

Pose Analysis

Jose Antonio Picazos Carrillo

Resumen— Este proyecto se centra en el proceso de reconocimiento y clasificación de las acciones que está realizando una persona, también conocido como Human Activity Recognition. Para realizar el reconocimiento se ha hecho uso de un algoritmo de pose estimation, el cual proporciona el esqueleto de la persona y es de ayuda para clasificar la acción que se está realizando. Para entrenar el modelo encargado de realizar la clasificación se ha hecho uso de una red neuronal recurrente como es LSTM y también se ha hecho uso AutoML. Todos estos métodos se han puesto a prueba con datos reales para comprobar su validez.

Palabras clave— Visión por computador, Human activity recognition, clasificador, AutoML, pose detection.

Abstract— This project focuses on the process of recognition and classification of the actions being performed by a person, also known as Human Activity Recognition. To perform the recognition, we have made use of a pose estimation algorithm, which provides the skeleton of the person and helps to classify the action being performed. To train the model in charge of the classification, a recurrent neural network such as LSTM has been used and AutoML has also been used. All these methods have been tested with real data to check their validity.

Index Terms— Computer vision, human activity recognition, classifier, AutoML, pose detection



1 INTRODUCCIÓN - CONTEXTO DEL TRABAJO

El deporte tradicional ha ido evolucionando en los últimos años en lo que respecta a la tecnología utilizada para transmitir: herramientas para el fair play, captación de las mejores jugadas, ayudas para las decisiones de los árbitros, herramientas de coaching y mucho más. Muchas de ellas han sido creadas con la ayuda de sistemas de Machine Learning y de visión por computador que, gracias a los avances de los últimos años, se han convertido en herramientas tecnológicas muy potentes.

Un sistema que los fanáticos del fútbol seguramente conozcan es la asistencia al árbitro por video, más conocido como VAR^[1], un sistema implementado por la FIFA. El VAR consiste en una serie de cámaras situadas en el campo con la finalidad de evitar los errores de arbitraje y, así, conseguir un juego más limpio.

-
- E-mail de contacte: joseantonio.picazos@e-campus.uab.cat
 - Menció realitzada: Computació
 - Treball tutoritzat per: Coen Antens (CVC)
 - Curs 2020/21

También se conocen algunos proyectos de deportistas muy conocidos como el de Xavi Hernandez, con el proyecto de Kognia^[2], o también el de Andrés Iniesta con First Vision^[3]. El proyecto de First Vision consiste en implementar una cámara en las camisetas de los jugadores para poder ver, desde el punto de vista del jugador, sus acciones durante el partido. Esto es realmente útil para aumentar la calidad de las retransmisiones, ya que se añade como experiencia para el

espectador. El proyecto de Kognia se basa en un sistema automático para analizar las tácticas de los equipos en los partidos de fútbol en tiempo real, o bien, después del partido. Para ello, Kognia recolecta datos de los partidos y las sesiones de entrenamiento y, con ellos, mediante la inteligencia artificial, se entrena un modelo de datos. Los datos obtenidos son revisados y complementados por los conocimientos de los entrenadores.

En el ámbito del coaching no existe una gran variedad de herramientas, pero una conocida en el mundo del básquet es HomeCourt^[4], una aplicación que cuenta con el soporte de la NBA. Esta es una herramienta muy interactiva que ayuda a mejorar las habilidades relacionadas con el básquet, ya que realiza la función de entrenador personal desde nuestro móvil. Todo esto lo consigue gracias a la captura y análisis de las imágenes obtenidas de los ejercicios que te propone, a la vez, te ofrece unos consejos guiados para mejorar tu rendimiento. Realiza una detección de la persona y, también, una detección de la pose del jugador para ver su posición a la hora de lanzar el balón o de realizar un ejercicio de habilidad.

Viendo el análisis de la pose que realiza la aplicación de HomeCourt, de la que hemos hablado en el párrafo anterior, nos podemos dar cuenta que en el mundo del deporte la posición del cuerpo del deportista en cualquier disciplina deportiva es muy importante, ya que, en función de esta, se realizará mejor o peor su movimiento o acción. Un ejemplo de deporte podría ser la gimnasia rítmica, en esta disciplina la posición del cuerpo de los gimnastas es muy importante para su puntuación y ejecución de la rutina.

Otro de los deportes en los que la pose del deportista es muy importante, y en el cual nos centraremos, es el atletismo. En el atletismo encontramos diferentes disciplinas como son velocidad, saltos, lanzamientos, carreras de resistencia y marcha. En todas ellas se necesita de una técnica específica para que el atleta ejecute bien el movimiento haciendo que su rendimiento y sus marcas sean buenas. En la ejecución de la técnica se debe tener en cuenta la posición del cuerpo del atleta, ya que una mala posición del cuerpo hará que sus marcas no sean las deseadas, su rendimiento empeore o, incluso, pueda llegar a sufrir alguna lesión. Para mejorar la técnica entra en juego el papel del entrenador, ya que será el que, desde fuera, verá cómo realiza el atleta el salto, por ejemplo, e indicará las correcciones necesarias para que no cometa los mismos errores y pueda corregirlos.

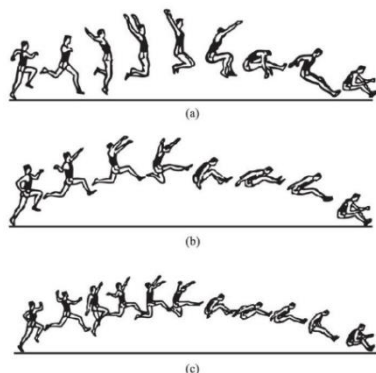


Ilustración 1: Fases salto longitud

Gracias al análisis de la pose también se puede llegar a deducir que actividad está realizando una persona, a esto se le llama Human Activity Recognition (HAR). Esto permite deducir la acción que está realizando una persona en un momento en concreto. Todas las acciones que se realizan tienen una pose en concreto, aunque siempre hay diferencias entre una persona y otra, pero siempre existen unas similitudes. Por ello obteniendo la pose de las personas se puede llegar a clasificar estas acciones. Esto es utilizado en muchas aplicaciones de entrenadores personales, las cuales están grabando el ejercicio que se realiza y gracias a la detección de la pose del cuerpo y un clasificador puede deducir si se realiza la acción correcta y contar el número de repeticiones si es necesario. Un ejemplo de aplicación de este estilo es FitnessAlly^[20], de la empresa Twenty Billion Neurons, el cual es una aplicación que realiza la función de asistente para el entrenamientos de fitness en casa ofreciendo una gran posibilidad de entrenamientos y posibilidades.

2 OBJETIVOS

Observando todo el movimiento que hay relacionado con nuevas iniciativas tecnológicas en el mundo del deporte relacionadas con pose detection y Machine Learning, son muchos los deportes en los que, todavía, no se ha intentado implementar esta tecnología y, por eso, mi objetivo es analizar la pose del atleta en atletismo.

Para ello me centraré en la gran importancia del entrenador y la corrección de la pose del atleta, mi objetivo es,

mediante un detector de pose, analizar la posición del cuerpo del atleta durante su movimiento. Mediante este análisis de la pose se puede crear una herramienta de soporte para los entrenadores que les permitiría obtener posibles mejoras de la pose del atleta en base a la posición analizada. Esta herramienta empezará siendo un prototipo a ordenador, gracias al software StreamLit^[5], que permite crear una base de prototipo para ejecutar todo el entorno. Como objetivo final, si fuese posible, sería conseguir crear un prototipo de aplicación móvil en la que se pueda implementar el pose detection y AutoML^[6], ya que es un dispositivo que llevamos todos en nuestro bolsillo y facilita mucho su uso al no depender de un ordenador.

Las tecnologías y los algoritmos de Machine learning y Deep Learning han ido evolucionando en el paso del tiempo dando a conocer nuevos métodos para la optimización del proceso de aprendizaje y para poder obtener modelos entrenados de una forma más eficiente y rápida. Una de las últimas tecnologías conocidas es el método de autoaprendizaje llamado AutoML. Esta nueva tecnología permite entrenar un modelo de datos y obtener el mejor modelo gracias a la optimización automática de los hyperparámetros. Por ello uno de los objetivos de este proyecto es poner a prueba este novedoso algoritmo de autoaprendizaje. Comprobaré si realmente este método proporciona la mejor opción para el modelo de datos. Este método se utilizará para entrenar el modelo de datos y, así, poder ahorrar una gran cantidad de tiempo en generar el modelo de datos, ya que esta técnica, al ser automática, proporciona un modelo optimizado. El modelo obtenido se compara con otros métodos de aprendizaje, y otros manuales a la hora de optimizar para verificar si, realmente, sería una mejora a la hora de entrenar modelos de datos.

3 METODOLOGIA

Para llevar a cabo el proyecto y gestionar todo el trabajo que este conlleva se ha aplicado una metodología Agile para gestionarlo de forma eficiente, llevar un control de todas las tareas y saber en el punto está el proyecto.

De todas las metodologías existentes se ha aplicado la metodología Scrum, en la cual se trabaja con periodos de tiempo fijos, también conocidos como sprints. Los sprints tendrán una duración de entre 1-2 semanas dependiendo de la fase del proyecto en la que se esté. Al finalizar cada sprint se espera obtener un avance significativo del proyecto aportándole más valor.

Siguiendo la metodología scrum se realizan reuniones semanales con el tutor del proyecto, para ver el avance y planificar el siguiente sprint y decidir qué tareas son prioritarias y cuáles no para que el avance del proyecto sea el correcto. Para llevar a cabo esta metodología se utilizará la herramienta ClickUp^[7], ya que proporciona todo el entorno necesario para crear el proyecto y las tareas a realizar durante cada iteración.

Se realizarán, también, reuniones semanales a través de la aplicación de Microsoft Teams para comprobar el progreso del sprint y el avance de las tareas, esto también servirá para

utilizarlo de “SandBox”, ya que toda la documentación que se va aportando durante el proyecto quedará almacenada en el equipo de Teams.

4 PLANIFICACIÓN

Todo el trabajo del proyecto se ha dividido en distintas fases, cada una de las fases corresponde a un objetivo.

1. Fase análisis AutoML

En esta primera fase, se va a realizar un análisis de los algoritmos de AutoML. Para realizar este análisis, se seleccionarán dos algoritmos AutoML y se realizará una comparativa con métodos más clásicos como son RandomForest, KNN, DecisionTree, etc. Una vez finalizada esta fase podremos seleccionar que algoritmo AutoML será utilizado para la clasificación de poses.

2. Fase reconocimiento de actividades

En esta fase se van a aplicar diferentes técnicas de Human Activity Recognition. Se van a aplicar técnicas más básicas haciendo uso de clasificadores de imágenes para realizar el reconocimiento. Para reforzar este tipo de técnicas se va a hacer uso de un algoritmo de pose estimation, el cual nos va a proporcionar los puntos del esqueleto y nos será de gran utilidad para la fase de clasificación de poses.

3. Fase clasificación de poses

En esta fase se creará un modelo para la clasificación de la pose. Se realizará un entrenamiento de diferentes modelos a partir del algoritmo AutoML seleccionado en la fase de análisis. También se hará uso de un segundo clasificador que nos permita realizar el análisis de la pose en una secuencia de imágenes, posiblemente uno de los algoritmos adecuados para este proceso se una red convulocional.

5 DESARROLLO

5.1 Datasets

Para realizar el desarrollo y la evaluación de los métodos aplicados en este proyecto se han utilizado distintos datasets, estos son:

- UCF101-Action Recognition^[16], se trata de un data set publico comúnmente usado para el reconocimiento de acciones, está compuesto por un total de 101 clases de acciones formadas por vídeos extraídos de Youtube. Algunas de las acciones que dispone este data set son: TaiChi, HighJump, Biking, Skate Boarding y muchas más.
- TREX
- Se ha creado un tercer dataset casero, el cual se han grabado diferentes videos realizando la acción de saltar y caminar.



Ilustración 2: Muestra dataset UCF101

5.2 Human Activity Recognition

Human Activity Recognition es la capacidad de poder clasificar la acción que está realizando una persona en un momento concreto como puede ser, caminar, saltar, sentarse en la silla y muchas otras acciones.

Para poder realizar el reconocimiento de una acción, se debe de seguir un proceso el cual ha sido desglosado en diferentes fases, estas fases son:

- Reconocimiento: es la encargada de realizar la detección de la persona y de aplicar un algoritmo de pose estimation para la detección.
- Clasificación: mediante un modelo de clasificación catalogar los puntos de la pose y obtener la acción resultante.

En la siguiente imagen se puede observar un diagrama del flujo completo que sigue nuestro algoritmo para el reconocimiento y clasificación de acciones.

*/** Diagrama de proceso **/*

5.2.1 Reconocimiento

La primera fase del proceso del algoritmo Human Activity Recognition es realizar el reconocimiento de la persona, en esta fase se han aplicado distintos métodos. En los dos primeros métodos aplicados no se ha tenido en cuenta las características pose de la persona, únicamente el frame. A continuación, se explican más detalles de cada uno de los métodos aplicados.

- Single-Frame CNN

El primer método aplicado es una forma sencilla de poder clasificar las acciones, se trata de una red convulocional de clasificación de imágenes. Este tipo de redes son muy efectivas para la clasificación y segmentación de imágenes.

Para entrenar el modelo, se ha hecho uso del dataset UCF-101, del cual únicamente se han utilizado 6 clases que han sido: HighJump, PoleVault, JavelinThrow, LongJump, ThrowDiscus y Biking. Por lo que respecta a la configuración del modelo de entrenamiento, se han hecho uso de dos capas convulocionales.

/ Grafica curva de loss & accuracy */*

Tras haber entrenado el modelo, se han realizado distintas validaciones utilizando videos extraídos de Youtube. Para realizar la validación hemos procesado el video frame a frame prediciendo la acción que se está realizando en cada frame. Analizando los resultados obtenidos tras procesar diferentes videos, se ha podido observar que en algunos casos se reconocía correctamente la acción que se realizaba en el momento, pero en otros casos era muy inestable. Esta inestabilidad es generada por el hecho de procesar nuestro video frame a frame y no hacer uso de una secuencia de frames.

/ Imagen resultado reconocimiento */*

- Red convolucional (ResNet)

En el segundo método aplicado, también se hace uso de una red convolucional, pero en este caso se ha escogido una red ya entrenada como es el caso de ResNet. Esta red ha sido entrenada con un dataset publico llamado DeepMind Kinetics human action video data set^[19]. Este data set contiene 400 clases de acciones de las cuales se han obtenido 400 clips de vídeo de cada una de esta clase para poder realizar el entrenamiento de la red.

Para poner a prueba este método, se ha seguido la misma estrategia utilizada para el método anterior. Se hecho uso de los mismos vídeos de Youtube y han sido procesados frame a frame. Analizando los resultados obtenidos se ha incrementado el número de aciertos en el reconocimiento de la acción y de una forma más estable que el método anterior. Se ha podido observar que en algunos si generaba algo de confusión y no reconocía correctamente la imagen, esto puede ser debido a la gran cantidad de clases que contiene la red que pueda encontrar muchas similitudes entre distintas acciones.

/* Imagen resultado reconocimiento */

- Pose Detection (OpenPose)

Se han aplicado dos primero metodos en los cuales se ha realizado el reconocimiento de la acción sin tener en cuenta la pose de la persona. Para este tercer método, se ha hecho uso de un algoritmo de pose estimation para reforzar el reconocimiento. Tras investigar entre las diferentes opciones que existen como DensePose, PoseNet, AlphaPose y muchos más, se ha decidido escoger OpenPose. OpenPose es un detector de pose en tiempo real y esto permitirá extraer los puntos (keypoints) del cuerpo detectado.

En la configuración de OpenPose podemos indicar dos modelos de detección diferentes, estos son COCO y BODY_25. En el caso de COCO, nos detecta un total de 18 puntos de la persona detectada, en cambio BODY_25 detecta un total de 25 punto. Para todas las pruebas realizadas se ha decidido utilizar BODY_25, ya que nos proporciona una mayor cantidad de puntos sobre el esqueleto y será de mayor utilidad para reconocer la acción realizada.



Ilustración 5: Ejemplo OpenPose Body_25

Inicialmente se ha decidido realizar un caso simple utilizando un data set casero creado a partir de una serie de vídeos caseros, en los cuales únicamente sale una persona y realiza dos acciones que son saltar y caminar. Tras obtener un total de 14 vídeos se ha realizado un preprocesado ellos, en el cual

se ha extraído frame a frame la imagen posteriormente procesar mediante OpenPose y poder obtener las características (keypoints) de cada una de ellas. En la siguiente tabla, se puede observar el número de vídeos y el total de frames que sean extraído de cada clase.

Acción	Nº Vídeos	Nº Frames
Caminar	8	1034
Saltar	6	464

Tras a ver extraído todos los frames de los vídeos, se ha procesado cada una de las imágenes con el algoritmo de OpenPose para extraer las características de la pose, en este caso los keypoints. Estos keypoints no son de utilidad para crear un data set reuniendo todas las poses detectadas y con ello entrenar nuestro propio clasificador. Para este caso simple se ha utilizado un clasificador clásico como es Random-Forest y posteriormente se ha hecho uso de un algoritmo de AutoML como es AutoKeras. Para los dos clasificadores, los entrenamientos han sido realizados con el mismo porcentaje de entrenamiento y de prueba, en este caso un 30% y 70% respectivamente.

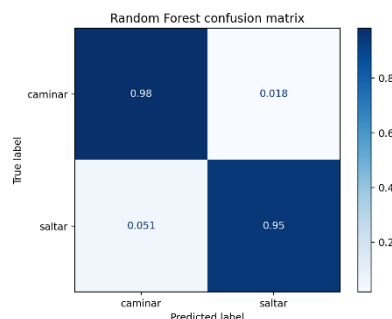


Ilustración 6: Matriz Confusión Random Forest

Ilustración 7: Matriz Confusión AutoKeras

Analizando las matrices de confusión que se han obtenido tras realizar el entrenamiento se puede observar que al tener una clasificación binaria y el hecho de tener más datos de la acción caminar respecto a la acción saltar hace que obtengamos un modelo el cual se adapta mucho a los datos, es decir se está produciendo overfitting. Otro de los puntos a destacar es que sería necesario grabar más vídeos desde diferentes puntos de vista para tener una mayor variedad de ángulos.

Tras realizar un primer caso simple con un dataset propio, se ha decidido realizar nuevamente el mismo proceso, pero en este caso utilizando el data set publico utilizado en los casos anteriores, el dataset UCF-101. Para realizar las primeras pruebas de reconocimiento se ha escogido únicamente una clase de todo el data set, esta ha sido la acción salto de longitud ("Long-Jump").

Al tratarse de un dataset compuesto por vídeo de YouTube, es muy difícil encontrar una gran cantidad de vídeos en los cuales únicamente salga una persona. Esto ha generado una problemática ya que OpenPose extrae la información de todas las

poses detectadas en la imagen y es difícil averiguar qué conjunto de puntos corresponde a la persona que está realizando la acción.



Ilustración 8: OpenPose sin tracking

Para solucionar este problema se ha aplicado un algoritmo de tracking, para poder hacer un seguimiento de la persona que realiza la acción a través de los diferentes frames del vídeo. El algoritmo aplicado para realizar el tracking de la persona ha sido el DeepSort. Este proceso facilita el poder extraer las características de la pose únicamente de la persona que realiza la acción y no de todas aquellas que se encuentran en el vídeo, pero no están realizando ninguna acción. Como se puede observar en la imagen se realiza el tracking únicamente de la persona que realiza la acción.

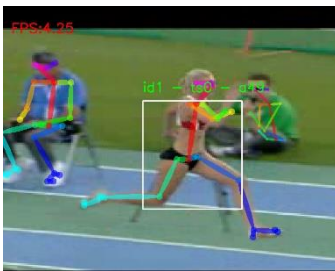


Ilustración 9: OpenPose con DeepSort

5.2.2 Clasificación

En esta segunda fase de clasificación se hace uso del tercer método aplicado en la fase de reconocimiento, que es la pose detección. Se utilizará este método ya que nos proporciona el esqueleto de la persona y esto nos permitirá realizar una clasificación de la pose. Se han aplicado dos métodos para entrenar nuestro modelo, un primer método en el cual se utiliza el algoritmo de autoaprendizaje denominado AutoML y un segundo método de una red convolucional LSTM.

Para realizar el entrenamiento de los dos clasificadores se ha hecho uso del dataset público UCF-101, del cual se han escogido 8 clases y de cada clase se han seleccionado 12 vídeos. Las clases escogidas han sido: archery, basketball, bowling, golfswing, highjump, longjump, playingguitar y shotput. De cada una de las clases se han escogido 8 vídeos.

- AutoML

Para comprobar cómo funcionan y familiarizarse con los algoritmos de AutoML, se ha realizado un primer caso básico con una base de datos pública como es la Breast Cancer Wisconsin (Diagnostic) [14]. Este dataset es utilizado para tareas de clasificación, dispone de un total de 569 filas y 32 atributos,

entre los cuales se encuentra el diagnóstico que es el target y cuenta con dos posibles valores M (Malignant) o B (Benign).

Entre todos los algoritmos de AutoML existentes, se han escogido Auto-Sklearn [13] que corresponde a la famosa librería de Machine learning de Scikit-learn y AutoKeras [15], es un algoritmo de AutoML basado en la librería Keras, la cual hace uso de los algoritmos de Deep Learning. A diferencia de Auto-Sklearn, AutoKeras hace uso de la librería de Tensorflow la cual ayuda a poder realizar el entrenamiento del modelo.

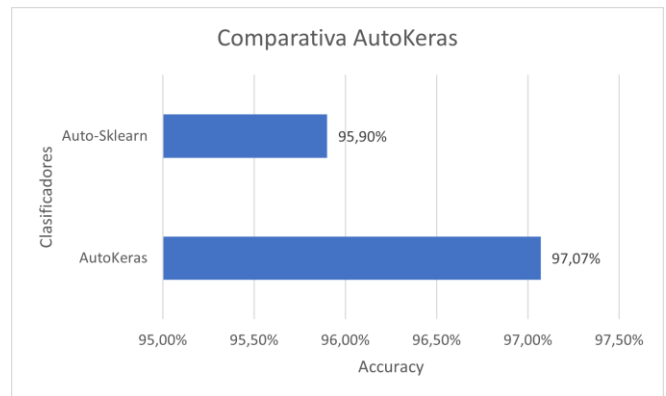


Ilustración 17: Comparativa AutoKeras

Tras realizar el mismo entrenamiento con los dos algoritmos de AutoML seleccionados, se ha realizado una comparativa del mejor accuracy obtenido en el modelo mejor optimizado. Observando la gráfica superior se puede ver que AutoKeras ha obtenido una mayor accuracy y rendimiento respecto a Auto-Sklearn, aproximadamente un 2% más de accuracy. Esta mejora de rendimiento se debe al aplicar algoritmos del Deep Learning, haciendo que nuestro modelo obtenga un mayor rendimiento y mejor optimización.

Para realizar el entrenamiento del clasificador, se ha escogido el algoritmo de AutoKeras, ya que ha sido el que mayor rendimiento y mejores resultados nos ha ofrecido. Para realizar el proceso de entrenamiento se ha seguido un proceso similar a la fase de reconocimiento y hemos entrenado el modelo en base a las características de la pose detectada.

/* Matriz de confusión y mejor modelo obtenido */

Para este clasificador se han realizado pruebas de validación clasificando la pose detectada frame a frame. En el caso de analizar secuencias de video este método de validación no es muy válido ya que una acción no sucede en un frame, si no que transcurre durante varios frames. Por esta razón, se ha implementado una red neuronal LSTM.

- LSTM

Una red neuronal LSTM, es un tipo de red que “recuerda” estados previos, es decir tiene en cuenta el tiempo, y esta información es utilizada para realizar la predicción. Esto nos es de gran utilidad, ya que nos permitirá predecir una acción a partir de una secuencia de frames.

Para crear el modelo LSTM, nos hemos basado en una técnica llamada Sentiment Analysis. Sentiment Analysis es una técnica utilizada para realizar el análisis y clasificación de textos en positivo, neutral o negativo. Aplicando esta técnica en el caso de la clasificación de poses, los keypoints de la pose detectada hacen referencia al texto y las posibles clases hacen referencia a positivo, negativo o neutral.

/* Diagrama modelo */

En este caso para extraer todo el rendimiento de este tipo de redes, en el proceso de validación se han escogido grupos de 10 frames con sus correspondientes keypoints y han sido procesados por el modelo obtenido del entrenamiento. Este tipo de redes nos proporciona una probabilidad respecto a cada una de las posibles clases, tras obtener las probabilidades se ha realizado la clase con mayor probabilidad para cada uno de los frames y se ha escogido la clase con mayor predicción entre los 10 frames.

Analizando los resultados obtenidos tras realizar diferentes análisis de videos, se ha podido comprobar que este tipo de red proporciona un gran rendimiento para el analizar secuencias de video. Esto ha hecho que se obtengas resultados mucho más estables y con mayor precisión al tener en cuenta una secuencia de videos y no un único frame.

6 CONCLUSIONES

Uno de los objetivos es hacer un análisis de los algoritmos de AutoML, para ello se ha hecho una comparativa con diferentes algoritmos de clasificación y comprobar su funcionamiento. Tras realizar una larga serie de pruebas y probar diferentes métodos, he podido comprobar que los algoritmos de autoaprendizaje como AutoKeras y AutoSkit-Learn, proporcionan un buen método de entrenamiento para obtener modelos de forma rápida y sin tener que invertir mucho tiempo en el proceso de optimización de los hyperparametros. Comparado con los algoritmos clásicos de clasificación como Random Forest, Decision Tree y muchos más he visto que es necesario invertir tiempo en una optimización óptima para poder obtener un modelo bien optimizado.

Por lo que respecta al reconocimiento de actividades, se han implementado diferentes métodos, el primer método se basa en la comparación de las imágenes para poder extraer la acción y el segundo método se basa en la pose de la persona que realiza la acción. En el caso del reconocimiento en base a la pose, he visto que tras hacer pruebas frame a frame, en muchos casos al no saber el contexto de la acción que se está realizando no clasifica correctamente la acción. Para solucionar esta falta de contexto, es necesario aplicar redes neuronales que tengan en cuenta el tiempo para poder reconocer la acción.

BIBLIOGRAFIA

- [1] FIFA [Online]. Disponible: <https://football-technology.fifa.com/es/media-tiles/video-assistant-referee-var>
- [2] Kognia [Online]. Disponible: <https://kogniasports.com>
- [3] Bnk To The Future [Online]. Disponible: <https://app.bnktothefuture.com/pitches/2079/first-vision-the-sports-broadcasting-revolution-with-andres-iniesta-and-serge-ibaka>
- [4] HomeCourt [Online]. Disponible: <https://www.homecourt.ai>
- [5] Streamlit [Online]. Disponible: <https://streamlit.io>
- [6] AutoML [Online]. Disponible: <https://www.automl.org>
- [7] ClickUp [Online]. Disponible: <https://clickup.com>
- [8] 2018. Unite the People [Online]. Disponible: <https://www.simonwenkel.com/2018/12/09/Datasets-for-human-pose-estimation.html>
- [9] 2017. A History of Machine Learning and Deep Learning [Online]. Disponible: <https://www.import.io/post/history-of-deep-learning/>
- [10] 2020. Creating a Human Pose Estimation Application with NVIDIA DeepStream [Online]. Disponible: <https://developer.nvidia.com/blog/creating-a-human-pose-estimation-application-with-deepstream-sdk/>
- [11] MPII Human Pose Dataset [Online]. Disponible: <http://human-pose.mpi-inf.mpg.de>
- [12] Hawk-Eye [Online]. Disponible: <https://www.hawkeyeinnovations.com/>
- [13] 2015. Efficient and Robust Automated Machine Learning, Feurer et al., Advances in Neural Information Processing Systems 28 (NIPS 2015) [Online]. Disponible: <https://proceedings.neurips.cc/paper/2015/file/11d0e6287202fcd83f79975ec59a3a6-Paper.pdf>
- [14] Breast Cancer Wisconsin (Diagnostic) Data Set [Online]. Disponible: <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29>
- [15] 2019. Haifeng Jin, Qingquan Song, and Xia Hu. "Auto-keras: An efficient neural architecture search system." Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining [Online]. Disponible: <https://autoke-ras.com>
- [16] 2012. University of Central Florida [Online]. Disponible: <https://www.crcv.ucf.edu/data/UCF50.php>
- [17] 2021. Taha Anwar. Introduction to Video Classification an Human Activity Recognition[Online]. Disponible: <https://learnopencv.com/introduction-to-video-classification-and-human-activity-recognition/>
- [18] 2019. Adrian Rosebrock. Human Activity Recognition with OpenCV and Deep Learning [Online]. Disponible: <https://www.pyimagesearch.com/2019/11/25/human-activity-recognition-with-opencv-and-deep-learning/>
- [19] 2017. The Kinetics Human Action Video Dataset [Online]. Disponible: <https://arxiv.org/abs/1705.06950>
- [20] FitnessAlly. Disponible: <https://fitnessallyapp.com/>

AGRAÏMENTS

APÈNDIX

A1. SECCIÓ D'APÈNDIX

.....
.....
.....
.....

A2. SECCIÓ D'APÈNDIX

.....
.....
.....
.....