

# Pose Analysis

## Jose Antonio Picazos Carrillo

**Resum**—Resum del projecte, màxim 10 línies. ....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....

**Paraules clau**—Paraules clau del projecte, màxim 2 línies. ....  
 .....

**Abstract**—Versió en anglès del resum. ....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....

**Index Terms**—Versió en anglès de les paraules clau. ....  
 .....



## 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<sup>[1]</sup>, 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.

También se conocen algunos proyectos de deportistas muy conocidos como el de Xavi Hernandez, con el proyecto de Kognia<sup>[2]</sup>, o también el de Andrés Iniesta con First Vision<sup>[3]</sup>. 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 és HomeCourt<sup>[4]</sup>, 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

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

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.

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<sup>[20]</sup>, de la empresa Twenty Billion Neurons, el cual es una aplicación que realiza la función de asistente para el entrenamiento 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.

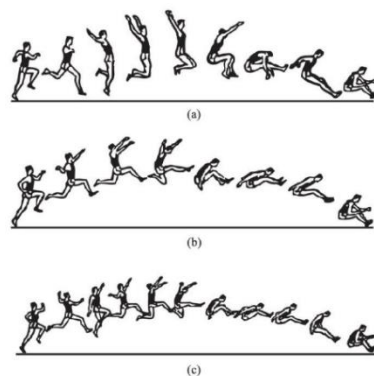


Ilustración 1: Fases salto longitud

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<sup>[5]</sup>, 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<sup>[6]</sup>, 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 hiperparámetros. 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 el trabajo que este conlleva se utilizará una metodología ágil para gestionarlo de forma eficiente, llevar un control de todas las tareas y saber en el punto en el que me vaya encontrando.

De todas las metodologías existentes aplicaré 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<sup>[7]</sup>, 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 me servirá para utilizarlo de “SandBox”, ya que toda la documentación que se va aportando durante el proyecto quedará almacenada en el equipo de Teams.

Para abordar el objetivo, sobre el funcionamiento del algoritmo de autoaprendizaje, AutoML, con el resto de algoritmos, inicialmente se escogerá un dataset público de imágenes relacionadas con el deporte. Dado que es un método novedoso analizaré sus resultados y comprobare si realmente es cierto lo que dice la teoría y proporciona un modelo bien entrenado y optimizado para su uso.

Para crear el dataset específico sobre el atletismo, dado que no existen datasets específicos de atletismo, realizaré una captura de imágenes de atletas realizando una disciplina. Seguidamente una vez creado el dataset se debe crear el modelo, para ello se utiliza el novedoso método de autoaprendizaje llamado AutoML. Este método permitirá obtener un modelo que ha sido entrenado automáticamente y el cual es el modelo mejor optimizado de todos.

Habiendo obtenido un modelo de datos específico, gracias a la herramienta Streamlit, podré realizar un primer prototipo de aplicación a ordenador en la cual se podrán ejecutar diferentes métodos para realizar el análisis de la pose del atleta de imágenes o videos, y también poder mostrar sugerencias y puntos de mejora en la posición del atleta. Más adelante, si es posible con el framework de Flutter, se creará un segundo prototipo multiplataforma el cual se pueda consultar desde nuestro dispositivo móvil.

## 4 PLANIFICACIÓN

Para realizar toda la planificación he tenido en cuenta el calendario que se proporciona sobre el proyecto y los informes que hay que entregar durante su ejecución. Teniendo en cuenta el calendario proporcionado únicamente dispongo de 22 semanas para realizar todo el proyecto. Durante el transcurso de las semanas, me encuentro con diferentes puntos de control, en los cuales he de entregar un informe sobre el trabajo realizado. Estos puntos de control son los siguientes:

Semana	Periodo	Entrega
--------	---------	---------

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

Tabla 1: Puntos de control del proyecto

Las primeras cinco semanas han sido utilizadas para organizar la idea y plantear los objetivos del proyecto y las últimas cinco semanas van a ser utilizadas para realizar la presentación del proyecto y últimos retoques del informe final. Revisando los diferentes puntos de control, me doy cuenta que aproximadamente dispongo de 14 semanas para trabajar en el corazón del proyecto.

Siguiendo la metodología Agile comentada anteriormente, realizare un proceso de análisis de los requisitos al inicio del proyecto, así, seleccionaré los requisitos prioritarios y los incluiré en la planificación de los sprints. Sabiendo el tiempo disponible para trabajar en el proyecto, este se ha dividido en diferentes fases como inicio del proyecto, análisis AutoML, creación dataset atletismo, análisis pose detection atletismo y cierre del proyecto.

A continuación, describo las diferentes tareas que incluiré en cada una de las fases del proyecto mencionadas anteriormente.

1. Inicio del proyecto
  - a. Reunión inicial con el tutor.
  - b. Definición de los objetivos.
  - c. Búsqueda de información necesaria.
  - d. Informe inicial.
2. Análisis AutoML
  - a. Preparación entorno para AutoML.
  - b. Análisis algoritmo AutoML.
  - c. Comparativa resultados obtenidos.
3. Creación dataset atletismo
  - a. Human Activity Recognition.
  - b. Recopilación de imágenes.
  - c. Creación dataset.
  - d. Entrega informe progreso 1
  - e. Analizar datos dataset.
4. Análisis pose detection atletismo
  - a. Entrenamiento modelo con AutoML.
  - b. Análisis modelo entrenado.
  - c. Entrega informe progreso 2
  - d. Prototipo inicial.
5. Cierre del proyecto
  - a. Entrega informe final.
  - b. Últimas modificaciones.
  - c. Preparación defensa del proyecto.
  - d. Presentación del proyecto.

e. Entrega y finalización del proyecto.

Tras realizar un desglose de cada una de las tareas que realizare en cada fase del proyecto e introducirlas en el programa ClickUp, teniendo en cuenta el calendario proporcionado y la carga de trabajo de cada fase. Esto me da la posibilidad de poder obtener un diagrama de Gantt que muestra la planificación de forma muy gráfica.

5

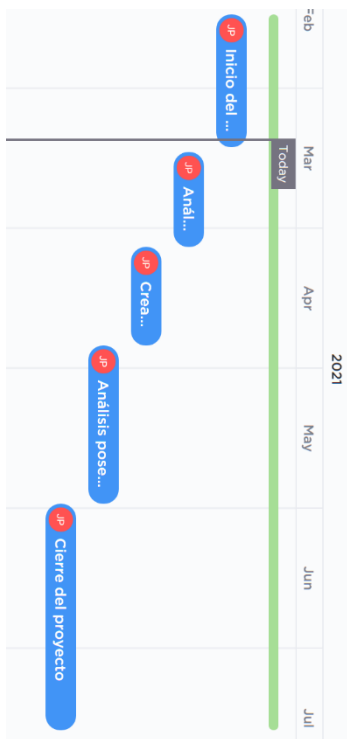


Ilustración 2: Diagrama de Gantt

## 5 DESARROLLO

En este apartado os explicaré toda la información respecto al desarrollo del proyecto durante sus fases.

### 5.1 Human Activity Recognition

Tras analizar los algoritmos de AutoML, llego a la fase de crear un data set propio para el atletismo, pero antes de crear este data set he realizado una pruebas iniciales con diferentes métodos de 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 realizar el reconocimiento se hace uso de metodos del Deep Learning como pueden ser redes convulocionales.

Para resolver el problema de Human Activity Recognition, he puesto a prueba dos métodos diferentes. Un primer método más manual en el cual realizó el entrenamiento del modelo en base un data set de actividades y un segundo método en el cual he hecho uso de una red convolucional ya entrenada.

En el primer método, hemos utilizado el data set UCF101-Action Recognition<sup>[16]</sup>, el cual es un data set para action recognition compuesto por 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.

Para poner a prueba este primer método he entrenado el modelo de clasificación utilizando las clases HighJump, PoleVault, JavelinThrow, LongJump, ThrowDiscus y Biking. La mayoría de las clases están relacionadas con el mundo del atletismo. Una vez entrenado el modelo gracias a los vídeos del data set he cogido algunos vídeos de Youtube en el que se realizan estas acciones y he comprobado su eficacia. En algunos casos si reconocía correctamente la acción, pero en otros casos no y era muy inestable al realizar el reconocimiento.



Ilustración 3: Activity Recognition con modelo manual

En el segundo método, como ya he mencionado anteriormente he utilizado una red convolucional (ResNet) entrenada con un data set llamado DeepMind Kinetics human action video data set<sup>[19]</sup>. 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 simplemente he utilizado los vídeos de Youtube utilizados en el primer método para comprobar su eficacia. En este caso sí que reconoce de una forma más estable las acciones, pero al ser una red entrenada con un mayor número de clases en algunos casos si generaba algo de confusión a la hora de reconocer las acciones.



Ilustración 4: Activity Recognition con red convolucional

### 5.1.1 OpenPose

He probado diferentes métodos para realizar el reconocimiento de la acción sin tener en cuenta la pose de la persona. Para realizar el reconocimiento de la pose, he decidido utilizar OpenPose, el cual es un detector de pose en tiempo real y me permitirá extraer los puntos (keypoints) del cuerpo, estos puntos hacen referencia a las articulaciones del cuerpo. En este detector se pueden utilizar dos modelos diferentes los cuales son COCO y BODY\_25, cada uno de ellos extrae un numero de puntos del cuerpo. En el caso del COCO extrae un total de 18 puntos y en cambio BODY\_25, un total de 25. En mi caso utilizare el modelo BODY\_25.



Ilustración 5: Ejemplo OpenPose Body\_25

Inicialmente he decidido grabar una serie de vídeos caseros en los cuales únicamente sale una persona y realiza dos acciones que son saltar y caminar. Tras obtener todos los vídeos he recortado los vídeos separando cada acción en un vídeo diferente, para así obtener dos carpetas con los vídeos de las acciones diferenciadas. Posteriormente se ha recorrido cada vídeo y he extraído cada frame generando una imagen y obteniendo un conjuntos de imágenes que posteriormente procesare para poder obtener los keypoints de cada una de ellas. En la tabla que hay más abajo se pueden ver la cantidad de vídeos obtenidos para cada clase y el total de frames que sean extraído.

Acción	Nº Vídeos	Nº Frames
Caminar	8	1034
Saltar	6	464

Habiendo extraído todos los frames, llega el momento de extraer las características de las imágenes, en este caso son los keypoints de la pose detectada en la imagen. Estos keypoints son necesarios para posteriormente poder entrenar el modelo de clasificación. Para extraer las características he procesado cada una de las imágenes con OpenPose extrayendo los keypoints de la pose detectada y posteriormente ha sido almacenado en un archivo .csv. Este archivo esta formado por todos los puntos detectados y su correspondiente acción.

El siguiente paso ha sido realizar el entrenamiento, he realizado un primer entrenamiento haciendo uso de RandomForest y posteriormente he utilizado AutoKeras para entrenar un segundo modelo. Los dos entrenamientos han sido realizados con el mismo porcentaje de entrenamiento y de test, en este caso un 30% y 70% respectivamente.

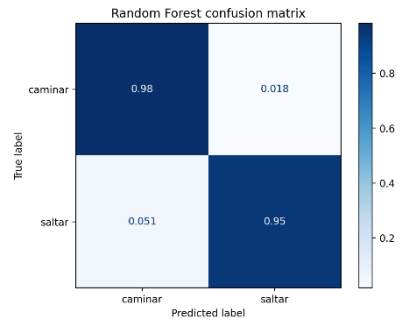


Ilustración 6: Matriz Confusión Random Forest

Ilustración 7: Matriz Confusión AutoKeras

Analizando las matrices de confusión que he obtenido tras realizar el entrenamiento se puede observar que en 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 esta produciendo overfitting.

Tras realizar la prueba con un data set propio, he decidido realizar nuevamente el mismo proceso pero en este caso con un data set publico, se trata del UFC101, el cual tiene disponible un total de 101 acciones deportivas. Inicialmente me he centrado en la acción salto de longitud (“Long-Jump”), la cual esta relacionada con el mundo del atletismo.

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 generaba una problemática ya que OpenPose extrae la información de todas las poses detectadas en la imagen y es difícil saber que listado de puntos corresponde a la persona que esta realizando la acción.

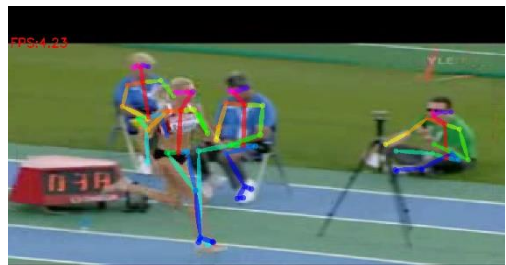


Ilustración 8: OpenPose sin tracking

Para solucionar este problema he decidido aplicar un algoritmo de tracking, para poder hacer un seguimiento de la persona que realiza la acción. El algoritmo de tracking escogido ha sido el DeepSort. El DeepSort me permite hacer una detección de la persona que realiza la acción y hacer un seguimiento de esta durante todo el transcurso del vídeo. 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.



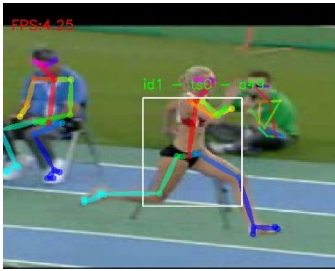


Ilustración 9: OpenPose con DeepSort

Tras realizar la implementación completa de OpenPose con DeepSort, ha llegado el momento de poner a prueba el algoritmo con diferentes deportes. Para ello he escogido 10 deportes/disciplinas del data set UFC101 que han sido: Long-Jump, Biking, Archery, Shotput, PlayingGuitar, GolfSwing, Bowling, Basketball y HighJump. De cada uno de estos deportes escogeré entre 20-25 vídeos para poder realizar la extracción de las características y así poder entrenar nuestro modelo. El modelo ha sido entrenado con 3 clasificadores diferentes para poder realizar una comparativa de ellos, se ha utilizado Random Forest, LSTM y AutoKeras. Se ha añadido la red neuronal LSTM, ya que es un tipo de red neuronal que “recuerda” estados previos, es decir tiene en cuenta el tiempo, y esta información es utilizada para realizar la predicción.

## 5.2 Análisis AutoML

Siguiendo la planificación realiza al inicio del proyecto, he comenzado con el primer sprint en el que se incluye el análisis AutoML. Durante este primer sprint he realizado una búsqueda relacionada con la tecnología AutoML, para conocer sus características, los diferentes tipos y librerías que existen y como hacerlas funcionar.

Tras analizar las librerías he decidido comenzar con el módulo Auto-Sklearn<sup>[13]</sup> que corresponde la famosa librería de Machine learning de Scikit-learn. Para poner a prueba este módulo he escogido un dataset genérico y que todo el mundo conoce como es el Breast Cancer Wisconsin (Diagnostic)<sup>[14]</sup>. Este dataset es utilizado para tareas de clasificación, dispone de un total de 569 filas y 32 atributos, entre los cuales tenemos del diagnóstico que es nuestro target que puede tener los valores M (Malignant) o B (Benign).

Para comprobar el funcionamiento del Auto-Sklearn he realizado el entrenamiento de un modelo en el cual se han obtenido los mejores valores de hiperparametros y el mejor clasificador para el conjunto de datos. Tras tener entrenado el modelo he realizado un proceso de validación y se ha obtenido un accuracy de un 95,9%. Analizando más a fondo los valores del modelo generado he visto que el mejor clasificador ha sido el MLP(Multi Layer Perceptron), es una red neuronal formada por múltiples capas y con la capacidad de resolver problemas que no son linealmente separables.

Para comprobar el rendimiento del Auto-Sklearn, se ha realizado el mismo entrenamiento con diferentes clasificadores como son: Decision Tree, Random Forest, KNN, Multi Layer Perceptron(MLP) y Gaussian Process. Todos estos

clasificadores han sido utilizados con los valores por defecto para comprobar su rendimiento y posteriormente de forma manual se han ajustado los hiperparametros de cada uno de ellos hasta obtener los mejores resultados.

A continuación, se pueden ver las diferentes matrices de confusión que he obtenido para cada uno de los clasificadores mencionados anteriormente.

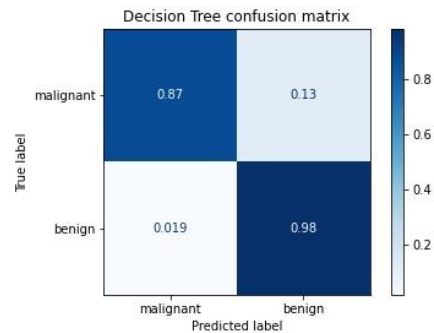


Ilustración 10: Matriz confusión Decision Tree

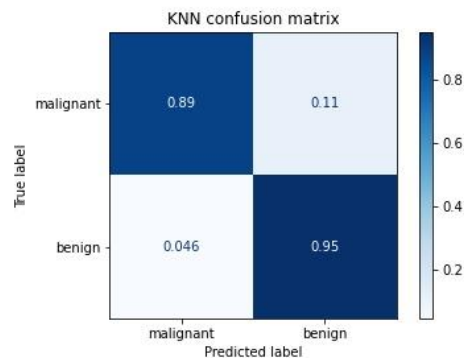


Ilustración 11: Matriz confusión KNN

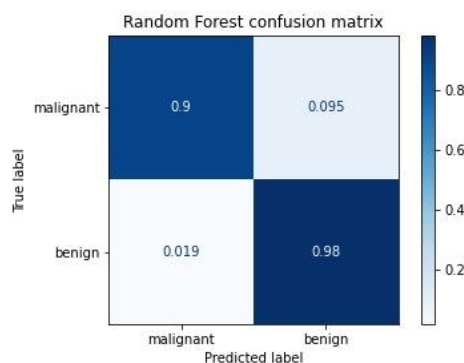


Ilustración 12: Matriz confusión Random Forest

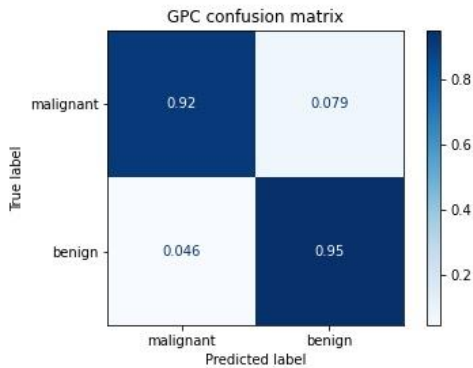


Ilustración 13: Matriz confusión Gaussian Process

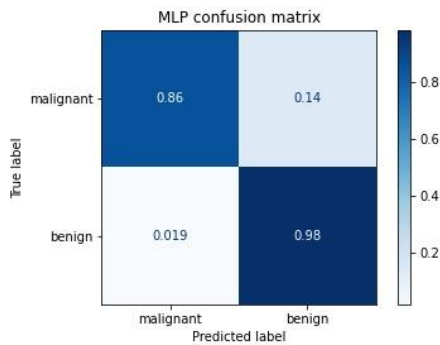


Ilustración 14: Matriz confusión MLP

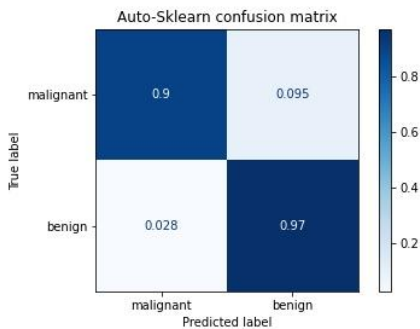


Ilustración 15: Matriz confusión Auto Sklearn

Como se puede ver en la gráfica que tenemos más abajo, he comparado el accuracy obtenido con cada uno de los clasificadores para comparar su rendimiento. Analizando los resultados obtenidos, se puede observar que todos los clasificadores han aumentado su accuracy aproximadamente entre un 1-2% una vez han sido optimizados sus hyperparametros. Comparando Auto-Sklearn respecto a los clasificadores, se puede observar que no hay una gran diferencia respecto a rendimiento incluso se puede ver que en algún caso como es el Random Forest se ha obtenido un accuracy mayor. Posiblemente añadiendo algunos parámetros al Auto-Sklearn se

podría obtener un rendimiento similar. Si comparo directamente el MLP optimizado manualmente con el Auto-Sklearn, se puede ver que el rendimiento obtenido con Auto-Sklearn es mayor, posiblemente esto se deba a un mejor trato de los datos y una mejor optimización de los hyperparametros.

Tras analizar Auto-Sklearn con algunos de los diferentes algoritmos de clasificación, he decidido compararlo con otro algoritmo de AutoML como es AutoKeras<sup>[15]</sup>. AutoKeras 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 nos ayuda a poder realizar el entrenamiento de nuestro modelo. Para poner a prueba AutoKeras se han utilizado los mismos datos que en los casos anteriores.

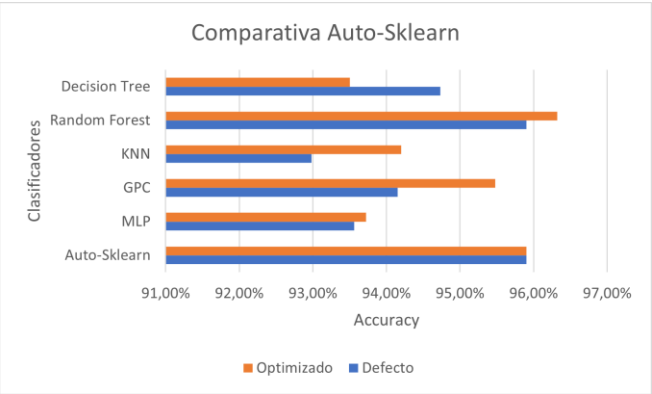


Ilustración 16: Comparativa Auto-Sklearn

Observando los resultados obtenidos por AutoKeras se puede ver en la gráfica que hay más abajo, se ha obtenido un mejor rendimiento respecto a Auto-Sklearn, aproximadamente un 2% más de accuracy. Posiblemente esta mejora de rendimiento se deba al uso del Deep Learning, como son las redes neuronales, para obtener una mejor optimización.

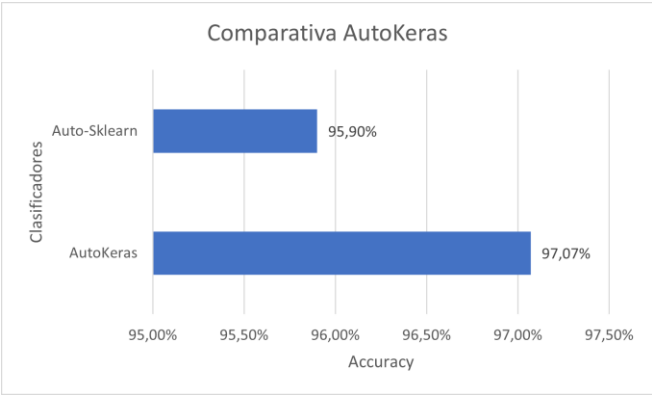


Ilustración 17: Comparativa AutoKeras

Tras ver los resultados obtenidos mi elección para realizar el entrenamiento del dataset propio, es AutoKeras. Pienso que gracias al uso del deeplearning para el caso en el que tenemos datos más complejos no servirá de gran ayuda para obtener el

mejor modelo. Posiblemente uno de los inconvenientes que presenta este algoritmo es la gran cantidad de recursos que utiliza cuando está realizando el entrenamiento del modelo.

### 5.3 Entorno prototipo

Tras realizar diferentes pruebas y tener una gran cantidad de informacion y funciones de todas ellas, he decidido crear un prototipo de entorno de ejecución mediante la libreria StreamLit. Este proceso me ha permitido tener un entorno mucho más organizado y poder ejecutar todos los procesos de una forma más rápida.

## 6 CONCLUSIÓN

.....

.....

.....

.....

## AGRAÏMENTS

.....

.....

.....

.....

## 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.bnkto-thefuture.com/pitches/2079/\\_first-vision-the-sports-broadcasting-revolution-with-andres-iniesta-and-serge-ibaka](https://app.bnkto-thefuture.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-deepestream-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/>



**APÈNDIX**

**A1. SECCIÓ D’APÈNDIX**

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

**A2. SECCIÓ D’APÈNDIX**

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