al de los otros dedos.

5. Confianza de Detección

La confianza de detección es un valor entre 0 y 1 (0% y 100%) que indica la precisión con la que el modelo ha identificado un dedo como extendido. En este proyecto, se utiliza un umbral de confianza (89% para los dedos y 20% para el pulgar) para determinar si un dedo está realmente extendido.

6. OpenCV

OpenCV (Open Source Computer Vision Library) es una biblioteca de código abierto que proporciona herramientas para el procesamiento de imágenes y videos. En este proyecto, se utiliza para:

- Capturar video en tiempo real desde la cámara web.
- Mostrar los resultados de la detección de dedos en una ventana.

7. NumPy

NumPy es una biblioteca de Python utilizada para realizar operaciones numéricas y de matrices. En este proyecto, se emplea para calcular distancias entre LandMarks y normalizar valores de confianza.

Metodología.

Sobre la metodología se muestra en la Figura 2, el cual sigue el proceso de:

la captura de video.

- El proceso termina o el proceso comienza.
- OpenCV captura frames de la cámara web.
- Cada frame se convierte de BGR a RGB para ser procesado por MediaPipe.
- MediaPipe Hands detecta las manos y sus LandMarks.
- Se verifica la posición de los LandMarks para determinar si los dedos están extendidos.
- Se calcula el nivel de confianza para cada dedo extendido.
- Los resultados se muestran en una ventana de OpenCV.
- Condición de salida. Si el usuario presiona ESC, el proceso termina; de lo contrario, se repite desde.

Se muestra a continuación la Figura 2.

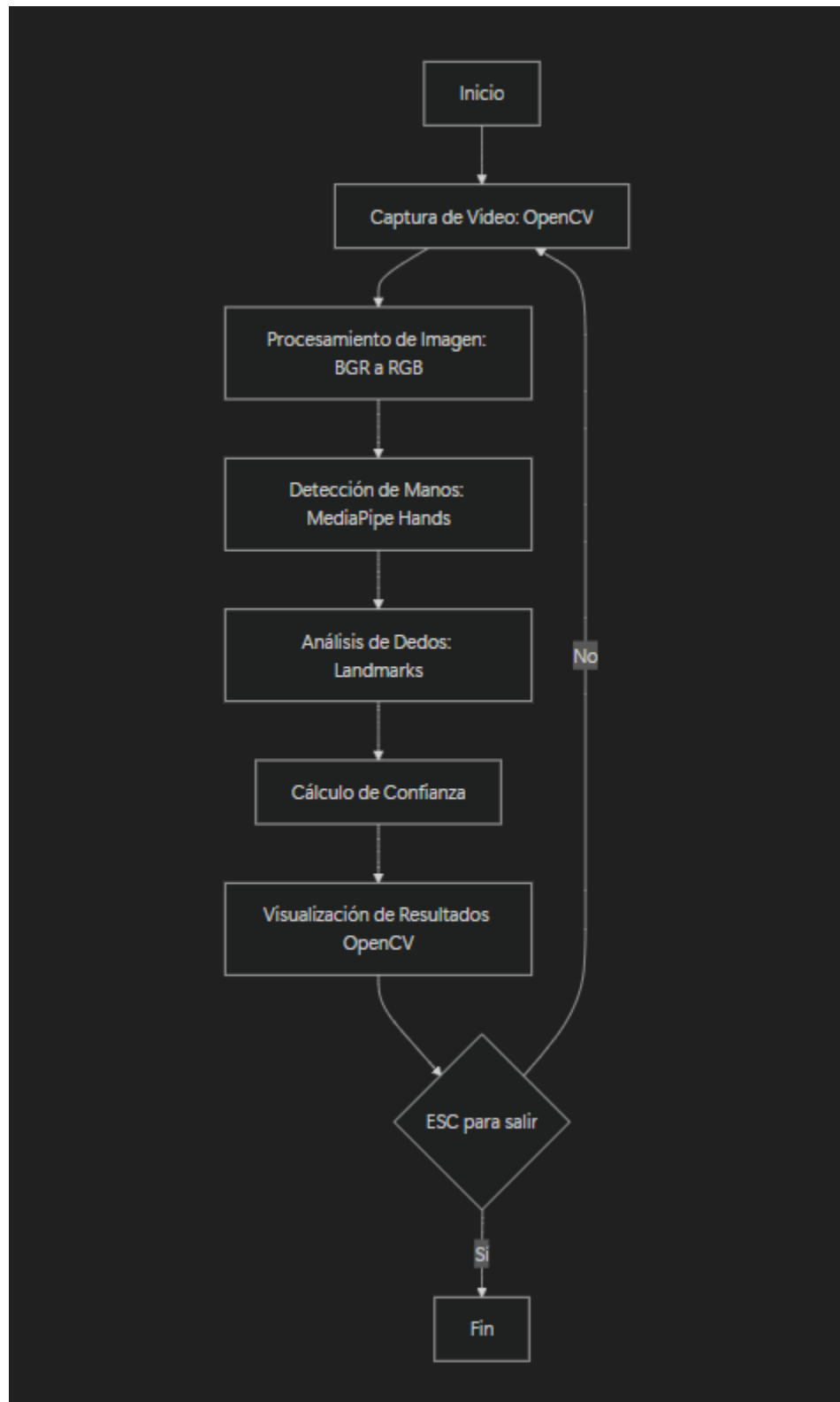


Figura 2 Metodología del Código

Implementación

Pseudocódigo: Detección de Dedos Extendidos con MediaPipe

El siguiente pseudocódigo describe la estructura general del proyecto para la detección de dedos extendidos utilizando MediaPipe y OpenCV. El proceso se divide en tres etapas principales: **inicialización**, **procesamiento de frames** y **visualización de resultados**.

Inicialización

1. **Importar bibliotecas:** Se importan las bibliotecas necesarias, como cv2 (OpenCV) para la captura de video y Mediapipe para la detección de manos.
2. **Configurar MediaPipe Hands:** Se inicializa el módulo de detección de manos de MediaPipe con parámetros como el número máximo de manos a detectar y los umbrales de confianza.
3. **Iniciar captura de video:** Se abre la cámara web utilizando OpenCV para capturar frames en tiempo real.

Procesamiento de Frames

1. **Capturar frame:** Se obtiene un frame de la cámara web.
2. **Convertir a RGB:** El frame se convierte de BGR (formato de OpenCV) a RGB (formato requerido por MediaPipe).
3. **Detectar manos:** MediaPipe procesa el frame y devuelve los LandMarks (puntos de referencia) de las manos detectadas.
4. **Analizar dedos:** Para cada mano detectada:
 - Se identifican los LandMarks correspondientes a las puntas y articulaciones de los dedos.
 - Se determina si cada dedo está extendido comparando las coordenadas de los LandMarks.
 - Se calcula el nivel de confianza basado en la distancia entre los LandMarks.
5. **Filtrar resultados:** Solo se consideran válidos los dedos con un nivel de confianza superior al umbral definido (89% para los dedos y 20% para el pulgar).

Visualización de Resultados

1. **Dibujar LandMarks:** Se dibujan los LandMarks y las conexiones entre ellos en el frame utilizando OpenCV.

2. **Mostrar estado de dedos:** Se muestra en pantalla el estado de los dedos (extendido o no) y el nivel de confianza para cada uno.
3. **Mostrar frame:** El frame procesado se muestra en una ventana de OpenCV.
4. **Condición de salida:** Si el usuario presiona la tecla ESC, el proceso termina; de lo contrario, se repite desde la captura de frame.

Finalización

1. **Liberar recursos:** Se libera la cámara y se cierran todas las ventanas de OpenCV.

Código de Ejemplo (Pseudocódigo)

Este pseudocódigo describe un bucle principal que captura frames de la cámara, detecta manos y dedos extendidos utilizando MediaPipe, y muestra los resultados en tiempo real con OpenCV. La lógica se basa en la comparación de landmarks y el cálculo de confianza para determinar el estado de los dedos. El proceso es eficiente y está diseñado para ejecutarse en tiempo real.

```
# Inicialización
importar cv2, mediapipe

configurar MediaPipe Hands
iniciar cámara web

mientras cámara esté abierta:
    # Procesamiento de Frames
    capturar frame
    convertir frame a RGB
    detectar manos con MediaPipe

    para cada mano detectada:
        identificar landmarks de dedos
        para cada dedo:
            determinar si está extendido
            calcular confianza
            si confianza > umbral:
                marcar dedo como extendido

    # Visualización de Resultados
    dibujar landmarks en frame
    mostrar estado de dedos y confianza
    mostrar frame en ventana

    si se presiona ESC:
        terminar proceso

# Finalización
liberar cámara
cerrar ventanascerrar ventanas
```

Resultados

Al ejecutar el script, se obtiene una ventana en tiempo real que muestra la captura de video de la cámara web, con anotaciones superpuestas que indican el estado de los dedos (extendido o no con ceros y unos) y el nivel de confianza para cada uno (en escala del 0% al 100%). El sistema es capaz de detectar ambas y, para cada una, identifica los dedos extendidos con un nivel de confianza que varía según la posición relativa de los LandMarks. Los resultados se muestran en la parte superior de la pantalla, indicando el estado de los dedos (1 para extendido, 0 para no extendido), y en la parte inferior, se muestra el porcentaje de confianza para cada dedo. El pulgar, aunque más difícil de detectar debido a su movimiento lateral, se identifica con un umbral de confianza más bajo (20%), mientras que los otros dedos requieren un umbral del 89%.

La precisión del sistema es aceptable cuando los dedos están claramente extendidos y la mano está frente a la cámara. Sin embargo, existen limitaciones y desafíos. En algunos casos, se producen **falsos positivos**, especialmente cuando los dedos están parcialmente flexionados o cuando la mano está en una posición que dificulta la detección de los LandMarks frente a la cámara. El pulgar, en particular, es más propenso a errores debido a su movimiento único y a la variabilidad en su posición. Además, la iluminación y la resolución de la cámara pueden afectar la calidad de la detección. A pesar de estas limitaciones, el sistema funciona bien en la mayoría de los escenarios, proporcionando una base sólida para aplicaciones de interacción basadas en gestos.

Análisis de Precisión, Limitaciones y Desafíos

Precisión

- **Dedos 1 al 4 (índice, medio, anular, meñique):** La precisión es alta (alrededor del 90%) cuando los dedos están claramente extendidos y la mano está bien posicionada.
- **Pulgar:** La precisión es menor (alrededor del 70-80%) debido a su movimiento lateral y a la dificultad para distinguir entre extensiones y flexiones parciales.

Limitaciones

- **Falsos positivos:** Ocurren cuando los dedos están parcialmente flexionados o cuando la mano está en una posición que dificulta la detección de landmarks.
- **Detección del pulgar:** Es menos precisa debido a su movimiento único natural y a la variabilidad en su posición.
- **Condiciones de iluminación:** La detección puede verse afectada por una iluminación deficiente o sombras.

- **Resolución de la cámara:** Una resolución baja puede reducir la precisión de la detección de LandMarks.

Desafíos

- **Mejorar la detección del pulgar:** Se requiere un ajuste más fino en el cálculo de confianza para mejorar la precisión.
- **Reducción de falsos positivos:** Se podrían implementar técnicas adicionales, como el uso de un modelo más avanzado o la incorporación de más LandMarks.
- **Robustez en diferentes condiciones:** Mejorar la detección en condiciones de iluminación variable y con cámaras de baja resolución.

Análisis de los Resultados

El proyecto se desarrolló en tres versiones principales, cada una con enfoques y mejoras distintas. La **Versión 1.0.1** fue efectiva para detectar la extensión de los dedos índice, medio, anular y meñique, mostrando un nivel de confianza preciso en la mayoría de los casos. Sin embargo, no lograba detectar correctamente el pulgar, lo que limitaba su funcionalidad. La **Versión 1.0.2** se centró en mejorar la detección del pulgar, utilizando un enfoque basado en coordenadas horizontales y un umbral de confianza más bajo (20%). Aunque mejoró la detección del pulgar, no integraba la funcionalidad completa de los otros dedos, lo que la hacía menos versátil. La **Versión 1.2.0** combinó las mejoras de las versiones anteriores, detectando todos los dedos, incluyendo el pulgar, y mostrando el estado de los dedos y el nivel de confianza en pantalla. Además, introdujo la verificación de la distancia entre la punta del dedo y la muñeca para reducir falsos positivos. A pesar de estos avances, aún existen desafíos, como la detección del pulgar en posiciones complejas y la sensibilidad a condiciones de iluminación deficiente. Para futuras versiones, se podrían implementar nuevas técnicas, como el uso de un modelo que tenga en cuenta la rotación del pulgar o la incorporación de filtros para reducir falsos positivos. También se podría explorar la integración del sistema en aplicaciones prácticas, como el control de dispositivos IoT o proyectos artísticos.

Conclusión

El proyecto de detección de dedos extendidos utilizando MediaPipe ha demostrado ser una solución efectiva y accesible para aplicaciones de visión por computadora. A lo largo de las tres versiones desarrolladas, se lograron avances significativos en la precisión y funcionalidad del sistema. La **Versión 1.2.0** representa la convergencia de dos scripts para la detección de todos los dedos, incluyendo el pulgar, y mostrando resultados en tiempo real con un nivel de confianza de al menos el 85%. Uno de los principales logros fue superar el desafío inicial de detectar el pulgar, optimizando su reconocimiento mediante el uso de coordenadas horizontales y un umbral de confianza ajustado. Además, se redujeron los falsos positivos al incorporar la verificación de la distancia entre la punta del dedo y la muñeca. Mediapipe, ofrece una solución eficiente y fácil de implementar para la detección de manos y LandMarks. Su capacidad para funcionar en tiempo real y su integración con OpenCV lo convierten en una opción ideal para proyectos de visión por computadora, tanto en entornos educativos como en aplicaciones prácticas. Este proyecto de educación tiene un gran potencial para ser implementado en diversas aplicaciones, como el control de dispositivos IoT, la interacción en realidad aumentada, mezclar el arte visual y la visión por computadora o herramientas educativas para enseñar lenguaje de señas. En conclusión, este proyecto no solo ha logrado sus objetivos iniciales, sino que también ha sentado las bases para futuras investigaciones y aplicaciones en el campo de la visión por computadora y la interacción basada en gestos.