

# Predicción Automática de Golpes de Pádel mediante Análisis de Video con Inteligencia Artificial (padelpose)

Máster en Inteligencia Artificial Avanzada y Aplicada

Autor:

José Julián Mazcuñán Navarro

Tutores:

Valero Laparra Pérez-Muelas

Jorge Vila Tomás

Valencia, Julio 2024

## Contents

1	Introducción .....	4
1.1	Contexto y motivación .....	4
1.2	Objetivos del trabajo .....	4
1.3	Resultados esperados y contribuciones .....	6
2	Estado del arte .....	7
2.1	Inteligencia Artificial en deportes .....	7
2.2	Inteligencia Artificial en el pádel .....	13
2.3	Modelos de estimación de pose .....	19
2.3.1	MoveNet .....	19
2.3.2	PoseNet .....	19
2.3.3	YOLOv8 pose estimation .....	20
2.3.4	MediaPipe .....	20
2.3.4.1	BlazePose .....	22
3	Metodología .....	23
3.1	Dataset .....	23
3.1.1	Datos de entrada: .....	24
3.1.2	Datos de salida .....	27
3.1.3	Análisis exploratorio de los datos de entrada .....	28
3.1.4	Transformaciones sobre los datos de entrada .....	30
3.2	Modelado del clasificador .....	33
3.2.1	Modelos de clasificación simples .....	33
3.2.2	Modelo basado en CNN y LSTM .....	34
3.2.3	Creación de modelos .....	36
3.3	Despliegue del modelo en una webapp .....	39
3.3.1	Ingesta de vídeo .....	40
3.3.2	Procesado e inferencia .....	42
3.3.3	Feedback .....	43
3.3.4	Visualizaciones adicionales .....	45
4	Resultados .....	48
5	Conclusiones y trabajos futuros .....	58
5.1	Conclusiones .....	58
5.2	Trabajos futuros .....	58

6. Bibliografía .....	60
7 Anexo .....	63
7.1 Informes de evaluación completos .....	63
7.1.1 Coordenadas .....	63
7.1.2 Distancias .....	66
7.1.3 Ángulos.....	69
7.1.4 LSTM.....	72
7.2 Pruebas en la webapp .....	73

# 1 Introducción

## 1.1 Contexto y motivación

El pádel ha experimentado un crecimiento exponencial en popularidad a nivel mundial en los últimos años. Según la Federación Internacional de Pádel [1], más de 25 millones de personas juegan al pádel en más de 90 países alrededor del mundo. En España, el pádel se ha convertido en el segundo deporte más popular, con más de 6 millones de jugadores activos y más de 20,000 pistas de pádel, solo por detrás del fútbol. Este incremento ha llevado a una mayor demanda de técnicas avanzadas para mejorar el rendimiento de los jugadores, optimizar el entrenamiento y prevenir lesiones.

El análisis detallado de los movimientos de los jugadores de pádel es crucial para alcanzar estos objetivos. La inteligencia artificial (IA), especialmente mediante técnicas de visión, ofrece herramientas avanzadas para este análisis. Utilizando IA, es posible monitorear y evaluar en tiempo real los diferentes golpes y movimientos de los jugadores, proporcionando retroalimentación instantánea y personalizada. Este enfoque no solo mejora la precisión y eficacia del entrenamiento, sino que también facilita el desarrollo de estrategias de juego más efectivas.

El presente trabajo aborda la clasificación de distintos tipos de golpe de pádel (derecha, revés, volea, bandeja, remate) a partir de vídeos de jugadores realizando estos movimientos. La cuestión principal que se busca responder es cómo optimizar la clasificación de estos golpes mediante el uso de IA y visión artificial.

Este proyecto no solo tiene relevancia académica, sino también una aplicabilidad práctica directa, ofreciendo una solución accesible para la comunidad deportiva del pádel. Al integrar estas tecnologías en una webapp, se busca democratizar el acceso a técnicas avanzadas de análisis deportivo, facilitando su adopción tanto por profesionales como por aficionados.

## 1.2 Objetivos del trabajo

El objetivo principal de este trabajo es la creación de un clasificador de golpes de pádel. La idea es desarrollar un modelo de IA capaz de identificar y clasificar automáticamente los diferentes tipos de golpe de pádel a partir de vídeos, proporcionando una herramienta precisa y eficiente para el análisis de movimientos.

### Objetivos específicos:

- Creación del dataset: Recolectar y etiquetar un conjunto de datos de vídeos de jugadores de pádel realizando diferentes tipos de golpes. Asegurar la calidad y diversidad del dataset para cubrir una amplia gama de variaciones en los movimientos, así como diferentes perspectivas. Hacer público el dataset para facilitar su acceso.
- Extracción de características a partir del vídeo: Implementar un modelo de reconocimiento de keypoints basado en IA, específicamente la solución de estimación de pose de MediaPipe.
- Entrenamiento y evaluación de modelos: Diseñar un flujo de trabajo que incluya la extracción de características a partir del conjunto de vídeos etiquetados, el entrenamiento del modelo y su evaluación.

- Desarrollo de una webapp: Implementar una webapp que permita la inferencia, donde los usuarios puedan subir vídeos y recibir predicciones sobre los tipos de golpe. Integrar una interfaz intuitiva para la visualización de resultados y la retroalimentación.
- Mejora a partir de datos de uso: Habilitar la posibilidad de que los usuarios puedan corroborar o corregir los resultados del modelo. Almacenar esta información para revisarla y retroalimentar al sistema en sucesivas iteraciones de entrenamiento para que aprenda a partir de nuevos datos y se mejore su robustez.

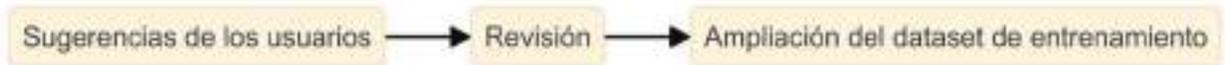
Flujo en entrenamiento:



Flujo en inferencia:



Flujo de feedback:



El ciclo de feedback se plantea como un medio para convertir el sistema en un ciclo iterativo (representado en la siguiente figura), de forma que se pueda retroalimentar el dataset en base al uso de la herramienta y, por lo tanto, el sistema se robustezca y siga aprendiendo en base al uso, siempre con una supervisión intermedia para consolidar las sugerencias (correcciones o confirmaciones) del usuario sobre el dataset de entrenamiento.

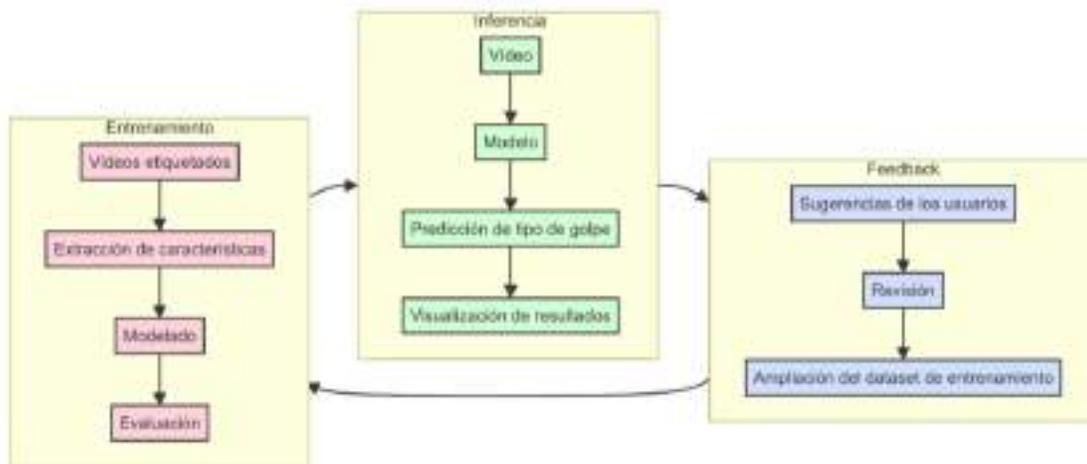


Figura 1 – Ciclo iterativo de desarrollo de la aplicación

### 1.3 Resultados esperados y contribuciones

El presente trabajo espera demostrar que el uso de técnicas de inteligencia artificial y visión artificial puede clasificar con alta precisión los diferentes tipos de golpe de pádel a partir de vídeos de jugadores. Específicamente, se anticipa que el modelo desarrollado logrará identificar correctamente los golpes de derecha, revés, volea, bandeja y remate en una amplia variedad de condiciones y estilos de juego.

Los resultados incluyen:

- a) Ejecución de una solución completa basada en análisis por IA:

El proyecto engloba la ejecución completa de una solución basada en IA, desde la fase de creación del dataset, extracción de características, procesado de las mismas, pruebas de distintos modelos, despliegue para inferencia y visualización, así como el ciclo de retroalimentación.

El modelo implementado en una webapp simula un entorno accesible para el usuario, que puede ser empleado desde dispositivos móviles, habilitando que la herramienta se pueda usar incluso desde la pista de pádel, grabando un vídeo corto del movimiento, subiéndolo a la web y visualizando el resultado tras un breve instante.

- b) Alta precisión en la clasificación de golpes:

Se espera que el modelo alcance una alta precisión en la identificación y clasificación de los tipos de golpe de pádel, con métricas de rendimiento superiores al 90% en precisión, recall y F1-score.

Como posibles futuras contribuciones al estado del arte, se enumeran los siguientes posibles usos prácticos:

Cuantificar la distribución de golpes en una partida: analizar la cantidad de distintos tipos de golpes, por ejemplo, utilizando el clasificador en cascada con un detector de golpes que funcionase con un stream de vídeo de un partido completo, que permitiría automatizar la extracción de estadística de juego: cuántos golpes de cada tipo ha habido por parte de cada jugador, etc.

Mejora del rendimiento: si bien el clasificador busca reconocer el tipo de golpe incluso aunque la técnica de ejecución no sea perfecta, podría añadirse información sobre el movimiento “maestro” en base al input de algún especialista en la materia, de forma que se podría comparar la secuencia de keypoints capturada por el usuario frente a la secuencia perfecta y ofrecer sugerencias al usuario para corregir su movimiento y mejorar su técnica.

Prevención de lesiones: Del mismo modo, se podría analizar la posición de las articulaciones del jugador y, empleando de nuevo información especializada, ofrecer avisos en el caso de que se detecten movimientos que puedan resultar lesivos.

## 2 Estado del arte

La inteligencia artificial (IA) ha tenido un impacto significativo en una amplia variedad de campos, y los deportes no son una excepción. Desde el análisis de rendimiento hasta la mejora de la experiencia del espectador, la IA está revolucionando la manera en que se practican y se disfrutan los deportes.

Este estado del arte se organiza de la siguiente manera: primero se discuten los trabajos previos en deportes en general, luego se abordan específicamente los estudios relacionados con el pádel, y finalmente se revisan los modelos de estimación de pose, incluyendo MoveNet, PoseNet y MediaPipe.

### 2.1 Inteligencia Artificial en deportes

En los últimos años, la inteligencia artificial (IA) ha transformado diversos aspectos del deporte, proporcionando herramientas avanzadas para el análisis de datos, monitoreo de atletas y optimización del rendimiento. La IA ha mejorado la comprensión de las dinámicas del juego, introduciendo métodos para prevenir lesiones y perfeccionar la técnica de los atletas. Permite el análisis en tiempo real de grandes volúmenes de datos generados durante entrenamientos y competiciones, incluyendo información sobre biomecánica, posición y velocidad de jugadores, y patrones estratégicos.

Un campo destacado dentro de la IA aplicada al deporte es la visión artificial, utilizada para capturar y analizar imágenes y videos, proporcionando **información detallada sobre la posición y movimientos de los jugadores**. En deportes como el fútbol y el baloncesto, esta tecnología rastrea el movimiento de jugadores y la trayectoria del balón, mejorando la estrategia y coordinación del equipo. Además, ayuda a corregir errores y mejorar habilidades específicas de los atletas. La IA y la visión artificial también juegan un papel crucial en la prevención de lesiones, identificando patrones de movimientos que podrían causar daños y alertando a entrenadores y médicos para tomar medidas preventivas.

En el estudio "**Implementación de un sistema de control de postura y velocidad en el patinaje de velocidad basado en procesamiento de imágenes**" (González et al., 2023), se desarrolló un sistema avanzado utilizando visión artificial para mejorar la postura y velocidad de los patinadores. La metodología incluyó la captura de imágenes en tiempo real con un dron Dji Tello, seguido de un procesamiento detallado de las mismas para detectar y analizar la postura mediante algoritmos de redes neuronales. Se evaluaron varias metodologías de detección de pose, incluyendo BlazePose, OpenPose y PoseNet, para determinar los puntos clave del cuerpo y construir un esqueleto digital del patinador. Estos puntos clave fueron comparados con un modelo óptimo predefinido para identificar desviaciones y proporcionar retroalimentación en tiempo real.



Figura 2 Imágenes de las fases de detección de postura y ángulos de patinadoras. En cada fase se detectan diferentes posturas y si se realiza de manera correcta, el sistema permite avanzar a la fase siguiente.

El análisis de los resultados mostró una mejora significativa en la postura y la velocidad de los patinadores. La velocidad media aumentó de 7.286 m/s a 7.852 m/s, y la clasificación de la postura mejoró notablemente. Este estudio demuestra la efectividad de las técnicas de visión artificial y redes neuronales en el análisis y optimización del rendimiento deportivo, subrayando el potencial de algoritmos como BlazePose, OpenPose y PoseNet para aplicaciones prácticas en la corrección de técnicas deportivas y la prevención de lesiones.

El artículo "**Martial arts, dancing and sports dataset: A challenging stereo and multi-view dataset for 3D human pose estimation**" (Zhang et al., 2017) presenta el MADS, un nuevo y desafiante conjunto de datos diseñado para la estimación de poses humanas en 3D. Este dataset incluye una variedad de acciones complejas provenientes de artes marciales (Tai-chi y Karate), danza (hip-hop y jazz), y deportes (baloncesto, voleibol, fútbol, rugby, tenis y bádmin-ton). Las acciones fueron realizadas por dos maestros de artes marciales, dos bailarines y un atleta, capturadas mediante múltiples cámaras y una cámara estéreo de profundidad. La riqueza y diversidad del MADS lo convierten en un recurso valioso para la investigación en la estimación de poses humanas, especialmente en entornos donde la variabilidad y la complejidad del movimiento humano son críticas.

En la evaluación del MADS, se utilizaron varios marcos de rastreo, incluyendo trackers generativos basados en filtros de partículas, trackers discriminativos utilizando procesos Gaussianos gemelos, y trackers híbridos como el Personalized Depth Tracker. Los resultados indican que los enfoques discriminativos tienden a tener un mejor desempeño cuando existen suficientes muestras representativas de entrenamiento, mientras que los métodos generativos son más robustos frente a la diversidad de poses.

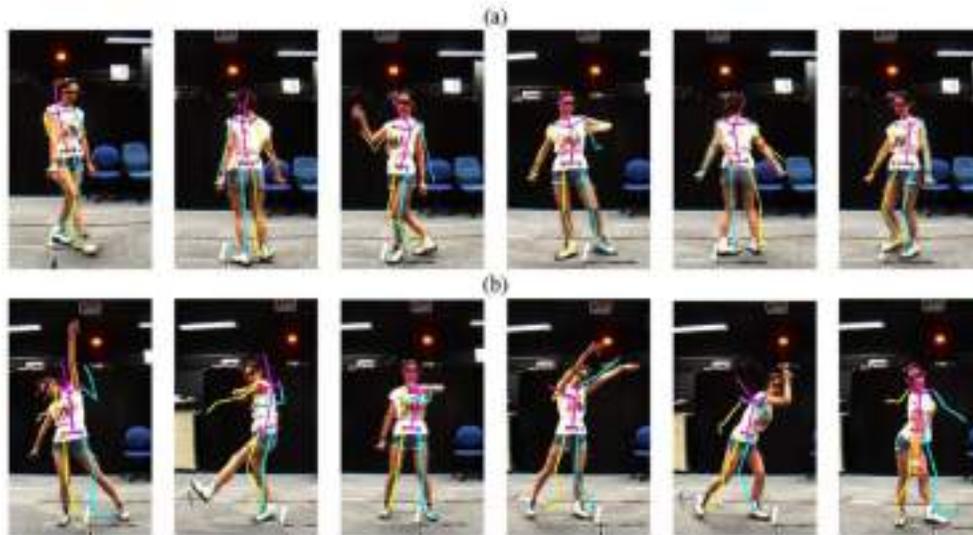


Figura 3 - a) Ejemplo de los resultados de aplicar el método discriminativo TGP (Tracking by Detecting General Pose) en acciones de giro en Jazz. b) Ejemplo de fallos de los resultados de TGP

Otra investigación que se enmarca en el contexto del creciente interés por la aplicación de inteligencia artificial en el ámbito deportivo es la investigación titulada "**Mejora de rendimiento deportivo mediante la detección de posturas por visión artificial**" (Mira Abad, Á., 2023) explora el uso de tecnologías avanzadas de visión artificial para analizar y evaluar posturas y movimientos de atletas, centrándose específicamente en la ejecución de sentadillas. El principal objetivo del trabajo es aplicar técnicas de aprendizaje automático supervisado para mejorar la técnica de las sentadillas. Para ello, se recopiló un conjunto de datos compuesto por vídeos clasificados como correctos o incorrectos en la ejecución de sentadillas. Estos vídeos fueron utilizados para entrenar modelos de clasificación, tales como Máquinas de Vectores de Soporte (SVM), árboles de decisión y regresión logística. Una vez entrenados, estos modelos se aplicaron al análisis de nuevos vídeos con el fin de identificar y corregir errores en la ejecución del ejercicio.

La metodología adoptada implica el uso de la librería MediaPipe para extraer y rastrear puntos clave del cuerpo en cada fotograma de los vídeos. Posteriormente, se calculan los ángulos formados por estos puntos clave, permitiendo evaluar la amplitud de movimiento y la alineación articular durante la realización de las sentadillas. Los resultados del estudio indican que los modelos de clasificación desarrollados son efectivos para identificar ejecuciones incorrectas y proporcionar recomendaciones precisas para la corrección de la técnica.



Figura 4 - Ejemplo de sentadilla clasificada como "bien", donde se visualiza el ángulo de la rodilla, se registran el número de repeticiones, se mide el tiempo por repetición y se identifica el tipo de sentadilla.

El artículo "Design of precise estimation algorithm for sports pose based on multifeature fusion" (Gao Songai et al., 2022) propone un algoritmo avanzado para la estimación precisa de posturas deportivas, combinando técnicas de visión artificial y redes neuronales. La metodología se centra en la construcción de un modelo de barras del cuerpo humano, donde se aplican filtros de Kalman para determinar los grados de libertad y el rango de movimiento de las principales articulaciones. Posteriormente, se utiliza un modelo de ocho estrellas para extraer características clave de la postura deportiva, procesando las imágenes en escala de grises mediante un método de promedio ponderado. La fusión de múltiples características permite un análisis exhaustivo y detallado de la información vectorial, creando un nuevo conjunto de vectores de características fusionadas.

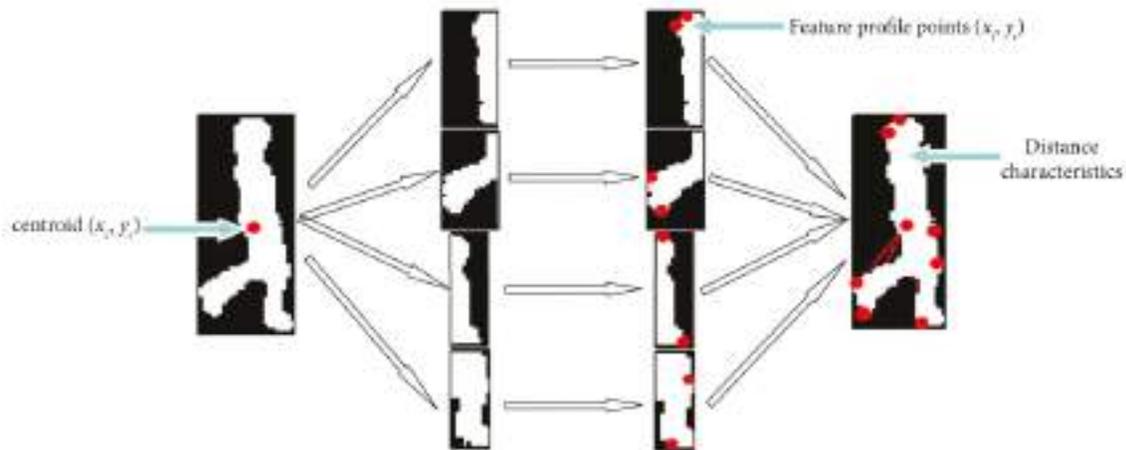


Figura 5 - Proceso de extracción de características de la distancia del polo del contorno local del modelo humano de ocho estrellas.

La precisión y eficiencia de este enfoque se potencian mediante el uso de un modelo de distribución gaussiana mixta para la estimación de la postura del atleta. Este modelo se integra con técnicas de redes neuronales para mejorar la identificación y clasificación de las posturas deportivas, logrando una estimación precisa con una alta eficiencia temporal. Los resultados experimentales muestran que el algoritmo propuesto supera significativamente a los métodos tradicionales, alcanzando una precisión del 100% en un tiempo de estimación de 5.3 segundos para conjuntos de datos de 900GB. Este estudio subraya la efectividad de combinar técnicas de visión artificial con redes neuronales para la optimización del rendimiento deportivo y la prevención de lesiones, ofreciendo una herramienta robusta y precisa para el análisis de posturas en actividades deportivas.

Más allá de los deportes convencionales, en el área del yoga, donde las posiciones del cuerpo son de vital importancia, también se ha investigado el uso de redes neuronales para el análisis de las posturas. El artículo "**YoNet: A neural network for yoga pose classification**" (Ashraf et al., 2023) desarrolla un modelo de red neuronal profunda que emplea convoluciones separables en profundidad para identificar cinco posturas de yoga con alta precisión. Este enfoque ha demostrado ser superior a otros modelos de clasificación de imágenes como ResNet y Xception, logrando una precisión del 94.91% con un conjunto de datos limitado. Esta investigación destaca cómo las redes neuronales pueden mejorar significativamente la precisión en la clasificación de posturas de yoga, facilitando la práctica correcta y la enseñanza de esta disciplina, especialmente en entornos remotos o asistidos por tecnología.

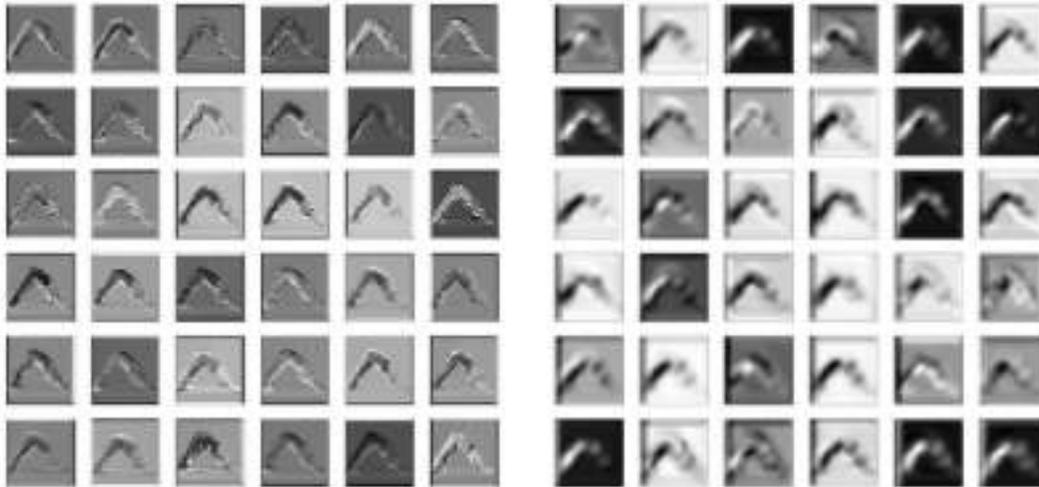
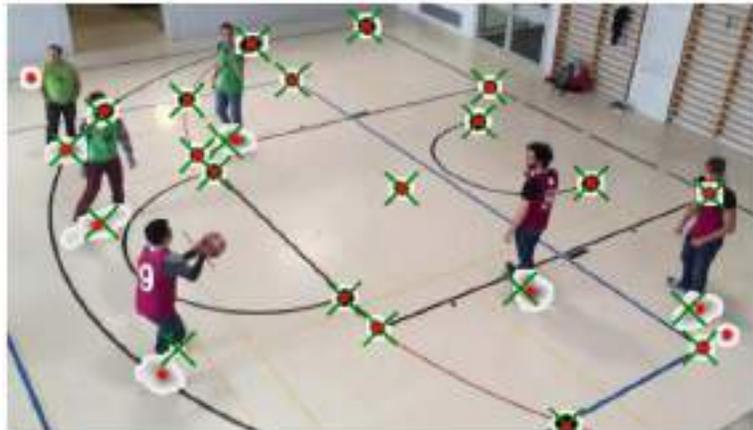


Figura 6 - En esta figura se muestran las salidas de los primeros 36 kernels. a) Mapa de características espaciales. Salida de la última capa Conv2d que extrae principalmente características espaciales de las imágenes. b) Mapa de características de profundidad. Salida de la última capa SeparableConv2d que se centra principalmente en la información de profundidad de las imágenes

La metodología implementada en YoNet se inspira en el modelo Xception, incorporando convoluciones separables en profundidad. Este enfoque permite extraer características espaciales y de profundidad de las imágenes por separado, para luego combinarlas en el proceso de clasificación. La arquitectura del modelo incluye capas de convolución para la extracción de características, normalización por lotes para estabilizar el proceso de aprendizaje y capas densas para la clasificación final. El conjunto de datos utilizado para el entrenamiento y la validación contiene 286 imágenes de cinco posturas de yoga: Adho Mukha, Sukhasana, Tadasana, Virabhadrasana I y Virabhadrasana II.

Además, la revisión "**Advancements in yoga pose estimation using artificial intelligence: A survey**" (Chamola, V. et al., 2024) destaca la importancia de la estimación de poses en tiempo real en yoga, proporcionando retroalimentación correctiva para mejorar la precisión de las posturas y permitiendo la instrucción remota por expertos. Esta revisión clasifica diversas técnicas de aprendizaje automático y visión artificial aplicadas a la estimación de posturas de yoga, destacando los conjuntos de datos y las métricas de evaluación disponibles. En este artículo, se discuten métodos basados en redes neuronales profundas, como las arquitecturas de convolución y las redes de regresión, que se utilizan para identificar y analizar las posturas de yoga a partir de imágenes y secuencias de video. La revisión menciona técnicas de estimación de puntos clave, donde los algoritmos identifican puntos específicos del cuerpo del practicante para evaluar la alineación y la precisión de las posturas. Además, se exploran enfoques de gemelos digitales y modelos de aprendizaje supervisado que ayudan a entrenar sistemas de IA para proporcionar retroalimentación en tiempo real. Esta combinación de tecnologías permite una evaluación detallada y precisa, mejorando tanto la práctica individual como la enseñanza del yoga en entornos remotos.

En el ámbito del análisis deportivo mediante visión artificial, no solo se ha investigado la detección de posturas corporales, sino también la estimación de la posición de la cámara en el área de juego. El artículo **"Real-time camera pose estimation for sports fields"** (Citraro, L. et al., 2020) presenta un enfoque innovador que utiliza una red neuronal convolucional tipo U-Net para detectar puntos clave específicos en las imágenes de campos deportivos, como las intersecciones de las líneas del campo y las posiciones de los jugadores.



*Figura 7 - Imagen de ejemplo con predicción superpuesta. Las cruces indican la verdadera proyección de los puntos de interés y los puntos rojos las ubicaciones 2D encontradas de la red entrenada. Los parches grises/blancos representan la puntuación devuelta por el U-Net. A pesar de que algunos puntos de interés están ocluidos, la red los localiza con precisión.*

A partir de estos puntos clave, se calcula una homografía inicial que mapea el plano de la imagen al plano del suelo, refinada posteriormente con las posiciones de los jugadores para obtener parámetros precisos de la cámara. Este enfoque, combinado con la consistencia temporal mediante un filtro de partículas, mejora significativamente la precisión y la robustez de la estimación de la pose de la cámara en tiempo real, proporcionando una herramienta valiosa para aplicaciones de realidad aumentada y análisis en 3D en deportes.

## 2.2 Inteligencia Artificial en el pádel

Se han revisado los casos de inteligencia artificial (IA) en deportes en general, observando que tiene múltiples aplicaciones muy beneficiosas, como la mejora del rendimiento deportivo, la prevención de lesiones y el análisis táctico. El siguiente apartado se va a centrar en la revisión de la IA en el pádel, un deporte que ha ganado considerable popularidad en los últimos años. Las tecnologías que se van a revisar para plasmar el estado del arte de la inteligencia artificial en el pádel son:

- **GAMETRAQ:** Utiliza cámaras de alta velocidad con resolución 4K para capturar los movimientos de los jugadores y proporcionar análisis en tiempo real del rendimiento del partido. Ofrece métricas detalladas como la distancia recorrida y la posesión de la pelota.
- **PadelCourt:** Emplea visión artificial para filmar partidos con un teléfono móvil, detectando y analizando los movimientos de los jugadores. Genera un esqueleto virtual para evaluar la posición y los tipos de golpes en tiempo real.
- **Aiball:** Desarrollada por HEAD, utiliza cámaras de alta resolución para capturar y analizar los movimientos en tiempo real. Emplea redes neuronales para detectar y seguir los movimientos, proporcionando datos precisos sobre el rendimiento.
- **SwingVision:** Captura vídeos desde un iPhone o iPad y utiliza seguimiento de objetos para rastrear la trayectoria de la pelota en 3D. Genera estadísticas de calidad profesional, resúmenes de partidos y llamadas de línea automáticas, facilitando un análisis integral del rendimiento deportivo.

La tecnología **GAMETRAQ**, desarrollada por la empresa sueca GameCam, representa un avance significativo en la aplicación de IA en el deporte del pádel. GAMETRAQ (Dale, 2023) utiliza cámaras de alta velocidad con resolución 4K para capturar todos los movimientos de los jugadores en la cancha, proporcionando un análisis detallado en tiempo real del rendimiento de los partidos. La tecnología se basa en la detección de puntos clave y el análisis de métricas específicas como la distancia recorrida, la posesión de la pelota, los errores de tiro, y la generación de mapas de calor que muestran las áreas de mayor actividad en la pista de juego.

Los informes generados por la IA ofrecen una visión integral del rendimiento de los jugadores, permitiendo identificar fortalezas y debilidades durante el juego. Estos datos son accesibles a través de plataformas web y móviles, facilitando el acceso a entrenadores y jugadores para mejorar estrategias y técnicas de juego.

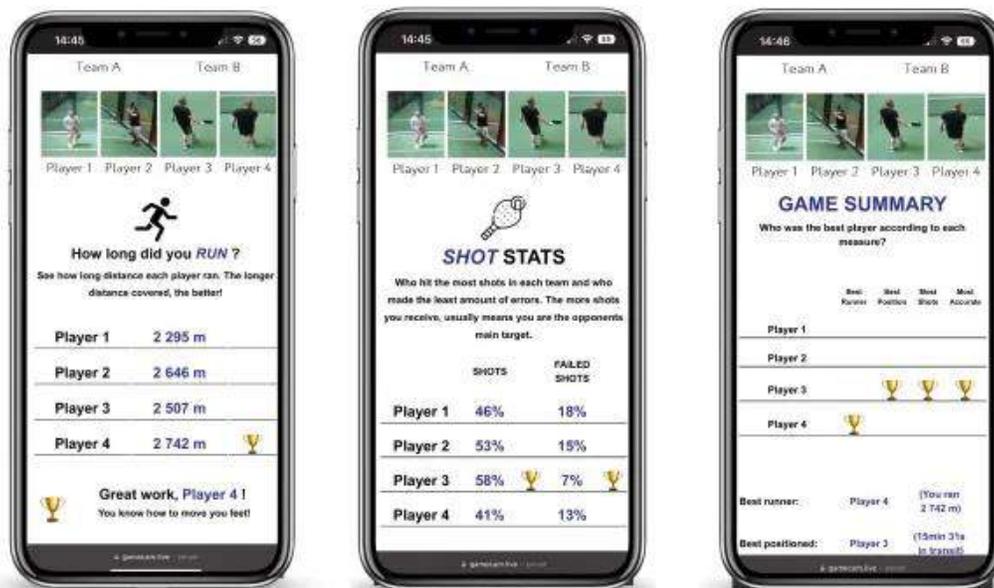


Figura 8 - Aplicación de GAMETRAQ para la visualización del partido de pádel y estadísticas (Dale, 2023)

Aunque GAMETRAQ utiliza cámaras de alta velocidad con resolución 4K y algoritmos de IA para analizar el rendimiento, **no se encontró evidencia específica de que la tecnología realice estimación de poses de los jugadores de pádel**, su enfoque principal está en el análisis estadístico del juego y no específicamente en la estimación de poses detalladas de los jugadores. Además, es una aplicación de pago.

La aplicación **PadelCourt** (Dale, 2023a) utiliza tecnología de visión artificial para mejorar la experiencia del jugador y proporcionar análisis detallados del rendimiento. PadelCourt permite a los jugadores filmar sus partidos con un teléfono móvil y utiliza un sistema de captura de movimiento que crea un esqueleto virtual de cada jugador, permitiendo determinar su posición en la cancha y los tipos de golpes que realizan. Esta tecnología permite un análisis detallado de las poses y movimientos durante el partido.

La aplicación también rastrea la trayectoria de la pelota y mantiene el marcador automáticamente. Al finalizar el partido, analiza los datos recopilados para ofrecer sugerencias de mejora técnica y estratégica, además de generar resúmenes de los mejores momentos y consejos personalizados basados en el rendimiento del jugador.



Figura 9 - Aplicación PadelCourt para el análisis de un partido de pádel que integra la puntuación del partido y la detección de la trayectoria de la pelota (superior). Detección de poses, pelota y pista de juego de un partido de pádel por la aplicación PadelCourt. (inferior) (PadelCourt - Artificial Intelligence For Padel, 2023)

Aunque PadelCourt ofrece avanzadas capacidades de inteligencia artificial para el análisis de juegos de pádel, presenta ciertas limitaciones que podrían mejorarse. Uno de los principales inconvenientes es que la aplicación es de pago y tiene un costo elevado, lo que puede ser una barrera para algunos usuarios. Además, al ser una aplicación de código cerrado, limita la capacidad de personalización y mejora por parte de la comunidad de desarrolladores.

La tecnología **Aiball** (Misha, 2024), desarrollada por HEAD, utiliza cámaras de alta resolución instaladas en las pistas para capturar y analizar en tiempo real los movimientos y las jugadas. Esta tecnología se basa en la inteligencia artificial para proporcionar datos precisos sobre el rendimiento de los jugadores.

Aiball analiza la posició de los jugadores en la pista, cubriendo zonas específicas como la red, transición y fondo. Evalúa la coordinación entre los jugadores para cubrir efectivamente la pista, y clasifica los golpes realizados, proporcionando estadísticas sobre la precisión y velocidad de los saques. Además, permite comparar las estadísticas del jugador con las de profesionales del World Padel Tour, identificando áreas de mejora. La aplicación mide la basculación horizontal y vertical para evaluar la coordinación en pareja y la cobertura de la pista. Aiball ofrece un análisis exhaustivo de los movimientos y estrategias del jugador, proporcionando información valiosa para optimizar el rendimiento y mejorar la técnica en el pádel. A pesar de sus capacidades avanzadas, no se proporciona información detallada sobre los tipos específicos de modelos de inteligencia artificial utilizados para la clasificación de golpes, lo que podría ser relevante para los usuarios interesados en los aspectos técnicos de la solución.



Figura 10 Aplicación de Aiball donde se muestra el análisis de los golpes y del partido de pádel. (Misha, 2024)

Sin embargo, como en todos los casos, Aiball es una aplicación de pago con un coste elevado, lo que limita su accesibilidad a algunos usuarios. Además, también es una aplicación con código cerrado, no permite ni la personalización ni la mejora continua por parte de la comunidad de desarrolladores.

**SwingVision** es una innovadora aplicación impulsada por inteligencia artificial que pretende expandirse más allá del tenis para incluir otros deportes de raqueta como el pádel. Utilizando la captura de vídeo desde un iPhone o iPad, la tecnología de seguimiento de objetos de SwingVision utilizando su tecnología de seguimiento de objetos patentada para rastrear la trayectoria de la pelota en 3D y genera estadísticas de calidad profesional, resúmenes de partidos y llamadas de línea automáticas en tiempo real. Esta tecnología de IA procesa los vídeos capturados para proporcionar datos detallados sobre la velocidad de los golpes, la colocación y la consistencia, además de crear mapas de calor que ayudan a los jugadores a entender mejor el impacto de sus golpes.



Figura 11 - Aplicación de SwingVision donde se muestra el análisis completo del partido de pádel (SwingVision Review: 5 Ways It Can Improve Your Tennis, 2023)

SwingVision se centra principalmente en el análisis de golpes y movimientos de la pelota, proporcionando estadísticas detalladas sobre la velocidad, precisión y ubicación de los golpes. La aplicación rastrea y analiza el rendimiento general del jugador teniendo en cuenta su posición dentro de la pista. Sin embargo, no se menciona específicamente que realice un análisis detallado de la postura del jugador como lo hacen otros modelos de estimación de poses.

En contraste con todas las aplicaciones mencionadas, donde el mayor inconveniente es que todas son de pago (elevado) y son de código cerrado, la aplicación que se propone desarrollar en este trabajo será de código abierto (opensource), lo que permitirá a cualquier desarrollador contribuir a su mejora y personalización según las necesidades específicas de los usuarios. Esto no solo democratizará el acceso a tecnologías avanzadas de análisis deportivo, sino que también fomentará una comunidad activa de desarrolladores que pueden innovar y mejorar continuamente la aplicación.

En resumen, estas tecnologías demuestran cómo la inteligencia artificial está transformando el deporte del pádel, ofreciendo herramientas avanzadas para el análisis del rendimiento y la mejora continua, proporcionando a los jugadores y entrenadores datos valiosos que antes no estaban disponibles.

## 2.3 Modelos de estimación de pose

La estimación de pose es un campo crucial dentro de la visión por computadora y la inteligencia artificial, enfocado en determinar la posición y orientación del cuerpo humano a partir de imágenes o secuencias de video. Este proceso implica la identificación de puntos clave en el cuerpo, como las articulaciones, y su disposición espacial, permitiendo aplicaciones que van desde la animación digital hasta el análisis biomecánico en deportes. En los últimos años, los avances en modelos de aprendizaje profundo han impulsado significativamente la precisión y eficiencia de la estimación de pose, facilitando el desarrollo de sistemas que pueden operar en tiempo real y con alta precisión en condiciones complejas. A continuación, se revisan algunos de los modelos más destacados y sus aplicaciones en diversos ámbitos.

### 2.3.1 MoveNet

MoveNet es un modelo de estimación de pose desarrollado por Google, conocido por su alta precisión y velocidad. Diseñado para aplicaciones en tiempo real, MoveNet puede detectar 17 puntos clave del cuerpo humano en una imagen o vídeo, permitiendo un análisis detallado de la pose en diversas posturas y movimientos. Este modelo es especialmente útil en aplicaciones que requieren una respuesta rápida y precisa, como la monitorización de ejercicios físicos y la interacción hombre-máquina en tiempo real.

MoveNet está implementado en TensorFlow y TensorFlow Lite, lo que facilita su uso en una amplia gama de dispositivos, desde servidores hasta dispositivos móviles. El modelo es de alta precisión y rapidez, y tiene fácil integración. Sin embargo, puede requerir hardware potente para alcanzar su máxima eficiencia en aplicaciones en tiempo real.

### 2.3.2 PoseNet

PoseNet es otro modelo popular para la estimación de pose, también desarrollado por Google. PoseNet puede funcionar tanto en dispositivos móviles como en plataformas web, lo que lo hace accesible para una variedad de aplicaciones. Este modelo puede detectar y localizar 17 puntos clave del cuerpo humano en una imagen, permitiendo a los desarrolladores crear aplicaciones que analicen la postura y el movimiento humano. PoseNet es conocido por su flexibilidad y facilidad de integración en diferentes entornos, desde aplicaciones deportivas hasta interfaces de usuario basadas en gestos.

PoseNet está disponible en TensorFlow.js, TensorFlow y TensorFlow Lite, lo que permite su ejecución en el navegador, en servidores y en dispositivos móviles. El modelo es capaz de operar en tiempo real, tiene flexibilidad en las plataformas de implementación y buena precisión. Sin embargo, puede no ser tan rápido o preciso comparado con modelos más recientes como MoveNet.

### 2.3.3 YOLOv8 pose estimation

La tecnología YOLO (You Only Look Once), conocida por su eficiencia en la detección de objetos en tiempo real, también ha sido adaptada para la estimación de poses humanas, específicamente en aplicaciones deportivas como el pádel. El modelo YOLOv8 de Ultralytics incorpora capacidades avanzadas para la detección de puntos clave del cuerpo, permitiendo un análisis detallado de las poses de los jugadores durante un partido.

YOLOv8 está implementado principalmente en Python y utiliza PyTorch como backend. Esto facilita la integración con otras herramientas de aprendizaje automático y procesamiento de imágenes. YOLOv8 es conocido por su rapidez, capaz de procesar imágenes en tiempo real con una latencia mínima. Esto es crucial en aplicaciones deportivas donde el análisis en tiempo real puede influir directamente en la estrategia y rendimiento de los jugadores.

El modelo YOLOv8-pose (*Finetuning YOLOv8 Pose Estimation For Padel Player Analysis Using trainYOLO*, s. f.) puede identificar múltiples puntos clave del cuerpo humano, lo que permite analizar la postura y los movimientos de los jugadores de pádel. Esto incluye la detección de las articulaciones principales como hombros, codos, caderas y rodillas. Puede ser adaptado y afinado para diferentes escenarios y deportes, lo que aumenta su versatilidad. Sin embargo, para obtener el máximo rendimiento, YOLOv8 puede requerir hardware potente, especialmente para aplicaciones en tiempo real en dispositivos móviles.

### 2.3.4 MediaPipe

MediaPipe, desarrollado por Google, es una plataforma de código abierto que proporciona soluciones de procesamiento multimedia en tiempo real, incluyendo la estimación de pose. El componente de estimación de pose de MediaPipe puede rastrear 33 puntos clave del cuerpo, ofreciendo una representación detallada de la postura humana. MediaPipe es altamente eficiente y está optimizado para funcionar en dispositivos móviles y en el navegador, lo que facilita su uso en aplicaciones de realidad aumentada, análisis deportivo y monitorización de salud.

Además, MediaPipe proporciona herramientas adicionales para el procesamiento de datos, como la detección de manos y el reconocimiento facial, lo que amplía su aplicabilidad. Permite combinar múltiples módulos (como BlazePose, BlazeFace y BlazeHand) para soluciones complejas.

MediaPipe está diseñado para integrarse fácilmente con TensorFlow, OpenCV y otras librerías de procesamiento multimedia. Tiene alta precisión con los 33 puntos clave, capacidad de integración con otras herramientas de procesamiento multimedia y eficiencia en tiempo real.

En su documentación (*Pose Landmark Detection Guide*, s. f.) muestra ejemplos de integración en Android, Python y entornos web, ofreciendo incluso una webapp para probar las capacidades del modelo ([https://mediapipe-studio.webapps.google.com/demo/pose\\_landmarker](https://mediapipe-studio.webapps.google.com/demo/pose_landmarker)).

El modelo de MediaPipe Pose Landmarker rastrea 33 puntos clave en el cuerpo humano, representando la ubicación aproximada de diversas partes. Los resultados del modelo incluyen coordenadas normalizadas (Landmarks) en 2D y coordenadas mundiales (WorldLandmarks) en 3D para cada punto de referencia, proporcionando una representación detallada y precisa de la postura y los movimientos del cuerpo humano en diferentes contextos. En la siguiente figura aparecen representados y listados los 33 puntos clave:

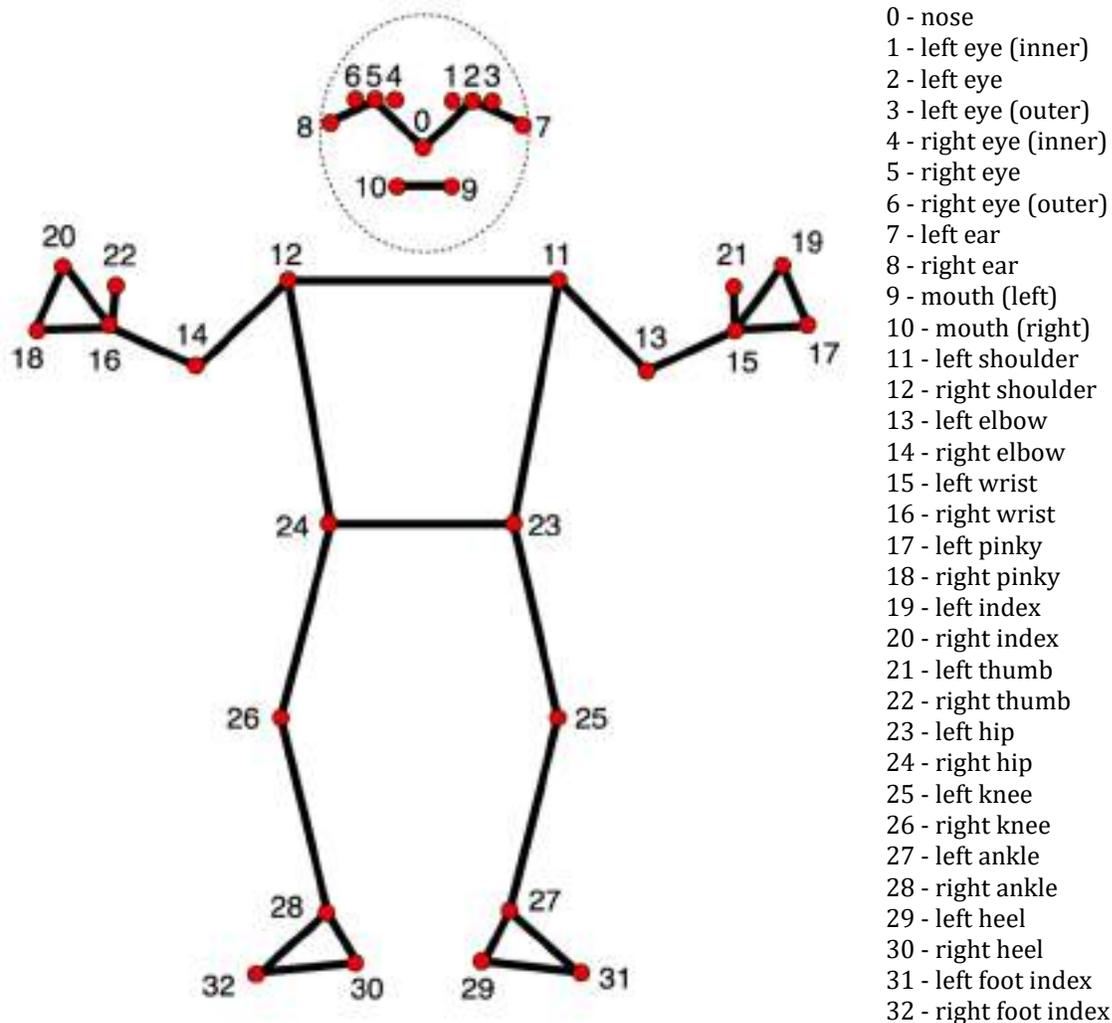


Figura 12 - esquema y listado de landmarks de MediaPipe

En la propia descripción de la librería, destacan como casos de uso la cuantificación de ejercicios físicos, mostrando incluso ejemplos visuales de ejercicios de yoga o de fitness en general (Google-Ai-Edge, s. f.).

En la documentación, Google destaca su eficiencia en tiempo real y su orientación a dispositivos móviles e incluso ejecución en web.

#### 2.3.4.1 BlazePose

BlazePose (*On-device, Real-time Body Pose Tracking With MediaPipe BlazePose*, s. f.) es un modelo de estimación de pose específico desarrollado por Google que se integra dentro del marco más amplio de MediaPipe, una plataforma de código abierto para el procesamiento multimedia en tiempo real. MediaPipe proporciona una infraestructura general y herramientas para implementar soluciones de visión por computadora, mientras que BlazePose es uno de los módulos específicos que se ejecuta dentro de esta infraestructura.

Este modelo está centrado exclusivamente en la estimación de poses humanas, detectando 33 puntos clave en 2D de un cuerpo humano a partir de un solo fotograma.

En combinación con GHUM (Generalized Human Model), un pipeline de modelado humano tridimensional, BlazePose es capaz de estimar la pose completa en 3D con coordenadas X,Y,Z a partir de imágenes o vídeos. BlazePose GHUM (*GHUM & GHUML: Generative 3D Human Shape And Articulated Pose Models*, 2020) mejora la precisión y el detalle del seguimiento de poses, ofreciendo una perspectiva más completa y realista del movimiento humano en 3D. Esta versión es ideal para aplicaciones que requieren un análisis más profundo de la biomecánica y la dinámica del cuerpo.

Actualmente, BlazePose GHUM 3D es la herramienta de detección de poses de Google en su ML Kit (Pose Detection, s. f.).

En este Trabajo Final de Máster se va a utilizar MediaPipe Pose, que es una solución basada en ML para seguimiento de pose y segmentación/enmascarado del fondo a partir de frames RGB de vídeo, utilizando la investigación desarrollada en BlazePose. La elección de BlazePose se debe a su optimización para dispositivos móviles y su capacidad de proporcionar un análisis detallado y preciso de la postura, lo que es crucial para aplicaciones de seguimiento y mejora del rendimiento en deportes como el pádel.

## 3 Metodología

### 3.1 Dataset

En este proyecto, se desarrolla un dataset propio específico para pádel en base a muestras de vídeos, dada la escasa disponibilidad de muestras publicadas.

Esto no solo optimizará la exactitud y eficacia de los modelos de IA para la estimación de poses y movimientos, sino que también contribuirá a llenar un vacío en la literatura y los recursos disponibles para futuros trabajos en este campo.

Un dataset propio presenta varias ventajas significativas:

1. Especificidad:
  - Un dataset propio permitirá capturar las dinámicas específicas y las particularidades del movimiento en pádel, mejorando la precisión y relevancia del modelo entrenado.
2. Personalización y adaptabilidad:
  - La capacidad de personalizar el dataset según las necesidades exactas del proyecto permite la inclusión de escenarios, condiciones y variaciones específicas que son relevantes para el análisis de pádel.
3. Mejor evaluación y validación:
  - Tener un dataset propio facilita la creación de conjuntos de datos de entrenamiento y test que reflejan de manera más precisa las situaciones reales que los jugadores de pádel encuentran, mejorando la evaluación del rendimiento del modelo.
4. Control de calidad:
  - Al desarrollar un dataset propio, se puede garantizar la calidad de las anotaciones y la coherencia de los datos, lo que es crucial para el entrenamiento de modelos de alta precisión.

Para crear el dataset, se recogen muestras de los distintos golpes a clasificar a partir de vídeos públicos de la plataforma Youtube.

El objetivo es conseguir vídeos del golpe realizados por distintos jugadores y capturados desde distintos ángulos, con una duración de 60 frames (2 segundos). Se recoge para ello material de varios vídeos sobre evaluaciones de palas en los que se prueban realizando distintos tipos de golpe o tutoriales de técnica.

Si bien la frecuencia de cada tipo de golpe no suele ser la misma durante el desarrollo de un partido, se crea un dataset balanceado para facilitar la creación y evaluación de los modelos, con un total de 300 muestras repartidas equitativamente entre las 6 clases.

El recuento de las muestras y las clases se representa en la siguiente tabla.

Golpe	Muestras
Derecha	50
Revés	50
Bandeja	50
Volea de revés	50
Volea de derecha	50
Remate	50

Tabla 1

El dataset está accesible públicamente en el repositorio de Hugging Face, en concreto en el siguiente enlace: <https://huggingface.co/datasets/josmazna/padelpose>

En la documentación se detalla la información que contiene, así como algunas funciones de ejemplo para acceder a la información de una detección o para acceder al dataset completo de forma análoga a como se cargan los datos para el entrenamiento de este proyecto.

### 3.1.1 Datos de entrada:

Los datos o características de cada vídeo son secuencias de keypoints, es decir, tramas que contienen la posición de las articulaciones de la persona en cada instante del tiempo.

Los principales retos que presentan este tipo de datos son la variación que pueden presentar por cambios de perspectiva o de la velocidad a la que se realice el movimiento. El mismo movimiento exacto puede presentar secuencias de keypoints con valores completamente diferentes si el vídeo es capturado desde posiciones diferentes. Del mismo modo, incluso desde la misma perspectiva, la secuencia estará más comprimida o será más larga en base a la velocidad a la que se haya realizado el movimiento.

Como se detalla en el punto 2.3.3, MediaPipe dispone de un pose landmarker basado en 33 landmarks que describen la posición de varias articulaciones.

Dichos landmarks son obtenidos mediante las funciones de detección de pose landmarks, que devuelven los resultados como poseLandmarkerResult con el siguiente formato:

```
PoseLandmarkerResult:
  Landmarks:
    Landmark #0:
      x      : 0.638852
      y      : 0.671197
      z      : 0.129959
      visibility : 0.9999997615814209
      presence  : 0.9999984502792358
    Landmark #1:
      x      : 0.634599
      y      : 0.536441
      z      : -0.06984
```

```

visibility : 0.999909
presence   : 0.999958
... (33 landmarks per pose)
WorldLandmarks:
Landmark #0:
x          : 0.067485
y          : 0.031084
z          : 0.055223
visibility : 0.9999997615814209
presence   : 0.9999984502792358
Landmark #1:
x          : 0.063209
y          : -0.00382
z          : 0.020920
visibility : 0.999976
presence   : 0.999998
... (33 world landmarks per pose)
    
```

Como se puede apreciar, la salida de la detección contiene tanto coordenadas normalizadas (Landmarks) como coordenadas globales (WorldLandmarks) para cada landmark.

Por un lado, los Landmarks contienen las siguientes coordenadas normalizadas:

- x, y: Coordenadas del punto de referencia normalizadas entre 0.0 y 1.0 por el ancho (x) y la altura (y) de la imagen.
- z: Es la profundidad del punto de referencia, con la profundidad en el punto medio de las caderas como origen. Cuanto menor sea el valor, más cerca estará el punto de referencia de la cámara. La magnitud de z usa más o menos la misma escala que x.
- visibility: Es la probabilidad de que el punto de referencia sea visible dentro de la imagen.

Estas coordenadas son, por tanto, muy dependientes de la posición de la persona dentro del frame de la imagen o de la perspectiva desde la que se capture el movimiento, ya que utilizan la propia imagen como sistema de referencia.

Por otra parte, los WorldLandmarks contienen información estimada sobre la posición global de cada landmark, usando en este caso a la persona como referencia y con las unidades en metros.

- x, y, z: Coordenadas tridimensionales del mundo real en metros, con el centro de las caderas como origen.
- visibility: Es la probabilidad de que el punto de referencia sea visible dentro de la imagen.

En resumen, para cada frame que se procese, se obtendrán sus landmarks 2D normalizadas al tamaño de la imagen de entrada así como las WorldLandmarks en 3D con las posiciones en coordenadas globales, como aparece representado en la siguiente figura.

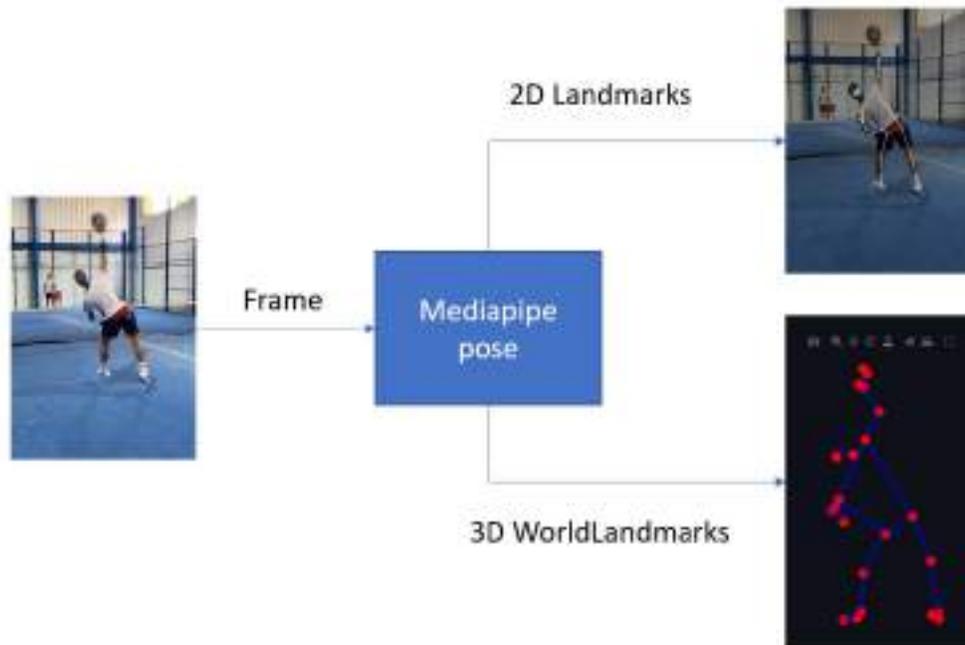


Figura 13 – Representación de los resultados en 2D y 3D obtenidos a partir de un frame de ejemplo

Como se ha descrito en el punto anterior, el dataset consiste en vídeos etiquetados, a los que se aplica el modelo de detección de pose de MediaPipe para la extracción de las secuencias de keypoints.

Para cada muestra del conjunto de datos se almacena el resultado de las detecciones (poseLandmarkerResult), que consiste en una secuencia de N frames con los Landmarks y WorldLandmarks para cada frame. A su vez, cada objeto de landmarks incluye, para cada uno de los 33 keypoints, las 3 coordenadas X, Y, Z.

Por lo tanto, extrayendo la información de los WorldLandmarks a partir de las detecciones, el tamaño de la matriz de características tras procesar los vídeos tendrá un tamaño de

$$[N\_samples, 60 (N\_frames), 33 (N\_keypoints), 3 (coordenadas x,y,z)]$$

### 3.1.2 Datos de salida

El dato de salida es el tipo de golpe estimado por el modelo en base a su aprendizaje previo.

A continuación, se enumeran y describen los golpes seleccionados:

- **Drive:**

Descripción: Golpe básico y de ataque que se realiza con la pala a la altura de la cintura o ligeramente más arriba. Se usa tanto en golpes de fondo como de aproximación a la red.

Técnica: Se realiza con un movimiento fluido y continuo, buscando profundidad y control en la pelota.

- **Revés:**

Descripción: Golpe básico que se ejecuta del lado opuesto al drive, generalmente con una mano.

Técnica: Similar al drive, se busca mantener la pala firme y realizar un swing controlado para mantener la pelota en juego con precisión.

- **Volea de derecha:**

Descripción: Golpe ofensivo que se realiza cerca de la red, antes de que la pelota bote en el suelo.

Técnica: Se golpea con la pala cerca de la red, con un movimiento corto y controlado para mantener la pelota baja y rápida.

- **Volea de revés:**

Descripción: Similar a la volea de derecha, pero del lado opuesto. Es crucial en el juego en la red.

Técnica: Se usa un movimiento corto y controlado, manteniendo la pala firme y cerca de la red para darle velocidad y precisión a la pelota.

- **Bandeja:**

Descripción: Golpe defensivo y de control, utilizado para mantener la posición en la red después de un globo del rival.

Técnica: Se ejecuta con un movimiento de arriba hacia abajo, con la pala alta, buscando colocar la pelota en el fondo de la cancha rival con efecto para complicar la devolución.

- **Remate:**

Descripción: Golpe ofensivo y potente que se realiza por encima de la cabeza, similar a una volea alta.

Técnica: Se busca imprimir la máxima potencia y precisión para que la pelota bote fuera de la pista o sea difícil de devolver, generalmente se utiliza para terminar el punto.

Cada uno de los vídeos del dataset está etiquetado con una de esas 6 clases. Los 3 primeros caracteres del nombre de cada fichero especifican la clase, siguiendo la codificación mostrada en la siguiente tabla.

Etiqueta (prefijo del fichero)	Nombre completo del golpe
der	Derecha
rev	Revés
vde	Volea de derecha
vre	Volea de revés
ban	Bandeja
rem	Remate

Tabla 2 – codificación de las etiquetas de los golpes

### 3.1.3 Análisis exploratorio de los datos de entrada

Como parte del análisis exploratorio de los datos, se realiza un ejercicio de clustering sobre los mismos.

En primer lugar, se separan los datos de cada una de las 6 clases, obteniendo 50 muestras en cada clase con su secuencia de coordenadas para cada keypoints.

Sobre estas muestras de cada clase se realiza un clustering empleando la librería KMeans de scikit learn para obtener un único cluster de cada una de ellas. A partir de este cluster, se extrae su centro y se visualiza haciendo modificaciones a las funciones de visualización 3D de la librería mediapipe para obtener una animación de la secuencia de keypoints.

En la siguiente figura se puede visualizar uno de los frames de la animación generada para visualizar el centro del cluster de la clase “Revés”.

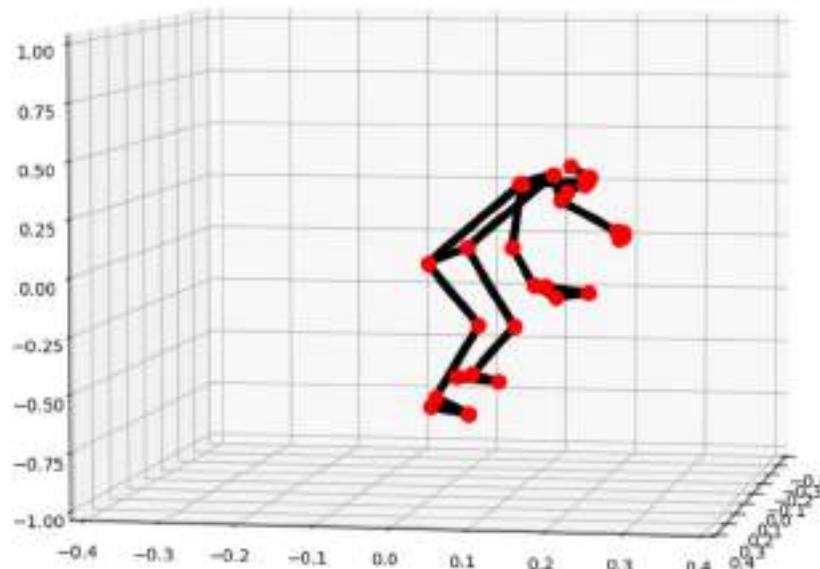


Figura 14 – Ejemplo de visualización de un frame de la animación del centro del cluster de la clase “Revés”

Se podría utilizar este concepto del centro de cluster de cada clase para generar una referencia del movimiento maestro y, por ejemplo, medir distancias de cada una de las nuevas muestras con respecto a este movimiento para evaluar su utilización como métrica de calidad del movimiento en cuanto a la técnica.

También puede ser interesante para encontrar distintas agrupaciones dentro de movimientos dentro de un mismo tipo de golpe. Por ejemplo, el movimiento que realiza el jugador en la volea puede cambiar en función de la altura a la que le llegue la pelota o la preparación para el golpe de derechas o de revés cambia si el jugador decide dejar entrar la pelota para impactar cuando ésta salga rebotada del cristal de fondo.

Por tanto, podría emplearse este mismo concepto de clustering dentro de las muestras de una misma clase, incrementando el número de clases de una sola a varias, para intentar capturar estos subgrupos de movimientos dentro de una misma clase que podrían emplearse incluso para mejorar la clasificación de forma más detallada en estos subtipos de golpe.

De forma exploratoria, también se realiza un clustering sobre todo el conjunto de datos conteniendo todas las clases, de nuevo con KMeans y con 6 clases.



Figura 15 – Visualización de componentes principales de PCA con etiquetas de clusters de KMeans

A priori se aprecia que los 6 clusters no terminan de estar claramente diferenciados sino que la representación muestra nubes de puntos continuas divididas en varias clases (como la 1 y la 5) o nubes de un mismo cluster dispersas (como la 4).

Para poder comparar con las etiquetas reales, se grafican también sobre este espacio reducido por PCA las etiquetas reales, como se muestra en la siguiente figura.

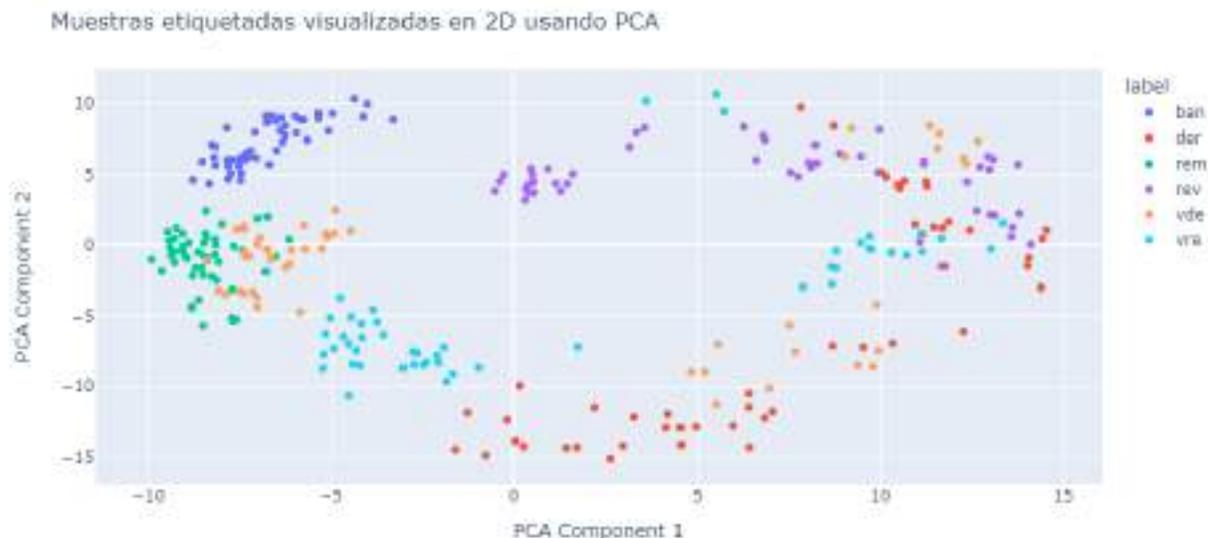


Figura 16 - Visualización de componentes principales de PCA con etiquetas reales de las muestras

De esta gráfica podemos concluir que la reducción de dimensionalidad del PCA sobre estas secuencias de frames no es capaz de diferenciar claramente sus características, ya que hay clases bastante dispersas como la vde (volea de derecha) o zonas con distintas clases en distancia cercanas, como la parte superior derecha que contiene muestras de 4 clases distintas entremezcladas.

Resulta interesante, pese a la escasa diferenciación en general que consigue sobre las clases, que los golpes de remate y bandeja (en la esquina superior izquierda) se mantengan cerca entre ellos, indicando que hay similitud entre ambos golpes altos y que presentan unos movimientos de preparación del golpe diferenciados del resto de clases.

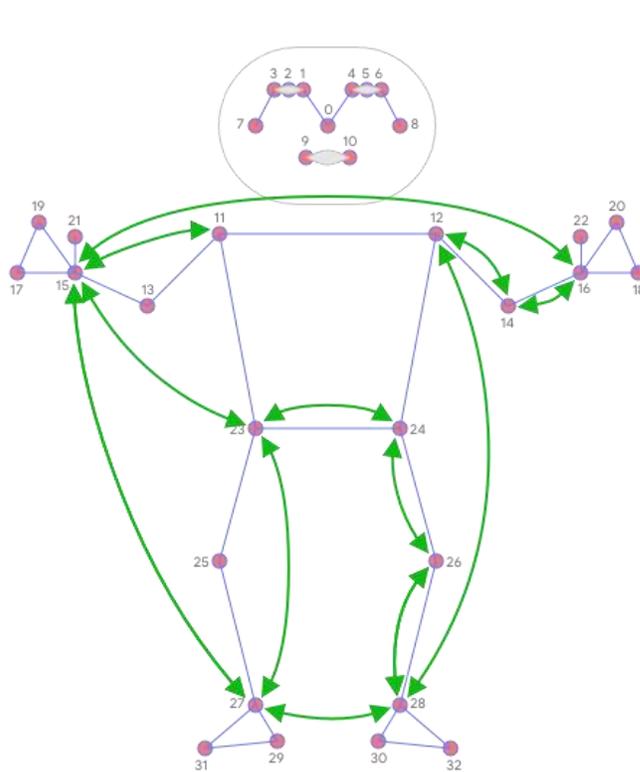
### 3.1.4 Transformaciones sobre los datos de entrada

Las coordenadas globales (WorldLandmarks) obtenidas a partir del algoritmo de detección de pose de MediaPipe pueden presentar distintas complicaciones para la tarea de clasificación. Por ejemplo, dos personas realizando un movimiento exactamente idéntico podrían tener dos representaciones completamente diferentes de sus coordenadas sólo con rotar su esqueleto sobre el eje vertical (Z).

Se han considerado para este proyecto dos transformaciones de dichas coordenadas para convertirlas en otras características que representen la pose en cada instante de tiempo, como son los ángulos entre articulaciones y las distancias entre articulaciones.

### 3.1.4.1 Conversión de coordenadas a distancias:

La idea principal es obtener una representación de la pose en base a distancias entre articulaciones. A partir de cada landmark de detecciones XYZ, se genera un embedding de distancias que contiene las siguientes distancias entre articulaciones medidas en cada uno de los 3 ejes:



- ('left\_shoulder', 'left\_elbow'),
- ('right\_shoulder', 'right\_elbow'),
- ('left\_elbow', 'left\_wrist'),
- ('right\_elbow', 'right\_wrist'),
- ('left\_hip', 'left\_knee'),
- ('right\_hip', 'right\_knee'),
- ('left\_knee', 'left\_ankle'),
- ('right\_knee', 'right\_ankle'),
- ('left\_shoulder', 'left\_wrist'),
- ('right\_shoulder', 'right\_wrist'),
- ('left\_hip', 'left\_ankle'),
- ('right\_hip', 'right\_ankle'),
- ('left\_hip', 'left\_wrist'),
- ('right\_hip', 'right\_wrist'),
- ('left\_shoulder', 'left\_ankle'),
- ('right\_shoulder', 'right\_ankle'),
- ('left\_hip', 'left\_wrist'),
- ('right\_hip', 'right\_wrist'),
- ('left\_elbow', 'right\_elbow'),
- ('left\_knee', 'right\_knee'),
- ('left\_wrist', 'right\_wrist'),
- ('left\_ankle', 'right\_ankle'),

Figura 17 – Representación gráfica de algunas distancias y listado de las 22 distancias procesadas entre articulaciones

La entrada se reduce, por tanto, de un vector de características de 33x3 (coordenadas de keypoints en 3 dimensiones) a un vector de 22x3 (distancias entre keypoints en 3 dimensiones). (Google-Ai-Edge, s. f.-a)

A su vez, la matriz de características pasa a tener unas dimensiones de

$$[N\_samples, 60 (N\_frames), 22 (N\_distancias), 3 (ejes x,y,z)]$$

3.1.4.2 Conversión de coordenadas a ángulos:

En su investigación **Fitness Action Recognition Based on MediaPipe** (Xu et al., 2023), en la extracción de características explican la necesidad de convertir las coordenadas absolutas en ángulos y proponen un conjunto de 11 ángulos que representan con fiabilidad la dinámica del movimiento corporal.

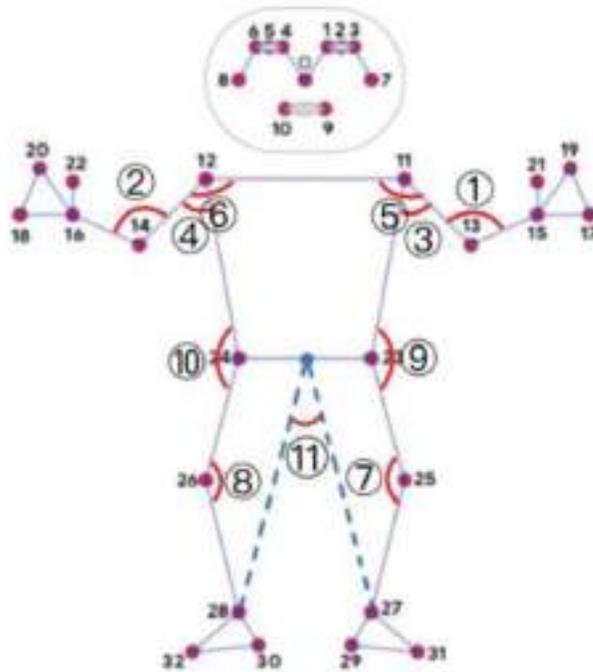


Figura 18 – Esquema gráfico de los ángulos

Índice	Ángulo	Keypoint 1	Keypoint 2	Keypoint 3
1	left_elbow_angle	11	13	15
2	right_elbow_angle	12	14	16
3	left_shoulder_angle	13	11	23
4	right_shoulder_angle	14	13	24
5	left_chest_angle	12	11	13
6	right_chest_angle	11	12	14
7	left_knee_angle	23	25	27
8	right_knee_angle	24	26	28
9	left_waist_angle	11	23	25
10	right_waist_angle	12	24	26

Tabla 3 – Listado de 10 ángulos con los índices de las articulaciones correspondientes

Aplicando el cálculo de los ángulos sobre el espacio tridimensional se obtiene una reducción de dimensionalidad de 33x3 (coordenadas de keypoints en 3 dimensiones) a un vector 10 ángulos.

La matriz de características del dataset pasa a tener unas dimensiones de

$$[N\_samples, 60 (N\_frames), 10 (N\_ángulos)]$$

## 3.2 Modelado del clasificador

Para obtener una predicción del tipo de golpe en base a las características extraídas del vídeo, es necesario disponer de un modelo clasificador que aprenda en base a ejemplos y sea capaz de entender las diferencias entre ellos, de forma que sea capaz de generalizar y predecir sobre nuevas muestras.

### 3.2.1 Modelos de clasificación simples

En primera instancia se entrenan modelos sencillos de clasificación basados en muestras unidimensionales.

Para ello, es necesario alisar el vector de entrada para que pase de tener

[N\_samples, 60 (N\_frames), 33 (N\_keypoints), 3 (coordenadas x,y,z)]

a tener

[N\_samples, N\_características]

Los modelos simples probados para evaluar su rendimiento en la clasificación de golpes de pádel son los siguientes:

- Regresión Logística

La regresión logística es un modelo estadístico que utiliza una función logística para modelar una variable dependiente categórica. Implementada en la biblioteca `sklearn.linear_model.LogisticRegression`, este modelo es eficaz para problemas de clasificación binaria y puede extenderse a multiclase. La regresión logística estima las probabilidades de las clases como una función logística de una combinación lineal de las características del conjunto de datos. Este modelo es útil por su interpretabilidad y eficiencia en problemas de clasificación sencillos.

- Random Forest

El Random Forest es un método de ensamble implementado en la biblioteca `sklearn.ensemble.RandomForestClassifier`. Este modelo construye múltiples árboles de decisión durante el entrenamiento y emite la clase que es el modo de las clases de los árboles individuales. Cada árbol se construye a partir de una muestra aleatoria con reemplazo del conjunto de entrenamiento y utiliza un subconjunto aleatorio de características para dividir cada nodo. Random Forest es adecuado para tareas de clasificación y regresión y es especialmente efectivo en conjuntos de datos grandes y con muchas características debido a su capacidad para manejar la variabilidad y reducir el sobreajuste.

- XGBoost

XGBoost (Extreme Gradient Boosting) es una implementación optimizada del algoritmo de gradient boosting, disponible en la biblioteca `xgboost`. Este modelo construye el modelo de forma secuencial, donde cada nuevo árbol intenta corregir los errores cometidos por los árboles anteriores. XGBoost incluye regularización para prevenir el sobreajuste, y se destaca por su velocidad y rendimiento en tareas de

clasificación y regresión. Es particularmente popular en competiciones de ciencia de datos debido a su eficacia y precisión en la mejora de las predicciones mediante el refinamiento iterativo del modelo.

### 3.2.2 Modelo basado en CNN y LSTM

Por último, se crea un modelo más complejo basado en LSTM con una CNN. Con esta red se busca combinar la capacidad de extracción de características espaciales de las CNN con la capacidad de modelado temporal de las RNN.

Las capas de convolución permiten la ingesta del dato con su tamaño original de [60 (N\_frames), 33 (N\_keypoints), 3 (coordenadas x,y,z)].

La definición de la arquitectura y su resumen son los siguientes:

```
# Definir la entrada
input_shape = (60, 33, 3) # (frames, keypoints, coordinates)
inputs = Input(shape=input_shape)

# CNN para cada frame
x = TimeDistributed(Conv2D(32, (3, 3), activation='relu'))(inputs)
x = TimeDistributed(MaxPooling2D((2, 2)))(x)
x = TimeDistributed(Flatten())(x)

# Remodelar para la entrada a LSTM
x = Reshape((60, -1))(x)

# LSTM para capturar la dependencia temporal
x = LSTM(128, return_sequences=False)(x)

# Capas densas para la clasificación
x = Dense(64, activation='relu')(x)
outputs = Dense(num_classes, activation='softmax')(x)
```

Layer (type)	Output Shape	Param #
input_layer_3 (InputLayer)	(None, 60, 33, 3)	0
time_distributed_9 (TimeDistributed)	(None, 60, 33, 3, 1)	0
time_distributed_10 (TimeDistributed)	(None, 60, 31, 3, 32)	128
time_distributed_11 (TimeDistributed)	(None, 60, 15, 3, 32)	0
time_distributed_12 (TimeDistributed)	(None, 60, 1440)	0
reshape_5 (Reshape)	(None, 60, 1440)	0
lstm_2 (LSTM)	(None, 128)	803,328
dense_4 (Dense)	(None, 64)	8,256
dense_5 (Dense)	(None, 6)	390

Explicación de cada componente:

- Entrada: (60, 33, 3) representa 60 frames, cada uno con 33 keypoints y 3 coordenadas (X, Y, Z).
- TimeDistributed Conv2D: Aplica una capa convolucional a cada frame individualmente.
  - TimeDistributed envuelve la capa Conv2D para aplicar la convolución a cada frame por separado.
  - Conv2D(32, (3, 3), activation='relu') aplica 32 filtros de convolución de tamaño 3x3.
- TimeDistributed MaxPooling2D: Realiza max pooling en cada frame.
  - MaxPooling2D((2, 2)) reduce las dimensiones espaciales (keypoints).
- TimeDistributed Flatten: Aplana la salida de cada frame individualmente.
- Reshape: Cambia la forma del tensor para que pueda ser procesado por la capa LSTM. Aquí, el tensor se remodela a (60, features), donde features es el número de características después de las operaciones convolucionales y de aplanado.
- LSTM: Capa LSTM de 128 unidades para capturar dependencias temporales entre frames.
- Dense Layers: Capas densas para la clasificación.
  - Dense(64, activation='relu') añade una capa densa con 64 unidades.
  - Dense(num\_classes, activation='softmax') añade la capa de salida con un número de unidades igual al número de clases (num\_classes) y softmax para clasificación multiclase.

### 3.2.3 Creación de modelos

#### 3.2.3.1 Extracción de características a partir de los vídeos etiquetados del dataset.

En primer lugar, es necesario extraer las características del dataset y adaptar su formato de entrada al modelo. Este paso es diferente para los distintos tipos de características analizadas, por lo que se detallan a continuación:

- **Coordenadas:**

Si la entrada al modelo son las coordenadas extraídas de los vídeos (detalladas en 3.1.1), el dataset se construye a partir de la función auxiliar `build_coordinates_dataset_from_videos` desarrollada en el fichero de `data_prep`.

Dicha función recorre la carpeta del dataset, busca todos los vídeos y, para cada uno de ellos, llama a la función `get_detection`.

Para mejorar la eficiencia, la función `get_detection` se ha desarrollado con la opción de almacenar los resultados de la detección de poses en un fichero pkl mediante la librería `joblib`, manteniendo el mismo formato original del `poseLandmarkerResult`.

En caso de que existan resultados previamente guardados, la función los recuperará y los devolverá, evitando replicar el procesamiento. En caso de que no existan registros previos, procesará el vídeo frame a frame, empleando la función `detect_for_video` del objeto `landmarker` para generar las detecciones.

De esta forma:

- Para cada vídeo, se extrae su etiqueta junto con sus detecciones.
- Las detecciones se recorren frame a frame para extraer los landmarks.
- Para cada landmark, se extraen las coordenadas  $x, y, z$ .

Cada uno de los vectores  $X, Y, Z$  que devuelve la función `build_coordinates_dataset_from_videos` tiene por tanto una longitud de  $(227, 60, 33)$  [muestras, frames, keyframes] y el conjunto de las 3 coordenadas da como resultado  $(227, 60, 33, 3)$ .

- **Distancias:**

En este caso, para emplear las distancias como características de entrada como se describe en el apartado 3.1.4.1, se emplea la función `build_distances_dataset_from_videos`, que busca los vídeos del dataset y llama a la función `get_distances_from_video`. Dentro de esta función se obtienen las detecciones de forma idéntica a como se hacía en el caso de las coordenadas, con la función `get_detections`. La diferencia es que, en lugar de devolver dichas detecciones, se pasan a la función `get_distances_from_detection` para obtener los embeddings de distancias para cada vídeo.

Esta función `get_distances_from_detection` recibe el embedding de distancias para cada frame del vídeo de la función `get_pose_distance_embedding`, en la que se llama en sucesivas ocasiones a una función

auxiliar que obtiene la distancia entre landmarks especificando el par de nombres de los landmarks objetivo.

De esta forma, se obtienen 22 características (distancias) en los 3 ejes X, Y, Z para cada frame de cada detección.

La matriz de distancias tiene, por tanto, unas dimensiones de (227, 60, 22, 3).

- **Ángulos:**

De forma análoga al punto anterior, para aplicar la transformación de coordenadas a ángulos como se detalla en el apartado 3.1.4.2, se construye una función `build_angles_dataset_from_videos` que replica la obtención de coordenadas para llamar a la función `get_angles_from_detection` sobre cada una de las detecciones de keypoints.

Dentro de esa función, para las detecciones de cada frame, se calcula el ángulo entre conjuntos de 3 keypoints definidos en un diccionario que contiene 10 valores. En este caso el ángulo entre los 3 puntos pasa a ser un elemento unidimensional, por lo que se reduce la dimensionalidad de 3 a 1.

La matriz que se obtiene de esta función de procesamiento tiene unas dimensiones de (225, 60, 10).

### 3.2.3.2 Transformación de las dimensiones de entrada

Antes de pasar las coordenadas al modelo, se modifica su formato de entrada mediante un reshape para que pase de (227, 60, 33, 3) a (227, 5940) para ser compatibles con los modelos de clasificación.

Para habilitar la matriz de distancias a los modelos de clasificación, se modifica su tamaño de (227, 60, 22, 3) a (227, 3960).

En el caso de los ángulos como descriptores, se aplana el formato de (227, 60, 10) a (227, 600).

### 3.2.3.3 Separación entre set de entrenamiento y test

La separación de los datos en conjuntos de entrenamiento y test es esencial para evaluar la capacidad del modelo para generalizar a datos no vistos. En este proceso, se utiliza la función `train_test_split` del módulo `sklearn.model_selection`, dividiendo los datos en un 80% para entrenamiento y un 20% para test.

Se utiliza el parámetro `stratify` para asegurar que cada clase mantiene la misma proporción de muestras, con lo que se obtienen siempre 40 ejemplos para el set de entrenamiento y 10 para el de test.

#### 3.2.3.4 Ajuste del modelo

Una vez preparados los datasets de entrenamiento y test, se procede al entrenamiento de los modelos de clasificación.

Se han dividido en dos subconjuntos, como se detallaba en el apartado 3.2:

- Por una parte, se entrenan los 3 algoritmos definidos en el apartado 3.2.1: RandomForestClassifier, LogisticRegression y XGBoost. Cada uno de ellos se entrena con
  - Las coordenadas extraídas de los vídeos (detalladas en 3.1.1)
  - Las distancias entre coordenadas (detalladas en 3.1.4.1)
  - Los ángulos entre coordenadas (detallados en 3.1.4.2)
- Por otra parte, se entrena el modelo con LSTM detallado en el punto 3.2.2. sobre las coordenadas extraídas de los vídeos con su tamaño de entrada original.

#### 3.2.3.5 Evaluación de métricas sobre el set de entrenamiento y de test

Como último paso de la fase de creación de modelos, se evalúan las métricas de desempeño de los mismos tanto en el conjunto de entrenamiento como en el conjunto de test.

Se desarrolla una función “evaluate model” que toma como entradas las etiquetas reales junto con las etiquetas de la inferencia del modelo, en la que se procesan los siguientes indicadores:

- Accuracy (exactitud): Es la proporción de predicciones correctas sobre el total de predicciones. Es una métrica simple y fácil de interpretar.
- Precision (precisión): Proporción de verdaderos positivos sobre el total de positivos predichos (verdaderos positivos + falsos positivos).
- Recall (exhaustividad): Proporción de verdaderos positivos sobre el total de positivos reales (verdaderos positivos + falsos negativos).
- F1-Score: Media armónica de precisión y exhaustividad, utilizada para balancear las dos métricas.

Además, se obtiene la matriz de confusión junto con el reporte de clasificación para los resultados a analizar.

Los resultados se detallan en el apartado 4.

### 3.3 Despliegue del modelo en una webapp

Con el objetivo de habilitar la predicción del modelo de clasificación en base a nuevas muestras, se habilita una aplicación web (webapp), a la que se puede acceder desde el siguiente enlace:

<https://padelpose.streamlit.app/>

#### **Infraestructura:**

El framework elegido para la aplicación es Streamlit, basado en Python y orientado al desarrollo de soluciones interactivas para proyectos de inteligencia artificial.

La aplicación se puede ejecutar de forma local, permitiendo su desarrollo de forma muy ágil, aunque su acceso queda restringido a aquellos equipos que puedan alcanzar el equipo que está alojando la aplicación. En el caso de estar en una red local, los equipos de esa misma red pueden acceder a la aplicación simplemente abriendo en un navegador la dirección y el puerto en el que se lance.

Para desplegar la aplicación de forma pública, se ha optado por Streamlit Cloud.

Streamlit Cloud es una plataforma especializada en la creación y despliegue de aplicaciones web desarrolladas en streamlit, permitiendo su acceso global simplemente enlazando el repositorio que contiene el código.

Desde una perspectiva técnica, Streamlit Cloud proporciona un entorno muy práctico para ejecutar y compartir aplicaciones web sin necesidad de desplegar la infraestructura necesaria, con algunas funciones básicas como estadísticas de acceso o cierto manejo de usuarios. En su versión gratuita, los recursos de los que dispone la aplicación son limitados; sin embargo, permiten compartir pruebas de concepto de forma muy ágil.

Para una solución escalable y robusta en cuanto a infraestructura, streamlit puede desplegarse sobre infraestructura genérica como podrían ser instancias de EC2 de AWS, con su consiguiente gestión de permisos de acceso y costes asociados. También existen plataformas especializadas en despliegue de webapps, como Posit Connect, que soportan streamlit de forma nativa.

**Funcionalidad:**

El flujo de información dentro de la aplicación se resume en la siguiente figura:

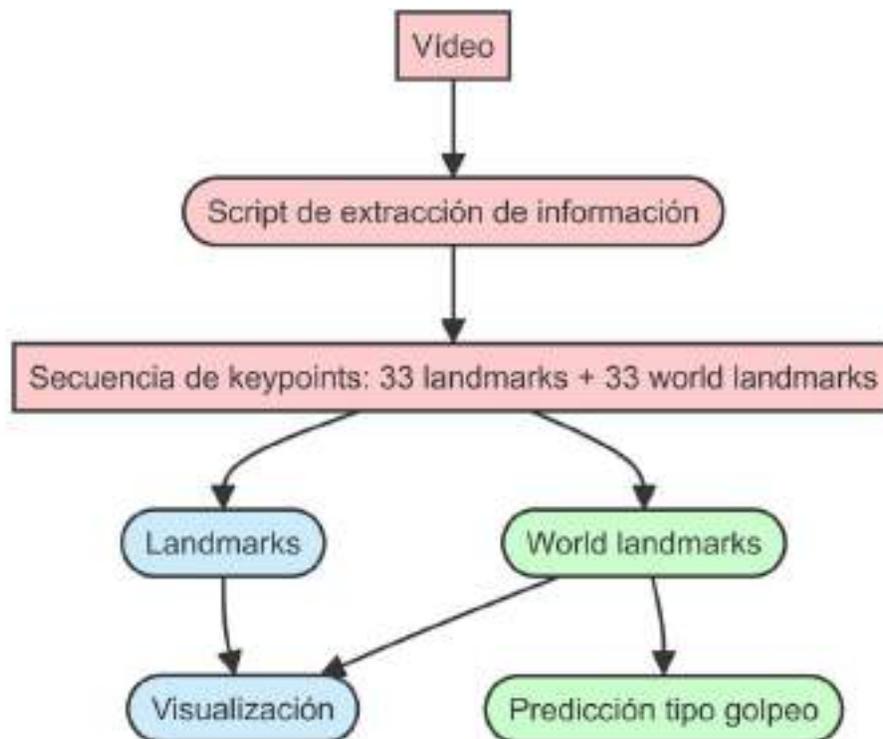


Figura 19 – Flujo de información desde la entrada (vídeo) hasta la salida (visualizaciones y predicción)

Para detallar las funcionalidades que soportan dicho flujo, se detallan sus componentes en las siguientes partes: ingesta de vídeo, procesado e inferencia, feedback y visualización.

3.3.1 Ingesta de vídeo

Al acceder a la aplicación, el usuario encuentra un widget como el mostrado en la siguiente figura, que le permite subir un vídeo desde su equipo (dispositivo móvil o PC).

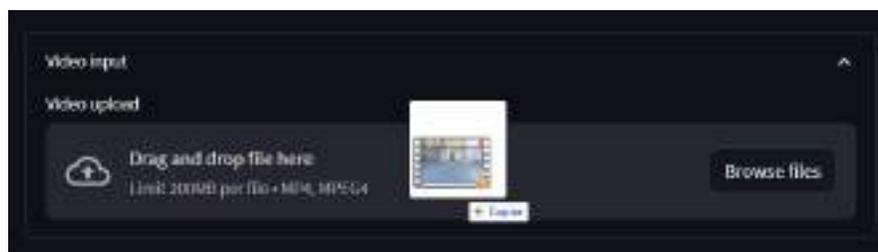


Figura 20 – subida de video a la aplicación en su versión de PC

La aplicació soporta vídeos en horitzontal o vertical, a distintes resolucions, donant resultats positius inclús amb vídeos de baixa qualitat. En l'anexo 7.2 se detallen els resultats de diverses proves realitzades amb diferents tipus de vídeo d'entrada.

Per adequar el vídeo a les condicions dels models entrenats, el primer pas és ajustar la duració a 60 frames (2 segons a 30fps). En cas de que la longitud siga inferior a 60 frames, se mostrarà un error indicant que no es pot processar el vídeo.

Per facilitar al usuari este recorte i garantir que el vídeo recortat recoja el moviment, se mostra una imatge en pantalla junt amb un deslizador que permet ajustar l'inici del vídeo, de forma que el usuari pot ir desplaçant fàcilment este selector per que la imatge mostre al jugador impactant la pilota.

En la següent figura se mostra un exemple en el que se ha desplaçat el deslizador fins al frame 41 per trobar el instant en el que el jugador està golpejant la pilota.

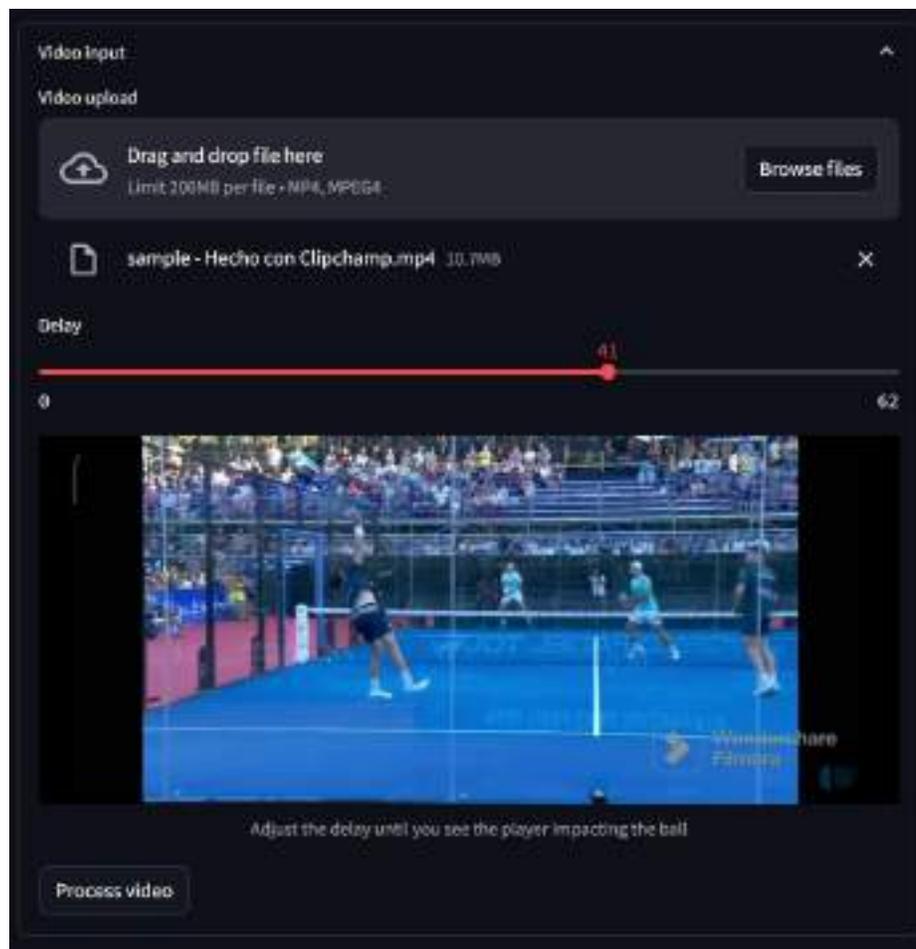


Figura 21 – Exemple de l'ajuste amb el deslizador fins a visualitzar al jugador golpejant la pilota

Una vegada ajustat el selector per que el vídeo recortat recoja el moviment correctament, l'usuari prem el botó "Process video", el vídeo recortat se almacena com un fitxer temporal en l'aplicació i comença la fase de processament.

### 3.3.2 Procesado e inferencia

El vídeo se almacena como un fichero temporal en la aplicación, permitiendo su procesado frame a frame para extraer y almacenar en memoria la información necesaria.

Cada frame, se procesa y almacena:

- El resultado de la detección de pose: los landmarks que contienen las posiciones de las articulaciones.
- La imagen anotada, que es la combinación del frame original con la información superpuesta de las coordenadas 2D de las articulaciones detectadas y las uniones entre ellas, que conforman un esqueleto.
- La figura en 3D con las coordenadas estimadas.

Por lo tanto, al terminar de procesar el vídeo, los resultados son:

- La secuencia de detecciones de articulaciones para cada uno de los frames.
- La secuencia de imágenes anotadas, que conforman un vídeo de salida.
- La secuencia de figuras del esqueleto 3D, que puede ser visualizada posteriormente.

Para obtener la predicción del tipo de golpe, se pasa la secuencia de detecciones de las articulaciones como entrada al modelo entrenado previamente para obtener una inferencia.

Se aplica el modelo de regresión logística aplicada directamente a las coordenadas extraídas de MediaPipe, dados sus buenos resultados (detallados en el apartado 4) y su sencillez, ya que el procesamiento resulta extremadamente eficiente para la webapp desplegada.

La salida del modelo es el tipo de golpe al que estima que corresponden los keypoints, codificada de forma numérica, que se decodifica para mostrar al usuario de forma intuitiva, tal y como se muestra en la siguiente figura.

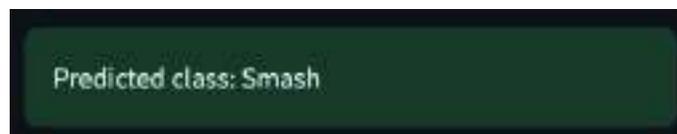


Figura 22 – Ejemplo de la visualización de la clase inferida por el modelo

Además de la clase más probable, el modelo devuelve también la probabilidad para cada una de las clases, que el usuario puede consultar en el desplegable “Prediction probability breakdown” que aparece en la siguiente figura:



Figura 23 – Ejemplo de la distribución de probabilidades para un remate

### 3.3.3 Feedback

Con el fin de robustecer el sistema y proporcionar un flujo cíclico de mejora, detallado en el apartado 1.2, se habilita el complemento mostrado en la siguiente figura para que el usuario pueda aportar su valoración.

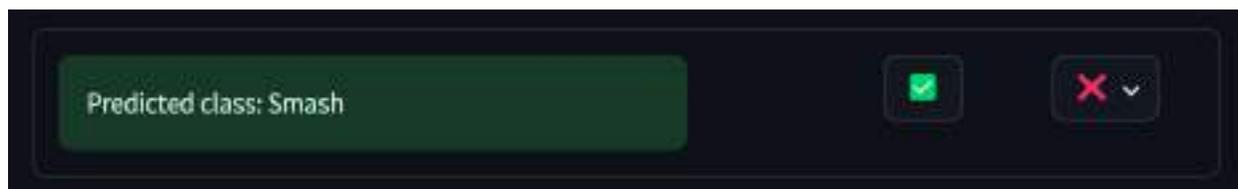


Figura 24 – Visualización de la clase inferida junto con los botones de feedback a su derecha

A pesar de que los resultados de evaluación son en general muy buenos, es posible que el modelo prediga de forma errónea el golpe realizado en algún vídeo; en ese caso, el usuario tiene disponible el botón **X**, que abre un menú, visible en la siguiente figura, en el cual puede seleccionar el golpe correcto entre las clases disponibles (todas excepto la que el modelo ha predicho).

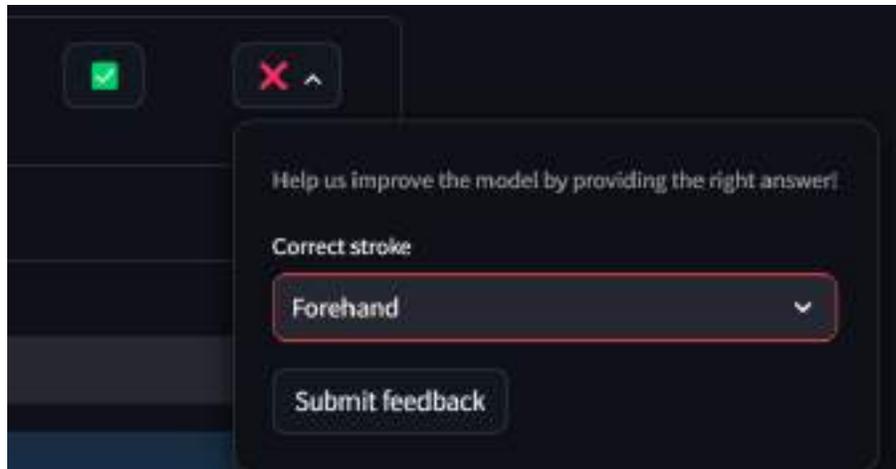


Figura 25 – Opciones de feedback negativo, con desplegable para sugerir el golpe correcto

También tiene disponible el botón  para corroborar que la predicción ofrecida por el modelo es correcta. Aunque a priori pueda parecer información menos valiosa, también tiene utilidad al ayudar a robustecer al modelo y enriquecer el dataset aportando variabilidad a las muestras iniciales.

Cuando el feedback es enviado a la webapp a través de cualquiera de los dos botones, se mantiene visible la opción seleccionada en la interfaz, aunque desactivada, ya que la información ya ha sido almacenada. En la siguiente figura se muestra

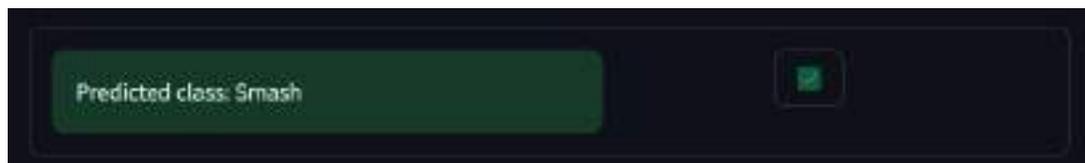


Figura 26 – Ejemplo de visualización tras feedback positivo

Internamente, el sistema almacena la siguiente información:

- Resultado de la detección (poseLandmarkerResult) con el mismo formato que en el dataset original
- Feedback del usuario (positivo/negativo)
- Tipo de golpe
- Fecha y hora

Esta información no se incorpora de forma automática al dataset sino que se deja a la disposición del desarrollador para su supervisión. En caso de que se visualicen los resultados de las detecciones y se confirme que el feedback ofrecido por los usuarios es correcto (o bien que se confíe ciegamente en el criterio de los usuarios), se podría añadir con total facilidad al dataset de origen con simplemente copiar y pegar los ficheros almacenados de la carpeta de feedback a la carpeta del dataset.

### 3.3.4 Visualizaciones adicionales

En la línea con los puntos expuestos en los posibles usos prácticos de la introducción del proyecto, la aplicación muestra algunas herramientas de visualización que puedan resultar de utilidad para analizar y mejorar la técnica.

Se han desarrollado las siguientes como ejemplos de la información que se puede extraer en base a las coordenadas extraídas por el detector de pose, preparando funciones específicas de procesamiento y generación de figuras 3D interactivas:

#### Vídeo anotado

Permite visualizar el esqueleto de forma sencilla e intuitiva, superpuesto al vídeo que se muestra en semitransparencia para resaltar los resultados de la detección, como se puede apreciar en la siguiente figura.



Figura 27 – Ejemplo del video anotado con el esqueleto del jugador superpuesto

#### Visualización de la pose en 3D:

Ofrece la posibilidad de visualizar la estimación de la pose tridimensional, almacenada en el momento de procesar el vídeo como una secuencia de figuras tridimensionales basadas en plotly express.

La visualización es interactiva y permite al usuario rotar la vista y alejar o acercar la figura, así como ver las coordenadas de cada una de las articulaciones, por lo que se puede revisar en detalle la información.

El instante de tiempo para el que se representa la pose puede ser seleccionado desde la barra superior; también se puede visualizar una animación de la secuencia completa de frames mediante el botón Animate. A continuación, se muestra una figura con un ejemplo de esta visualización:



Figura 28 – Visualizador 3D para análisis de las poses en cada frame

### Seguimiento de articulaciones:

Ofrece al usuario la posibilidad de seleccionar una o varias articulaciones y de visualizar su posición de forma detallada, tanto en vídeo (figura 27) como en gráficas 3D donde se visualiza un seguimiento de la posición en cada instante (figura 28).

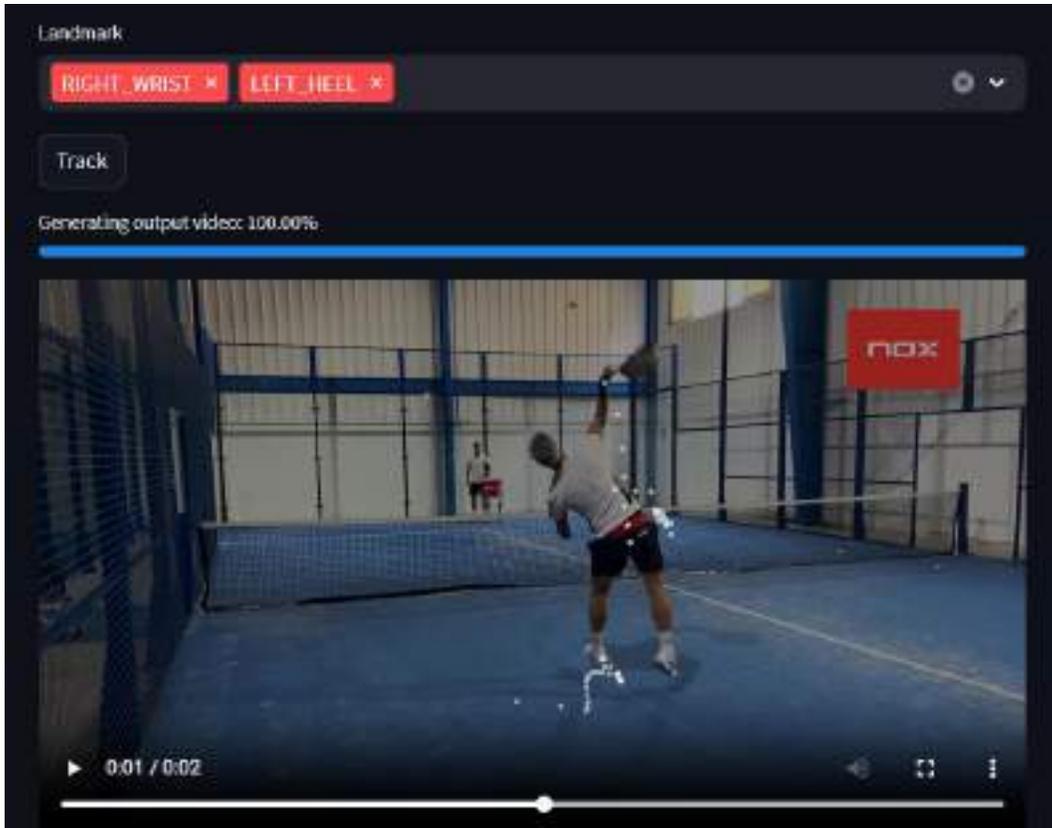


Figura 29 – Vídeo con la superposición de las articulaciones seleccionadas (muñeca derecha y talón izquierdo)

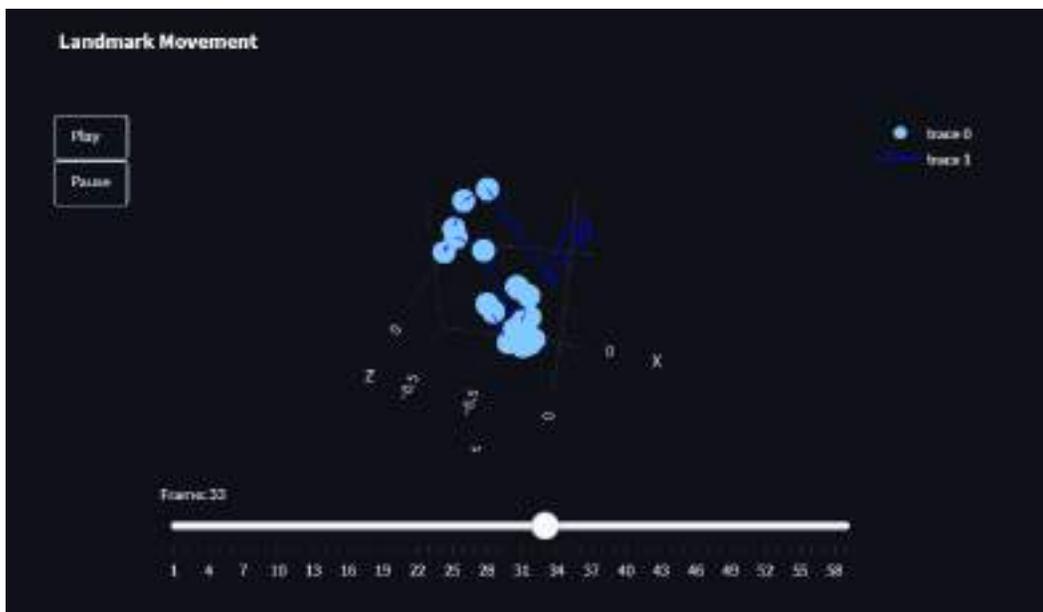


Figura 30 – Visualización 3D de la evolución de la posición de una articulación seleccionada

## 4 Resultados

En el desarrollo del proyecto se han completado los objetivos y cumplido con los resultados esperados descritos en los apartados introductorios 1.2 y 1.3.

En concreto, en este apartado se destacan los resultados obtenidos en la creación del dataset, del modelo clasificador y, finalmente, de la webapp.

### **Extracción de características y creación del dataset:**

Se ha creado un dataset de 300 muestras de vídeo etiquetadas con los 6 tipos de golpe a analizar y se han publicado las secuencias de poses procesadas con MediaPipe para cada una de las muestras, contribuyendo con ello a facilitar avances en el estado del arte.

Durante la extracción de características de los vídeos, alrededor del **X%** de muestras presentaron algún problema de detección por parte del algoritmo de MediaPipe, por lo que tuvieron que ser desechadas. Es esperable que en la aplicación desplegada también se produzcan errores en la fase de extracción de poses, por lo que se lista una posible acción de mejora en los trabajos futuros del apartado 5.2.

### **Creación de la webapp y experiencia de usuario:**

Se ha creado la webapp como plataforma para habilitar el flujo de información entre el usuario, el modelo y cerrar el ciclo de vuelta al dataset de entrenamiento a través de un ciclo de feedback que permita mejorar las inferencias futuras.

La webapp resulta ágil e intuitiva tanto en PC como en dispositivos móviles de cualquier gama, ya que el procesamiento se lleva a cabo del lado del servidor de la aplicación.

Se han realizado pruebas con vídeos de distintas resoluciones y calidades, grabados en entornos interiores y al aire libre, con resultados en general muy positivos.

También se ha testado la aplicación con vídeos generados a partir de grabaciones de pantalla desde un dispositivo móvil con resultados satisfactorios.

El algoritmo de detección de poses de MediaPipe juega un papel crucial para ofrecer esta versatilidad, ya que abstrae al modelo desarrollado de la complejidad y variabilidad de los vídeos de entrada.

Para probar la robustez de la detección y probar sus límites, se han llevado diversas pruebas con vídeos de baja resolución, con fondos complejos o el sujeto alejado.

En el anexo 7.2 se detallan los resultados de varias de estas pruebas con capturas de la webapp.

**Modelos clasificadores:**

A continuación, se muestran los resultados de los clasificadores entrenados.

En todos los casos se alcanza un 100% en las métricas durante el entrenamiento, lo cual indica que las características de entrada son suficientes para diferenciar los distintos tipos de golpes.

Aunque el modelo sea capaz de inferir perfectamente sobre el set de datos que ha utilizado durante el entrenamiento, interesa evaluar su rendimiento sobre el set de test para analizar su capacidad de generalización y evitar modelos sobreajustados a los datos de entrada.

Por ello, se detallan a continuación las métricas sobre el set de test, que contiene exclusivamente muestras que no han sido utilizadas durante la fase de entrenamiento.

En el anexo se pueden consultar los informes de evaluación completos obtenidos para cada escenario (cada modelo con sus sets de entrenamiento y test), que contienen las métricas definidas en el apartado 3.2.3.5: accuracy, precisión, recall, F1-score y el reporte de clasificación.

Como métrica principal para evaluar y comparar los distintos escenarios, se utiliza el F1-score, ya que destaca como una métrica de resumen robusta y equilibrada para evaluar clasificadores. Su capacidad para balancear precisión y recall lo convierte en una opción preferida frente a otras métricas que pueden no captar adecuadamente el rendimiento del modelo en contextos específicos.

Los resultados de cada uno de los 3 modelos de clasificación simples sobre los conjuntos de evaluación de las distintas características de entrada son los siguientes:

F1-Score – Test Set	Logistic Regression	Random Forest	XGBoost
Coordenadas	100.00%	93.25%	96.73%
Distancias	98.33%	100.00%	98.33%
Ángulos	100.00%	98.33%	93.29%

*Tabla 4 – Resumen de indicador F1 para cada modelo entrenado*

Matrices de confusión – Set de evaluación – Datos de entrada: coordenadas

Logistic Regression – Test Set – Confusion Matrix

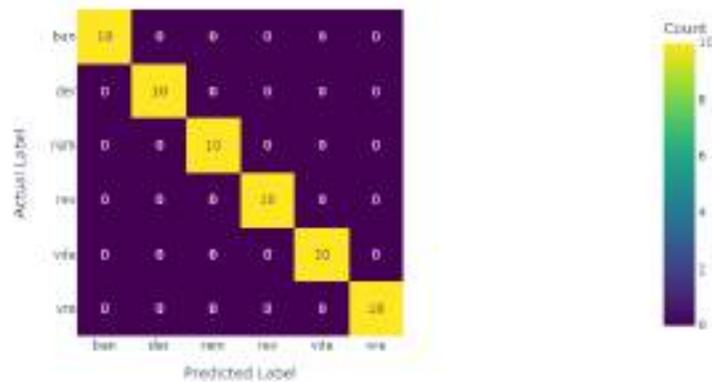


Figura 31 - Matriz de confusión del set de test para el modelo Logistic Regression con coordenadas

Random Forest – Test Set – Confusion Matrix

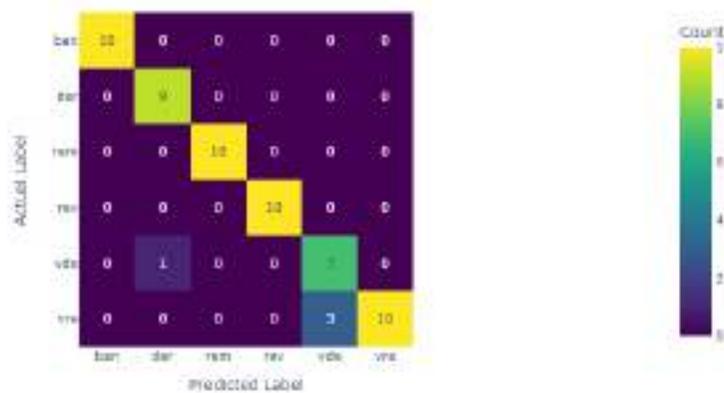


Figura 32 - Matriz de confusión del set de test para el modelo Random Forest con coordenadas

XGBoost – Test Set – Confusion Matrix

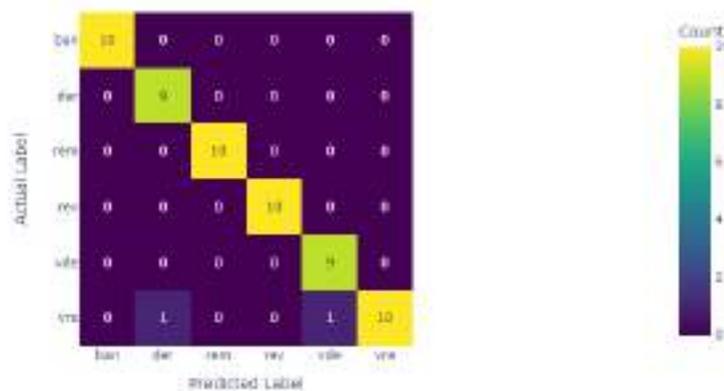


Figura 33 - Matriz de confusión del set de test para el modelo XGBoost con coordenadas

Matrices de confusión – Set de evaluación – Datos de entrada: distancias

Distancias - Logistic Regression - Test Set - Confusion Matrix

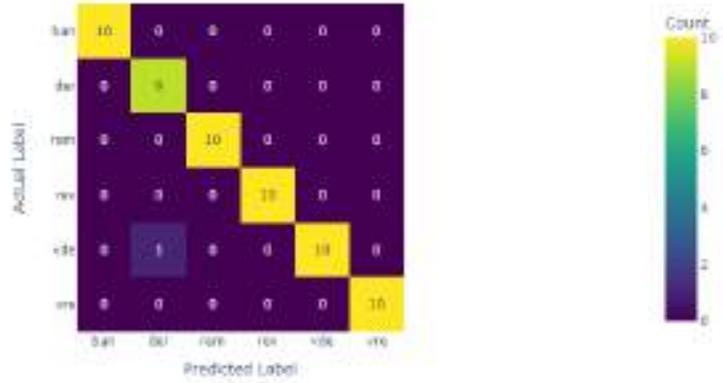


Figura 34 Matriz de confusión del set de test para el modelo Logistic Regression con distancias

Distancias - Random Forest - Test Set - Confusion Matrix

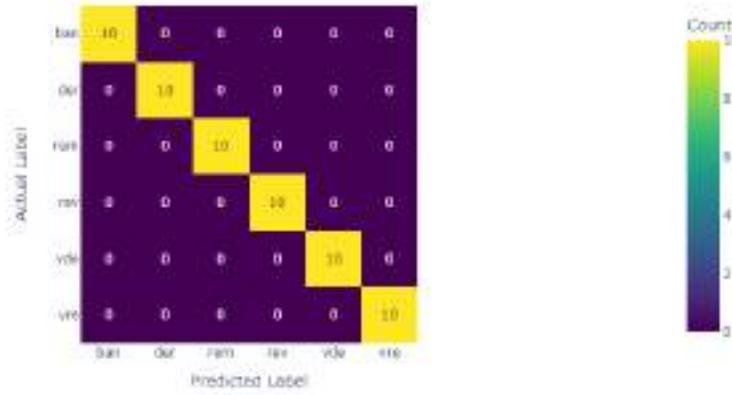


Figura 35 Matriz de confusión del set de test para el modelo Random Forest con distancias

Distancias - XGBoost - Test Set - Confusion Matrix

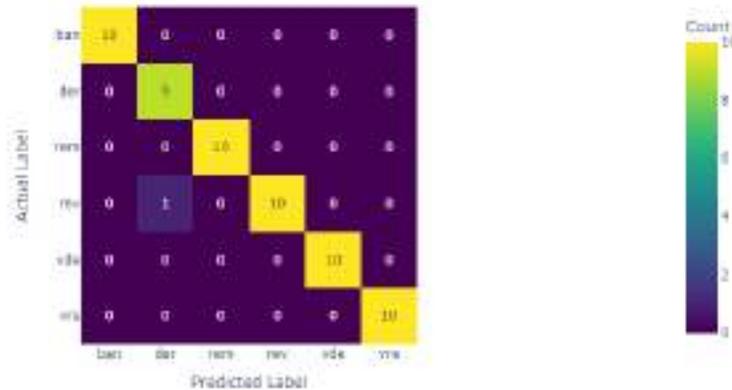


Figura 36 - Matriz de confusión del set de test para el modelo XGBoost con distancias

Matrices de confusión – Set de evaluación – Datos de entrada: ángulos

Angles - Logistic Regression - Test Set - Confusion Matrix:

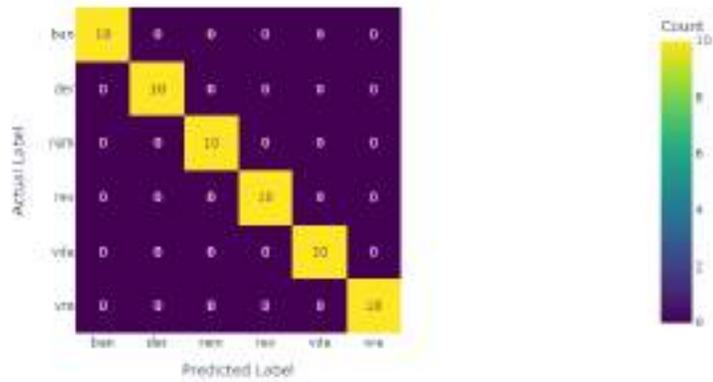


Figura 37 - Matriz de confusión del set de test para el modelo Logistic Regression con ángulos

Angles - Random Forest - Test Set - Confusion Matrix:

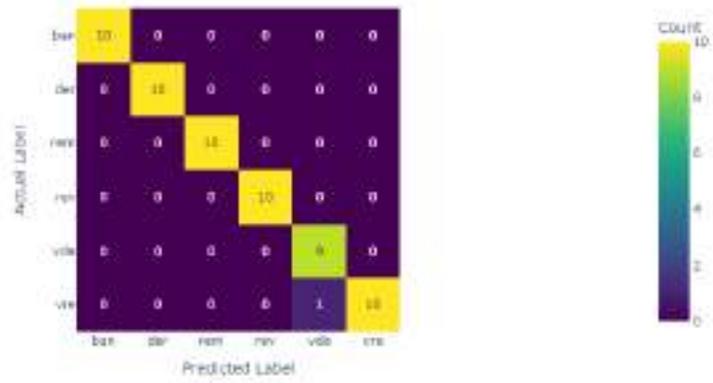


Figura 38 Matriz de confusión del set de test para el modelo Random Forest con ángulos

Angles - XGBoost - Test Set - Confusion Matrix:

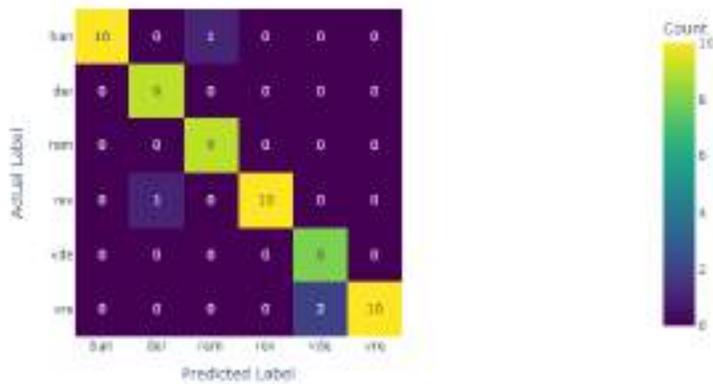


Figura 39 - Matriz de confusión del set de test para el modelo XGBoost con ángulos

### Resultados del entrenamiento de la red LSTM:

El entrenamiento de la red descrita en 3.2.2 consigue un 100% de exactitud en ambos sets de datos tras aproximadamente 40 épocas de entrenamiento, como se puede observar en las siguientes figuras:

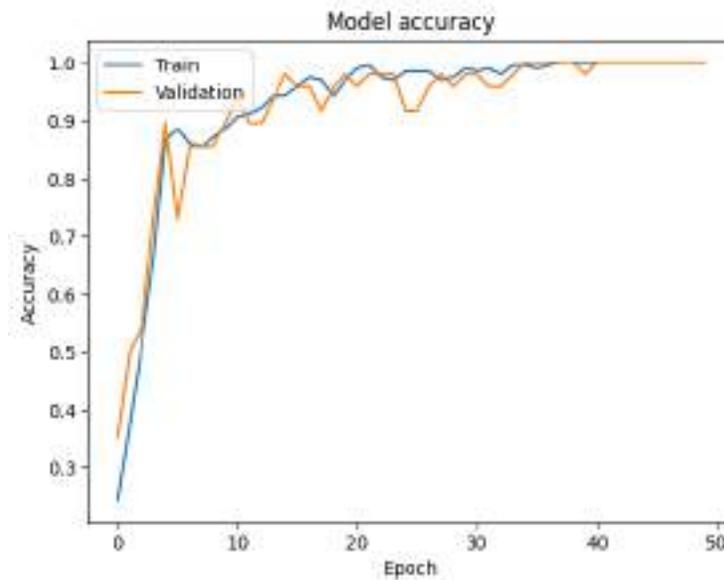


Figura 40 – Evolución de la accuracy del modelo en su entenamiento

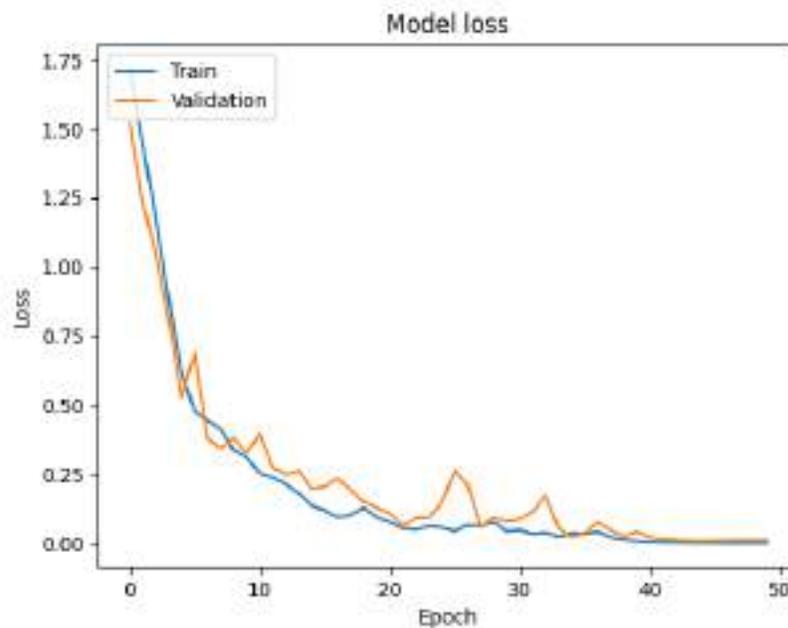


Figura 41 – Evolución de la loss del modelo durante su entrenamiento

Las matrices de confusión de ambos sets reflejan por tanto la clasificación ideal. Se añade la figura del set de test:

LSTM - Test Set - Confusion Matrix

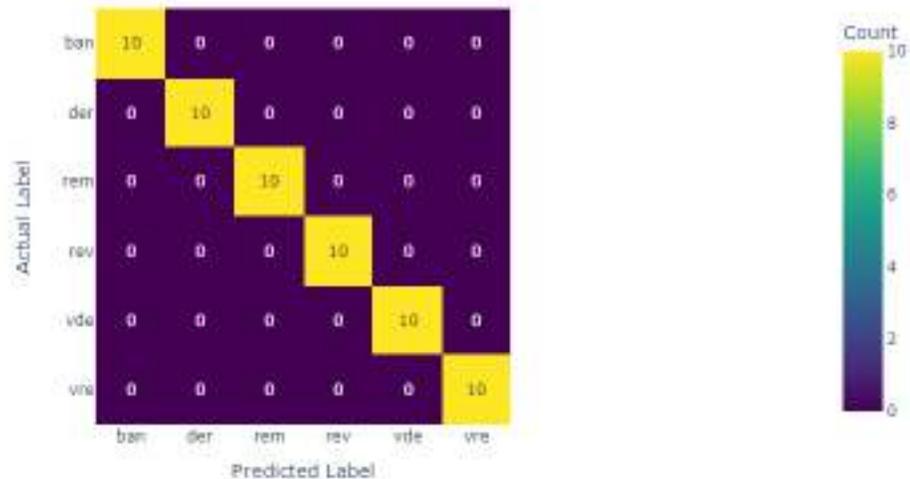


Figura 42 – Matriz de confusión del set de test para el modelo LSTM

Una de las ventajas de los modelos simples como Logistic Regression es que los pesos del modelo pueden ser fácilmente consultados. En este caso en particular, se extraen los pesos tras el entrenamiento, accesibles en el atributo `model.coef_`.

Presentan originalmente la forma (6, 5940) [N\_clases, N\_características]; deshaciendo las operaciones de alisado que se realizan antes del entrenamiento, se pueden reconstruir las dimensiones de los pesos a las dimensiones de [N\_clases, N\_frames, N\_keypoints, N\_ejes], por lo que estos pueden ser visualizados para cada clase.

En particular, se muestran en las siguientes gráficas algunos de los pesos que el modelo ha otorgado a la capa Y de las coordenadas en cada uno de los instantes (frames) para ser capaz de diferenciar entre los distintos golpes.

En la siguiente figura se muestran los pesos que atribuye el modelo a la clase Remate.

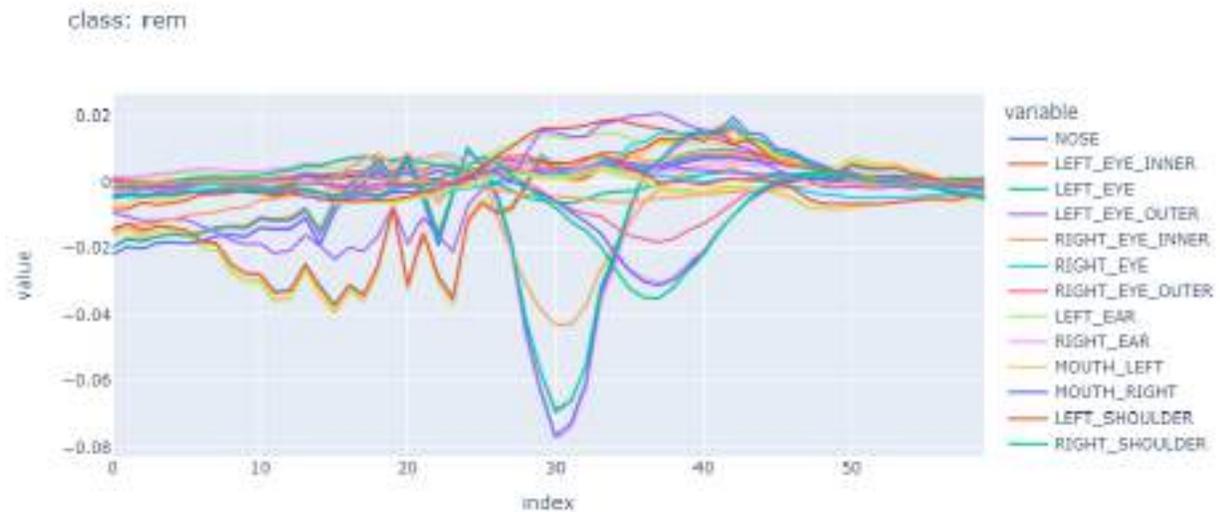


Figura 43 – pesos del modelo para cada keypoint (variable) en los distintos frames (index) para la clase rem (Remate)

En un primer análisis podemos ver cómo el modelo otorga los mayores pesos a los keypoints asociados a la mano derecha en el instante central del vídeo, en el que se produce el golpeo. La línea naranja que sigue destaca en el instante central se corresponde a los pesos otorgados a la posición del codo derecho.

También cabe destacar los pesos que el modelo otorga a la mano izquierda durante los instantes anteriores al golpeo (frames 10 al 25 aprox) en los cuales el jugador alza esa mano durante la preparación del golpeo.

Por otro lado, cabe destacar que varias de las secuencias de pesos tiene pesos prácticamente idénticos entre ellas, lo cual refleja que el modelo está tratando de forma equivalente a esos datos, por lo que, en origen, sus valores serán prácticamente idénticos.

Filtrando por ejemplo por los keypoints de la muñeca izquierda y del índice, meñique y pulgar de la mano izquierda, podemos apreciar en la siguiente figura que sus valores confluyen prácticamente en paralelo, dado que los valores de las posiciones estimados por mediapipe serán prácticamente idénticos.

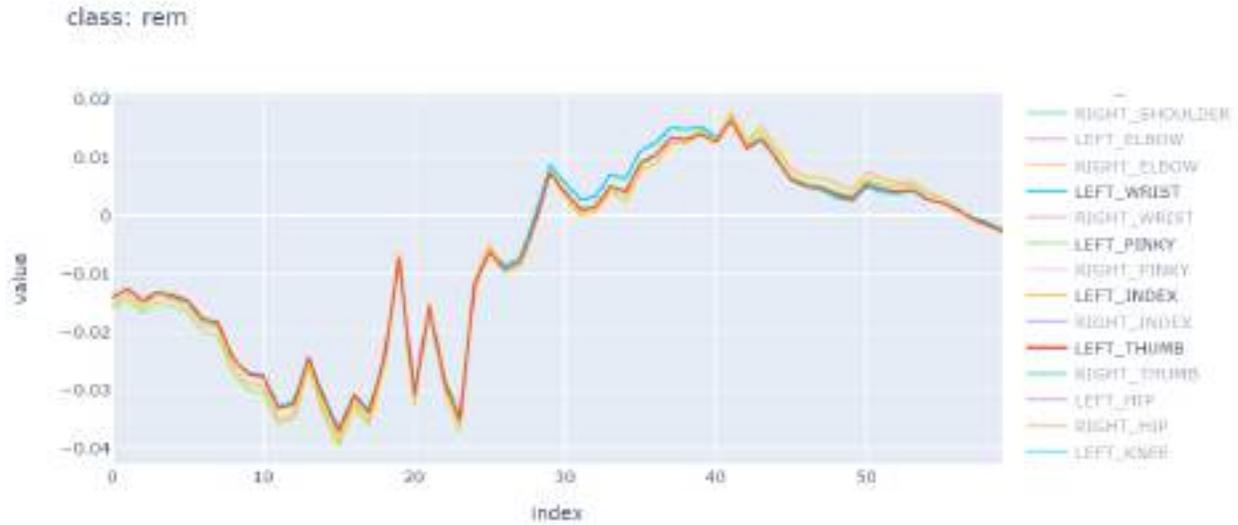


Figura 44 – pesos filtrados para muñeca, meñique, índice y pulgar izquierdos

Al tener estos datos una correlación tal alta entre ellos por su proximidad en el cuerpo, se podría probar a simplificar la entrada manteniendo solamente uno de los keypoints del conjunto y eliminando el resto.

En la siguiente figura se muestra la evolución de pesos para otro golpe, en este caso de revés. Vemos como principalmente los pesos destacan para las posiciones de las manos derecha e izquierda.

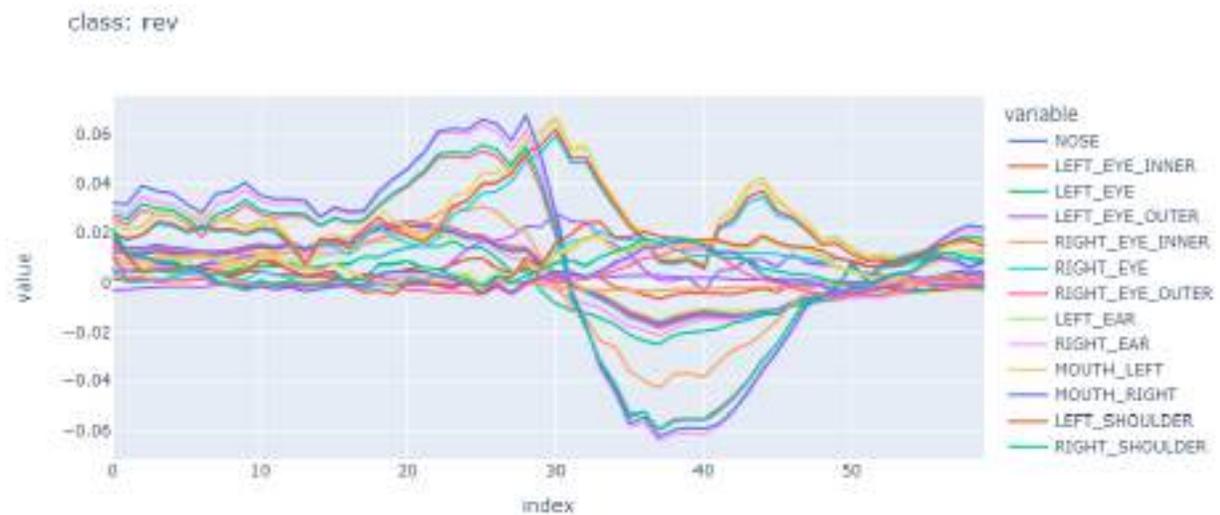


Figura 45 - pesos del modelo para cada keypoint (variable) en los distintos frames (index) para la clase rev (Revés)

En las siguientes figuras se recogen también los pesos de los modelos en una fase temprana de la preparación de modelos en la cual se aprecia el comportamiento “espejo” de los pesos relacionados con la mano derecha (de nuevo, se aprecia un comportamiento similar para los keypoints asociados a la mano) para los movimientos de derecha y de revés sobre el eje X.

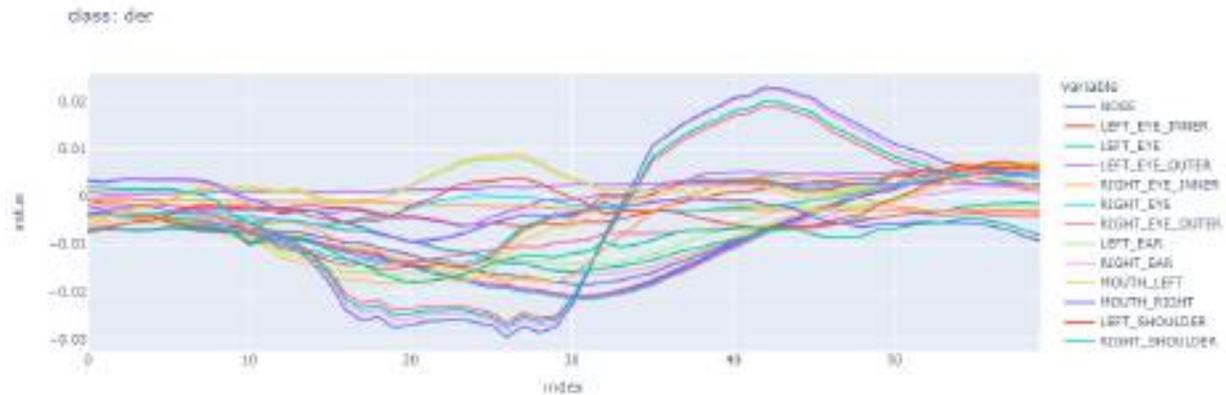


Figura 46 – visualización de pesos del modelo LR para el golpe de derechas

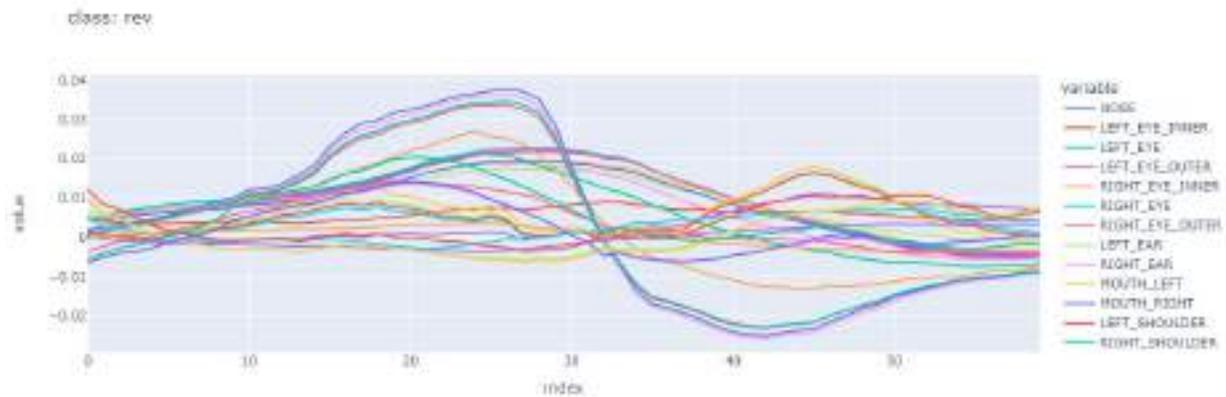


Figura 47 – visualización de pesos del modelo LR para golpe de revés

## 5 Conclusiones y trabajos futuros

### 5.1 Conclusiones

El presente trabajo ha demostrado la viabilidad y eficacia de utilizar técnicas de inteligencia artificial (IA) y visión artificial para la clasificación automática de golpes en el pádel. Los resultados obtenidos han sido satisfactorios, logrando una alta precisión en la identificación de diferentes tipos de golpes como derecha, revés, volea, bandeja y remate. La implementación de modelos de estimación de pose, especialmente MediaPipe, ha permitido extraer características detalladas de los movimientos de los jugadores, facilitando una clasificación precisa y eficiente.

1. Alta precisión en la clasificación de golpes: El modelo desarrollado ha alcanzado métricas de rendimiento cercanas al 100% en precisión, recall y F1-score, demostrando su capacidad para identificar correctamente los golpes de pádel en diversas condiciones ambientales, distintas perspectivas, etc.
2. Creación de un dataset específico y público: La construcción de un dataset propio ha sido crucial para el éxito del modelo. Este dataset, compuesto por vídeos etiquetados de golpes de pádel, no solo ha permitido un entrenamiento robusto del modelo, sino que también se ha hecho público para la comunidad, facilitando futuras investigaciones y desarrollos en el campo del análisis deportivo con IA.
3. Desarrollo de una webapp accesible: La integración del modelo en una webapp proporciona una plataforma fácil de usar que permite a los usuarios subir vídeos y obtener predicciones en tiempo real. Esta accesibilidad democratiza el uso de tecnologías avanzadas, permitiendo a los usuarios el acceso sin necesidad de equipos costosos (ya que el procesamiento se realiza en el servidor y no en el dispositivo) o conocimientos técnicos avanzados.
4. Mejora continua mediante feedback de usuarios: La implementación de un sistema de retroalimentación dentro de la webapp permite a los usuarios corroborar o corregir las predicciones del modelo. Este ciclo de feedback es esencial para la mejora continua del modelo, ya que permite ajustar y refinar las predicciones basadas en el uso real y las correcciones proporcionadas por los usuarios.

### 5.2 Trabajos futuros

A pesar de los avances logrados, existen varias áreas en las que se puede seguir trabajando para mejorar y expandir esta solución:

1. Ampliación del dataset: Continuar recolectando más datos, incluyendo variaciones en las condiciones de iluminación, ángulos de cámara y diferentes niveles de habilidad de los jugadores, para mejorar la robustez y generalización del modelo. Ampliar a más tipos de golpes como podrían ser la chiquita, dejada, bajada de pared, etc.
2. Implementar mejoras de robustez: en ocasiones la detección de MediaPipe falla en el primer frame o en algún frame posterior, causando que no se puedan extraer las posiciones y, por lo tanto, provocando un error en la aplicación. Este caso se podría intentar solucionar imputando

esos valores faltantes con interpolaciones a partir de los valores anexos, por ejemplo. El propio landmarker de MediaPipe dispone de parámetros que pueden ser ajustados para intentar reducir la ocurrencia de este efecto.

3. Habilitar el clasificador sobre un stream completo de vídeo, de forma que el clasificador actúe en cascada con un detector. Por ejemplo, el detector podría reconocer y localizar a varios jugadores sobre cada frame y llamar al clasificador cuando detecte que se produce un golpe. De esta forma podría automatizarse la extracción de métricas de juego, que permitirían elaborar mapas de calor con la posición en la que los jugadores han realizado los golpes o métricas avanzadas orientadas a la mejora táctica.
4. Integración de análisis avanzado de movimiento: Incorporar técnicas de análisis de biomecánica para ofrecer retroalimentación no solo sobre el tipo de golpe, sino también sobre la técnica de ejecución, ayudando a los jugadores a corregir y perfeccionar sus movimientos.
5. Prevención de lesiones: Desarrollar módulos adicionales que analicen la postura y los movimientos de los jugadores para identificar patrones que podrían resultar en lesiones, proporcionando alertas y recomendaciones preventivas.
6. Evaluación en escenarios reales: Realizar estudios en condiciones reales de juego y entrenamiento para validar la efectividad de la herramienta en la práctica diaria, añadiendo nuevas funcionalidades o ajustando las existentes según sea necesario. Sería especialmente interesante contar con la opinión de expertos deportivos (entrenadores y expertos en pádel) para refinar las funcionalidades en base a su feedback, asegurando que las recomendaciones y análisis proporcionados sean de máxima utilidad para los jugadores.
7. Desarrollo de funciones adicionales en la webapp: Añadir funcionalidades como la comparación de desempeño a lo largo del tiempo, la personalización de entrenamientos basados en el análisis

Este proyecto ha sentado las bases para la aplicación de IA en el análisis y mejora del rendimiento en pádel, abriendo numerosas posibilidades para futuras investigaciones y desarrollos que sigan beneficiando a la comunidad deportiva.

## 6. Bibliografía

- [1] Czermak, C. (2023, 11 agosto). Padel Statistics 2022: How Popular is Padel Tennis? - Tennis Creative. *Tennis Creative*. <https://tenniscreative.com/padel-statistics/>
- [2] González González, D. T., & Muñoz Muñoz, D. J. (2023). *Implementación de un sistema de control de postura y velocidad en el patinaje de velocidad basado en procesamiento de imágenes* (Trabajo de titulación). Universidad Nacional de Chimborazo, Riobamba, Ecuador.
- [3] Zhang, W., Liu, Z., Zhou, L., Leung, H., & Chan, A. B. (2017). Martial Arts, Dancing and Sports dataset: A challenging stereo and multi-view dataset for 3D human pose estimation. *Image And Vision Computing*, 61, 22-39. <https://doi.org/10.1016/j.imavis.2017.02.002>
- [4] Mira Abad, Á.(2023). *Mejora de rendimiento deportivo mediante la detección de posturas por visión artificial* (Trabajo Final de Grado). Universitat Politècnica de València, España.
- [5] Songai, G., Ningning, Z., & Xiang, Z. (2022). Design of Precise Estimation Algorithm for Sports Pose Based on Multifeature Fusion. *Journal Of Mobile Information Systems*, 2022, 1-9. <https://doi.org/10.1155/2022/1224939>
- [6] Ashraf, F. B., Islam, M. U., Kabir, M. R., & Uddin, J. (2023). YoNet: A Neural Network for Yoga Pose Classification. *SN Computer Science/SN Computer Science*, 4(2). <https://doi.org/10.1007/s42979-022-01618-8>
- [7] Chamola, V., Gummana, E. P., Madan, A., Rout, B. K., & Rodrigues, J. J. P. C. (2024). Advancements in Yoga Pose Estimation Using Artificial Intelligence: A Survey. *Current Bioinformatics*, 19(3), 264-280. <https://doi.org/10.2174/1574893618666230508105440>
- [8] Citraro, L., Márquez-Neila, P., Savarè, S., Jayaram, V., Dubout, C., Renaut, F., Hasfura, A., Shitrit, H. B., & Fua, P. (2020). Real-time camera pose estimation for sports fields. *Machine Vision And Applications*, 31(3). <https://doi.org/10.1007/s00138-020-01064-7>
- [9] Dale, M. (2023, 11 mayo). *AI analytics for padel is here - The Padel Paper*. The Padel Paper. <https://thepadelpaper.com/gametraq-gamecam-padel-ai-video-analysis/>
- [10] *GAMETRAQ® 3 AI camera kit — GameCam® AI cameras for sports*. (s. f.). GameCam® AI Cameras For Sports. <https://www.gamecam.se/products/p/8pa47edy1z6tz2ja3cke4ygjke8pyl>
- [11] Dale, M. (2023a, enero 26). *PadelCourt app offers live scores, video highlights and coaching tips - The Padel Paper*. The Padel Paper. <https://thepadelpaper.com/padelcourt-app-offers-live-scores-video-highlights-and-coaching-tips/>
- [12] *PadelCourt - Artificial Intelligence for padel*. (2023, 20 marzo). Artificial Intelligence For Padel. <https://padelcourt.app/>
- [13] Misha. (2024, 21 mayo). *Aiball - It's not all about technique*. Aiball. <https://www.aiball.io/>
- [14] *SwingVision Review: 5 Ways it can Improve Your Tennis*. (2023, 5 diciembre). My Tennis Coaching. <https://mytenniscoaching.com/2022/08/29/swingvision-review/>
- [15] *Finetuning YOLOv8 pose estimation for padel player analysis using trainYOLO*. (s. f.). <https://www.trainyolo.com/blog/padel-player-analysis>

- [16] *Pose landmark detection guide*. (s. f.). Google For Developers. [https://ai.google.dev/edge/mediapipe/solutions/vision/pose\\_landmarker](https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker)
- [17] Google-Ai-Edge. (s. f.). *mediapipe/docs/solutions/pose.md at master · google-ai-edge/mediapipe*. GitHub. <https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/pose.md>
- [18] *On-device, Real-time Body Pose Tracking with MediaPipe BlazePose*. (s. f.). <https://research.google/blog/on-device-real-time-body-pose-tracking-with-mediapipe-blazepose/>
- [19] *GHUM & GHUML: Generative 3D Human Shape and Articulated Pose Models*. (2020, 1 junio). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/9157563>
- [20] *Pose detection*. (s. f.). Google For Developers. <https://developers.google.com/ml-kit/vision/pose-detection>
- [21] Google-Ai-Edge. (s. f.-a). *mediapipe/docs/solutions/pose\_classification.md at master · google-ai-edge/mediapipe*. GitHub. [https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/pose\\_classification.md#classification](https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/pose_classification.md#classification)
- [22] Xu, Z., Lou, Y., Song, Y., & Lin, Z. (2023, 7 marzo). *Fitness Action Recognition Based on MediaPipe*. <https://publications.waset.org/10012992/fitness-action-recognition-based-on-mediapipe>

#### Páginas web (vídeos)

- Dani 13. (2024, 5 marzo). SIUX FENIX PRO 4 / Review y sensaciones - Dani13 [Vídeo]. YouTube. <https://www.youtube.com/watch?v=53yiyicPVEw>
- Dani 13. (2024b, marzo 7). OXDOG HYPER PRO 2.0 , ¿ LA MEJOR PALA HÍBRIDA? // Review y sensaciones - Dani13 [Vídeo]. YouTube. <https://www.youtube.com/watch?v=3KX3I8x9bGo>
- Dani 13. (2024c, abril 28). PROBAMOS LAS JOMA SLAM PRO 2023 y 2024 CON GUILLE , JUGADOR DE JOMA [Vídeo]. YouTube. <https://www.youtube.com/watch?v=6hHVYa6lkpA>
- Dani 13. (2024d, junio 7). NOX ML10 BAHÍA 2024 // Review y sensaciones - Dani13 [Vídeo]. YouTube. <https://www.youtube.com/watch?v=SkpbdlQgvck>
- Dani 13. (2023, 11 junio). HEAD SPEED PRO // Review y sensaciones Dani13 [Vídeo]. YouTube. [https://www.youtube.com/watch?v=rceY\\_-9xRJY](https://www.youtube.com/watch?v=rceY_-9xRJY)
- Dani 13. (2023b, junio 16). HEAD SPEED PRO X // Review y sensaciones Dani 13 [Vídeo]. YouTube. <https://www.youtube.com/watch?v=PnZfDeSMRWs>
- Dani 13. (2023c, noviembre 6). NOX AT10 18K 2023 // Review y sensaciones -Dani13 [Vídeo]. YouTube. [https://www.youtube.com/watch?v=\\_4Qo9tVJMjA](https://www.youtube.com/watch?v=_4Qo9tVJMjA)
- Mejora tu Padel. (2023, 2 noviembre). MI FAVORITA: NOX AT10 de AGUSTIN TAPIA y LEO AUGSBURGER 2024 | Mejora tu padel [Vídeo]. YouTube. <https://www.youtube.com/watch?v=uf1UfPuecOQ>

Mejora tu Padel (2023b, noviembre 2). MI FAVORITA: NOX AT10 de AGUSTIN TAPIA y LEO AUGSBURGER 2024 | Mejora tu padel [Vídeo]. YouTube. <https://www.youtube.com/watch?v=uf1UfPuecOQ>

MasyMasPadel. (2023, 8 marzo). REVIEW AT10 18K 2023 la pala de AGUSTIN TAPIA! ¿NUEVA TENOLOGÍA? ¿POTECIA? ¿CONTROL? | MasyMasPadel [Vídeo]. YouTube. <https://www.youtube.com/watch?v=05pTBpsT6KE>

Paddelea. (2016, 27 julio). Curso pádel iniciación - El golpe de derecha [Vídeo]. YouTube. <https://www.youtube.com/watch?v=AGEFLIIScnA>

Probando Palas. (2024, 21 enero). NOX AT10 18K 2024 [La pala de AGUSTÍN TAPIA] ¿La más CÓMODA?! #padel [Vídeo]. YouTube. [https://www.youtube.com/watch?v=B\\_QMVGutsM8](https://www.youtube.com/watch?v=B_QMVGutsM8)

## 7 Anexo

### 7.1 Informes de evaluaci3n completos

#### 7.1.1 Coordenadas

```

Logistic Regression - Training Set - Accuracy: 1.0
Logistic Regression - Training Set - Precision (macro): 100.00%
Logistic Regression - Training Set - Recall (macro): 100.00%
Logistic Regression - Training Set - F1 Score (macro): 100.00%
Logistic Regression - Training Set - Precision (per class): [100. 100. 100. 100. 100. 100.]
Logistic Regression - Training Set - Recall (per class): [100. 100. 100. 100. 100. 100.]
Logistic Regression - Training Set - F1 Score (per class): [100. 100. 100. 100. 100. 100.]
Logistic Regression - Training Set - Classification Report:
      precision    recall  f1-score   support

   ban         1.00      1.00      1.00         40
   der         1.00      1.00      1.00         40
   rem         1.00      1.00      1.00         40
   rev         1.00      1.00      1.00         40
   vde         1.00      1.00      1.00         40
   vre         1.00      1.00      1.00         40

 accuracy                   1.00         240
 macro avg         1.00      1.00      1.00         240
weighted avg         1.00      1.00      1.00         240

```

```

Logistic Regression - Test Set - Accuracy: 1.0
Logistic Regression - Test Set - Precision (macro): 100.00%
Logistic Regression - Test Set - Recall (macro): 100.00%
Logistic Regression - Test Set - F1 Score (macro): 100.00%
Logistic Regression - Test Set - Precision (per class): [100. 100. 100. 100. 100. 100.]
Logistic Regression - Test Set - Recall (per class): [100. 100. 100. 100. 100. 100.]
Logistic Regression - Test Set - F1 Score (per class): [100. 100. 100. 100. 100. 100.]
Logistic Regression - Test Set - Classification Report:
      precision    recall  f1-score   support

   ban         1.00      1.00      1.00         10
   der         1.00      1.00      1.00         10
   rem         1.00      1.00      1.00         10
   rev         1.00      1.00      1.00         10
   vde         1.00      1.00      1.00         10
   vre         1.00      1.00      1.00         10

 accuracy                   1.00         60
 macro avg         1.00      1.00      1.00         60
weighted avg         1.00      1.00      1.00         60

```

```

Random Forest - Training Set - Accuracy: 1.0
Random Forest - Training Set - Precision (macro): 100.00%
Random Forest - Training Set - Recall (macro): 100.00%
Random Forest - Training Set - F1 Score (macro): 100.00%
Random Forest - Training Set - Precision (per class): [100. 100. 100. 100. 100. 100.]
Random Forest - Training Set - Recall (per class): [100. 100. 100. 100. 100. 100.]
Random Forest - Training Set - F1 Score (per class): [100. 100. 100. 100. 100. 100.]
Random Forest - Training Set - Classification Report:
      precision    recall  f1-score   support

   ban         1.00      1.00      1.00         40
   der         1.00      1.00      1.00         40
   rem         1.00      1.00      1.00         40
   rev         1.00      1.00      1.00         40
   vde         1.00      1.00      1.00         40
   vre         1.00      1.00      1.00         40

 accuracy
macro avg         1.00      1.00      1.00         240
weighted avg         1.00      1.00      1.00         240
    
```

```

Random Forest - Test Set - Accuracy: 0.9333333333333333
Random Forest - Test Set - Precision (macro): 93.33%
Random Forest - Test Set - Recall (macro): 94.07%
Random Forest - Test Set - F1 Score (macro): 93.25%
Random Forest - Test Set - Precision (per class): [100. 90. 100. 100. 70. 100.]
Random Forest - Test Set - Recall (per class): [100. 100. 100. 100. 87.5 100.]
Random Forest - Test Set - F1 Score (per class): [100. 94.73684211 100. 100. 77.77777778 86.95652174]
Random Forest - Test Set - Classification Report:
      precision    recall  f1-score   support

   ban         1.00      1.00      1.00         10
   der         0.90      1.00      0.95          9
   rem         1.00      1.00      1.00         10
   rev         1.00      1.00      1.00         10
   vde         0.70      0.88      0.78          8
   vre         1.00      0.77      0.87         13

 accuracy
macro avg         0.93      0.94      0.93         60
weighted avg         0.95      0.93      0.93         60
    
```

```
XGBoost - Training Set - Accuracy: 1.0
XGBoost - Training Set - Precision (macro): 100.00%
XGBoost - Training Set - Recall (macro): 100.00%
XGBoost - Training Set - F1 Score (macro): 100.00%
XGBoost - Training Set - Precision (per class): [100. 100. 100. 100. 100. 100.]
XGBoost - Training Set - Recall (per class): [100. 100. 100. 100. 100. 100.]
XGBoost - Training Set - F1 Score (per class): [100. 100. 100. 100. 100. 100.]
XGBoost - Training Set - Classification Report:
```

	precision	recall	f1-score	support
ban	1.00	1.00	1.00	40
der	1.00	1.00	1.00	40
rem	1.00	1.00	1.00	40
rev	1.00	1.00	1.00	40
vde	1.00	1.00	1.00	40
vre	1.00	1.00	1.00	40
accuracy			1.00	240
macro avg	1.00	1.00	1.00	240
weighted avg	1.00	1.00	1.00	240

```
XGBoost - Test Set - Accuracy: 0.9666666666666667
XGBoost - Test Set - Precision (macro): 96.67%
XGBoost - Test Set - Recall (macro): 97.22%
XGBoost - Test Set - F1 Score (macro): 96.73%
XGBoost - Test Set - Precision (per class): [100. 90. 100. 100. 90. 100.]
XGBoost - Test Set - Recall (per class): [100. 100. 100. 100. 100. 100.]
100. 100.
83.33333333]
XGBoost - Test Set - F1 Score (per class): [100. 94.73684211 100.
100. 90.90909091]
```

```
XGBoost - Test Set - Classification Report:
```

	precision	recall	f1-score	support
ban	1.00	1.00	1.00	10
der	0.90	1.00	0.95	9
rem	1.00	1.00	1.00	10
rev	1.00	1.00	1.00	10
vde	0.90	1.00	0.95	9
vre	1.00	0.83	0.91	12
accuracy			0.97	60
macro avg	0.97	0.97	0.97	60
weighted avg	0.97	0.97	0.97	60

### 7.1.2 Distancias

```
Distancias - Logistic Regression - Training Set - Accuracy: 1.0
Distancias - Logistic Regression - Training Set - Precision (macro): 100.00%
Distancias - Logistic Regression - Training Set - Recall (macro): 100.00%
Distancias - Logistic Regression - Training Set - F1 Score (macro): 100.00%
Distancias - Logistic Regression - Training Set - Precision (per class): [100.
100. 100. 100. 100.]
Distancias - Logistic Regression - Training Set - Recall (per class): [100. 100.
100. 100. 100.]
Distancias - Logistic Regression - Training Set - F1 Score (per class): [100. 100.
100. 100. 100.]
Distancias - Logistic Regression - Training Set - Classification Report:
      precision    recall  f1-score   support

   ban         1.00      1.00      1.00         40
   der         1.00      1.00      1.00         40
   rem         1.00      1.00      1.00         40
   rev         1.00      1.00      1.00         40
   vde         1.00      1.00      1.00         40
   vre         1.00      1.00      1.00         40

 accuracy                   1.00         240
 macro avg          1.00      1.00      1.00         240
 weighted avg       1.00      1.00      1.00         240
```

```
Distancias - Logistic Regression - Test Set - Accuracy: 0.9833333333333333
Distancias - Logistic Regression - Test Set - Precision (macro): 98.33%
Distancias - Logistic Regression - Test Set - Recall (macro): 98.48%
Distancias - Logistic Regression - Test Set - F1 Score (macro): 98.33%
Distancias - Logistic Regression - Test Set - Precision (per class): [100. 90.
100. 100. 100. 100.]
Distancias - Logistic Regression - Test Set - Recall (per class): [100.
100. 100. 100. 100. 90.90909091
100. ]
Distancias - Logistic Regression - Test Set - F1 Score (per class): [100.
94.73684211 100. 100. 95.23809524
100. ]
Distancias - Logistic Regression - Test Set - Classification Report:
      precision    recall  f1-score   support

   ban         1.00      1.00      1.00         10
   der         0.90      1.00      0.95          9
   rem         1.00      1.00      1.00         10
   rev         1.00      1.00      1.00         10
   vde         1.00      0.91      0.95         11
   vre         1.00      1.00      1.00         10

 accuracy                   0.98         60
 macro avg          0.98      0.98      0.98         60
 weighted avg       0.98      0.98      0.98         60
```

```
Distances - Random Forest - Training Set - Accuracy: 1.0
Distances - Random Forest - Training Set - Precision (macro): 100.00%
Distances - Random Forest - Training Set - Recall (macro): 100.00%
Distances - Random Forest - Training Set - F1 Score (macro): 100.00%
Distances - Random Forest - Training Set - Precision (per class): [100. 100. 100.
100. 100. 100.]
Distances - Random Forest - Training Set - Recall (per class): [100. 100. 100.
100. 100. 100.]
Distances - Random Forest - Training Set - F1 Score (per class): [100. 100. 100.
100. 100. 100.]
Distances - Random Forest - Training Set - Classification Report:
      precision    recall  f1-score   support

   ban         1.00      1.00      1.00         40
   der         1.00      1.00      1.00         40
   rem         1.00      1.00      1.00         40
   rev         1.00      1.00      1.00         40
   vde         1.00      1.00      1.00         40
   vre         1.00      1.00      1.00         40

 accuracy                   1.00         240
macro avg         1.00      1.00      1.00         240
weighted avg         1.00      1.00      1.00         240
```

```
Distances - Random Forest - Test Set - Accuracy: 1.0
Distances - Random Forest - Test Set - Precision (macro): 100.00%
Distances - Random Forest - Test Set - Recall (macro): 100.00%
Distances - Random Forest - Test Set - F1 Score (macro): 100.00%
Distances - Random Forest - Test Set - Precision (per class): [100. 100. 100.
100. 100. 100.]
Distances - Random Forest - Test Set - Recall (per class): [100. 100. 100. 100.
100. 100.]
Distances - Random Forest - Test Set - F1 Score (per class): [100. 100. 100. 100.
100. 100.]
Distances - Random Forest - Test Set - Classification Report:
      precision    recall  f1-score   support

   ban         1.00      1.00      1.00         10
   der         1.00      1.00      1.00         10
   rem         1.00      1.00      1.00         10
   rev         1.00      1.00      1.00         10
   vde         1.00      1.00      1.00         10
   vre         1.00      1.00      1.00         10

 accuracy                   1.00         60
macro avg         1.00      1.00      1.00         60
weighted avg         1.00      1.00      1.00         60
```

```

Distances - XGBoost - Training Set - Accuracy: 1.0
Distances - XGBoost - Training Set - Precision (macro): 100.00%
Distances - XGBoost - Training Set - Recall (macro): 100.00%
Distances - XGBoost - Training Set - F1 Score (macro): 100.00%
Distances - XGBoost - Training Set - Precision (per class): [100. 100. 100. 100.
100. 100.]
Distances - XGBoost - Training Set - Recall (per class): [100. 100. 100. 100.
100. 100.]
Distances - XGBoost - Training Set - F1 Score (per class): [100. 100. 100. 100.
100. 100.]
Distances - XGBoost - Training Set - Classification Report:
      precision    recall  f1-score   support

   ban         1.00      1.00      1.00         40
   der         1.00      1.00      1.00         40
   rem         1.00      1.00      1.00         40
   rev         1.00      1.00      1.00         40
   vde         1.00      1.00      1.00         40
   vre         1.00      1.00      1.00         40

 accuracy                1.00         240
macro avg         1.00      1.00      1.00         240
weighted avg         1.00      1.00      1.00         240

Distances - XGBoost - Test Set - Accuracy: 0.9833333333333333
Distances - XGBoost - Test Set - Precision (macro): 98.33%
Distances - XGBoost - Test Set - Recall (macro): 98.48%
Distances - XGBoost - Test Set - F1 Score (macro): 98.33%
Distances - XGBoost - Test Set - Precision (per class): [100.  90. 100. 100. 100.
100.]
Distances - XGBoost - Test Set - Recall (per class): [100.          100.
100.    90.90909091 100.
100.    ]
Distances - XGBoost - Test Set - F1 Score (per class): [100.          94.73684211
100.    95.23809524 100.
100.    ]
Distances - XGBoost - Test Set - Classification Report:
      precision    recall  f1-score   support

   ban         1.00      1.00      1.00         10
   der         0.90      1.00      0.95          9
   rem         1.00      1.00      1.00         10
   rev         1.00      0.91      0.95         11
   vde         1.00      1.00      1.00         10
   vre         1.00      1.00      1.00         10

 accuracy                0.98         60
macro avg         0.98      0.98      0.98         60
weighted avg         0.98      0.98      0.98         60
    
```

### 7.1.3 Ángulos

```

Angles - Logistic Regression - Training Set - Accuracy: 1.0
Angles - Logistic Regression - Training Set - Precision (macro): 100.00%
Angles - Logistic Regression - Training Set - Recall (macro): 100.00%
Angles - Logistic Regression - Training Set - F1 Score (macro): 100.00%
Angles - Logistic Regression - Training Set - Precision (per class): [100. 100.
100. 100. 100. 100.]
Angles - Logistic Regression - Training Set - Recall (per class): [100. 100. 100.
100. 100. 100.]
Angles - Logistic Regression - Training Set - F1 Score (per class): [100. 100.
100. 100. 100. 100.]
Angles - Logistic Regression - Training Set - Classification Report:
      precision    recall  f1-score   support

   ban           1.00         1.00         1.00         40
   der           1.00         1.00         1.00         40
   rem           1.00         1.00         1.00         40
   rev           1.00         1.00         1.00         40
   vde           1.00         1.00         1.00         40
   vre           1.00         1.00         1.00         40

 accuracy                   1.00         240
 macro avg           1.00         1.00         1.00         240
 weighted avg        1.00         1.00         1.00         240
    
```

```

Angles - Logistic Regression - Test Set - Accuracy: 1.0
Angles - Logistic Regression - Test Set - Precision (macro): 100.00%
Angles - Logistic Regression - Test Set - Recall (macro): 100.00%
Angles - Logistic Regression - Test Set - F1 Score (macro): 100.00%
Angles - Logistic Regression - Test Set - Precision (per class): [100. 100. 100.
100. 100. 100.]
Angles - Logistic Regression - Test Set - Recall (per class): [100. 100. 100.
100. 100. 100.]
Angles - Logistic Regression - Test Set - F1 Score (per class): [100. 100. 100.
100. 100. 100.]
Angles - Logistic Regression - Test Set - Classification Report:
      precision    recall  f1-score   support

   ban           1.00         1.00         1.00         10
   der           1.00         1.00         1.00         10
   rem           1.00         1.00         1.00         10
   rev           1.00         1.00         1.00         10
   vde           1.00         1.00         1.00         10
   vre           1.00         1.00         1.00         10

 accuracy                   1.00         60
 macro avg           1.00         1.00         1.00         60
 weighted avg        1.00         1.00         1.00         60
    
```

```

Angles - Random Forest - Training Set - Accuracy: 1.0
Angles - Random Forest - Training Set - Precision (macro): 100.00%
Angles - Random Forest - Training Set - Recall (macro): 100.00%
Angles - Random Forest - Training Set - F1 Score (macro): 100.00%
Angles - Random Forest - Training Set - Precision (per class): [100. 100. 100.
100. 100. 100.]
Angles - Random Forest - Training Set - Recall (per class): [100. 100. 100. 100.
100. 100.]
Angles - Random Forest - Training Set - F1 Score (per class): [100. 100. 100.
100. 100. 100.]
Angles - Random Forest - Training Set - Classification Report:
      precision    recall  f1-score   support

   ban         1.00      1.00      1.00         40
   der         1.00      1.00      1.00         40
   rem         1.00      1.00      1.00         40
   rev         1.00      1.00      1.00         40
   vde         1.00      1.00      1.00         40
   vre         1.00      1.00      1.00         40

 accuracy                1.00         240
macro avg         1.00      1.00      1.00         240
weighted avg         1.00      1.00      1.00         240
    
```

```

Angles - Random Forest - Test Set - Accuracy: 0.9833333333333333
Angles - Random Forest - Test Set - Precision (macro): 98.33%
Angles - Random Forest - Test Set - Recall (macro): 98.48%
Angles - Random Forest - Test Set - F1 Score (macro): 98.33%
Angles - Random Forest - Test Set - Precision (per class): [100. 100. 100. 100.
90. 100.]
Angles - Random Forest - Test Set - Recall (per class): [100.          100.
100.          100.          100.
90.90909091]
Angles - Random Forest - Test Set - F1 Score (per class): [100.          100.
100.          100.          94.73684211
95.23809524]
Angles - Random Forest - Test Set - Classification Report:
      precision    recall  f1-score   support

   ban         1.00      1.00      1.00         10
   der         1.00      1.00      1.00         10
   rem         1.00      1.00      1.00         10
   rev         1.00      1.00      1.00         10
   vde         0.90      1.00      0.95          9
   vre         1.00      0.91      0.95         11

 accuracy                0.98         60
macro avg         0.98      0.98      0.98         60
weighted avg         0.98      0.98      0.98         60
    
```

```

Angles - XGBoost - Training Set - Accuracy: 1.0
Angles - XGBoost - Training Set - Precision (macro): 100.00%
Angles - XGBoost - Training Set - Recall (macro): 100.00%
Angles - XGBoost - Training Set - F1 Score (macro): 100.00%
Angles - XGBoost - Training Set - Precision (per class): [100. 100. 100. 100. 100. 100.]
Angles - XGBoost - Training Set - Recall (per class): [100. 100. 100. 100. 100. 100.]
Angles - XGBoost - Training Set - F1 Score (per class): [100. 100. 100. 100. 100. 100.]
Angles - XGBoost - Training Set - Classification Report:
      precision    recall  f1-score   support

   ban         1.00      1.00      1.00         40
   der         1.00      1.00      1.00         40
   rem         1.00      1.00      1.00         40
   rev         1.00      1.00      1.00         40
   vde         1.00      1.00      1.00         40
   vre         1.00      1.00      1.00         40

 accuracy                1.00         240
macro avg         1.00      1.00      1.00         240
weighted avg         1.00      1.00      1.00         240

Angles - XGBoost - Test Set - Accuracy: 0.9333333333333333
Angles - XGBoost - Test Set - Precision (macro): 93.33%
Angles - XGBoost - Test Set - Recall (macro): 94.19%
Angles - XGBoost - Test Set - F1 Score (macro): 93.29%
Angles - XGBoost - Test Set - Precision (per class): [100.  90.  90. 100.  80. 100.]
Angles - XGBoost - Test Set - Recall (per class): [ 90.90909091 100.          100. 90.90909091 100.          83.33333333]
Angles - XGBoost - Test Set - F1 Score (per class): [95.23809524 94.73684211 94.73684211 95.23809524 88.88888889 90.90909091]
Angles - XGBoost - Test Set - Classification Report:
      precision    recall  f1-score   support

   ban         1.00      0.91      0.95         11
   der         0.90      1.00      0.95          9
   rem         0.90      1.00      0.95          9
   rev         1.00      0.91      0.95         11
   vde         0.80      1.00      0.89          8
   vre         1.00      0.83      0.91         12

 accuracy                0.93         60
macro avg         0.93      0.94      0.93         60
weighted avg         0.94      0.93      0.93         60

```

### 7.1.4 LSTM

LSTM - Train Set - Accuracy: 1.0  
 LSTM - Train Set - Precision (macro): 100.00%  
 LSTM - Train Set - Recall (macro): 100.00%  
 LSTM - Train Set - F1 Score (macro): 100.00%  
 LSTM - Train Set - Precision (per class): [100. 100. 100. 100. 100. 100.]  
 LSTM - Train Set - Recall (per class): [100. 100. 100. 100. 100. 100.]  
 LSTM - Train Set - F1 Score (per class): [100. 100. 100. 100. 100. 100.]  
 LSTM - Train Set - Classification Report:

	precision	recall	f1-score	support
ban	1.00	1.00	1.00	40
der	1.00	1.00	1.00	40
rem	1.00	1.00	1.00	40
rev	1.00	1.00	1.00	40
vde	1.00	1.00	1.00	40
vre	1.00	1.00	1.00	40
accuracy			1.00	240
macro avg	1.00	1.00	1.00	240
weighted avg	1.00	1.00	1.00	240

LSTM - Test Set - Accuracy: 1.0  
 LSTM - Test Set - Precision (macro): 100.00%  
 LSTM - Test Set - Recall (macro): 100.00%  
 LSTM - Test Set - F1 Score (macro): 100.00%  
 LSTM - Test Set - Precision (per class): [100. 100. 100. 100. 100. 100.]  
 LSTM - Test Set - Recall (per class): [100. 100. 100. 100. 100. 100.]  
 LSTM - Test Set - F1 Score (per class): [100. 100. 100. 100. 100. 100.]  
 LSTM - Test Set - Classification Report:

	precision	recall	f1-score	support
ban	1.00	1.00	1.00	10
der	1.00	1.00	1.00	10
rem	1.00	1.00	1.00	10
rev	1.00	1.00	1.00	10
vde	1.00	1.00	1.00	10
vre	1.00	1.00	1.00	10
accuracy			1.00	60
macro avg	1.00	1.00	1.00	60
weighted avg	1.00	1.00	1.00	60

## 7.2 Pruebas en la webapp

Se muestran a continuación algunas de las pruebas llevadas a cabo en la webapp con distintos tipos de formato de entrada y calidades.

- Pruebas con grabaciones de partidos con calidad baja de vídeo:



Figura 48 – Ejemplo de vídeo recortado con baja resolución



Figura 49- Detalle de la predicción de pose correcta pese a la baja calidad del vídeo

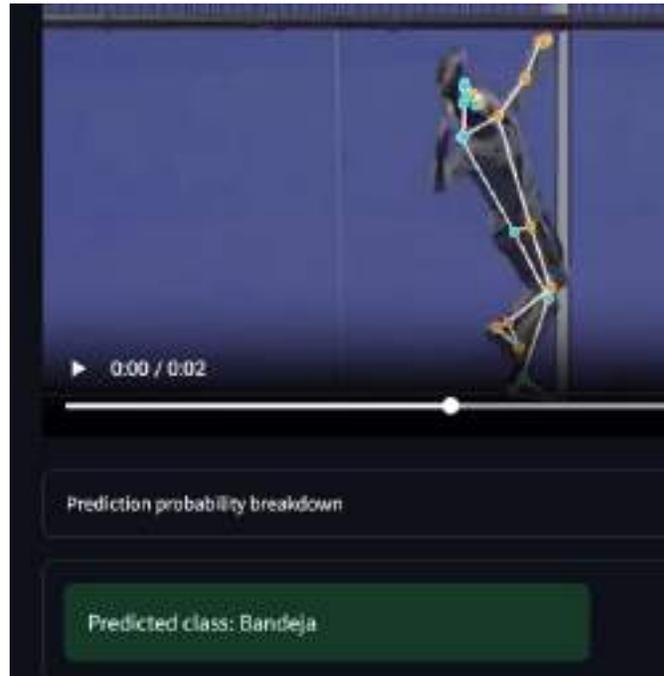


Figura 50 – Resultado de la predicción correcto

- Predicciones con jugadores de distintos niveles de experiencia

A pesar de que los modelos han sido entrenados con capturas de movimiento de jugadores experimentados, las predicciones sobre jugadores menos nivel técnico son correctas.

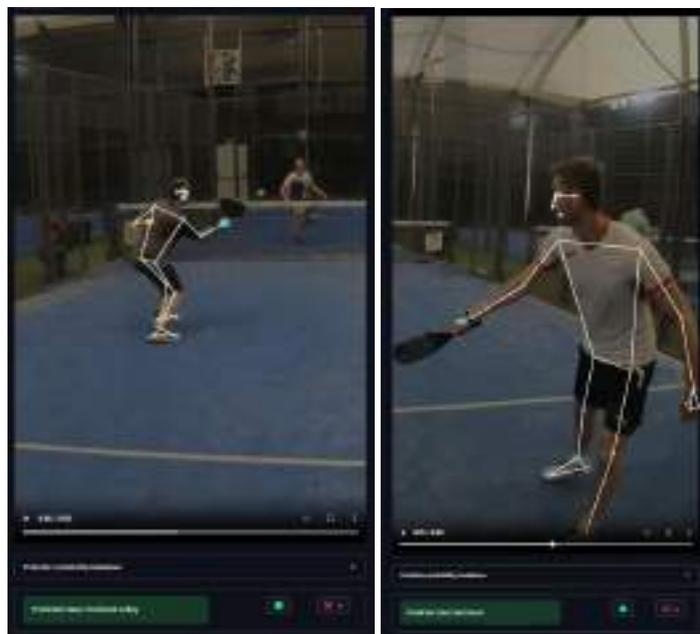


Figura 51 – Resultados correctos sobre muestras de jugadores amateur

- Predicciones correctas sobre grabaciones de pantalla de contenido en redes sociales

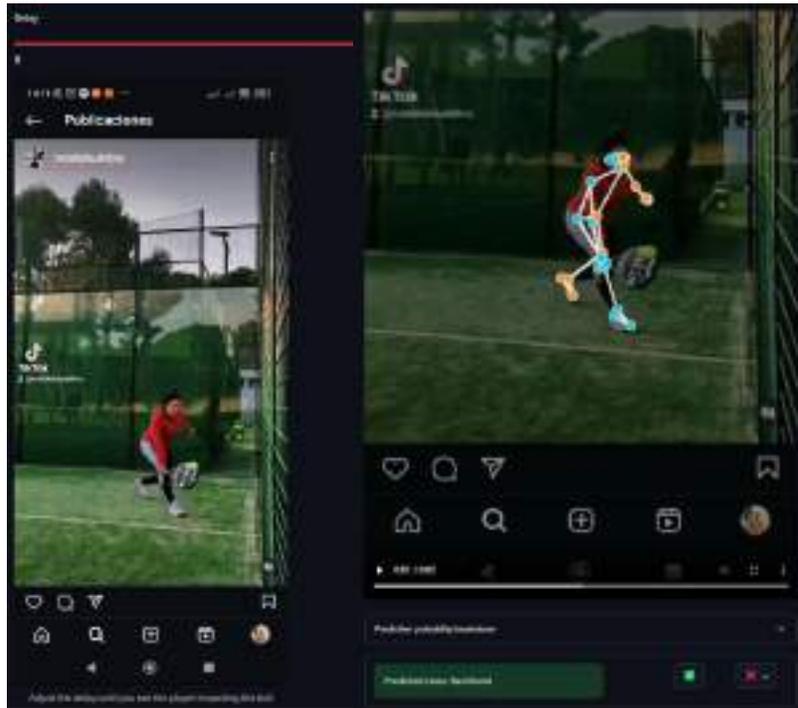


Figura 52 – Predicción correcta sobre una grabación de pantalla de contenido publicado en redes sociales

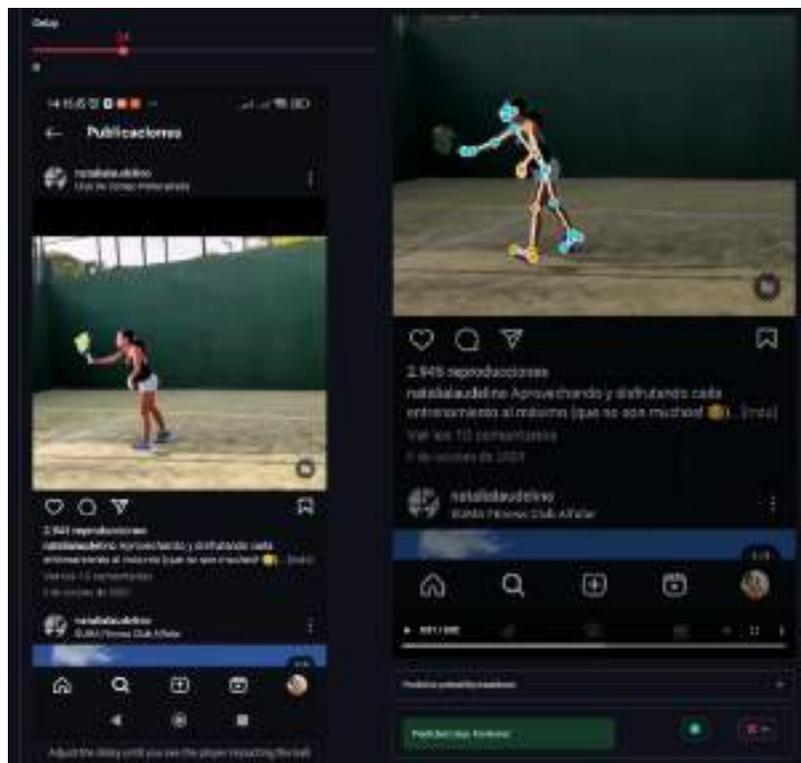


Figura 53 - Predicción correcta sobre una grabación de pantalla de contenido horizontal publicado en redes sociales

- Fallo por confusión con fondo complejo y baja calidad del vídeo



Figura 54 – Fallo de estimación de pose: el fondo hace confundir la posición del brazo, que está adelantado

- Pérdida de detección de keypoints durante el vídeo en condiciones complejas (oclusión por la red y baja calidad)



Figura 55 – Pérdida de detección de keypoints en secuencia de movimiento