UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

SISTEMA INTERACTIVO DE SEGUIMIENTO DE POSICIÓN CORPORAL MEDIANTE VISIÓN ARTIFICIAL PARA ACTIVIDAD DEMOSTRATIVA.

POR

Ricardo Francisco Ríos Díaz

Informe de Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero Civil Biomédico

Profesores Guía
Esteban Pino Quiroga
Mabel Urrutia Martínez

Comisión
Pamela Guevara Alvez

Septiembre 2025
Concepción (Chile)

Agradecimientos

A mi familia, por darme su amor incondicional durante todo este viaje. Por las palabras de ánimo y cariño que fueron abrazos cálidos a la distancia.

A mis amigos, con los cuales comparto invaluables recuerdos que me llenan de alegría el alma. Fueron la energía y fortaleza en los días que más lo necesitaba. Fueron apoyo no solamente en el ámbito emocional, sino también en el ámbito académico y desarrollo personal. Al apoyarme cuando no existía claridad en el camino.

Agradezco también al Proyecto Fondecyt Exploración 13220040 por los recursos proporcionados y por dar la iniciativa de esta memoria de titulo; la idea nace originalmente de integrar modelos de visión artificial al servicio de la interpretación emocional en usuarios y niños.

Finalmente, a los profesores que generosamente compartieron su ayuda, guía y conocimientos, mi más profundo agradecimiento. Han sido verdaderos ejemplos a seguir y su apoyo ha sido invaluable.

Resumen

El presente informe de memoria de título expuso el desarrollo de un sistema de reconocimiento postural en tiempo real utilizando la visión por computador, enfocado en la comparación automática de posturas corporales humanas en una interfaz gráfica con asignación de puntaje. Una postura fue capturada en tiempo real y la otra fue de referencia proveniente de datasets internos del proyecto. El sistema fue implementado usando el lenguaje de programación Python y sustentado por el modelo MediaPipe Holistic de Google, el cual permitió la estimación de puntos articulares clave del cuerpo humano a partir de la imagen captada por la cámara, y sobre estos se dibujaron líneas de unión, simulando un esqueleto sobre la imagen. El modelo detectaba puntos en regiones faciales, corporales y en manos.

Se realizó una revisión bibliográfica y estudio de técnicas de procesamientos internos de la visión por computador basados en Deep Learning (DL) supervisado. Se seleccionó un modelo de estos mediante comparación de ventajas y desventajas entre los modelos de Computer Vision (CV) actuales.

Para verificar si las posturas eran similares o diferentes, fue necesario el desarrollo de una lógica comparativa postural que constaba de la comparación de diferencias promedio basadas en distancias euclidianas entre puntos articulares correspondientes y la asignación de pesos sobre regiones anatómicas con un umbral de tolerancia definido empíricamente. Si la postura del usuario captada por la cámara coincidía con la imagen de referencia, se otorgaban puntos como retroalimentación y se continuaba con la siguiente.

Se implementaron además tres dataset ordenados por dificultad para enriquecer la experiencia de la actividad demostrativa y se evaluó el logro de posturas por nivel de dificultad por parte de usuarios reales. Se expusieron muestras de funcionamiento de los software desarrollados, umbral definido empíricamente y evaluación de logro de posturas por nivel de dificultad.

En la actividad con usuarios, el porcentaje promedio de posturas logradas disminuye al aumentar la dificultad (80 % en el nivel fácil, 62 % en el intermedio y 40 % en el difícil). En términos generales, el proyecto logró demostrar la factibilidad de construir un sistema de reconocimiento postural en tiempo real, capaz de otorgar retroalimentación inmediata, establecer niveles de dificultad y cuantificar el desempeño del usuario de forma objetiva.

Abstract

The present thesis report presented the development of a real-time posture recognition system using computer vision, focused on the automatic comparison of human body postures within a graphical interface with score assignment. One posture was captured in real time and the other was taken as a reference from internal project datasets. The system was implemented using the Python programming language and supported by Google's MediaPipe Holistic model, which enabled the estimation of key joint points of the human body from the image captured by the camera, upon which connecting lines were drawn, simulating a skeleton over the image. The model detected points in facial, body, and hand regions.

A bibliographic review and study of internal processing techniques in computer vision based on supervised Deep Learning (DL) were carried out. One of these models was selected through a comparison of advantages and disadvantages among current Computer Vision (CV) models.

To verify whether the postures were similar or different, it was necessary to develop a comparative posture logic consisting of the comparison of average differences based on Euclidean distances between corresponding joint points and the assignment of weights on anatomical regions with a tolerance threshold defined empirically. If the user's posture captured by the camera matched the reference image, points were awarded as feedback and the system proceeded to the next posture.

In addition, three datasets ordered by difficulty were implemented to enrich the experience of the demonstrative activity, and the achievement of postures by difficulty level was evaluated with real users. Samples of the developed software's operation, empirically defined threshold, and evaluation of posture achievement by difficulty level were presented.

In the user activity, the average percentage of achieved postures decreased as the difficulty increased (80 % at the easy level, 62 % at the intermediate level, and 40 % at the difficult level). In general terms, the project demonstrated the feasibility of building a real-time posture recognition system, capable of providing immediate feedback, establishing difficulty levels, and objectively quantifying user performance.

Tabla de contenidos

Índice de Tablas

Índice de Figuras

Capítulo 1. Introducción

1.1 Introducción general

Las nuevas tecnologías como los modelos de inteligencia artificial, visión por computador y procesamiento de imágenes, nos permiten la adquisición, análisis, síntesis y visualización de imágenes de la vida real sobre el cuerpo humano, por ejemplo. Pudiendo clasificar: ejes, distancia, profundidad, entre otros parámetros para la captura de la imagen del sujeto en tiempo real.

Gracias a modelos de visión artificial se pueden capturar fotogramas de un usuario en tiempo real y aplicar sobre este distintos modelos por región anatómica. Es decir, se pueden aplicar modelos de reconocimiento postural, facial, de manos, entre otros.

En este contexto, el presente informe de memoria de título se enfoca en la implementación de un sistema interactivo de seguimiento de posición corporal, que hace uso de modelos de visión artificial para detectar, representar y comparar la postura del usuario en tiempo real. A través de una cámara web, el sistema captura la imagen del sujeto y genera un esqueleto virtual mediante el modelo MediaPipe Pose, que identifica las principales articulaciones del cuerpo humano. La imagen capturada puede ser comparada con una imagen de referencia, permitiendo determinar si la postura actual corresponde a la posición que se busca y si esta es correcta, retroalimentarla con un sistema de puntos para la actividad demostrativa.

A partir de este proyecto, además de la parte técnica, podemos desprender aplicaciones en los ámbitos de actividades demostrativas de visión artificial por cámara web, aplicaciones médicas en fisioterapia mediante el reconocimiento, corrección postural y movilidad del sujeto. Este análisis es posible gracias a la incorporación de revisión y comparación entre distintos modelos de visión artificial, incluyendo modelos como MediaPipe Holistic y otros modelos de cuerpo completo, evaluando su desempeño y adaptabilidad en el sistema.

1.2 Objetivos

1.2.1 Objetivo general

Implementar un sistema interactivo de seguimiento en línea de la posición corporal de un sujeto, mediante un modelo de visión artificial, que permita comparar posturas captadas en tiempo real con imágenes de referencia para actividad demostrativa de forma interactiva.

1.2.2 Objetivos específicos

- Desarrollar software que obtenga y visualice la posición en línea del usuario en tiempo real mediante modelos de visión artificial.
- Incorporar una lógica de comparación entre la postura del usuario y una imagen de referencia, con asignación de puntaje cuando se detecta coincidencia.
- Diseñar una interfaz gráfica que permita: visualizar el esqueleto generado, la imagen de referencia y la retroalimentación de puntaje para actividad demostrativa.
- Validar el funcionamiento de la solución mediante pruebas preliminares.

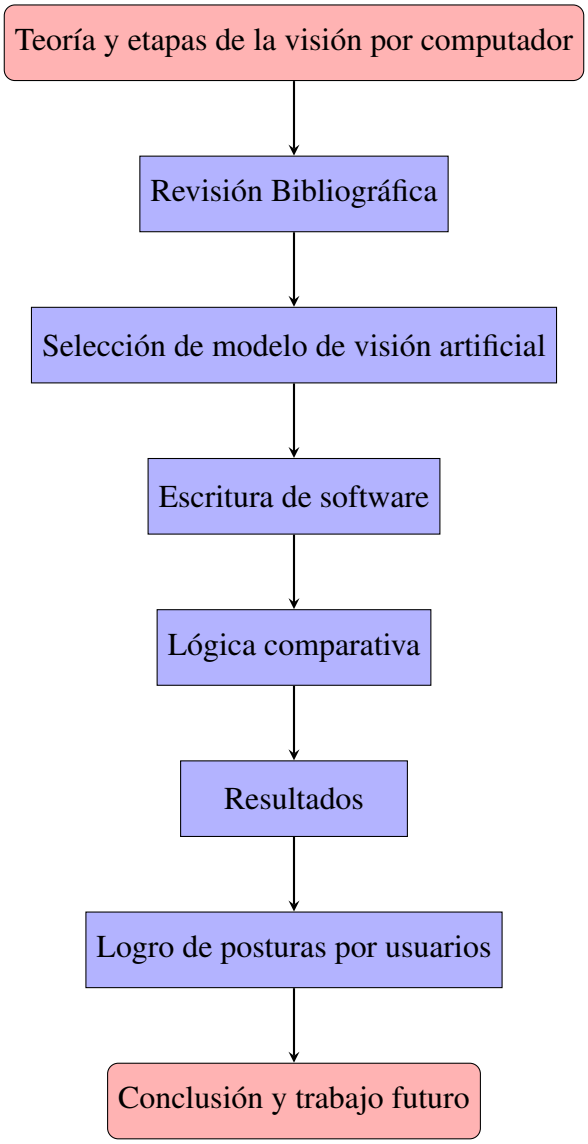
1.3 Alcance y limitaciones

- Se cuenta con una cámara web que deberá ser utilizada en el proyecto.
- El software estará limitado a las capacidades y recursos del computador disponible.
- Se requiere buena iluminación para un correcto reconocimiento artificial.

1.4 Metodología

La metodología desarrollada en este trabajo se estructuró de manera integral, iniciando por la teoría y etapas de la visión por computador; Un análisis bibliográfico asociado al estado del arte de modelos de visión artificial existentes, luego a modelo experimental, se probaron diferentes modelos de visión artificial, comparando sus ventajas y desventajas. Luego, tras la selección de modelo de visión artificial,

se implementó el software de captura, dibujo de cuerpo completo, se incorporó la captura de imágenes de referencia, la lógica de comparación y se diseñó una interfaz gráfica simple para mostrar la imagen objetivo y un sistema de puntaje asociado a la similitud detectada. Finalmente, se evaluó el logro de posturas por nivel de dificultad en usuarios.; Y se obtuvieron conclusiones a partir de estas.



Esquema de Metodología a seguir

1.5 Temario

- **Capítulo 1. Introducción**: Introducción de manera general indicando la metodología, objetivos y alcances del proyecto.
- **Capítulo 2. Marco Teórico**: En esta sección se tiene el marco teórico y la revisión bibliográfica, las cuales sientan las bases para entender las etapas de la visión por computador, modelos de CV existentes y sus características.
- **Capítulo 3. Desarrollo de software**: Se elige el modelo de visión por computador con el que trabaja el proyecto, se escriben los software requeridos para el funcionamiento completo. Se desarrolla una lógica comparativa postural, se añaden datasets para la actividad interactiva y se evalúa el logro de posturas por nivel de dificultad en usuarios reales.
- **Capítulo 4. Resultados**: Se exponen los principales resultados obtenidos por los software donde se aprecia el funcionamiento del modelo de CV y se observan los porcentajes de posturas logradas por los usuarios en la experiencia propuesta.
- **Capítulo 5. Conclusiones**: En esta sección se extrae información relevante a partir de los resultados, concluyendo la efectividad del sistema y posibles optimizaciones que deba experimentar este mismo en el apartado de trabajo futuro.

Capítulo 2. Marco Teórico

2.1 Introducción

El presente capítulo está enfocado en la revisión bibliográfica y el estado del arte de distintos modelos de visión artificial existentes. Para ello, se presentan de manera breve las principales etapas de la visión por computador, procesamientos internos que estos realizan y cómo éstas se integran en los modelos modernos basados en *Deep Learning* (DL). Luego, se describen los modelos más conocidos en la actualidad, sus características y aplicaciones, para finalmente compararlos y analizar cuál resulta más pertinente para implementar en el presente proyecto.

2.2 Introducción a la visión por computador

La visión por computador es una rama de la inteligencia artificial que permite a las máquinas "ver", interpretar y comprender el entorno que nos rodea. Utiliza algoritmos y modelos matemáticos para procesar imágenes o videos y extraer información relevante a partir de ellos [1]. Una vez que la información es extraída, el sistema puede tomar decisiones o ejecutar acciones según las características detectadas.

En este proyecto el foco se encuentra en técnicas modernas de visión por computador basadas en DL supervisado, donde las etapas clásicas de procesamiento se integran dentro de los modelos. El objetivo de esta disciplina es automatizar tareas que el ojo humano realiza de forma natural, como:

- Reconocimiento de objetos: por ejemplo, personas, partes del cuerpo o elementos del entorno [2].
- Seguimiento de movimiento en escenas dinámicas, aplicado en seguridad o análisis de comportamiento [3].
- Detección de anomalías, como las presentes en imágenes médicas, entre ellas las de endoscopía [4].
- Interpretación del entorno en aplicaciones robóticas, por ejemplo en los autos autónomos [5].

2.3 Etapas de la visión por computador

De manera clásica, la visión por computador puede dividirse en distintas etapas [6], las cuales permiten adquirir, transformar e interpretar información visual. Aunque en este proyecto no se implementaron manualmente estas etapas, es útil presentarlas brevemente para entender cómo evolucionaron hacia los modelos actuales de DL.

2.3.1 Adquisición de imágenes

Corresponde a la captura de datos mediante cámaras o sensores (RGB, infrarrojos, profundidad, entre otros). En este proyecto se utilizó una cámara web convencional como fuente de imágenes en tiempo real.

2.3.2 Preprocesamiento

Incluye operaciones como la conversión a escala de grises, la reducción de ruido o la normalización de tamaños. Estos pasos buscaban mejorar la calidad de la imagen para facilitar su análisis posterior. En los enfoques modernos basados en DL, este tipo de transformaciones se realizan de forma automática dentro del modelo, sin la necesidad de preprocesamiento manual.

2.3.3 Procesamiento e interpretación

El procesamiento se enfocaba tradicionalmente en operaciones como convoluciones y detección de bordes (Sobel, Canny, entre otros), fundamentales en la evolución hacia las redes neuronales convolucionales (CNN) [7]. Estas técnicas permitían extraer características locales de las imágenes, como texturas, contornos o gradientes.

La interpretación, por su lado, consiste en dar significado a la información extraída, como identificar posturas humanas o reconocer objetos en una escena. En este proyecto, estas funciones son realizadas de manera automática por el modelo seleccionado, que integra tanto la detección como la interpretación de los puntos articulares clave en un sujeto.

2.3.4 Actuador

Hace referencia a la respuesta del sistema tras la interpretación de la imagen. Puede ser física (mover un robot, accionar un dispositivo) o lógica (mostrar un resultado en pantalla, entregar retroalimentación al usuario). En el caso de este proyecto, el actuador corresponde a la retroalimentación visual y la asignación de puntaje dentro de la interfaz gráfica.

2.4 OpenCV

OpenCV (o Open Source Computer Vision Library) es una biblioteca de código abierto [8], [9] enfocada al procesamiento en imágenes y visión por computador. Primero fue desarrollada por Intel y se encuentra disponible para diversos lenguajes de programación, como: C++, Python y Java [10]. Esta biblioteca nos permite trabajar con imágenes, video en tiempo real y es ampliamente usada en:

- Reconocimiento facial y corporal.
- Detección de objetos y movimiento.
- Procesamiento de bordes, contornos y filtros.
- Aplicaciones de seguridad, biomédicas, fitness e industriales.

Para la aplicación del proyecto, OpenCV permite: Leer imágenes y videos en tiempo real desde cámara web. Aplicar filtros, redimensionamiento, operaciones de convolución internas, ajuste de brillo y contraste y es el puente entre la cámara web (recolección de frames) y el modelo de detección corporal seleccionado.

2.5 Tipos de modelo de visión artificial

Ya conociendo las etapas de trabajo de visión por computador, pasamos a conocer los modelos más conocidos el desarrollo del proyecto, que trabajan utilizando el soporte de OpenCV.

2.5.1 MediaPipe

MediaPipe es una biblioteca de Google que se especializa en visión por computador en tiempo real, se encuentra bastante optimizada para dispositivos móviles y computadores. Es muy eficiente al poder funcionar con muy pocos recursos de CPU en vivo y tiene una precisión del 99.2 % [11]. Esta biblioteca entrega estimaciones relativas en unidades aproximadamente métricas de objetos y personas en 3D con imágenes de entrada en 2D, gracias a su modelo de aprendizaje profundo entrenado sobre grandes datasets. Está diseñada para funcionar con imágenes, videos y cámaras web que no detecten profundidad directamente [12]. Entre sus modelos más conocidos se encuentran:

- Pose: Estimación de la pose corporal completa. Cuenta con la detección de 33 puntos para el cuerpo. Se destaca que al considerar una región 3D , el eje Z está en dirección de la cámara y el torso del cuerpo suele estar cerca de la coordenada (0, 0, 0). Más adelante, estos puntos (landmarks) serán altamente requeridos. Véase la figura 2.1.

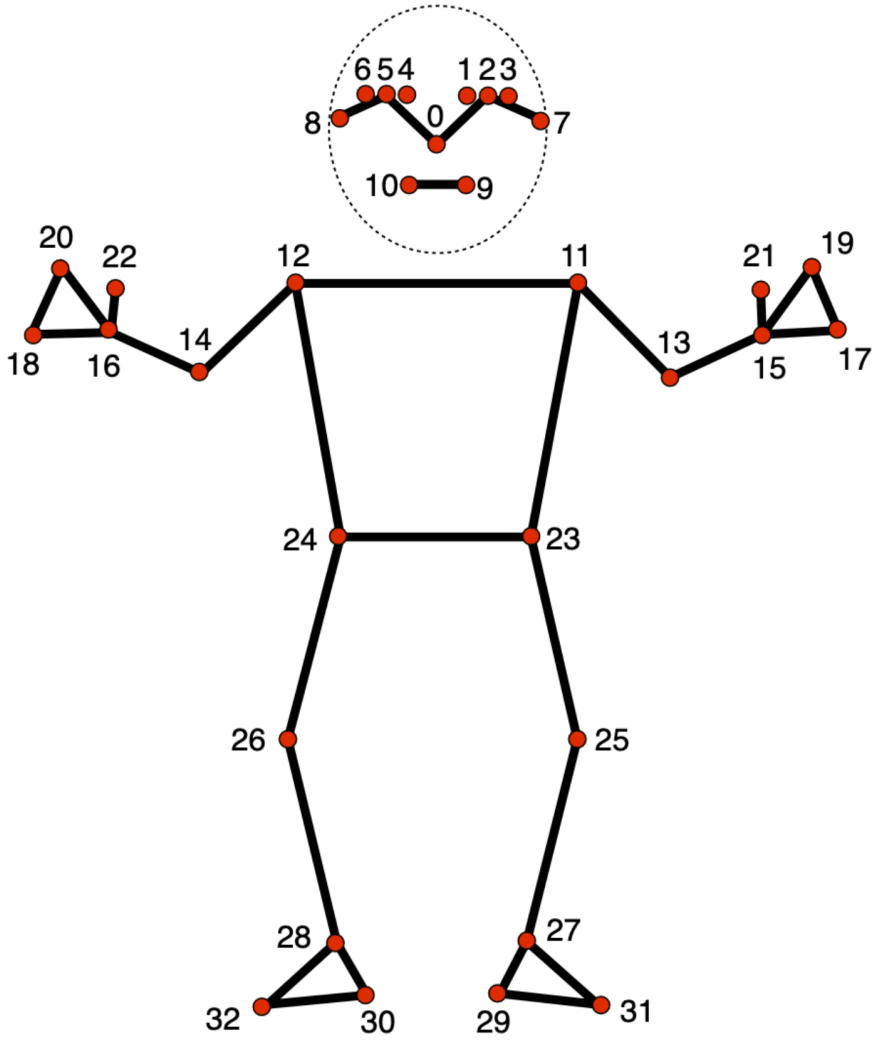


Figura 2.1: Landmarks del modelo Mediapipe Pose.

- Hands: Detección y seguimiento de manos. Cuenta con la detección de 21 puntos por cada mano [13]. Véase la figura 2.1 [14].

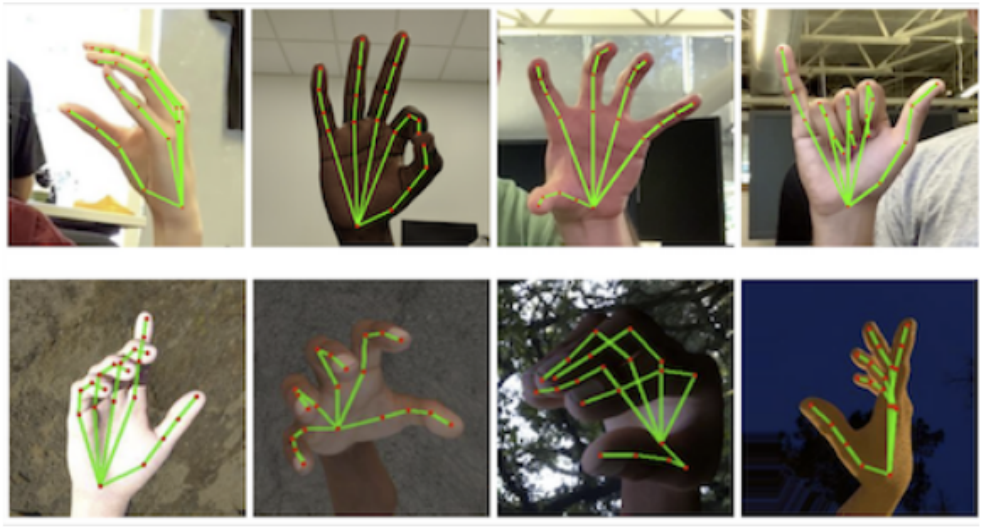


Figura 2.2: Ejemplos de funcionamiento del modelo Mediapipe Hands [14].

- Face Detection: Detección de rostros en un recuadro.
- Face Mesh: Malla facial en 3D. Cuenta con la detección de 468 puntos para el rostro, un modelo bastante robusto y detallado [15]. Véase la figura 2.3.

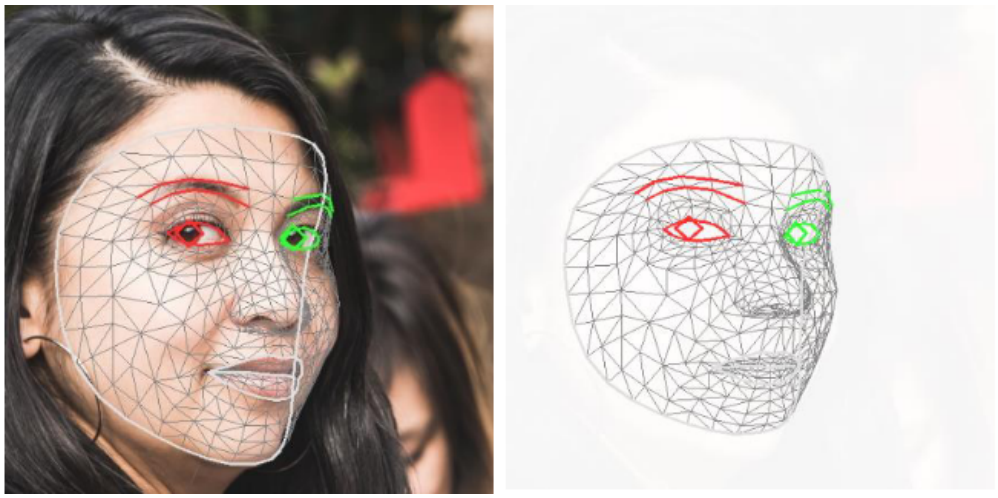


Figura 2.3: Ejemplo de funcionamiento del modelo Mediapipe Mesh [15].

- Holistic: Modelo muy completo, es la combinación de los modelos de estimación de pose, manos y rostro (Face Mesh) en un solo modelo integrado. Altamente eficiente y requiere de pocos recursos computacionales [16]. Opera secuencialmente en regiones de interés (ROI) para mantener bajo consumo de recursos. Funciona en base al modelo de estimación postural y recortes sobre manos y

cara reemplazándolos con sus modelos completos correspondientes, sus canales deben ser corregidos al ingresar una imagen, ya que OpenCV los carga en formato BGR y MediaPipe acepta RGB. Véase la figura 2.4.

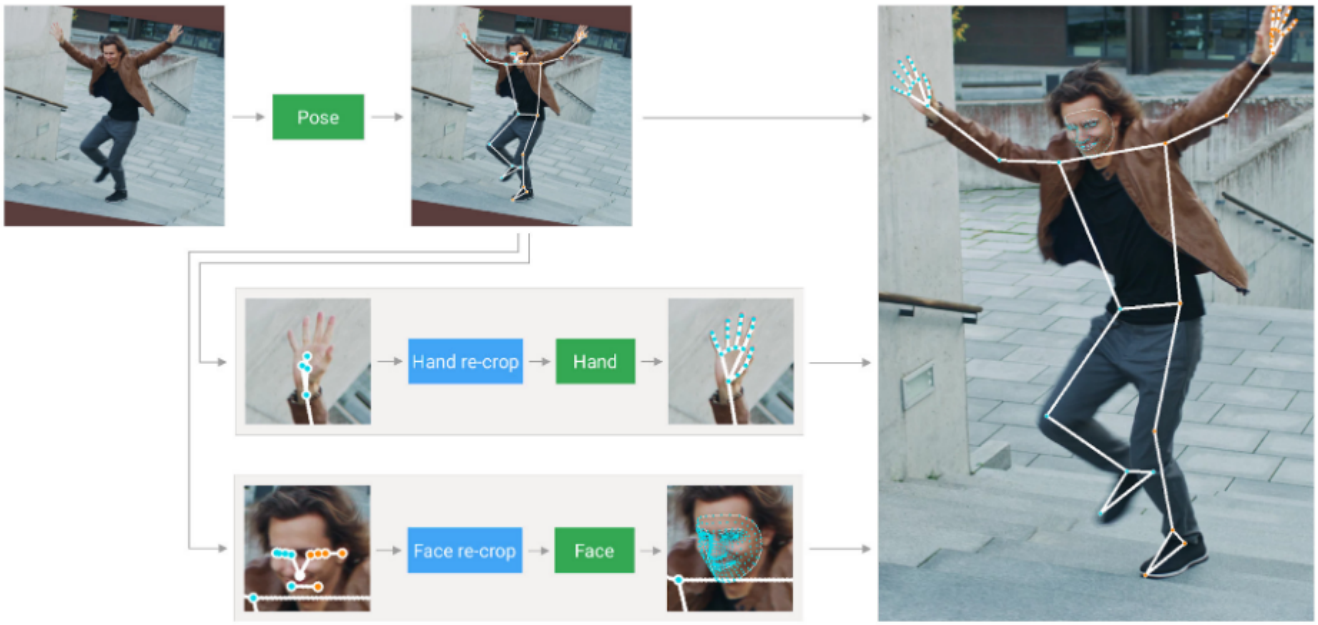


Figura 2.4: Descripción general del modelo Mediapipe Holistic [16].

- Objectron: Detección de objetos en 3D.
- Selfie Segmentation: Segmentación de fondo en selfies.

2.5.2 OpenPose

OpenPose fue desarrollado por la Universidad Carnegie Mellon, fue el primer sistema en tiempo real para estimación de objetos y poses en varias personas simultáneas [17]. Utiliza la arquitectura de Campos de Afinidad de Partes (*Part Affinity Fields*, PAFs) que vinculan los puntos articulares entre varias personas en una imágen, permitiendo detectar hasta 135 puntos (cuerpo, manos, rostro y pies) . Es un modelo de muy alta precisión, soporta imágenes 2D y 3D, manos y rostro, pero, requiere GPU para funcionar en tiempo real aunque es ideal cuando hay muchos sujetos en escena [18]. Véase las figuras 2.5,2.6 y 2.7 extraídas del paper oficial de OpenPose [17].

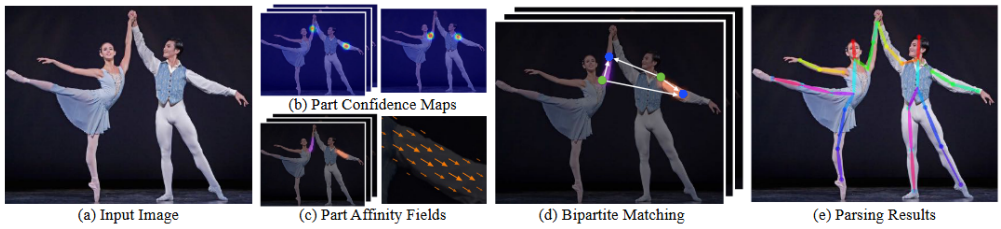


Figura 2.5: Uso de campos de afinidad de partes del modelo OpenPose [17].

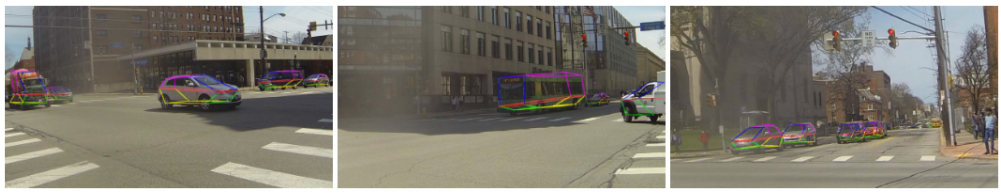


Figura 2.6: Ejemplo de detección de objetos usando el modelo OpenPose [17].

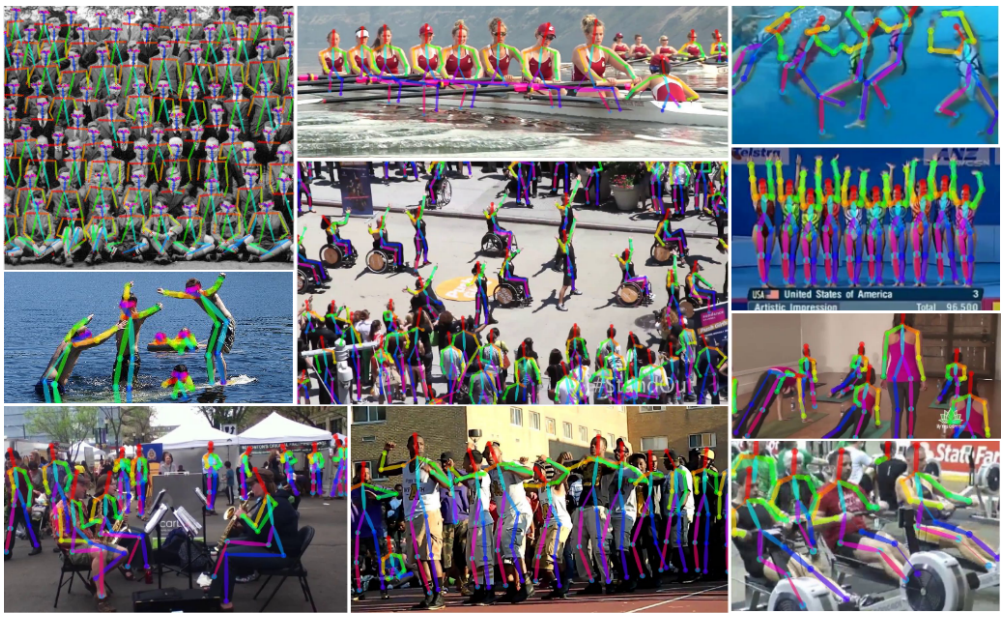


Figura 2.7: Ejemplo de detección de personas usando el modelo OpenPose [17].

2.5.3 BlazePose

Es un modelo de Google, bastante similar a Mediapipe (lo incluye) pero optimizado específicamente para dispositivos móviles, dado que el modelo es extremadamente ligero pero menos preciso, al ser un modelo no tan complejo. Tiene menor precisión en escenarios de movimiento (por ejemplo, en escenas

de ejercicio físico) y al igual que MediaPipe Pose, posee 33 puntos de detección en el cuerpo humano [19] con una frecuencia superior a 30 FPS en dispositivos como Pixel 2. Su arquitectura está basada en mapas de calor (probabilidades de encontrar puntos articulares claves en zonas de la imagen detectada) y en regresión directa a coordenadas (transformar esas distribuciones de probabilidades a coordenadas fijas) Véase la figura 2.8 [20].

Figura 2.8: Ejemplo de funcionamiento del modelo BlazePose [20].

2.5.4 DeeplabCut

Es una herramienta basada en deep learning diseñada para etiquetar los puntos anatómicos de especies sin usar marcadores [21]. No funciona en tiempo real y debe ser alimentada con imágenes del usuario; Sin embargo, es muy utilizada para investigación científica relacionada a la biomecánica, neurociencia, movimiento de desplazamiento en animales y estudios de laboratorio [22]. Véase la figura 2.9.

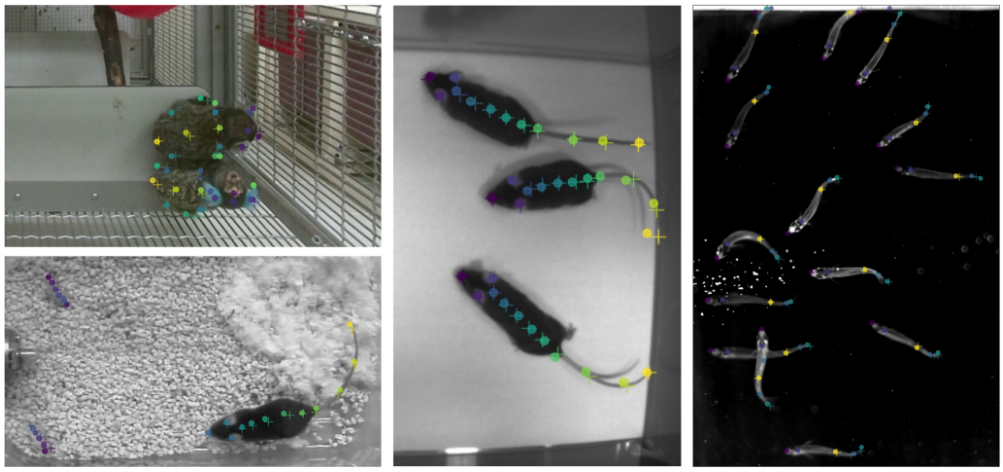


Figura 2.9: Ejemplo de funcionamiento del modelo DeeplabCut [22].

2.5.5 Haarcascade

Haarcascade es un método clásico de detección de objetos y rostros basado en clasificadores en cascada. Fue popularizado por Paul Viola y Michael Jones [23]. Se utilizan métodos como Fisherfaces, Eigenfaces o LBPH, que analizan patrones para compararlos con una base de datos [24] [25]. Este modelo requiere muchos recursos computacionales, debe ser utilizado convertido a escala de grises, posee alta tasa de falsos positivos y tiene menor precisión en comparación a los métodos modernos, sin embargo, tiene la ventaja de detectar varios rostros simultáneos a la vez. Véase la figura 2.10.

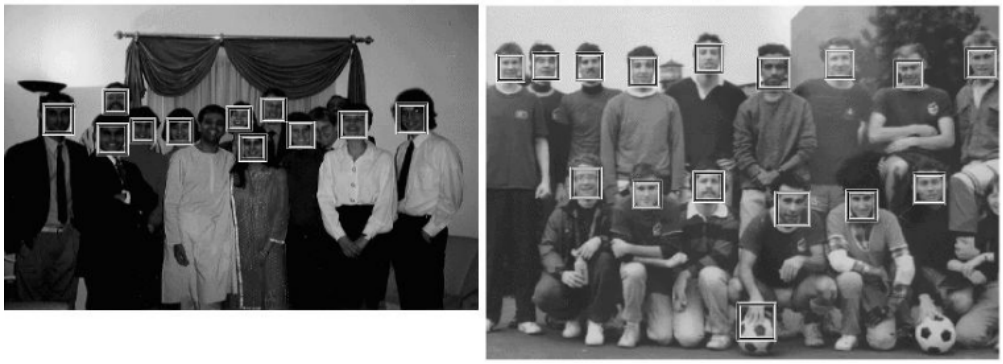


Figura 2.10: Ejemplo de funcionamiento del modelo Haarcascade [25].

2.6 Comparativa entre modelos

Se realiza una tabla comparativa para ver las ventajas y desventajas de cada modelo descrito, en la tabla 2-1.

Tabla 2-1: Comparativa entre modelos

Modelo	**Precisión**	**Tiempo real**	**Req. Comp.**	**Aplicación**	**Desventaja**	**Arquitectura**	**Multi-persona**
MediaPipe	Muy alta, 99.2 %	Sí	Bajo	Móviles, PC, fitness	Sensible a la iluminación y a fondos complejos	DL, estimación espacial	No
OpenPose	Alta	Sí (GPU)	Alto	Investigación, apps multi-persona	Alto consumo de GPU y tiempo de procesamiento	PAFs	Sí
BlazePose	Moderada	Sí	Muy bajo	Dispositivos portátiles y salud	Baja precisión en escenas con movimiento rápido o baja luz	Mapas de calor	No
DeepLabCut	Muy alta	No	Alto	Biomecánica, animales, investigación	No apto para uso en tiempo real y requiere entrenamiento previo	DL	No
Haarcascade	Moderada	Sí	Alto	Detección facial básica	Alta tasa de falsos positivos y sensibilidad a cambios en la imagen	Clasificadores en cascada	Sí (rostros)

2.7 Discusión

Este capítulo sienta las bases teóricas y bibliográficas de la visión por computador, el cómo trabaja a grandes rasgos, las etapas que experimenta este en el procesamiento de imágenes, aplicaciones que se pueden extraer, modelos más conocidos actualmente y una tabla comparativa de estos; Donde se exponen sus ventajas y desventajas a la hora de elegir uno de ellos. Con esto en mente, en el próximo capítulo se toma una decisión sobre cuál modelo es el adecuado para trabajar en este proyecto considerando precisiones y recursos computacionales para el correcto funcionamiento.

Capítulo 3. Desarrollo de software

3.1 Introducción

En este capítulo se selecciona el modelo de visión por computador a trabajar según los vistos en el capítulo anterior, siendo elegido por sus características y ventajas principales en comparación con el resto de modelos. Una vez que el modelo es elegido, se desarrolla el software de reconocimiento postural, la actividad interactiva mediante un procesamiento postural a imágenes de un pequeño dataset y se desarrolla la lógica comparativa para que ambas imágenes adquiridas puedan ser comparadas. Se añaden tres dataset ordenados por dificultad y se evalúa el cumplimiento de posturas en usuarios reales. Todos estos factores contribuyen al funcionamiento integral del software desarrollado.

3.2 Selección de modelo de visión por computador

Se selecciona el modelo de MediaPipe Holistic por ser un robusto, liviano y preciso modelo de visión por computador con más de 540 puntos de referencia (33 puntos para el cuerpo, 468 puntos para el rostro y 21 puntos por mano). Este modelo integra los landmarks models anteriores de MediaPipe en unidades aproximadamente métricas en uno solo y no requiere de una cámara que detecte profundidad, admite imágenes de entrada 2D. Este modelo además admite una amplia gama de configuraciones [26], [27] propias tales como:

- static image mode: Determina si el modelo trata cada imagen como independiente o como parte de un flujo de video.
- model complexity: Controla el tamaño y precisión del modelo de cuerpo (pose).
- smooth landmarks: Suaviza los landmarks (puntos articulares) a lo largo del tiempo para evitar temblores visuales.
- min detection confidence: Umbral mínimo de confianza para aceptar una detección inicial del cuerpo.
- min tracking confidence: Umbral mínimo para mantener el seguimiento de una detección ya realizada.

3.3 Escritura de software de reconocimiento de posturas por webcam

Se trabaja la escritura de software con el lenguaje de programación Python en su versión 3.12 gracias a su versatilidad, librerías soportadas y al ser un lenguaje informático de alto nivel. Se decide usar el programa Pycharm como soporte de escritura para todos los software desarrollados en este proyecto, gracias a su alta compatibilidad y rendimiento, con entorno virtual del propio sistema operativo y eficiencia que este programa ofrece. Para la escritura y correcto funcionamiento del modelo MediaPipe Holistic fueron requeridas y utilizadas las librerías: cv2 en su versión 4.10 y mediapipe en su versión 0.10.18 de OpenCV y la libreria screeninfo 0.8.1 para redimensionar la ventana al monitor, las cuales deben ser instaladas previamente en el entorno. Véase el diagrama de flujo del código 3.1.

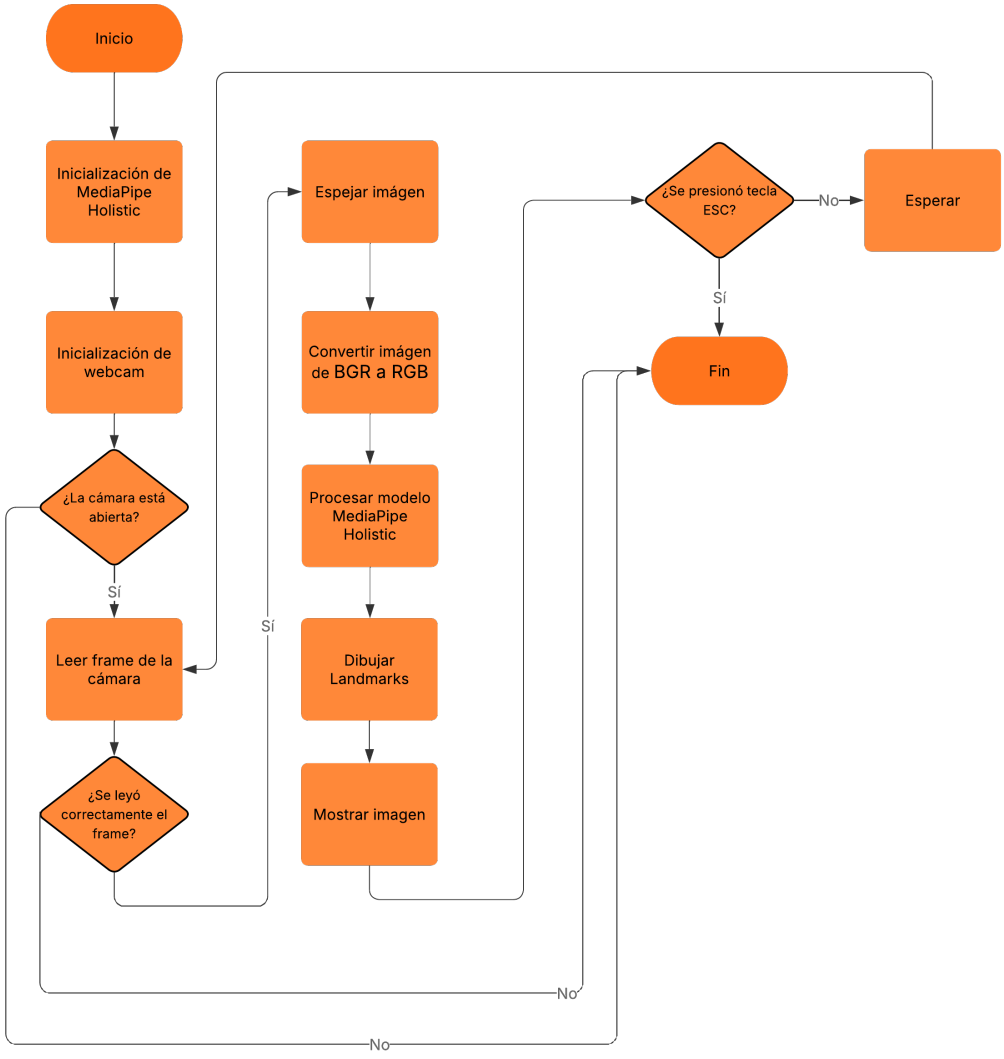


Figura 3.1: Diagrama de flujo del código de reconocimiento de posturas por webcam usando MediaPipe Holistic.

Y véanse las figuras: 3.2, 3.3 y 3.4 de ejemplo de captura de imágenes en tiempo real usando el modelo.

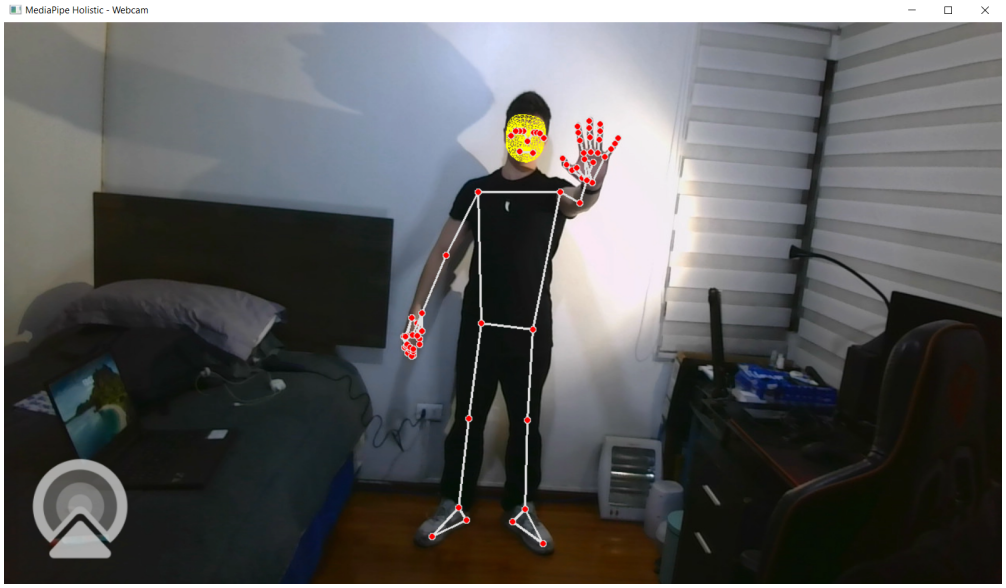


Figura 3.2: Ejemplo de captura de imágenes en tiempo real usando MediaPipe Holistic.

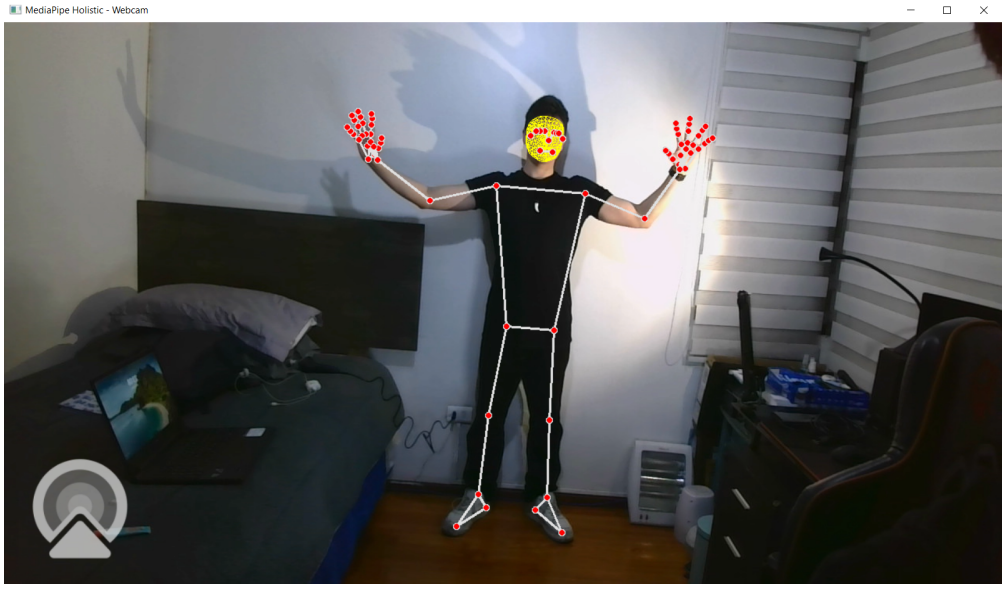


Figura 3.3: Ejemplo de captura de imágenes en tiempo real usando MediaPipe Holistic.

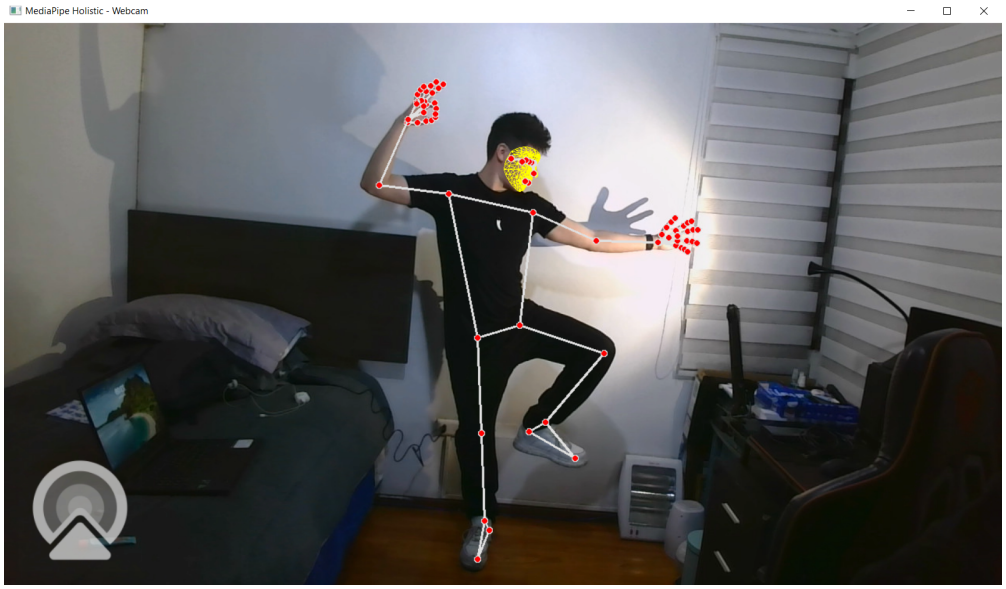


Figura 3.4: Ejemplo de captura de imágenes en tiempo real usando MediaPipe Holistic.

3.4 Comparador de 2 imágenes posturales

Se busca asociar el software de reconocimiento postural en línea con una actividad demostrativa interactiva, es por esto que se desarrolla como complemento una actividad de comparación para el usuario donde debe imitar poses preexistentes de un dataset propio del proyecto. Para llevar a cabo esto, se seleccionaron 4 imágenes posturales de distinto tamaño (en píxeles) para ver si esto afectaba o no al modelo de MediaPipe. Se desarrolló una métrica que comparaba si las posturas realizadas en las imágenes eran similares o distintas. También fueron obtenidos los Pose world landmarks (nube de puntos en coordenadas 3D) de cada una de las imágenes.

3.4.1 Mini Dataset postural

Se eligieron 2 casos de comparación. Unos karatecas dado que sus posturas son complejas y muy distintivas; Y dos mujeres haciendo la postura de árbol de navidad de yoga. Dado que también puede considerarse compleja para el software. Destaquemos que estas 4 fotos son de tamaños completamente distintos (en píxeles). Véase la figura 3.5:

Figura 3.5: Mini Dataset postural.

3.5 Lógica de comparación postural

Para la comparativa postural entre dos imágenes, nos enfocamos únicamente en el modelo de posicionamiento corporal. MediaPipe Pose detecta 33 puntos del cuerpo humano (landmarks), y para cada punto nos entrega una tupla tridimensional (x, y, z), que representa su posición normalizada en el espacio. A partir de esto, diremos que una pose es una lista de 33 puntos 3D, como se muestra en la ecuación 3.5.1:

$$\text{pose} = [(x_0, y_0, z_0), (x_1, y_1, z_1), \ldots, (x_{32}, y_{32}, z_{32})] \tag{3.5.1}$$

Para saber si dos poses son similares, comparamos sus puntos correspondientes uno a uno. La lógica asumida es: “Si dos personas hacen la misma pose, entonces sus conjuntos de landmarks deben estar muy cerca entre sí en el espacio”. Para medir esta cercanía usamos la distancia euclidiana entre cada par de puntos equivalentes. Véase la ecuación 3.5.2:

$$\text{Distancia} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{3.5.2}$$

Se calcula distancia para los 33 puntos de referencia corporales correspondientes recorriendo el listado de puntos (pose) en el mismo orden establecido en la figura 2.1.

Sin embargo, se otorga el mismo peso a todas las regiones del cuerpo, lo cual no siempre es representativo. Para mejorar la precisión, se propone agregar pesos diferentes a cada región anatómica mediante multiplicación, siguiendo la distribución de landmarks propuesta por el manual MediaPipe Pose. Cada región es definida mediante cierta cantidad de puntos (landmarks). Los pesos definidos son mostrados en la tabla 3-2:

Tabla 3-2: Regiones anatómicas, puntos asociados y pesos asignados

Región	**Puntos asociados (índices MediaPipe)**	**Peso**
Cabeza	0–10	0.1
Brazos	11–22 (desde hombros a dedos)	0.3
Piernas	23–32 (desde caderas a pies)	0.3
Tronco	11, 12, 23, 24 (desde hombros a caderas, compartidos)	0.3

Para cada región anatómica r, se calcula el promedio de las distancias euclidianas de los puntos que la componen 3.5.3:

$$\bar{d_r} = \frac{1}{n_r} \sum_{i \in r} d_i \tag{3.5.3}$$

Con estos pesos, calculamos la distancia ponderada 3.5.4:

$$\bar{d_p} = 0.1 \left(\frac{1}{n_{\text{cabeza}}} \sum_{i \in \text{cabeza}} d_i \right) + 0.3 \left(\frac{1}{n_{\text{brazos}}} \sum_{i \in \text{brazos}} d_i \right) + 0.3 \left(\frac{1}{n_{\text{piernas}}} \sum_{i \in \text{piernas}} d_i \right) + 0.3 \left(\frac{1}{n_{\text{tronco}}} \sum_{i \in \text{tronco}} d_i \right) \tag{3.5.4}$$

O de forma resumida 3.5.5, 3.5.6:

$$\bar{d_p} = \sum_r w_r \left(\frac{1}{n_r} \sum_{i \in r} d_i \right) \tag{3.5.5}$$

$$\bar{d_p} = \sum_r w_r \cdot \bar{d_r} \tag{3.5.6}$$

Donde:

- r representa cada región anatómica (cabeza, brazos, piernas, tronco)
- w_r es el peso asignado a la región r
- n_r es la cantidad de puntos en dicha región r

- d_i es la distancia euclidiana entre el punto i del usuario y el correspondiente a la imagen de referencia

Finalmente, la distancia ponderada obtenida se compara con un umbral de tolerancia τ. Si la distancia ponderada es menor a este umbral, se considera que la pose fue realizada correctamente:

$$\bar{d_p} < \tau \Rightarrow \text{Pose Similar}$$

$$\bar{d_p} \geq \tau \Rightarrow \text{Pose Diferente}$$

3.6 Diagrama de flujo del comparador de 2 imágenes

Tras el desarrollo de la lógica comparativa postural y el minidataset de este, se presenta el diagrama de flujo del comparador de 2 imágenes en la figura 3.6:

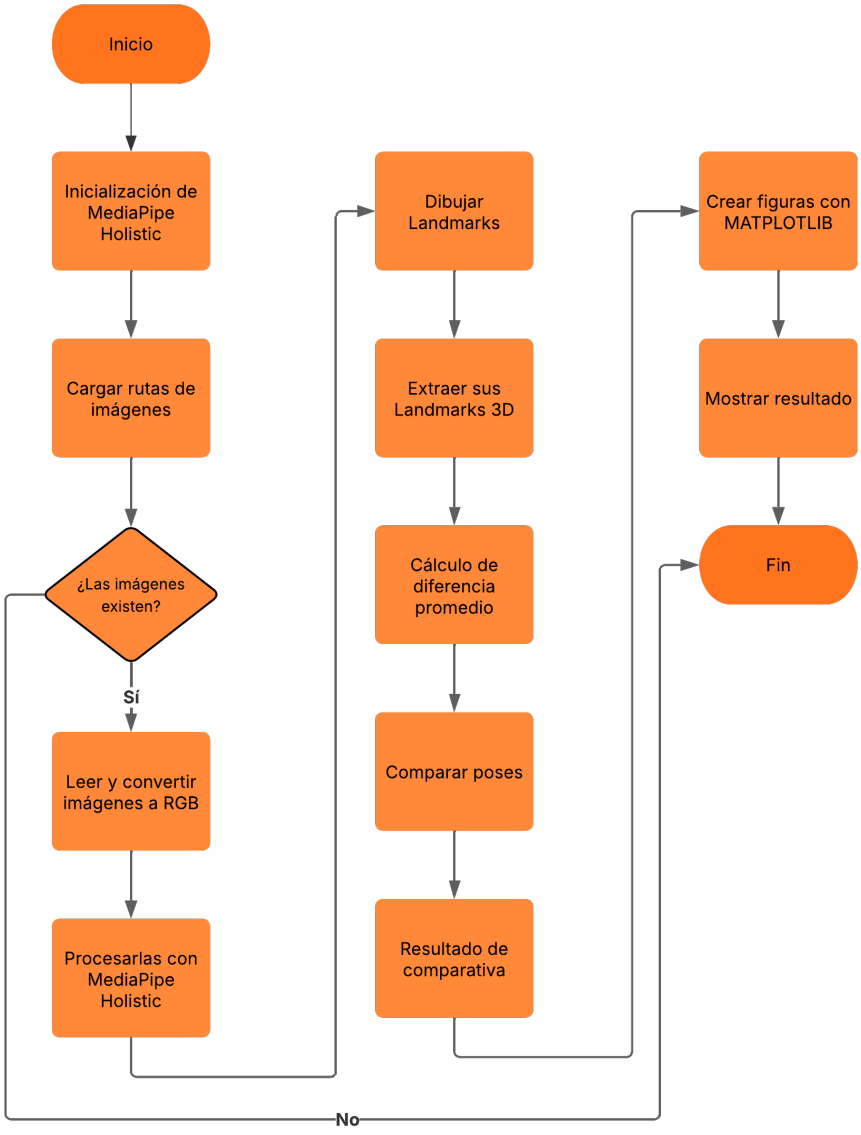


Figura 3.6: Diagrama de flujo del comparador de 2 imágenes usando MediaPipe Holistic.

3.7 Desarrollo de software comparador postural en tiempo real

Se combinan los dos software anteriormente desarrollados (uso del modelo MediaPipe Holistic y el comparador postural de dos imágenes) gracias a la lógica de comparación postural, para crear un único software que compara imágenes posturales capturadas en tiempo real con imágenes posturales de un dataset interno previamente seleccionado, como actividad didáctica y demostrativa. Se presentan imágenes de referencia que el usuario debe imitar en tiempo real, si las posturas son similares se otorgarán

puntos al usuario y se continuará con las siguientes imágenes, si las posturas son diferentes, se dirá en pantalla que lo son y se esperará hasta que el usuario las realice. Véase el diagrama de flujo del código en la figura 3.7:

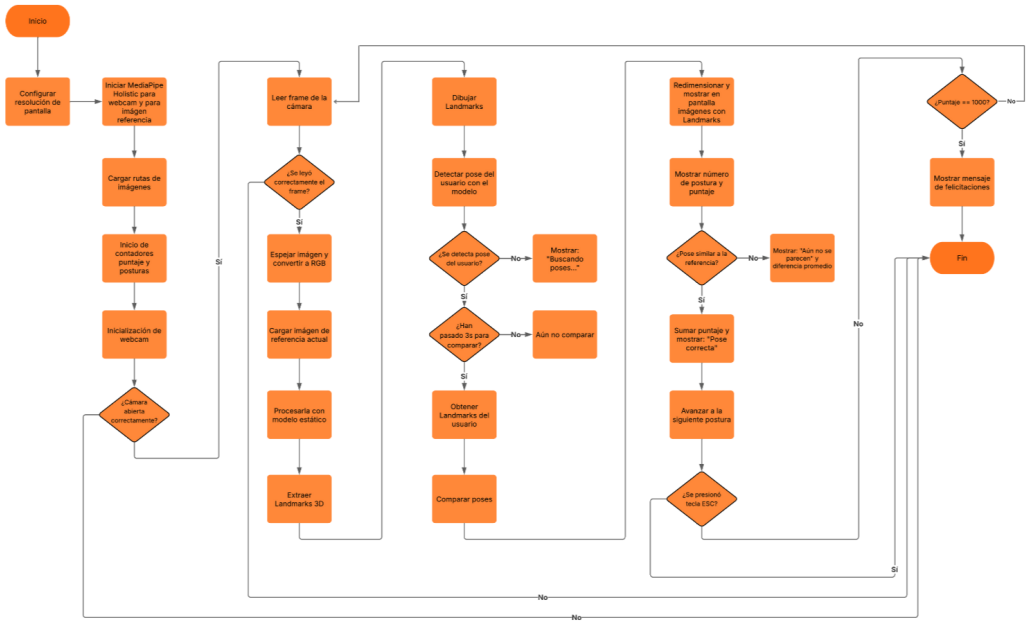


Figura 3.7: Diagrama de flujo del software comparador postural en tiempo real.

3.7.1 Dataset de prueba

Para el desarrollo de este software, fue usado un dataset de prueba de 9 imágenes y solo se consideró el uso del modelo MediaPipe Pose para las imágenes de referencia, dado que solo nos interesa el reconocimiento de posturas. Para el dataset fueron seleccionadas posturas complejas y de distinto tamaño (dimensiones en píxeles) a propósito para exigir y verificar el correcto funcionamiento del software. Véase la figura 3.8 del dataset de prueba utilizado.

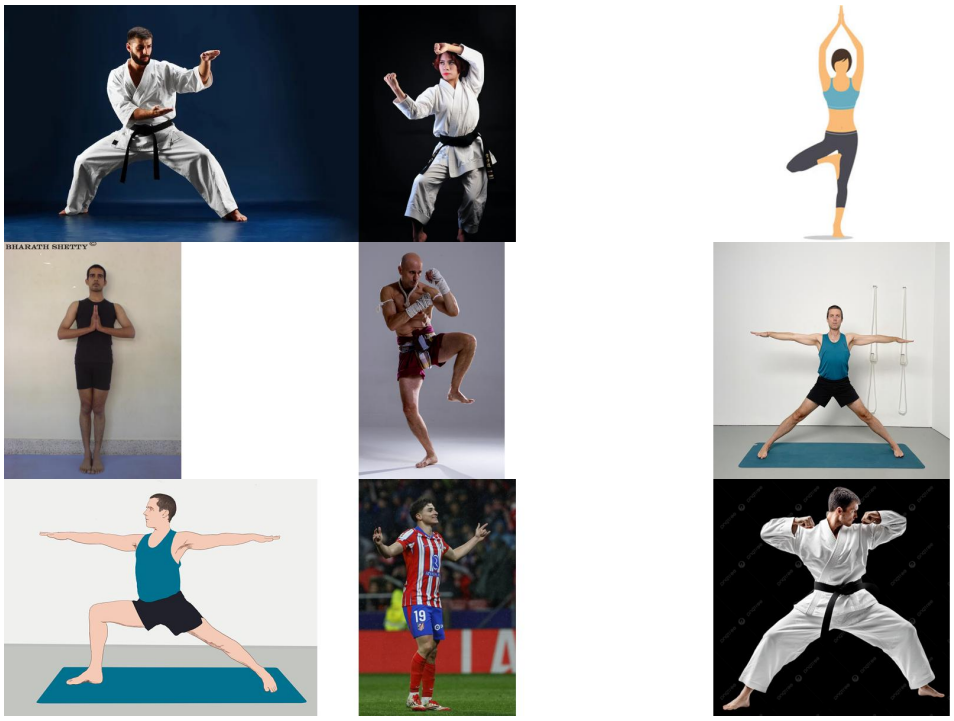


Figura 3.8: Dataset de prueba utilizado (9 imágenes posturales).

3.8 Comparador postural de yoga en tiempo real por dificultad

Se plantea añadir datasets de yoga ordenados por dificultad (fácil, intermedio y difícil) para complementar el dinamismo de la actividad demostrativa con las siguientes figuras de los dataset 3.9, 3.10 y 3.11. Se debe destacar que las imágenes de estos dataset se encuentran en formato .webp y no .jpg por lo que, se añade la libreria PIL en su versión 10.2 para leerlas correctamente y se añade la libreria tkinter 8.6 para seleccionar la carpeta dataset antes de que empiece la actividad comparativa postural.

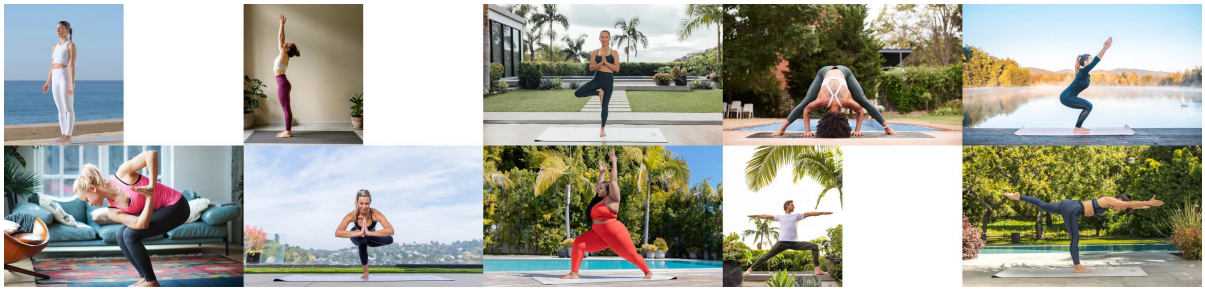


Figura 3.9: Dataset de dificultad fácil (10 imágenes posturales de yoga).

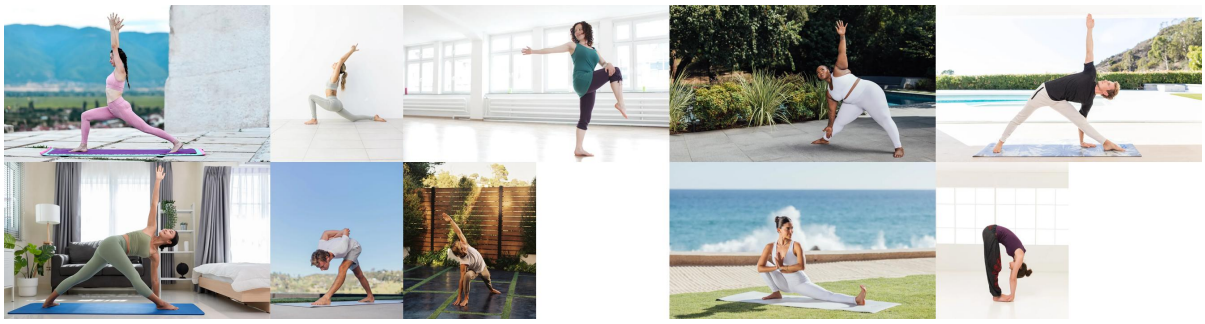


Figura 3.10: Dataset de dificultad intermedia (10 imágenes posturales de yoga).

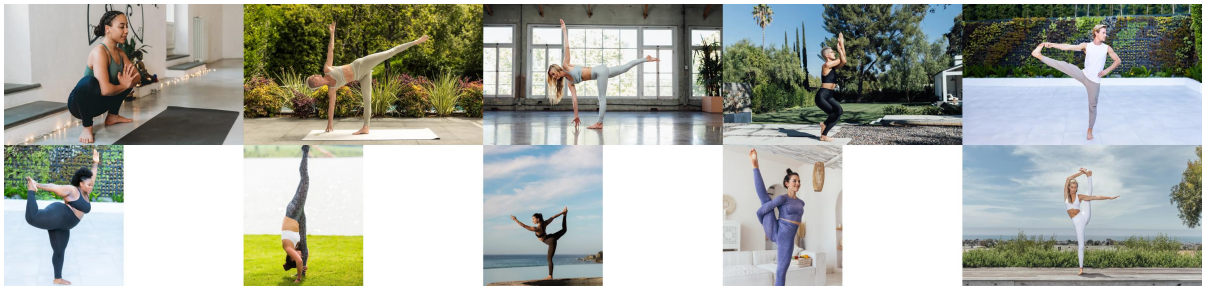


Figura 3.11: Dataset de dificultad difícil (10 imágenes posturales de yoga).

3.9 Evaluación del logro de posturas según dificultad

Se plantea que cinco usuarios realicen las 30 posturas del sistema (10 por cada dataset), con el objetivo de medir la tasa de éxito de realización según el nivel de dificultad de estos (fácil, intermedio, difícil) y analizar cómo varía dicho éxito entre las distintas posturas y diferentes personas. Se considera que cada usuario posee diferentes aptitudes físicas en cuanto a altura, peso y flexibilidad.

3.10 Discusión

En el presente capítulo fue seleccionado el modelo de visión artificial que impulsa el proyecto, el desarrollo de los software necesarios para el funcionamiento completo de este, datasets usados para la actividad demostrativa, la lógica de comparación postural requerida y, por último, una experiencia para usuarios de evaluación del logro de posturas según la dificultad de los dataset propuestos. Gracias al desarrollo de estos, se obtuvieron resultados que serán presentados en el siguiente capítulo.

Capítulo 4. Resultados

4.1 Introducción

En esta sección se exponen los resultados obtenidos tras el desarrollo de los software vistos en el capítulo anterior, se comprobará el funcionamiento de estos mediante resultados y desempeño entregado. Se evaluará el modelo de visión por computador seleccionado, verificando si es el correcto para el proyecto, y finalmente se observará el comportamiento de los dataset aplicado a usuarios reales en la realización de posturas según la dificultad.

4.2 MediaPipe Holistic mediante webcam en tiempo real

El modelo funciona correctamente, detecta los puntos de las manos, cara y posturales. Funciona de manera bastante fluida considerando el modelo predeterminado MediaPipe Holistic Model Complexity = 1 de las 3 complejidades de modelo existentes. Según los propios desarrolladores, el modelo MediaPipe Holistic tiene una precisión del 99.2 % [11] de aciertos, con muy pocos falsos positivos. Esto es corroborado cuando en escena el sujeto no alcanza a verse completamente y el modelo no dibuja sobre el extremidades falsas. Según la experiencia probando el modelo, podemos decir que depende bastante de una buena iluminación para la detección de sujetos en la imagen, y que el modelo solo reconoce una persona a la vez.

4.3 Comparador de 2 imágenes

Se exponen los resultados de las imágenes tras ser procesadas con MediaPipe Holistic de manera estática, sus nubes de puntos en coordenadas 3D y sus comparativas mediante la métrica descrita en el capítulo anterior. Sobre las imágenes cargadas se dibujan los landmarks con bastante precisión. Se debe destacar que el modelo crea nubes de puntos en 3D en base a la estimación relativa sobre una imagen 2D de entrada, dado que MediaPipe fue diseñado para funcionar con imágenes de dos dimensiones. Si se utilizaran cámaras que detectan profundidad como Kinect, se trabajarían con coordenadas reales del espacio. Para la comparativa, el umbral se configuró hasta llegar a resultados correctos. Se comenzó con un umbral de tolerancia = 0.1, el cual funcionaba en el caso de los karatecas, más no en el caso de las

poses árbol de navidad. Se determinó de manera empírica un umbral de tolerancia de 0.2 que funciona para ambos casos. Véanse las figuras 4.1 y 4.2.

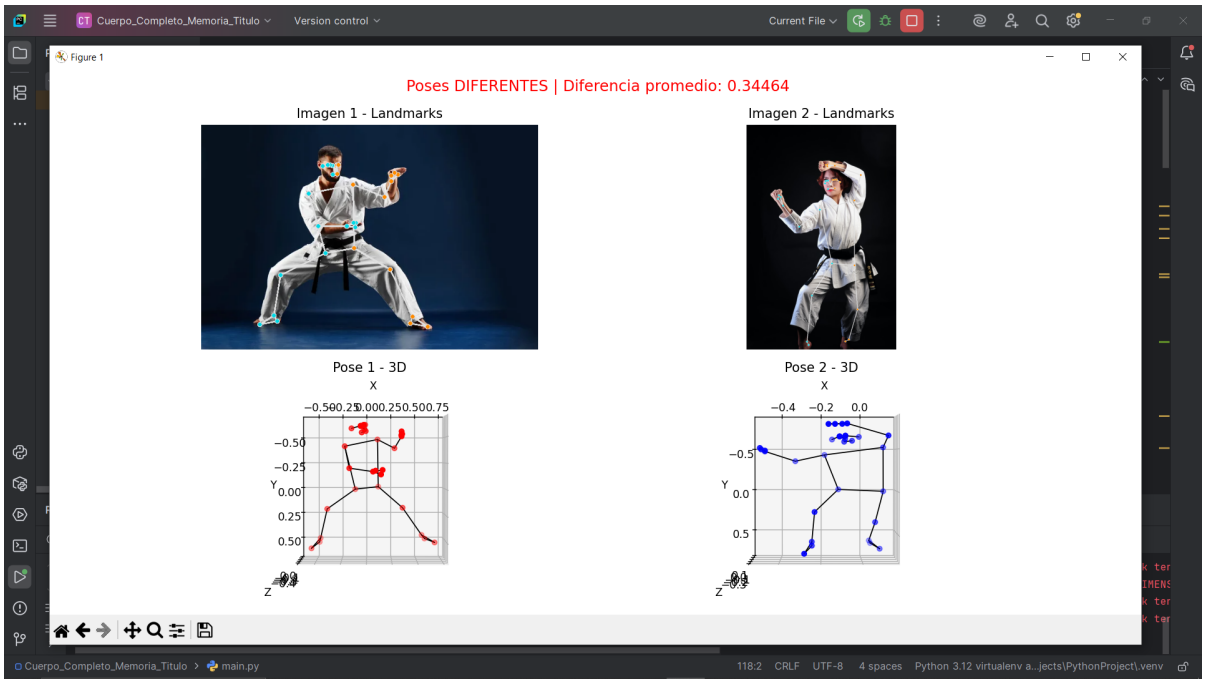


Figura 4.1: Resultado de procesamiento a karatecas con MediaPipe Holistic y su comparativa.

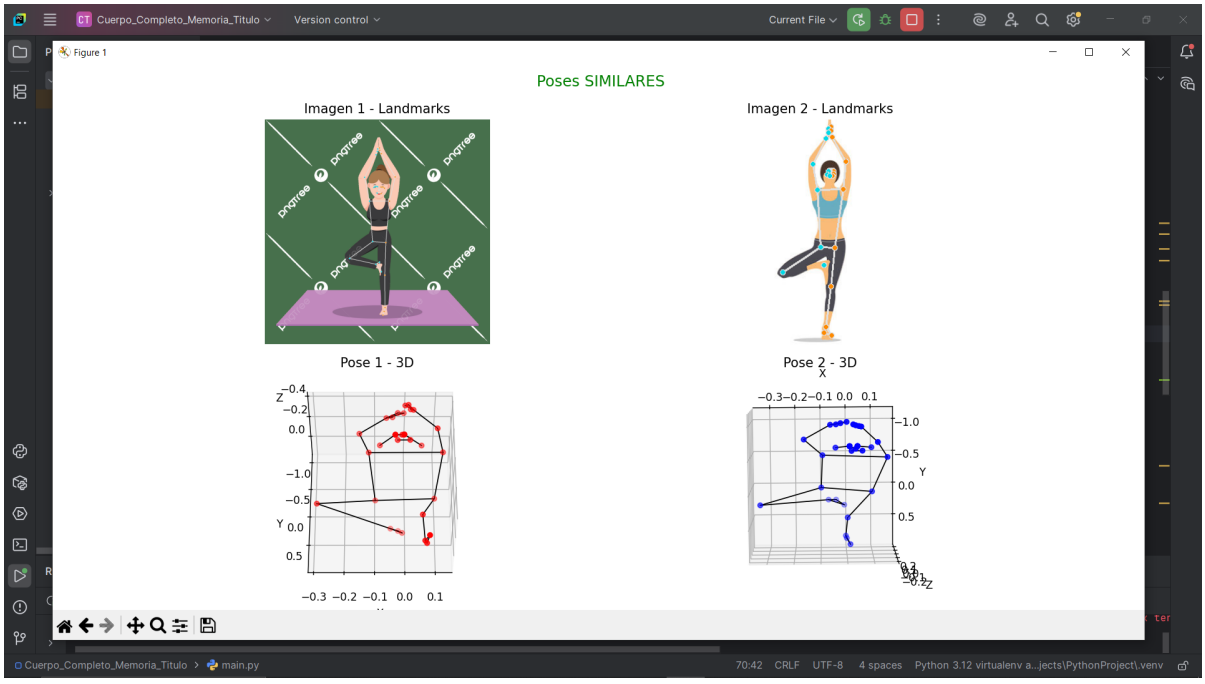


Figura 4.2: Resultado de procesamiento a postura de árbol de navidad con MediaPipe Holistic y su comparativa.

4.4 Software comparador postural en tiempo real

En términos generales, el software funciona correctamente. Se muestran landmarks para ambas imágenes, se pide al usuario realizar la postura de la imagen de referencia mientras se realiza el cálculo de comparación postural internamente; si esta es realizada correctamente, se avanza a la imagen siguiente hasta llegar al final y al terminar, el programa se cierra. Sin embargo, notemos algunas observaciones:

- Las imágenes son redimensionadas en el software, más el modelo funciona correctamente sobre estas
- El umbral debió ser reconfigurado a 0.15 tras múltiples intentos, finalmente se obtuvo éxito. Las posturas realizadas rondan este umbral y es por esto que se determina su valor. Ver figura 4.3
- El software requiere de muy buena iluminación para funcionar correctamente y no detectar falsos positivos
- Las imágenes deben ser seleccionadas cuidadosamente, ya que formatos incorrectos/incompatibles o archivos corruptos hacen que el software se cierre inesperadamente

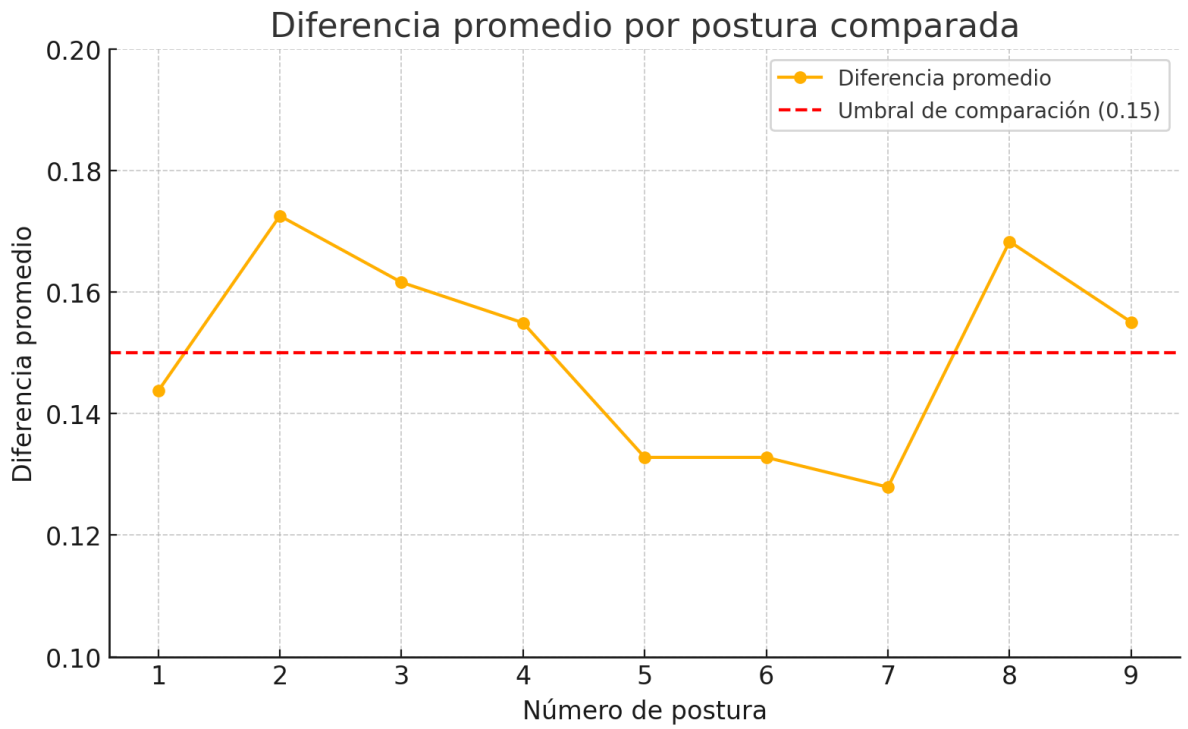


Figura 4.3: Comparación de diferencia promedio en posturas vs umbral definido.

Véanse las figuras 4.4, 4.5, 4.6 y 4.7 como ejemplo de funcionamiento del software comparador postural en tiempo real.

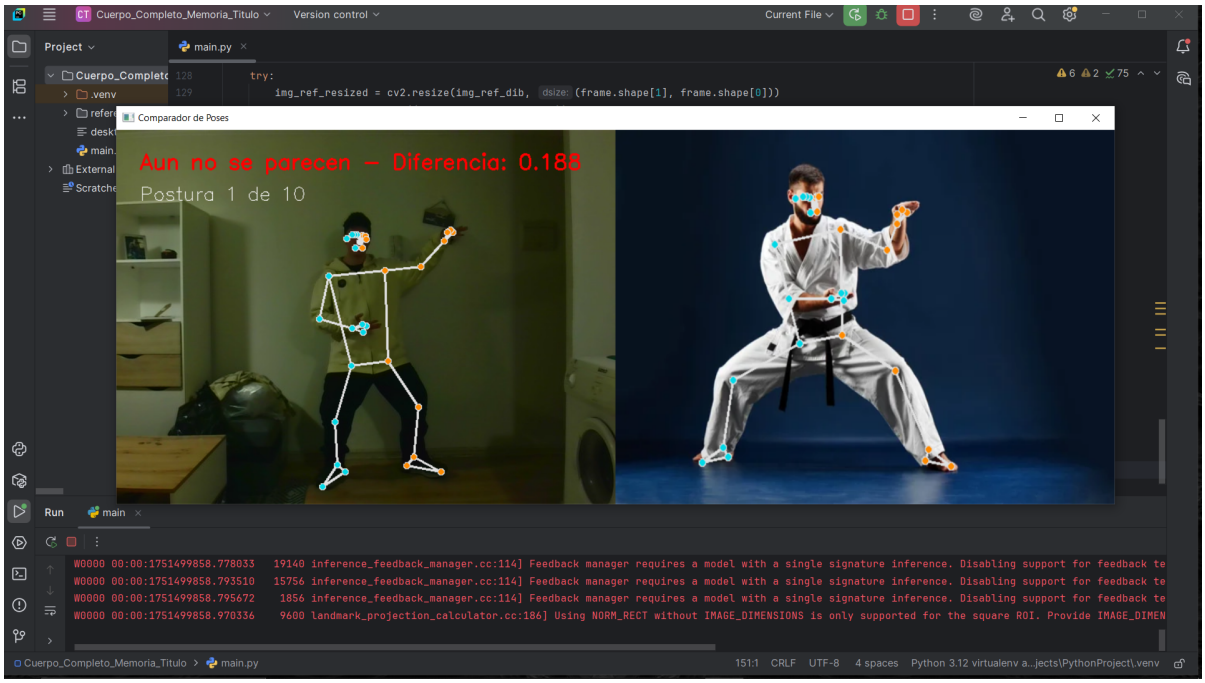


Figura 4.4: Funcionamiento del software comparador postural en tiempo real caso: Poses diferentes.

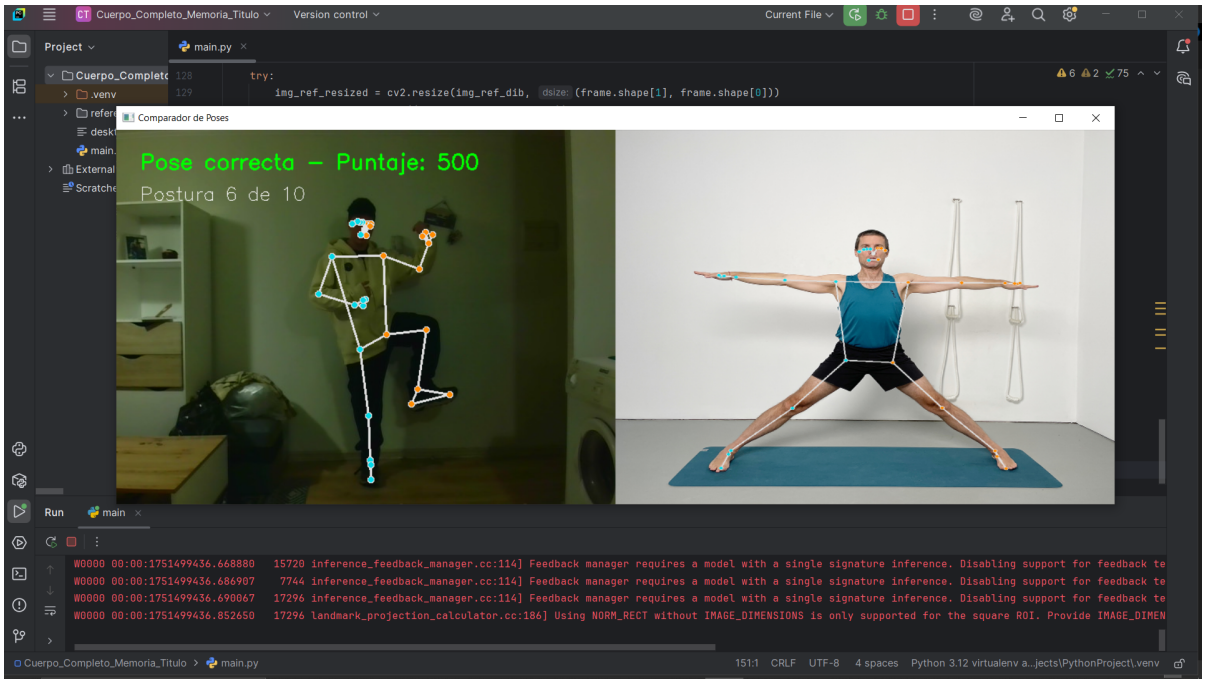


Figura 4.5: Funcionamiento del software comparador postural en tiempo real caso: Pose correcta 1.

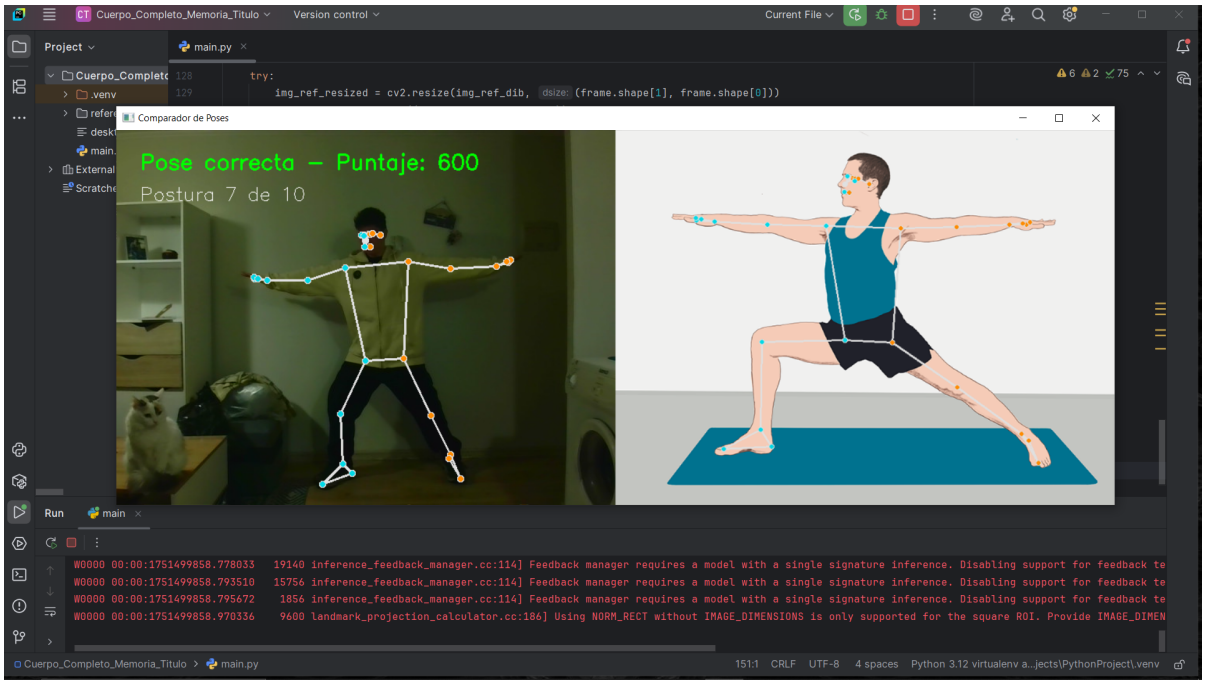


Figura 4.6: Funcionamiento del software comparador postural en tiempo real caso: Pose correcta 2.

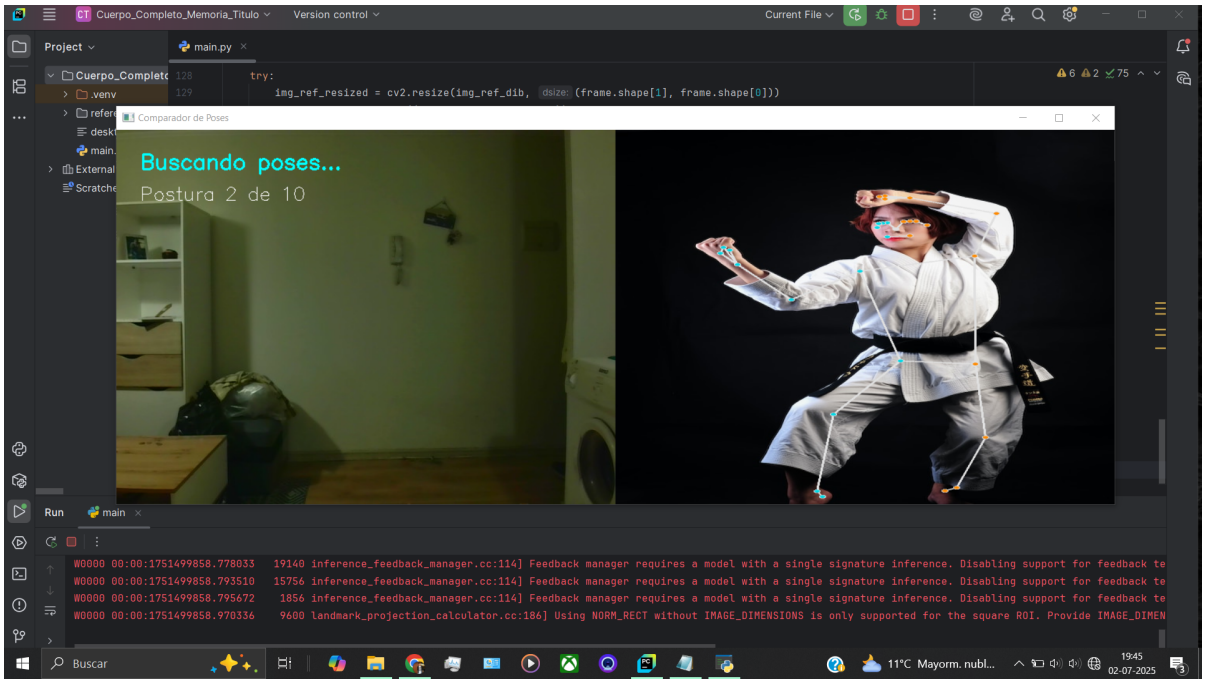


Figura 4.7: Funcionamiento del software comparador postural en tiempo real caso: No se detecta sujeto.

4.5 Comparador postural de yoga en tiempo real por dificultad

Funcionan correctamente los dataset, se pide al usuario que elija la dificultad que desea y debe seleccionar su carpeta (Ver figura 4.8). Se cumple con el objetivo de clasificación por dificultad. Sin embargo, para el caso de posturas de yoga, se requiere cierto nivel de control sobre el cuerpo, algunos usuarios sugieren que las poses son difíciles.

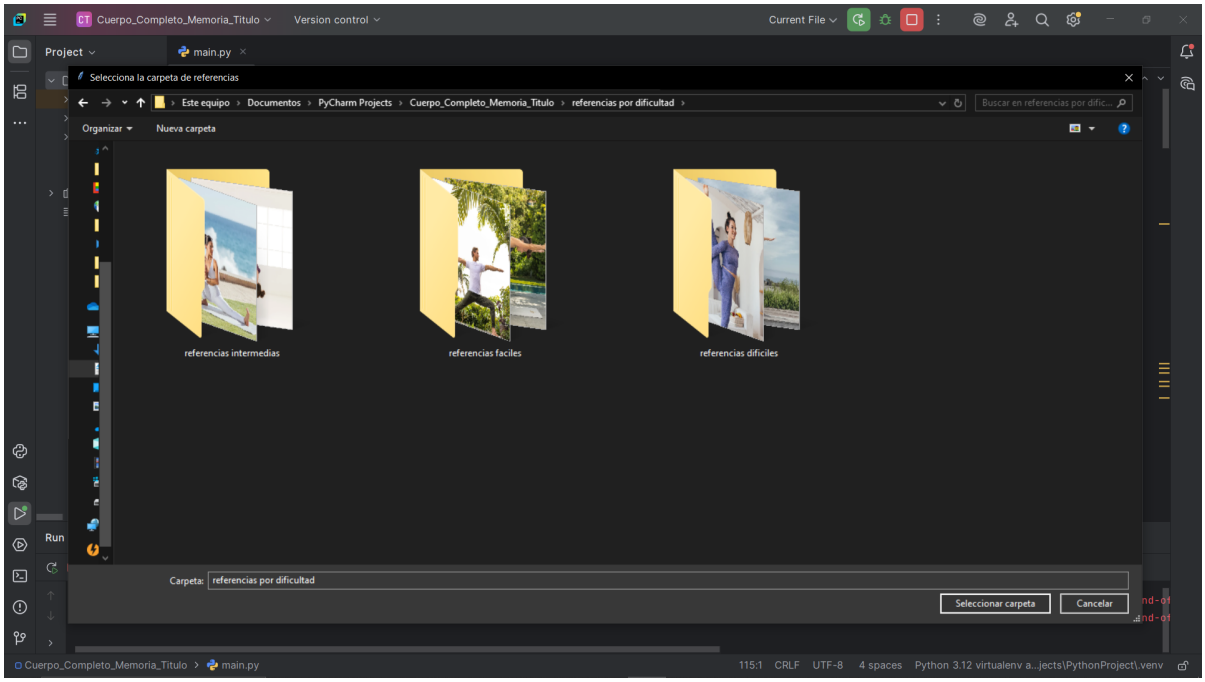


Figura 4.8: Selección de dataset en carpeta según dificultad.

Véanse también las figuras de ejemplo: 4.9, 4.10 y 4.11.

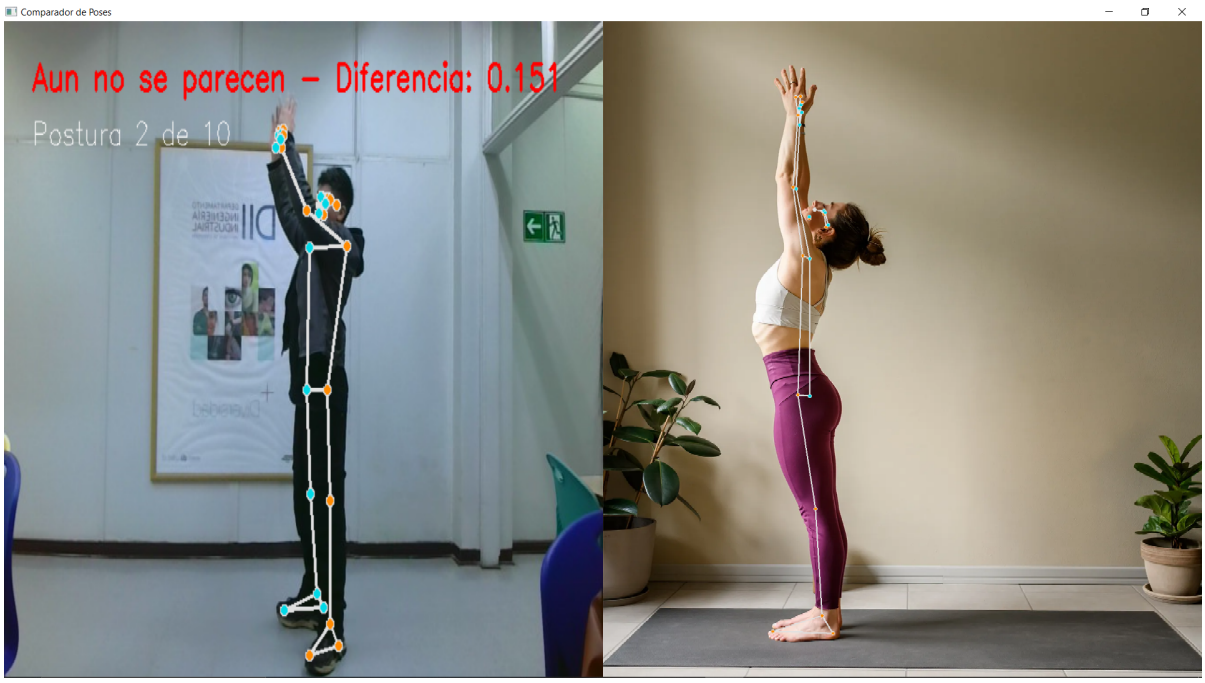


Figura 4.9: Ejemplo del comparador postural en tiempo real en dificultad fácil.

Figura 4.10: Ejemplo del comparador postural en tiempo real en dificultad intermedia.

Figura 4.11: Ejemplo del comparador postural en tiempo real en dificultad difícil.

4.6 Evaluación del logro de posturas según dificultad

Tras la experiencia de evaluación postural por dificultad a cinco usuarios, se obtuvieron los siguientes resultados. Ver tabla 4-3.

Tabla 4-3: Cantidad de personas que lograron cada postura por dataset de dificultad

Postura	Dataset fácil	Dataset intermedio	Dataset difícil
1	5 logran	5 logran	4 logran
2	5 logran	5 logran	3 logran
3	4 logran	4 logran	3 logran
4	3 logran	5 logran	5 logran
5	5 logran	3 logran	0 logran
6	5 logran	3 logran	3 logran
7	2 logran	2 logran	0 logran
8	5 logran	3 logran	2 logran
9	5 logran	1 logra	0 logran
10	1 logra	0 logran	0 logran

En general:

- **Dataset fácil:** promedio de **80 %** de posturas logradas por los usuarios.
- **Dataset intermedio:** promedio de **62 %** de posturas logradas.
- **Dataset difícil:** promedio de **40 %** de posturas logradas.

Véase el gráfico del porcentaje de posturas logradas según nivel de dificultad 4.12:

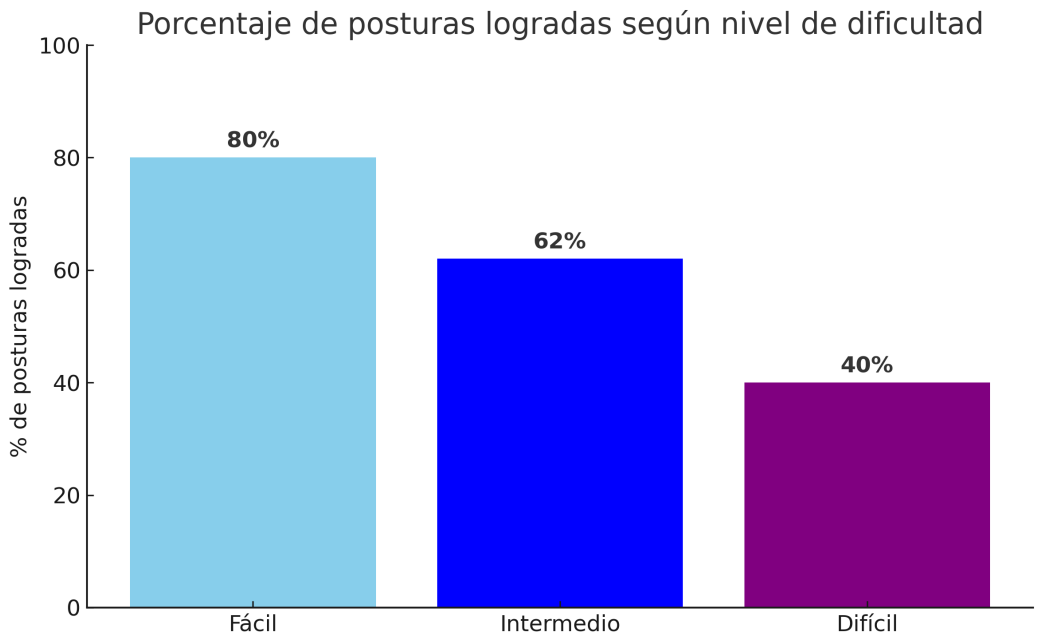


Figura 4.12: Porcentaje de posturas logradas según nivel de dificultad.

4.7 Discusión

Los resultados obtenidos gracias al desarrollo de los software, la lógica de comparación de posturas y la experiencia del logro de posturas según su nivel de dificultad, dan paso para extraer conclusiones a partir de estos, que serán vistos en el capítulo siguiente junto a sugerencias de trabajo futuro para el sistema, tales como optimización y mejoras que pueda experimentar este mismo.

Capítulo 5. Conclusiones

En esta sección se presentan las conclusiones finales del trabajo desarrollado, considerando el modelo seleccionado, software desarrollado, la lógica comparativa postural y la evaluación de logro de posturas por nivel de dificultad. Finalmente, se incluye un breve apartado de trabajo futuro.

5.1 Conclusión

Se concluye que los objetivos planteados en el Capítulo 1 fueron cumplidos en su totalidad. El modelo de visión por computador seleccionado, MediaPipe Holistic, resultó adecuado, permitiendo obtener y visualizar en tiempo real la postura del usuario y compararla con una imágenes de referencia mediante una lógica de comparación postural desarrollada en el proyecto. El sistema asigna puntaje al usuario al reconocer posturas correctas y muestra información visual asociada al desempeño, incluyendo landmarks corporales, visualización de la postura actual y la puntuación acumulada. Además, se incorporaron tres datasets de distinta dificultad para la actividad demostrativa, lo que permite a los usuarios seleccionar la experiencia más adecuada para cada uno de estos.

Los resultados mostraron que el porcentaje promedio de posturas logradas disminuye progresivamente al aumentar la dificultad: 80 % en el nivel fácil, 62 % en el nivel intermedio y 40 % en el nivel difícil. Esto refleja una relación directa entre el nivel de dificultad del dataset y el desempeño del usuario, aunque con algunas excepciones. Se observaron posturas clasificadas como difíciles que fueron logradas por todos los participantes de la experiencia, y también, posturas fáciles que no fueron alcanzadas por la totalidad de los usuarios. Lo anterior indica que la clasificación de dificultad realizada en los datasets no siempre coincide con el desafío real que implica cada postura, ya que factores físicos y contextuales, como flexibilidad, condición física, edad y experiencia previa en actividades similares (por ejemplo yoga), influyen de manera significativa en el resultado. En este sentido, el sistema permite evidenciar que variables personales del usuario influyen directamente en el logro postural.

El sistema depende de condiciones externas. Factores como la iluminación del entorno, la calidad de la cámara utilizada (resolución), la estabilidad de la señal de video y la correcta calibración del umbral de comparación influyen en la precisión de los resultados. El umbral definido empíricamente funcionó de manera adecuada para las pruebas, aunque se reconoce que podría ajustarse según las características del usuario. Esto abre la posibilidad de personalizar el sistema, haciéndolo más permisivo o más estricto

de acuerdo con las necesidades.

Varios participantes manifestaron que las posturas basadas en yoga resultaron complejas debido a la falta de familiaridad con este tipo de actividad física, lo que evidencia que la percepción de dificultad no depende únicamente del algoritmo sino también de la experiencia previa del usuario. Esta observación complementa la interpretación cuantitativa de los resultados.

En términos generales, el proyecto logró demostrar la factibilidad de construir un sistema de reconocimiento postural en tiempo real, capaz de otorgar retroalimentación inmediata, establecer niveles de dificultad y cuantificar el desempeño del usuario de forma objetiva. Es un aporte en el área de aplicaciones interactivas basadas en visión por computador, con potencial para extenderse a contextos educativos, recreativos y clínicos.

5.2 Trabajo futuro

El sistema desarrollado abre diversas líneas de trabajo futuro. En primer lugar, se podría mejorar la precisión de la comparación postural mediante el uso de técnicas más avanzadas de análisis de similitud, como métricas de aprendizaje profundo. Otra posible mejora corresponde a la incorporación de un mayor número de datasets, con una variedad más amplia de posturas, incluyendo tanto ejercicios físicos cotidianos como movimientos empleados en rehabilitación o entrenamiento deportivo. Esto permitiría evaluar de mejor forma la generalización del sistema y su capacidad de adaptación a diferentes contextos.

Desde el punto de vista técnico, el sistema puede beneficiarse con uso de cámaras de mayor calidad, sensores de profundidad o incluso dispositivos de captura de movimiento, lo cual entregaría información tridimensional más precisa y reduciría la dependencia de la iluminación del entorno. Asimismo, sería posible optimizar la interfaz gráfica, haciéndola más amigable y accesible para distintos grupos de usuarios, considerando criterios de usabilidad e inclusión.

En cuanto a aplicaciones, el alcance del sistema puede extenderse más allá de la actividad demostrativa presentada. Podría utilizarse en educación física, en rehabilitación, como herramienta de apoyo en terapias de movimiento; en biometría y seguridad, para el reconocimiento de gestos; y en robótica o realidad aumentada, como método de control basado en posturas humanas. También es factible su uso en contextos de fitness y entrenamiento personal, permitiendo a los usuarios recibir feedback inmediato en sus rutinas de ejercicio.

Finalmente, un aspecto clave de trabajo futuro es la validación con un número mayor de participantes

y en entornos más diversos. Esto permitiría obtener métricas más robustas y generalizables, identificar patrones de error recurrentes y evaluar la aceptación del sistema en contextos más reales.

Glosario

CV	Computer Vision
DL	Deep Learning
LF	Low Filter
HF	High Filter
CNNs	Redes neuronales convolucionales
CPU	Unidad Central de Procesamiento
GPU	Unidad de Procesamiento Gráfico

Referencias

[1] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[3] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," 2016. [Online]. Available: https://arxiv.org/abs/1602.00763

[4] S. Aggarwal, I. Gupta, A. Kumar, S. Kautish, A. S. Almazyad, A. W. Mohamed, F. Werner, and M. Shokouhifar, "Gastrofuse-net: An ensemble deep learning framework designed for gastrointestinal abnormality detection in endoscopic images," *Mathematical Biosciences and Engineering*, vol. 21, no. 8, pp. 6847–6869, 2024. [Online]. Available: https://www.aimspress.com/article/doi/10.3934/mbe.2024300

[5] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," 2016. [Online]. Available: https://arxiv.org/abs/1604.07316

[6] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Pearson, 2018.

[7] M. Cheng and M. Liu, "Image convolution techniques integrated with yolov3 algorithm in motion object data filtering and detection," *Scientific Reports*, vol. 14, no. 1, p. 7651, 2024.

[8] G. Bradski, "The opencv library," *Dr. Dobb's Journal of Software Tools*, 2000, https://opencv.org/about/.

[9] OpenCV Foundation. (2025) Opencv: Open source computer vision library. Official website of the OpenCV project. [Online]. Available: https://opencv.org/

[10] OpenCV Contributors. (2024) Opencv documentation, version 4.11.0. Technical documentation maintained by the OpenCV development team. [Online]. Available: https://docs.opencv.org/4.11.0/

[11] V. Bazarevsky, I. Kartynnik, A. Vakunov, A. Tkachenka, and M. Grundmann, "Mediapipe holistic: Simultaneous face, hand and pose prediction, on-device," https://research.google/blog/mediapipe-holistic-simultaneous-face-hand-and-pose-prediction-on-device/, 2020, google AI Blog, acceso el 18 de mayo de 2025.

[12] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. Yong, J. Lee *et al.*, "Mediapipe: A framework for perceiving and processing reality," in *Third workshop on computer vision for AR/VR at IEEE computer vision and pattern recognition (CVPR)*, vol. 2019, 2019.

[13] A. Moryossef, "Optimizing hand region detection in mediapipe holistic full-body pose estimation to improve accuracy and avoid downstream errors," 2024.

[14] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "Mediapipe hands: On-device real-time hand tracking," 2020.

[15] J. M. Wachter, "Schwellwertanalyse bezüglich der limitierung von mediapipe face mesh anhand von informationsreduziertem ausgangsmaterial," B.S., 2022.

[16] I. Grishchenko and V. Bazarevsky, "Mediapipe holistic," *URL: https://google. github. io/mediapipe/solutions/holistic. html*, 2020.

[17] Z. e. a. Cao, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *arXiv preprint arXiv:1812.08008*, 2018. [Online]. Available: https://arxiv.org/abs/1812.08008

[18] "Validation of openpose for detecting postural control deficits," *Scientific Reports*, vol. 11, no. 1, 2021. [Online]. Available: https://www.nature.com/articles/s41598-021-00212-x

[19] V. e. a. Bazarevsky, "Blazepose: On-device real-time body pose tracking," 2020. [Online]. Available: https://arxiv.org/abs/2006.10204

[20] V. Bazarevsky, I. Grishchenko, K. Raveendran *et al.*, "Blazepose: On-device real-time body pose tracking," *arXiv preprint arXiv:2006.10204*, 2020. [Online]. Available: https://arxiv.org/abs/2006.10204

[21] J. Lauer, M. Zhou, S. Ye, W. Menegas, T. Nath, M. M. Rahman, V. Di Santo, D. Soberanes, G. Feng, V. N. Murthy *et al.*, "Multi-animal pose estimation and tracking with deeplabcut," pp. 2021–04, 2021.

[22] J. Lauer, M. Zhou, S. Ye, W. Menegas, S. Schneider, T. Nath, M. M. Rahman, V. Di Santo, D. Soberanes, G. Feng *et al.*, "Multi-animal pose estimation, identification and tracking with deeplabcut," *Nature Methods*, vol. 19, no. 4, pp. 496–504, 2022.

[23] I. G. Rivera Itúrburu and D. F. Zambrano Guaranda, "Implementación de reconocimiento facial y visión artificial en robot nao con python y opencv," B.S., 2022.

[24] D. Costa Mari, “Análisis de un sistema de reconocimiento facial a partir de una base de datos realizado mediante python,” B.S. thesis, Universitat Politècnica de Catalunya, 2020.

[25] P. Viola and M. J. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001, pp. 511–518. [Online]. Available: https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf

[26] Google AI Edge Team. (2024) Mediapipe holistic - full body solution. Official documentation maintained by the MediaPipe development team at Google. [Online]. Available: https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/holistic.md

[27] MediaPipe Development Team, Google. (2024) Mediapipe pose landmarker python api. Official API reference for the Pose Landmarker solution in MediaPipe. [Online]. Available: https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker/python?hl=es-419

Anexo A. Programación desarrollada en software.

Este anexo describe funciones importantes implementadas en el desarrollo de los software:

Tabla 0-4: Funciones principales del sistema de comparación de poses

Función	**Descripción**
`get_pose_world_landmarks(result)`	Extrae coordenadas tridimensionales (x, y, z) de los 33 landmarks corporales detectados por MediaPipe. Se utiliza para obtener las poses del usuario como de la imagen de referencia.
`comparar_poses_ponderada(p1, p2, tolerancia)`	Compara dos listas de puntos 3D (una del usuario y otra de referencia) mediante una distancia ponderada por región anatómica (cabeza, brazos, piernas y tronco). Devuelve si la pose es similar y la distancia.
`modelo_usuario.process(rgb_frame)`	Procesa un fotograma en vivo desde la cámara para detectar los landmarks del cuerpo del usuario usando MediaPipe Holistic.
`modelo_referencia.process(img_ref_rgb)`	Procesa la imagen de referencia cargada y extrae los landmarks corporales.
`mp_drawing.draw_landmarks(...)`	Dibuja los puntos landmarks y sus conexiones sobre el usuario o la imagen de referencia, para visualización en pantalla.
`cv2.imshow(...)`	Muestra en una ventana la comparación entre la imagen de referencia y la captura en vivo del usuario, junto con mensajes sobre puntaje y pose actual.

UNIVERSIDAD DE CONCEPCIÓN – FACULTAD DE INGENIERÍA
RESUMEN DE MEMORIA DE TÍTULO

Departamento : Departamento de Ingeniería Civil Eléctrica
Carrera : Ingeniería Civil Biomédica
Nombre del memorista : Ricardo Francisco Ríos Díaz
Título de la memoria : Sistema interactivo de seguimiento de posición corporal mediante visión artificial para actividad demostrativa.
Fecha de la presentación oral : 3 de Septiembre de 2025
Profesor(es) Guía : Esteban Pino Quiroga, Mabel Urrutia Martínez
Profesor(es) Revisor(es) : Pamela Guevara Alvez
Concepto :
Calificación :

Resumen

El presente trabajo consiste en el desarrollo de un software de comparación entre dos posturas en tiempo real, utilizando un modelo de visión por computador seleccionado, MediaPipe Holistic de Google, el cual detecta puntos articulares clave en el usuario y compara su postura con imágenes de referencia propios del proyecto usando una interfaz gráfica de visualización. Para llevar a cabo esto, se desarrolla una lógica comparativa postural basada en diferencias promedio con un umbral de tolerancia; Se añaden datasets por dificultad y se evalúa el logro de posturas por nivel de dificultad en usuarios.

Se concluye que el sistema funciona correctamente, cumple objetivos descritos y queda sujeto a actualizaciones de trabajo futuro.