



Escuela Técnica Superior de Ingeniería Informática

Ingeniería Informática. Ingeniería del Software.

## TRABAJO FIN DE GRADO

Reconocimiento de imágenes aplicado en pruebas deportivas

Autor/es:  
Antonio Manuel Solís Miranda

Tutor/es:  
Jose Antonio Troyano Jiménez

Segunda convocatoria

Curso 2023/2024

# Índice

Índice de ilustraciones .....	3
Introducción .....	5
Objetivos .....	6
Motivación .....	7
Resumen.....	8
Planificación .....	11
1. Historia del cronometraje .....	12
2. Tecnologías existentes .....	15
2.1. Tecnología RFID.....	15
2.1.1. ¿Qué es la tecnología RFID?.....	15
2.1.2. Empresas que utilizan RFID .....	16
2.1.3. Ventajas y desventajas de RFID .....	16
2.1.4. Uso habitual en pruebas deportivas .....	17
2.2. Tecnología de reconocimiento facial .....	18
2.2.1. ¿Qué es el reconocimiento facial?.....	18
2.2.2. Funcionamiento .....	19
2.2.3. Empresas y reconocimiento facial en el deporte .....	20
3. Marco teórico de la implementación.....	21
3.1. Adquisición de imágenes .....	22
3.2. Técnicas de preprocesamiento .....	24
3.2.1. Transformaciones afines .....	25
3.2.2. Filtros.....	26
3.2.3. Transformación del histograma .....	30
3.3. Técnicas de detección de rostros.....	34
3.3.1. Algoritmo Viola-Jones .....	35
3.3.2. Histograma de gradientes orientados (HOG) .....	37
3.3.3. Redes neuronales convoluciones (CNN) .....	38
3.4. Extracción de características.....	43
3.5. Reconocimiento .....	44
3.6. Algoritmos complementarios .....	47
4. Marco Practico .....	49
4.1. Tecnologías utilizadas .....	49
4.2. Entorno de desarrollo y versiones .....	51

4.3.	Equipos utilizados .....	52
4.4.	Configuración de los equipos.....	53
4.5.	Desarrollo practico.....	54
4.5.1.	Toma de imágenes de los rostros de cada atleta .....	54
4.5.2.	Almacenamiento de imágenes .....	56
4.5.3.	Reducción de cantidad de imágenes .....	57
4.5.4.	Detección del rostro.....	58
4.5.5.	Aumento de datos (Data Augmentation) .....	65
4.5.6.	Reconocimiento de rostros .....	66
4.6.	Resultados obtenidos.....	71
5.	Inclusión real del sistema de cronometraje.....	74
5.1.	Organización de un evento real .....	75
5.2.	Organización de un evento real que use reconocimiento facial .....	76
6.	Problemas y limitaciones del sistema .....	78
7.	Análisis a futuro y mejoras a aplicar .....	79
	Conclusiones .....	81
	Bibliografía: .....	83

## Índice de ilustraciones

Ilustración 1. Identidad; Reflexión; Escalado; Rotación; Cizallamiento. ....	26
Ilustración 2. Etapas en la aplicación de un filtro frecuencial .....	27
Ilustración 3. Imagen original; Imagen con filtro paso baja; Imagen con filtro paso alta .....	27
Ilustración 4. Comparativa de los filtros Sobel, Prewitt y Robert .....	29
Ilustración 5. Imagen en blanco y negro y su histograma. ....	30
Ilustración 6. Tipos de funciones de transformación simples .....	31
Ilustración 7. Imagen y su histograma original junto con la imagen con desplazamiento k=70 y su histograma.....	31
Ilustración 8. Función de transformación para desplazamiento k=70. ....	32
Ilustración 9. Imagen, histograma y función resultante tras aplicar Stretching. ....	32
Ilustración 10. Imagen, histograma y función resultante tras aplicar Shrinking.....	33
Ilustración 11. Imagen ecualizada, CDF e histograma de la imagen ecualizada. ....	34
Ilustración 12. Ecuación del cálculo de descriptores Haar y estos con respecto a un rostro. ....	35
Ilustración 13. Imagen original con descriptor Haar señalado, junto a su imagen integral y la operación a realizar.....	35
Ilustración 14. Proceso de entrenamiento con clasificadores debiles, junto con el sistema de filtros en cascada.....	36
Ilustración 15. Imagen y su representación con HOG, donde se pueden ver los gradientes.....	37
Ilustración 16. Estructura de una neurona artificial y tipos de funciones de activación. .....	39
Ilustración 17. Max-pooling de 2x2. ....	40
Ilustración 18. Estructura de una CNN. ....	41
Ilustración 19. Funcionamiento simplificado de KNN. ....	45
Ilustración 20. Representación en un plano 2D del funcionamiento de una SVM. ....	45
Ilustración 21. Representación de los kernels de forma bidimensional. ....	47
Ilustración 22. PCA aplicado a una imagen con distintos niveles de varianza. ....	48
Ilustración 23. Funcionamiento simplificado de K-Medias. ....	49
Ilustración 49. Principales empresas promotoras del reconocimiento facial. ....	20
Ilustración 50. Mediacoach de La Liga.....	21
Ilustración 24. Logo de OpenCV. ....	50
Ilustración 25. Logo TensorFlow.....	51
Ilustración 26. DLib. ....	51
Ilustración 27. Logo del IDE PyCharm.....	52
Ilustración 28. Cada una de las fases de la toma de imágenes del rostro del participante. ....	55

Ilustración 29. Configuración de doble cámara simultánea.....	55
Ilustración 30. Proceso de captación de imágenes. ....	56
Ilustración 31. Estructura de almacenamiento. ....	57
Ilustración 32. Método de reducción de número de imágenes (PCA y K-Means). ....	58
Ilustración 33. Clasificador y detector de rostros frontales. ....	59
Ilustración 34. Imagen de ejemplo con las coordenadas devueltas por el clasificador. ....	60
Ilustración 35. Imágenes extraídas de los videos de prueba. ....	62
Ilustración 36. Resultados de aplicar Viola-Jones (izquierda) y HOG+SVM (derecha)..	64
Ilustración 37. Generador de imágenes (Data Augmentation). ....	66
Ilustración 38. Resultado de aplicar dos veces Data Augmentation para una misma imagen. ....	66
Ilustración 39. Conjunto de muestra de LFW. ....	67
Ilustración 40. Rostro promedio de todas las imágenes del dataset. ....	68
Ilustración 41. Creación y entrenamiento de PCA+SVM. ....	68
Ilustración 42. Diagrama de una capa residual a la derecha junto con la estructura básica de un ResNet. ....	69
Ilustración 43. Configuración del modelo ResNet-50.....	70
Ilustración 44. Entrenamiento de ResNet-50. ....	70
Ilustración 45. Estadísticas de entrenamiento y evaluación. ....	72
Ilustración 46. Reconocimiento de rostro mediante PCA+SVM.....	73
Ilustración 47. Etapas/épocas de entrenamiento del modelo. ....	73
Ilustración 48. Resultados de la prueba con el modelo ResNet.....	74
Ilustración 51. Entrega de dorsales de la Maratón de Sevilla. ....	75
Ilustración 52. Salida de la Maratón de Sevilla.....	76
Ilustración 53. Representación de la toma de imágenes del rostro del atleta. ....	77
Ilustración 54. Representación de la disposición de las cámaras en el arco de salida/meta.....	77
Ilustración 55. Representación de la implementación ideal. ....	78

# Introducción

Desde hace siglos los eventos deportivos han sido los acontecimientos más concurridos del mundo reuniendo miles de personas, uniéndolas en un mismo instante y en un mismo lugar, gracias a una misma pasión, el deporte. Desde los Juegos Olímpicos hasta eventos amateurs, el deporte ha generado en aquellos que participaban, una gran cantidad de emociones, experiencias e historias increíbles dignas de recordar. Pero no solo ha generado esto, sino que también ha generado grandes cantidades de información que pueden ser procesadas y aprovechadas para ejercer un impacto positivo en las vidas de los deportistas.

En la era del Big Data, el dato es el bien más demandado y cotizado por las empresas e incluso por los particulares, aunque en primera instancia no lo sepan. Es en los datos donde se define el marco contextual de este Trabajo de Fin de Grado, más concretamente en los datos de carácter visuales, las imágenes.

El enfoque principal de este TFG consiste en explorar de manera efectiva la implementación de técnicas de reconocimiento de imágenes en el ámbito de las pruebas deportivas, siendo las pruebas de atletismo donde se aplicará, aunque también se abordará superficialmente el uso de estas técnicas en otras pruebas y ámbitos deportivos. Desde la precisa identificación de atletas hasta el análisis detallado de sus rostros, el reconocimiento de imágenes es una tecnología que ha llegado para quedarse y trae una perspectiva diferente, tanto para pruebas como para rendimiento deportivo.

Conforme se avance en el documento, se abordarán tanto los avances tecnológicos como los desafíos que surgen de la integración de esta tecnología en el ámbito de las pruebas deportivas. Se examinarán casos de estudio, se evaluarán aplicaciones prácticas y se presentará un modelo realista en base a las conclusiones obtenidas.

En última instancia, este trabajo pretende aportar un pequeño grano de arena al creciente campo de la tecnología de reconocimiento de imágenes al deporte, destacando el potencial que esta tiene para revolucionar la organización de eventos, la experiencia de usuario e incluso el propio rendimiento del atleta.



## Objetivos

Los objetivos principales que se quieren alcanzar para este trabajo de fin de grado son los siguientes:

- Desarrollar de forma básica un sistema de reconocimiento de imágenes: Diseñar y desarrollar un sistema de reconocimiento de imágenes capaz de identificar de manera razonablemente precisa los rostros de los atletas (además de otras propiedades que puedan ayudar a la eficiencia del modelo) durante una prueba deportiva.
- Optimizar el modelo de Reconocimiento Facial: Investigar y aplicar técnicas de optimización para mejorar la precisión y el rendimiento del modelo implementado.
- Análisis de los resultados: Realizar pruebas, en un entorno de prueba deportiva real o simulada, para la posterior validación de los resultados para evaluar su eficacia y eficiencia en una situación real, analizando cada parámetro, llegando a unas conclusiones realistas.
- Explorar aplicaciones prácticas y potenciales: Investigar y discutir posibles aplicaciones de este sistema como, por ejemplo, para la mejora de la gestión de eventos, el seguimiento de tiempos y posiciones, y la experiencia general de los participantes. Al igual que analizar el uso en otros eventos deportivos o incluso fuera del ámbito del deporte.
- Comparar resultados, analizar el mercado y búsqueda de competencia: Realizar comparativas con otras tecnologías de reconocimiento similares ya existentes, para plasmar las ventajas y desventajas del sistema en comparación con las necesidades del mercado y la competencia.
- Recurso de conocimiento: Contribuir a la comunidad con los conocimientos que se plasmen en este documento relacionados con el reconocimiento de imágenes especialmente en eventos deportivos.

En resumen, los objetivos que se plantean para este TFG son el desarrollo de un sistema básico de reconocimiento de imágenes y optimizarlo para su análisis posterior, explorar las posibles aplicaciones reales y potenciales, analizar un mercado en auge y buscar la competencia relacionada, para posteriormente discutir cómo sería una implementación real.

## Motivación

Se trata de una idea que, aunque no es nueva y ya se está utilizando (como veremos más adelante), está poco explotada y son pocas las empresas a nivel español que utilizan esta tecnología. Además, puede ser un tema introductorio al reconocimiento de imágenes con una curva de aprendizaje y con una cantidad de ejemplos prácticos que resulte fácil de asimilar por estudiantes y personas que quieran introducirse en el mundo de la IA y el reconocimiento facial y corporal.

Además, este tema tiene una motivación personal detrás, ya que fue mi entorno quien me alentó a realizar este trabajo de fin de grado con esta temática, pues tanto mis familiares como mi persona colaboramos en pruebas deportivas, mayoritariamente de atletismo. Por lo tanto, este trabajo lo considero como un “ikigai”, una combinación de pasión, profesión y necesidad.

- Pasión por las nuevas tecnologías y la vanguardia, como es el reconocimiento facial y la inteligencia artificial. Tecnologías que ya existían pero que han tomado gran relevancia en estos últimos años, popularizada por las enormes capacidades de procesamiento de lenguaje y automatización. Siempre criticadas, debido a la privacidad y control por parte de las empresas y gobiernos, pero usada por todos en el día a día, ya sea para desbloquear móviles, agilizar compras o vigilar tu propia casa.
- Profesión en el mundo de los eventos deportivos, uniendo el ámbito que estudio con este, pues parece que se ha quedado un tanto estancado y no es capaz de subirse al tren de los avances científicos y tecnológicos.
- Necesidad de abaratamiento de costes a largo plazo, problemas de abastecimiento de chips y facilidades para pequeños eventos y pruebas privadas.

Estos tres puntos son los que hacen que para mí tenga sentido realizar este trabajo de fin de grado, y en los que iré profundizando a lo largo de este trabajo, del que espero que salgan con una idea clara de cómo se podría aplicar esta tecnología a este campo y como podría afectar al futuro del deporte.



## Resumen

El trabajo presentado se centra en la implementación y evaluación de un sistema de reconocimiento facial aplicado al cronometraje de pruebas deportivas, sirviendo como estudio teórico-práctico para la aplicación real de este sistema. Se posiciona como una alternativa a la tecnología actual de cronometraje, el RFID, ofreciendo una solución innovadora basada en técnicas de visión por computadora y aprendizaje profundo, y un aire de renovación a un sector que utiliza la misma tecnología que se usaba hace 25 años. Para lograrlo se ha presentado un sistema completo tanto tecnológicamente como físicamente, detallando todo lo necesario para que se pueda llevar a cabo esta idea.

La base principal de este sistema, como es lógico, es la creación de un sistema completo de reconocimiento facial óptimo, para ello se han explicado todos los conceptos necesarios de las técnicas utilizadas, de forma que quede completamente claro todos las técnicas, parámetros y algoritmos que han sido necesarios en la implementación y que serán necesarios para una futura implementación real. Para lograrlo se ha dividido en 5 partes fundamentales: Adquisición de Imágenes, Preprocesamiento de Imágenes, Detección de Rostros, Extracción de Características y Reconocimiento Facial. A continuación, se explicarán las implementaciones realizadas y de la que se ha concluido que es la mejor opción en base a pruebas reales y teóricas:

### **Adquisición de Imágenes**

En esta fase, se emplearon las librerías OpenCV y Dlib para capturar y almacenar imágenes de rostros localmente, para asegurar un acceso y guardado rápido de los datos se debe almacenar en un sistema de archivos local. Para optimizar el almacenamiento, se aplicaron técnicas de reducción de datos como K-Means y Análisis de Componentes Principales (PCA). Durante las pruebas comparativas entre los métodos Viola-Jones y Histogram of Oriented Gradients (HOG) para la detección de rostros, se concluyó que HOG ofrece un mejor desempeño en la identificación de múltiples rostros y en diferentes ángulos dentro de una misma imagen.

### **Preprocesamiento de Imágenes**

El preprocesamiento de las imágenes capturadas se realizó utilizando técnicas de aumento de datos (Data Augmentation) proporcionadas por TensorFlow. Estas técnicas incluyen rotación, escalado, y transformación de las imágenes, con el objetivo de mejorar la robustez del modelo de reconocimiento ante variaciones en las condiciones de captura.

### **Detección de Rostros**

La detección de rostros se integró con la fase de adquisición, empleando el método HOG de Dlib. Este método se seleccionó por su eficacia demostrada en la fase de pruebas comparativas. Aclarar que las imágenes almacenadas, corresponden a los rostros

reconocidos por HOG Dlib, lo que permite reducir el almacenamiento, pues no se requiere el almacenaje más allá del rostro del atleta.

### **Extracción de Características**

Para la extracción de características, se utilizó una Red Neuronal Convolucional (CNN) específica: ResNet, implementada con TensorFlow. Esta fase se combinó con la fase de reconocimiento pues se usó una capa de extracción de características, ahorrando la implementación de otros algoritmos. El modelo fue importado con los pesos de un entrenamiento previo con ImageNet, entrenado con el dataset de LFW y se le hizo fine-tuning con los datos propios de dos atletas.

### **Reconocimiento Facial**

Se evaluaron dos enfoques para el reconocimiento facial: PCA combinado con una Máquina de Vectores de Soporte (SVM) y ResNet. Los resultados mostraron que ResNet supera ampliamente a PCA+SVM en términos de precisión, debido a que las SVM tienen problemas con los soportes en escenarios con variabilidad alta en los datos, mientras que ResNet es capaz de generalizar muy por encima de SVM.

Los experimentos y pruebas demostraron que un sistema que implemente HOG para identificación de rostros y ResNet para la identificación de estos, ofrece una mayor precisión en el reconocimiento facial comparado con Viola-Jones, PCA+SVM y sus posibles combinaciones. La facilidad de HOG para reconocer grandes cantidades de rostros y la capacidad de ResNet para manejar variaciones en los datos de entrada se tradujo en un rendimiento superior en pruebas prácticas.

### **Propuesta de Sistema Real**

Se ha diseñado una arquitectura para la implementación del sistema en un entorno real. Consta de las siguientes partes:

Comenzando con la preparación del material, ya sea cámaras, focos y equipos, como software, datos de atletas que hayan participado en pruebas anteriores con este sistema que pueda servir como información para entrenamiento de la ResNet, además del ajuste de esta para su posterior entrenamiento.

Una vez comenzado la exposición del corredor, será necesario haber instalado las cámaras que recopilaran las imágenes de los rostros de los atletas. Al igual que los equipos que almacenarán la información, que será un potente equipo con mucho almacenamiento capaz de almacenar toda la información. Tras la finalización de cada día de exposición deberá de ser entrenada la ResNet con los datos capturados, optimizando el tiempo y evitando tener una gran cantidad de datos la noche antes del día grande.

En cuanto al día del evento, se deberá instalar en el arco de salida y en puntos estratégicos del recorrido, focos para asegurar la correcta iluminación y cámaras para

acometer el reconocimiento de los rostros, es imperativo el posicionamiento de estas últimas con una inclinación de 30 grados con la horizontal que asegure un reconocimiento óptimo. Además, en salida/meta deberá de estar el equipo principal responsable de la detección y reconocimiento de rostros en tiempo real, que también recibirá los datos de los distintos puntos de control colocados a lo largo de todo el recorrido. Se utilizará una base de datos NoSQL local para almacenar las horas de inicio y fin de los atletas, registrando la última detección del rostro en el punto de salida y llegada, respectivamente.

El trabajo ofrece una visión detallada del desarrollo y evaluación de un sistema de cronometraje basado en reconocimiento facial, destacando la eficacia de técnicas modernas de aprendizaje profundo y proponiendo una implementación práctica para eventos deportivos. Este sistema no pretende mejorar la precisión y eficiencia del cronometraje que con las etiquetas RFID es muy alta, sino proporcionar una solución escalable, económica a largo plazo, innovadora y extensible a distintos servicios.

## Planificación

La planificación del tiempo es fundamental en la realización de cualquier proyecto, ya que permite organizar las tareas de manera eficiente y asegurar que se cumplan los objetivos dentro del plazo establecido. Para este proyecto, hemos dividido el trabajo en ocho tareas principales, asignando a cada una un tiempo específico según su complejidad y extensión. A continuación, se detalla esta planificación:

Nombre	Descripción	Horas estimadas	Horas reales
Estudio de competidores	Realización de un estudio con las empresas que utilizan las tecnologías descritas.	10	4
Análisis del problema y soluciones	Búsqueda de información, solución e ideación del sistema propuesto.	16	10
Análisis de herramientas	Estudio de las mejores herramientas para el correcto desarrollo del proyecto.	6	20
Desarrollo	Engloba el desarrollo de los modelos, la instalación de las herramientas y la implementación de técnicas complementarias.	200	250
Obtención de recursos	Tiempo utilizado para obtención de los recursos para realizar las pruebas.	2	1
Pruebas	Pruebas realizadas para comprobar el funcionamiento y extraer datos sobre el desempeño.	10	7
Redacción de la memoria	Engloba todos los puntos aparte de la introducción.	100	160
Preparación para la entrega	Referente al repaso y refinamiento de la entrega final.	12	20
		356	472

## 1. Historia del cronometraje

“Para ver el futuro hay que mirar al pasado”. Esta frase popular es el hilo conductor de este punto, nos adentraremos en la historia del cronometraje deportivo desde las primeras pruebas deportivas de la historia hasta sistemas actuales muy avanzados de seguimiento de estadísticas deportistas.

Tenemos que remontarnos a la Antigua Grecia (1200 a.C - 146 a.C) para observar las primeras actividades deportivas de las que tenemos evidencia real, gracias a historiadores como Heródoto, Pausanias y Tucídides. De ellos podemos extraer que los primeros eventos que reunían actividades deportivas de distintas índoles fueron los Juegos Olímpicos realizados en Olimpia en el año 776 a.C, donde se reunían deportistas de toda Grecia para realizar pruebas atléticas e hípcas donde destacaban las carreras de caballos y las pruebas de velocidad, siendo el estadio (200 metros) y el doble estadio, las principales.

A principios del siglo VI a.C se realizan los Juegos Panhelénicos, un conjunto de eventos deportivos (Juegos Olímpicos, Píticos, Nemeos e Ístmicos) realizados en diferentes ciudades del sur de la antigua Grecia, donde los atletas podían disfrutar de una continuidad en las competiciones atléticas e hípcas. Es en este marco histórico donde situamos las primeras necesidades de un cronometraje real, pues se daban casos en los que los atletas llegaban con diferencias realmente pequeñas, dando lugar a empates. Aunque la capacidad tecnológica de la época limitó en ese sentido el correcto desarrollo de los juegos y de las aptitudes de los deportistas, pues la disponibilidad de un sistema de cronometraje que permitiese realizar entrenamientos de más calidad hubiese sido clave para una evolución en el rendimiento físico traducido a tiempos menores en las competiciones. Hay que aclarar que para la época existían sistemas de conteo de tiempo, por ejemplo, los relojes de agua, sol o arena, pero no tenían un nivel de precisión suficiente como para poder usarse para estas funciones.

Este problema tecnológico es recurrente a lo largo de la historia, no solo en la sociedad griega, sino que también en las famosas carreras de carros en la antigua Roma, las carreras de natación en el río Nilo o la increíble carrera entre Menfis y el oasis de El Fayum (de unos 100 km a lo largo del desierto) en el antiguo Egipto.

No es hasta el siglo XVII cuando empieza una carrera en búsqueda de la mejora en la relojería medieval, que al igual que los métodos de conteo del tiempo, carecían de la precisión necesario para ser útiles en el ámbito del cronometraje. Con la invención del péndulo en 1656 por parte de Christiaan Huygens, y su aplicación a los relojes de pared,

se dio un paso considerable hacia la precisión, pues se pasó un error de 15 minutos diarios a unos 15 segundos. Esta mejora, motivo a otros relojeros y científicos a crear sistemas cada vez más precisos de medición, como fue el sistema de escape de áncora, inventado por William Clement en 1680 y que se popularizó por su famoso diseño del reloj de pie (también llamado reloj del abuelo), sistema que basaba su funcionamiento en el movimiento del péndulo conectado a un dispositivo con forma de ancla que hace mover el engranaje necesario para el movimiento del reloj. También se inventó años antes de manos de Richard Towneley, el escape sin retroceso, sistema que era más preciso y que no se popularizó hasta el 1715 de la mano de George Graham, destronando así al sistema de escape de ancla. Es en este momento cuando se añadió el minuterio, cercanos así a la precisión necesaria.

La siguiente gran revolución vendría de la mano de John Harrison en 1730 con la invención del cronometro (reloj de gran precisión) motivado por el premio económico de la Junta de Longitud, que incentivaba a los relojeros, físicos y tecnólogos, a realizar un invento que solucionase el problema de inexactitud en la determinación de la longitud en alta mar, clave para la supervivencia de la embarcación, es decir, un reloj que conservase la hora con precisión durante largas travesías marítimas. Durante 31 años, y tras distintos prototipos y versiones, John Harrison consiguió el premio con su modelo H4, que se retrasó tan solo 5 segundos en una travesía de 80 días. Sentando las bases de la relojería de precisión.

A partir de este periodo nos “separamos” del término reloj para dar con el cronógrafo. Un cronógrafo, a diferencia de un cronometro, se trata de un instrumento de medición que registra el transcurso de tiempo entre el arranque y la parada del mismo. Siendo la invención de esta herramienta de la mano de Louis Moinet en 1816, teniendo la esfera central una aguja que daba una vuelta en un segundo; una segunda esfera que contaba los segundos transcurridos hasta 60; una tercera que contaba los minutos transcurridos hasta 60; y una última esfera que contaba hasta las 24 horas. Mas tarde Abraham-Louis Breguet mejoró la precisión del cronógrafo de la época, que se fueron mejorando con respecto los años hasta la actualidad.

Es en este marco tecnológico del siglo XIX cuando se realiza la que muchos consideran la primera prueba deportiva cronometrada, la Carrera de Remo Doggett de 1829. Este evento se llevó a cabo en el río Támesis, Inglaterra. Esta competencia involucraba a remeros que navegaban desde Londres hasta Chelsea. Se utilizó un cronometro para ver cuanto tardaban en realizar la distancia entre el Puente de Londres a Chelsea. En atletismo fueron las pruebas atléticas de la universidad de Oxford hacia 1850, las que comenzaron a utilizar estos sistemas para el cronometraje de las pruebas.

A finales del siglo XIX tiene lugar el comienzo del evento deportivo más importante de su época y de la actualidad. Estamos hablando ni más ni menos que de los juegos olímpicos modernos. En 1894, el barón Pierre de Coubertin comenzó con el proyecto, fruto de la idea de reunir en una misma prueba a deportistas de distintas nacionalidades similar a los juegos olímpicos realizados por el empresario griego Evangelos Zappas, que buscaban replicar los antiguos juegos olímpicos pero que carecían de la

internacionalidad del proyecto que presentaba Coubertin. Es, por tanto, en 1896 cuando da lugar los primeros juegos olímpicos internacionales, que contaron con la participación de 14 países. Este mismo año se utilizaron cronógrafos y relojes para medir los tiempos oficiales, aunque contaban solo con una precisión de 1/5 de segundo lo que propiciaba errores de tiempo en las clasificaciones oficiales. No fue hasta 1916 que la empresa TAG Heuer desarrollo un sistema de cronometro con una precisión de 1/50, lo que le hizo el cronometrador oficial de los juegos olímpicos hasta 1928, cuando la empresa Longines fabricó un cronógrafo de bolsillo de precisión 1/100.

Pese a los avances tan significativos en cuanto a precisión, las carreras de velocidad como los 100 o 200 metros seguían siendo imprecisas para resolver el empate entre dos atletas, pues dos atletas podían llegar a meta con escasos milímetros o centímetros que, a ojos de los jueces, era imperceptible y hacía que la decisión que estos tomaran fueran erróneas o en casos más graves imparciales. Es para resolver este problema, cuando entra en la ecuación la fotografía, con la invención en el año 1912 de la popular foto-finish, en los juegos olímpicos de Estocolmo (al mismo tiempo que el cronometro eléctrico).

La televisación de los juegos olímpicos creó una fiebre deportiva en todo el mundo, motivando al público general a empezar a entrenar, reunirse y en última instancia crear eventos donde poder medirse y competir unos con otros, nace así las carreras populares. Los formatos de estas carreras en sus orígenes eran muy primitivos, al no tener poder económico ni tecnológico que poseían grandes marcas como los Juegos Olímpicos, la única forma en la que podían cronometrar los organizadores era apuntando los tiempos de los primeros a mano. Pero esta forma de cronometrar tenía carencias claramente notables en pruebas con muchos participantes o en distancias en las que los atletas entrasen en meta muy cerca los unos de los otros.

Con la proliferación de los códigos de barra vino los dorsales con estos mismos. El sistema constaba de una persona pulsando un botón que registraba los tiempos en un computador para después más tarde en meta, formando una cola en orden de llegada, se le asociase ese tiempo registrado, al dorsal mediante la lectura del código de barras.

Este sistema duró poco pues en 1993, una empresa llamada ChampionChip creo un sistema de cronometraje basado en tecnología RFID. Su funcionamiento se basa en el uso de radio frecuencias para la identificación inequívoca de un objeto, es decir, la etiqueta o chip transmite una onda de radio con la información de esa etiqueta que es única. La aplicación de esta tecnología en pruebas deportivas es evidente, cada atleta lleva un chip RFID que comienza el temporizador en salida y se para en llegada. Este sistema es en el que se basa la mayoría de sistemas actuales, con sus respectivas mejoras con forme pasan los años, pero aun funcionando con el mismo sistema.

Como bien hemos visto a lo largo de la historia, e hilando con el principio de este punto, si miramos la evolución natural del cronometraje de pruebas deportivas podemos darnos cuenta que hay una recursividad, el cronometraje siempre tiende a avanzar junto con la tecnología de la época. Es aquí donde está enmarcado este TFG. La inteligencia

artificial y el reconocimiento de imágenes son tecnologías, aunque no estrictamente nuevas, emergentes y de las que empresas como Microsoft, Google y Apple se están haciendo eco y están aplicando a sus nuevos productos. Es por esto que la evolución natural que el cronometraje deportivo debería de tomar es el uso de estas tecnologías para su beneficio, aunque como ya veremos, no todo es color de rosas con estas tecnologías.

Gracias a mirar al pasado podemos ver más claramente que es una evolución natural y necesaria, aunque teniendo sus limitaciones será revolucionario para el sector y podría marcar un nuevo comienzo de digitalización en un deporte con un carácter tradicional muy marcado como podría ser el atletismo popular, ayudando también a la cohesión con otros sistemas más asentados como la medición de constantes vitales (pulso o respiración) o ritmo del atleta.

## 2. Tecnologías existentes

### 2.1. Tecnología RFID

Como se ha comentado anteriormente, la tecnología más usada actualmente es la tecnología RFID. Prácticamente el 95% de eventos deportivos cuentan con esta tecnología, fácil de usar, posicionada y con una gran comunidad y empresas que las comercializan. Vamos ahondar un poco más en los aspectos técnicos a lo largo de este apartado para que todo aquel que no sepa de ella quede con una muy buena idea del funcionamiento principal, aplicaciones, ventajas, desventajas y las empresas que la comercializan.

#### 2.1.1. ¿Qué es la tecnología RFID?

RFID (Radio Frequency Identification) como su nombre indica es un sistema basado en el uso de ondas de radio para la identificación de un objeto único, normalmente una etiqueta o chip que se deposita dentro o se adhiere a lo que se quiere identificar, pudiendo contener desde un simple número de serie hasta información más compleja que ayude a la identificación en sistemas más complejos. La forma en la que funciona depende del tipo de etiqueta de la que se trate, pero a grandes rasgos consta de un circuito que genera la señal con la información de la etiqueta tras recibir la señal de la antena solicitante de esta información. Existen dos tipos de etiquetas RFID:

- Etiquetas activas: Contienen una batería que activa el circuito integrado, permitiendo la lectura de la información que contiene el chip de la etiqueta y lo envía con la diminuta antena. Tienen un gran alcance (0 a 500 metros) pero una duración de batería limitada y se usan en condiciones más extraordinarias (agua, viento, ...), en nuestro caso, en pruebas acuáticas como triatlones o cuando hace falta lecturas con distancias mayores.



- Etiquetas pasivas: No contiene ningún tipo de batería, a diferencia de su hermana la etiqueta activa, contiene un transductor que transforma las ondas de radio del emisor a corriente eléctrica permitiendo así que la antena envíe la información que contiene el chip, es por esto que la distancia de funcionamiento es menor, pues necesita de una señal de radio entrante mayor. En contraparte a las activas, duran mucho más al no tener baterías, pero fallan a largas distancias alcanzando los 6 metros como máximo.
- Etiquetas semipasivas: Es un híbrido entre activa y pasiva, se caracteriza por tener una batería al igual que la activa pero solo es utilizada para alimentar al chip, pues lo que se busca con este tipo de etiqueta es que el chip guarde información que le envía el emisor para futuras respuestas. Suelen tener el rango de una etiqueta pasiva pero una fiabilidad y durabilidad cercana a la activa.

### 2.1.2. Empresas que utilizan RFID

En el mundo y en el cualquier ámbito esta tecnología está muy extendida y tiene un uso general, por ejemplo: control de acceso, seguimiento y registro de objetos/paquetes (muy utilizado en grandes almacenes), información sobre productos, identificación de animales perdidos, automatización de procesos, y muchos más. Pero no es el uso generalista el que estamos tratando en este trabajo, en el ámbito deportivo se usa en el 99% de las pruebas deportivas populares y en muchas de carácter profesional.

En su mayoría en carreras, desde 5 kilómetros pasando por los 42 y llegando a las ultramaratonas (más de 42km), usándose las etiquetas pasivas. También se utilizan en carreras como duatlones, triatlones o acuatlones, donde se utilizan las etiquetas activas debido a las complicaciones del terreno por donde compiten los atletas.

Son muy variadas las empresas que los comercializan. La pionera fue ChampionChip que introdujo esta tecnología en el sector deportivo, la que le siguieron empresas como Race Timing Systems, Ipico, Winning Time, además existen otras como la española Timing Sense o MyLaps, la antiguamente y ya citada ChampionChip.

Cada una de estas empresas basan sus sistemas en la tecnología RFID para más tarde adaptarlos segundo las necesidades del mercado además de para diferenciarse de la competencia, algunas utilizan ondas de radio con menor frecuencia (campo cercano) y otras con mayor frecuencia (campo lejano).

### 2.1.3. Ventajas y desventajas de RFID

Los sistemas RFID como se puede intuir tienen un gran número de ventajas al ser el actual estándar en cronometraje deportivo, pero no todo es bueno, existen multitud de problemas asociados al uso de ondas de radio, uso y limitaciones técnicas.

En cuanto a las VENTAJAS de esta tecnología:

1. Precio de las etiquetas: Gracias al avance en la tecnología de los últimos tiempos se ha podido alcanzar y miniaturizar circuitos a escalas sorprendentes, además de que este sistema es muy simple, estas dos cosas hacen que el precio de una etiqueta oscile entre los 5 y 10 céntimos por unidad.
2. Flexibilidad: Como hemos visto existen distintos tipos de etiquetas y frecuencias dependiendo de las necesidades.
3. Gran distancia de identificación: Gran amplitud a la hora de identificar los objetos, llegando incluso a los 600 metros.
4. Reutilizables o desechables: Gracias, de nuevo, a la gran variedad de opciones, pueden ser desechado por el poco coste que tienen, o en caso contrario pueden ser reutilizados, por ejemplo, en circuito de pruebas atléticas donde valga el mismo dorsal para distintas pruebas.
5. Rápida lectura: Se trata de un sistema que posee una lectura de la etiqueta que puede llegar hasta los 0.003 segundos.
6. Fiabilidad: Tiene una fiabilidad en el mejor de los casos del 99,8% con una gran cantidad de participantes simultáneos.

Por último, las DESVENTAJAS son:

1. Sensible: Tiene sensibilidad al magnetismo, al ser un sistema que se instala en calzadas y suelos con posibles capacidades magnéticas, puede verse afectado por estas y comprometer la fiabilidad. Además de que si no se escoge correctamente el tipo de etiqueta y no se configuran bien las antenas emisoras y receptoras pueden a ver fallos.
2. Complejidad técnica: Se necesita unos conocimientos avanzados para configurar estos sistemas.
3. Interacción del usuario: Depende en gran parte del usuario que la etiqueta este en buen estado, pues en la mayoría de pruebas, esta se le dé pegado al dorsal que se lleva a su casa. Dependiendo así que no sea maltrato por el atleta.
4. Requieren de un gasto continuo: Aunque las etiquetas pueden ser reutilizables, pocas veces se hace utilidad de esta funcionalidad, suponiendo un coste que, aunque ridículo, existe.
5. Problemas de seguridad: La información que envía el chip o etiqueta puede ser interceptada por usuarios externos a la prueba y por tanto suponer una brecha en la seguridad de la misma.

#### 2.1.4. Uso habitual en pruebas deportivas

El uso habitual en pruebas deportivas se da en conjunto con el dorsal. Usualmente se utiliza la etiqueta RFID recubierta con una espuma solida adherida a la parte trasera del dorsal. El motivo por el cual se usa esta espuma es para evitar el contacto con el cuerpo, pues como hemos explicado anteriormente, las etiquetas que se utilizan para este tipo

de cometidos son sensibles a condiciones con agua o materiales con mucha densidad, como puede ser el cuerpo de un atleta en carrera. En pruebas como triatlón, se utilizan otro tipo de chips o etiquetas, que al tener que estar en todo momento junto al atleta no pueden llevar un dorsal mientras que están nadando (un triatlón consta de una misma prueba de natación, ciclismo y correr), por lo tanto, se utilizan chips normalmente activos encapsulados en una pulsera que lleva el atleta durante toda la prueba. Existen, además, otras alternativas en las que el chip va enrollado en los cordones del zapato, ubicación que puede ser beneficioso para chips con menor capacidad de transmisión, pero va en desuso.

En cuanto a los sistemas que reciben las señales que envían los chips o etiquetas, son en la mayoría de los casos, unas especies de alfombras con antenas incorporadas a lo largo de la misma que captan las ondas de radio emitidas por las etiquetas y se envían de forma alámbrica a los sistemas cronometradores, en los que no voy a entrar en detalles pues cada empresa fabricante tiene su propio sistema.

## 2.2. Tecnología de reconocimiento facial

Como hemos visto a lo largo de la historia, los avances tecnológicos se ponen a disposición del público general para hacer nuestras vidas más sencillas. Los avances de la inteligencia artificial y el reconocimiento de imágenes en los últimos años hacen que el siguiente paso natural del cronometraje deportivo sea el uso de reconocimiento facial en los eventos. No solo haría la vida más fácil a los cronometradores que se ahorrarían dinero y esfuerzos, si no también aportaría un gran valor en cuanto a servicios que se lo podrían brindar al atleta y la comodidad que representan. Aunque no todo es color de rosa con esta tecnología, en este punto ahondaremos en su funcionamiento, puntos fuertes y débiles, además de que se expondrán los motivos por los que podría convertirse en el estándar del cronometraje en los eventos deportivos.

### 2.2.1. ¿Qué es el reconocimiento facial?

El reconocimiento facial, por definición, es la acción de un software que identifica a una persona en base a sus rasgos faciales. Esta identificación del rostro humano se hace en base a unos algoritmos matemáticos que tienen como entrada los rasgos faciales (líneas de expresión, forma de la cara, ojos, ...) que, tras procesarse, da un resultado de acierto o error según superen el umbral de confianza definido.

Se puede considerar a Woodrow Wilson Bledsoe como el pionero de esta tecnología pues en 1960 comenzó lo que se considera el primer proyecto de reconocimiento de facial, un sistema de clasificación de rasgos humanos a través de la tabla RAND, aunque como es lógico, no es comparable con los sistemas actuales en tiempo real, pues se trataba de un sistema de clasificación prácticamente manual.

No es hasta el final de los 80 y tras la aportación de algunos científicos al campo, que Sirovich y Kirby aplican el álgebra lineal y supone un gran avance. Mas tarde en 1991, se consigue la primera detección de un rostro humano en una fotografía, marcando un hito en la automatización del reconocimiento facial. Entrando en el siglo XXI, las agencias de seguridad de los Estados del mundo se interesan por esta tecnología, y es cuando se populariza, creándose sistemas avanzados de reconocimiento de objetos y rostros. Mas tarde, a comienzos de 2010 se aplica los avances en redes neuronales convolucionales (popularmente conocido como IA) a este campo, suponiendo el mayor avance hasta la fecha alcanzo niveles de éxito nunca antes vistos.

### 2.2.2. Funcionamiento

Hay que aclarar que el reconocimiento facial es una subcategoría del reconocimiento de objetos que engloba: procesamiento de imágenes, reconocimiento de patrones y visión por ordenador. El funcionamiento simplificado es el siguiente: Se parte de una imagen en la que se tiene que identificar un rostro conocido o desconocido; con el uso de técnicas y algoritmos se identifica un según la cantidad de muestras que tenga ese conjunto de datos el algoritmo devolverá un acierto o fracaso y la confianza de este resultado.

Profundizando en el funcionamiento, esta tecnología cuenta con 5 etapas:

1. Adquisición de la imagen: Como es obvio, se debe adquirir una imagen o video en la que queremos que opere el modelo que hemos entrenado anteriormente, también puede hacerse en vivo lo que requiere de una capacidad computacional superior.
2. Preprocesamiento de la imagen: Se le aplica a la imagen original un tratamiento para asegurar que la imagen que toma el algoritmo puede ser leída con facilidad. Aunque una imagen no esté correctamente tomada, gracias a transformaciones de la imagen se puede obtener un resultado significativamente superior a la original, lo que aumenta los niveles de confianza en este sistema. Aclarar que este paso no tiene por qué ser exclusivo y/o ir antes o después de otro si no que, de hecho, son utilizadas por algunos algoritmos de detección de rostros. Existen diferentes tipos de modificaciones que se aplican a la imagen que dependen del tipo de resultado que se quiere obtener, que a su vez está condicionado por la implementación del algoritmo de reconocimiento facial. Se adentrará en las técnicas que se pueden aplicar en los siguientes puntos.
3. Detección del rostro: Este es el paso principal del proceso de reconocimiento facial, pues con la imagen se localiza la posición de los rostros utilizando algoritmos complejos que trabajan a nivel de pixel, con métodos estadísticos o mediante aprendizaje profundo. Aclarar que para este paso también se utilizan técnicas llamadas de preprocesamiento de imágenes, que se explicaran a continuación, para mejorar el rendimiento y la efectividad de estos algoritmos. Al igual que las técnicas de preprocesamiento de la imagen, los algoritmos que son más utilizados actualmente se explicarán más adelante.

4. Extracción de características: Detectado el rostro se extraen las características propias de la persona que comprenden desde la posición de los ojos, nariz, boca o cejas, hasta la forma de la cara. Estas características se extraen mediante algoritmos que forman un vector para después ser comparados con los almacenados.
5. Reconocimiento: Con los datos en forma de vector se buscan coincidencias en la base de datos, en caso de ser positivo se devuelve la identidad de la persona si supera un umbral predefinido junto con el nivel de confianza con el que se está dando el resultado, o en caso de una comprobación de identidad se devuelve si se trata de esa persona o no lo es.

### 2.2.3. Empresas y reconocimiento facial en el deporte

En cuanto a las empresas que implementan el reconocimiento facial y de imágenes entre sus tecnologías, no existe un sistema ampliamente utilizado ni acogido por grandes o medianas competiciones. Aunque existen distintas aplicaciones en el ámbito de las carreras populares, ninguna ha intentado aplicar esta tecnología al cronometraje de los participantes en base al reconocimiento facial.

Los sistemas de reconocimiento facial en el atletismo se orientan a sistemas de reconocimiento de dorsal, en su mayoría para agrupar las imágenes obtenidas en la carrera y poder compartirlas en línea, ofreciéndotelas mediante una aplicación con marcas de agua para adquirirlas mediante pago. Esto lo consiguen con la implementación de las técnicas explicadas en este trabajo, pero con datasets de dígitos. Existen empresas como PIC2GO que utilizan estas tecnologías para este fin, lastimosamente las tecnologías utilizadas no son de carácter público al tratarse de una empresa privada que basa su producto en esto.

Existen sistemas como el de MarathonFoto que puede implementar algún tipo de reconocimiento facial gracias a una asignación del rostro del atleta con el dorsal, pues pueden asignar fotos en las que no sale el número de dorsal, pero si el rostro, a la biblioteca de imágenes de la carrera del atleta. No se pueden sacar conclusiones certeras debido a la falta de información oficial, pero si sospecha que es un sistema similar, pues suele haber alguna fotografía que no corresponde ni al rostro ni al número de dorsal, lo que denota un error de identificación. En cambio, la empresa Marathons Photo Live implementa este sistema reconocido de forma oficial, además explican que si no consiguen identificar tu número de dorsal deberás adjuntar en su sitio web una fotografía propia de tu rostro para facilitar esta función.



*Ilustración 1. Principales empresas promotoras del reconocimiento facial.*

El sistema conocido que más se puede acercar al propuesto en este trabajo posiblemente sea el de la maratón de Hangzhou International Marathon en China de 2018, donde se puso a prueba un sistema de reconocimiento facial para acabar con las trampas relacionadas con el cambio de dorsales. Los atletas tenían que subir una foto de sus rostros previamente al día de la carrera para asegurar que se trataba de ellos. Este sistema fue desarrollado por Alibaba Group y se desconoce el éxito o no que tuvo el sistema.

En otros ámbitos deportivos, como el fútbol, beisbol o eventos de estadio, se suelen utilizar para labores de acceso o seguridad como, por ejemplo, el control de seguridad en la entrada a estadios ya sea reconocimiento facial para entrar o salir, o facial / postural para prevenir altercados o reconocer a sospechosos. Siendo el primer uso útil conocido en 2001 llevado a cabo por la policía de Tampa (Florida) para la identificación de 19 criminales en un partido de la Super Bowl. Otra función muy utilizada, en el desempeño físico de los atletas, dando datos muy valiosos que, junto con otros sensores, se puede alcanzar un nivel de detalle del rendimiento de los atletas nunca visto.

En España, el deporte rey, hace una apuesta importante por el reconocimiento de imágenes en tiempo real. Los sistemas de la liga como Mediacoach (para el análisis de los partidos), Players (librería de fotos para los jugadores de fútbol) o XGOAL (predicciones de gol en base a imágenes).

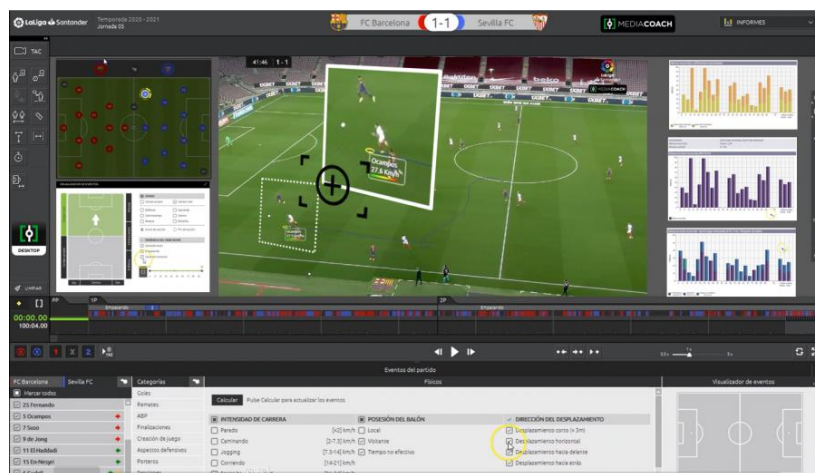


Ilustración 2. Mediacoach de La Liga.

### 3. Marco teórico de la implementación

En este punto se profundizará en las técnicas utilizadas, dando una base matemática y teórica sobre el reconocimiento facial, lo que ayudará a comprender las decisiones y algoritmos que se han utilizado durante el desarrollo.

### 3.1. Adquisición de imágenes

El reconocimiento de objetos y facial depende en total medida de las imágenes que se toman, siendo el resultado final una culminación de las decisiones de diseño con una cimentación en la imagen original, ya sea video o fotografía.

La forma en la que se han tomado las imágenes ha variado mucho con el paso del tiempo, desde la invención de la cámara oscura, de la que toman su nombre las cámaras modernas, no era más que un habitáculo cerrado donde había un pequeño orificio por el que entraba la luz, lo que hacía que se proyectase la imagen invertida del exterior. Entorno al 1840, fueron apareciendo distintas variaciones de cámara oscura, como el daguerrotipo y el calotipo, que mejoraban en prototipo inicial. En 1888, la empresa norteamericana Kodak invento la primera cámara portable, que no requería de ser un experto para utilizarla, desde entonces se extendió la fotografía, llegando en 1930 las cámaras instantáneas y más tarde en 1975 la primera cámara digital portátil.

Paralelamente las cámaras de video marcaban su camino nutriéndose de los avances de la fotografía, pues un video no es más que una cantidad de imágenes por segundo. Fue gracias al cine, que las cámaras de video se desarrollaron de la forma en la que lo hicieron, gracias a Edison y los hermanos Lumière que comenzaron una carrera por la invención de la cinematografía. Pero son a estos últimos a los que se le atribuye la invención del cinematógrafo. Al igual que Kodak en la fotografía, Sony presentó la primera cámara de video verdaderamente portátil.

Desde este punto, las cámaras han avanzado inimaginablemente, permitiendo tomar fotos y videos con un mismo dispositivo, además de que han surgido especializaciones como, por ejemplo, las cámaras deportivas, las cámaras de seguridad, cámaras lentas o cámaras de 360 grados, cada una con sus características que permiten una ventaja competitiva con respecto a las de carácter general o las de otras categorías.

Para el ámbito del reconocimiento facial se pueden utilizar todo tipo de cámaras, pero hay que atenerse a ciertas características para que el resultado sea el correcto. Usualmente el reconocimiento facial suele utilizarse como sistemas de control y seguridad lo que a priori nos hace pensar que el medio físico idóneo para un sistema de cronometraje sería una cámara de seguridad, aunque vamos a repasar que tipos de cámaras serían las idóneas para esta función.

Para que los resultados sean los esperados, es decir, que se realice el reconocimiento facial correctamente, se deben de tener las siguientes consideraciones:

- **Resolución:** La resolución es uno de los puntos más importantes a considerar, pues la capacidad de diferenciar las características de los rostros que habrá en la imagen dependerá directamente de la cantidad de píxeles que tenga la imagen, y por tanto de la resolución. Cuanto mayor sea la resolución más cantidad de píxeles tiene una imagen, las resoluciones más comunes son la HD (1280x720 píxeles), FullHD (1920x1080 píxeles) y UHD o 4K (3840x2160), existen resoluciones tanto inferiores como superiores, pero no se consideran relevantes



para aplicación que nos atañe. La resolución que sería recomendable es la UHD o 4K, debido a la cantidad de detalles que la imagen contendrá, para este caso, la cantidad de gente que puede haber en una salida de una carrera popular supone que se tenga que elegir esta resolución, además de que la distancia entre los participantes y la cámara tiene que ser superior a la distancia de uso convencional.

- **Rango dinámico (HDR):** El rango dinámico es la capacidad que tiene la cámara de captar más detalles tanto en zonas oscuras como brillantes. Esto hace al HDR una característica digna de tener en consideración a la hora de seleccionar un modelo.
- **Velocidad de fotogramas (FPS):** La velocidad de fotogramas hace referencia a la cantidad de imágenes que es capaz de capturar la cámara por segundo, puesto que esencialmente un video es una secuencia de imágenes, cuanto más imagen mayor es la cantidad de información que se puede procesar. Por lo tanto, es recomendable buscar una cámara con una tasa de fotogramas alta. Las tasas recomendables parten de los 30 FPS, 60 FPS (es habitual encontrarnos modelos con 59.94 FPS pues es una limitación heredada de estándares antiguos) o 120 FPS.

A la pregunta de cuál usar, hay que tener en cuenta que una mayor cantidad de fotogramas ayuda a captar más expresiones y por tanto ayuda a que se realice un reconocimiento satisfactorio. Pero hay que plantearse si la actividad deportiva requiere de tantos fotogramas, en nuestro caso al ser una prueba de atletismo, normalmente carreras populares, las velocidades de los atletas tanto en salida como en meta no requieren usualmente de velocidad de captura tan altas como pueden ser los 120FPS o superior. Pues el uso de frames elevados supondría una capacidad de cómputo bastante alta y de una capacidad de almacenamiento superior si encima se utiliza junto a resolución 4K.

- **Sensor y sensibilidad a la luz:** Una buena cámara de video necesita tener un buen sensor que capte la cantidad de luz de forma correcta, ya que puede hacer un día tanto soleado como nublado, o incluso atardeciendo donde la luz es menor y se suele granular y añadir ruido a la imagen. Es por esto que es importante que cuente con un sensor de calidad, además que cuente con adaptación automática a la luz puede ser un punto a favor.
- **Conectividad, fiabilidad y seguridad:** Es esencial que la cámara que se vaya a utilizar tenga conectividades como red, de imagen entre otros. También es importante que cuente con sistemas homologados de protección ante atacantes, debido a que el uso objetivo trata con información sensible como son los datos biométricos. Todas estas características no suponen un gasto menor, así que el dispositivo que se obtenga tiene que tener un mínimo de fiabilidad y calidad de construcción que permita como mínimo amortizar la inversión.
- **Coste:** El coste es uno de los puntos más a tener en cuenta en los proyectos. Hay que tener en cuenta, como es lógico, que si se busca un dispositivo con altas prestaciones el costo será superior, pudiendo alcanzar hasta los 5000€.



- **Ángulos:** Para el reconocimiento facial contar con una lente con ángulos de visión superiores a los normales (inferiores a 60 grados), es decir, una lente gran angular, no es recomendable porque, aunque permitiría captar más detalles, se produce una distorsión en las dimensiones de los objetos que aparecen en una imagen, infiriendo directamente en la propiedad principal que utilizan los modelos de reconocimiento para cumplir su función. No solo es importante los grados de la lente, sino que también lo es que cuente con capacidad de rotación a distancia, es decir, que se pueda regular el ángulo de inclinación sin necesidad de modificar el dispositivo manualmente. Esto último **es esencial** para una identificación correcta, pues el ángulo ideal es el ángulo horizontal, esto en casos prácticos como el que tratamos no es posible, es por ello que un buen grado de inclinación sin comprometer la comodidad de los participantes sería de unos 30 grados del plano horizontal, pues permite colocar la cámara a una altura correcta.

Con obtener una cámara de gran calidad no basta, es a veces más importante la configuración y el posicionamiento que la propia calidad, pudiendo convertir un dispositivo de gama media que obtenga imágenes de mediana calidad en imágenes de gran calidad propias de un dispositivo de gama alta. Es por eso que se requiere de una configuración acorde a los datos arrojados por un análisis previo de la geografía, iluminación y posición.

### 3.2. Técnicas de preprocesamiento

El preprocesamiento de una imagen es una transformación o serie de transformaciones que se le aplican a una imagen con el objetivo de mejorar la información que muestra. Desempeñan un papel fundamental en la mejora de la calidad y legibilidad de una imagen, consiguiendo una mayor tasa de acierto en la resolución final. Aunque se haya situado, en el punto anterior, posterior a la detección de rostro, se puede realizar tanto antes como después, incluso forma parte del propio proceso de reconocimiento de rostros, pues este mismo necesita de transformaciones para el funcionamiento y aumento de las posibilidades de identificación. Los objetivos que se persiguen suelen ser los siguientes:

- **Corrección de iluminación:** La iluminación es uno de los problemas que nos encontramos en las imágenes y alcanzar el nivel de iluminación idóneo en cada situación es prácticamente imposible, puesto que cada lugar posee unas características de iluminación propias y cambiantes. Asumir, a la hora de desarrollar el modelo de reconocimiento, que la iluminación va a ser perfecta suele ser un error común que dificulta su correcto funcionamiento.
- **Eliminación de ruido:** La posición y calidad de la cámara, ondas electromagnéticas, incorrecta configuración de la cámara, ... Todo esto son motivos por los que se puede añadir ruido a la imagen empeorando la extracción

de las características del rostro en fases posteriores lo que afecta directamente a la decisión final del modelo.

- **Normalización de la posición:** La orientación, inclinación y rotación de los individuos es, obviamente, una situación en la que nos vamos a encontrar siempre que se esté haciendo un reconocimiento facial. Además de que la posición y el ángulo en el que se haya instalado la cámara también influye. Es por esto que una de las transformaciones más comunes es la de la posición con respecto a ejes normales para asegurar la igualdad en la comparación de imágenes.
- **Realzar y detectar bordes:** Para ciertos usos que veremos más adelante, el uso realce de bordes puede ser muy útil para identificar y diferenciar características muy importantes como: bordes del rostro, arrugas o diferencias ojos, boca y nariz. Fuera de ámbitos como la detección facial, es muy útil para la detección de números y letras.
- **Normalización de la resolución:** Las imágenes faciales pueden variar en tamaño y resolución, haciendo que pueda afectar tanto al rendimiento del modelo como a la solución final.

Para cada fin existen uno o varios tipos de preprocesamiento, cada uno con sus características de rendimiento, fiabilidad y resultado. Por tanto, no podemos definir cuáles serán las técnicas de preprocesamiento que se usarán ya que depende de las características de la imagen y las implementadas por los métodos que se utilizarán. Una vez enumerados los objetivos que se persiguen con este tipo de transformaciones, procedemos a ver que tipos existen.

### 3.2.1. Transformaciones afines

Se trata de las transformaciones de imágenes más simples, pero de las más utilizadas, pues comprenden operaciones básicas en el tratamiento de imágenes. Estas operaciones trabajan a nivel de pixel, siendo el resultado final, la aplicación de las transformadas a la imagen en forma matricial, es decir:

$$\vec{y} = A * \vec{x}; \text{ donde } A \text{ es la transformada.}$$

- **Identidad:** Método trivial que devuelve la misma imagen de origen.

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- **Reflexión:** Método que devuelve una imagen especular a la anterior, similar a lo que se vería si es figura se reflejase en un espejo.

$$A = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- **Escalado:** Aplicación lineal que aumenta o disminuye la imagen dados unos factores de escala aplicados a cada eje de la imagen. Útil para casos donde es necesario realizar un estiramiento del rostro de un individuo o redimensionarlo.

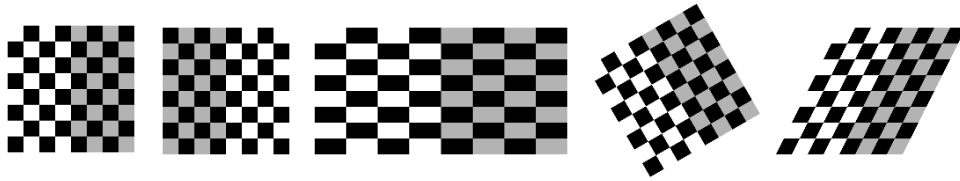
$$A = \begin{pmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- **Rotación:** Cambio en el ángulo de disposición de la imagen. Útil en infinidad de casos, para rotar rostros que se alejen de ángulos próximos a 0 grados.

$$A = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- **Cizallamiento:** Similar en concepto al escalado, pero con diferentes coeficientes, sirve para la corrección de perspectivas de rostros en imágenes tomadas con diferentes ángulos.

$$A = \begin{pmatrix} 1 & c_x & 0 \\ c_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



*Ilustración 3. Identidad; Reflexión; Escalado; Rotación; Cizallamiento.*

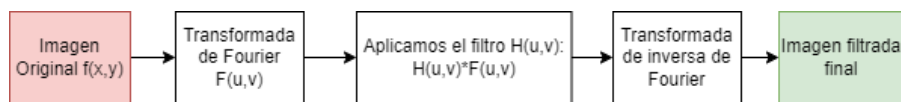
### 3.2.2. Filtros

Los filtros son herramientas fundamentales en el proceso de preprocesamiento de imágenes, empleadas para mejorar la calidad y utilidad de las imágenes faciales. Estas herramientas desempeñan diversas funciones como la reducción de ruido para eliminar información no deseada, el realce de características para resaltar detalles faciales importantes, la corrección de problemas de iluminación para asegurar consistencia en la apariencia de las imágenes, y el filtrado espectral para destacar o suavizar ciertas características espaciales. Existen dos tipos principales, los filtros en el dominio de la frecuencia y en el dominio del espacio.

Los **filtros en el dominio de la frecuencia** procesan la imagen trabajando sobre el dominio de la frecuencia gracias a la Transformada de Fourier de la imagen, siguiendo el Teorema de la Convulación:

$$G(u, v) = F(u, v) * H(u, v)$$

Se sigue el siguiente flujo para aplicar un tipo u otro de filtro:



*Ilustración 4. Etapas en la aplicación de un filtro frecuencial*

Se suelen aplicar dos tipos de filtros  $H(u, v)$ :

- **Filtro paso de baja:** Atenúa las frecuencias altas y mantiene intactas las bajas. Con la aplicación de este filtro se consigue una imagen suavizada, con cambios más suaves entre distintos niveles de intensidad. Muy útil para reducir el ruido de imágenes.
- **Filtro paso de alta:** Atenúa las frecuencias bajas y mantiene las altas. El resultado de la aplicación de este filtro es una imagen con los bordes resaltados (puesto que las altas frecuencias corresponden en las imágenes a cambios bruscos de densidad) y las áreas de tonalidad uniforme atenuadas, lo que facilita la identificación de objetos. Es muy utilizado por la facilidad para la detección de bordes, pero por el contrario genera ruido en la imagen.



*Ilustración 5. Imagen original; Imagen con filtro paso baja; Imagen con filtro paso alta*

En cuanto a los filtros en el dominio espacial se caracterizan por operar directamente en el pixel a diferencia de los frecuenciales, clasificándose como líneas o no lineales.

Los **no lineales** ordenan los valores de los vecinos, dentro de una ventana de tamaño  $n \times n$ , de cada pixel de menor a mayor valor y obtienen el resultando operando sobre esa lista dependiendo del tipo de filtro no lineal que se esté utilizando, suelen utilizarse tres tipos:

- **Máximo:** Se selecciona el mayor valor de gris de la lista ordenada de vecinos del pixel. Este tipo de filtro se utiliza para eliminar el ruido de pimienta (pixeles negros). El resultado de aplicar este filtro es un aclarado de la imagen original.
- **Mínimo:** Se selecciona el menor valor de gris de la lista ordenada de vecinos del pixel. Este tipo de filtro se utiliza para eliminar el ruido de sal (pixeles blancos). El resultado es un oscurecimiento de la imagen original.

- **Mediana:** Se selecciona la mediana de los valores de gris de la lista ordenada de vecinos del pixel. Se suele utilizar para eliminar tanto el ruido sal y pimienta como efectos engañosos creados por este tipo de ruido. En contraposición, se pierden detalles de la imagen original, como líneas y picos.

Los lineales operan basándose en la información de los píxeles próximos a este a razón de una matriz de tamaño  $n \times n$ , conocida como núcleo. Siguiendo el Teorema de la Convolución, la imagen final sigue la formula:

$$I'(x, y) = \sum_{m=-k}^k \sum_{n=-k}^k I(x + m, y + n) * H(m, n)$$

Siendo  $k$  la mitad del tamaño del núcleo y  $H(m, n)$  el tipo de filtro que se quiere aplicar. El tamaño del núcleo y el tipo de filtro dependerá del objetivo de calidad que se quiere obtener incluso del rendimiento con que operará el modelo de reconocimiento facial. Existen distintos tipos dependiendo de los objetivos y artefactos que contenga la imagen original:

- **Filtro promedio:** También conocido como blurring, se utiliza principalmente para quitar pequeños detalles de una imagen. Se trata de un filtro muy simple, ya que consiste en sustituir el valor de cada pixel por el promedio de los vecinos. El resultado de esta transformación es muy similar a la aplicación de un filtro paso de baja, y al igual que este emborrona la imagen y hace que se pierda información.
- **Filtro detección de bordes:** Respecto a la detección de bordes, muy útil para segmentación de objetos y extracción de características de los mismos, se suelen utilizar tres tipos de filtros basados en la búsqueda del gradiente en cada pixel dada una máscara característica de cada tipo de operador:
  - Operador Sobel: Utiliza máscaras de convolución en ambas direcciones, tanto horizontal como en vertical, obteniendo el módulo del gradiente para decidir si se trata de un borde en caso de que sea un valor alto. La máscara viene dada por la siguiente matriz:

$$G_x = \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix} \quad G_y = \begin{pmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

- Operador Prewitt: Al igual que Sobel, opera con máscaras de convolución horizontales y verticales. A diferencia del anteriormente descrito, Prewitt

consta de una menor sensibilidad al ruido. Las máscara de convolución es la siguiente:

$$G_x = \begin{pmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{pmatrix} \quad G_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{pmatrix}$$

- **Operador Robert:** Este tipo de filtro se diferencia por utilizar máscaras de tamaño 2x2. Aunque es sensible al ruido y tiene una respuesta más débil a la detección de bordes que Sobel y Prewitt, es un filtro más eficiente gracias al tamaño de su máscara, lo que lo hace una alternativa viable si se busca un modelo de reconocimiento más rápido. Su máscara de convolución es:

$$G_x = \begin{pmatrix} +1 & 0 \\ 0 & -1 \end{pmatrix} \quad G_y = \begin{pmatrix} 0 & +1 \\ -1 & 0 \end{pmatrix}$$

En la siguiente imagen podemos apreciar el resultado de aplicar los tres tipos de filtro en una misma imagen. Nótese el ruido que genera Sobel en comparación con Prewitt, y la menor fiabilidad en la detección de bordes de Robert en comparación.

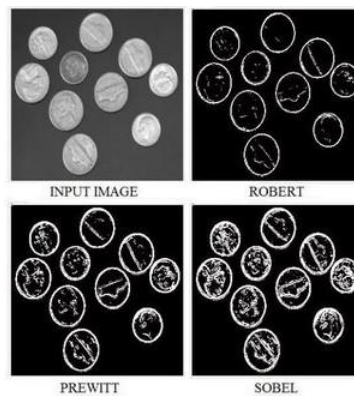


Ilustración 6. Comparativa de los filtros Sobel, Prewitt y Robert

- **Filtro realce:** De forma frecuente se necesitan tipos de filtros en los que se realce la imagen sin causar mucha alteración en ella. En este caso se utiliza el filtro Laplaciano, pero al ser altamente sensible al ruido se suma o resta (dependiendo de la máscara utilizada) a la imagen original, resultando en una imagen un poco más granulada, y que es útil para la detección de características.

$$G_{3 \times 3} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & +4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad G_{5 \times 5} = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & +16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$

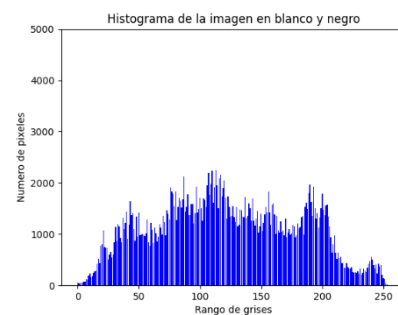
- **Filtro gaussiano:** Similar al filtro blur o promedio, funciona suavizando la imagen según la máscara dada, aunque con una diferencia principal. El filtro gaussiano como su propio nombre indica sigue una función gaussiana bidimensional

matricial como máscara, asignando un peso mayor a la intensidad de los píxeles del centro y disminuyendo conforme se aleja de él. La intensidad de la máscara variará dependiendo de los valores de desviación estándar de la función, es decir, cuanto mayor sea la desviación más amplia la distribución de la intensidad en cada píxel y por tanto más suave. La principal diferencia entre el filtro de promedio y el filtro gaussiano es la capacidad de variación de la intensidad con la que se suaviza la imagen.

$$G_{3 \times 3} = \frac{1}{20} * \begin{pmatrix} 1 & 2 & 1 \\ 2 & 8 & 2 \\ 1 & 2 & 1 \end{pmatrix} \quad G_{5 \times 5} = \frac{1}{273} * \begin{pmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{pmatrix}$$

### 3.2.3. Transformación del histograma

En primer lugar, un histograma es la representación gráfica de la frecuencia (cantidad de píxeles) de aparición de un nivel de gris en la imagen, siendo el eje x el nivel de gris yendo de 0 a 255 de izquierda a derecha (siendo 0 el color negro y 255 el blanco), y el eje y la frecuencia de ese nivel de gris en la imagen yendo desde 0 al máximo de píxeles que tenga la imagen.



*Ilustración 7. Imagen en blanco y negro y su histograma.*

Nótese que todas las técnicas que se comentarán en este apartado están aplicadas sobre imágenes bidimensionales, es decir, en blanco y negro, pudiéndose aplicar en imágenes tridimensionales (a color) aplicando la técnica correspondiente a cada canal, es decir, aplicándose en el histograma del rojo, verde y azul, dando por tanto el mismo resultado a color al combinarse los canales. En lugar de niveles de gris en el eje x, hablaríamos de intensidad de color dependiendo del canal.

Para usar el uso de técnicas de transformación del histograma es necesario definir una función, que representa como va a producirse cada cambio de nivel del gris de entrada de cada uno de los píxeles de la imagen, a cada nuevo valor de nivel de gris de salida que tomará cada píxel.

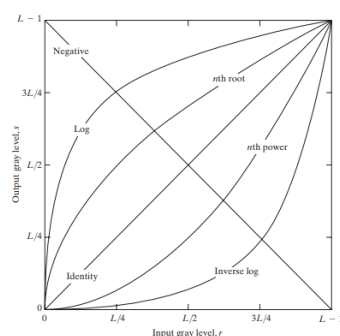


Ilustración 8. Tipos de funciones de transformación simples

Es, por tanto, de vital importancia definir cuál va a ser la función que maquetrará el cambio de nivel de gris de cada pixel porque de ella dependerá plenamente la imagen final. Una vez sabido los conceptos básicos que sirven de base para entender este tipo de transformaciones, es hora de presentar las operaciones elementales aplicables a histogramas:

- **Desplazamiento (Sliding):** Como su propio nombre indica, esta operación se basa en el desplazamiento de los niveles de gris en el histograma según un variable “k” que se sumará o restará (según su valor) al nivel de gris de cada pixel. Es decir, si el pixel 1 tiene un nivel de gris 105 y la variable  $k = 70$ , el nuevo nivel de gris de ese pixel es igual a 175, lo que supone un desplazamiento lateral a la derecha del histograma.

A este tipo de transformación se le tiene que aplicar clipping, que no es más que evitar el desbordamiento para valores mayores de 255 o menores de 0, asignando 255 para valores que superan este mismo y 0 para valores que quedan por debajo de 0.

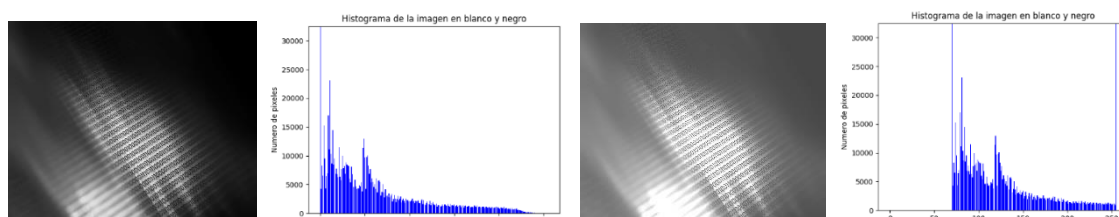


Ilustración 9. Imagen y su histograma original junto con la imagen con desplazamiento  $k=70$  y su histograma.

En las imágenes anteriores se puede apreciar como el histograma situado a la derecha representa el desplazamiento con  $k=70$ , pues todo valor que anteriormente se encontraba inferior al nivel de gris 70 se encuentra desplazado en 70 posiciones a la derecha. Nótese, además, el efecto de aplicar clipping en el último valor de 255 como aumenta la frecuencia de valores. Es, por tanto, fácilmente deducible que el resultado de aplicar este desplazamiento equivaldría a un cambio en el nivel de intensidad de la imagen, en el caso de blanco y negro, se produciría un aclarado en la imagen si el valor de  $k$  es **positivo** y un oscurecimiento si  $k$  es **negativo**. La función de transformación es la siguiente:



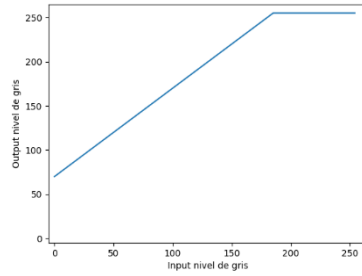


Ilustración 10. Función de transformación para desplazamiento  $k=70$ .

- **Estiramiento (Stretching):** El estiramiento de histograma es una técnica que redistribuye los niveles de gris de una imagen desde un valor mínimo hasta otro máximo, utilizándose así un rango mas amplio de grises que los tenía la imagen original. Resultando en una imagen con mayor contraste, lo que en la mayoría de casos puede aumentar la calidad de la misma diferenciando áreas que antes no podía.

La función de transformación sigue la siguiente ecuación matemática:

$$Output(i,j) = \begin{cases} \frac{255}{Max - Min} (Input(i,j) - Min), & Min \leq Input(i,j) \leq Max \\ 0, & Input(i,j) < Min \\ 255, & Input(i,j) > Max \end{cases}$$

Dándole los valores a Min y Max de 35 y 125 respectivamente, tenemos una función de transformación y la imagen resultante de la siguiente forma:

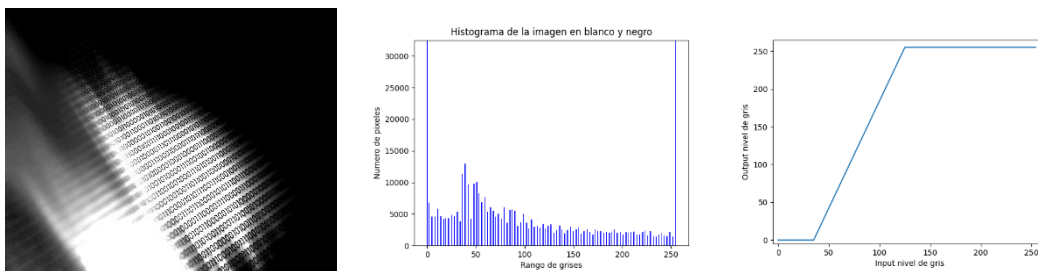


Ilustración 11. Imagen, histograma y función resultante tras aplicar Stretching.

Se puede ver como se distribuyen uniformemente los niveles de grises comprendidos entre los valores 35 y 125, haciendo que aumente en contraste.

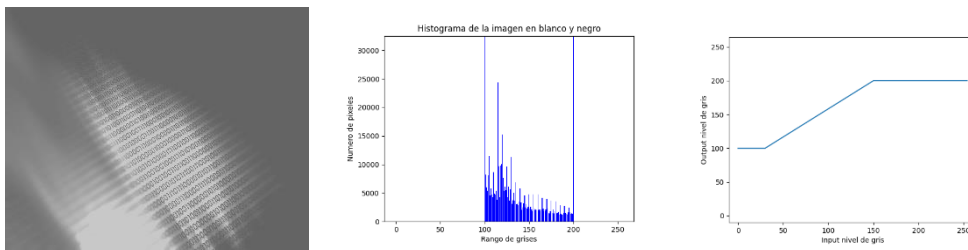
- **Compresión (Shrinking):** La compresión de histograma realiza la operación inversa al estiramiento, por tanto, realiza una reducción de intensidades, es decir, reduce la diferencia entre los niveles de grises de los distintos pixeles. Es necesario definir unos valores de máximos y mínimos de la imagen de entrada ( $I_{max}$ ,  $I_{min}$ ) y salida ( $R_{max}$ ,  $R_{min}$ ) El resultado final de aplicar este método es una

reducción en el contraste, suavizando así las transiciones entre las áreas más oscuras y claras.

La función de transformación sigue la siguiente ecuación matemática:

$$Output(i, j) = \begin{cases} \frac{R_{max} - R_{min}}{I_{max} - I_{min}} (Input(i, j) - I_{min}) + R_{min}, & I_{min} \leq Input(i, j) \leq I_{max} \\ R_{min}, & Input(i, j) < I_{min} \\ R_{max}, & Input(i, j) > I_{max} \end{cases}$$

Dándole los valores a  $I_{max} = 150$ ,  $I_{min} = 30$ ,  $R_{max} = 200$ ,  $R_{min} = 100$ , tenemos una función de transformación y la imagen resultante de la siguiente forma:



*Ilustración 12. Imagen, histograma y función resultante tras aplicar Shrinking.*

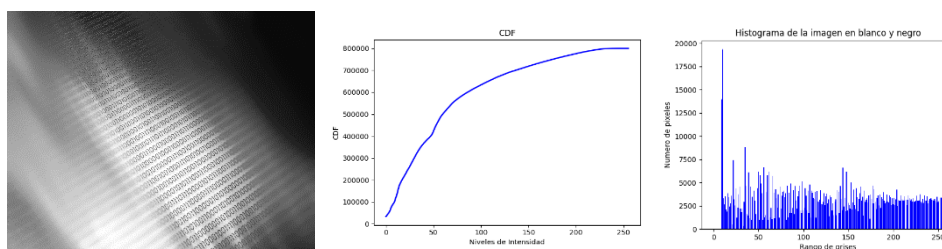
Si se ha prestado atención a los resultados obtenidos en los apartados anteriores se podrá observar que la operación de compresión del histograma se asemeja a una combinación de desplazamiento y estiramiento, pero en ningún caso hay que confundirlos pues cada uno realiza una operación sobre el histograma y tiene principios teóricos distintos.

- **Ecualización:** Es una técnica que busca redistribuir los niveles de grises del histograma de forma uniforme manteniendo la frecuencia de los niveles de la imagen original. Con esto conseguimos un contraste mayor entre las zonas claras y oscuras, permitiendo así diferenciar mejor características que originalmente no se podían apreciar gracias al aprovechamiento de todos los niveles de grises posibles del histograma.

Esta técnica implica el uso de una función de distribución acumulativa (CDF), que no es más que una función que calcula la suma acumulativa de las frecuencias de los niveles de grises del histograma, es decir, una función matemática que nos dice como están distribuidos los niveles de gris en todo el rango de 0 a 255. El cálculo de la CDF para un nivel de gris “i” sigue la siguiente distribución:

$$CDF(i) = \sum_{j=0}^i P(j)$$

Donde  $P(j)$  es la probabilidad del nivel de gris  $j$ , siendo igual a la frecuencia normalizada de este en el histograma. Una vez tenemos la función CDF calculada, que está comprendida entre 0 y 1, para obtener el histograma final se tiene que escalar  $CDF(j)$  para cada " $j$ " a un rango entre 0 y 255, para así poder obtener el nuevo valor de intensidad



*Ilustración 13. Imagen ecualizada, CDF e histograma de la imagen ecualizada.*

Se puede apreciar cómo se realiza un cambio en los niveles de grises sobre todo en la esquina inferior izquierda aumentando la legibilidad, pero sin comprometer zonas muy oscuras como la esquina superior derecha donde no había prácticamente información.

### 3.3. Técnicas de detección de rostros

Las técnicas de detección de rostros distintos algoritmos que pretenden cumplir el objetivo de reconocer rostros de personas de distintas formas dependiendo del tipo que sea. Es normal que se produzcan equivocaciones con este mismo punto y con el propio reconocimiento facial, ya que este último equivale al conjunto de pasos descritos anteriormente que comprenden el sistema completo mientras que la detección de rostros es el proceso de identificación de una cara en una imagen, determinando si se continúa o no con el siguiente proceso.

La elección correcta del algoritmo es un proceso crítico y más en el sistema que se presenta. La no identificación, aunque sea de un solo participante, puede desacreditar el prestigio de la organización. Por eso es de vital importancia no solo hacer un estudio previo riguroso y que se ajuste a cada caso, sino que también sea testeado para asegurarnos de que funcione todo correctamente.

Para elegir la mejor técnica de detección de rostros se debe tener en cuenta la precisión del modelo, teniendo muy en cuenta cuáles tienen las tasas de acierto mayores con el mejor rendimiento posible. Además, también es importante sopesar la velocidad de procesamiento de las imágenes.

Existen muchas técnicas cada una con sus diferentes características, se van a presentar tres de las técnicas más populares, y se verán cuáles son sus defectos y virtudes.

### 3.3.1. Algoritmo Viola-Jones

Este algoritmo presentado en 2001 por Paul Viola y Michael Jones, supuso una revolución en los algoritmos que existían hace ya más de 20 años. Marcó un antes y después en el sector del reconocimiento facial, principalmente en tiempo real pues como se verá, aumenta la rapidez y reduce la capacidad de cómputo notablemente.

Este algoritmo basa su funcionamiento en imágenes en escala de grises, operando gracias a la sustracción de valores de gris de unas regiones definidas como descriptores Haar, restándose los valores de gris que se encuentran bajo la parte blanca menos los que se encuentra bajo la negra. Gracias a esto se pueden detectar características propias de objetos o rostros, pues los descriptores Haar suelen coincidir con características propias de rostros como se puede apreciar en la siguiente imagen.

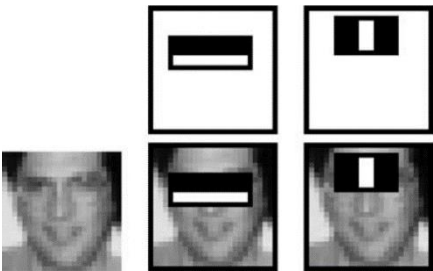
$$\Delta = \text{blanco} - \text{negro} = \frac{1}{n} \sum_{\text{blanco}}^n I(x) - \frac{1}{n} \sum_{\text{negro}}^n I(x)$$


Ilustración 14. Ecuación del cálculo de descriptores Haar y estos con respecto a un rostro.

Los cálculos de todos los descriptores Haar no serían eficientes si no se utilizase la **imagen integral**. Se trata de un array del tamaño de la imagen que representa la suma del propio pixel y todos los superiores y los de la izquierda, guardando la suma en la posición de ese pixel, pero en el nuevo array. Esta operación se repite para cada uno de los pixeles de forma consecutiva. Dando como resultado una matriz con las sumas totales de los pixeles a las que se les pueden realizar operaciones por conjunto, lo que facilita inmensamente la carga computacional que requeriría realizar la suma sobre todo con descriptores de gran tamaño. En la siguiente imagen se puede apreciar que la suma a realizar para el conjunto seleccionado en la imagen original sería:  $46 - 20 - 22 + 10 = 14$

1	2	2	4	1
3	4	1	5	2
2	3	3	2	4
4	1	5	4	6
6	3	2	1	3

Input Image

1	3	5	9	10
4	10	13	22	25
6	15	21	32	39
10	20	31	46	59
16	29	42	58	74

Integral Image

Ilustración 15. Imagen original con descriptor Haar señalado, junto a su imagen integral y la operación a realizar.

Los descriptores Haar por sí solos son capaces de identificar rostros con un porcentaje de acierto un poco superior a la aleatoriedad. Con esto en cuenta el algoritmo de Viola-Jones no supondría una opción a tener en cuenta, en cambio para solucionar este problema de tasa de acierto se utiliza Adaboost. Este es un algoritmo que esencialmente toma los descriptores Haar, clasificadores débiles, y los combina para formar conjuntos

de clasificadores fuertes, haciendo que la tasa de acierto se considerablemente superior.

Adaboost es un algoritmo que requiere de entrenamiento supervisado para ser capaz de obtener los clasificadores fuertes, para ello se entrena al algoritmo con los clasificadores débiles y con datos de rostros, ponderando mayormente los errores en caso de equivocación. Recursivamente, se consiguen unos clasificadores fuertes que serán utilizados en el proceso de detección de la imagen original para detectar si hay o no rostros. Para este último paso se utilizará un método de cascada tomando como entrada una región de la imagen que tiene que pasar unos filtros, siendo estos los definidos por el entrenamiento, es decir, los clasificadores fuertes. En caso de que pase el primer filtro se someterá el segundo filtro, y así hasta que pase todos los filtros y se considerará que en esa región hay un rostro. En caso contrario, que no pase alguno de los filtros se descarta y no se vuelve a comprobar esa área.

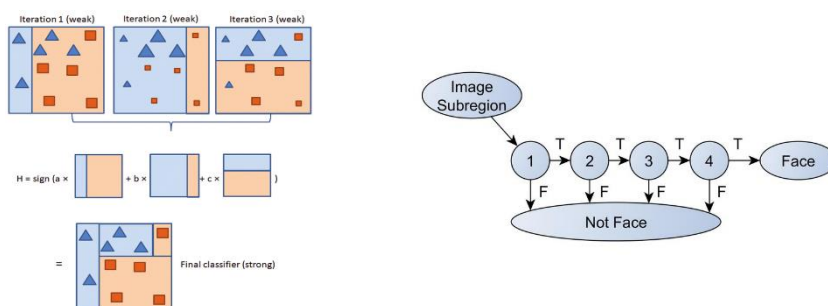


Ilustración 16. Proceso de entrenamiento con clasificadores débiles, junto con el sistema de filtros en cascada.

Las principales ventajas de este sistema son: la rapidez y eficiencia, gracias al uso de imágenes integrales se optimiza en mayor medida la cantidad de accesos a la imagen y los cálculos para cada subconjunto. Por otro lado, la sencillez de conceptos ayuda a la implementación del mismo, requiriendo unos conocimientos matemáticos mínimos. Los dos puntos positivos se ven aún más reforzados gracias a su operabilidad en sistemas de bajos recursos, presentándose como un algoritmo capaz de ejecutarse incluso en un dispositivo con un microprocesador StrongARM de baja potencia sin operaciones de punto flotante a dos fotogramas por segundo.

En contraparte, existen una serie de desventajas notables sobre todo para el dominio de este TFG, como, por ejemplo, la sensibilidad a cambios en los datos que supone que no sea un modelo recomendable para casos en los que aparecen rostros de perfil o con ángulos que se alejen de lo normal, se pueden realizar transformaciones, pero requeriría de un modelo más complejo. Además, requiere que el rostro se encuentre ocupando un gran porcentaje de la imagen, difícil de que se de en la mayoría de ocasiones.

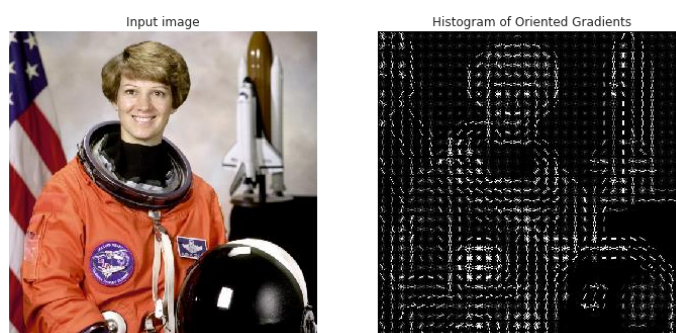
### 3.3.2. Histograma de gradientes orientados (HOG)

Es una técnica ideada por Navneet Dalal y Bill Triggs en 2005, que supone una modernización, tras el éxito de Viola y Jones, de la forma en la que la visión por computación es capaz de reconocer objetos de todo tipo en una imagen, suponiendo una revolución en el mundo del reconocimiento de imágenes e impulsando a la comunidad a mejorar e investigar nuevas modificaciones e incluso a la invención de nuevos métodos.

Este método se basa en la distribución de gradientes de intensidad en una imagen, es decir, la distribución de la diferencia de intensidades. Esto permite que se pueda reconocer formas para su posterior comparación y determinación. Por lo tanto, el objetivo de esta técnica es la generalización de un objeto, en este caso rostros, de tal forma que, a la hora de compararlo con otro objeto de la misma clase en distintas condiciones, pueda tener una distribución de gradientes o características lo más cercanas posibles a esta generalización.

El funcionamiento de esta técnica se divide en etapas, comenzando con una imagen de un solo canal o escala de grises, característica que hace de HOG un método eficiente al tratar con datos de menor dimensión. Para reducir los defectos de iluminación y sombras que pueda haber en la imagen se normaliza la imagen, se pueden utilizar múltiples técnicas, tomaremos como ejemplo la ecualización del histograma, técnica explicada en puntos anteriores.

Tras la normalización y ecualización de la imagen, esta queda preparada para el comienzo de la siguiente etapa, el cálculo de gradientes. En este contexto, un gradiente es el cambio de intensidad (nivel de gris) entre un pixel y otro. Para esta etapa el cálculo de los cambios de intensidad se hace entre pixeles contiguos tanto en la dirección horizontal como vertical, esto se suele representar con una línea que representa la dirección donde existe un cambio abrupto de intensidad, ya pueda ser de claro a oscuro o viceversa.



*Ilustración 17. Imagen y su representación con HOG, donde se pueden ver los gradientes.*

A continuación y con la intención de reducir la dimensionalidad y aumentar el rendimiento, se agrupan los gradientes en celdas, usualmente de 8x8 pixeles y se crea

un histograma de los gradientes de cada grupo, es decir, se cuentan las direcciones de cada uno de los píxeles y se agrupan según el comportamiento (dirección del gradiente) general, normalmente se suelen agrupar en 9 comportamientos que representan de 0 a 180 grados, pero depende de la implementación que se realice pueden ser más o menos comportamientos, lo que representa más o menos detalle en los ángulos en los que hay un cambio de intensidad.

En la siguiente fase de esta técnica de nuevo volvemos a agrupar esta vez en grupos de 2x2 celdas, es decir, de 16x16 píxeles. Se concatenan los histogramas de cada una de las celdas y se normalizan, normalmente usando normalización  $L_p$ , donde  $\epsilon$  representa una constante pequeña, para evitar división entre cero:

$$v_{normalizado} = \frac{vector_{concatenado}}{\sqrt[p]{\|vector_{concatenado}\|^p + \epsilon^2}}$$

Esto se repite para cada uno de los grupos de celdas y se concatenan, dando como resultado un vector de características HOG que contiene información sobre la forma y la estructura de la imagen.

La implementación de los distintos recursos matemáticos explicados y que constituye a la técnica de HOG como, por ejemplo, el cálculo de gradientes, hace que pueda procesarse una imagen de la misma manera aun si es muy oscura o clara, lo que aumenta considerablemente las opciones de éxito y la robustez ante las detecciones. Al igual que las agrupaciones de gradientes, que reduce la cantidad de datos que maneja la técnica.

Como conclusión, esta técnica ha sido ampliamente aplicada en campos variados, desde la detección de rostros, pasando por detección de peatones y hasta escenas complejas, consolidándose como una herramienta muy importante en la visión por ordenador. A pesar de sus fortalezas, el rendimiento de HOG puede verse afectado por la escala y orientación del objeto, así como por la complejidad del fondo en las imágenes. HOG continúa siendo un método versátil y robusto para la extracción de características visuales, aunque hayan pasado prácticamente dos décadas, enriqueciendo y ampliando las capacidades de los sistemas modernos de visión artificial.

### 3.3.3. Redes neuronales convoluciones (CNN)

Las redes neuronales convolucionales son un tipo de red neuronal artificial, a su vez subconjunto de machine learning o aprendizaje automático, que supone una de las ramas más importantes de la conocida como inteligencia artificial. Una red neuronal intenta imitar el funcionamiento del cerebro y las interacciones entre las neuronas, para llegar a un resultado similar al que llegaría un ser humano. Su funcionamiento parte de un conjunto de nodos o neuronas artificial que, conectadas entre sí, producen un resultado de salida.



Este modelo constituye una unión entre el mundo de la medicina, biología y la informática, gracias al trabajo conjunto de estas ramas de la ciencia se ha conseguido una de las herramientas más potentes que el ser humano a conocido y ha sentado las bases para un futuro donde las maquinas trabajarán mano a mano con el ser humano en la detección de patrones como enfermedades, identificación de actos delictivos o compartimiento de masas. Se considera la invención del perceptrón por Frank Rosenblatt en 1957, el detonante de esta revolución. A lo largo de los 80s y 90s se fueron añadiendo los conceptos que hoy día componen la base de una red neuronal.

Las redes neuronales convolucionales funcionan gracias a una amplia interconexión de nodos, separados por capas. Estos nodos presentan una estructura similar al perceptrón propuesto por Frank Rosenblatt, tenemos unos inputs de información llamados conectores (ya pueda ser una imagen o datos) y un peso asignado a cada uno de ellos, similar a las dendritas de una neurona. Para procesar la información proveniente de los conectores se encuentra suman sus entradas multiplicado por los pesos que se le han dado a cada una de ellas, a esto se le conoce como función de red o propagación. Por último, la salida se obtiene aplicando una función limitante al valor de la función de red a la que se le llama función de activación, se pueden utilizar distintos tipos, pero la más usada suele ser la sigmoidea gracias a su similitud con las neuronas biológicas.

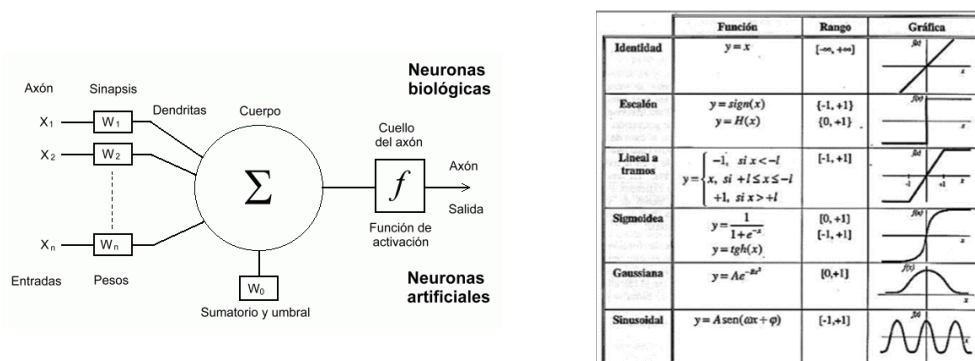


Ilustración 18. Estructura de una neurona artificial y tipos de funciones de activación.

A las agrupaciones de neuronas se les llama capas, normalmente existen entre decenas y centenas de capas de los siguientes tipos:

- **Capa de entrada:** Esta es única y se ubica al principio del proceso. Recepciona los datos de entrada, como una imagen o audio. Para una imagen de 10x10 resultará en una capa con 100 nodos.
- **Capas convolucionales:** Estas son de vital importancia en las CNN, aplica una convolución de varios de los filtros mostrados en el punto de técnicas de preprocesamiento. En esta capa hay múltiples nodos, cada uno realiza la convolución de los datos de entrada con un kernel o filtro asignado.

Estas operaciones las realizan cada nodo en paralelo, además de que se comparten los pesos para cada uno de los nodos. Gracias a la aplicación de estos filtros se obtienen mapas de características (feature mapping) que representan

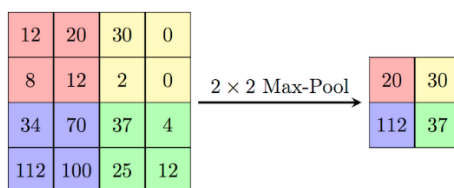


la activación de ese nodo y que se utilizarán como entrada para las siguientes capas.

Es decir, si se escogen 10 filtros, existirán 10 nodos y cada uno devolverá un mapa de características dependiendo del filtro escogido. Si se usase una imagen de 10x10, con la convolución de 10 filtros de 3x3 se obtendría un total de  $(10-3+1) \times (10-3+1) \times 10 = 640$  nodos, que representan las 10 imágenes filtradas de tamaño 8x8 (10 matrices).

- Capas de activación (ReLU): Su labor es la devolver el valor de entrada si este es positivo o 0 en caso de que sea negativo. Estas capas son muy simples pero muy útiles, ayudando a la eficiencia del modelo.
- Capas de agrupación: Por ejemplo, si se escoge una imagen de 10x10, como se ha desarrollado los ejemplos anteriores, aplicar otra capa de convolución de 10 filtros, esta vez con 640 nodos de entrada que equivaldrían a una imagen de 8x8 pero con 10 canales, sería igual a  $(8-3+1) \times (8-3+1) \times 10 \times 10 = 3600$  nodos (tamaño de la imagen filtrada multiplicada por los 10 canales y por los 10 filtros), continuando las iteraciones: 16000 y por ultimo 40000.

Se puede observar con este ejemplo que el incremento es exponencial, es por ello que existe esta capa, cuya función principal es la de reducir la dimensionalidad de los datos de entrada. Principalmente se usan el max-pooling y el average-pooling, siendo la más usada el max-pooling. Esta técnica funciona de la siguiente manera, se recorre la imagen dependiendo del tamaño del filtro y escoge el pixel con mayor nivel de color o gris, por ejemplo, si es un max-pooling de 2x2 se recorre en grupos de 2x2 pixeles y se escoge el pixel de mayor valor.



*Ilustración 19. Max-pooling de 2x2.*

Estas capas excepto, la de inicio se repiten cuantas veces sea necesario para obtener el resultado deseado. Con un mayor número de estas tres capas se presume que el modelo será más profundo, con una capacidad superior de aprendizaje de características complejas y de generalización para datos desconocidos, además de que evita el infra ajuste. No todo es positivo, pues resulta más difícil de entrenar, puede incurrir en errores de precisión como el sobreajuste.

- Capa completamente conectada: Una vez terminado las capas de convolución y agrupación, el resultado de todo el proceso se aplana transformándolas en neuronas tradicionales, y se conectan con otras capas completamente conectadas formando una red neuronal tradicional.

Dependiendo de la casuística, antes de la capa de salida se pasa el resultado de las capas completamente conectadas por una función de activación llamada Softmax. Se trata de una función de clasificación multiclase que normaliza la capa de salida permitiéndonos así interpretar los datos correctamente, pues transforma el vector de entrada en un vector de salida con las probabilidades de pertenecer a una clase específica. Siguiendo la siguiente formula:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

Donde  $z$  es el vector de entrada de  $K$  dimensiones, es decir, el número de clases. Produciendo un vector  $\sigma(z)_i$  que representa la probabilidad de que la entrada pertenezca a la clase  $i$ .

- Capa de salida: Por último, está la capa de salida que puede devolver una imagen, valores numéricos o cualquier valor dependiendo del dominio del problema.

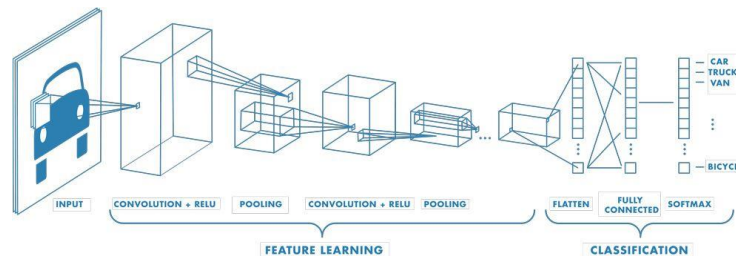


Ilustración 20. Estructura de una CNN.

Para el uso de una red neuronal debe pasar por distintas fases que aseguran que los datos resultantes sean correctos y funcionales para un entorno de producción estas fases son las siguientes:

- Recopilación de datos: Se ha de recopilar tanto los datos de entrada para los que quiere ser usada la CNN como los datos de entrenamiento, testeo y validación. En la mayoría de casos se entrenan con dataset ampliamente reconocidos que aseguran en cierta manera el correcto etiquetado de los datos.
- Definir la composición de la CNN: Este paso consta de la elección del número de capas de convolución, agrupación y completamente conectadas, junto con las funciones de activación que se van a usar y la forma en la que se van a distribuir los pesos inicialmente.

- **Entrenamiento:** Durante esta fase, se busca ajustar los pesos de los nodos o neuronas mediante un algoritmo de optimización que minimiza las variaciones entre las muestras del conjunto de datos y las predicciones del modelo. Es importante tener un conjunto amplio y bien etiquetado para que el entrenamiento, que suele ser la fase de mayor duración, sea lo más óptimo posible para no tener que volver a entrenar el modelo. Existen varios tipos de entrenamiento:
  - **Supervisado:** Dados un conjunto de datos etiquetados, es decir, sus respectivos datos y el resultado ideal, es el proceso por el cual la red neuronal aprende a asociar patrones ajustando sus ponderaciones hasta que el modelo se aproxime al esperado. Esto se consigue a su vez dividiéndose en etapas, donde en un primer lugar se le pasa los datos de entrada dando un resultado que se compara gracias a una función de pérdida (Error Cuadrático Medio o Absoluto Medio, entre otras) que calcula la diferencia entre el resultado y lo esperado. Posteriormente se aplica la retropropagación (o backpropagation), que ajusta los pesos de la red para minimizar la función de pérdida. Por último, se itera hasta que la función de pérdida converja en un punto, entendiéndose que no hay más margen de mejora con el dataset.
  - **No supervisado:** A diferencia del supervisado, este tipo de entrenamiento se caracteriza por no tener etiquetas en los datos, es decir, sin tener una salida esperada. Es por tanto que la red neuronal aprende de patrones que reconoce en la imagen. Existen varias técnicas dependiendo de la finalidad del modelo, pero la más popular es agrupamiento (o clustering). Esta técnica pretende reunir los datos de modo que los que más similitudes tienen los unos entre los otros pasan a formar parte de un mismo grupo o cluster.
  - **Híbrido:** Este tipo de entrenamiento utiliza técnicas tanto del entrenamiento supervisado como el no supervisado, utilizando por ejemplo el entrenamiento supervisado inicialmente para después ajustar el modelo con entrenamiento no supervisado.
  - **Por refuerzo:** Se basa en un sistema de recompensas, donde el modelo intenta maximizar la recompensas acumuladas y obtenidas mediante la interacción del modelo con el entorno.
- **Validación:** Se evalúa el rendimiento resultante del entrenamiento con un dataset de validación, con la intención de reajustar posibles parámetros del modelo o el ajuste de las muestras de entrenamiento.
- **Evaluación:** Se vuelve a evaluar el modelo, esta vez con un dataset distinto a los anteriores y con casos reales. Esto supone el primer contacto con el producto

final, en caso de no ser positivo se debería de repetir las fases hasta lograr un resultado satisfactorio.

Las principales ventajas de este modelo frente a otros son poseen menor sensibilidad a las variaciones de los datos gracias a su generalización y su estructura jerárquica (comienza reconociendo características simples que se combinan dando resultados complejos), esto quiere decir que pueden llegar a detectar individuos con, por ejemplo, gafas o pañuelos mientras que otros modelos son susceptibles a esas variaciones. Además, no se ven alterados por rotaciones del rostro o ángulos diferentes de lo normal, al igual que pueden reconocer rostros de distintas dimensiones en la imagen.

En contraparte, las CNN requieren de una gran cantidad de datos para que el reconocimiento de rostros sea óptimo. Por tanto, el procesado de tanta información requiere de unos dispositivos con grandes especificaciones computacionales.

Como conclusión, los tres modelos funcionan de una forma muy diferente y tienen su fortalezas y virtudes. Mientras que Viola-Jones despunta en velocidad y capacidad de procesamiento por su enfoque en el análisis de características simples, falla en las condiciones de iluminación y posición del rostro. A su vez, HOG tienen los mismos defectos de sensibilidad que su antecesor Viola-Jones, pero en menor cantidad siendo menos susceptible a cambios de intensidad, además también tiene la capacidad devolver un vector de características ahorrando la necesidad de implementar un algoritmo para ello. Por otra parte, CNN no tiene la misma magnitud de fallo ante iluminación y ángulo, tiene mejor capacidad de generalización y es capaz de devolver un vector con el resultado de clasificación multiclase, pero depende mucho de los datos con que sea entrenado y es más lento que HOG y Viola-Jones.

Por tanto, la elección del modelo siempre será dependiendo de estas características y las necesidades de la aplicación.

### 3.4. Extracción de características

La extracción de características es el paso previo al reconocimiento final y supone el proceso de identificar las características de un rostro en una imagen normalmente normalizada para la etapa del proceso en el que se encuentra. Existen desde algoritmos hasta modificaciones en los pasos anteriores para añadir esta etapa a su modelo, dependiendo de los pasos anteriores y la optimización con la que se quiera operar.

En la mayoría de casos los vectores de características vienen dados por el algoritmo seleccionado en la fase de detección de rostros. Para obtener este en una CNN se debería acceder a una capa convolucional de extracción de características tras las capas ocultas, donde devuelva el vector con estas mismas, normalmente es la capa anterior a

la capa de salida que devuelve el vector de probabilidades. En el caso de HOG está diseñado para devolver un vector de características HOG para posteriormente ser analizado.

### 3.5. Reconocimiento

Por último, en la fase de reconocimiento, y una vez obtenido el vector de características del rostro, se suele utilizar la distancia euclidiana como comparador entre dos clases para averiguar las similitudes que hay entre ellas. El único caso que no es necesario esta fase al igual que la anterior es con las CNN que gracias a su estructura de capas convolucionales son capaces de integrar estos dos últimos pasos en uno.

La distancia euclidiana o distancia L2 es la distancia entre dos puntos de un espacio euclídeo, esta se define para dos vectores de puntos P y Q en un espacio euclídeo como:

$$d_E(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

Aplicando el cálculo de la distancia euclídea entre los vectores obtenemos una medida de dispersión, cuanto menor sea esta distancia mayor es la similitud entre los vectores y por tanto mayor probabilidad de que se trate de la misma persona. Existen además distintas variaciones de esta distancia como la distancia Manhattan que también suele ser utilizada:

$$d_M(P, Q) = \sum_{i=1}^n |p_i - q_i|$$

Para que el sistema pueda determinar que es o no una persona determinada, tenemos que definir un valor numérico denominado umbral que servirá como discriminante para decidir.

Otro enfoque sería utilizar el algoritmo KNN (K Nearest Neighbors). Por su simpleza y eficiencia, el método KNN busca los K puntos “vecinos” más cercanos al conjunto de puntos dado haciendo uso de una medida de medición como es la distancia euclidiana, clasificándolo según la etiqueta que pertenezca los K vecinos, que en este caso serían los datos del individuo identificado. Este algoritmo se engloba dentro de la categoría de modelos de "aprendizaje perezoso". Esto implica que solo retiene un conjunto de datos de entrenamiento en memoria en lugar de realizar una fase de entrenamiento previa, todos los cálculos se realizan al momento de llevar a cabo una clasificación o predicción. La complejidad de este modelo reside en la elección correcta del valor K, que tiene que tener un equilibrio entre sesgo y varianza para que sea preciso y correcto.

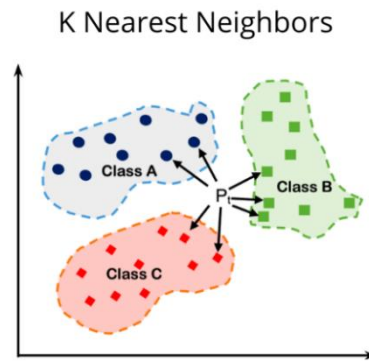


Ilustración 21. Funcionamiento simplificado de KNN.

El problema del sesgo y la varianza es insalvable, pues al aumentar el sesgo hace que la varianza baje y viceversa, es por esto que existen técnicas como la validación cruzada para la elección óptima de este valor. Esta técnica utiliza el conjunto de datos y lo divide en k subconjuntos llamados “folds” (k-folds), esta división sirve para entrenar el modelo k veces rotando en cada iteración el conjunto de datos de entrenamiento con el de prueba. Para obtener el valor de K óptimo se realiza la media aritmética de los valores de error obtenido en cada iteración según la siguiente fórmula:

$$E = \frac{1}{K} \sum_{i=1}^K E_i$$

Una de las técnicas más usada a nivel práctico son las Support Vector Machine (SVM) siendo un modelo muy extendido en cuanto a problemas de clasificación (SVC). En los 90, Vladimir Vapnik y AT&T presentaron al mundo un conjunto de algoritmos de aprendizaje supervisado que solucionaba de forma eficiente el clásico problema de clasificación. El principal objetivo de este algoritmo es de encontrar un hiperplano, que sea capaz de separar de la mejor forma dos conjuntos de datos, aunque podrían ser más (tres, cuatro, cinco ... infinitos). Aclarar que se considera hiperplano a un plano en 3D, una recta en 2D, así dependiendo de la dimensionalidad.

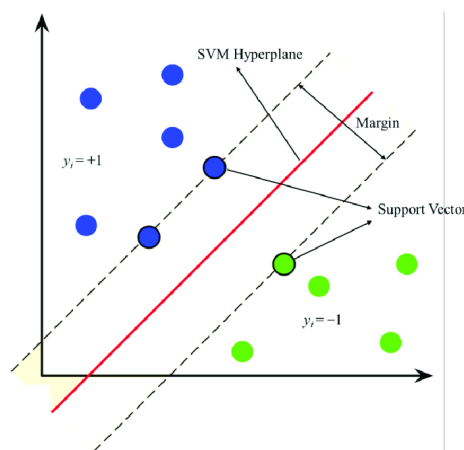


Ilustración 22. Representación en un plano 2D del funcionamiento de una SVM.

A los puntos más cercanos al hiperplano se les llama “vectores de soporte” y cumplen una función muy importante en la tarea de separar un plano de la mejor forma posible. Esto supone la maximización de la distancia más cercana del hiperplano a los datos, es decir, cuanto más separados estén los vectores de soporte del hiperplano mejor será el modelo en características importantes como la capacidad de generalización cuando se presenten datos nuevos, pues estos tendrán más espacio donde clasificarse. A la hora de entrenar al modelo se puede tener en cuenta cierto error pues normalmente, y más en el reconocimiento de rostros, los datos no son linealmente separables, por lo tanto, se utiliza un hiperparametro a la hora de entrenar el modelo para regular este margen de error (soft-margin). A este se le conoce como coste, usualmente es representado por una “C”.

La búsqueda de un valor correcto de “C” es clave para el resultado final del modelo, pues un valor alto significa que se penaliza mucho el error y por tanto el modelo se centrará en evitar este mismo, lo que puede llevar a sobreajuste, lo que culminará en una baja capacidad de generalización. Por otro lado, un valor de “C” bajo puede llevar a un subajuste y a una baja precisión. Normalmente la búsqueda del valor de C correcto se hace de forma manual, comprobando cual se ajusta mejor al problema.

En la práctica, y más en el ámbito del reconocimiento de imágenes, el conjunto de datos a separar no presenta un caso idílico en el que se pueda separar por un hiperplano, es por ello que se usan los “kernels”. Un kernel es una función matemática que permite transformar los datos originales en otros con mayor dimensionalidad, donde posiblemente sea más probable encontrar un hiperplano que divida los datos.

Existen distintos tipos de kernels, pero los principales son tres:

- Kernel Polinomial: Tiene la capacidad de relacionar características no lineales.

$$K_p(x_i, x_j) = (x_i * x_j + c)^d$$

- Kernel RBF o Gaussiano: Es muy efectivo por su capacidad de relacionar características no lineales complejas.

$$K_{RBF}(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

- Kernel Sigmoide: Tiene una gran capacidad de relacionar características no lineales que siguen una curva sigmoide.

$$K_{Sigmoide}(x_i, x_j) = \tanh(\alpha x_i * x_j + c)$$

Para visualizarlo de forma sencilla, se muestra la siguiente imagen que representa al hiperplano definido por los distintos kernels:

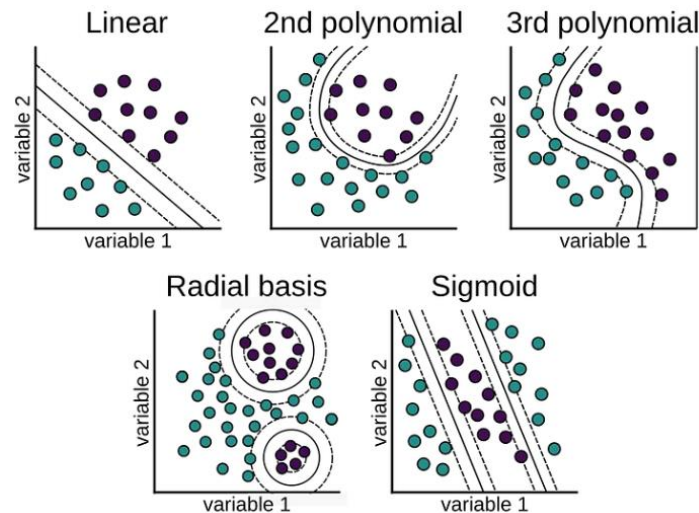


Ilustración 23. Representación de los kernels de forma bidimensional.

### 3.6. Algoritmos complementarios

Existen otras técnicas complementarias que no tienen por qué constituir una fase del proceso de reconocimiento de rostros por sí mismas, pero que pueden ser útiles de implementar junto con los algoritmos de dicha fase. Algoritmos de reducción de dimensionalidad y de categorización, suelen ser útiles para esta casuística. Se adentrará en dos técnicas que podrían servir para el desarrollo de un sistema de reconocimiento de rostros optimo.

El primer algoritmo está especializado en la reducción de los datos, siendo uno de los más conocidos, PCA (Análisis de Componentes Principales). A grandes rasgos, este algoritmo estadístico reduce la dimensionalidad de los datos, mapeando el conjunto de datos a subespacios derivados de menor dimensión, permitiendo así operar con los datos con un coste computacional menor. Este procedimiento se consigue gracias a la transformación del sistema de coordenadas original a uno nuevo donde los datos están representados por componentes principales, nuevas variables creadas por la combinación lineal de los datos, que representan las direcciones en las que la variabilidad de los datos es máxima. Para ello se calculan las matrices de covarianza de los datos, para posteriormente extraer los valores (cantidad de varianza) y la dirección que tienen los datos. Una vez hecho los cálculos se seleccionan los “k” vectores con más varianza y se proyectan sobre los vectores, consiguiendo así reducir los datos. No se ha considerado profundizar más en conceptos matemáticos pues no se considera que sea de utilidad para este trabajo.

Es un concepto un tanto complejo y técnico pero que en el contexto de las imágenes toma más sentido de la siguiente manera, el cambio de intensidad de los píxeles representa la varianza de los datos, por lo que en base a eso y la dirección de los cambios de intensidad de los píxeles podemos elegir el número de componentes que se quieren mantener siendo este valor los “k” vectores con más varianza, o también llamado los



números de componentes ( $n\_components$ ). El resultado dependerá directamente de este valor, si el número de componentes seleccionadas es mayor implica que la reducción es mejor y, por tanto, la imagen conservará más características originales. Al contrario, si el número de componentes es menor, se seleccionarán solo los componentes con mayor varianza, resultando en una imagen con menos características, más borrosa.



*Ilustración 24. PCA aplicado a una imagen con distintos niveles de varianza.*

La última técnica que se suele utilizar en el contexto de reconocimiento de imágenes y que puede ser útil para este trabajo es el algoritmo de clasificación de K-Means o K-Medias. Propuesto por Stuart Lloyd en 1957 supone uno de los algoritmos más usados en muchos contextos, desde segmentación de datos o análisis de patrones hasta compresión de imágenes, pero todos tienen algo en común, el agrupamiento de datos. Su funcionamiento principal es la agrupación en K grupos o clusters en base en características comunes, esto se lleva a cabo mediante la definición inicial de K puntos centrales de forma aleatoria (estos serán los centros de los clusters), posteriormente cada dato será asignado al cluster de cuyo centro este más cercano en base a la distancia euclidiana. Después se sustituye el centro del cluster por la media de todos los datos de ese cluster. Esto se realiza de forma iterativa hasta que se alcance un número de iteraciones máximo o la variabilidad del centro de los clusters no cambien.

Esta técnica tiene un coste computacional dependiente de la dimensionalidad de los datos que puede ser un inconveniente en el caso de imágenes. Es por eso que usualmente se utiliza junto a PCA que reduce la dimensionalidad de estos sin modificar mucho los datos, aligerando así la carga computacional de K-Medias. El uso que se le puede dar en el campo del reconocimiento de rostros va desde la clasificación de imágenes, agrupamiento por similitud o detección de objetos.

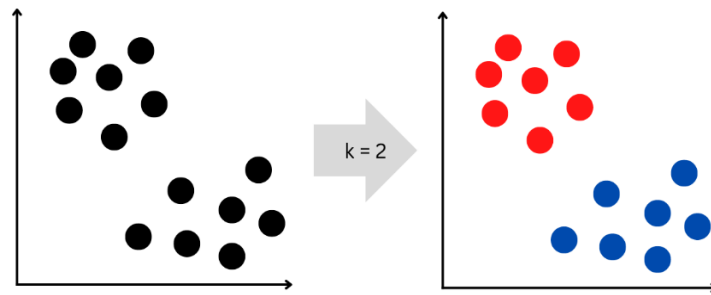


Ilustración 25. Funcionamiento simplificado de K-Medias.

## 4. Marco Práctico

En este punto se describirá en que ha consistido el desarrollo de la parte práctica. Se describirá la metodología propuesta para desarrollar un modelo viable del sistema, igual que se explicarán tanto las tecnologías usadas como los algoritmos aplicados y de los que se hará un análisis previo de los óptimos según los resultados y el contexto de la aplicación.

Como recordatorio, se propone un sistema de cronometraje basado en reconocimiento facial, con tecnologías basadas en modelos de aprendizaje automático e inteligencia artificial que permita saber con certeza el rostro del participante para su posterior procesamiento y contabilización del tiempo en carrera.

### 4.1. Tecnologías utilizadas

En primer lugar, para todo proyecto de programación la selección del lenguaje en el que se va a cimentar es el punto de partida del viaje, por tanto, la correcta elección del lenguaje de programación ahorrará o supondrá dolores de cabeza dependiendo de la elección. En este caso el lenguaje seleccionado es Python.

Python es un lenguaje de programación de alto nivel creado por Guido van Rossum y utilizado en muchos ámbitos, siendo la principal opción para analizar y procesar datos grandes, aprendizaje automático o inteligencia artificial. Este lenguaje ofrece una sencillez sintáctica, facilitando a los programadores la tarea de entender, crear y mantener código, lo que predispone a la comunidad a crear proyectos de uso libre (librerías) que otros programadores puedan usar en otros proyectos, ahorrando esfuerzo y agilizando el proceso de creación. Es por esto el motivo por el que se ha elegido este lenguaje, su fácil comprensión y el uso de librerías.

Para este proyecto se han hecho uso de las librerías de OpenCV, Tensorflow y Pytorch, explicadas a continuación:

- **OpenCV:** Es una librería de software de código abierto que se utiliza principalmente para el procesamiento de imágenes. Creada en 1999 por Intel tiene una gran comunidad activa de desarrolladores, que la hacen la herramienta principal para procesar imágenes, visión por computador y visión en tiempo real. Presenta una gran variedad de usos que van desde la transformación de imágenes aplicando los conceptos explicados en la parte teórica, pasando por el reconocimiento de rostros mediante aprendizaje automático y llegando hasta el uso de complejos modelos de redes neuronales para la predicción y clasificación de imágenes. Es muy popular su uso junto a Python.

El uso que se le ha dado a esta herramienta es del tratamiento de imágenes en tiempo real y almacenamiento posterior de estas. Gracias a OpenCV y su módulo de captura de imágenes se puede tratar cada imagen de la webcam para tratarla de forma individual con tan solo una línea y bucle while, lo que facilita enormemente la labor de programación.



*Ilustración 26. Logo de OpenCV.*

- **Tensorflow:** Al igual que OpenCV es una librería, comúnmente utilizada junto con Python (aunque es compatible con C++ y Java), que se centra en el cómputo de tareas de deep learning, aprendizaje automático e incluso de procesamiento de lenguaje natural. Creada y mantenida por Google, comenzó siendo un sistema propietario parte de Google Brain en 2011, que pasó a ser de código abierto en 2015 gracias al tremendo éxito dentro de Alphabet (empresa matriz de Google). Desde entonces ha dado servicio y miles de proyectos suponiendo un ahorro de tiempo y esfuerzo a los desarrolladores, reduciendo el muro de conocimiento necesario para empezar a desarrollar con esta tecnología gracias a que, aunque orientada al bajo nivel, proporciona modelos pre entrenados y dispuesto a ser configurados para el refinamiento del mismo.

El uso dado en este proyecto es el de crear un modelo de deep learning capaz de ser entrenado para reconocer rostros en una imagen y diferenciar entre distintos tipos de rostros almacenados. Por si solo Tensorflow al estar orientado al bajo nivel supondría un nivel de abstracción bajo y por tanto sería difícil y poco

accesible esta tecnología, es por eso que Tensorflow proporciona una API de alto nivel llamada **Keras**. Esta API nos proporciona una interfaz simple y amigable para casos de uso comunes, además de que es modular, configurable y extensible, lo que se traduce en que se puede configurar los distintos módulos de cada modelo añadiendo o substrayendo las capas necesarias para cada uno de los usos que se le quiera dar. Es por todo esto que se usara la librería de Tensorflow y Keras sobre el lenguaje Python.



*Ilustración 27. Logo TensorFlow.*

- **DLib**: Otra librería que se ha utilizado en el proyecto es DLib. Esta biblioteca de código abierto, desarrollada en C++, se destaca por su detector de caras basado en HOG (Histogram of Oriented Gradients), que ha sido aplicado exitosamente en el reconocimiento facial en este trabajo. DLib es conocida por su eficiencia y precisión en la detección facial, lo que la hace una opción popular en la visión por computadora. Al igual que Pytorch, DLib cuenta con una comunidad activa y un amplio conjunto de herramientas que facilitan el desarrollo de aplicaciones de inteligencia artificial. Se ha optado por la implementación de modelos con DLib debido a su robustez y eficiencia en la detección facial, buscando comparar su rendimiento con otras soluciones en el mercado.



*Ilustración 28. DLib.*

## 4.2. Entorno de desarrollo y versiones

Con respecto a entornos de programación se ha utilizado el IDE de la desarrolladora JetBrains para el lenguaje Python, PyCharm. Motivado por la sencillez de la interfaz, completado de código inteligente, sistema de control de versiones, depurador integrado y, sobre todo, que supone un IDE dedicado en exclusiva a Python, a diferencia de otros como puede ser Eclipse con PyDev (herramienta de Eclipse) o Komodo (enfocado a múltiples lenguajes). Se ha rechazado al igual, Visual Studio Code por motivos similares a los anteriores y por ser un editor de código no especializado. Se ha podido apreciar un

aumento de productividad durante el desarrollo gracias al uso de PyCharm. No ha generado ningún tipo de problema, si no justo lo contrario facilitando la labor de programación.



*Ilustración 29. Logo del IDE PyCharm.*

Una de las labores que ha facilitado este IDE es la de creación de entornos virtuales. A modo de resumen un entorno virtual es una herramienta que nos permite aislar el proyecto de dependencias y configuraciones externas, haciendo que solo se aplique a la hora de compilar y ejecutar las configuraciones realizadas sobre este entorno. Esto nos permite instalar y crear tantos entornos como sea necesario con versiones alternativas de las librerías en cada uno de ellos si fuese necesario. Para este proyecto se ha utilizado un entorno virtual de fácil configuración gracias al IDE, aunque para crear uno no es más que usar dos líneas de código en la terminal del equipo.

La versión de Python utilizada en este entorno es la 3.9, la versión más actualizada que maximiza la compatibilidad con todas las librerías. La versión a destacar son las de: OpenCV con la versión 4.10.0.84 específicamente la “contrib”, que brinda un mayor número de herramientas. Tensorflow con la versión 2.12.1. Y por último Dlib, con la versión 19.6. El resto de versiones y paquetes vienen definidos en un archivo requirements.txt en el repositorio del proyecto.

### 4.3. Equipos utilizados

Se ha contado con dos dispositivos para la realización del proyecto, un PC de sobremesa y un portátil. El PC de sobremesa ha hecho de ordenador principal de desarrollo y el portátil ha hecho de ordenador secundario con una capacidad de cómputo superior en ciertas situaciones, que serán explicadas a continuación. Las especificaciones de los dispositivos son las siguientes:

#### **PC sobremesa:**

- Placa base: ASUS TUF Gaming B550-plus WIFI II
- CPU: AMD Ryzen 7 5700 X
- GPU: AMD Radeon RX 6700 XT
- RAM: 32GB DDR4 3200mhz
- Almacenamiento: WD Black SN850X SSD 1TB M.2
- PSU: Corsair CX750 750 W 80 Plus Bronze

#### PC Portátil:

- Marca: MSI
- CPU: Intel Core i7-9750H
- GPU: RTX 2060 Portátil
- RAM: 32GB DDR4 2666MHz
- Almacenamiento: 1TB NVMe PCIe Gen3x4 SSD
- Cámara: 720p

#### Cámara:

- Marca: Apple iPad
- Cámara trasera: 12 Mpx y 4K
- Cámara frontal: 12 Mpx

### 4.4. Configuración de los equipos

La parte práctica se ha realizado en su totalidad en el PC Portátil, mientras que el desarrollo del trabajo se ha realizado en el PC de sobremesa. Aun así, se han probado varios modelos en el PC de sobremesa para el testeo de funcionamiento en otro ordenador, además también se han sacado distintas estadísticas de rendimiento. El uso de esta configuración viene motivado por la falta de una tarjeta gráfica Nvidia en el dispositivo de sobremesa. Esto hubiese supuesto tiempos de entrenamiento de los modelos de inteligencia artificial entre 8 y 9 veces inferiores a dispositivos que si cuentan con una de ellas.

Como se puede apreciar no contar con una tarjeta gráfica de la marca Nvidia supone una gran limitación de la capacidad de cómputo, ralentizando así las pruebas y el desarrollo. Las tarjetas de esta marca cuentan con los núcleos CUDA, unidades de procesamiento de cálculo en paralelo, ideales para los cálculos matriciales y tensoriales propios del tratamiento de imágenes y técnicas de deep learning. Con el mismo nombre, Nvidia proporciona a los programadores una API que capacita para paralelizar subprocesos, por ello y al ser más populares las tarjetas Nvidia, tanto en el mundo profesional como doméstico, propiciando que los principales desarrolladores como Pytorch y Tensorflow opten por el uso de esta tecnología. El homónimo a CUDA en GPUs de AMD serían los AMD Stream Processors, y la API utilizada sería OpenCL, que al igual que CUDA permite acceder a los procesadores para realizar computación paralela.

Además, para la aceleración de modelos de inteligencia artificial Nvidia cuenta con una librería especializada en aceleración por GPU llamada cuDNN. Por parte de AMD hay una librería lanzada en 2016, ROCm, que cuenta soporte oficial de alguna de las librerías más utilizadas, como Pytorch.

Parece que es una desventaja para el desarrollo de redes neuronales tener una tarjeta gráfica que no sea de Nvidia, lo cierto es que lo es. Para afianzar su posición en el mercado de la IA, Nvidia plantea el lanzamiento de un chip especializado para el

entrenamiento de IA, llamado Blackwell, que según cifras proporcionadas por la propia Nvidia supone una mejora en el consumo de 25 veces y, en rendimiento, 30 veces superior al máximo competidor del mercado, es decir, la propia Nvidia.

## 4.5. Desarrollo practico

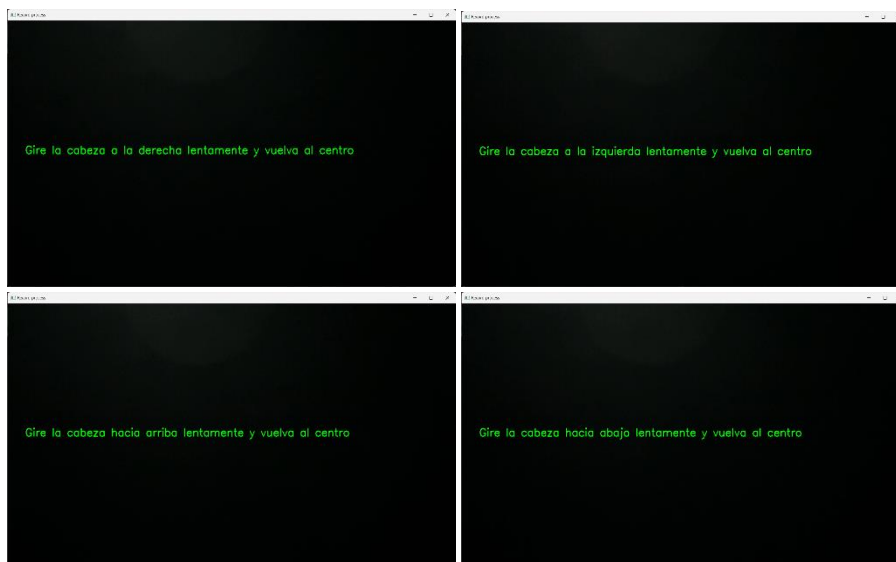
Se han realizado distintos modelos con la finalidad de comparar las diferencias entre métodos clásicos y modernos. En este punto se recorrerá el camino de tomas de decisiones que se han llevado a cabo durante el desarrollo y se concluirá cual es el más efectivo para la tarea de cronometrar una prueba atlética, siempre con la premisa de diseñar un sistema seguro, eficiente y preciso. Se analizará tanto los resultados como las decisiones tomadas para cada uno de estos modelos en base a estas, sus posibles puntos a mejorar y sus fortalezas.

### 4.5.1. Toma de imágenes de los rostros de cada atleta

Un método común básico para todo sistema de aprendizaje automático es la toma de imágenes necesarias para la posterior labor de aprendizaje del modelo elegido. Ha sido un proceso meticuloso y cuidadosamente planificado, asegurando la mayor precisión y calidad en los resultados. Para llevar a cabo esta tarea, se ha utilizado una configuración de cámara fija en un entorno controlado. La elección de cámaras de una alta resolución ha sido fundamental, ya que permiten capturar detalles faciales con la claridad necesaria para el análisis posterior. El posicionamiento de las cámaras ha sido optimizado para obtener una vista frontal de los sujetos, garantizando que cada imagen contenga las características faciales esenciales para un reconocimiento efectivo. Se ha considerado la iluminación como un factor crítico, utilizando una iluminación uniforme y sin sombras, lo que reduce las variaciones y errores durante el procesamiento de las imágenes.

Para esta labor se ha utilizado un iPad Pro como cámara para reconocer la imagen gracias a unos programas privativos llamados CAMO y DroidCam, que, con su plan gratuito, nos proporciona una calidad de imagen de 720p a 60FPS, aprovechando la profundidad y calidad de imagen de la cámara de este dispositivo. Además, se ha utilizado la webcam integrada del equipo secundario para testeo con diferentes tipos de cámaras e iluminaciones.

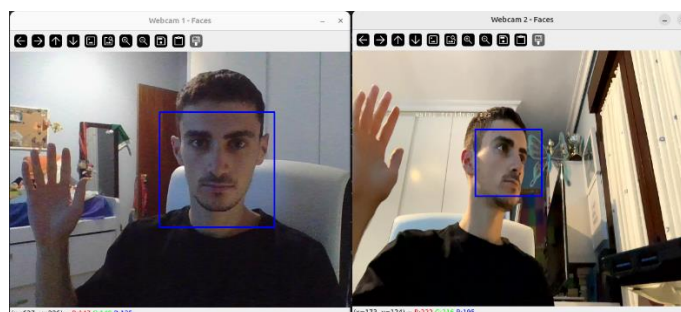
Una vez configurada la cámara, se da comienzo a la toma de imágenes usando un foco que asegura que no haya posibles sombras en el rostro. Se ha organizado en 4 fases, con una duración de 5 segundos cada una: Gira la cabeza a la derecha, giro la cabeza a la izquierda, gire la cabeza hacia arriba y gire la cabeza hacia abajo. Estas 4 fases nos permiten tener un número elevado de fotogramas del rostro y en distintas posiciones, lo que nos servirá para predecir con más facilidad al individuo. Como se ha comentado la toma de estas imágenes es común a los tres métodos, por lo que no se ha hecho diferencia en la implementación para cada uno de ellos.



*Ilustración 30. Cada una de las fases de la toma de imágenes del rostro del participante.*

La toma de imágenes se ha llevado a cabo gracias a la librería de OpenCV y su modulo VideoCapture, que permite capturar la imagen generada por la webcam de forma sencilla y con tan solo un par de líneas.

En cuanto a la implementación, el método principal *“fase\_adquisition\_start\_process”* es el método principal que cumple con la tarea de leer la información recibida por la webcam gracias al módulo VideoCapture de OpenCV (en el código como cv2), el 0 que se pasa como parámetro a esta clase denota el número del dispositivo de video, siendo posible utilizar distintas webcams con tan solo cambiar este número; esto es clave para una implementación real pues cuanto mayor densidad de participantes en línea de salida o meta, mayor será esta, y por tanto requerirá de un mayor número de cámaras. Además, también podría ser muy útil para la captación en la fase de entrenamiento de los modelos, permitiendo alcanzar un mayor número de imágenes desde distintos ángulos. Cabe aclarar que la configuración de distintas cámaras modificaría el código citado anteriormente levemente. A continuación, se muestra un ejemplo de una configuración de doble cámara.



*Ilustración 31. Configuración de doble cámara simultánea.*

Continuando, se comprueba que exista el directorio con número de identificación del atleta por parámetro (DNI sin letra o nombre) donde se va almacenar las imágenes y se procede a entrar en el bucle while que itera hasta que se cierra la conexión con la



cámara. Dentro de este bucle es donde se realiza el almacenamiento de cada imagen, guardando 1 de cada 2 frames totales, es decir: tenemos una cámara de 60 FPS, con 4 fases y 5 segundos por cada una, nos da un total máximo (se explicará más adelante porque máximo) 1200 frames o imágenes almacenadas, pero al guarda 1 de cada 2, tenemos un total de 600 imágenes almacenadas. Con una configuración de cámara múltiple, el total de imágenes seguiría la siguiente ecuación:

$$N_{imágenes} = \frac{FPS_{camara} * S_{fase} * N_{fase}}{2} * N_{camaras}$$

Los métodos “*adjust\_text*” y “*write\_text*” son funciones auxiliares que sirven para ajustar el texto según la resolución del video y para mostrar cada texto correspondiente (“Gire la cabeza a la ...”) respectivamente, para mantener limpieza y legibilidad del código.

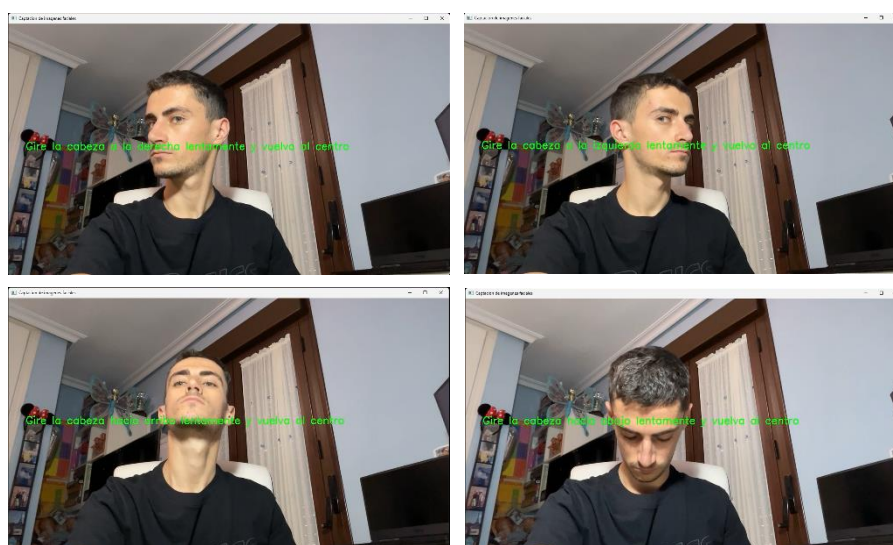


Ilustración 32. Proceso de captación de imágenes.

#### 4.5.2. Almacenamiento de imágenes

Aquí surge la primera toma de decisión del proyecto, el almacenamiento de las imágenes capturadas. Tras un análisis de posibilidades, se redujo a dos opciones, almacenar las imágenes de forma local en el dispositivo o en una base de datos. Optando por la velocidad de procesamiento, las imágenes quedan almacenadas de forma local en el dispositivo mediante sistema de ficheros, guardándose en una carpeta con nombre, el número de identificación (DNI sin letra o el nombre) del atleta. Esto permite que los tiempos de acceso a los datos sean mínimos, pero por el contrario limita la información que se puede almacenar a imágenes o archivos.

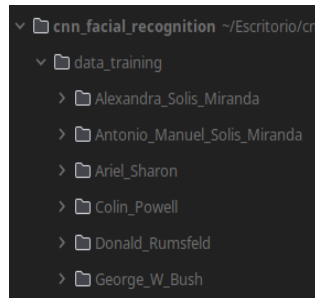


Ilustración 33. Estructura de almacenamiento.

Para mitigar este problema se plantearon dos opciones ambas validas pero con sus limitaciones, la primera es usar un sistema de base de datos NoSQL como MongoDB para el almacenamiento de información, pudiendo almacenar toda la información necesaria y recopilándola con una simple consulta, la base de datos tiene que contener un atributo con la dirección en local de la carpeta con los datos de atleta; la siguiente opción es almacenar una pequeña cantidad de información en el nombre del fichero separado por un carácter especial como la barra baja para más tarde “parsear” el nombre recuperando la información, pudiendo almacenar información como fechas, localizaciones del rostro o evento en el que participa, permitiendo acceder de forma más rápida y sencilla pero sin poder almacenar tanta información.

Cada una de las opciones tiene sus pros y contras, en el caso del uso de una base de datos supone una estructura de los datos más clara y sencilla, al igual que poder almacenar mayor cantidad de información simple, a expensas de tiempos de accesos mayores y posibles fallos de conexión, además de que no se pueden almacenar grandes cantidades de imágenes, como supondría una prueba deportiva. Por el contrario, el sistema de archivos es más robusto y rápido, y puede almacenar datos de grandes dimensiones como imágenes, pero falla en la cantidad de información que se puede almacenar. El uso de ambas opciones abarca todas las posibilidades y minimiza los problemas del uso de una única opción.

#### 4.5.3. Reducción de cantidad de imágenes

Otro de los problemas que surgen es el gran número de imágenes que podría suponer 600 imágenes por participante sin que estas nos proporcionen en su totalidad un beneficio importante, siendo en algunos casos incluso perjudicial (sobreajuste de los modelos). La solución más fácil sería disminuir la frecuencia de almacenamiento de las imágenes, pero esto podría ser negativo pues se puede perder información importante, siendo clave algunos ángulos.

Para solucionar este problema, se ha implementado un sistema de reducción y clustering, es decir, un sistema que combina la reducción de la dimensionalidad mediante PCA (Análisis de Componentes Principales) y el clustering o agrupación en K grupos según la similitud entre las imágenes mediante el algoritmo K-Means (K medias).

El proceso de reducción de imágenes comienza con la extracción de características mediante el uso de una red neuronal convolucional, en concreto con el modelo VGG16, que devuelve un diccionario con el nombre de la imagen como clave y las características como valor. A continuación, se reduce la dimensionalidad de las características mediante PCA de 4096 (imágenes de 224\*224) hasta el requerido, definido por “*n\_components*” que en este caso es 40, lo que supone una reducción considerable. Posteriormente, se aplica K-means al conjunto de las imágenes y características reducidas, con el número de agrupaciones que se desee, devolviéndonos un diccionario con el número de cluster por nombre y los archivos contenidos en el por valor.

Una vez que tenemos la agrupación de las imágenes por similitud solo queda quedarnos con una imagen de cada cluster, que representará a las demás. Es debido a que se agrupan por similitud las imágenes que quedarnos con todas supone un exceso, que puede detonar en falta de memoria para almacenar más imágenes. Por tanto, la elección de la imagen más representativa de cada cluster es, en efectos prácticos, aleatoria, pues en el caso ideal habría que ir seleccionando una a una cual sería la mejor, siendo esto ineficiente y poco práctico. Por último, se eliminan las imágenes sobrantes. Por comodidad se han reducido a 150 el número de imágenes, suponiendo una reducción del 75% de las imágenes totales.

Las ventajas de esta implementación son evidentes, pero las desventajas son apreciables a la hora de la ejecución de este proceso. En dispositivos sin una gráfica Nvidia (testado en el PC de sobremesa) puede llegar a tardar 20 segundos para 120 imágenes, en cambio en dispositivos con gráfica Nvidia (testado en el PC portátil) tarda 10 segundos con los mismos parámetros. Aunque con una tarjeta gráfica Nvidia los tiempos de reducción son menores, siguen siendo altos, pues para 600 imágenes llega a tardar entre 1 minuto y 30 segundos dependiendo de la gráfica. No obstante, tarda un tiempo que dada la función que cumple, no supone mayor problema.

```
def reduce_images(identification):
    start = time.time()
    model = VGG16()
    model = Model(inputs=model.inputs, outputs=model.layers[-2].output)
    data_dict = feature_vector_of_images(model, identification)
    feat = np.array(list(data_dict.values())).reshape(-1, 4096)
    feat_reduced = pca_algorithm(feat.reshape(-1, 4096), n_components=40, random_state=22)
    new_images = kmeans_algorithm(data_dict, feat_reduced, n_clusters=150, random_state=22)
    end = time.time()
    print(f'Tiempo de ejecución: {end - start}')
    delete_images_not_in_list(new_images, identification)
```

*Ilustración 34. Método de reducción de número de imágenes (PCA y K-Means).*

#### 4.5.4. Detección del rostro

El punto más importante en un sistema de reconocimiento facial es la detección del rostro, al igual que los puntos anteriores se trata de un punto en común con los 3 tipos de métodos que se han desarrollado y comparado. Se han usado principalmente dos tipos de tecnologías proporcionadas por dos librerías diferentes, un sistema de

detección basado en el algoritmo Viola-Jones de la librería OpenCV y otro basado en HOG+SVM de Dlib, ambos explicados en el marco teórico. Los métodos que nos proporcionan ambas librerías son similares, diferenciándose como es lógico, en el nombre de los métodos y su funcionamiento interno.

En el caso de OpenCV, se crea un clasificador en cascada “CascadeClassifier()” que implementa un método basado en el algoritmo Viola-Jones y AdaBoost, y al ser un modelo de aprendizaje automático, necesita las características Haar seleccionadas y los umbrales de decisión utilizados en el proceso de aprendizaje, en el caso que nos compete de rostros faciales (un poco más adelante se añadirá para rostros laterales). Cabe destacar que OpenCV usó el dataset CALTECH para obtener estos resultados. Una vez cargado estas características y umbrales, se llama al método “detectMultiScale()” del clasificador, que dado unos parámetros en los que vamos a profundizar, devuelve el rostro detectado en la imagen:

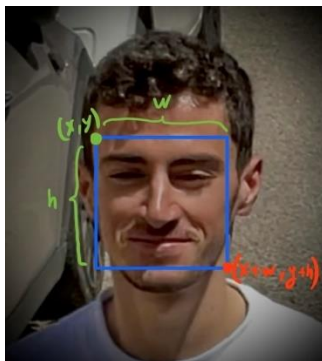
```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
face_frontal = cv2.CascadeClassifier('../training/data/haarcascade_frontalface_default.xml')
frontal_faces_detector = face_frontal.detectMultiScale(gray, scaleFactor=1.5, minNeighbors=7, minSize=(200, 200))
```

*Ilustración 35. Clasificador y detector de rostros frontales.*

- **Imagen en escala de grises**: Es necesario para el detector, pues funciona sobre una sola dimensión.
- **ScaleFactor**: Define el factor porque el que se va a reducir la imagen en cada iteración, en este caso el valor 1.5 quiere decir que se reduce 50%. Es un valor alto, pues normalmente se usa 1.2 o 1.3 que proporciona un equilibrio, pero para la tarea que cumple y al esperar que el rostro en el proceso de recopilación de imágenes sea de gran tamaño en la imagen no es necesario valor inferiores, además de que aporta eficiencia y rapidez en el proceso.
- **MinNeighbors**: Denota el número de vecinos que debe de tener el rectángulo actual para que se considere un rostro válido, por lo tanto, un valor mayor restringe el resultado final, siendo más preciso. Usualmente se usa entre 3 y 5, siendo 7 un valor elevado, pero al igual que el parámetro explicado anteriormente, se busca la reducción de falsos positivos, por lo que se ha utilizado un número mayor. Es importante no generar falsos positivos, pues la existencia de estos afectará en el rendimiento de los modelos entrenados con las imágenes recopiladas.
- **MinSize**: Establece el tamaño mínimo que considerar para que un conjunto de píxeles se considere rostro. El uso de un número mayor, como lo puede ser 200x200, implica que el rostro tiene que estar más cerca de la cámara, para el proceso de captación de rostros esto no es un problema pues se espera que el atleta este cercano a esta.

Este método nos devuelve un array con las coordenadas de los rostros reconocidos, en caso contrario devuelve un array vacío. Una vez se tienen los rostros, el clasificador es capaz de reconocer varios rostros en una misma imagen, tan solo habría que recorrerlo

con un bucle “for”. Cada rostro identificado por el clasificador, viene definido por las coordenadas  $x, y, w, h$  (width y height), según la siguiente imagen.



*Ilustración 36. Imagen de ejemplo con las coordenadas devueltas por el clasificador.*

En la toma de imágenes hay que destacar que las imágenes almacenadas no son completas, es decir, solo se almacena el rostro del atleta. Esto es así porque las imágenes ocuparían gran tamaño y tomarían tiempos de almacenamiento mayores, pues la imagen completa no llega a aportar mucho a no ser que se quiera hacer una revisión manual posterior. Esto nos ayuda posteriormente para entrenar los modelos solo con información relevante para el dominio del problema pues elementos externos al rostro como un objeto o una persona, induciría a error y bajaría la precisión de los modelos.

Dos puntos a destacar en el módulo de toma de imágenes son la implementación de un clasificador de rostros laterales y un sistema de recuerdo del último punto. En primer lugar, se pretende captar un mismo rostro desde distintos ángulos, maximizando así la cantidad de información obtenida. En segundo lugar, el uso de un sistema de recuerdo del último punto es importante, pues como se puede apreciar en este modelo de clasificador Haar en cascada (Viola-Jones), tiende a no identificar rostros en la imagen bastante a menudo, por lo que se pierde información que puede ser relevante. El sistema guarda la última posición identificada de un rostro durante 0.25 segundos, lo que permite que, si el clasificador no encuentra al atleta en ese periodo de tiempo, se use la última posición conocida. Esto supone una mejora considerable en la captación de frames, pues la implementación tendía a no identificar rostros cuando el rostro estaba girando y se encontraba en una posición intermedia entre lateral y frontal, pues ni el clasificador frontal ni el lateral, eran capaces de identificarlo. Además, si el atleta produce un movimiento brusco también es capaz de seguir el movimiento del atleta y captar un gran parte del rostro. Si el atleta sale del plano, se da la vuelta o el sistema no es capaz de reconocer un rostro en la imagen durante más de 0.25 segundos se deja de captar información hasta que vuelva al plano o pasen 20 segundos (4 fases de 5 segundos cada una).

A continuación, se presentan datos sobre el desempeño del clasificador en condiciones de captación de imágenes: Luminosidad ideal, rostro cercano a la cámara y parámetros de configuración óptimos para esta función. (1.5, 7, (200, 200)). Se presentan dos test con el sistema básico y con las mejoras citadas anteriormente.

Nombre de la prueba	Acierto	Error	No identificado
Captación de rostros en tiempo real	92.5%	0%	7.5%

\* Las pruebas en tiempo real, se han realizado según la Ilustración 32. Proceso de captación de imágenes. Los porcentajes mostrados son las medias de las imágenes de los rostros de cada una de las pruebas.

Nombre de la prueba	Acierto	Error	No identificado
Captación de rostros en tiempo real – CON CLASIFICADOR LATERAL Y RECUERDO	99%	0%	1%

En las pruebas a tiempo real se aprecia un rendimiento menor, esto puede ser debido a una calidad de imagen inferior y movimientos de cabeza que clasificador frontal no es capaz de apreciar. En cambio, con las mejoras descritas el acierto alcanza prácticamente al 100% de los frames reales.

En cuanto al desempeño real del modelo en circunstancias semejantes a las que podría ser el día del evento, se ha testado distintas configuraciones (solo con el clasificador frontal) pues algunos parámetros como “minSize” deben de cambiar para poder reconocer a mayor distancia al participante. Las pruebas están evaluadas sobre videos de un corredor en posición frontal simulando la llegada a meta.

Nombre de la prueba: <u>running/4k_01.MOV</u>	Acierto	Error	Frames reales/ideales	Frames obtenidos	Frames no identificado	Frames sobrantes
ScaleFactor: 1.2 MinNeighbors: 5 MinSize: (30, 30)	37.67% (220)	62.33% (364)	240	584	-	344
ScaleFactor: 1.2 MinNeighbors: 7 MinSize: (30, 30)	51.55% (216)	48.45% (203)	240	419	-	179
ScaleFactor: 1.5 MinNeighbors: 5 MinSize: (30, 30)	81.57% (248)	18.43% (56)	240	304	-	64
ScaleFactor: 1.5 MinNeighbors: 7 MinSize: (30, 30)	97.78% (220)	2.22% (5)	240	225	15	-
ScaleFactor: 1.5 MinNeighbors: 10 MinSize: (30, 30)	100% (136)	0% (0)	240	136	104	-

Nombre de la prueba: <u>running/4k_30_01.MOV</u>	Acierto	Error	Frames reales/ideales*	Frames obtenidos	Frames no identificado	Frames sobrantes
ScaleFactor: 1.2 MinNeighbors: 5 MinSize: (30, 30)	61.39% (194)	38.61% (122)	195	316	-	121
ScaleFactor: 1.2 MinNeighbors: 7 MinSize: (30, 30)	85.52% (195)	14.47% (33)	195	228	-	33
ScaleFactor: 1.5 MinNeighbors: 5 MinSize: (30, 30)	96.90% (219)	3.10% (7)	195	226	-	31
ScaleFactor: 1.5 MinNeighbors: 7 MinSize: (30, 30)	100% (204)	0% (0)	195	204	-	9
ScaleFactor: 1.5 MinNeighbors: 10 MinSize: (30, 30)	100% (134)	0% (0)	195	134	61	-



\*Los frames ideales representan el número idóneo de frames que tiene que tener puesto que cada frame del video contiene a la cara del atleta.



Ilustración 37. Imágenes extraídas de los videos de prueba.

En estos frames extraídos de los videos comentados en la tabla de pruebas, se puede apreciar que el video sin el “30” en el nombre, significa que está grabado con un ángulo de inclinación de 0 grados con la horizontal del rostro del atleta. Por el contrario, el que contienen el “30”, significan que están grabados con un grado inclinación de 30 grados (o menos 30 grados) con respecto a la horizontal, simulando la disposición ideal para el reconocimiento facial y la que se aconseja para un resultado optimo.

Las conclusiones extraídas de estos datos, es que se trata de un modelo bastante aceptable para el reconocimiento de rostros frontales en condiciones de luminosidad idóneas y con calidades de imagen alta como se ha podido ver en las pruebas de video. Además de que la configuración de los parámetros que se le pasan al clasificador influye mucho en el resultado, donde se puede apreciar que la mejor opción es:

- ScaleFactor: 1.5
- MinNeighbors: 5
- MinSize: (30, 30)

Esta configuración tiene la mejor combinación en ambas pruebas, pues el porcentaje de acierto es muy alto (97.78% y 100%) y prácticamente no difiere en los números de frames ideales, faltando y sobrando pocos frames en ambas pruebas. Se busca que el número de frames sea lo mejor posible, es necesario una captación del mayor número de frames sin que esto perjudique al rendimiento, siempre y cuando el objeto captado sea un rostro. Una vez obtenido la mejor configuración para este caso, se mostrará el resultado global en banco de videos de prueba:

Nombre de la prueba	Acierto	Error	Frames reales/ideales	Frames obtenidos	Frames no identificado	Frames sobrantes
<u>running/4k 01.MOV</u>	97.78% (220)	2.22% (5)	240	225	15	-
<u>running/4k 02.MOV</u>	100% (128)	0% (0)	212	128	84	-

<u>running/4k_03.MOV</u>	100% (123)	0% (0)	189	123	66	-
<u>running/4k_30_01.MOV</u>	100% (204)	0% (0)	195	204	-	9
<u>running/4k_30_02.MOV</u>	100% (102)	0% (0)	174	102	72	-

Los datos arrojan que se trata de una buena configuración en global, con un 99.5% de acierto, pero que difiere un poco de los frames reales. Esto significa que se pierde (frames no identificados) una media de 47 frames entre las cinco pruebas, lo que podría provocar pérdida de rostros en la línea de meta/salida. Esto podría solucionarse haciendo pruebas reales “in situ” y adaptando el clasificador a los resultados, pero no se puede asegurar con rotundidad que sea capaz de identificar un número igual al número de rostros totales que salen y llegan a meta, siendo una de las características principales de una prueba deportiva.

En el segundo caso, se ha usado la librería de Dlib, la cual apenas difiere en implementación con OpenCV. Las principales diferencias es que Dlib tiene una personalización prácticamente nula, lo que lo hace a su vez más simple y fácil de usar. Esta falta no supone un problema pues al tratarse de un algoritmo diferente como lo es HOG+SVM hace que el proceso y los resultados sean completamente diferentes. Para implementar esta solución, se necesita crear un detector con el método “*get\_frontal\_face\_detector*” que más tarde se le pasará la imagen en escala de grises y que este nos devolverá un array con las caras detectadas, pudiendo acceder a las coordenadas con *face.left*, *face.top*, *face.width*, *face.height*, siendo estas x, y, w y h, respectivamente a la implementación con OpenCV.

Se ha testeado en las mismas pruebas y se han obtenido los siguientes resultados:

Nombre de la prueba	Acierto	Error	Frames reales/ideales	Frames obtenidos	Frames no identificado	Frames sobrantes
<u>running/4k_01.MOV</u>	96.70% (176)	3.30% (6)	240	182	58	-
<u>running/4k_02.MOV</u>	99.05% (210)	0.95% (2)	212	212	-	-
<u>running/4k_03.MOV</u>	100% (189)	0% (0)	189	189	-	-
<u>running/4k_30_01.MOV</u>	100% (193)	0% (0)	195	193	2	-
<u>running/4k_30_02.MOV</u>	100% (136)	0% (0)	174	136	38	-

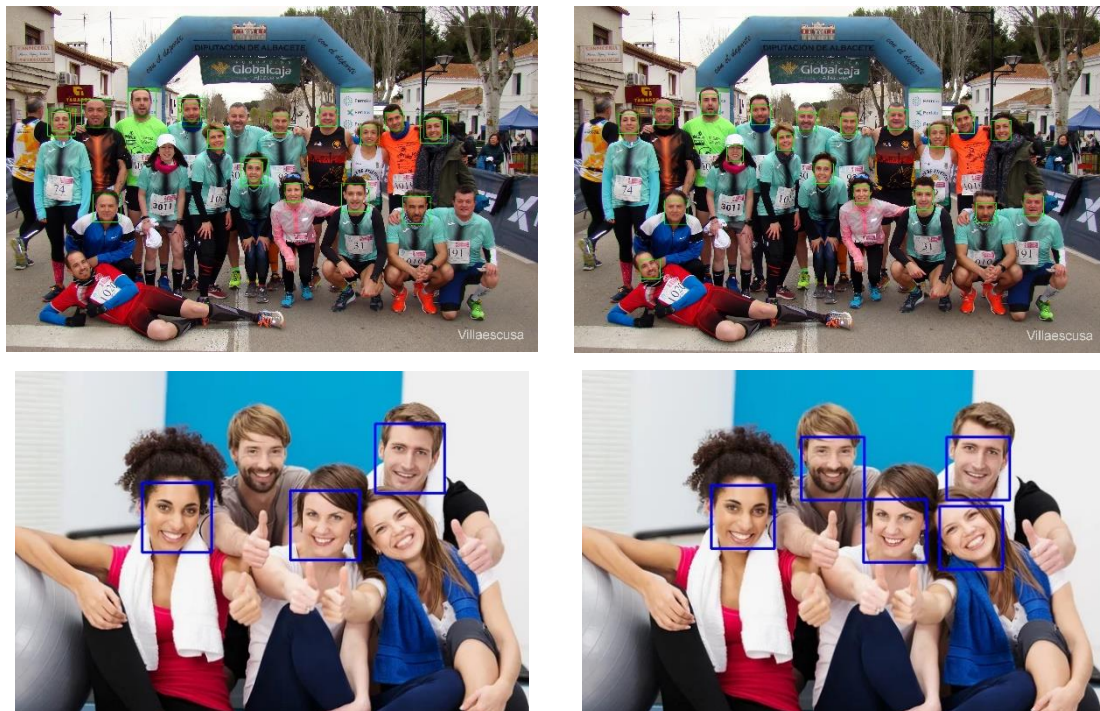
Nombre de la prueba	Acierto	Error	No identificado
Captación de rostros en tiempo real	100%	0%	0%

Según los datos obtenidos se puede apreciar la robustez ante el fallo y la poca variación entre los frames de frames perdidos con una media de 19 frames. Siendo el acierto de



un 99% de media en todas las pruebas, nos encontramos al igual que con el método de Viola-Jones, con un modelo muy robusto y con una tasa de acierto muy alta. Cabe destacar que en una de las pruebas se ha obtenido un resultado perfecto, sin falta ni sobra de frames y con un resultado del 100% de acierto, característica que el modelo basado en clasificadores Haar no ha conseguido.

Para asegurar que modelo puede ajustarse mejor a la necesidad de una prueba deportiva, se ha testado los modelos en imágenes con múltiples rostros y se ha observado una problemática que no había surgido en las pruebas de video, y que se puede apreciar en las siguientes imágenes:



*Ilustración 38. Resultados de aplicar Viola-Jones (izquierda) y HOG+SVM (derecha).*

En la detección de múltiples rostros en una misma imagen, parece que Dlib es muy superior, aunque OpenCV no lo hace tan mal detecta varias veces el mismo rostro, lo que puede inducir a error en los datos, pues en la misma imagen parece que faltan los mismos rostros que sobran, es decir, hay cuatro participantes a los que no se le ha detectado mientras que a otros cuatro se le ha detectado doblemente, pareciendo que ha habido una detección perfecta cuando no lo ha sido. Esto posiciona a Dlib un escalón superior en cuanto a cobertura de necesidades, pues a diferencia de OpenCV, este si ha detectado de forma correcta todos los rostros sin duplicidades.

Comparando ambos modelos, estamos ante dos modelos muy parecidos a nivel de resultados pero que difieren mucho en la implementación. Si se busca opción de personalización, ya sea de parámetros o incluso para la detección de otros objetos, el método de Viola-Jones es el método a elegir, aunque ya hemos visto que para asegurar una alta tasa de acierto se tienen que usar unos parámetros muy rigurosos que hace que el clasificador no identifique rostros que debería. En cambio, Dlib es menos personalizable, pero es capaz de acertar en la detección de todos los rostros deseados

en una imagen. Gracias al algoritmo que utiliza puede identificar rostros en más ángulos y con rostros más laterales.

Rompiendo una lanza a favor de OpenCV, la velocidad que este ha demostrado ha sido superior al de su homónimo Dlib, acabando 3 veces antes que su homónimo Dlib. Esto se traduce en que tiempo real es capaz de procesar 5 frames por segundo más. En este caso Dlib tiene la capacidad de usar una tarjeta gráfica Nvidia para procesar las imágenes más rápido, pero no se ha conseguido instalar esta opción, por lo que es un punto a tener en cuenta para Dlib.

Concluyendo este punto, se ha elegido Dlib por encima de OpenCV por múltiples motivos:

- La capacidad de identificar múltiples rostros en distintas imágenes.
- Mayor ángulo de captación del rostro, abarcando mayor posibilidad de una correcta identificación.

Sería recomendable el uso de esta librería en un ordenador con tarjeta Nvidia, esto requeriría una configuración que no se ha podido obtener, por lo que no se ha podido cuantificar y por tanto tampoco documentar.

#### 4.5.5. Aumento de datos (Data Augmentation)

El aumento de datos es una técnica utilizada en el campo del aprendizaje automático consistente en aumentar artificialmente la diversidad y cantidad de imágenes a partir de una o varias imágenes. El aumento de datos se realiza aplicando técnicas de modificación de imágenes, como las explicadas anteriormente en el punto 3.2 Técnicas de preprocesamiento. Se ha utilizado el módulo de ImageDataGenerator de la librería de Tensorflow.

En un comienzo se utilizaron transformaciones con la librería de OpenCV, pero requería de una gran cantidad de memoria. Por cada imagen se guardaban 5 más, correspondientes a las transformaciones, en total 6 con la original, y si la toma de imágenes del rostro en el proceso de captación de imágenes del atleta recogía 600 fotografías como máximo, había un total de 3600 imágenes por atleta. Esto supondría que en eventos deportivos había que tener unos sistemas de almacenamiento bastante amplio y recorrer todos esos archivos, lo que también ralentiza todo el proceso de entrenamiento.

Por ello se buscó otra alternativa, encontrando el módulo de Tensorflow que permite generar datos de forma dinámica sin necesidad de guardar en el disco las imágenes completas. Esto ha permitido ahorrar un espacio y tiempo considerable.

```

datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

```

*Ilustración 39. Generador de imágenes (Data Augmentation).*

Las transformaciones que se realizan en el código implementado equivalen a:

- **Rotation range**: Especifica el ángulo de rotación que va a tomar la imagen, si tomamos el valor de la imagen, la imagen rotará en un valor entre 20 y -20.
- **Width shift range**: Se refiere a la transformación relacionada con el desplazamiento de una imagen horizontalmente de forma aleatoria entre el valor dado, expresado entre 0 y 1. Por ejemplo: 0.2, se desplazará aleatoriamente entre -0.2 y 0.2.
- **Height shift range**: Igual, pero con desplazamiento vertical.
- **Shear range**: Referente a la transformación de cizallamiento, siendo el valor el rango comprendido donde se tomará un valor aleatorio. Por ejemplo: 0.2, se aplicará cizallamiento con un ángulo aleatorio entre -0.2 y 0.2.
- **Zoom range**: Referente a la ampliación o reducción de una imagen, siendo el valor el rango comprendido donde se tomará un valor aleatorio. Por ejemplo: 0.2, se una ampliación aleatoria entre -0.2 y 0.2.
- **Horizontal flip**: Referente a la transformación de reflexión. El valor es un booleano.
- **Fill mode**: Técnica que se utiliza para rellenar los píxeles faltantes al haber aplicado técnicas de transformación como el cizallamiento o rotación. En este caso se utiliza la técnica del píxel más cercano.



*Ilustración 40. Resultado de aplicar dos veces Data Augmentation para una misma imagen.*

Nótese que el resultado es aleatorio cada vez que se aplica. Lo que es idóneo para el entrenamiento al modelo, aportándole generalización y menor sobreajuste.

#### 4.5.6. Reconocimiento de rostros

Llegados a este punto del trabajo, se procede a explicar los tres modelos principales de identificación y comparación de rostros, que en última instancia completarán el funcionamiento de sistema, cada uno de una forma, características y resultados

diferentes, profundizando en ellos y estudiando el porqué de estos. En este punto se dividirá el proyecto en tres caminos que parten desde un mismo punto en común, los modelos explicados anteriormente y en los que se basa: la captación de imágenes, el almacenamiento y aumento de las mismas, y el sistema de identificación de rostros en una imagen.

Antes de profundizar en este punto, cabe destacar que los modelos que se han utilizado para clasificar e identificar a que atleta pertenece cada rostro capturador difieren en un aspecto importante de los modelos de reconocimiento de rostros utilizados y explicados anteriormente como Dlib y Viola Jones. Esta diferencia radica principalmente en que estos dos últimos modelos no les hacen falta un entrenamiento previo, excepto Viola-Jones que se le pasa un archivo con los pesos, pero que a efectos práctico no ha hecho falta un entrenamiento en este proyecto. Los tres modelos de clasificación por su parte, requieren de este paso para poder diferenciar entre cada uno de los rostros (teóricamente clases) y que representa junto con la configuración del modelo el paso más importante para un correcto funcionamiento. Es por esto, que se ha puesto tanto hincapié en que hay que minimizar la utilización de capturas del rostro erróneas, porque pueden introducir gran cantidad de error en el modelo a la hora de obtener el modelo entrenado.

El primer modelo implementado es un conjunto de técnicas, PCA para la reducción de dimensionalidad de la imagen y SVM para la clasificación. Esta combinación, que denominaremos como PCA + SVM, supone un método clásico de clasificación ampliamente utilizado en problemas de reconocimiento facial.

El proceso de clasificación de rostros comienza con la carga del dataset utilizado. En todos los modelos han sido implementados y probados con uno de los dataset con más recorrido en el reconocimiento de rostros, **Labeled Faces in the Wild**, es importante el entrenamiento de ambos modelos con el mismo dataset para poder compararlos en igualdad de condiciones. Este dataset, creado y mantenido por la Universidad de Massachusetts, es público y cuenta con más de 5749 personas, con un total de 13233 imágenes de rostros sacados de internet de personajes de vida pública. Para asegurar un rendimiento correcto de los modelos, se ha decidido eliminar las personas que no tenga más de 50 imágenes, quedando así un número de personas menores, pero en teoría, con mejor capacidad para diferenciar entre ellas.



*Ilustración 41. Conjunto de muestra de LFW.*

Una vez cargado el dataset, se divide en dos conjuntos de imágenes/datos: los de entrenamiento y los de validación. Esta separación del dataset de entrenamiento es recurrente en los tres modelos, ya que sirve para evaluar el rendimiento del modelo de forma sencilla, gracias a la existencia de métodos en las librerías que proporcionan estadísticas del entrenamiento. Un punto muy importante que hay que tener en cuenta con este modelo es que sea llegado a la conclusión que no es recomendable el uso de aumento de datos con esta implementación, viendo decrementos del 50% en la precisión del modelo cuando se usan los valores de Data Augmentation explicados en el apartado anterior.

A continuación, hay que ajustar el modelo a los datos de entrenamiento (no de validación) mediante el método “fit” de PCA, que calcula los datos necesarios (media, covarianza y vectores) para aplicar las correspondientes transformaciones de reducción, la imagen promedio que devuelve para el dataset utilizado es la siguiente.



*Ilustración 42. Rostro promedio de todas las imágenes del dataset.*

Ya con el modelo ajustado, los datos de entrenamiento son procesados por PCA para transformar una alta dimensionalidad de los datos (los píxeles de la imagen) a un espacio de menor dimensión. Esta reducción de dimensionalidad facilita la generalización y el rendimiento del modelo.

```
def train_and_load_models():
    lfw_people = fetch_lfw_people(min_faces_per_person=50, resize=1.0)
    n_samples, h, w = lfw_people.images.shape
    lfw_people.images = np.array([resize(image, output_shape=(224, 224)) for image in lfw_people.images])
    X, y = lfw_people.images, lfw_people.target
    target_names = lfw_people.target_names
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42, stratify=y)
    X_train_flat = X_train.reshape((X_train.shape[0], -1))
    X_test_flat = X_test.reshape((X_test.shape[0], -1))
    n_components = 250
    pca = PCA(n_components=n_components, svd_solver='randomized', whiten=True).fit(X_train_flat)
    X_train_pca = pca.transform(X_train_flat)
    X_test_pca = pca.transform(X_test_flat)
    # plt.imshow(pca.mean_.reshape(lfw_people.images[0].shape), cmap=plt.cm.bone)
    # plt.show()
    param_grid = {'C': [1e1, 5e1, 1e2, 5e2, 1e3], 'gamma': [0.0001, 0.0005, 0.001, 0.005, 0.01]}
    clf = GridSearchCV(SVC(kernel='rbf', class_weight='balanced', probability=True), param_grid, cv=5)
    clf.fit(X_train_pca, y_train)
    print(f'Best estimator found by grid search: {clf.best_estimator_}')
    y_pred = clf.predict(X_test_pca)
    y_prob = clf.predict_proba(X_test_pca)
```

*Ilustración 43. Creación y entrenamiento de PCA+SVM.*

El número de componentes que se ha utilizado es de 250 (no se utilizó la notación en porcentaje) y los valores de C y gamma del modelo SVM se asignan mediante una



búsqueda del valor más óptimo (GridSearch) entre los siguientes valores: {C: [10,50,100,500,1000], Gamma: [0.0001, 0.0005, 0.001, 0.005, 0.01]}, además el kernel utilizado es RBF.

Una vez transformadas las imágenes mediante PCA, se utiliza SVM para entrenar un modelo de clasificación que puede diferenciar entre diferentes rostros. SVM es efectivo en esta tarea por su capacidad para encontrar el hiperplano óptimo que separa las distintas clases (los distintos rostros) con el mayor margen posible. El resultado es un sistema que puede clasificar nuevos rostros con una cierta precisión, aprovechando las ventajas de ambas técnicas: la capacidad de PCA para simplificar y condensar la información, y la robustez de SVM para la clasificación precisa.

El siguiente modelo se basa en una CNN, pero con una variación denominada ResNet o Red Residual. Esta variación fue presentada por Kaiming He en 2015 en un estudio sobre el reconocimiento de imágenes, y se caracteriza por la capacidad para entrenar modelos con muchas capas, lo que es especialmente útil para el reconocimiento de imágenes. La diferencia principal con una CNN convencional es la existencia de unas conexiones entre capas llamadas residuales.

Estas conexiones residuales, también conocidas como *skip connections*, permiten que la entrada de una capa se salte una o más capas y se sume a la salida de esas capas. Este diseño permite a la red aprender mapeos residuales en lugar de mapeos directos. La arquitectura residual se puede describir de la siguiente manera: si una capa tiene una entrada “ $x$ ” y produce una salida “ $F(x)$ ”, en una ResNet la salida de esa capa sería “ $y=F(x)+x$ ”. Estas conexiones ayudan a mitigar problemas comunes en redes profundas, como el desvanecimiento de gradiente, problema que hace que los pesos en el entrenamiento de una red neuronal cambien poco o no lo hagan impidiendo avanzar en el aprendizaje. Por lo tanto, estas conexiones permiten entrenar redes con muchas más capas de las que sería posible con una CNN convencional, además de que pueden mejorar la precisión y optimización de estas.

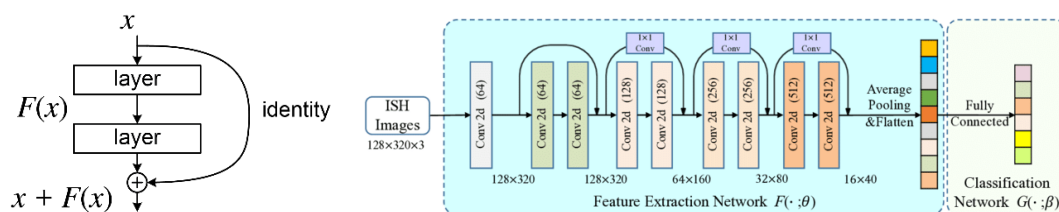


Ilustración 44. Diagrama de una capa residual a la derecha junto con la estructura básica de un ResNet.

Existen varios tipos de ResNet que se denotan por el número de capas después del nombre, de la forma ResNet-N, siendo N el número de capas. Normalmente se usan los siguientes tipos: ResNet-18, ResNet-34, ResNet-50, ResNet-101 y ResNet-152, cuanto mayor número de capas mayor precisión y menor rendimiento, y es por eso que en este proyecto se ha implementado una ResNet-50 (aunque se han testado también la 101 y 152), ideal para tener un balance entre rendimiento y precisión.

Al igual que PCA+SVM se ha utilizado el dataset de LFW con las personas que tenga 50 fotos o más, y además se han añadido imágenes de familiares simulando el proceso de captación de imágenes para testear el modelo con la mayor similitud al proceso real que seguiría el sistema de cronometraje basado en reconocimiento facial propuesto en este trabajo, siguiendo todos los pasos a su vez de reducción del número de imágenes y aumento de datos, pues este modelo a diferencia de PCA+SVM si se ve altamente beneficiado por el aumento de datos.

```
def create_model(p_labels, show_summary=False):
    base_model = tf.keras.applications.resnet50(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
    x = base_model.output
    x = GlobalAveragePooling2D()(x)
    x = Dense(units=512, activation='relu')(x)
    x = BatchNormalization()(x)
    x = Dropout(0.5)(x)
    number_labels = len(np.unique(p_labels))
    print(f'Numero de Labels (personas en el dataset): {number_labels}')
    predictions = Dense(number_labels, activation='softmax')(x)
    new_model = tf.keras.models.Model(inputs=base_model.input, outputs=predictions)
    for layer in base_model.layers:
        layer.trainable = False
    new_model.compile(optimizer=Adam(learning_rate=1e-4), loss='sparse_categorical_crossentropy', metrics=['accuracy'])
    if show_summary: new_model.summary()
    return new_model
```

Ilustración 45. Configuración del modelo ResNet-50.

Se puede observar en la configuración del modelo como se está utilizando el modelo citado anteriormente, pre entrenado con los pesos del dataset de ImageNet, que nos ayuda por así decirlo a centrar el modelo en el reconocimiento de imágenes (no significa que se haya utilizado este dataset, se ha usado LFW). Posteriormente se modifica las últimas capas del modelo base para especializarlo en la clasificación de rostros (fine-tuning o ajuste fino). A destacar de este punto es la adición de una capa Dropout del 50%, es una capa que anula la contribución de algunas neuronas hacia la siguiente capa y deja sin modificar todas las demás, ayudando a prevenir el sobreajuste del modelo.

Por último, se añade una capa de salida “*predictions*” con el número de clases del dataset y la función de activación “*softmax*”. Se compila con el optimizador Adam y se devuelve el modelo.

```
def train_model(p_model, p_images, p_labels, p_minimo_img) -> None:
    le = LabelEncoder()
    labels_enc = le.fit_transform(p_labels)
    save_label_encoder(le, file_path='label_encoder.pickle')
    x_train, x_test, y_train, y_test = train_test_split(p_images, labels_enc, test_size=0.2, random_state=42)
    learning_rate_reduction_callback = tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss', patience=2, verbose=0, factor=0.5, min_lr=0.00001)
    model_checkpoint_callback = tf.keras.callbacks.ModelCheckpoint(filepath='model_checkpoint.h5', monitor='val_loss', verbose=0, save_best_only=True)
    callbacks = [learning_rate_reduction_callback, model_checkpoint_callback]
    v_batch_size = 32
    datagen = tf.keras.preprocessing.image.ImageDataGenerator(
        rotation_range=20,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True,
        fill_mode='nearest'
    )
    datagen.fit(x_train)
    train_data = p_model.fit(
        datagen.flow(x_train, y_train, batch_size=v_batch_size),
        steps_per_epoch=len(x_train) // v_batch_size,
        validation_data=(x_test, y_test),
        epochs=50,
        callbacks=callbacks
    )
    with open('trained_model-history.pickle', 'wb') as file:
        pickle.dump(train_data.history, file)
    p_model.save('trained_model.h5')
```

Ilustración 46. Entrenamiento de ResNet-50.

Del entrenamiento del modelo podemos destacar:

- El uso del 80% del dataset para entrenamiento y el 20% restante para validación.
- Uso de “*Callbacks*”, que no es más que una función que personaliza el funcionamiento del modelo en cada etapa (epoch o época). Se están utilizando dos, una para guardar el progreso del modelo y otra que ajusta dinámicamente la tasa de aprendizaje cuando una de las muestras a dejado de presentar mejoría con el paso de las etapas.
- Uso de Data Augmentation.
- Entrenamiento del modelo con 50 etapas, se ha apreciado que el entrenamiento con más etapas no hace que el modelo aumente en precisión significativamente, aunque si se busca una precisión máxima se podría aumentar. No se ha realizado con más etapas pues supone un aumento en el tiempo de entrenamiento, mientras que la mejora no era considerable.
- Por último, se guarda el modelo utilizando “*pickle*”.

## 4.6. Resultados obtenidos

En cuanto a los resultados obtenidos y presentados son los obtenidos durante la fase de entrenamiento, evaluación y pruebas prácticas reales como las realizadas para el apartado de detección de rostros, es decir, pruebas en tiempo real y en video. En primer lugar, se van a presentar los datos del modelo clásico implementado, PCA+SVM, y posteriormente el modelo moderno, ResNet. Las pruebas se han realizado con el dataset de LFW seleccionando las clases que tuvieran más de 50 imágenes, además se le han añadido dos clases con nombres Antonio\_Manuel\_Solis\_Miranda y Alexandra\_Solis\_Miranda. Cada clase representa a un atleta que ha registrado su rostro, aunque estos dos últimos son los que más se acercan a lo que sería las imágenes obtenidas en el proceso de captación de los atletas.

Para PCA+SVM el resultado obtenido durante la fase de entrenamiento nos arroja unos datos bastante esperanzadores, donde se puede apreciar que la precisión es del 80%, con un rango de precisión para cada clase del 56% al 100%. Se ha realizado una prueba con imágenes del dataset que no se habían utilizado para el entrenamiento, dando valores positivos.



	precision	recall	f1-score	support
Ariel Sharon	0.68	0.68	0.68	19
Colin Powell	0.80	0.88	0.84	59
Donald Rumsfeld	0.75	0.80	0.77	30
George W Bush	0.91	0.88	0.90	133
Gerhard Schroeder	0.67	0.81	0.73	27
Hugo Chavez	0.93	0.78	0.85	18
Jacques Chirac	0.57	0.62	0.59	13
Jean Chretien	0.92	0.79	0.85	14
John Ashcroft	0.56	0.69	0.62	13
Junichiro Koizumi	0.83	0.67	0.74	15
Serena Williams	1.00	0.62	0.76	13
Tony Blair	0.81	0.81	0.81	36
Alexandra_Solis_Miranda	0.67	1.00	0.80	22
Antonio_Manuel_Solis_Miranda	0.00	0.00	0.00	11
accuracy			0.80	423
macro avg	0.72	0.72	0.71	423
weighted avg	0.79	0.80	0.79	423













predicted: Bush true: Bush confidence: 58.34%	predicted: Powell true: Powell confidence: 43.27%	predicted: Bush true: Bush confidence: 75.46%	predicted: Blair true: Blair confidence: 91.53%
			
predicted: Powell true: Powell confidence: 80.84%	predicted: Bush true: Bush confidence: 63.29%	predicted: Sharon true: Sharon confidence: 51.22%	predicted: Blair true: Blair confidence: 79.54%
			
predicted: Ashcroft true: Schroeder confidence: 37.21%	predicted: Powell true: Powell confidence: 56.66%	predicted: Sharon true: Sharon confidence: 82.30%	predicted: Bush true: Bush confidence: 73.64%
			

Ilustración 47. Estadísticas de entrenamiento y evaluación.

Parece que el modelo proporciona predicciones medianamente aceptables para los rostros de LFW, pero existe un gran problema que parece pasar desapercibido y que puede estar falseando todas las estadísticas. Se trata de la columna support, representa cuantas veces aparece cada clase en los datos de entrenamiento y evaluación. Esto hace por tanto que no se esté seleccionando todas las imágenes que se debería, lo que hace que tenga un soporte desequilibrado. Esto puede estar motivado por la diferencia de numero de imágenes entre las clases, lo que hace que el modelo esté aprendiendo mejor las clases con más imágenes y peor las que menos tienen. Se puede apreciar como “George W Bush” posee un soporte completamente desproporcionado con respecto al resto de las clases, un valor correcto de support vendría dado por un equilibrio entre las clases.

Esta falta de soporte hace que en casos en los que se presenta una imagen no mostrada anteriormente, tienda a identificarla erróneamente, es decir, el modelo generaliza mal o muy mal. Se puede apreciar en el caso del reconocimiento en tiempo real y video, donde los datos que se presentan son realmente desesperanzadores para este modelo. Se ha podido apreciar una identificación errónea en todos los casos de los videos de prueba (4K\_01.MOV, 4K\_02.MOV, 4K\_03.MOV, 4K\_30\_01.MOV y 4K\_30\_02.MOV) y un pequeño mejor desempeño en tiempo real con condiciones de luz optimas y posiciones ideales, llegando a ser capaz de reconocer un rostro real con el número de soporte mínimo. También se ha probado la identificación simultánea simulando un rostro con iluminación ideal (Hugo Chávez) y otro con peor iluminación (Antonio Manuel Solís Miranda), siendo capaz de reconocer el rostro ideal y fallar en el peor iluminado.



Ilustración 48. Reconocimiento de rostro mediante PCA+SVM.

Estos problemas se podrían intentar aminorar aplicando muestreo estratificado o reequilibrando el número de imágenes por clase, ninguna se ha aplicado pues se ha considerado que se querían comparar ambos modelos en las mismas condiciones y no se ha visto la necesidad de introducir técnicas más complejas, además de que los resultados de por si no son buenos.

Pasando con el siguiente modelo, ResNet, se puede apreciar unos resultados completamente diferentes. Mientras que PCA+SVM se veía un decremento considerable de la precisión al aplicar técnicas de Data Augmentation debido a que modificaban las proporciones de los rostros, las redes neuronales ResNet se ven ampliamente beneficiadas de esta técnica. El proceso de entrenamiento de una red neuronal es bastante superior al de un sistema SVM, pero por consiguiente el modelo es mucho mejor en cuanto a generalización. Con el entrenamiento buscamos que el modelo vaya aprendiendo con respecto pasan las etapas/épocas, comenzando con un nivel de precisión bajo y un nivel de pérdida alto, para finalizar con un nivel de precisión alto (cuanto más cercano a 1 mejor) y un nivel de pérdida menor (cuanto más cercano a 0 mejor), podemos ver como en el caso de una ResNet-50 el modelo es capaz de llegar a valores increíbles.

```

51/51 [.....] - 11s 272ms/step - loss: 2.3175 - accuracy: 0.3700 - val_loss: 1.4479 - val_accuracy: 0.5505 - lr: 1.0000e-04
Epoch 2/50
51/51 [.....] - 11s 253ms/step - loss: 1.5135 - accuracy: 0.5548 - val_loss: 1.1744 - val_accuracy: 0.6618 - lr: 1.0000e-04
Epoch 3/50
51/51 [.....] - 11s 248ms/step - loss: 1.3213 - accuracy: 0.6102 - val_loss: 0.9211 - val_accuracy: 0.7398 - lr: 1.0000e-04
Epoch 4/50
51/51 [.....] - 11s 244ms/step - loss: 1.0768 - accuracy: 0.6611 - val_loss: 0.8148 - val_accuracy: 0.7585 - lr: 1.0000e-04
Epoch 5/50
51/51 [.....] - 11s 267ms/step - loss: 1.0652 - accuracy: 0.6958 - val_loss: 0.7295 - val_accuracy: 0.7683 - lr: 1.0000e-04
Epoch 6/50
51/51 [.....] - 11s 238ms/step - loss: 0.9602 - accuracy: 0.7086 - val_loss: 0.6372 - val_accuracy: 0.8098 - lr: 1.0000e-04

51/51 [.....] - 11s 248ms/step - loss: 0.3013 - accuracy: 0.9122 - val_loss: 0.2054 - val_accuracy: 0.9512 - lr: 1.5800e-05
Epoch 47/50
51/51 [.....] - 11s 255ms/step - loss: 0.2630 - accuracy: 0.9203 - val_loss: 0.2045 - val_accuracy: 0.9443 - lr: 1.5800e-05
Epoch 48/50
51/51 [.....] - 11s 232ms/step - loss: 0.2674 - accuracy: 0.9134 - val_loss: 0.2067 - val_accuracy: 0.9439 - lr: 1.5800e-05
Epoch 49/50
51/51 [.....] - 11s 239ms/step - loss: 0.2797 - accuracy: 0.9128 - val_loss: 0.2069 - val_accuracy: 0.9439 - lr: 1.5800e-05
Epoch 50/50
51/51 [.....] - 11s 237ms/step - loss: 0.2542 - accuracy: 0.9203 - val_loss: 0.2079 - val_accuracy: 0.9439 - lr: 1.5800e-05

```

Ilustración 49. Etapas/épocas de entrenamiento del modelo.

Se puede apreciar que el entrenamiento acaba con un valor del 94% de precisión en el conjunto de validación y de 93% aproximadamente en el conjunto de entrenamiento. Por otra parte, el nivel de pérdida, que no viene representado en porcentaje, es bastante bajo al final, pues comienza con valores de 2.3 y 1.46, y termina con 0.25 y 2.8. Que los valores de pérdida y precisión de los conjuntos de entrenamiento y validación sean tan similares, denota una buena generalización sin sobreajuste. Además “lr” representa la tasa de aprendizaje, que no es más que el ajuste que se hace a los pesos del modelo que se realiza en cada época según la pérdida, es común que vaya decrementando con respecto al paso de las fases, lo que significa que el modelo está respondiendo correctamente al entrenamiento.

Los resultados obtenidos, son muy correctos pues en los videos se ha logrado obtener con precisión del 100% en algunos frames cercanos a la cámara donde el rostro es más apreciable. Se ha configurado un umbral alto para que el sistema no identifique rostros erróneamente, configurándolo al 90% de confianza, así hemos asegurado que en todos los videos se identifica a correctamente al atleta. También el tiempo real se obtienen resultados sin fallos, incluso con imágenes de LFW.



*Ilustración 50. Resultados de la prueba con el modelo ResNet.*

Aunque no se trata de un modelo ideal, debido a que con un umbral tanto alto el modelo reconoce los rostros el 50% de las ocasiones, pero lo realiza con una gran precisión. Se han testado además los modelos de ResNet-101 y ResNet-152, dando resultados similares, esto puede deberse a la poca variedad de imágenes y clases lo que hace que ResNet-50 sea mas que suficiente e incluso mejor que sus hermanos, pues no cae en el sobreajuste. Aunque se espera que en un uso real, se trate de una mejor opcion usar 101 o 152, al tener mas capas (su numero respectivamente) y pueda aprender de mejor forma los detalles de los rostros.

Por tanto, sea comprobado que los modelos más idóneos para el reconocimiento en pruebas deportivas populares es el de las Redes Neuronales Convolucionales (CNN) que representan a las técnicas modernas de reconocimiento de imágenes. Su nivel de aprendizaje es superior a cualquier método clásico, aunque solo se ha testado oficialmente con PCA+SVM. Aunque las CNN necesiten mas datos que un método clásico estos desempeñan entorno un 40% mejor, lo que supone una gran mejora. Cabe destacar que la configuración de una red neuronal es un proceso complejo y requiere de grandes conocimientos sobre estas, a diferencia de modelos clásicos. Es por esto que los sistemas actuales en su mayoría aplican sistemas basado en Redes Neuronales.

## 5. Inclusión real del sistema de cronometraje

Lo expuesto durante todo el trabajo trata en su mayoría sobre el reconocimiento facial y que algoritmos y técnicas serían las idóneas, sin hablar del fin de este sistema, que es el sistema de cronometraje. En este punto se explicará de forma progresiva como sería

la implementación real y funcional que debería de llevarse a cabo para lograr un resultado satisfactorio para la prueba. Aclarar que estos pasos al igual que el sistema, no se ha testado en una prueba real, si no que representan los pasos ideados durante el desarrollo.

## 5.1. Organización de un evento real

Un evento deportivo, en este caso de pruebas populares, consta de un al menos un día si el evento es de pequeña magnitud o de tres o cuatro días si el número de participantes super el millar, normalmente se celebran los fines de semana, en los que el viernes y sábado (jueves si la prueba es de muchos participantes +15000) están destinados al reparto de dorsales y bolsas del corredor, donde se celebra la exposición del corredor. Por ultimo y el día de la carrera donde se culmina el evento.

El equipo de cronometraje tiene que estar en todas ellas pues su función es la columna principal del evento, pero la función de la organización y del equipo de cronometraje difieren en distintos puntos y funciones:

- Día de preparación previo al evento: Hay que organizar el material que se va a necesitar y testear que los sistemas de cronometraje RFID funcionen correctamente. Además, también debe de asignarse cada uno de los dorsales con su chip a su respectivo atleta.
- Días de la Exposición del corredor: La función del equipo de cronometraje en estos 2 o 3 días es muy importante pues si surge algún problema con los dorsales, es necesario algún cambio de nombre o información, existen problemas con los ordenadores que escanean los dorsales para entregarlos, o en general, hay algún problema que influya al cronometraje del atleta es responsabilidad del equipo de cronometraje. Estos días se centran en el reparto de los dorsales con las etiquetas RFID o sistemas similares.



*Ilustración 51. Entrega de dorsales de la Maratón de Sevilla.*

- Día del evento: Este es el día más importante, pues es donde la función principal de este equipo brilla. En este día se comprueban los sistemas, se carga la base de datos de los atletas, se configuran los distintos puntos de cronometraje del recorrido y se da comienzo a la prueba. Cuando llega el ultimo atleta se da por

concluida la prueba y se para el sistema RFID de cronometraje. Posteriormente se suben las clasificaciones oficiales a la aplicación o página oficial del evento.



*Ilustración 52. Salida de la Maratón de Sevilla.*

- Días posteriores al evento: Estos últimos días no requieren de un gran esfuerzo, pues se procesan los datos, se extraen estadísticas que puedan arrojar algún tipo de información importante sobre los equipos de cronometraje o la prueba, y se atienden cualquier tipo de reclamación en cuanto a tiempos oficiales.

En conclusión, el equipo de cronometraje desempeña un papel fundamental en la organización y ejecución de un evento deportivo, especialmente en pruebas populares de atletismo. Su trabajo abarca desde la preparación y verificación de los sistemas de cronometraje antes del evento, la gestión de cualquier incidencia durante la entrega de dorsales, hasta la configuración y supervisión de los sistemas el día de la carrera. Su labor asegura la precisión y la integridad de los tiempos registrados, proporcionando resultados fiables y justos para todos los participantes. Posteriormente, el equipo procesa los datos y gestiona cualquier reclamación, consolidando la credibilidad y el éxito del evento.

## 5.2. Organización de un evento real que use reconocimiento facial

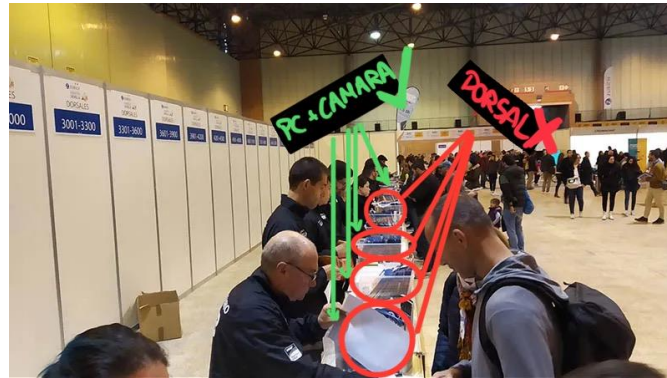
Las funciones del equipo siguen siendo en esencia las mismas, únicamente cambia el sistema que se utiliza.

En primer lugar, los días previos al evento se deberían de testear y ajustar los modelos de reconocimiento facial a los datos que se hayan obtenidos en otras pruebas anteriores para aumentar la tasa de éxito, una opción muy interesante para aumentar la probabilidad de éxito es utilizar imágenes de atletas que ya hayan corrido en otros eventos con este sistema, ayudando así al modelo a tener más datos y por tanto aumentado la confianza individual de cada atleta. Se puede hacer un pre entreno con estos datos y así reducir el tiempo de entrenamiento posterior.

El día de la exposición del corredor, se prepara la zona donde se van a escanear cada rostro de cada atleta. Esta zona debe de tener varios ordenadores o sistemas de captación de imágenes, cuantos más mejor será el flujo de atletas, en pruebas de 12000



inscritos sería necesario al menos de 45 a 50 equipos. Está zona además deberá de tener una buena iluminación que pueda captar imágenes optimas del rostro de los atletas. En paralelo el equipo de cronometraje podrá estar visualizando que no haya ningún problema con las imágenes que estén llegando al sistema de almacenamiento central, pues todos los sistemas de captación de imágenes estarán conectados a un mismo sistema de almacenamiento central.



*Ilustración 53. Representación de la toma de imágenes del rostro del atleta.*

Cada día de exposición del corredor, una vez finalizado esta, se deberá de realizar un entrenamiento del modelo con los datos recopilados ese mismo día, para así agilizar los tiempos de entrenamiento y no realizarlo todo el ultimo día con cantidades enormes de datos.

El día anterior al día del evento, se deberá de colocar las cámaras en el arco de salida y meta. El ángulo que tienen que tener las cámaras es de 30 grados con la horizontal apuntado al suelo, esto garantiza un reconocimiento optimo, más de 30 grados puede generar frames en los que no se detecten rostros, y menos supondría que se tiene que detectar los rostros desde más lejos. Es recomendable el uso de focos complementarios de alta potencia que aseguren un nivel de luz adecuado, haciendo que las condiciones meteorológicas no sea un problema para el sistema.



*Ilustración 54. Representación de la disposición de las cámaras en el arco de salida/meta.*

En cuanto al sistema de cronometraje, se ha planteado un sistema básico que marque como tiempo de comienzo y llegada, el último instante en el que aparece el rostro del atleta en cámara, esto es muy simple de implementar, pero carece de mucha exactitud

pues debido a la localización de la cámara no puede ver el momento exacto en el que entra en meta.

Una solución a este problema sería alejar la cámara del arco de salida/meta, aunque habría que implementar un sistema más complejo, en el que se calcule el centro del rostro y mediante cálculos geométricos se trace una línea en la que si el centro del rostro atraviesa esa línea se registraría el tiempo. Esta opción sería la ideal, pero debido a su complejidad innecesaria para este trabajo, sea obviado la idea.



*Ilustración 55. Representación de la implementación ideal.*

El tiempo se registraría en una base de datos NoSQL, que facilita el acceso y registro de datos de los atletas que salen y entran en meta en tiempo real, gracias a su gran velocidad. Además, la facilidad de realizar consultas complejas nos proporcionaría un sistema fácil y rápido junto con Python, para la extracción de clasificaciones oficiales.

Para asemejar este sistema a los sistemas RFID actuales, también se podrían instalar cámaras en puntos de control, como en una maratón los 10km o 21km, para que se puedan tener información del ritmo y localización del atleta. Solo difiere en la forma en la que se capta la información, que en vez de ser con RFID es con un reconocimiento de rostro igual al de salida y meta, pero sin necesidad de una localización tan específica como necesitan estas. La información se envía de forma inalámbrica mediante 5G desde la unidad de procesamiento local del punto de control.

Toda la información generada del evento se guarda en discos duros, para su posterior utilización para mejoras del sistema y sacar estadísticas sobre el rendimiento. Además, como hemos comentado antes, las imágenes de los atletas se pueden utilizar para eventos diferentes en los que participen, aumentando así la precisión del sistema al tener más muestras.

## 6. Problemas y limitaciones del sistema

Como se ha podido observar los resultados de las CNN en concreto de las ResNet, son considerablemente superior a los métodos clásicos de reconocimiento de imágenes

como el explicado en este trabajo, es decir, PCA+SVM. Esto era de esperar pues a nivel comercial la mayoría de implementaciones de reconocimiento facial implementan un método basado en CNN.

Por tanto, el sistema a implementar constaría de una ResNet en la fase de reconocimiento de rostros. Asegurando al menos un 95% de precisión a la hora de reconocer los rostros.

Pero hay algunos problemas y limitaciones en los que el sistema puede fallar o no rendir en condiciones óptimas, alguno de los motivos son los siguientes:

- Falta de variedad en las imágenes recopiladas: Este es principal problema al que se tiene que enfrentar el sistema propuesto, carece de variedad en los datos. Aunque el módulo de Data Augmentation puede proporcionar una variación simulada, no es suficiente para alcanzar los valores ideales.
- Rendimiento medio: Aunque el modelo no es extremadamente lento, sí que sufre de lentitud media, lo que hace que baje los FPS de la captura en vivo, a la hora de procesar grandes cantidades de datos simultáneamente. Esto se ha apreciado en un equipo de características media, pudiendo mejorar para equipos más potentes.
- Tiempo de entrenamiento: En las pruebas no se ha apreciado una gran cantidad de tiempo, pero es debido a que la cantidad de imágenes de prueba es menor a la que sería con una prueba de más de 100 participantes. Esto podría incurrir en mayores tiempos de entrenamiento. Al igual que el punto anterior, con equipos más potentes se podría solucionar. Otra solución sería realizar los entrenamientos por la noche o en momentos clave, y no realizar uno con una gran cantidad de información horas antes del día de la prueba.
- Coste elevado de los equipos: Como se ha podido deducir, para asegurar el correcto funcionamiento del sistema de reconocimiento facial, es necesario tener equipos de calidad, sobre todo en los equipos de procesamiento en los que se van a cargar y entrenar los modelos. También las cámaras instaladas el día de la carrera. Además, poseer un número de cámaras tan elevado para la captación de imágenes del rostro requiere de una inversión importante, pero rentable a largo plazo.
- Modificaciones del rostro: Es posible que los atletas modifiquen su rostro el día de la prueba, con el uso de gafas o algún tipo de adorno que dificulte y tape las facciones de este. Este es sin duda, uno de los mayores problemas a los que se enfrenta este sistema.

## 7. Análisis a futuro y mejoras a aplicar

Este sistema supone una revolución en el mundo del deporte, abriendo un gran abanico de posibilidades a un deporte que parece estancado y sin innovaciones desde hace



décadas. Aunque no es un sistema perfecto con el paso del tiempo, el descubrimiento de nuevas técnicas y modelos que mejoren los resultados, se podrá implementar de forma efectiva un modelo de reconocimiento facial en pruebas deportivas con todas las garantías.

Algunas posibles implementaciones a realizar en un futuro sería la prueba con otros modelos de reconocimiento facial basados en CNN como FaceNet, VGG-Face o DeepFace, que por la información que recogen algunos usuarios pueden llegar a niveles de precisión del 99%, aunque habría que testarlo con el mismo dataset de prueba que sea utilizado en este proyecto para ver si es capaz de alcanzar la precisión que promete.

Se ha rechazado la idea de implementar otros métodos como pueden ser LBPH (de OpenCV) o K-Means debido a su bajo nivel de acierto y rendimiento en tiempo real, respectivamente.

En cuanto a mejoras, se puede optimizar líneas de código, como líneas duplicadas u optimización de variables, esperando que aumente el rendimiento general buscando una reducción de tiempo de entrenamiento o velocidad de procesamiento en tiempo real.

## Conclusiones

Este trabajo de fin de grado ha explorado de manera teórica y práctica el desarrollo de un sistema de cronometraje para pruebas deportivas basado en el reconocimiento facial, abordando cada etapa del proceso desde la adquisición de imágenes hasta el reconocimiento final. Se han empleado técnicas avanzadas de visión por computadora e inteligencia artificial, usando herramientas como OpenCV, Dlib y TensorFlow para construir un sistema robusto y eficiente, desde cero y completamente funcional.

Se ha logrado desarrollar un sistema que captura, preprocesa y reconoce rostros de forma eficiente, proporcionando una alternativa viable, aunque no igual de precisa que los métodos tradicionales de cronometraje como el RFID.

También se han evaluado diversas técnicas de forma exhaustiva, y se determinó que el método HOG de Dlib es superior a Viola-Jones para la detección de rostros en términos de manejo de múltiples rostros y ángulos. Además, se demostró que ResNet supera ampliamente a PCA+SVM en precisión de reconocimiento facial, gracias a su capacidad para gestionar variaciones en los datos la que lo convierte en un modelo muy capaz de generalizar.

Además, se han implementado técnicas de reducción de datos como K-Means y PCA, junto con el uso de Data Augmentation, que permitió optimizar tanto el almacenamiento como el rendimiento del sistema.

Y por último, una vez explicado todos los puntos y porque es mejor el sistema de HOG+ResNet, se diseñó una arquitectura práctica para la implementación del sistema en eventos deportivos, asegurando la captura precisa de datos y el entrenamiento continuo de la red neuronal para mejorar su desempeño.

El trabajo ha contribuido a revitalizar el campo del cronometraje deportivo, proponiendo una solución innovadora y detallada que puede ser aplicada en diversos eventos de múltiples dimensiones. Además, ofrece una base sólida para futuras investigaciones y mejoras, como la integración de técnicas más avanzadas de inteligencia artificial y optimización de código.

El uso de un sistema de reconocimiento facial para el cronometraje no solo moderniza y a porta otro punto de vista a esta área, tan desconocida a los ojos de las personas, sino que también abre nuevas posibilidades en la gestión y análisis de eventos deportivos. La implementación de este sistema podría reducir significativamente los errores humanos y proporcionar datos en tiempo real, beneficiando a organizadores, atletas y espectadores.

En conclusión, este trabajo demuestra el potencial del reconocimiento facial en aplicaciones deportivas, estableciendo una información de soporte para futuras innovaciones en el campo del cronometraje. La combinación de técnicas de aprendizaje profundo y visión por computadora ofrece una herramienta poderosa y versátil, capaz de revolucionar la manera en que se gestionan y cronometran los eventos deportivos.

## Bibliografía:

colaboradores de Wikipedia. (2023a, julio 16). Cronógrafo. Wikipedia, la Enciclopedia Libre.

<https://es.wikipedia.org/wiki/Cron%C3%B3grafo#:~:text=El%20t%C3%A9rmino%20E2%80%9Ccron%C3%B3grafo%E2%80%9D%20fue%20utilizado,un%20peque%C3%B1o%20dep%C3%B3sito%20de%20tinta.>

colaboradores de Wikipedia. (2023b, agosto 15). Deporte en el Antiguo Egipto. Wikipedia, la Enciclopedia Libre.

[https://es.wikipedia.org/wiki/Deporte\\_en\\_el\\_Antiguo\\_Egipto](https://es.wikipedia.org/wiki/Deporte_en_el_Antiguo_Egipto)

colaboradores de Wikipedia. (2024a, marzo 5). Christiaan Huygens. Wikipedia, la Enciclopedia Libre. [https://es.wikipedia.org/wiki/Christiaan\\_Huygens](https://es.wikipedia.org/wiki/Christiaan_Huygens)

colaboradores de Wikipedia. (2024b, mayo 7). Juegos Olímpicos de Atenas 1896. Wikipedia, la Enciclopedia Libre.

[https://es.wikipedia.org/wiki/Juegos\\_Ol%C3%ADmpicos\\_de\\_Atenas\\_1896](https://es.wikipedia.org/wiki/Juegos_Ol%C3%ADmpicos_de_Atenas_1896)

colaboradores de Wikipedia. (2024c, junio 12). Antiguo Egipto. Wikipedia, la Enciclopedia Libre. [https://es.wikipedia.org/wiki/Antiguo\\_Egipto](https://es.wikipedia.org/wiki/Antiguo_Egipto)

De Ancos Presidente de Ochobre, A. (2023, 1 junio). El deporte en los palacios egipcios - fusionasturias.com. fusionasturias.com. <https://fusionasturias.com/deportes/pasion-y-deporte/el-deporte-en-los-palacios-egipcios.htm>

Eulalia, S. (2021, 11 agosto). El segundero muerto. Moda de Lujo - Blog Santa Eulalia. <https://blog.santa-eulalia.com/lifestyle/alta-relojeria-el-segundero-muerto/>

García, F. (2020, 26 marzo). Los juegos en Grecia: Olímpicos, Ístmicos, Píticos y Nemeos. historia.nationalgeographic.com.es. [https://historia.nationalgeographic.com.es/a/juegos-grecia-olimpicos-istmicos-piticos-y-nemeos\\_9163#](https://historia.nationalgeographic.com.es/a/juegos-grecia-olimpicos-istmicos-piticos-y-nemeos_9163#)

La Universidad de Osaka, P. P. D. P. E. (s. f.). Cronometrando los Juegos Olímpicos. <https://www.horalatina.com/archivos/8723-cronometrando-los-juegos-olimpicos.html>

Martin, P. (s. f.). Breve historia del cronometraje. <https://historiatletismo.blogspot.com/2010/12/breve-historia-del-cronometraje.html>

¿Quién inventó el cronómetro? (s. f.). www.my-alarm-clock.com. <https://www.my-alarm-clock.com/es/articulos/quien-invento-el-cronometro>

Serra, A. (2018, 5 julio). Historia del cronometraje de carreras - timingsense. Timingsense. <https://timingsense.com/blog/historia-del-cronometraje-de-carreras/>

Wikipedia contributors. (2023, 2 septiembre). George Graham (clockmaker). Wikipedia. [https://en.wikipedia.org/wiki/George\\_Graham\\_\(clockmaker\)](https://en.wikipedia.org/wiki/George_Graham_(clockmaker))

Wikipedia contributors. (2024a, marzo 15). Richard Towneley. Wikipedia. [https://en.wikipedia.org/wiki/Richard\\_Towneley](https://en.wikipedia.org/wiki/Richard_Towneley)

Wikipedia contributors. (2024b, abril 21). Anchor escapement. Wikipedia. [https://en.wikipedia.org/wiki/Anchor\\_escapement](https://en.wikipedia.org/wiki/Anchor_escapement)

colaboradores de Wikipedia. (2023, 7 agosto). Transductor. Wikipedia, la Enciclopedia Libre. <https://es.wikipedia.org/wiki/Transductor>

colaboradores de Wikipedia. (2024, 23 enero). RFID. Wikipedia, la Enciclopedia Libre. <https://es.wikipedia.org/wiki/RFID>

RACE RESULT - Cronometraje y Valoración de eventos deportivos. (s. f.). <https://www.raceresult.com/es-es/systems/index.php>

Serra, A. (2020, 29 enero). Sistema de cronometraje por chip: ¿campo cercano o lejano? Timingsense. <https://timingsense.com/blog/cronometraje-deportivo-con-chip/>

Serra, A. (2021, 31 agosto). Sistemas de cronometraje activos y pasivos - timingsense. Timingsense. <https://timingsense.com/blog/sistemas-cronometraje-activos-y-pasivos/>

What's the difference between an active and passive timing system? (s. f.). <https://blog.mylaps.com/active-or-passive-timing-system>

Características que se deben tener en cuenta al momento de invertir en un Sistema de Cámaras de Seguridad. (2022, April 18). Cloud Based and Central Management. <https://www.iptechview.com/es/caracteristicas-que-se-deben-tener-en-cuenta-al-momento-de-invertir-en-un-sistema-de-camaras-de>

Directrices de la cámara para Face Recognition. (n.d.). <https://help.avigilon.com/design4analytics/es/face-recognition/camera-placement.htm>

Editaclia. (2017, March 4). Repasamos un poco de historia de la creación de la cámara cinematográfica así como sus primeros pasos y su evolución en el tiempo. Tu Editor De Videos. <https://tueditordevideos.com/historia-camara-video/>

Krings, E. (2023, September 25). Comprendiendo los Valores FPS Comunes: Guía avanzada de frecuencias de imagen de vídeo [2023 Update]. Dacast. <https://www.dacast.com/es/blog-es/frame-rate-fps/#:~:text=La%20velocidad%20de%20fotogramas%20se,que%20se%20muestran%20esas%20im%C3%A1genes.>

Thomsen, M. R. (2018, October 19). Los orígenes de la fotografía. La Vanguardia. <https://www.lavanguardia.com/vida/junior-report/20180411/442459480247/historia-fotografia-camara-kodak-polaroid-digital-daguerrotipo.html#:~:text=En%201839%20Louis%20Daguerre%20patent%C3%B3,c%C3%A1mara%20fotogr%C3%A1fica%20de%20la%20historia.&text=En%20paralelo%2C%20el%20cient%C3%ADfico%20brit%C3%A1nico,papel%20sensible%20a%20la%20luz.>

Mahalle, A. G., & Shah, A. M. (2017). FPGA implementation of gradient based edge detection algorithms. Int J Innov Res Comput Commun Eng, 5(5), 10610-10616.

8.2. Matriz de convolución. (n.d.). <https://docs.gimp.org/2.6/es/plugin-convmatrix.html>

colaboradores de Wikipedia. (2023, April 11). Transformación afín. Wikipedia, La Enciclopedia Libre. [https://es.wikipedia.org/wiki/Transformaci%C3%B3n\\_af%C3%ADn](https://es.wikipedia.org/wiki/Transformaci%C3%B3n_af%C3%ADn)

colaboradores de Wikipedia. (2024a, May 10). Procesamiento digital de imágenes. Wikipedia, La Enciclopedia Libre. [https://es.wikipedia.org/wiki/Procesamiento\\_digital\\_de\\_im%C3%A1genes](https://es.wikipedia.org/wiki/Procesamiento_digital_de_im%C3%A1genes)

colaboradores de Wikipedia. (2024b, May 31). Sistema de reconocimiento facial. Wikipedia, La Enciclopedia Libre. [https://es.wikipedia.org/wiki/Sistema\\_de\\_reconocimiento\\_facial](https://es.wikipedia.org/wiki/Sistema_de_reconocimiento_facial)

García Mateos, G. & Dept. de Informática y Sistemas, Universidad de Murcia. (n.d.). PROCESAMIENTO DE IMÁGENES [Slide show]. <https://dis.um.es/~ginesgm/files/doc/pi/tema2.pdf>

Jesús. (2021, July 6). Cómo Crear un Reconocedor Facial con Eigenfaces con scikit-learn y OpenCV. DataSmarts. <https://www.datasmartr.net/como-crear-un-reconocedor-facial-con-eigenfaces-con-scikit-learn-y-opencv/>

marketingbcnvision. (2024, February 9). Algoritmos de reconocimiento facial, casos de uso y polémica. <https://bcnvision.es/blog-vision-artificial/algoritmos-de-reconocimiento-facial-casos-de-uso-y-polemica/>

Mejoramiento de las imágenes en el dominio espacial. (n.d.). [Slide show]. <http://dsp1.materia.unsl.edu.ar/Mejoramiento%20Espacial.pdf>

Solis Miranda, A. M. (2024, January). Fuente de imagenes. ImagenesPSM. <https://imagenespsm.streamlit.app/>

del Prado, N. G., Gonzalez-Castro, V., Alegre, E., & Fernandez, E. F. (s/f). COMPARACION DE METODOS DE DETECCION DE ROSTROS EN IMAGENES DIGITALES. Udc.es. Recuperado el 13 de junio de 2024, de [https://ruc.udc.es/dspace/bitstream/2183/25830/3/2017\\_Garcia\\_del\\_Prado\\_Natalia\\_Comparacion\\_metodos\\_deteccion\\_rostros\\_imagenes\\_digitales.pdf](https://ruc.udc.es/dspace/bitstream/2183/25830/3/2017_Garcia_del_Prado_Natalia_Comparacion_metodos_deteccion_rostros_imagenes_digitales.pdf)

colaboradores de Wikipedia. (2023, November 12). Geometría del taxista. Wikipedia, La Enciclopedia Libre. [https://es.wikipedia.org/wiki/Geometr%C3%ADa\\_del\\_taxista](https://es.wikipedia.org/wiki/Geometr%C3%ADa_del_taxista)

Dshahid. (2021, December 8). Convolutional Neural Network - towards data science. Medium. <https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529>

Na, & Na. (2019, February 26). Breve Historia de las Redes Neuronales Artificiales | Aprende Machine Learning. Aprende Machine Learning.

<https://www.aprendemachinelearning.com/breve-historia-de-las-redes-neuronales-artificiales/>

Papers with Code - Max Pooling Explained. (n.d.). <https://paperswithcode.com/method/max-pooling>

¿Qué es el algoritmo de k vecinos más cercanos? | IBM. (n.d.). <https://www.ibm.com/es-es/topics/knn>

¿Qué es el aprendizaje supervisado? | IBM. (n.d.). <https://www.ibm.com/es-es/topics/supervised-learning>

¿Qué son las redes neuronales convolucionales? | 3 cosas que debe saber. (n.d.). MATLAB & Simulink. <https://es.mathworks.com/discovery/convolutional-neural-network.html>

Tyagi, M. (2022, January 6). Viola Jones Algorithm and haar cascade classifier - towards data science. Medium. <https://towardsdatascience.com/viola-jones-algorithm-and-haar-cascade-classifier-ee3bfb19f7d8>

Vila, M. a. S. (n.d.). [Azure Machine Learning] La dicotomía Varianza-sesgo (Bias-Variance). Epicalsoft — Superheroic Software Development Blog. [https://epicalsoft.blogspot.com/2019/02/azure-machine-learning-la-dicotomia.html#:~:text=El%20algoritmo%20k%2Dnn%20\(k,aumenta%20el%20sesgo%20del%20modelo](https://epicalsoft.blogspot.com/2019/02/azure-machine-learning-la-dicotomia.html#:~:text=El%20algoritmo%20k%2Dnn%20(k,aumenta%20el%20sesgo%20del%20modelo)

Wikipedia contributors. (2024a, April 29). AdaBoost. Wikipedia. <https://en.wikipedia.org/wiki/AdaBoost>

Wikipedia contributors. (2024b, April 29). Viola–Jones object detection framework. Wikipedia. [https://en.wikipedia.org/wiki/Viola%E2%80%93Jones\\_object\\_detection\\_framework](https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework)

Clasificación de imágenes usando redes neuronales convolucionales en Python, C. (s/f). Trabajo Fin de Grado Grado en Ingeniería de las Tecnologías de Telecomunicación. Idus.us.es. Recuperado el 13 de junio de 2024, de <https://idus.us.es/bitstream/handle/11441/89506/TFG-2402-ARTOLA.pdf?sequence=1>

Tamaulipas, I. C. (s/f). Ponente: D r. Wilfrido Gómez Flores. Cinvestav.mx. Recuperado el 13 de junio de 2024, de [https://www.tamps.cinvestav.mx/~wgomez/toptamps/presentacion2\\_2016.pdf](https://www.tamps.cinvestav.mx/~wgomez/toptamps/presentacion2_2016.pdf)

Jesús. (2020, March 3). Cómo Extraer Features de una Imagen con HOG en Scikit-Image. DataSmarts Español. <https://datasmarts.net/es/como-extraer-features-de-una-imagen-con-hog-en-scikit-image/>

HOG Histograma de Gradientes Orientados. (2017, December 5). WordPress.com. <https://carlosjuliopardoblog.wordpress.com/2017/12/04/hog-histograma-de-gradientes-orientados/>

Mittal, K. (2024, March 12). A gentle introduction into the histogram of oriented gradients. Medium. <https://medium.com/analytics-vidhya/a-gentle-introduction-into-the-histogram-of-oriented-gradients-fdee9ed8f2aa>

What does it mean an histogram vector normalization with L1/L2 norms? (n.d.). Cross Validated. <https://stats.stackexchange.com/questions/224723/what-does-it-mean-an-histogram-vector-normalization-with-l1-l2-norms>

Rodrigo, J. A. (s/f). Máquinas de Vector Soporte (Support Vector Machines, SVMs). Cienciadedatos.net. Recuperado el 13 de junio de 2024, de [https://cienciadedatos.net/documentos/34\\_maquinas\\_de\\_vector\\_soporte\\_support\\_vector\\_machines](https://cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines)

Navarro, S. (2024, April 16). ¿Qué son las SVM o Support Vector Machines? KeepCoding Bootcamps. <https://keepcoding.io/blog/que-son-las-svm-o-support-vector-machines/>

colaboradores de Wikipedia. (2024, March 5). Máquinas de vectores de soporte. Wikipedia, La Enciclopedia Libre. [https://es.wikipedia.org/wiki/M%C3%A1quinas\\_de\\_vectores\\_de\\_soporte](https://es.wikipedia.org/wiki/M%C3%A1quinas_de_vectores_de_soporte)

3.6.10.14. The eigenfaces example: chaining PCA and SVMs — Scipy lecture notes. (n.d.). [https://scipy-lectures.org/packages/scikit-learn/auto\\_examples/plot\\_eigenfaces.html](https://scipy-lectures.org/packages/scikit-learn/auto_examples/plot_eigenfaces.html)

GeeksforGeeks. (2021, November 30). ML Face Recognition using PCA Implementation. GeeksforGeeks. <https://www.geeksforgeeks.org/ml-face-recognition-using-pca-implementation/>

Labelled faces in the Wild (LFW): images of various people with. . . (n.d.). ResearchGate. [https://www.researchgate.net/figure/Labelled-faces-in-the-Wild-LFW-images-of-various-people-with-different-pose\\_fig6\\_334725817](https://www.researchgate.net/figure/Labelled-faces-in-the-Wild-LFW-images-of-various-people-with-different-pose_fig6_334725817)

Potrimba, P. (2024, April 15). What is ResNet-50? Roboflow Blog. <https://blog.roboflow.com/what-is-resnet-50/>

colaboradores de Wikipedia. (2024, January 28). Problema de desvanecimiento de gradiente. Wikipedia, La Enciclopedia Libre. [https://es.wikipedia.org/wiki/Problema\\_de\\_desvanecimiento\\_de\\_gradiente](https://es.wikipedia.org/wiki/Problema_de_desvanecimiento_de_gradiente)

Archivo:Logo US.png - SebaseNet. (n.d.). [https://www.uco.es/investigacion/proyectos/SEBASENet/index.php?title=Archivo:Logo\\_US.png](https://www.uco.es/investigacion/proyectos/SEBASENet/index.php?title=Archivo:Logo_US.png)

Pic2Go. (n.d.). <https://www.paulpoole.co.th/edm/may19/P2G.html>

Race Photos | Saskatchewan Marathon. (n.d.). <https://saskmarathon.ca/race-photos/>

(1) Marathon Foto (@MarathonFotoEPG) / x. (n.d.). X (Formerly Twitter). <https://x.com/marathonfotoepg>



Cómo los clubes utilizan los datos de Mediacoach, desarrollada por LaLiga, para imponerse al rival | Fútbol Global. (n.d.). Fútbol Global. <https://newsletter.laliga.es/futbol-global/como-los-clubes-utilizan-los-datos-de-mediacoach-desarrollada-por-laliga-para-imponerse-al-rival>

colaboradores de Wikipedia. (n.d.). Archivo:Checkerboard identity.svg - Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Archivo:Checkerboard\\_identity.svg](https://es.wikipedia.org/wiki/Archivo:Checkerboard_identity.svg)

colaboradores de Wikipedia. (n.d.). Archivo:Checkerboard reflection.svg - Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Archivo:Checkerboard\\_reflection.svg](https://es.wikipedia.org/wiki/Archivo:Checkerboard_reflection.svg)

colaboradores de Wikipedia. (n.d.). Archivo:Checkerboard scale.svg - Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Archivo:Checkerboard\\_scale.svg](https://es.wikipedia.org/wiki/Archivo:Checkerboard_scale.svg)

colaboradores de Wikipedia. (n.d.). Archivo:Checkerboard rotate.svg - Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Archivo:Checkerboard\\_rotate.svg](https://es.wikipedia.org/wiki/Archivo:Checkerboard_rotate.svg)

colaboradores de Wikipedia. (n.d.). Archivo:Checkerboard shear.svg - Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Archivo:Checkerboard\\_shear.svg](https://es.wikipedia.org/wiki/Archivo:Checkerboard_shear.svg)

Fig. 9: (a)Edge Output for Robert, Prewitt & Sobel Operators (b) Output. . . (n.d.). ResearchGate. [https://www.researchgate.net/figure/aEdge-Output-for-Robert-Prewitt-Sobel-Operators-b-Output-for-various-steps-of\\_fig7\\_337153747](https://www.researchgate.net/figure/aEdge-Output-for-Robert-Prewitt-Sobel-Operators-b-Output-for-various-steps-of_fig7_337153747)

Jplenio. (2024, March 5). Pistas Acero Rieles - Foto gratis en Pixabay. Pixabay. <https://pixabay.com/es/photos/pistas-acero-rieles-viaje-8613278/>

Figure 5. Scheme of haar features [18]. (n.d.). ResearchGate. [https://www.researchgate.net/figure/Scheme-of-haar-features-18\\_fig5\\_357946620](https://www.researchgate.net/figure/Scheme-of-haar-features-18_fig5_357946620)

Tyagi, M. (2022, January 6). Viola Jones Algorithm and haar cascade classifier - towards data science. Medium. <https://towardsdatascience.com/viola-jones-algorithm-and-haar-cascade-classifier-ee3bfb19f7d8>

Récord de participación extranjera en el Maratón de Sevilla 2024 - Muchodeporte. (n.d.). <https://www.muchodeporte.com/hoy-es-noticia/74194/record-de-participacion-extranjera-en-la-maraton-de-sevilla/>

EnriqueTictac. (2019, April 24). Arcos - portos sonorizaciones. Portos Sonorizaciones. <http://www.sonorizacionesportos.com/arcos/>

Ávila, S. A. (2024, February 16). Dónde ver en televisión y online el Zurich Maratón de Sevilla 2024. Diario ABC. <https://www.abc.es/deportes/ver-television-online-zurich-maraton-sevilla-2024-20240216143543-nts.html>

Fig. 2 A schematic of the AdaBoost process. Blue triangles and orange. . . (n.d.). ResearchGate. [https://www.researchgate.net/figure/A-schematic-of-the-AdaBoost-process-Blue-triangles-and-orange-squares-represent\\_fig1\\_351719735](https://www.researchgate.net/figure/A-schematic-of-the-AdaBoost-process-Blue-triangles-and-orange-squares-represent_fig1_351719735)

Figure 2. Illustration of cascade classifier. (n.d.). ResearchGate. [https://www.researchgate.net/figure/Illustration-of-cascade-classifier\\_fig2\\_371812964](https://www.researchgate.net/figure/Illustration-of-cascade-classifier_fig2_371812964)

Fig. 5. Results of applying HOG. (n.d.). ResearchGate. [https://www.researchgate.net/figure/Results-of-applying-HOG\\_fig5\\_343263011](https://www.researchgate.net/figure/Results-of-applying-HOG_fig5_343263011)

A. Requena y col.: “Nuevas Tecnologías y Contaminación de Atmósferas para PYMEs.” (n.d.). <https://www.um.es/LEQ/Atmosferas/Ch-VI-3/F63s4p3.htm>

Figura 2.5: Funciones de Activación más utilizadas para las redes. . . (n.d.). ResearchGate. [https://www.researchgate.net/figure/Figura-25-Funciones-de-Activacion-mas-utilizadas-para-las-redes-neuronales-artificiales\\_fig2\\_358008222](https://www.researchgate.net/figure/Figura-25-Funciones-de-Activacion-mas-utilizadas-para-las-redes-neuronales-artificiales_fig2_358008222)

Figure 4.3 -Exemple d’une opération de max-pooling. (n.d.). ResearchGate. [https://www.researchgate.net/figure/Exemple-dune-operation-de-max-pooling\\_fig4\\_337635267](https://www.researchgate.net/figure/Exemple-dune-operation-de-max-pooling_fig4_337635267)

Figure 3: Example of an image detector using convolutional neural network. (n.d.). ResearchGate. [https://www.researchgate.net/figure/Example-of-an-image-detector-using-convolutional-neural-network\\_fig3\\_324231417](https://www.researchgate.net/figure/Example-of-an-image-detector-using-convolutional-neural-network_fig3_324231417)

Figure 5. kNN classifier example 2.6.4 Leave One Out Cross Validation. . . (n.d.). ResearchGate. [https://www.researchgate.net/figure/kNN-classifier-example-264-Leave-One-Out-Cross-Validation-LOOCV-The-LOOCV-is-a-method\\_fig2\\_343233412](https://www.researchgate.net/figure/kNN-classifier-example-264-Leave-One-Out-Cross-Validation-LOOCV-The-LOOCV-is-a-method_fig2_343233412)

Figure 3. Illustration of linear SVM Classifier separating the two. . . (n.d.). ResearchGate. [https://www.researchgate.net/figure/Illustration-of-linear-SVM-Classifier-separating-the-two-classes-Illustration-of-linear\\_fig1\\_359803757](https://www.researchgate.net/figure/Illustration-of-linear-SVM-Classifier-separating-the-two-classes-Illustration-of-linear_fig1_359803757)

6 Classifying based on probabilities and hyperplanes: naive Bayes and support vector machines · Machine Learning with R, the tidyverse, and mlr. (n.d.). <https://livebook.manning.com/book/machine-learning-with-r-the-tidyverse-and-mlr/chapter-6/v-4/111>

Omarzai, F. (2024, April 26). K-Means Clustering in depth - Fraidoon Omarzai - medium. Medium. <https://medium.com/@fraidoonomarzai99/k-means-clustering-in-depth-2162352779e4>

Von Schmeling, J. G. (n.d.). Proyectos beta. <https://proyectosbeta.net/2017/10/opencv-3-3-en-linux-mint-18-2-sonya/>

colaboradores de Wikipedia. (n.d.). Archivo:TensorFlow Logo with text.png - Wikipedia, la enciclopedia libre. [https://es.m.wikipedia.org/wiki/Archivo:TensorFlow\\_Logo\\_with\\_text.png](https://es.m.wikipedia.org/wiki/Archivo:TensorFlow_Logo_with_text.png)

Wikipedia contributors. (2024, June 16). Dlib. [https://en.wikipedia.org/wiki/Dlib#/media/File:Dlib\\_c++\\_library\\_logo.png](https://en.wikipedia.org/wiki/Dlib#/media/File:Dlib_c++_library_logo.png)

colaboradores de Wikipedia. (n.d.). Archivo:PyCharm Icon.svg - Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Archivo:PyCharm\\_Icon.svg](https://es.wikipedia.org/wiki/Archivo:PyCharm_Icon.svg)

Redacción. (2020, March 1). Victoria contundente. La Tribuna De Albacete. <https://www.latribunadealbacete.es/noticia/ze8a5f42a-9805-d10f-fbd43a387dcd39d0/202003/victoria-contundente>

Labelled faces in the Wild (LFW): images of various people with. . . (n.d.). ResearchGate. [https://www.researchgate.net/figure/Labelled-faces-in-the-Wild-LFW-images-of-various-people-with-different-pose\\_fig6\\_334725817](https://www.researchgate.net/figure/Labelled-faces-in-the-Wild-LFW-images-of-various-people-with-different-pose_fig6_334725817)

colaboradores de Wikipedia. (2024, June 15). Red neuronal residual. [https://es.wikipedia.org/wiki/Red\\_neuronal\\_residual#/media/Archivo:ResBlock.png](https://es.wikipedia.org/wiki/Red_neuronal_residual#/media/Archivo:ResBlock.png)

Fig. 3. The architecture of our deep residual network (ResNet) in the. . . (n.d.). ResearchGate. [https://www.researchgate.net/figure/The-architecture-of-our-deep-residual-network-ResNet-in-the-precise-stage-prediction\\_fig2\\_344779798](https://www.researchgate.net/figure/The-architecture-of-our-deep-residual-network-ResNet-in-the-precise-stage-prediction_fig2_344779798)

Blogmaldito. (2019, February 18). BLOGMALDITO: Maratón de Sevilla (9). BLOGMALDITO. <https://blogmaldito.com/2019/02/18/a-examen-zurich-maraton-sevilla/maraton-de-sevilla-9/>