

Pose Analysis

Marc Picazos Carrillo

Resumen— Este documento se centra en el uso de sistemas de pose estimation para analizar actividades Deportivas. El trabajo pasa desde la extracción de los keypoints mediante los sistemas de pose estimation, hasta la detección y tracking de los atletas para finalizar estudiando y comparando las actividades realizadas. Para analizar y detectar patrones en una actividad realizada por dos personas diferentes se ha utilizado el algoritmo Dynamic Time Warping, el cual permite la detección de patrones y alineamiento de dos secuencias temporales distintas.

Paraules clau— Pose estimation, dynamic time warping, visión por computador

Abstract— This paper focuses on the use of pose estimation systems to analyze sports activities. The work goes from the extraction of the keypoints using pose estimation systems, to the detection and tracking of the athletes and finally comparing the activities performed. In order to analyze and detect patterns in an activity performed by two different people we have used the Dynamic Time Warping algorithm, which allows the detection of patterns and alignment of two different time sequences.

Index Terms— Pose estimation, dynamic time warping, computer vision

1 INTRODUCCIÓN

Alexa, ¿qué tiempo hace hoy? , Ok Google, activar alarma a las 8:00 son frases que se utilizan a diario en la vida de las personas y esto no sería posible sin el Machine Learning. En la vida cotidiana se utilizan multitud de aplicaciones que aplican Machine Learning para ofrecer una mejor experiencia o resultados más similares a los que podría dar una persona, y es que en los últimos años la evolución del Machine Learning ha permitido que campos como el procesamiento del lenguaje natural o computer vision desarrollen aplicaciones cada vez más complejas y potentes.

Un claro ejemplo de procesamiento del lenguaje natural es una de las aplicaciones más utilizadas por los estudiantes cuando se está realizando un trabajo y se necesita traducir un texto. Para ello, se acude a traductores como el Google Translate [1] o Deepl [2]. Estos traductores han ido mejorando mucho en los últimos años gracias al Machine Learning, pasando de ofrecer una traducción directa a una respuesta más estructurada según el contenido del texto. Esto es posible a que la inteligencia artificial que utilizan ya no sólo traduce el contenido del texto, sino que además les permite comprenderlo para ofrecer una mejor traducción.

Otro ejemplo es el Autopilot de Tesla [3]. Una tecnología muy emergente y que sin la evolución de la visión artificial no sería posible la conducción autónoma. Para ello, se utiliza un sistema de cámaras en el automóvil que permiten captar imágenes de todo su alrededor. Una vez procesadas, facilita a la inteligencia artificial la obtención de datos como señales de tráfico, detección de líneas de la calle, coches u obstáculos que se encuentran a su alrededor y a la distancia que se encuentran para poder tomar decisiones de forma autónoma y conducir el coche con seguridad.

La visión artificial o visión por computador no siempre ha sido tan potente como lo es ahora. En los inicios de siglo XXI se realizaban concursos, en los que se retaba a los investigadores a leer bases de datos de imágenes y clasificarlas según su contenido. Una de las bases de datos más conocidas con la que se realizaban estos retos y que sirve a los investigadores para comprobar la eficacia de sus proyectos es ImageNet [4]. Gracias a estos concursos en el año 2012 se realiza un gran avance en la utilización de redes neuronales convolucionales. Alex Krizhevsky juntos con sus compañeros presenta un sistema de redes convolucionales conocido como AlexNet [5], que les permite clasificar en 1000 clases diferentes los 1,2 millones de imágenes de alta resolución de la base de datos de ImageNet con un porcentaje de error de 17%, mejorando así el estado del arte actual en cuanto a la clasificación de imágenes. Esto da el pistoletazo de salida para el uso de las redes neuronales convolucionales en la visión por computador atrayendo las miradas de grandes empresas, que aprovechando los recursos que tienen en sus manos han llevado estos sistemas a otro nivel en el que prácticamente obtienen un 99% de aciertos al clasificar imágenes.

Dentro de la visión por computador encontramos que una de sus utilidades más destacables es la pose detection.

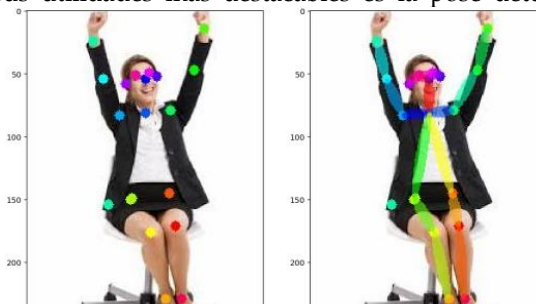


Figure 1 - Detección de puntos clave con pose detection

Esta técnica permite mediante el estudio de las imágenes o vídeos predecir, localizar y extraer los keypoints del individuo como se muestra en la Figura 1. Esta capacidad abre un gran rango de posibilidades para distintas aplicaciones y en múltiples entornos. Es el caso, por ejemplo, de la aplicación creada por Amazon para mantener la distancia de seguridad entre sus empleados debido a la COVID-19 [6]. En esta aplicación mediante la detección de personas en las imágenes obtenidas por las cámaras y la creación de los puntos de unión de las articulaciones del individuo, es posible calcular una distancia de seguridad y, en el caso que no se cumpla dicha distancia, se comunica con un elemento visual conforme no se está manteniendo la distancia de seguridad necesaria.

En la robótica es habitual el uso de pose detection para el uso de tareas colaborativas humano-robot, como es el caso del paper *Formulating Intuitive Stack-of-Tasks with Visuo-Tactile Perception for Collaborative Human-Robot Fine Manipulation* [7], en el que se utilizan cámaras para trackear a las personas cercanas y mediante human pose estimation obtener los keypoints de la persona. De esta manera el robot puede interactuar de manera segura con la persona al manipular objetos porque puede procesar dónde están los límites y en caso necesario no interactuar para proteger al empleado.

En el artículo *Using Kinect™ sensor in observational methods for assessing postures at work* aplican la tecnología de pose detection a la Ergonomía en el trabajo [8]. En este proyecto se presenta una tecnología capaz de detectar las malas posturas de distintos individuos durante su jornada laboral en oficinas mediante cámaras Kinect con el objetivo de mejorar la ergonomía y reducir las lesiones causadas por la mala postura.

La utilización de la visión por computador en el deporte está permitiendo añadir valor al contenido consumido por las grandes audiencias. En el caso proyecto de Leonardo Citraro (2020) *Real-Time Camera Pose Estimation for Sports Fields* [9], mediante el uso de cámaras permite convertir la escena 2d a 3d detectando en ella a los jugadores del campo de juego. Esto permite obtener métricas de los jugadores de forma automática y ofrecer al espectador un valor añadido. Otra utilidad en el deporte es el ejercicio guiado como el que propone Hou (2017) en su proyecto *Dancing like a superstar: Action guidance based on pose estimation and conditional pose alignment* [10], en el que utilizan el posicionamiento de los keypoints de la persona, obtenidos con pose estimation gracias a cámaras Kinect, para crear una plataforma en la que ayuda a mejorar los pasos de baile de manera interactiva.

Estos ejemplos son solo algunos de los avances en el campo de la visión por computador que están permitiendo automatizar procesos. De esta forma, se facilita el trabajo a las personas y a su vez se adapta a los diferentes entornos.

En este trabajo profundizaremos sobre el uso de human pose estimation dentro del ámbito del deporte para analizar qué sistemas se adaptan mejor a los problemas que plantea este entorno, para así con uno de ellos implementar un sistema de detección de patrones que permita visualizar cuando se está realizando un movimiento no esperado o mejorable técnicamente durante la realización de una disciplina deportiva.

2 OBJETIVOS

En este apartado se tratarán los objetivos del proyecto con el fin de tener una visión general del trabajo a realizar. Para ello se dividirán los objetivos en apartados para poder explicar con más detalle cada uno de ellos y mostrar en el orden que deberán ser realizados, ya que el cumplimiento de un objetivo dependerá de su sucesor.

2.1 Analizar pose estimation en el deporte

El deporte es uno de los ámbitos en los que hoy en día ya se están utilizando técnicas de pose estimation. Algunas de sus aplicaciones actuales son obtener estadísticas en tiempo real de los jugadores en un partido de fútbol, ayudar en el entrenamiento de los deportistas o recrear jugadas en tres dimensiones. Debido a las múltiples aplicaciones y los diferentes contextos en los que se realiza el deporte, nuestro principal objetivo será comparar las librerías de pose estimation en el ámbito Deportivo.

2.3 Dataset en entorno real

Una vez hemos analizado qué sistemas de pose estimation se adaptan mejor al deporte, se aplicará uno de ellos a un entorno real con el objetivo de obtener todos los datos posibles de la realización de una práctica deportiva.

Como objetivo opcional se realizará un pequeño evento deportivo para la recopilación de datos de forma masiva. Debido a la pandemia que se está sufriendo es posible que este objetivo no se pueda llevar a cabo, es por eso por lo que se indica como objetivo opcional.

2.4 Detección de patrones

Como objetivo final después de crear un dataset con datos reales de la práctica de una disciplina deportiva, se aplicará un algoritmo para detectar posibles fallos durante la práctica de esta, mostrando los fotogramas en los que ha sucedido. De esta manera podremos completar el círculo de analizar los sistemas de pose estimation, escoger el que mejor se ajusta al deporte y finalmente utilizar el dataset para observar fallos en la técnica.

3 METODOLOGÍA

La metodología utilizada para el desarrollo del trabajo de fin de grado será Kanban [11] con algunos principios de metodologías ágiles. Esta elección viene debida a la experiencia obtenida durante el grado en diferentes asignaturas en las que hemos practicado con estas metodologías,

donde el objetivo es organizar y gestionar las tareas de equipos de trabajo. Algunas de estas metodologías necesitan de varios integrantes para un buen funcionamiento y en este caso se trata de un proyecto unipersonal. Es por esto que la estrategia de trabajo escogida ha sido Kanban para gestionar y organizar las tareas del proyecto.

Para implementar Kanban se utiliza un tablero dividido en diferentes columnas, en este caso TO DO, IN PROGRESS y RESOLVED, que hace referencia al estado de una tarea de tal manera que se pueda ver el estado actual del proyecto en todo momento. Para implementar Kanban se utilizará la plataforma Monday[13], que permite crear un tablero con nuestras tareas.

Además, se implementará el principio de trabajo iterativo, en el que mediante sprints de 1 o 2 semanas se irá evolucionando el proyecto y permitirá organizar el trabajo necesario para cada fase del proyecto. Al finalizar cada iteración se realizará una reunión con el tutor para verificar el estado del trabajo y planificar próximos pasos de los siguientes sprints.

Como modo de comunicación se utilizará Microsoft Teams para mantener reuniones con el tutor, posibilitando crear un canal de comunicación a modo Sandbox en el que poder almacenar documentos de interés. De la misma manera se utilizará un canal a modo de portfolio para tener un registro del progreso del trabajo realizado durante toda la evolución de este.

4 PLANNING

En este apartado se detallarán las cargas de trabajo y las tareas a realizar del proyecto para los próximos meses. Para ello se tendrá en cuenta el calendario disponible para realizar el proyecto y los informes a realizar durante su transcurso.

Si se dividen todos los meses del calendario escolar para realizar el proyecto en semana, se obtiene un total de 22 semanas para trabajar en él.

Las semanas importantes que se tendrán en cuenta para añadir a la planificación por entregas de un informe o punto de control son las siguientes:

- Semana 4 (08/03 – 14/03): Informe inicial
- Semana 10 (19/04 – 25/04): Informe progreso 1
- Semana (24/05 – 30/05): Informe progreso 2
- Semana 18 (14/06 – 20/06): Propuesta informe final
- Semana 19 (21/06–27/06): Propuesta presentación final
- Semana 20 (28/06 – 04/07): Entrega informe final
- Semana 22 (12/07 – 18/07): Defensa y presentación del proyecto

Según estas semanas y las entregas de control que se tienen que realizar, se dispone aproximadamente de 13-14 semanas para trabajar en el núcleo del proyecto, ya que las últimas 5 semanas están destinadas a pulir el informe

final y preparar la presentación y las primeras 5 semanas son para organizar y plantear el kickoff del proyecto.

Conociendo las semanas disponibles para trabajar en el proyecto y las fechas de entrega de informes o puntos de control necesarios para realizarlo, se detallan las tareas a realizar de cada fase del proyecto.

4.1 Tareas del Proyecto

El proyecto tiene fases muy diferenciadas que son el inicio del proyecto, el análisis de la pose estimation en el deporte, la creación de un dataset en un entorno real, la detección de anomalías y cierre del proyecto. A continuación, se describen las tareas que contiene cada fase del proyecto.

Inicio del proyecto

Esta primera fase se centra en organizar las primeras reuniones con el tutor con el fin de marcar los objetivos iniciales del proyecto para poder crear el informe inicial, el cual marcará las bases del proyecto.

Analizar pose estimation en el deporte

La fase de análisis de sistemas de pose estimation comprende la primera fase de desarrollo, en ella se escogerán 3 sistemas de pose estimation para poder prepararlos y realizar una prueba de precisión y una de tiempo de inferencia que permitan decidir con que sistema de pose estimation aplicar el resto de las fases. Una vez escogido sistema de pose estimation se crea un script como base para la detección de patrones, en el cual se implementa un sistema de detección y tracking utilizando los outputs recibidos tras aplicar el sistema de pose estimation sobre un vídeo.

Crear dataset en un entorno real

Esta fase se centra en la creación de dos dataset de una actividad deportiva con el fin de preparar los datos necesarios para ejecutar la fase final de detección de patrones. El primer dataset se obtiene de una grabación online de una competición oficial. El segundo dataset se crea un dataset propio en un entorno controlado.

Detección de patrones

Última fase del desarrollo del proyecto se centra en implementar un sistema de detección de patrones mediante el algoritmo Dynamic Time Warping. En esta fase se analizan y se extraen las conclusiones de los resultados obtenidos al aplicar la detección de patrones en los datasets creados en fases anteriores.

Cierre del proyecto

Fase final del proyecto, en la que se centra en finalizar obtener las últimas conclusiones para cerrar el informe final, presentación y la preparación de la presentación del trabajo realizado.

5 DESARROLLO

La sección de Desarrollo permite explicar las fases del proyecto de una manera más detallada. En cada una de ellas se puede ver el trabajo que se ha realizado para cumplir con los objetivos marcados en la planificación.

5.1 Comparativa sistemas pose estimation

En esta primera fase del desarrollo se han escogido tres sistemas de pose estimation para hacer la comparativa y analizar sus puntos fuertes. Los sistemas de pose estimation escogidos son OpenPose [14], Detectron 2 [15] y AlphaPose [16]. OpenPose ha sido escogido debido a su popularidad en los últimos años, AlphaPose porque es el sistema de pose estimation con el método top-down más popular y por último Detectron2 debido a que una gran empresa como Facebook está detrás de este sistema de pose estimation.

Para realizar esta tarea se ha utilizado Google Colab, un producto de Google Research que nos permite escribir y ejecutar código desde el navegador utilizando los recursos computacionales proporcionados por parte de Google de forma gratuita y sin necesidad de una configuración previa. Gracias a estas características encontramos multitud de archivos colaborativos en el entorno del Machine Learning que nos permiten tener una primera visión del funcionamiento y resultados sin necesidad de pasar por el proceso de instalación y configuración en una máquina propia.

En el caso de los sistemas de pose estimation seleccionados, los desarrolladores facilitan un archivo de Google Colab en el que ya contiene las librerías y requisitos necesarios para utilizar su sistema de pose estimation con un conjunto de ejemplos básicos para obtener una primera interacción. Estos archivos colaborativos se han preparado para poder realizar las mismas pruebas comparativas.

Las pruebas que se realizan son una de precisión y otra de tiempo de inferencia donde se tiene en cuenta la cantidad de personas en las imágenes. Para las dos pruebas se utilizará el dataset oficial de COCO, uno de los datasets más utilizados para la evaluación y puntuación del rendimiento del estado del arte en los modelos de visión por computador.

5.2 Extracción de keypoints

En la primera fase del Proyecto se han utilizado varios sistemas de pose estimation como son OpenPose [14], Detectron 2 [15] y AlphaPose [16] para obtener los keypoints de las personas que aparecen en los fotogramas de los vídeos.

Las pruebas se han enfocado en extraer y mostrar los keypoints de un video, como el mostrado en el ejemplo de la Figura 3.

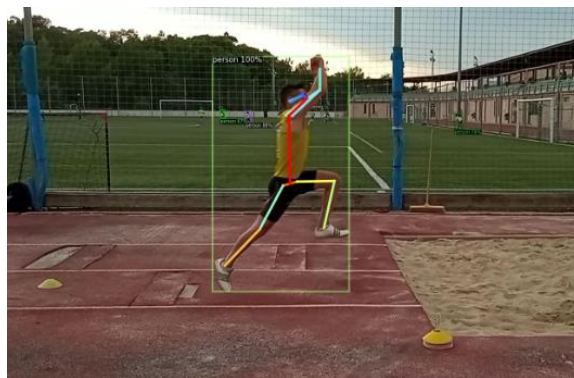


Figure 3 – Ejemplo pose estimation aplicado con Detectron2 [15]

Todos los sistemas de pose estimation testeados devuelven un vector de valores con la posición x, y del keypoint en el fotograma. Además, algunos añaden la precisión del punto para tener una referencia de cuan fiable es la posición obtenida. La posición del conjunto de valores x, y y precisión en el vector hace referencia a un punto del esqueleto formado por los keypoints, para reconocer a qué parte del esqueleto hace referencia. Para ello se necesita consultar la documentación de cada sistema de pose estimation. En el ejemplo mostrado se utiliza OpenPose con el formato BODY_25 que representa el esqueleto de la persona con 25 keypoints (Figura 4).

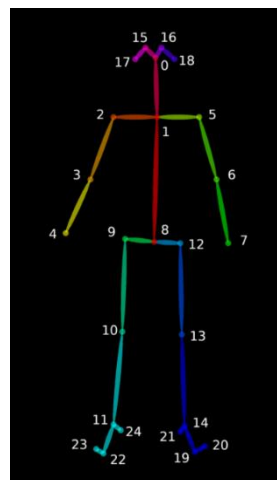


Figure 4 – Estructura de keypoints BODY_25

5.2 Tracking y detección de personas

Un problema observado durante las pruebas de los sistemas de pose estimation es la necesidad de relacionar los conjuntos de keypoints del atleta de un fotograma con el fotograma anterior. Para ello se ha implementado 2 sistemas de detección y tracking de personas para localizarlas durante todo el video con el fin de relacionar cada conjunto de keypoints con la persona pertinente.

El primer sistema de multitasking implementado utiliza como base las librerías de Dlib para trackear y hacer la

correlación entre detecciones, la librería OpenCV [17] para la gestión y tratamiento de frames del video y crear una red neuronal con el modelo MobileNetSSD para la detección de individuos.

El funcionamiento está basado en 3 fases, la primera de ellas es la obtención de un fotograma del vídeo, en segundo lugar, la detección de personas en el fotograma y en último lugar el seguimiento de las personas detectadas. En caso de no tenerlas trackeadas se crea un tracker y si ya existe se actualiza el tracker para situar la posición obtenida en el fotograma actual. Finalmente, para visualizar el seguimiento de cada persona detectada en el vídeo se encuadra cada detección, como en el ejemplo de la Figura 5.



Figure 5 – Detección y tracking de persona

El segundo sistema de detección y multitracking utilizado ha sido la librería Norfair[18] de la empresa Tryolabs. Esta librería permite utilizar los keypoints obtenidos del sistema de pose estimation como detección. Para el tracking utiliza el algoritmo distancia euclidiana para detectar el punto más cercano entre frames y Kalman filter [19] para así predecir donde se situará el punto en el siguiente fotograma.

Con esta librería se han obtenido mejores resultados que con el sistema anterior. Esto es debido a que la detección de la persona con la red neuronal MobileNetSSD en algunas fases del salto no detecta la persona y en cambio el sistema de pose estimation prácticamente devuelve todos los keypoints durante la actividad. Este último punto y su fácil implementación ha hecho que sea el sistema seleccionado.

5.3 Detección de patrones

En esta sección del proyecto se detalla la fase de detección de patrones. Para ello, se ha utilizado el algoritmo Dynamic Time Warping o DTW que permite la detección de patrones y alineación de secuencias temporales. Un ejemplo aplicado de DTW es Speech recognition, en el que mediante las ondas de sonido producidas al hablar permite encontrar patrones de similitud independientemente

de la velocidad de pronunciación o acento, como se puede observar en el artículo [20].

En el ejemplo de la Figura 7 se muestra la gráfica del audio de dos personas pronunciando la frase, “Doors and corners, kid. That’s where they get you”. El audio 1 será utilizado como la base para la detección de patrones y el audio 2 es con el que se va a comparar. El audio 2 se ha exagerado la entonación y pronunciación, provocando que los dos audios tengan longitudes distintas y las palabras se pronuncien en diferentes momentos en el tiempo.

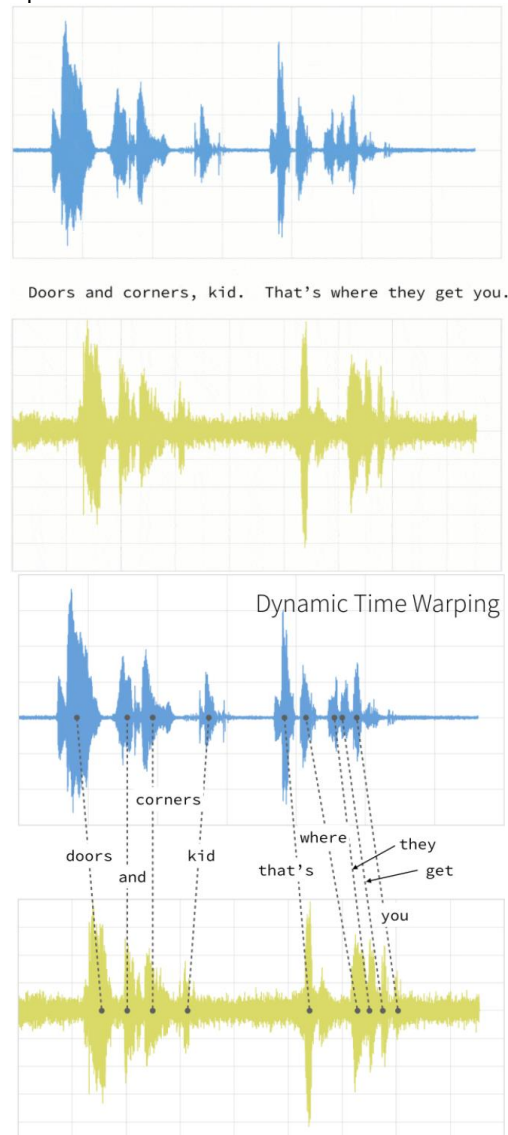


Figure 7 –Ejemplo Dynamic Time Warping para la frase “Doors and corners kid that’s where they get you”

Para aplicar el algoritmo DTW en este caso se utiliza la librería de Python Fastdtw [21]. Al invocar el método Fastdtw se le envía por parámetro en primer lugar el audio base y en segundo lugar el audio 2 con el que será comparado. El método devuelve dos valores, la distancia mínima y un vector con el camino mínimo entre los dos audios. El camino mínimo permite marcar los pun-

tos en común de los dos audios. De esta forma se puede observar como la correlación de los momentos en que se pronuncia una palabra coinciden en los dos audios pese a la entonación al pronunciar la misma frase.

Aplicar el algoritmo DTW al problema del proyecto posibilita comparar una misma actividad realizada por dos personas. Esto permite aplicar una comparación eficaz de una actividad realizada por un atleta profesional con una realizada por un amateur y detectar las fases de la actividad que tienen en común y aquellas que se pueden mejorar.

Para la detección de patrones en el problema se utilizan los keypoints de los atletas al realizar un salto de longitud. Los keypoints han sido obtenidos en fases anteriores utilizando el sistema de pose estimation OpenPose y tratados para obtener solo los puntos de interés del atleta. En este caso, se calcula el ángulo de la pierna izquierda del saltador durante el ejercicio mediante los puntos de la cadera, rodilla y tobillo. Con el objetivo de reducir posibles errores producidos por la pérdida de algún keypoint en alguno de los fotogramas, se aplica smooth average. Con ello se tiene en cuenta una cantidad determinada de datos de fotogramas anteriores, en este caso 5 fotogramas, y se hace una media para obtener el valor final.

Una vez aislados y preparados los datos necesarios para comparar las secuencias, se aplica el algoritmo DTW utilizando la librería Dtwalign [22]. En la Figura 8 se muestra la matriz de costes y el camino mínimo encontrado que indica la similitud entre las dos secuencias. Con esta información se crea la correspondencia de fotogramas del vídeo de los atletas obteniendo un vídeo comparativo de la actividad realizada.

6 RESULTADOS

En esta sección se detalla el proceso seguido para obtener los resultados en cada una de las fases del proyecto.

6.1 Análisis sistema pose estimation

Para los tres sistemas de pose estimation escogidos se han adaptado un archivo de Google Colab para realizar las pruebas de precisión y de tiempo de inferencia. Se ha utilizado la librería Pycocotools para acceder al dataset oficial de COCO y obtener imágenes y sus anotaciones.

Utilizando el dataset COCO de 2015 se han obtenido las siguientes precisiones medias variando el threshold utilizado.

Tabla 1: Precisión sistemas pose estimation

Method	AP @0.5:0.95	AP @0.5	AP @0.75	AP me- dium	AP large
OpenPose	61.8	84.9	67.5	57.1	68.2
Detectron	67.0	88.0	73.1	62.2	75.6
AlphaPose	73.3	89.2	79.1	69.0	78.6

En la prueba de tiempo de inferencia se ha comparado el tiempo tardado en hacer la inferencia por cada sistema de pose estimation. Las imágenes han sido clasificadas según la cantidad de personas que se muestran para ver cómo afecta en el rendimiento a cada sistema en las diferentes situaciones.

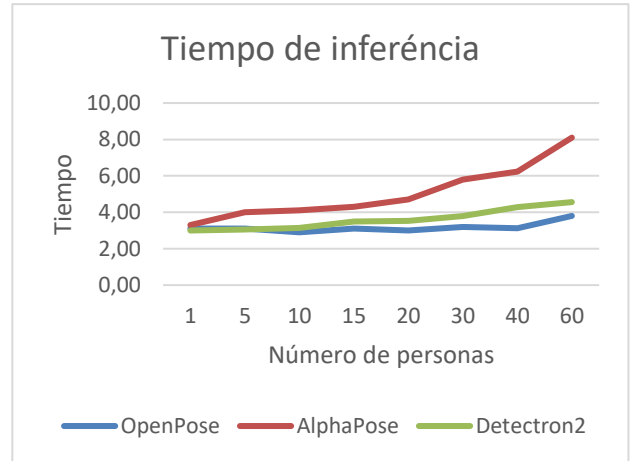


Figure 8 –Gráfica tiempo inferencia

En la Figura 8 se puede observar los resultados obtenidos tras aplicar la prueba de inferencia. AlphaPose adquiere un movimiento ascendente mucho mayor que el resto de los sistemas a medida que aumenta el número de personas por imagen, esto puede ser debido a que aplica el método top-down en el que primero detecta a las personas para luego extraer los keypoints de cada una, al contrario que los otros dos sistemas que usan el método bottom-up detectando primero los keypoints para finalizar relacionándolos entre ellos y obtener la detección.

Debido a que OpenPose y Detectron2 obtienen en ambas pruebas unos resultados similares, se ha acabado escogiendo OpenPose como sistema de pose estimation para aplicar durante el proyecto debido a su popularidad, la gran cantidad de documentación disponible y que pone dispone de la estructura de keypoints BODY_25, la cual permite añadir más información para la detección de patrones.

6.2 Comparativa ángulo pierna de batida

En esta primera comparativa se analiza el comportamiento de la pierna izquierda de cada saltador ya que es la pierna predominante al realizar la batida. Se ha seleccionado la pierna de batida para realizar el análisis del salto porque aporta mucha información durante el ejercicio y permite obtener un análisis significativo del ejercicio reduciendo la cantidad de información necesaria para realizar el estudio. Para ello se han utilizado los keypoints de la pierna izquierda, cadera, rodilla y tobillo, para calcular el ángulo durante todo el salto.

Se analizan los resultados obtenidos de comparar los saltos de dos atletas profesionales y el salto de un atleta aficionado con el de un profesional. Para ello en el primer caso se han seleccionado dos saltos realizados el mismo día y en la misma competición, permitiendo que los ángulos de la cámara y la perspectiva sea la misma para los dos saltadores. Para el segundo caso se utilizan dos saltos realizados en competiciones diferentes y con perspectivas y ángulos de cámara diferentes.

Para poner a prueba la detección de patrones y en especial el algoritmo Dynamic Time Warping la longitud temporal de los saltos difiere dos segundos una de otra, comprendiendo para todos casos comparados las fases de carrera, batida, vuelo y caída de un salto de longitud [23].

Para esta comparativa se ha utilizado un script en el que tras pasar por las fases de extracción de keypoints y tracking permite calcular y comparar el ángulo de la pierna izquierda de los saltadores.

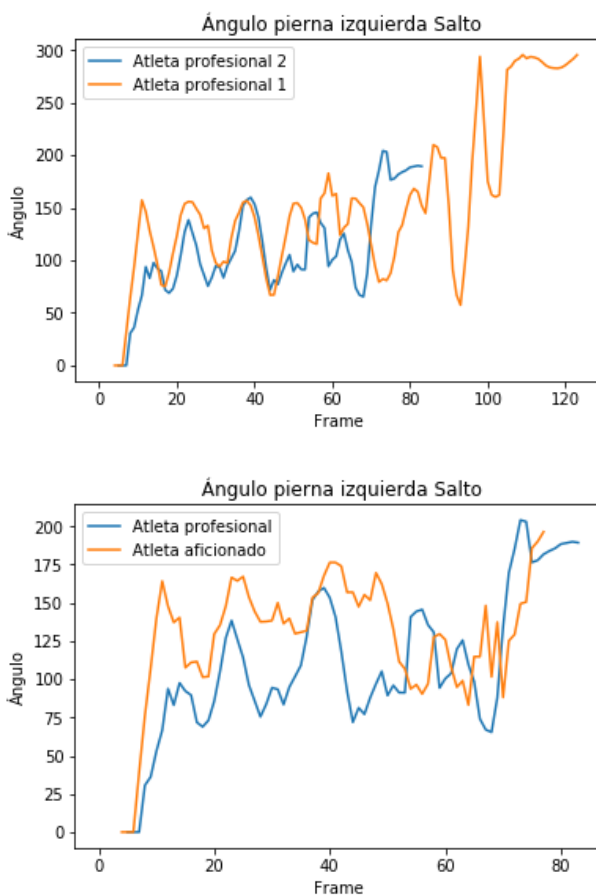


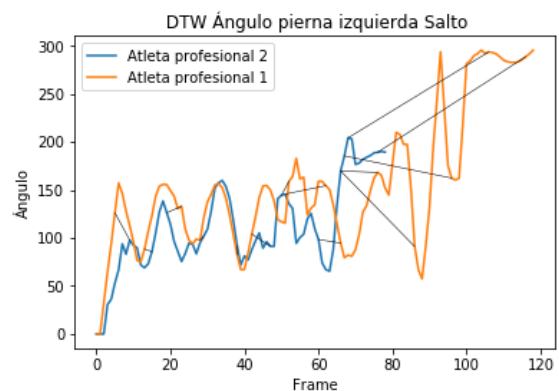
Figure 9 –Comparativa ángulo pierna izquierda atletas

En la Figura 9, se observan los resultados obtenidos. En los dos casos se aprecia como el ángulo tiene un comportamiento distinto a lo largo de todo el salto. Este comportamiento nos puede ayudar a detectar las diferentes fases del salto a simple vista. Durante la fase de carrera el án-

gulo tiene un comportamiento sinusoidal constante en el mismo rango hasta aproximadamente el fotograma 40 -50 donde finaliza y se realiza la fase de batida. La fase de batida es poco apreciable ya que es muy corta, pero se puede identificar como el momento en el que se rompe con la tendencia adquirida por la fase de carrera. A partir de ese momento el rango de los ángulos se acorta y pasa a oscilar más rápidamente, es aquí donde podemos detectar la fase de vuelo. Esta fase tiene la particularidad de que existen varias técnicas de vuelo. En el caso de la comparativa los atletas utilizan técnicas diferentes. Uno de ellos utiliza la técnica de extensión en la que tras la fase de batida se busca extender brazos y piernas para agrupar rápidamente brazos y pies en el mismo punto. En el caso del otro saltador utiliza la técnica de pasos en el que el saltador simula realizar un seguido de pasos en el aire. Al ser técnicas diferentes se pueden observar diferentes patrones en los ángulos, pero en los dos casos se cumple que se reduce el rango del ángulo y oscilaciones más seguidas hasta el fotograma 70 aproximadamente en el caso de unos de los saltadores y 80-85 en el otro.

Por último, se realiza la fase de caída, en la que el atleta intenta crear un efecto muelle al tocar con los pies en la arena del foso buscando que su cadera acabe lo más cerca posible donde han caído sus pies, reduciendo así la marca que dejará en el foso y optimizando la caída. Esta fase es muy corta, pero se puede apreciar como el rango del ángulo aumenta siendo el más grande de toda la fase del salto.

Una vez analizadas y comprendidas las relaciones del ángulo de la pierna de batida con las diferentes fases del salto de longitud se aplica a estas mismas series de datos el algoritmo DTW. Para facilitar la comprensión se ha mostrado gráficamente el ángulo de la pierna izquierda del atleta a lo largo del salto y cada 10 fotogramas se ha unido el punto del salto base con su correspondiente con el salto que se está comparando según el resultado obtenido al aplicar el algoritmo.



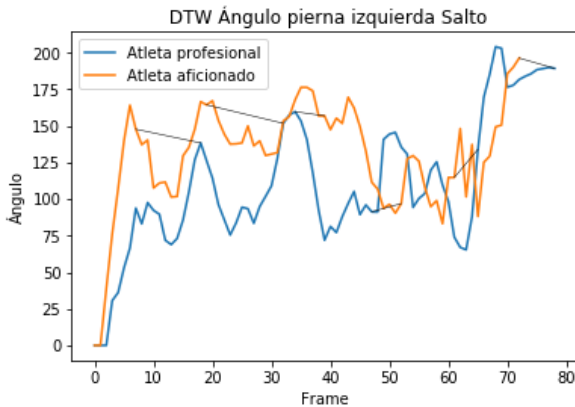


Figure 10 –Comparativa DTW ángulo pierna izquierda atletas

En la comparativa de la Figura 10 se utiliza el algoritmo DTW obteniendo resultados dispares. En la fase de carrera comprendida entre los fotogramas [0,40] y en la fase de caída en los últimos 20 fotogramas de cada saltador podemos ver que a pesar de la diferencia de tiempo en la fase de caída el algoritmo relaciona puntos correctamente. En cambio, en la fase de vuelo como se ha comentado en el análisis anterior, los dos atletas realizan una técnica diferente y el detector de patrones busca puntos en los que ambos atletas tienen ángulos similares.

6.2 Comparativa avanzada

En esta segunda comparativa se van a analizar la fase de carrera de dos atletas aficionados pero esta vez utilizando todos los keypoints que nos proporcionan los sistemas de pose estimation. Esto permitirá añadir más información como la posición de los brazos o la inclinación del cuerpo para ayudar y mejorar la detección de patrones.

El dataset utilizado para esta comparativa es un dataset propio grabado durante una sesión de entrenamientos. Esto ha propiciado que los saltos a comparar estén grabados con un mismo ángulo y distancia, obteniendo así un plano 2D que permite comprender fácilmente la representación gráfica en 3D de todos los keypoints del atleta durante el vídeo como se puede observar en la Figura 11.

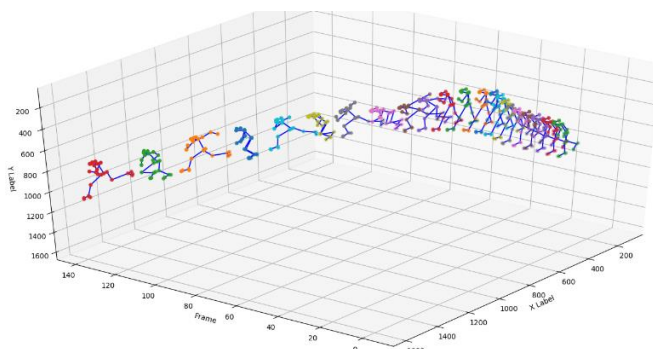


Figure 11 –Representación gráfica en 3D de los keypoints del atleta durante la fase de salida

Para la representación gráfica de los keypoints del atleta durante la actividad se han seguido las mismas fases que en la comparativa anterior, pero utilizando un gráfico en 3D. En este caso al añadir más información a la comparativa dificulta la observación de los gráficos si se mostrara la información de los dos atletas. Para solucionar este problema se ha utilizado el camino mínimo obtenido al aplicar el algoritmo DTW, el cual si mostramos gráficamente mediante una matriz de distancias como en la Figura 12 permite entender el funcionamiento interno del algoritmo DTW. El algoritmo calcula para cada fotograma del vídeo base la distancia euclidiana con el resto de los fotogramas del vídeo con el que se está comparando, obteniendo de esta manera una matriz de distancias. Con la matriz de distancias se escoge uno a uno el fotograma con menor distancia obteniendo así un camino mínimo para recorrer la matriz de distancias. Gracias al camino mínimo se crea un vídeo en el que se muestra el fotograma del vídeo base con el fotograma del vídeo que se está utilizando para comparar la actividad realizada que ha asignado el algoritmo DTW.

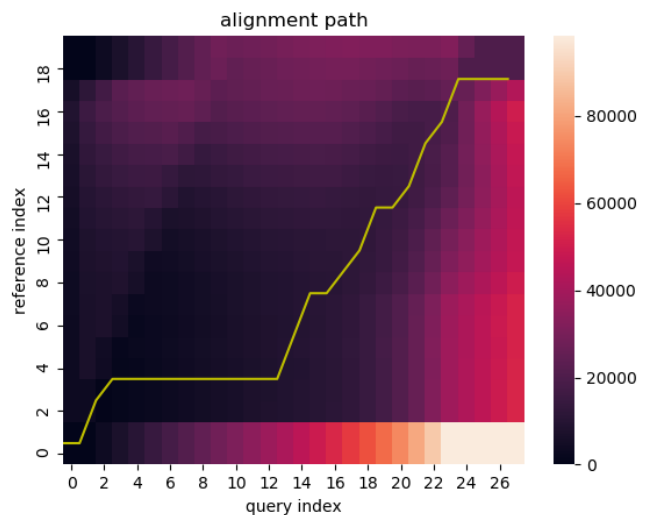


Figure 12 – Matriz de distancias y path obtenidos tras comparar dos atletas

Al observar los resultados obtenidos se puede ver que pese a la diferencia de duración de los dos vídeos y que los dos atletas utilizan piernas distintas para lanzar el inicio de su fase de carrera, el algoritmo detecta de forma correcta tanto el inicio de la carrera como los apoyos realizados durante esta, como se puede observar en la Figura 13.



Figure 13 – Comparativa avanzada dos atletas en fase carrera

7 CONCLUSIONES

A lo largo del proyecto se ha implementado un proceso que utiliza los datos obtenidos de los sistemas de pose estimation para analizar y detectar patrones de una actividad física. En este caso se ha utilizado para analizar la prueba de salto de longitud obteniendo de esta forma dos tipos de conclusiones.

7.1 Sistema pose estimation

El sistema pose estimation escogido puede afectar al resultado y rendimiento de una posible aplicación final. Para escoger el más adecuado se tiene que tener en cuenta las características del tipo de deporte y las personas que pueden aparecer durante el transcurso de la actividad, ya que como hemos visto dependiendo la cantidad de personas que aparecen durante la actividad puede variar el tiempo en aplicar la inferencia. En este caso se ha visto que Openpose ofrece un rendimiento estable sin depender de la cantidad de personas que aparecen en pantalla a diferencia de AlphaPose.

7.2 Detección de patrones

En el proceso previo de preparación de los datos obtenidos con el sistema de pose estimation se ha tenido que crear un sistema que relacione los conjuntos de datos de cada persona de fotograma a fotograma. Esto ha sido necesario debido a que los sistemas de pose estimation no tienen sistemas de multitracking aplicados o están en fases tempranas. Este es uno de los aspectos a tener en cuenta en el caso de aplicar un sistema de pose estimation en una aplicación real.

La detección de patrones se ha aplicado mediante el algoritmo DTW. Después de observar los resultados se pueden extraer dos conclusiones:

1. Perspectiva: Para comparar dos actividades es necesario una perspectiva similar para obtener unos buenos resultados. En este caso si se utilizara en un futuro esta

aplicación debería solo aplicarse en actividades en las que se observa al atleta con una vista lateral en 2D reduciendo así la complejidad del problema.

2. Actividad: La actividad a comparar debe ser la misma o compartir parte de la misma actividad. En el caso de la fase de vuelo se puede ver que la detección de patrones no funciona correctamente porque la técnica de los atletas es diferente.

Pese a las complicaciones encontradas el proceso de detección de patrones aplicado ha permitido detectar patrones en dos saltos de longitud realizados por dos personas diferentes.

El código desarrollado durante el proyecto se encuentra disponible en un repositorio de GitHub (https://github.com/pika91/pose_analysis_tfg)

8 AGRADECIMIENTOS

A mi tutor, Coen Antens por todo el conocimiento que ha compartido conmigo y por todo el apoyo y comprensión durante todo el proyecto.

A mi familia, por el apoyo en los malos momentos y en especial a mi hermano Jose Antonio Picazos por todas las charlas y proceso que hemos compartido para sacar adelante nuestros proyectos.

BIBLIOGRAFIA

- [1] AutoML translation [Online] Disponible: <https://cloud.google.com/translate>
- [2] Deepl [Online]: <https://www.deepl.com>
- [3] Tesla, Autopilot and Full Self-Driving Capability [Online]. Disponible: <https://www.tesla.com/support/autopilot>
- [4] Image-net [Online]. Disponible: <http://image-net.org>
- [5] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton (2012). ImageNet Classification with Deep Convolutional Neural Networks [Online]. Disponible : <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>
- [6] Brad Porter (2020). Amazon introduces 'Distance Assistant' [Online]. Disponible: <https://www.aboutamazon.com/news/operations/amazon-introduces-distance-assistant>
- [7] Sunny Katyara, Fanny Ficuciello, Tao Teng, Fei Chen, Bruno Siciliano, Darwin G. Caldwell (2021). Formulating Intuitive Stack-of-Tasks with Visuo-Tactile Perception for Collaborative Human-Robot Fine Manipulation [Online] Disponible: <https://arxiv.org/abs/2103.05676>
- [8] Jose Antonio Diego-Mas, Jorge Alcaide-Mazal (2013). Using Kinect sensor in observational methods for assessing postures at work [Online]. Disponible: <https://doi.org/10.1016/j.apergo.2013.12.001>
- [9] Leonardo Citraro, Pablo Márquez-Neila, Stefano Savaré, Vivek Jayaram, Charles Dubout, Félix Renaut, Andrés Hasfura, Horresh Ben Shitrit, and Pascal Fua (2020). Real-Time Camera Pose Estimation for Sports Fields [Online]. Disponible: <https://arxiv.org/pdf/2003.14109.pdf>
- [10] Hou, Y., Yao, H., Li, H., & Sun, X. (2017). Dancing like a superstar: Action guidance based on pose estimation and conditional pose alignment. [Online] Disponible: <https://ieeexplore.ieee.org/document/8296494>
- [11] Henrik Kniberg & Mattias Skarin (2010). Kanban y Scrum - obteniendo lo mejor de ambos [Online]. Disponible: https://www.academia.edu/download/38261265/KanbanVsScrum_Castellano_FINAL-printed.pdf
- [12] MPII Human Pose Dataset [Online]. Disponible: <http://human-pose.mpi-inf.mpg.de>
- [13] Monday [Online]. Disponible: <http://monday.com>
- [14] Documentación OpenPose [Online]: <https://cmu-perceptual-computing-lab.github.io/openpose/web/html/doc/>
- [15] Documentación Detectron 2 DensePose [Online]: <https://detectron2.readthedocs.io>
- [16] Documentación AlphaPose [Online]: <https://www.mvig.org/research/alphapose.html>
- [17] OpenCv [Online]: <https://opencv.org>
- [18] Norfair [Online] Disponible: <https://github.com/tryolabs/norfair>
- [19] Filtro de Kalman [Online] Disponible: https://es.wikipedia.org/wiki/Filtro_de_Kalman
- [20] Douglas Coimbra de Andrade (2018). Recognizing Speech Commands Using Recurrent Neural Networks with Attention [Online] Disponible: <https://towardsdatascience.com/recognizing-speech-commands-using-recurrent-neural-networks-with-attention-c2b2ba17c837>
- [21] FastDtw [Online]: <https://github.com/slaypni/fastdtw>
- [22] DtwAlign [Online]: <https://dtwalign.readthedocs.io/en/latest/index.html>
- [23] Ginés Pérez Iniesta (2014) El salto de longitud, una modalidad del atletismo muy adaptable para trabajar en Educación Física [Online] Disponible: <https://www.efdeportes.com/efd192/el-salto-de-longitud-en-educacion-fisica.htm>
- [24] Adrian Rosebrock (2018). Multi-object tracking with dlib [Online] Disponible: <https://www.pyimagesearch.com/2018/10/29/multi-object-tracking-with-dlib/>
- [25] Ricardo Portilla, Brenner Heintz and Denny Lee (2019). Understanding Dynamic Time Warping [Online] Disponible: <https://databricks.com/blog/2019/04/30/understanding-dynamic-time-warping.html>
- [26] Berkeley MHAD dataset [Online]: https://tele-immersion.citris-uc.org/berkeley_mhad
- [27] Donald J. Berndt and James Clifford (1994). Using Dynamic Time Warping to Find Patterns in Time Series [Online] Disponible: <https://www.aai.org/Library/Workshops/1994/ws94-03-031.php>