



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH



FACULTAD DE INFORMÁTICA DE BARCELONA
GRADO EN INGENIERÍA INFORMÁTICA
Especialidad en Computación

Detección de fatiga y somnolencia del conductor mediante visión por computador

Autor: Adrian Martínez López
Director: Joan Climent Vilaró

19 de marzo de 2022

Índice

| | |
|---|-----------|
| 1. Introducción | 10 |
| 1.1. Contextualización y alcance | 10 |
| 1.1.1. Contexto | 10 |
| 1.1.2. Descripción del problema | 10 |
| 1.1.3. Actores implicados | 11 |
| 1.2. Antecedentes | 12 |
| 1.3. Alcance | 14 |
| 1.3.1. Objetivos | 14 |
| 1.3.2. Requisitos | 15 |
| 1.3.3. Obstáculos y riesgos | 16 |
| 1.4. Metodología y seguimiento | 16 |
| 1.4.1. Metodología | 16 |
| 1.4.2. Seguimiento | 17 |
| 1.4.3. Modificaciones respecto a la metodología inicial | 17 |
| 2. Planificación temporal | 18 |
| 2.1. Descripción de las tareas | 18 |
| 2.1.1. Gestión del Proyecto [GP] | 18 |
| 2.1.2. Estudio Previo [EP] | 19 |
| 2.1.3. Desarrollo de un producto viable mínimo [MVP] | 19 |
| 2.1.4. Desarrollo de un producto completo [PC] | 20 |
| 2.1.5. Resumen de las tareas | 20 |
| 2.2. Recursos | 21 |
| 2.2.1. Recursos Humanos | 21 |
| 2.2.2. Recursos Materiales | 21 |
| 2.3. Resumen de la dedicación temporal y Gantt | 22 |
| 2.4. Gestión del Riesgo | 22 |
| 2.5. Desviaciones respecto a la planificación inicial | 23 |
| 3. Conocimientos previos | 26 |
| 3.1. Características de Haar | 26 |
| 3.2. Algoritmo de Viola-Jones | 26 |
| 3.3. Magnificación Euleriana de Vídeos | 27 |
| 3.4. Modelos de <i>Machine Learning</i> utilizados | 28 |
| 3.4.1. Árboles de decisión | 28 |
| 3.4.2. Regresión Logística | 28 |
| 3.4.3. K-Nearest Neighbors | 29 |
| 3.4.4. Linear Support Vector Machines (Linear SVM) | 30 |
| 3.4.5. Gradient Boosting | 30 |
| 3.5. Evaluación de clasificadores binarios | 30 |
| 4. Solución propuesta | 32 |
| 4.1. Regiones de interés seleccionadas | 32 |
| 4.2. Características seleccionadas | 32 |
| 4.2.1. Características relativas a una imagen estática | 32 |

| | | |
|-----------|---|-----------|
| 4.2.2. | Características relativas a una secuencia temporal | 34 |
| 4.3. | Funcionamiento del sistema implementado | 37 |
| 4.4. | Procesado de Imagen | 37 |
| 4.5. | Segmentación de las regiones de interés | 38 |
| 4.5.1. | Detección de la cara | 38 |
| 4.5.2. | Generación de la malla facial | 39 |
| 4.5.3. | Selección de los puntos faciales de interés | 39 |
| 4.6. | Extracción de características | 41 |
| 4.6.1. | Obtención de las mediciones | 42 |
| 4.6.1.1. | Mediciones del ojo | 42 |
| 4.6.1.2. | Mediciones de la boca | 44 |
| 4.6.1.3. | Mediciones de la cabeza | 44 |
| 4.6.1.4. | Alternativas valoradas | 45 |
| 4.6.2. | Determinación del estado de las regiones de interés | 46 |
| 4.6.2.1. | Análisis del estado del ojo | 46 |
| 4.6.2.2. | Análisis del estado de la boca | 49 |
| 4.6.2.3. | Análisis de la postura de la cabeza | 50 |
| 4.6.3. | Detección de parpadeos, bostezos y cabeceos | 50 |
| 4.6.4. | Cálculo de las características de carácter temporal | 51 |
| 4.7. | Reconocimiento de la somnolencia del conductor | 51 |
| 4.8. | Entrenamiento de los clasificadores | 51 |
| 4.8.1. | Preprocesado de los datos de entrenamiento | 52 |
| 4.8.2. | Elección de las características relevantes | 52 |
| 4.8.3. | Elección del mejor tipo de clasificador | 52 |
| 4.8.4. | Búsqueda de los hiperparámetros óptimos | 52 |
| 5. | Implementación | 53 |
| 5.1. | Consideraciones planteadas para el sistema | 53 |
| 5.2. | Lenguaje de programación utilizado | 53 |
| 5.3. | Procesado de imagen | 53 |
| 5.3.1. | Librerías utilizadas | 53 |
| 5.4. | Segmentación de las regiones de interés | 53 |
| 5.4.1. | Librerías utilizadas | 54 |
| 5.5. | Entrenamiento de los modelos predictivos | 55 |
| 5.5.1. | Librerías utilizadas | 55 |
| 5.5.2. | Dataset utilizado: NTHUDDD | 57 |
| 5.5.2.1. | Introducción | 57 |
| 5.5.2.2. | Tratamiento | 59 |
| 5.5.3. | Entrenamiento de los modelos | 60 |
| 5.6. | Mecanismo de inferencia | 61 |
| 5.6.1. | Entrada | 61 |
| 5.6.2. | Salida | 61 |
| 5.6.3. | Ejecución del programa | 62 |
| 5.7. | Tiempo de ejecución del sistema | 62 |

| | |
|--|-----------|
| 6. Resultados | 64 |
| 6.1. Experimentos relacionados con los métodos de obtención de características | 64 |
| 6.1.1. Detección de parpadeos sobre secuencias de vídeos | 64 |
| 6.1.2. Detección de bostezos sobre secuencias de vídeo | 71 |
| 6.1.3. Detección de cabeceos sobre secuencias de vídeo | 76 |
| 6.2. Experimentos relacionados con los modelos de detección del cansancio | 81 |
| 7. Informe de Sostenibilidad | 88 |
| 7.1. Estudio del impacto Ambiental | 88 |
| 7.2. Estudio del impacto Económico | 89 |
| 7.2.1. Costes de personal | 89 |
| 7.2.2. Costes por actividad | 90 |
| 7.2.3. Costes genéricos | 92 |
| 7.3. Coste Total del Proyecto | 93 |
| 7.4. Control de Gestión | 93 |
| 7.5. Desviaciones respecto a la planificación inicial | 94 |
| 7.6. Estudio del impacto Social | 97 |
| 8. Conclusiones y trabajo futuro | 99 |
| 8.1. Conclusiones | 99 |
| 8.2. Trabajo futuro | 99 |

Índice de figuras

| | |
|---|----|
| 1. Accidente de tráfico. | 10 |
| 2. Evolución del número total de accidentes de tráfico en España entre 2011 y 2020. . | 11 |
| 3. Resultados que han de mostrar los algoritmos de detección. | 15 |
| 4. Esquema de la metodología <i>Agile</i> | 17 |
| 5. Dependencias de las tareas del proyecto. | 21 |
| 6. Diagrama de Gantt de la dedicación temporal al proyecto. | 25 |
| 7. Características de Haar. | 26 |
| 8. <i>Cascade of classifiers</i> utilizado por el algoritmo de Viola-Jones. | 27 |
| 9. Resultados de aplicar la magnificación euleriana para visualizar el pulso. | 27 |
| 10. Árbol de decisión de 4 variables. | 28 |
| 11. Funcionamiento del algoritmo de KNN con $k=3$ | 29 |
| 12. Hiperplano que maximiza el margen entre las observaciones | 30 |
| 13. Relación de aspecto del ojo | 33 |
| 14. Relación de aspecto de la boca | 33 |
| 15. Ángulos de navegación de la cabeza | 34 |
| 16. Diagrama de flujo correspondiente al funcionamiento del sistema | 37 |
| 17. Diagrama de flujo correspondiente a la segmentación de las regiones de interés . | 38 |
| 18. Resultado de aplicar el algoritmo de detección de caras | 38 |
| 19. Malla facial generada a partir de la imagen recortada de la cara | 39 |
| 20. Puntos faciales seleccionados de los ojos | 40 |
| 21. Puntos faciales seleccionados de la boca | 40 |
| 22. Puntos faciales seleccionados de la cara | 41 |
| 23. Comparación del funcionamiento de Viola-Jones en función del estado de los ojos. | 41 |
| 24. Diagrama de flujo correspondiente a la obtención de características del <i>frame</i> actual | 42 |

| | | |
|-----|---|----|
| 25. | Cálculo del EAR euclidiano | 42 |
| 26. | Cálculo del EAR no euclidiano | 43 |
| 27. | Cálculo de la apertura del ojo respecto al diámetro del iris | 43 |
| 28. | Comparación del método geométrico en función del ángulo de <i>yaw</i> de la persona. La línea indica la dirección estimada de la pose de la cabeza. | 46 |
| 29. | Evolución del EAR durante el transcurso del vídeo | 47 |
| 30. | Comportamiento del <i>threshold</i> dinámico con $n=100$ y $n=5$, respectivamente. | 48 |
| 31. | Comportamiento del intervalo de incertidumbre con $gz=0.05$ y $gz=0.1$, respectivamente. | 49 |
| 32. | Evolución del MAR para un suavizado con $m=30$ y $m=5$, respectivamente | 50 |
| 33. | Diagrama de flujo correspondiente al entrenamiento de los clasificadores | 52 |
| 34. | Soluciones ofrecidas por Mediapipe | 54 |
| 35. | Mallas faciales generadas por el modelo | 55 |
| 36. | Contenido de los ficheros de anotaciones | 59 |
| 37. | Características extraídas del <i>frame</i> actual y de la secuencia temporal | 60 |
| 38. | Características del <i>frame</i> puntual en formato de diccionario | 60 |
| 39. | Extracto del fichero csv que contiene las características del vídeo tratado | 60 |
| 40. | Evolución de la calidad de las respuestas según el valor del <i>threshold</i> estático | 66 |
| 41. | Comparación del rendimiento de los métodos para los distintos sujetos | 67 |
| 42. | Evolución de los resultados según los dos parámetros a valorar para los métodos 1, 2 y 3, respectivamente. | 68 |
| 43. | Comparativa del rendimiento de los métodos implementados para los distintos sujetos | 69 |
| 44. | Evolución del rendimiento de los métodos según el valor del <i>threshold</i> estático | 72 |
| 45. | Evolución de las respuestas para el método 1 | 73 |
| 46. | Evolución de las respuestas para el método 2 | 74 |
| 47. | Evolución de las respuestas según el valor del <i>threshold</i> estático | 77 |
| 48. | Evolución de los parámetros para el método 1 | 78 |
| 49. | Evolución de los parámetros para el método 2 | 78 |
| 50. | Importancia de las variables sobre las respuestas del modelo. | 83 |
| 51. | Matriz de confusión de las predicciones del modelo sobre el <i>dataset</i> de <i>test</i> | 86 |
| 52. | Curvas de precision-exhaustividad y ROC | 87 |
| 53. | Rendimiento del clasificador final para los diferentes sujetos del <i>dataset</i> de <i>test</i> | 87 |

Índice de tablas

| | | |
|-----|--|----|
| 1. | Comparativa de los distintos tipos de soluciones para la detección del cansancio en conductores. | 13 |
| 2. | Tiempo real dedicado a las tareas. | 24 |
| 3. | Matriz de confusión para un problema de clasificación binario | 31 |
| 4. | Escenarios posibles para el tratamiento de los intervalos | 50 |
| 5. | Rendimiento de los diferentes métodos implementados | 67 |
| 6. | Comparativa del rendimiento de los métodos implementados | 68 |
| 7. | Rendimiento de los diferentes métodos implementados | 70 |
| 8. | Rendimiento de los diferentes métodos implementados | 72 |
| 9. | Rendimiento de los diferentes métodos implementados | 74 |
| 10. | Rendimiento de los diferentes métodos implementados | 75 |
| 11. | Rendimiento de los diferentes métodos implementados | 76 |

| | | |
|-----|---|----|
| 12. | Rendimiento de los diferentes métodos implementados | 78 |
| 13. | Rendimiento de los diferentes métodos implementados | 79 |
| 14. | Comparativa del rendimiento de los métodos implementados | 80 |
| 15. | Calidad de las respuestas del clasificador inicial | 82 |
| 16. | Calidad de las respuestas del clasificador tras eliminar las variables irrelevantes . | 83 |
| 17. | Calidad de las respuestas de cada clasificador | 84 |
| 18. | Tiempo de entrenamiento y predicción de cada clasificador | 84 |
| 19. | Comparativa del modelo base con el modelo optimizado | 85 |
| 20. | Calidad de las respuestas del modelo que se ha escogido | 86 |
| 21. | Costes de los diferentes roles del proyecto | 90 |
| 22. | Resumen de los costes por actividad del proyecto. | 91 |
| 23. | Costes de los recursos <i>Software</i> del proyecto. | 92 |
| 24. | Costes de los recursos <i>Hardware</i> del proyecto. | 92 |
| 25. | Resumen de los costes genéricos estimados. | 93 |
| 26. | Suma de los costes totales del proyecto. | 93 |
| 27. | Resumen de los desvíos en los costes por actividad. | 95 |
| 28. | Comparación de los costes estimados y los costes reales del proyecto. | 95 |

Agradecimientos

Quiero agradecer al tutor de mi TFG, Joan Climent Vilaró, por haberme acompañado en el desarrollo del proyecto y haberme guiado en los momentos difíciles. Sin él, la realización de este TFG no habría sido posible.

Por otro lado, quiero agradecer también al Departamento de Visión por Computador de la Universidad Nacional de Tsing Hua, por haberme dado acceso al *dataset*[8] mediante el cual he podido llevar a cabo el desarrollo del proyecto.

Muchas gracias!

Abstract

Nowadays, traffic accidents are events that happen very frequently and that result in grave consequences for all the users of the road that see themselves directly involved. Their impact is not only limited to those victims which are involved directly, but their repercussions also affect a big amount of individuals in an indirect manner.

One of the main causes of these accidents is the presence of drivers that suffer from fatigue and/or somnolence on the asphalt, because their level of attention to the road is severely reduced and they are much more prone to be distracted or to carry out erratic maneuvers.

The main goal of this project is to develop an algorithm capable of detecting the fatigue and/or somnolence of drivers through the application of computer vision techniques, in order to allow the application of strategies to keep the drivers alert, anticipate possible accidents and keep road safety.

After analyzing the set of available alternatives, the system has been designed to perform a manual extraction of a set of features that describe the drivers' behaviour through the use of computer vision techniques. Once these features have been generated, machine learning techniques are applied to determine the drivers' fatigue level based on the set of features.

Resumen

Hoy en día los accidentes de tráfico son eventos que ocurren con bastante frecuencia y que acarrear consecuencias muy graves para todos los usuarios de la vía que se encuentran implicados. El impacto de estos no solo se limita a las víctimas que se ven involucradas de manera directa, sino que muchos individuos se ven afectados indirectamente por sus repercusiones.

Una de las principales causas de estos accidentes es la presencia de conductores en la vía que padecen de síntomas de fatiga y/o somnolencia, puesto que su grado de atención a la carretera es mucho más reducido y resultan mucho más susceptibles a distracciones y a realizar maniobras erráticas.

Este proyecto tiene como objetivo desarrollar un algoritmo capaz de detectar los síntomas de fatiga y/o somnolencia de los conductores mediante técnicas de visión por computador, para habilitar la aplicación de estrategias que alerten a los conductores, prevengan posibles accidentes y mantengan la seguridad vial.

Después de plantear las diferentes alternativas disponibles, para desarrollar este sistema se ha optado por realizar una extracción manual de un conjunto de indicadores que describen el comportamiento de los conductores mediante técnicas de visión por computador. Una vez se han obtenido dichos indicadores, se ha optado por aplicar algoritmos de *Machine Learning* para determinar el estado de fatiga de los conductores a partir de estas características.

Resum

Avui dia els accidents de trànsit són events que succeeixen amb bastant freqüència i que comporten conseqüències molt greus per a tots els usuaris de la via que es troben implicats. L'impacte d'aquests no solament es limita a les víctimes que es veuen involucrades de manera directa, sinó que molts individus es veuen afectats indirectament per les repercussions que aquests impliquen.

Una de les principals causes d'aquests accidents és la presència de conductors a la via que pateixen de símptomes de fatiga i/o somnolència, ja que el seu grau d'atenció a la carretera és molt més reduït i resulten molt més susceptibles a distraccions i a realitzar maniobres erràtiques.

Aquest projecte té com a objectiu desenvolupar un algorisme capaç de detectar els símptomes de fatiga i/o somnolència dels conductors mitjançant tècniques de visió per computador, per habilitar l'aplicació d'estratègies que alertin als conductors, previnguin possibles accidents i mantinguin la seguretat vial.

Després de plantejar les diferents alternatives disponibles, per a desenvolupar aquest sistema s'ha optat per realitzar una extracció manual d'un conjunt d'indicadors que descriuen el comportament dels conductors mitjançant tècniques de visió per computador. Un cop s'han obtingut aquests indicadors, s'ha optat per aplicar algorismes de *Machine Learning* per a determinar l'estat de fatiga dels conductors a partir d'aquestes característiques.

Introducción

1.1 Contextualización y alcance

1.1.1 Contexto

Hoy en día los accidentes de tráfico son eventos que ocurren con bastante frecuencia y que acarrear consecuencias muy graves para todos los usuarios de la vía que se encuentran implicados.



Figura 1: Accidente de tráfico.

Fuente: [2]

Una de las principales causas de estos accidentes es la presencia de conductores que muestran síntomas de fatiga y/o somnolencia en la carretera, puesto que su grado de atención a la vía es mucho más reducido y son mucho más susceptibles a distracciones.

Debido a las graves consecuencias que estos accidentes acarrear, es de vital importancia dotar a los vehículos de mecanismos de seguridad que, no solo reduzcan las consecuencias de los accidentes, sino que también traten de anticiparse a ellos y prevenirlos.

Teniendo esto en cuenta, el objetivo de este trabajo consiste en el desarrollo de un algoritmo capaz de detectar con precisión los síntomas de fatiga y/o somnolencia que presentan los conductores, y de alertarlos para así prevenir posibles accidentes, mantener la seguridad vial y, alineado con la meta 3.6 de los ODS de la ONU [33] en referencia a la reducción del número de muertes y de lesiones causados por los accidentes de tráfico.

Dentro del marco de la Facultad de Informática de Barcelona, este Trabajo de Fin de Grado pertenece a la mención de Computación, y, más concretamente, al ámbito de Visión por Computador.

1.1.2 Descripción del problema

Como ya se ha introducido en el apartado anterior, los accidentes de tráfico están a la orden del día y las consecuencias que acarrear causan un perjuicio muy significativo para nuestra sociedad.

Como podemos observar en la figura 2, el número de accidentes graves y mortales que suceden cada año en España ha sufrido un crecimiento significativo desde 2011, y, si tenemos en cuenta que

los datos de 2020 se encuentran distorsionados por una pandemia que redujo significativamente la actividad en las carreteras, la tendencia que se observa es bastante preocupante.

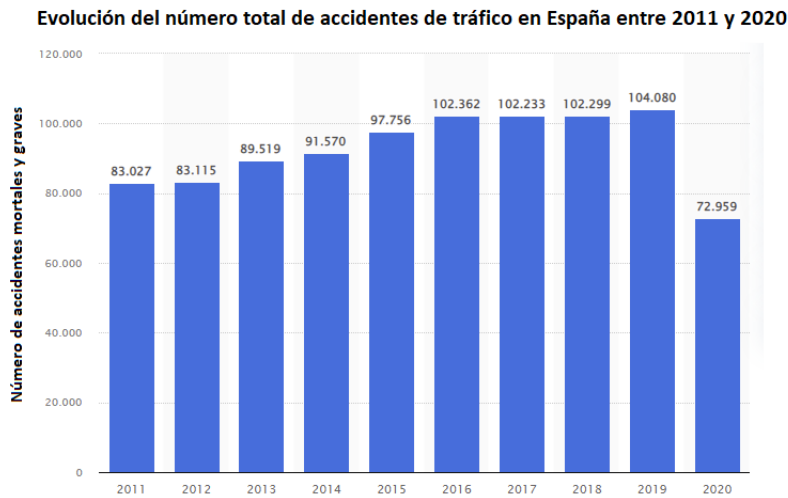


Figura 2: Evolución del número total de accidentes de tráfico en España entre 2011 y 2020.

Fuente: [40]

Teniendo en cuenta que una de las principales causas de estos accidentes es la presencia de conductores que padecen de fatiga en las carreteras, la ausencia de mecanismos que puedan detectar la somnolencia de los conductores y alertarlos para así prevenir los accidentes resulta inaceptable.

El objetivo de este trabajo, por lo tanto, consiste en el desarrollo de un algoritmo capaz de detectar con precisión y robustez los síntomas de fatiga y somnolencia que presentan los conductores, además de ser fácilmente integrable en posibles dispositivos *hardware* que puedan ser incorporados en los vehículos del futuro.

Este trabajo se centrará puramente en el desarrollo del *software* capaz de realizar la detección de los síntomas de fatiga mediante **visión por computador**, y únicamente planteará la posibilidad de ser integrado en dispositivos *hardware* que puedan aprovecharlo para alertar a los conductores en caso de detección de fatiga.

1.1.3 Actores implicados

Respecto a la realización del proyecto, en el personal del mismo se pueden destacar dos roles.

- **Director del proyecto:** Guiará al autor del TFG en la realización del trabajo. Se trata de un docente de la FIB, en este caso **Joan Climent Vilaró**.
- **Autor del proyecto:** Principalmente llevará a cabo el desarrollo del trabajo. Se trata de un estudiante de la facultad, en este caso **Adrián Martínez López**.

Respecto a los resultados de este proyecto, estos van dirigidos principalmente a dos grandes colectivos.

Principales Usuarios

- **Industria Automotriz**

En el proceso de manufactura de los vehículos se podrán incorporar dispositivos que obtengan imágenes de la cara del conductor y que utilicen los resultados de este proyecto para detectar síntomas de fatiga y/o somnolencia y alertar al conductor en caso de que sea necesario.

- **Conductores de vehículos**

Utilizarán los dispositivos diseñados por la industria automotriz, que se encontrarán incorporados en sus vehículos, de manera pasiva. En caso de mostrar síntomas de fatiga y de perder la concentración, serán alertados para volver a dirigir su atención a la vía.

Colectivos Beneficiados

El colectivo que resultará más beneficiado de que este proyecto sea llevado a cabo con éxito es, principalmente, el conjunto de todos los usuarios de la vía, y, si realizamos una valoración más profunda, podemos llegar a hablar de toda la sociedad.

Una reducción significativa en el número de los accidentes supondría, por un lado, una **disminución del número de fallecidos y heridos**, reduciendo así potenciales consecuencias psicológicas, el número de situaciones familiares comprometidas, el grado de ocupación en hospitales, posibles daños materiales, etc.

Por otro lado, la congestión del tráfico se vería reducida, dando lugar a una **circulación más fluida** y causando menos perjuicio a todos los usuarios de la vía.

1.2 Antecedentes

Existen infinidad de enfoques y estrategias que se han llevado a cabo en la historia para tratar de realizar una detección de la fatiga en los conductores lo más precisa posible.

A gran escala, podemos dividir los diferentes tipos de soluciones en tres grandes grupos:

- Soluciones basadas en **parámetros del comportamiento del conductor**[30]: Analizan características como la frecuencia de parpadeo, la detección de los bostezos, la postura del conductor, expresiones faciales, etc.

Son soluciones pertenecientes principalmente al ámbito de Visión por Computador, puesto que todas estas características se extraen de secuencias de imágenes de los conductores.

- Soluciones basadas en **parámetros del vehículo**[30]: Valoran características como la variación de velocidad del vehículo, el ángulo del volante, patrones de movimiento del vehículo, etc.

- Soluciones basadas en **parámetros fisiológicos del conductor**[30]: Hacen uso de diferentes tipos de sensores para analizar parámetros como el pulso, señales de EEG y ECG, etc.

Son soluciones bastante sofisticadas pero que resultan invasivas para el usuario y no tan fáciles de aplicar a gran escala, puesto que suelen requerir el uso de aparatos de medición.

| | Parámetros del comportamiento del conductor | Parámetros del comportamiento del vehículo | Parámetros fisiológicos del conductor |
|------------------------|--|---|--|
| Características | - Frecuencia de parpadeo - Detección de bostezos - Postura del conductor - Expresiones faciales - Apertura de los ojos | - Velocidad del vehículo - Ángulo del volante - Patrones de movimiento del vehículo | - Pulso del conductor - Señales de EEG - Señales de ECG - Frecuencia de respiración |
| Ventajas | - Rendimiento aceptable - No resulta intrusivo - Técnicas fáciles de aplicar | - Rendimiento aceptable - No resulta intrusivo | - Rendimiento muy preciso y robusto - Eficiente |
| Desventajas | - Poco robusto a cambios de iluminación | - Poco robusto a cambios en las condiciones de la carretera | - Muy intrusivo |

Tabla 1: Comparativa de los distintos tipos de soluciones para la detección del cansancio en conductores.

Fuente: Elaboración propia.

La solución elegida para el desarrollo de este proyecto tiene que ser **no invasiva**, puesto que, idealmente, ha de formar parte de la vida cotidiana de los conductores. Además, otro requisito muy importante es que se trate de una solución **eficiente, precisa y robusta**. Teniendo también en cuenta que este trabajo pertenece al ámbito de Visión por Computador, las soluciones que se han explorado están basadas principalmente en el análisis de los **parámetros relacionados con el comportamiento del conductor**.

Cabe destacar que existen diferentes métodos de obtención de los parámetros relacionados con el comportamiento del conductor, entre los que se puede destacar una obtención explícita de las pautas de comportamiento a partir de sistemas de detección, y una extracción implícita de estas, que se basa en el uso de descriptores, como por ejemplo el histograma de gradientes orientados (HOG).

Una vez se han obtenido dichas características, un algoritmo de clasificación se encarga de determinar si corresponden a un sujeto cansado o no.

Ambos métodos de extracción pueden proporcionar buenos resultados, y se ha considerado más interesante realizar una extracción explícita de un cierto conjunto de comportamientos que, a priori, pueden ser buenos indicadores de la somnolencia de una persona:

- **Frecuencia de parpadeo**
- **Bostezos del conductor**

- **Expresiones faciales**
- **Grado de apertura de los ojos**

Es indudable que la extracción explícita de características que pueden ser identificadas con el comportamiento de una persona, como la frecuencia de los bostezos, da lugar a modelos clasificadores cuya respuesta puede ser entendida más fácilmente.

Por otro lado, cabe destacar que la naturaleza de este proyecto es exploratoria, y, por lo tanto, es de esperar que se añadan o modifiquen nuevas características conforme el desarrollo del trabajo evolucione con el tiempo.

Por último, cabe también mencionar que, como ya se ha mencionado, se han entrenado modelos de Machine Learning a partir de los conjuntos de características obtenidos para tratar de resolver el problema de la detección del cansancio, puesto que se han podido encontrar conjuntos de datos con la suficiente volumetría.

Modificaciones respecto a la planificación inicial

Durante el transcurso del proyecto han ocurrido acontecimientos que han provocado la necesidad de cambiar las vías de trabajo planteadas en las etapas iniciales de este proyecto.

Los cambios más significativos que se han realizado son los siguientes.

- **Descarte de los parámetros fisiológicos** – Inicialmente, durante la planificación del proyecto se planteó trabajar principalmente con parámetros relacionados con el comportamiento del conductor. También se tuvo en cuenta que este tipo de soluciones presentan un rendimiento más limitado cuando las condiciones de iluminación no son favorables, y, por lo tanto, se planteó introducir parámetros fisiológicos, como el pulso del conductor, para aumentar la robustez de las mismas.

Finalmente, no ha sido posible introducir estos indicadores, puesto que los métodos que se han valorado para llevar a cabo su detección no eran lo suficientemente precisos, y, por lo tanto, se han descartado y se ha optado por implementar soluciones basadas completamente en el comportamiento del conductor.

- **Utilización de algoritmos de *Machine Learning*** – En las etapas iniciales del proyecto no se tenía acceso a ningún *dataset* que permitiera llevar a cabo un entrenamiento de modelos de *Machine Learning* y no se tenía la seguridad de que fuera posible obtener acceso a uno. Por esta razón se descartó inicialmente la opción de utilizar clasificadores para resolver el problema.

Finalmente, durante el transcurso del proyecto se ha conseguido acceso a un *dataset*[8] muy utilizado para valorar la calidad de las soluciones propuestas por diferentes proyectos del Estado del Arte, y esto ha motivado el desarrollo de soluciones basadas en *Machine Learning*.

1.3 Alcance

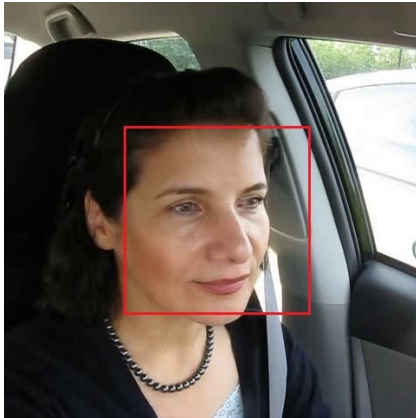
1.3.1 Objetivos

El objetivo de este proyecto consiste en el desarrollo de un algoritmo capaz de detectar con precisión la fatiga y somnolencia de los conductores mediante **visión per computador**.

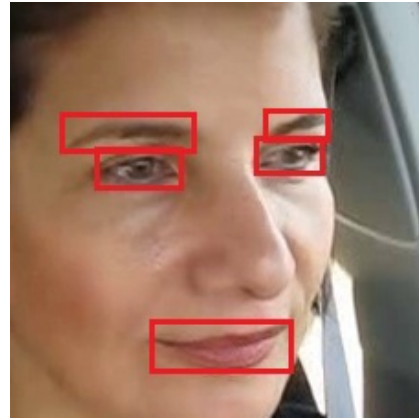
Subobjetivos

Los subobjetivos en los que se puede dividir el proyecto son los siguientes:

- **Localización de la cara del conductor:** Implementación de un algoritmo que detecte con precisión la *bounding box* de la cara del conductor, como se puede ver en la figura 3, para así reducir el tamaño de la imagen a analizar.
- **Localización de las regiones de interés:** Desarrollo de un mecanismo que localice con precisión las regiones de interés, que se puede observar en la figura 3, de la cara del conductor: ojos, cejas, boca, etc.



(a) *Bounding box* de la cara del conductor



(b) *Bounding boxes* de las regiones de interés

Figura 3: Resultados que han de mostrar los algoritmos de detección.

Fuente: Elaborado a partir de [37]

- **Obtención de características a partir de las regiones de interés:** Obtención de variables como la frecuencia de parpadeo, la frecuencia de movimiento de las pupilas, la posición de las cejas o el pulso cardíaco a partir de las regiones de interés.
- **Integración de las diferentes características:** Realización de una ponderación sobre las características de cada región de interés para dar más importancia a ciertas regiones de la cara.
- **Detección de la fatiga del conductor a partir de las características:** Desarrollo de un sistema que determine si el conductor padece síntomas de fatiga a partir de las características obtenidas.

1.3.2 Requisitos

- **Precisión** — El producto resultante ha de resolver el problema que se plantea con un cierto grado de precisión.
- **Eficiencia** — El algoritmo resultante ha de ser eficiente y ha de dar respuestas rápidas. Es crucial que el retraso en la respuesta sea lo más mínimo posible, puesto que los accidentes pueden ocurrir en cuestión de segundos.

- **Usabilidad** – Es importante que el sistema pueda ser integrado en dispositivos pequeños y no invasivos para el usuario, como por ejemplo, aquellos diseñados para formar parte del conjunto de mecanismos de seguridad de los vehículos.
- **Reusabilidad** – Es interesante que el sistema sea fácilmente integrable de manera modular en otros proyectos de mayor escala.

1.3.3 Obstáculos y riesgos

Uno de los principales obstáculos a tener en cuenta es el hecho de que no se tiene la certeza de que se pueda obtener acceso a un conjunto de datos que permita determinar con certeza si una respuesta del sistema es correcta o no. Se tendrán que realizar **valoraciones intuitivas** para determinar la calidad de las respuestas, puesto que no siempre se puede saber con certeza si una persona se encuentra fatigada.

Otro obstáculo a valorar es el hecho de que el proyecto se realiza en un periodo de tiempo relativamente limitado. La capacidad de exploración de las diferentes estrategias se verá limitada y será necesario llevar a cabo una metodología ágil.

Un riesgo a tener en cuenta es la posibilidad de que las características obtenidas no sean suficientes para dar respuestas lo suficientemente precisas, en cuyo caso se trataría de explorar nuevas pautas de comportamiento más sofisticadas con el objetivo de aumentar la robustez del sistema.

Un último riesgo a tener en cuenta es la posibilidad de que los algoritmos de detección no sean lo suficientemente robustos, en cuyo caso la calidad de las características obtenidas y de las respuestas del algoritmo de detección del cansancio no tendrá tanta importancia, puesto que las regiones de interés de las que parte el análisis pueden no ser correctas.

1.4 Metodología y seguimiento

1.4.1 Metodología

Puesto que el tiempo de desarrollo del proyecto es bastante limitado, se ha escogido una metodología que permite explorar diversas soluciones y estrategias para los diferentes subobjetivos que se han planteado. Teniendo en cuenta, además, que estos presentan un orden ya establecido por sus dependencias, se ha planteado el uso de la metodología *Agile*[1].

La idea principal de esta metodología consiste en dividir el desarrollo de un proyecto en *sprints*. Cada una de estas iteraciones tiene como objetivo el desarrollo de un sistema mínimamente viable, cuya complejidad aumentará de manera incremental, permitiendo así la exploración rápida de diferentes estrategias y enfoques para resolver un problema complejo.

Como podemos observar en la figura 4, en cada *sprint* se llevan a cabo las mismas etapas de desarrollo:

1. **Planificación:** Consiste en el planteamiento de los diferentes enfoques y estrategias para cumplir con el objetivo, y de las tareas que deberían llevarse a cabo en el *sprint*.
2. **Diseño:** Una vez se ha optado por un enfoque para cumplir con el objetivo, plantear con más profundidad los detalles técnicos para materializar dicha solución.
3. **Desarrollo:** Consiste en la etapa en la que se materializa el diseño planteado y se desarrolla el código necesario para resolver el problema.

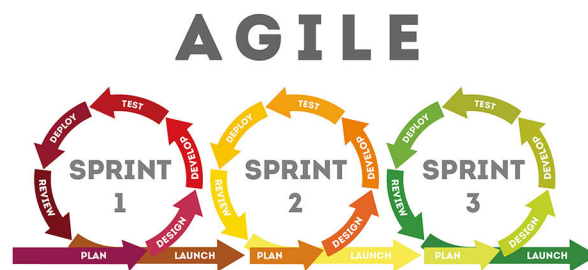


Figura 4: Esquema de la metodología *Agile*.

Fuente: [15]

4. **Verificación:** Una vez se ha desarrollado un mecanismo capaz de dar respuestas al problema planteado, se determinará la calidad de estas respuestas y el grado de robustez del sistema mediante tests.
5. **Publicación:** Una vez se han verificado unos resultados aceptables, se publica el modelo y se avanza al siguiente *sprint* con el objetivo de obtener un sistema más elaborado.

Método de validación de los modelos

Puesto que finalmente se ha obtenido acceso a conjuntos de datos que permiten valorar de manera estadística la calidad de los resultados de las iteraciones, se ejecutarán los sistemas sobre estos *datasets* y se valorará la precisión de los resultados obtenidos.

De esta manera, los sistemas obtenidos podrán ser comparados entre ellos de manera objetiva, además de obtener una referencia acerca del rendimiento de los productos desarrollados en comparación con las soluciones del Estado del Arte que se han valorado sobre los mismos *datasets*.

1.4.2 Seguimiento

Como herramienta de seguimiento del proyecto, se ha utilizado *git* para mantener un repositorio en el que se ha realizado un seguimiento de las diferentes iteraciones del proyecto y mantenido una copia de seguridad para prevenir la pérdida del código. Las nuevas implementaciones y estrategias se han llevado a cabo en primer lugar en ramas de desarrollo, y, una vez se han superado los tests de la etapa de verificación, los cambios han sido incorporados a la rama principal.

Además, se han programado reuniones con el director del TFG de manera periódica para valorar el desarrollo del proyecto y resolver los problemas o inquietudes que han surgido durante el transcurso de este.

1.4.3 Modificaciones respecto a la metodología inicial

El único cambio que se ha llevado a cabo en la metodología respecto al planteamiento inicial ha sido el sistema de valoración de los productos obtenidos en cada *sprint*, puesto que se ha conseguido acceso a conjuntos de datos anotados que no estaban disponibles en el momento en que se realizó el planteamiento inicial del proyecto.

Planificación temporal

La finalidad de esta sección consiste en determinar y describir las diferentes tareas que se han llevado a cabo para la exitosa realización del proyecto y en describir las desviaciones que han ocurrido durante el transcurso del desarrollo.

El proyecto empezó el día 14 de febrero de 2022 y ha acabado el 20 de junio del mismo año.

2.1 Descripción de las tareas

2.1.1 Gestión del Proyecto [GP]

La gestión del proyecto abarca la contextualización y definición del trabajo a realizar, así como la documentación y planificación temporal del mismo. También comprende las reuniones con el director del proyecto para validar los objetivos propuestos y valorar las soluciones desarrolladas. Se trata de una base fundamental para cimentar la correcta realización del proyecto y se le han dedicado alrededor de 140 horas.

Contexto y Alcance [GP_1]

Consiste en la fase de descripción de la identidad del proyecto. Se proporciona un contexto y una motivación que justifican la realización del trabajo, así como una descripción más detallada de los objetivos que se planea cumplir y de las metodologías de trabajo que se van a aplicar. Se han invertido 15 horas para completar esta tarea.

Planificación temporal [GP_2]

Consiste en la fase de definición de las tareas en las que se divide el proyecto, su estimación temporal y la descripción de sus posibles dependencias. También se describen los recursos que han sido necesarios para la realización del proyecto, así como los planes alternativos que se valoran ante los obstáculos descritos en la definición del contexto. La duración de esta fase ha sido de 10 horas.

Presupuesto y Sostenibilidad [GP_3]

Se ha realizado una estimación del presupuesto necesario para poder llevar a cabo el proyecto. Se ha hecho una partida por cada tarea teniendo en cuenta los costes del personal, así como una cuantificación global de costes genéricos y partidas de imprevistos. Por otro lado, se han realizado estimaciones sobre el impacto social, económico y medioambiental que puede conllevar la realización del proyecto. Puesto que el trabajo a realizar es de investigación y los recursos materiales necesarios son escasos, la dedicación ha sido modesta, de 10 horas.

Reuniones [GP_4]

Se han realizado reuniones periódicamente con el director del TFG para revisar el progreso del proyecto, establecer los siguientes objetivos a cumplir y aclarar las posibles dudas e inquietudes han surgido. La dedicación ha sido de unas 15 horas.

Documentación [GP_5]

La documentación del proyecto es una parte fundamental de la realización del mismo, puesto que permitirá en el futuro una comprensión rápida de la naturaleza del trabajo y de las conclusiones obtenidas en la finalización del mismo. Cabe destacar que la documentación se ha llevado a cabo de forma paralela a la realización del proyecto, tratando de plasmar fielmente las diferentes etapas de este. Debido a la importancia que tiene la presentación de este documento, se han dedicado 75 horas para completarla.

Presentación [GP_6]

Finalmente, una vez finalizada la documentación y habiendo obtenido unas conclusiones acerca de los resultados del proyecto, se realizará una presentación para el tribunal del TFG, donde se expondrán las diferentes etapas por las que ha pasado el proyecto y se realizará una defensa de los resultados obtenidos. Se preparará material de soporte audiovisual para la presentación y se invertirá tiempo en la realización de ensayos. Se estima que la dedicación será de 10 horas.

2.1.2 Estudio Previo [EP]

Estudio del estado del arte [EP_1]

Para determinar la dirección en la que debería encararse el desarrollo del proyecto es necesario investigar las técnicas actuales que obtienen mejores resultados y determinar como pueden ser incorporadas y mejoradas. Debido a que el autor del proyecto ya posee unos fundamentos sólidos de **visión por computador** gracias a haber cursado la asignatura homónima (GRAU-VC), la complejidad de esta tarea se ha visto reducida significativamente. Por lo tanto, se ha realizado una dedicación de 15 horas.

2.1.3 Desarrollo de un producto viable mínimo [MVP]

Puesto que se han aplicado metodologías ágiles, el transcurso del desarrollo del proyecto se ha dividido en *sprints* cuya duración ha sido aproximadamente de 2 semanas, a excepción del primero de ellos, para el que se han invertido 4 semanas debido a que comprende gran parte de la gestión del proyecto.

En el primer *sprint* se ha llevado a cabo el desarrollo de un producto viable mínimo, es decir, una versión funcional del algoritmo no muy sofisticada y que producía resultados bastante imprecisos. Se trata de una base a partir de la cual se han podido construir versiones más elaboradas durante el transcurso de los diferentes *sprints*.

Planificación [MVP_1]

Tras haber realizado un estudio del estado del arte, se han valorado diferentes estrategias punteras para la detección del cansancio, se han estudiado las subtareas necesarias para el correcto desarrollo de un MVP. Se han invertido 15 horas.

Diseño [MVP_2]

Una vez se ha optado por un enfoque y se han determinado los subobjetivos a cumplir, se ha realizado un planteamiento más profundo de los detalles técnicos necesarios para materializar dicha estrategia. La dedicación ha sido de 10 horas aproximadamente.

Implementación [MVP_3]

Una vez se ha realizado un diseño del producto, se ha desarrollado el código necesario para obtener un mecanismo funcional. A grandes rasgos, se puede dividir la implementación en los subobjetivos descritos en el contexto y alcance del proyecto. Puesto que esta implementación ha sido la base a partir de la que se ha construido el resto del proyecto, se trata de una tarea realmente importante, de manera que se han dedicado 30 horas para completarla.

Validación [MVP_4]

Otra tarea de suma importancia para el proyecto ha sido la validación del rendimiento de los algoritmos que se han implementado. La correctitud de los algoritmos implementados ha sido verificada mediante tests de validación. Cabe destacar que la validación se ha realizado de forma paralela a la implementación del sistema, sobre los subobjetivos que se han ido cumpliendo, y también de manera posterior al desarrollo del MVP, teniendo en cuenta que el rendimiento del producto no ha sido el objetivo principal en este primer *sprint*. Se han dedicado 10 horas para completar esta tarea.

2.1.4 Desarrollo de un producto completo [PC]

Este grupo de tareas ha ocupado los 6 *sprints* restantes de la realización del proyecto. Como ya se ha comentado, se han llevado a cabo mejoras o modificaciones sobre los MVP ya obtenidos con el objetivo de desarrollar productos resultantes más sofisticados. De manera análoga al desarrollo del primer MVP, por cada *sprint* se han definido las tareas de:

- **Planificación [PC_#S_1]**
- **Diseño [PC_#S_2]**
- **Implementación [PC_#S_3]**
- **Desarrollo [PC_#S_4]**

Donde #S corresponde al identificador del *sprint* al que pertenecen las tareas.

Puesto que cada *sprint* ha tenido una dedicación temporal distinta, en la tabla XX se presenta un resumen del tiempo dedicado a cada tarea .

2.1.5 Resumen de las tareas

En la figura 5 se presenta un grafo que muestra las diferentes tareas que comprenden el desarrollo del proyecto junto con sus dependencias. Cabe destacar que se han omitido algunos *sprints* con el objetivo de que el grafo resultante sea más legible, debido a que todos los grupos de tareas *PC* mantienen la misma estructura y las mismas dependencias.

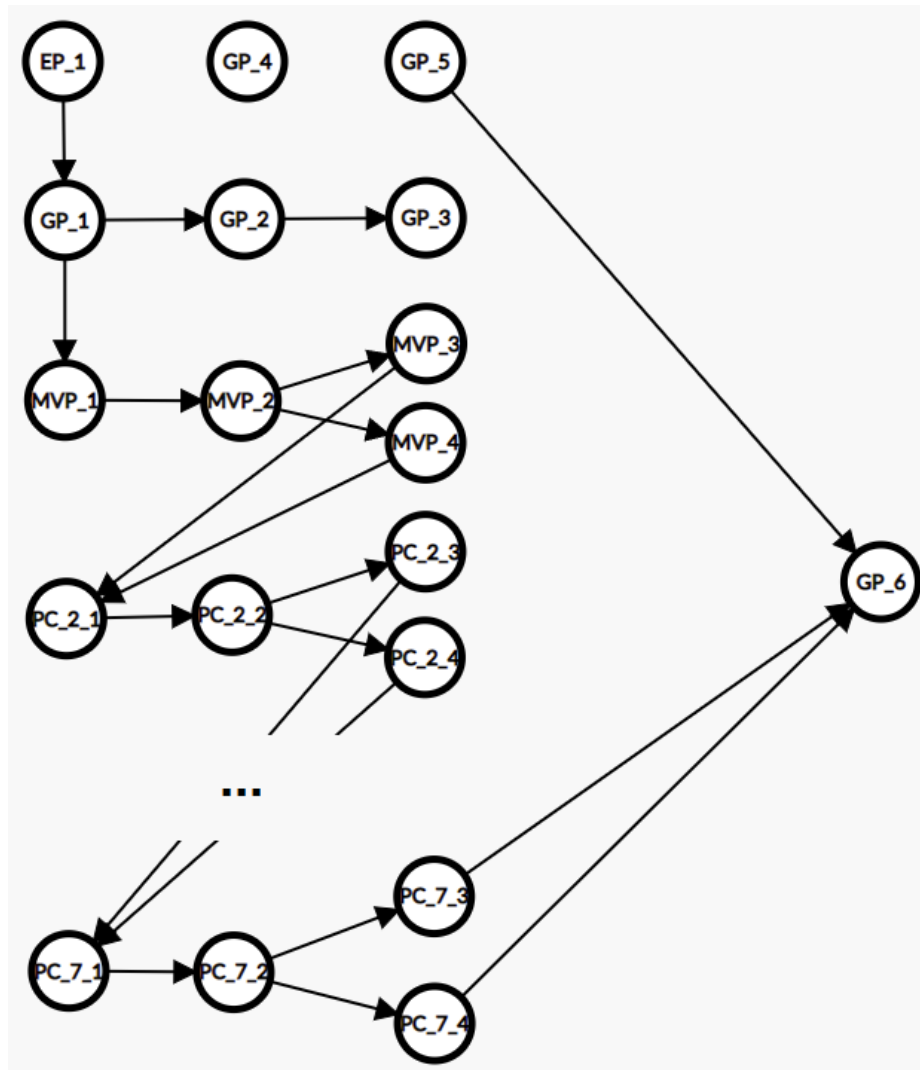


Figura 5: Dependencias de las tareas del proyecto.

Fuente: Elaboración propia

2.2 Recursos

2.2.1 Recursos Humanos

- **Director del TFG** - Se trata de un docente de la FIB, especializado en el ámbito del proyecto, que ha guiado al autor del TFG en la realización del trabajo.
- **Ingeniero Informático** - Se trata del autor del proyecto y ha llevado a cabo el desarrollo del trabajo.

2.2.2 Recursos Materiales

Los recursos materiales necesarios para la realización del proyecto han sido los siguientes:

Hardware

- Ordenador de prestaciones medias con tarjeta gráfica dedicada.

Software

- *Overleaf* - Entorno de desarrollo de documentos con \LaTeX .
- *Python* - Lenguaje de programación utilizado para el desarrollo de los diferentes algoritmos que forman parte de la solución propuesta.
- *git* - Herramienta para llevar a cabo un control de versiones y mantener copias de seguridad del código.
- *Ganttter* - Herramienta para poder realizar diagramas de Gantt.

2.3 Resumen de la dedicación temporal y Gantt

Como se puede observar en las figura 6, se ha elaborado un diagrama de Gantt que refleja la disposición temporal de todas las tareas que comprende la realización del proyecto. El diagrama se ha realizado con la herramienta Ganttter. También se puede ver un resumen de dichas tareas en la tabla 2.

2.4 Gestión del Riesgo

A continuación se exponen las principales fuentes de riesgo y obstáculos que pueden surgir en la realización del proyecto, así como las posibles soluciones y alternativas para estos. Es muy importante la realización de este análisis, puesto que prever los posibles imprevistos y tener preparada una respuesta ante ellos puede evitar un retraso muy significativo respecto a la planificación temporal.

- **Incertidumbre en la valoración de los productos desarrollados:** Debido a que existe un cierto grado de incertidumbre en el momento de determinar si un conductor se encuentra realmente fatigado, no resultará sencillo desarrollar sistemas de validación que permitan determinar el rendimiento de un cierto algoritmo. Por lo tanto, la determinación de la calidad del producto dependerá en cierto grado del criterio del desarrollador del mismo. Si se determina que las valoraciones no son concluyentes, se tendrá que aumentar la dedicación a las tareas de validación de los algoritmos que se van desarrollando.
- **Conjunto de características insuficiente:** Dada la complejidad del problema que se está tratando, es probable que el conjunto de características obtenido en los primeros *sprints* de la realización del proyecto sea insuficiente para la construcción de un mecanismo robusto. Debido a esto, se valora siempre la posibilidad de explorar nuevos indicadores a incorporar en la toma de decisiones del algoritmo y siempre se analizará desde un punto de vista crítico la calidad del conjunto de características a utilizar.
- **Falta de tiempo en la implementación del producto completo:** Debido a la gran variedad de técnicas que se pueden explorar y a la complejidad de desarrollar productos que obtengan un buen rendimiento, es probable que en etapas más avanzadas del proyecto se determine que se necesita más tiempo para acabar el trabajo. La solución que se propone consiste en el aumento de la dedicación diaria al proyecto.

- **Rendimiento insuficiente de los algoritmos de detección:** Debido a que la detección de las regiones de interés de la cara del conductor supone un pilar fundamental para el buen rendimiento del resto del sistema, resultaría bastante problemático que los algoritmos de detección mostraran un comportamiento poco robusto. Si esta falta de rendimiento llega a bloquear el desarrollo del proyecto, se tendrá que aumentar significativamente la dedicación a esta parte del trabajo.

Como medida para responder ante los desvíos temporales que han surgido durante la realización del trabajo, se han llevado a cabo nuevas estimaciones temporales sobre las tareas que quedaban por llevar a cabo para ajustar su realización a la situación real del proyecto. De esta manera, se ha podido tener una visión más realista del tiempo que se le podía dedicar a cada tarea y se ha podido responder de manera más efectiva antes los desvíos temporales.

2.5 Desviaciones respecto a la planificación inicial

Respecto a la planificación temporal del proyecto, cabe decir que a grandes rasgos no han ocurrido cambios en la estructura de los *sprints* en los que se divide el proyecto. No obstante, como es natural, han habido desviaciones en la duración de estos, y no todos se han completado en el mismo tiempo.

Como es lógico y como se había presentado en la planificación inicial, el primer *sprint* ha requerido más tiempo que el resto, aunque cabe destacar que el tiempo dedicado ha sido superior al esperado, puesto que las tareas de planificación y diseño han llevado más tiempo del que se esperaba inicialmente.

No obstante, esto se ha visto compensado conforme los *sprints* avanzaban, puesto que a medida que el proyecto se acercaba al final, los *sprints* requerían menos tiempo de planificación y diseño, que ya habían sido invertidos en los productos iniciales, y requerían más tiempo de implementación y validación para incorporar mejoras a los productos ya funcionales.

Cabe destacar también, que los *sprints* intermedios, dedicados principalmente a la introducción de métodos de extracción de características de nuevas regiones de interés, han requerido menos tiempo del previsto, puesto que se ha llevado a cabo una metodología muy similar para las distintas regiones, que ha permitido obtener nuevos productos funcionales en un tiempo menor del esperado inicialmente.

Respecto a las tareas de gestión del proyecto, cabe decir que no se ha invertido tanto tiempo en las reuniones como se esperaba, mientras que la documentación de la memoria ha requerido más dedicación de la que se estimaba inicialmente.

Teniendo en cuenta todo esto y analizando el tiempo invertido en cada tarea, se calcula que el tiempo final dedicado al proyecto ha sido de 471 horas, superando el tiempo estimado inicialmente por las razones que se han expuesto en este apartado.

| Código | Tarea | Tiempo | Dependencias | Recursos |
|-------------|------------------------------|-------------|----------------------|---|
| GP | Gestión del Proyecto | 140h | - | - |
| GP_1 | Contexto y Alcance | 15h | EP_1 | Ordenador, <i>Overleaf</i> |
| GP_2 | Planificación temporal | 10h | GP_1 | Ordenador, <i>Overleaf</i> , <i>Gantter</i> |
| GP_3 | Presupuesto y Sostenibilidad | 10h | GP_2 | Ordenador, <i>Overleaf</i> |
| GP_4 | Reuniones | 15h | - | Ordenador |
| GP_5 | Documentación | 75h | - | Ordenador, <i>Overleaf</i> |
| GP_6 | Presentación | 10h | GP_5, PC_7_3, PC_7_4 | Ordenador |
| EP | Estudio Previo | 15h | - | - |
| EP_1 | Estudio del estado del arte | 15h | - | Ordenador |
| MVP | Desarrollo de un MVP | 80h | GP_1 | - |
| MVP_1 | Planificación | 25h | GP_1 | Ordenador |
| MVP_2 | Diseño | 15h | MVP_1 | Ordenador |
| MVP_3 | Implementación | 30h | MVP_2 | Ordenador, <i>Python</i> |
| MVP_4 | Validación | 10h | MVP_2 | Ordenador |
| PC | Producto completo | 236h | MVP | - |
| PC_2 | Sprint 2 | 50h | MVP | - |
| PC_2_1 | Planificación | 10h | MVP_3, MVP_4 | Ordenador |
| PC_2_2 | Diseño | 8h | PC_2_1 | Ordenador |
| PC_2_3 | Implementación | 22h | PC_2_2 | Ordenador, <i>Python</i> |
| PC_2_4 | Validación | 10h | PC_2_2 | Ordenador |
| PC_3 | Sprint 3 | 40h | PC_2 | - |
| PC_3_1 | Planificación | 5h | PC_2_3, PC_2_4 | Ordenador |
| PC_3_2 | Diseño | 5h | PC_3_1 | Ordenador |
| PC_3_3 | Implementación | 20h | PC_3_2 | Ordenador, <i>Python</i> |
| PC_3_4 | Validación | 10h | PC_3_2 | Ordenador |
| PC_4 | Sprint 4 | 28h | PC_3 | - |
| PC_4_1 | Planificación | 5h | PC_3_3, PC_3_4 | Ordenador |
| PC_4_2 | Diseño | 2h | PC_4_1 | Ordenador |
| PC_4_3 | Implementación | 13h | PC_4_2 | Ordenador, <i>Python</i> |
| PC_4_4 | Validación | 8h | PC_4_2 | Ordenador |
| PC_5 | Sprint 5 | 24h | PC_4 | - |
| PC_5_1 | Planificación | 5h | PC_4_3, PC_4_4 | Ordenador |
| PC_5_2 | Diseño | 2h | PC_5_1 | Ordenador |
| PC_5_3 | Implementación | 9h | PC_5_2 | Ordenador, <i>Python</i> |
| PC_5_4 | Validación | 8h | PC_5_2 | Ordenador |
| PC_6 | Sprint 6 | 44h | PC_5 | - |
| PC_6_1 | Planificación | 5h | PC_5_3, PC_5_4 | Ordenador |
| PC_6_2 | Diseño | 5h | PC_6_1 | Ordenador |
| PC_6_3 | Implementación | 24h | PC_6_2 | Ordenador, <i>Python</i> |
| PC_6_4 | Validación | 10h | PC_6_2 | Ordenador |
| PC_7 | Sprint 7 | 50h | PC_6 | - |
| PC_7_1 | Planificación | 5h | PC_6_3, PC_6_4 | Ordenador |
| PC_7_2 | Diseño | 5h | PC_7_1 | Ordenador |
| PC_7_3 | Implementación | 26h | PC_7_2 | Ordenador, <i>Python</i> |
| PC_7_4 | Validación | 14h | PC_7_2 | Ordenador |
| - | Total | 471h | - | - |

Tabla 2: Tiempo real dedicado a las tareas.

Fuente: Elaboración propia.

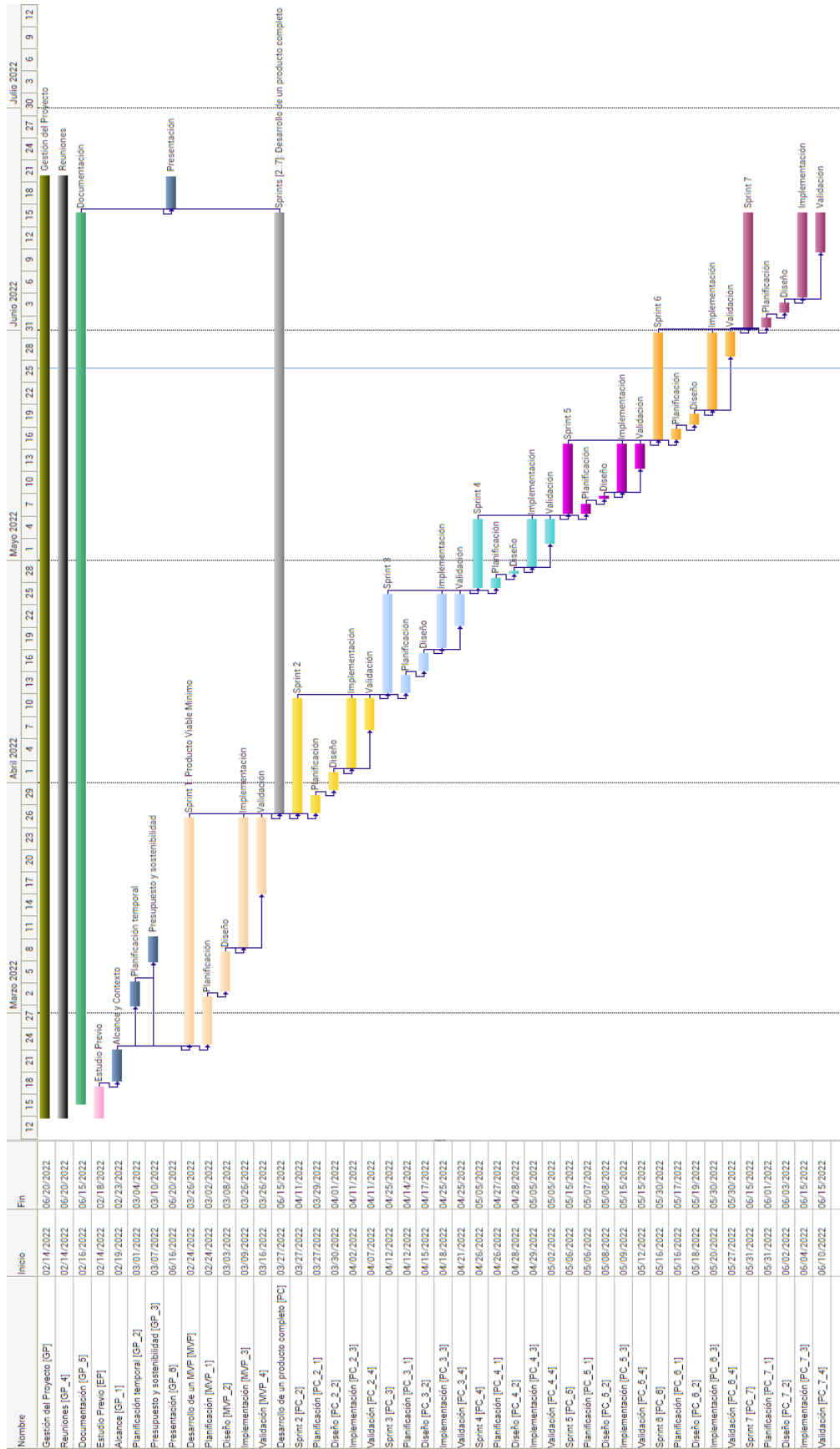


Figura 6: Diagrama de Gantt de la dedicación temporal al proyecto.
Fuente: Elaboración propia mediante *Gantter*

Conocimientos previos

3.1 Características de Haar

Las características de Haar[6] son ventanas rectangulares utilizadas para comparar la oscuridad de una región de píxeles con respecto a otra y así poder detectar contornos y líneas en las imágenes. Como se puede ver en la figura 7, existen tres tipos básicos de características de Haar:

- **Edge features:** son las ventanas que detectan contornos, es decir, regiones en las que se observa un contraste alto en el brillo de los píxeles y existe un paso de una región clara a una oscura o viceversa. Un ejemplo relacionado con la detección de las facciones de la cara podría ser la detección del contorno entre las cejas, que suelen ser oscuras, y la frente, que suele reflejar la luz y mostrarse más clara.
- **Line features:** son las ventanas que detectan líneas, es decir, regiones en las que se observa un patrón oscuro-claro-oscuro o claro-oscuro-claro. Una de sus aplicaciones es la detección de los ojos, puesto que permite distinguir una región oscura, el iris, entre dos regiones más claras.
- **Four-rectangle features:** son ventanas que detectan líneas diagonales en la imagen. Pueden utilizarse para detectar contornos de la cara o de la mandíbula, arrugas, etc. Se trata de ventanas más especializadas.

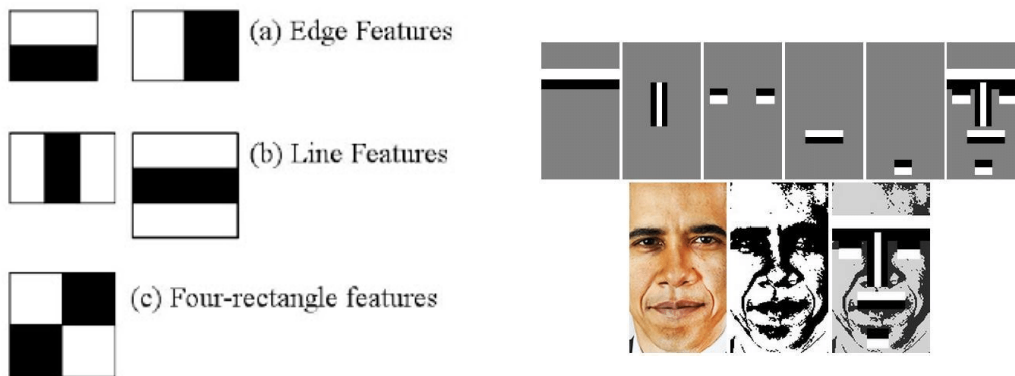


Figura 7: Características de Haar.

Fuente: [34][23]

3.2 Algoritmo de Viola-Jones

El algoritmo de Viola-Jones[44] consiste en un rápido y robusto sistema de detección de objetos en tiempo real, que puede ser aplicado para la detección de caras. Se trata de un algoritmo que hace uso de un conjunto de clasificadores en cascada.

En esencia, los clasificadores consisten en algoritmos, que, utilizando las características de Haar, aprenden a determinar si una imagen contiene el objeto que se quiere detectar o no.

Como se muestra en la figura 8, para llevar a cabo la detección de objetos, el algoritmo de Viola-Jones ejecuta un conjunto de clasificadores simples en cascada sobre subregiones de la imagen a analizar. Es decir, si el resultado de uno de los clasificadores es negativo, se determina que no hay objetos en esa subregión, mientras que si el resultado es positivo, se ejecuta el siguiente clasificador con la misma imagen como entrada.

Así, el algoritmo detectará un objeto si todos los clasificadores determinan que la subregión a analizar contiene dicho objeto.

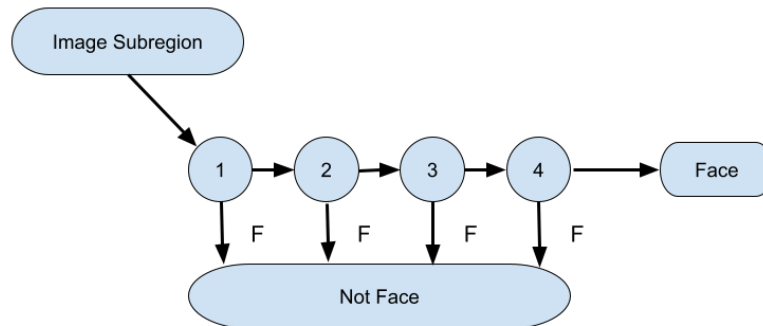


Figura 8: *Cascade of classifiers* utilizado por el algoritmo de Viola-Jones.

Fuente: [4]

De esta manera, la combinación de muchos clasificadores simples da lugar a un algoritmo de detección de objetos muy robusto.

3.3 Magnificación Euleriana de Vídeos

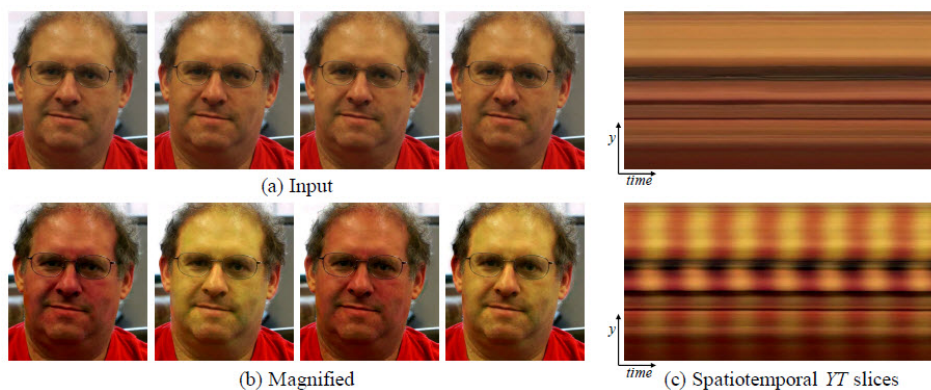


Figura 9: Resultados de aplicar la magnificación euleriana para visualizar el pulso.

Fuente: [19]

Consiste en una técnica que trata de amplificar las variaciones de color en un vídeo o secuencia de imágenes para revelar los cambios que resultan imperceptibles al ojo humano[19].

Como se puede ver en la figura 9, puede resultar de mucha utilidad para hacer posible la obtención de variables como el pulso de una persona mediante técnicas de Visión por Computador.

3.4 Modelos de *Machine Learning* utilizados

3.4.1 Árboles de decisión

Un árbol de decisión[38] es un modelo predictor en forma de diagrama de flujo construido a partir de un conjunto de datos de entrenamiento, donde cada nodo interno representa una decisión en función de los valores de una de las variables predictoras, y las hojas contienen las respuestas que proporciona el modelo para cada uno de los posibles escenarios que modela el árbol.

Para dar respuesta a una nueva observación, se recorre el árbol desde su raíz, de acuerdo con las decisiones que se toman en cada uno de los nodos internos. La hoja en la que se termina el recorrido contiene el valor predicho para la observación.

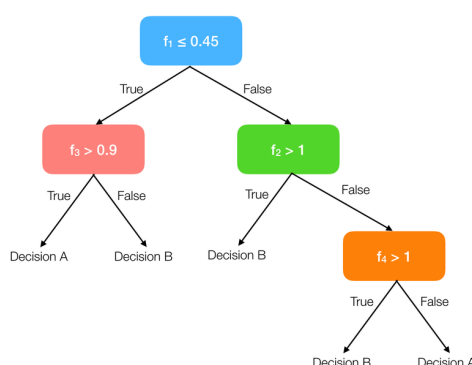


Figura 10: Árbol de decisión de 4 variables.

Fuente: [29]

Estos modelos son muy inestables y producen árboles muy distintos dependiendo del conjunto de datos de entrenamiento.

3.4.2 Regresión Logística

La regresión logística[36] consiste en un tipo de análisis de regresión que modela la probabilidad de que una observación pertenezca a una clase en función de un conjunto de variables predictoras. La predicción se obtiene a partir de la clase para la que se modela una mayor probabilidad y dichas probabilidades se obtienen a partir de la aplicación de una **función logística**.

La función logística es una función sigmoide, cuyo dominio es el conjunto de los números reales y cuyo rango es el intervalo $[0, 1]$.

$$\sigma(t) = \frac{1}{1 + e^{-t}} \quad (1)$$

Donde t se trata de una combinación lineal de todas las variables predictoras disponibles.

$$t = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n \quad (2)$$

Donde x_i representa la variable predictora i , n representa el número de variables predictoras y β_i representa el coeficiente asignado a la variable i .

Teniendo todo esto en cuenta, la regresión logística consiste en la estimación de los valores de los coeficientes que minimizan el error en la clasificación de las observaciones.

El error de las predicciones se modela mediante la *logistic loss*. Dada una observación x_k y la clase a la que pertenece y_k , $p_k = \sigma(x_k)$ corresponde a la probabilidad de que $y_k = 1$, mientras que $1 - p_k$ corresponde a la probabilidad de que $y_k = 0$.

La *logistic loss* para una observación k se define como:

$$\begin{cases} -\ln(p_k) & \text{si } y_k = 1 \\ -\ln(1 - p_k) & \text{si } y_k = 0 \end{cases} \quad (3)$$

3.4.3 K-Nearest Neighbors

El algoritmo de K-nearest neighbors [12] consiste en un modelo de *Machine Learning* muy simple que hace uso de un conjunto de datos de entrenamiento para realizar predicciones sobre nuevas observaciones. Los modelos de KNN son normalmente utilizados para resolver problemas de clasificación.

El algoritmo computa la distancia de la observación que se quiere predecir a todas las instancias presentes en el conjunto de datos de entrenamiento, y determina la respuesta a partir del número de apariciones pertenecientes a cada posible clase en las k observaciones más cercanas. A la observación inicial se le asigna la clase que ha tenido más apariciones en el conjunto de las k observaciones más cercanas.

En el ejemplo de la figura 11, la nueva observación será clasificada como un triángulo, mientras que si $k=5$, la observación sería clasificada como cuadrado.

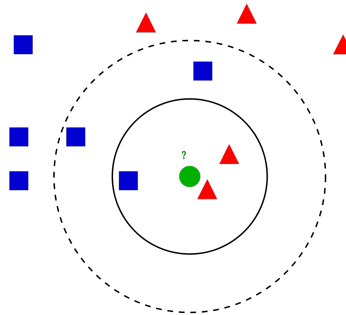


Figura 11: Funcionamiento del algoritmo de KNN con $k=3$.

Fuente: [9]

Funciones utilizadas para calcular la distancia

A continuación se presentan algunas de las funciones más comunes para determinar la distancia entre las observaciones.

- **Distancia Euclidiana** — Se trata de la función más comúnmente utilizada para medir la distancia. Geométricamente, se traza una línea recta entre las dos observaciones valoradas.

$$d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2} \quad (4)$$

- **Distancia Manhattan** — Se trata de otra función muy común, que define la distancia como la suma de las diferencias absolutas entre las coordenadas de las observaciones valoradas.

$$d(x, y) = \sum_{i=1}^m |x_i - y_i| \quad (5)$$

3.4.4 Linear Support Vector Machines (Linear SVM)

Una máquina de vector de soporte lineal [11] o *Linear Support Vector Machine* es un modelo que tiene como objetivo encontrar un hiperplano que separe las observaciones en dos conjuntos. En un problema de clasificación binaria, cada conjunto resultante se corresponde con una de las clases.

Para conseguir esto, se trata el problema como uno de optimización y se busca el hiperplano que maximiza el margen entre los dos conjuntos de observaciones.

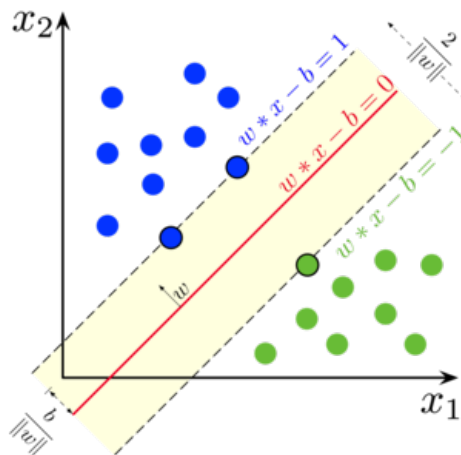


Figura 12: Hiperplano que maximiza el margen entre las observaciones

Fuente: [10]

3.4.5 Gradient Boosting

El *gradient boosting*[32] consiste en una técnica de *Machine Learning* utilizada para resolver problemas de clasificación y regresión. La idea detrás de su funcionamiento se basa en la combinación de clasificadores débiles para obtener un modelo capaz de modelar mejor el problema.

Los clasificadores débiles utilizados son normalmente árboles de decisión.

La idea detrás del *Gradient Boosting* consiste en entrenar un conjunto de modelos de manera iterativa, donde un modelo $i+1$ trata de corregir los errores de su predecesor i .

3.5 Evaluación de clasificadores binarios

En un problema de clasificación binaria, existen cuatro combinaciones posibles para las categorías reales y para las categorías predichas.

- **Verdaderos positivos (tp)** — Se trata de asignaciones a la clase positiva correctas.
- **Verdaderos negativos (tn)** — Asignaciones a la clase negativa correctas.

- **Falsos positivos (fp)** – Asignaciones a la clase positiva incorrectas.
- **Falsos negativos (fn)** – Asignaciones a la clase negativa incorrectas.

Estas combinaciones se pueden presentar de manera compacta en una tabla, que se conoce como matriz de confusión.

| | Valor real positivo | Valor real negativo |
|----------------------------|----------------------------|----------------------------|
| Predicción positiva | Verdadero positivo | Falso positivo |
| Predicción negativa | Falso negativo | Verdadero negativo |

Tabla 3: Matriz de confusión para un problema de clasificación binario

Fuente: Elaboración propia

A partir de la matriz de confusión se pueden extraer las siguientes métricas, que proporcionan información acerca de la calidad de la clasificación.

- **Exactitud** – Representa el porcentaje de aciertos del sistema respecto a todas las observaciones del conjunto de datos.

$$exactitud = \frac{tp + tn}{tp + tn + fp + fn} \quad (6)$$

- **Precisión** – Representa la fracción de las asignaciones positivas correctas respecto a todas las asignaciones positivas.

$$precision = \frac{tp}{tp + fp} \quad (7)$$

- **Exhaustividad** – Representa la fracción de asignaciones positivas respecto a todas las observaciones positivas reales.

$$exhaustividad = \frac{tp}{tp + fn} \quad (8)$$

- **F1-score** – Representa la media armónica de las métricas de precisión y exactitud. Permite realizar una valoración general del rendimiento de estas dos métricas.

$$f1_score = \frac{2 \cdot precision \cdot exhaustividad}{precision + exhaustividad} \quad (9)$$

Solución propuesta

En este apartado se pretende describir las diferentes etapas que conforman la solución propuesta para resolver el problema planteado en la sección 1.1.2, así como las decisiones que se han tomado y las alternativas que se han valorado durante el desarrollo del proyecto.

4.1 Regiones de interés seleccionadas

A continuación se describen las regiones de interés que el sistema tiene en cuenta durante el proceso de extracción de características.

- **Ojos** — Los ojos son, sin duda alguna, una de las regiones de la cara que más información proporciona acerca del cansancio y estado de una persona. La cantidad de características e información que se puede extraer de ellos es bastante elevada y supone una base fundamental para el correcto funcionamiento del sistema.
- **Boca** — Otra región de la cara que puede aportar mucha información sobre el comportamiento y estado del conductor es la boca, puesto que comportamientos como los bostezos resultan muy buenos indicadores de la fatiga en una persona.
- **Cabeza** — Finalmente, también se tiene en cuenta la cabeza del conductor y, más específicamente, la postura de la misma, puesto que se pueden extraer indicadores de gran valor como los cabeceos para añadir una dimensión más al modelaje del estado de la persona.

4.2 Características seleccionadas

Como ya se ha presentado en la sección 1.2, este proyecto plantea llevar a cabo una extracción explícita de un conjunto limitado de características de interés para poder realizar predicciones acerca de la somnolencia de una persona.

Por lo tanto, la elección de este conjunto de características supone un pilar fundamental para el correcto desarrollo del proyecto, puesto que son los indicadores que modelarán el comportamiento y el estado de la persona a analizar.

Dentro del conjunto de indicadores seleccionados, se pueden distinguir dos grandes grupos, dependiendo de la dimensión temporal que representan.

4.2.1 Características relativas a una imagen estática

Este primer grupo abarca las características que describen el estado de la persona en un único *frame* de la secuencia de imágenes a analizar.

El sistema obtiene unas mediciones que posteriormente utiliza para determinar el estado de las regiones de interés.

Relación de aspecto del ojo

Un indicador muy útil para extraer información acerca de los ojos de una persona es la relación de aspecto de los mismos.

A partir de ahora se define la relación de aspecto de los ojos como EAR (Eye Aspect Ratio).

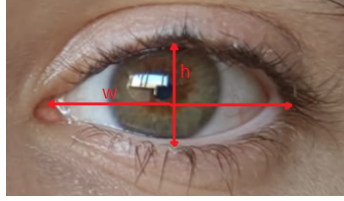


Figura 13: Relación de aspecto del ojo

Fuente: Elaboración propia

La relación de aspecto del ojo representa el grado de apertura de este, y se obtiene dividiendo la altura del ojo entre su anchura.

Matemáticamente, el EAR queda definido de la siguiente forma:

$$EAR = \frac{h}{w}$$

A partir del EAR se puede computar el **estado del ojo**, que presenta dos posibles valores: **abierto** o **cerrado**.

Relación de aspecto de la boca

Análogamente a la relación de aspecto de los ojos, un indicador muy útil para extraer información acerca del estado de la boca es la relación de aspecto de la misma.

A partir de ahora se define la relación de aspecto de la boca como MAR (Mouth Aspect Ratio).

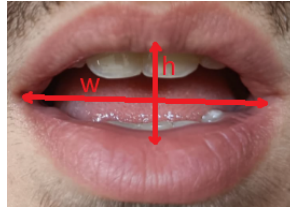


Figura 14: Relación de aspecto de la boca

Fuente: Elaboración propia

La relación de aspecto de la boca representa el grado de apertura de esta, y, de manera análoga al EAR, se obtiene dividiendo la altura de esta entre su anchura.

Matemáticamente, el MAR queda definido de la siguiente forma:

$$MAR = \frac{h}{w}$$

A partir del MAR se puede computar el **estado de la boca**, que presenta dos posibles valores: **abierta** o **cerrada**.

Ángulos de navegación

Los ángulos de navegación, también conocidos como ángulos de **Tait-Bryan**[31], son unos indicadores utilizados para describir la orientación de un objeto en tres dimensiones.

Estos consisten en un sistema de coordenadas angulares, definidas de forma similar a los ángulos de Euler, pero que utilizan el corte entre dos planos no homólogos como línea de nodos. Los ángulos de navegación son utilizados normalmente para definir la posición de aeronaves, satélites o embarcaciones, pero pueden ser utilizados para definir la orientación de la cabeza del conductor, como se muestra en la figura 15.

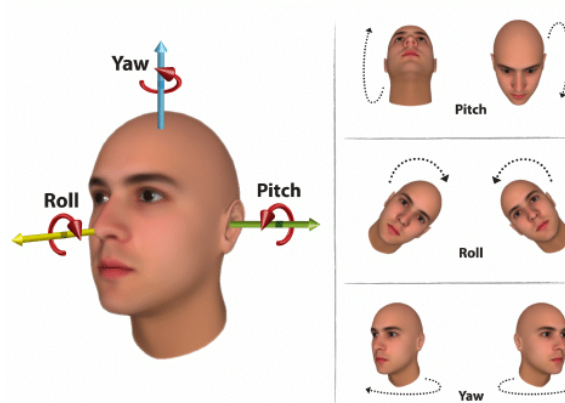


Figura 15: Ángulos de navegación de la cabeza

Fuente: [43]

- **Yaw** – Representa la orientación horizontal de la cabeza del conductor. Resulta de mucha utilidad para detectar variaciones en la orientación del conductor, que influirán en la extracción de las características, y realizar correcciones para evitar detecciones erróneas.
- **Pitch** – Representa la orientación vertical de la cabeza del conductor. Resulta de mucha utilidad para detectar la inclinación de la cabeza y los cabeceos causados por la somnolencia.
- **Roll** – Representa la orientación en el eje Z de la cabeza del conductor.

Para los objetivos de este proyecto se han definido dos posibles **estados para la inclinación vertical de la cabeza**, dependiendo de la medición del ángulo de *pitch* extraída de la imagen.

- **Inclinada hacia abajo** – Representa los escenarios en que la cabeza se encuentra visiblemente inclinada hacia abajo. Es el estado a partir del que se detectan los cabeceos derivados del cansancio.
- **Posición neutral o inclinada hacia arriba** – Puesto que únicamente la inclinación hacia abajo de la cabeza proporciona información relevante para la detección de los cabeceos, se ha decidido definir el resto de inclinaciones como un único estado.

4.2.2 Características relativas a una secuencia temporal

En este segundo grupo se describen las características que tienen en cuenta información del pasado y proporcionan un modelaje temporal del estado y comportamiento de la persona a analizar.

Parpadeos

Los parpadeos representan intervalos de tiempo muy breves donde los ojos se encuentran cerrados. Estos finalizan cuando los ojos vuelven a abrirse.

A partir de las detecciones de parpadeos y del estado del ojo, se computan los siguientes indicadores.

- **Frecuencia de parpadeo** — Consiste en la cantidad de parpadeos que se han detectado en un intervalo de tiempo. Normalmente, el intervalo de tiempo que se suele utilizar para medir este indicador es de un minuto.

J.A. Stern *et al.* llevaron a cabo una revisión de un conjunto de experimentos para determinar la relación entre la frecuencia de parpadeo y el tiempo de realización de tareas (TOT) en pilotos y controladores de tráfico aéreo [22]. En las conclusiones del artículo se expone que, efectivamente, la frecuencia de parpadeo se suele ver incrementada cuando el TOT aumenta.

Teniendo esto en cuenta y, pese a que las variaciones en los valores de esta variable pueden ser dependientes de cada sujeto, este indicador, junto con otras variables, puede resultar en un buen discriminador del estado de fatiga de los sujetos a analizar.

$$frecuencia_parpadeo = \frac{\text{Número de parpadeos}}{\text{Número de frames transcurridos}}$$

- **PERCLOS** — El nombre de esta variable consiste en un acrónimo de **Percentage of Eye Closure**, y representa el porcentaje del tiempo que los ojos se encuentran cerrados al menos en un 80 %.

Esta característica es fundamental para la detección del cansancio mediante el uso de técnicas de Visión por Computador y proporciona más información sobre la naturaleza de los parpadeos de los individuos analizados.

D.Dinges y R.grace llevaron a cabo una evaluación de distintas medidas de detección del cansancio en escenarios controlados de laboratorio y llegaron a la conclusión de que el PERCLOS era la medida más fiable para determinar el nivel de alerta de los sujetos [13].

Teniendo en cuenta estos resultados, resulta interesante plantear utilizar esta variable para el modelaje de los niveles de alerta de los conductores a analizar.

$$PERCLOS = \frac{\text{Número de frames ojos cerrados}}{\text{Número de frames transcurridos}}$$

- **Tiempo medio de parpadeo** — Consiste en la cantidad de tiempo media que suele durar un parpadeo. De manera similar a las dos variables que ya han sido presentadas, este indicador proporciona una perspectiva diferente para analizar la naturaleza de los parpadeos de los sujetos.

Más allá de contabilizar el número de parpadeos, esta variable representa el tiempo medio que suelen durar estos, o, dicho de otra manera, su velocidad media.

Michael Ingre *et al.* llevaron a cabo un experimento para evaluar la correlación entre la duración de los parpadeos y los desvíos en la dirección del vehículo sobre un simulador de conducción [21]. Llegaron a la conclusión de que existe una alta correlación entre el cansancio de los conductores y los valores de ambos indicadores.

Teniendo esto en cuenta, esta característica puede resultar un indicador útil para modelar el comportamiento de los conductores desde una dimensión distinta a las que se han propuesto anteriormente.

$$tiempo_medio_parpadeo = \frac{\text{Número de frames ojos cerrados} \cdot \text{frames por segundo}}{\text{Número de parpadeos}}$$

- **Tiempo que los ojos llevan cerrados** — Todas las características que se han visto hasta el momento representan diferentes propiedades de los parpadeos del individuo a analizar, pero lo hacen en una escala temporal relativamente grande y, por lo tanto, un cambio relevante en un intervalo de tiempo pequeño puede ser negligido. Un sistema incapaz de modelar estos cambios rápidamente es inaceptable para los objetivos del proyecto, puesto que los accidentes de tráfico pueden ocurrir en cuestión de segundos.

Es por esto que resulta de gran utilidad contar con una variable capaz de modelar con un tiempo rápido de respuesta si el individuo padece síntomas de fatiga.

Este indicador representa el tiempo que los ojos llevan cerrados desde la última vez que estuvieron abiertos, o dicho de otra manera, desde que empezó el parpadeo. Los peligros que acarrea un conductor con los ojos cerrados por más de un segundo son evidentes, por lo que es necesario responder inmediatamente ante estas situaciones, y este indicador es capaz de modelarlas.

Por lo tanto, se podría afirmar que la inclusión de esta variable en un modelo predictor dotaría a dicho modelo de una visión temporal local, que aportaría mucho valor, puesto que los indicadores presentados previamente únicamente lo dotarían de una visión temporal global.

$$tiempo_ojos_cerrados = \text{número de frames parpadeo actual} \cdot \text{frames por segundo}$$

Bostezos

De manera análoga a los parpadeos, los bostezos se definen como intervalos temporales prolongados donde el MAR presenta valores muy elevados.

Indudablemente, los bostezos prolongados durante un intervalo de tiempo son muy buenos indicadores de la fatiga y, por lo tanto, esta variable resulta valiosa para el análisis del cansancio y proporciona una dimensión más para modelar el estado del individuo a analizar.

A partir de los bostezos, se computa el siguiente indicador.

- **Frecuencia de los bostezos** — De manera análoga a la frecuencia de parpadeo, este indicador representa la cantidad de bostezos que se han detectado en un intervalo de tiempo. De nuevo, este intervalo de tiempo suele ser de un minuto.

$$frecuencia_bostezos = \frac{\text{número de bostezos}}{\text{número de frames transcurridos}}$$

Cabeceos

Los cabeceos representan intervalos prolongados donde el ángulo de *pitch* es muy bajo y, por lo tanto, la cabeza se encuentra inclinada hacia abajo.

De manera análoga a los bostezos, las personas que se encuentran en un estado de alerta raramente dan cabeceos, mientras que es mucho más común detectarlos en personas que padecen síntomas de fatiga. Indudablemente, los cabeceos suponen un indicador muy claro de la fatiga de una persona y resultan de mucho valor para el análisis del cansancio.

A partir de los cabeceos, se computa el siguiente indicador.

- **Frecuencia de cabeceo**

Como ya se ha definido previamente con la frecuencia de parpadeo y de los bostezos, la frecuencia de cabeceo representa la cantidad de cabeceos que se han detectado en un intervalo de tiempo, que suele ser de un minuto.

$$frecuencia_cabeceos = \frac{\text{número de cabeceos}}{\text{número de frames transcurridos}}$$

4.3 Funcionamiento del sistema implementado

El sistema que se ha implementado para solucionar el problema se divide principalmente en cuatro grandes etapas, como se puede ver en la figura 16. Dentro de cada etapa se exponen las técnicas implementadas, así como las alternativas que se han valorado durante la realización del proyecto.



Figura 16: Diagrama de flujo correspondiente al funcionamiento del sistema

Fuente: Elaboración propia

4.4 Procesado de Imagen

En esta etapa del sistema se lleva a cabo un procesado de la imagen para facilitar y mejorar el rendimiento de los procesos de segmentación de las regiones de interés y de extracción de características que se llevarán a cabo en las siguientes etapas del sistema.

Puesto que las librerías que se han utilizado para tratar las imágenes trabajan en distintos espacios de color, el único tratamiento que ha sido necesario llevar a cabo consiste en una transformación del espacio de color BGR a RGB.

Otras valoraciones

Algunos de los vídeos sobre los que se ha trabajado durante el desarrollo del proyecto presentan una resolución muy baja y una iluminación muy pobre. A causa de esto, algunas regiones de interés presentan un contraste muy alto con el resto de la cara y su detección se vuelve más compleja.

Para tratar de solucionar este problema se han intentado aplicar técnicas de ecualización de histograma adaptativo, pero no se ha visto una mejora significativa en la calidad de detección y el rendimiento del procesado de imagen se veía reducido en algunos escenarios.

Por lo tanto, se ha descartado la idea por falta de tiempo para explorarla más a fondo, pese a que resulta una vía de trabajo interesante que volver a valorar en el futuro.

4.5 Segmentación de las regiones de interés

En esta etapa del sistema se lleva a cabo la segmentación de las regiones de interés, una vez se ha realizado el tratamiento sobre la imagen de entrada.

Como se puede ver en la figura 17, esta etapa se divide principalmente en tres procesos.

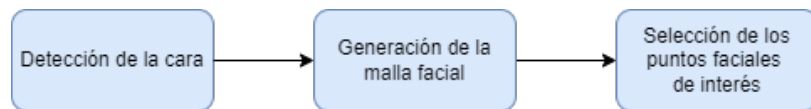


Figura 17: Diagrama de flujo correspondiente a la segmentación de las regiones de interés

Fuente: Elaboración propia

4.5.1 Detección de la cara

Para detectar la cara de los sujetos se ha utilizado un algoritmo basado en el funcionamiento de un *Single Shot Multibox Detector*[26], que presenta una calidad de detección alta junto con un tiempo de predicción bastante reducido.

A grandes rasgos, el algoritmo consiste en una red neuronal convolucional que produce un conjunto de *bounding boxes* junto con una métrica que indica la presencia del objeto que se quiere detectar en dicha *bounding box*. Para realizar las detecciones finales, se lleva a cabo una supresión de no-máximos.

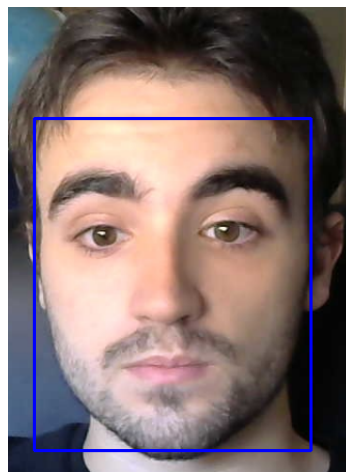


Figura 18: Resultado de aplicar el algoritmo de detección de caras

Fuente: Elaboración propia

A partir de la *bounding box* obtenida, se recorta la cara de la imagen y se trabaja con ella en los siguientes pasos.

4.5.2 Generación de la malla facial

Para llevar a cabo una segmentación de las regiones de interés se ha optado por un enfoque basado en la generación de una malla de puntos faciales 3D a partir de la imagen de la cara de los sujetos.

Esta malla proporciona información acerca de todas las regiones de interés de la cara, y permite realizar una segmentación muy rápida de estas.

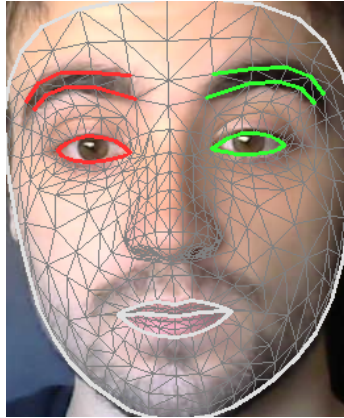


Figura 19: Malla facial generada a partir de la imagen recortada de la cara

Fuente: Elaboración propia

Como se puede ver en la figura 19, la malla facial aporta una cantidad de información inmensa acerca de la cara, y permite centralizar la segmentación de la cara en un único algoritmo, en lugar de un conjunto de modelos para cada una de las regiones a analizar. El módulo utilizado para generar la malla se describe en la sección 5.4.

4.5.3 Selección de los puntos faciales de interés

Puesto que la malla facial proporciona una cantidad inmensa de información, es necesario llevar a cabo una selección de los puntos faciales que son relevantes para la extracción de las características del comportamiento de los conductores.

A continuación se presentan las regiones de interés para la extracción de características y los puntos faciales que se han seleccionado para cada una de ellas.

Ojos

Cabe destacar que se ha decidido trabajar con un único ojo para realizar la extracción de las características. Esto es debido a que pueden haber escenarios donde la cara de los conductores esté muy rotada lateralmente y únicamente se tenga visibilidad sobre uno de los ojos. Por lo tanto, la utilización de un único ojo evita la necesidad de calcular oclusiones para determinar si estos son visibles o no, y no resulta en una pérdida de capacidad de detección.

El criterio que se ha seleccionado para la elección del ojo a analizar es la cercanía de este a la cámara, que se puede computar fácilmente gracias a la naturaleza 3D de los puntos faciales, puesto que los ojos ocultos son siempre los que se encuentran más alejados. Se ha determinado el ojo más cercano a partir de la componente Z de los puntos faciales.

Teniendo esto en cuenta, se han seleccionado los extremos horizontales y verticales del ojo más cercano, para poder obtener información acerca de su altura y anchura. Se han escogido los 3 puntos verticales más centrados para obtener mediciones más fiables, como se puede ver en la figura 20.

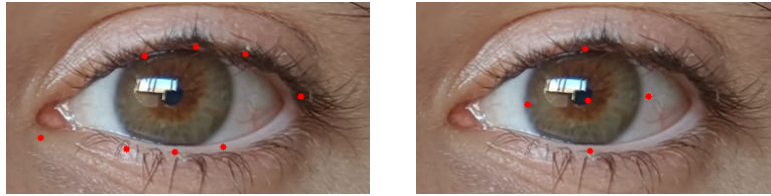


Figura 20: Puntos faciales seleccionados de los ojos

Fuente: Elaboración propia

Por otro lado, también se han seleccionado los puntos faciales referentes al iris del ojo, ya que resulta una fuente de información muy valiosa.

Boca

Respecto a la región de la boca, se ha realizado un tratamiento análogo al de los puntos faciales del ojo más cercano. Como se puede ver en la figura 21, se han seleccionado los tres puntos verticales más centrados, puesto que son los que muestran una mayor variación de distancia entre el estado abierto y cerrado de la boca.

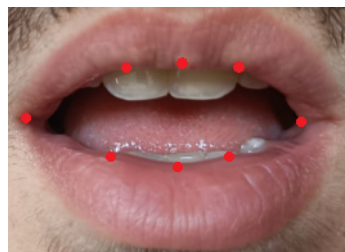


Figura 21: Puntos faciales seleccionados de la boca

Fuente: Elaboración propia

Cabeza

Respecto a la región de la cabeza, puesto que uno de los métodos implementados para la obtención de características, que se presentará en la sección 4.6.1.3, requiere la localización de diferentes puntos faciales de la cara, se han seleccionado los puntos de la malla más cercanos a estas localizaciones. En la figura 22 se pueden ver estos puntos de interés.

Alternativas valoradas

Inicialmente se planteó utilizar el algoritmo de Viola-Jones para llevar a cabo la detección de la cara de los sujetos y de las diferentes regiones de interés, pero este presentaba un comportamiento

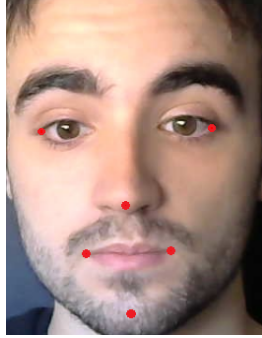


Figura 22: Puntos faciales seleccionados de la cara

Fuente: Elaboración propia

inestable, y la calidad de las detecciones no era lo suficientemente precisa como para poder llevar a cabo una extracción de características robusta.

Cabe destacar que el detector del ojo utilizado inicialmente, también basado en el algoritmo de Viola-Jones, no era capaz de localizar los ojos cuando estos estaban cerrados, como se puede ver en la figura 23. Esto suponía un gran problema para la extracción de las características, puesto que este estado aporta información fundamental para poder describir el comportamiento de los sujetos.

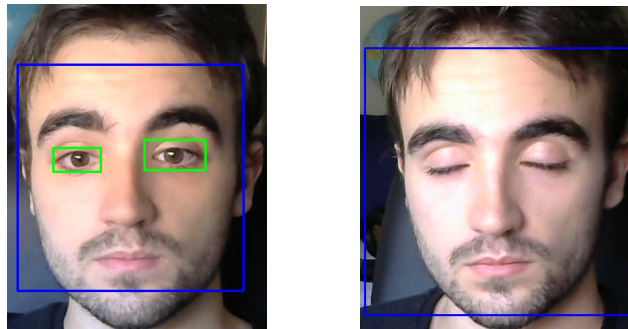


Figura 23: Comparación del funcionamiento de Viola-Jones en función del estado de los ojos.

Fuente: Elaboración propia

A causa de estos problemas y teniendo en cuenta que la generación de los puntos faciales centraliza la segmentación de las regiones de interés en un único modelo, se decidió descartar esta opción.

4.6 Extracción de características

Una vez se ha llevado a cabo el procesado de la imagen y la segmentación de las regiones de interés del sujeto a analizar, el siguiente paso a realizar es la extracción de las características del *frame* actual de la secuencia, y la actualización de los indicadores del comportamiento que modelan la secuencia temporal. En la figura 24 se presenta el diagrama de flujo correspondiente a esta etapa.

Cabe destacar que esta es la etapa más importante del sistema implementado, puesto que las características obtenidas modelan el problema que se quiere resolver.

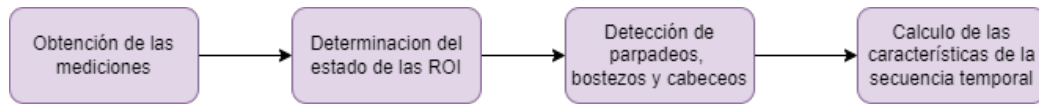


Figura 24: Diagrama de flujo correspondiente a la obtención de características del *frame* actual

Fuente: Elaboración propia

4.6.1 Obtención de las mediciones

En esta fase se obtienen mediciones que aportan información acerca de la fisionomía de las diferentes regiones de interés.

4.6.1.1 Mediciones del ojo

EAR euclidiano

Se ha seleccionado el EAR del ojo de la persona mediante distancias euclidianas para determinar el estado de este. Los motivos por los que se ha decidido utilizar este método se detallan en los experimentos que serán presentados en la sección 6.1.1.

Como se puede ver en la figura 25, el EAR es computado realizando una media de las distancias que presentan los diferentes pares de puntos verticales y dividiéndola entre la distancia que presentan los extremos horizontales del ojo.

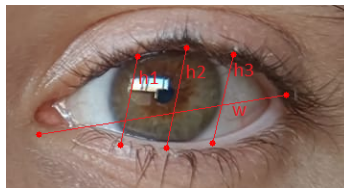


Figura 25: Cálculo del EAR euclidiano

Fuente: Elaboración propia

Matemáticamente, queda definido de la siguiente forma:

$$EAR = \frac{h1 + h2 + h3}{3w} \quad (10)$$

- **Ventajas** — El método es invariante a los cambios en el ángulo de *Roll*, puesto que las distancias que se computan son euclidianas.
- **Desventajas** — Las mediciones se ven afectadas si los pares de puntos a analizar no se encuentran alineados en la coordenada *x*. Debido a esto, las mediciones pueden tener algo de ruido y no siempre representar la realidad de manera fiel.

Otras mediciones valoradas

- **EAR no euclidiano** — Como contraparte al EAR euclidiano, en un inicio se valoró utilizar distancias no euclidianas para determinar el estado del ojo. La distancia entre cada par de puntos verticales se definía como la diferencia entre sus coordenadas *y*, mientras que para los puntos horizontales se aplicaba el mismo criterio con sus coordenadas *x*.

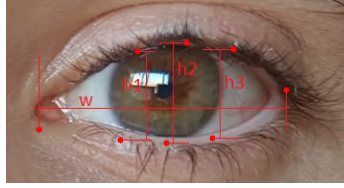


Figura 26: Cálculo del EAR no euclidiano

Fuente: Elaboración propia

Matemáticamente, quedaba definido de la siguiente forma:

$$EAR = \frac{h1 + h2 + h3}{3w} \quad (11)$$

- **Ventajas** — Las mediciones son muy cercanas a la realidad, puesto que únicamente se tienen en cuenta las coordenadas relevantes para la medición de ambas magnitudes. Se elimina el ruido de la coordenada x para la medición de la altura, y el de la y para la anchura.
- **Desventajas** — A diferencia del EAR euclidiano, el método no es invariante a posiciones de la cabeza con un ángulo de *Roll* alejado de 0. Esto es debido a que conforme el *Roll* se aleja del 0, la diferencia de altura entre los dos párpados va disminuyendo, aunque la distancia real entre estos se mantenga, y, por lo tanto, las mediciones resultantes no son realistas.
- **Apertura del ojo respecto al diámetro del iris** — Se trata una medición alternativa para el grado de apertura del ojo que se define como la altura de este entre el diámetro del iris.

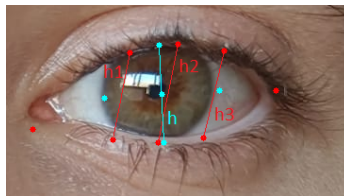


Figura 27: Cálculo de la apertura del ojo respecto al diámetro del iris

Fuente: Elaboración propia

Matemáticamente, queda definida de la siguiente forma:

$$eye_closure = \frac{h1 + h2 + h3}{3h}$$

- **Ventajas** — El método es invariante a los cambios en el ángulo de *Roll*, puesto que las distancias que se computan son euclidianas. También es invariante a los cambios en el ángulo de *Yaw*, puesto que la medición no depende de la anchura del ojo.
- **Desventajas** — De manera similar al EAR euclidiano, las mediciones se ven afectadas si los pares de puntos a analizar no se encuentran alineados en la coordenada x . Debido a esto, las mediciones pueden tener algo de ruido y no siempre representar la realidad de manera fiel. Puesto que el iris no siempre es completamente visible, el diámetro del iris computado puede no estar representado correctamente.

4.6.1.2 Mediciones de la boca

MAR euclidiano

Este método es completamente análogo al que se ha presentado para el EAR en la sección 4.6.1.1. Los motivos que han llevado a la utilización de este método se describen de manera más detallada en la sección 6.1.2.

Otras mediciones valoradas

- **MAR no euclidiano** — De la misma manera, este método es completamente análogo al que se ha presentado para el EAR en la sección 4.6.1.1. Esta medición se ha descartado, puesto que no ofrece la misma robustez a las variaciones en el ángulo de *Roll*.

4.6.1.3 Mediciones de la cabeza

Medición mediante modelo de *deep learning*

Puesto que la detección de los ángulos de navegación resulta crucial para poder llevar a cabo la extracción de las características, ha sido necesario encontrar un método realmente robusto. En la sección 6.1.3 se detallarán los motivos por los que se ha escogido este método con más detalle.

Tras llevar a cabo una búsqueda de diferentes alternativas, se ha optado por aplicar una red neuronal convolucional entrenada específicamente para llevar a cabo esta tarea, capaz de predecir los ángulos de navegación a partir de la imagen recortada de la cara de una persona.

Thorsten Hempel *et al.* describen la arquitectura del modelo de manera muy detallada en [20].

- **Ventajas** — El método es muy robusto y proporciona resultados muy precisos. Además, no proporciona detecciones erróneas de los ángulos cuando la persona mira de frente a la cámara.
- **Desventajas** — El método no utiliza los puntos faciales ya computados y eleva bastante el tiempo de cómputo necesario para completar la extracción de las características.

Otras mediciones valoradas

- **Método geométrico** — En un primer lugar se valoró llevar a cabo un enfoque geométrico para obtener los ángulos de navegación de la cabeza, que hacía uso de los puntos faciales obtenidos en el procesado de la imagen y requería la siguiente información.
 - **Coordenadas 2D de los siguientes puntos faciales:** Se trata de los puntos seleccionados en la sección 4.5.3.
 1. Extremo izquierdo del ojo izquierdo
 2. Extremo derecho del ojo derecho
 3. Extremo izquierdo de la boca
 4. Extremo derecho de la boca
 5. Punta de la nariz
 6. Barbilla
 - **Coordenadas 3D de los mismos puntos** — Para obtener la localización 3D de estos puntos se decidió utilizar un modelo 3D genérico como referencia.

- **Parámetros intrínsecos de la cámara** – Puesto que el sistema que se quiere diseñar no ha de resultar invasivo para el usuario, el producto final no ha de exigir una calibración de la cámara a los conductores. Por lo tanto, se realizó una aproximación de los parámetros intrínsecos de esta.
 - **Centro óptico de la cámara** – Representa el punto situado en el centro de la lente de la cámara. Se puede aproximar al centro de la imagen.
 - **Distancia focal** – Representa la distancia entre el centro óptico de la lente de la cámara y el foco. Se puede aproximar a la anchura de la imagen en píxeles.
 - **Distorsión radial** – Se asume que en las imágenes no hay ninguna distorsión.

A partir de los parámetros descritos, se obtuvieron las matrices de rotación y translación, que transforman los puntos del sistema de coordenadas del mundo real a las coordenadas de la cámara, mediante la resolución de un sistema de ecuaciones no lineal entre los pares de puntos de la imagen y los puntos del mundo real utilizados como parámetros.

Una vez obtenidas las matrices de transformación, se realizó una proyección del punto (0, 0, 1000) del sistema de coordenadas del mundo real sobre el plano del *frame*, y se calculó el vector que lo unía al punto facial que representa la localización de la punta de la nariz.

Finalmente, los ángulos de navegación se obtenían calculando el ángulo entre cada eje y el vector computado.

$$\text{pitch} = \arccos\left(\frac{x}{\sqrt{x^2 + y^2 + z^2}}\right)$$

$$\text{yaw} = \arccos\left(\frac{y}{\sqrt{x^2 + y^2 + z^2}}\right)$$

$$\text{roll} = \arccos\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right)$$

Donde las variables x , y , z representan, respectivamente, el valor de cada componente del vector computado.

- **Ventajas** – El método es rápido de computar y hace uso de los puntos faciales obtenidos.
- **Desventajas** – Los resultados obtenidos son bastante inestables y varían mucho entre dos *frames* consecutivos. Además, el ángulo de *pitch* obtenido cuando la persona mira de frente a la cámara es erróneo. En la figura 28 se presenta este problema.

Estas desventajas han comportado la necesidad de buscar una alternativa y, por lo tanto, este método se ha descartado.

4.6.1.4 Alternativas valoradas

Durante las etapas iniciales de desarrollo del proyecto se planteó la posibilidad de extraer el pulso de los conductores para tener una dimensión más de modelaje de sus síntomas de fatiga.

Se ha valorado principalmente la técnica de magnificación euleriana de vídeos, pero esta ha sido descartada porque requiere de un proceso de optimización de parámetros exhaustivo que es diferente para cada vídeo, y, por lo tanto, la técnica no presenta la suficiente robustez.

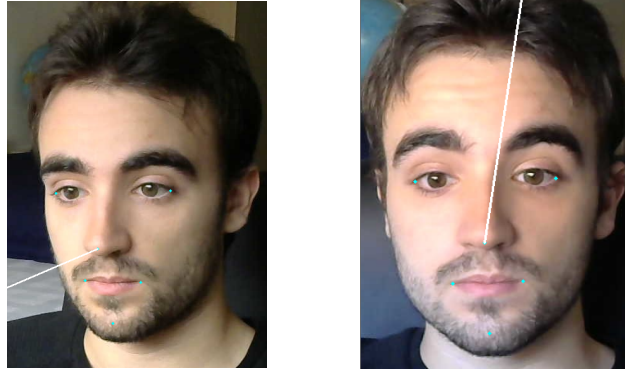


Figura 28: Comparación del método geométrico en función del ángulo de *yaw* de la persona. La línea indica la dirección estimada de la pose de la cabeza.

Fuente: Elaboración propia

Otro factor muy decisivo para no explorar técnicas alternativas ha sido el hecho de que los vídeos del *dataset* que se ha utilizado, descrito con más detalle en la sección 5.5.2, se encuentran en escala de grises, y por lo tanto resulta imposible extraer el pulso a través de las variaciones de color en la imagen.

No obstante, esta característica presenta mucho potencial y presenta una vía de trabajo futuro muy interesante.

4.6.2 Determinación del estado de las regiones de interés

Una vez se han obtenido las medidas, se lleva a cabo un análisis de estas junto con las mediciones obtenidas en el pasado para determinar el estado de las regiones en el *frame* actual.

4.6.2.1 Análisis del estado del ojo

Para entender el funcionamiento del algoritmo que determina el estado del ojo y las decisiones que se han tomado respecto a su diseño, es necesario presentar las diferentes iteraciones que se han implementado junto con sus debilidades.

El algoritmo basa sus respuestas fundamentalmente en la comparación de la medición obtenida del ojo respecto a un *threshold*, cuya definición varía en las diferentes versiones implementadas.

Para valorar el rendimiento de los diferentes algoritmos implementados y valorar si los cambios han introducido mejoras significativas, se ha llevado a cabo un análisis de la *performance* de cada uno de los algoritmos, que se presentará en la sección 6.1.1.

Algoritmo inicial

Inicialmente, el análisis se realizaba mediante un *threshold* estático (Th), pasado como parámetro. Este método es muy simple y no tiene en cuenta la fisonomía del ojo de la persona, de manera que los resultados son irregulares, y la calidad de estos depende mucho del sujeto que se está analizando.

En la ecuación 12 se describe el funcionamiento del algoritmo inicial, basado en el uso de un *threshold* estático. Por otro lado, en la figura 29 se proporciona un ejemplo visual del funcionamiento del algoritmo sobre los valores de EAR obtenidos durante el transcurso de una secuencia de imágenes.

$$f(EAR, Th) = \begin{cases} \text{abierto,} & \text{si } EAR \geq Th \\ \text{cerrado,} & \text{en caso contrario.} \end{cases} \quad (12)$$

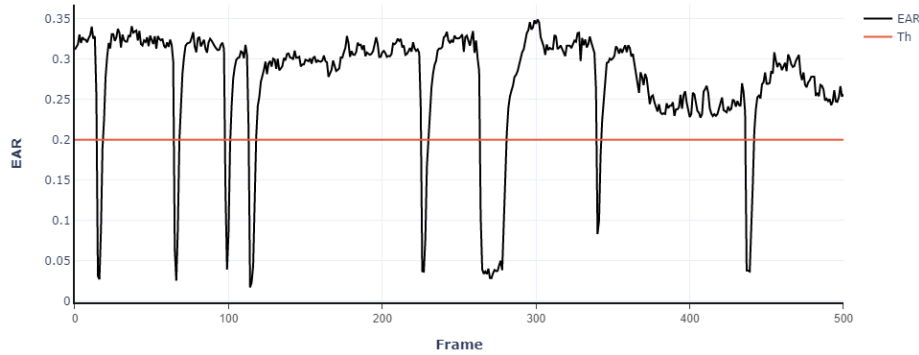


Figura 29: Evolución del EAR durante el transcurso del vídeo

Fuente: Elaboración propia

Implementación de un *threshold* dinámico

Dadas las debilidades que presentaba el sistema inicial, se decidió implementar un sistema que determina el valor del *threshold* utilizado de manera dinámica, capaz de adaptarse a la fisionomía del ojo del sujeto que se está analizando.

Este sistema hace uso de dos nuevos parámetros, que pueden ser modificados por el usuario y que sustituyen el parámetro de *threshold* estático.

- **Número de frames a pasado (n)** – Determina el número de *frames* del pasado **donde el ojo estaba abierto** para los que se almacena el valor del EAR. A partir de estas medidas, se obtiene la media de estas, que representa el valor esperado del EAR cuando el ojo está abierto y que se utilizará como base para calcular el nuevo *threshold*.

$$expected_EAR = \frac{\sum_{i=1}^n EAR_i}{n} \quad (13)$$

Donde EAR_i representa el valor del EAR de i frames en el pasado.

- **Porcentaje del *threshold* (threshold_perc)** – Este parámetro se utiliza para determinar el nuevo valor del *threshold* a partir de la media computada mediante los *frames* del pasado.

$$Th' = threshold_perc \cdot expected_EAR \quad (14)$$

Donde Th' representa el nuevo valor del *threshold* que será utilizado en lugar del parámetro estático utilizado inicialmente.

En la figura 30 se presenta un ejemplo visual de la evolución del *threshold* dinámico en función de los valores del EAR obtenidos durante el transcurso de la misma secuencia de imágenes y del número de *frames* que se tienen en cuenta para calcular el valor esperado del EAR.

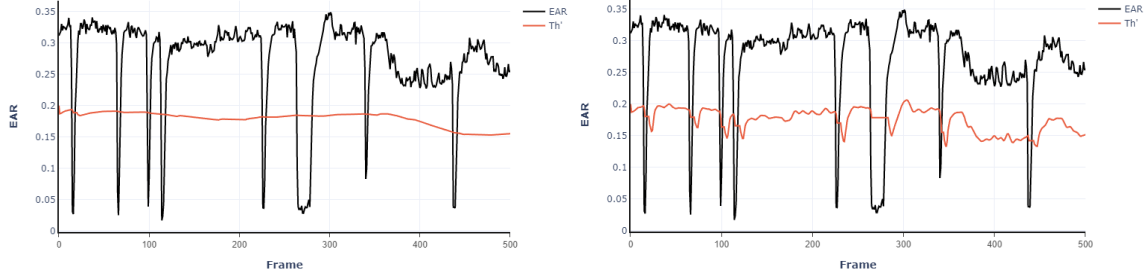


Figura 30: Comportamiento del *threshold* dinámico con $n=100$ y $n=5$, respectivamente.

Fuente: Elaboración propia

Pese a que la introducción de este nuevo cambio ha comportado una mejora significativa de rendimiento y un aumento en la robustez de las detecciones, este sistema presenta una debilidad más. Cuando las mediciones extraídas eran muy cercanas al valor del *threshold*, podía ocurrir una secuencia alternante de detecciones positivas y negativas, causando así muchas detecciones falsas en el algoritmo de detección de parpadeos, que se presentará en la sección 4.6.3.

Esto es debido a que la transición entre los dos estados del ojo es demasiado inmediata.

Implementación de un intervalo de incertidumbre

Para remediar este problema, se decidió introducir un nuevo parámetro, que añade un intervalo de incertidumbre para las mediciones en la función que determina el estado del ojo.

- **Amplitud de incertidumbre (gz)** – Determina el porcentaje de la media computada con los *frames* del pasado que se utilizará para definir una *zona gris* para las mediciones. Este porcentaje permite definir un *threshold* superior y uno inferior, y las mediciones que se encuentren entre estos dos valores no serán concluyentes para determinar el estado del ojo, de manera que este se determinará teniendo en cuenta el estado del *frame* anterior.

$$Th'_u = Th' \cdot (1 + gz) \quad (15)$$

$$Th'_l = Th' \cdot (1 - gz) \quad (16)$$

Donde Th'_u representa el valor del *threshold* superior y Th'_l representa el valor del *threshold* inferior. El intervalo de incertidumbre es, por lo tanto, $[Th'_l, Th'_u]$.

La función que resulta de añadir este nuevo parámetro queda presentada en la ecuación 17.

$$f(EAR, Th_u, Th_l, ultimo_estado) = \begin{cases} \text{abierto,} & \text{si } EAR > Th_u \\ \text{cerrado,} & \text{si } EAR < Th_l \\ \text{último_estado,} & \text{en caso contrario.} \end{cases} \quad (17)$$

En la figura 31 se presenta un ejemplo visual del funcionamiento del algoritmo mediante el nuevo intervalo de incertidumbre.

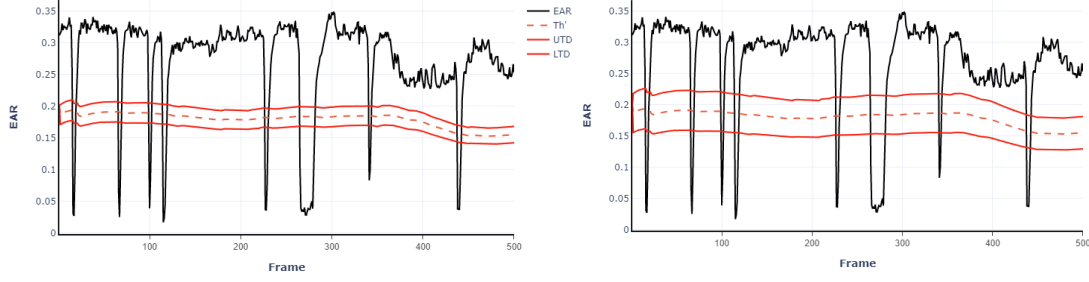


Figura 31: Comportamiento del intervalo de incertidumbre con $gz=0.05$ y $gz=0.1$, respectivamente.

Fuente: Elaboración propia

4.6.2.2 Análisis del estado de la boca

De manera similar al sistema que lleva a cabo un análisis del estado del ojo, el estado de la boca se determina mediante un *threshold*.

No obstante, en el caso del estado de la boca no resulta tan sencillo determinar un *threshold* dinámico, puesto que esta, de manera contraria a los ojos, suele estar cerrada la mayor parte de tiempo. Afortunadamente, la fisonomía de la boca de los sujetos no es tan diversa como ocurre con los ojos, y resulta relativamente sencillo determinar si está abierta o no.

Debido a esto, el sistema implementa un *threshold* estático, pasado como parámetro por el usuario.

Cabe destacar también, que este sistema presenta las mismas debilidades a las falsas detecciones que el sistema del ojo, causadas por la inmediatez de la transición entre los dos estados. Por lo tanto, se ha implementado el mecanismo que introduce un intervalo de incertidumbre en las mediciones del MAR, mediante el parámetro de zona gris, junto con un suavizado de las mediciones teniendo en cuenta la información del pasado mediante un nuevo parámetro.

- **Número de frames de suavizado (m)** – Determina el número de *frames* del pasado para los que se almacena la medida de la boca. A partir de estas medidas y el MAR del *frame* actual, se computa una media que será utilizada como la nueva medición para determinar el estado de la boca.

$$MAR' = \frac{MAR_0 + \sum_{i=1}^m MAR_i}{m + 1} \quad (18)$$

Donde MAR_x representa el valor de la medición de hace x *frames*.

En la figura 32 se presenta el comportamiento del MAR cuando se aplica el suavizado.

Finalmente, la función resultante de añadir estos parámetros es muy similar a la utilizada por el sistema del ojo.

$$f(MAR', Th_u, Th_l, ultimo_estado) = \begin{cases} \text{abierta,} & \text{si } MAR' > Th_u \\ \text{cerrada,} & \text{si } MAR' < Th_l \\ \text{último_estado,} & \text{en caso contrario.} \end{cases} \quad (19)$$

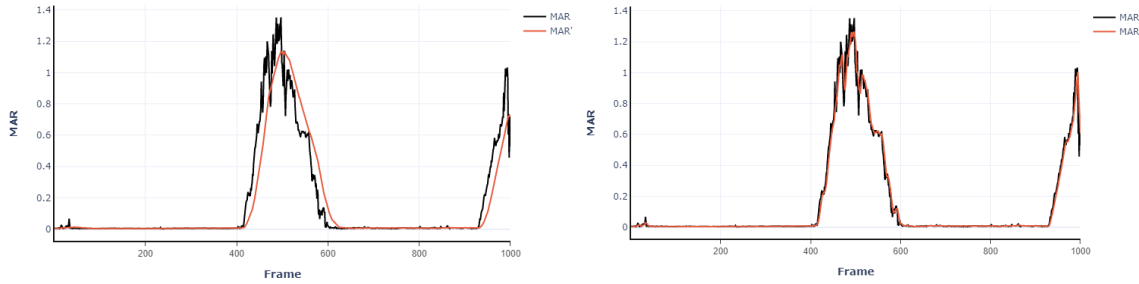


Figura 32: Evolución del MAR para un suavizado con $m=30$ y $m=5$, respectivamente

Fuente: Elaboración propia

4.6.2.3 Análisis de la postura de la cabeza

De nuevo, el análisis de la postura de la cabeza se lleva a cabo mediante el uso de un *threshold*.

El sistema implementado plantea los mismos parámetros y estrategias que se han visto para el sistema del ojo, añadiendo la estrategia de suavizado de las mediciones introducida en el sistema de la boca.

$$f(\text{pitch}', Th_u, Th_l, \text{ultimo_estado}) = \begin{cases} \text{abajo} & \text{si } \text{pitch}' > Th_u \\ \text{no_abajo} & \text{si } \text{pitch}' < Th_l \\ \text{último_estado}, & \text{en caso contrario.} \end{cases} \quad (20)$$

4.6.3 Detección de parpadeos, bostezos y cabeceos

Como ya se ha presentado en la sección 4.2.2, los parpadeos, bostezos y cabeceos son intervalos donde las regiones de interés se encuentran en el estado de detección positiva de manera consecutiva.

Para poder detectarlos, es necesario tener en cuenta el estado del *frame* actual y del anterior. El algoritmo utilizado es muy sencillo.

| Estado del <i>frame</i> actual | Estado del <i>frame</i> pasado | Respuesta |
|--------------------------------|--------------------------------|--------------------------------|
| Positivo | Positivo | El intervalo continúa |
| Positivo | Negativo | Inicio de un nuevo intervalo |
| Negativo | Positivo | Finaliza el intervalo |
| Negativo | Negativo | No hay ningún intervalo activo |

Tabla 4: Escenarios posibles para el tratamiento de los intervalos

Fuente: Elaboración propia

En la tabla 4 se describen los diferentes escenarios a valorar para determinar los intervalos de detección, donde se considera que el estado positivo es aquel para el que se activan dichos intervalos. Cada nuevo intervalo cuenta como una nueva detección.

4.6.4 Cálculo de las características de carácter temporal

Finalmente, teniendo en cuenta la información extraída del *frame* actual, se recalculan las características temporales de acuerdo con las definiciones que se han presentado en la sección 4.2.2.

Por otro lado, se lleva a cabo un almacenamiento de las características extraídas durante esta etapa para ser utilizadas durante la extracción de los siguientes *frames* de la secuencia. Cabe destacar que se ha decidido almacenar las características correspondientes a los *frames* del último minuto procesado de la secuencia, con el objetivo de dar más importancia a las características más recientes.

4.7 Reconocimiento de la somnolencia del conductor

En esta etapa del sistema se lleva a cabo la predicción de la somnolencia del conductor a partir de una combinación de las características temporales y de las características del *frame* actual.

La predicción se lleva a cabo mediante un clasificador previamente entrenado. Se ha decidido utilizar un modelo basado en un *gradient boosting* de árboles de decisión, puesto que se trata de un algoritmo que ofrece tiempos de entrenamiento y predicción muy reducidos, además de proporcionar un rendimiento alto en la clasificación del problema que se plantea resolver. Así mismo, el modelo también ofrece métodos que permiten explicar con mucha facilidad las predicciones realizadas y entender las variables que tienen más relevancia.

Cabe destacar que el sistema tarda unos pocos segundos para dar respuestas fiables, puesto que la información del pasado que se tiene en el inicio de la ejecución es muy reducida como para poder sacar conclusiones acerca del estado de los conductores.

Otros clasificadores valorados

Para obtener un entendimiento más profundo de la naturaleza del problema a tratar, se han valorado otros algoritmos de clasificación, teniendo en cuenta características como el tiempo de entrenamiento y predicción, la calidad de las respuestas y la explicabilidad del modelo. El funcionamiento de estos modelos se ha descrito con más detalle en la sección 3.4.

- **Regresión logística**
- **KNN**
- **SVM lineal**

Finalmente, estos algoritmos han sido descartados porque el *gradient boosting* de árboles de decisión presenta la combinación de características más beneficiosa para los objetivos del proyecto. En el experimento que se presentará en la sección 6.2 se puede ver una comparación más detallada de los distintos modelos.

4.8 Entrenamiento de los clasificadores

En este apartado se detalla el proceso de entrenamiento de los diferentes clasificadores que se han utilizado en la etapa de reconocimiento de la somnolencia.

Cabe destacar que este proceso no forma parte del programa final, pese a que resulta imprescindible para el buen funcionamiento de este.

Como se puede ver en la figura 33, el entrenamiento de los clasificadores se divide en varias fases.



Figura 33: Diagrama de flujo correspondiente al entrenamiento de los clasificadores

Fuente: Elaboración propia

4.8.1 Preprocesado de los datos de entrenamiento

Como se describe en la sección 3.4, algunos de los clasificadores valorados hacen uso de una función de distancia para determinar la calidad de las predicciones durante el proceso de entrenamiento.

Debido a esto, resulta de vital importancia llevar a cabo un tratamiento de los datos, para evitar que presenten diferentes magnitudes y tengan un impacto diferente sobre las respuestas de los modelos.

Para evitar esto, se ha llevado a cabo un reescalado de los datos, también conocido como *min-max scaling* al intervalo de valores [0, 1]. En la ecuación 21 se presenta la función utilizada para realizar el reescalado.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (21)$$

4.8.2 Elección de las características relevantes

Una vez se han tratado los datos de entrenamiento, se ha utilizado un clasificador con un conjunto de hiperparámetros estándar y se han llevado a cabo entrenamientos para determinar las características que aportan información relevante acerca del estado de los conductores. Los resultados de este experimento se presentarán con más detalle en la sección 6.2.

4.8.3 Elección del mejor tipo de clasificador

En esta etapa del entrenamiento se ha llevado a cabo una comparación de los diferentes clasificadores, entrenados mediante el conjunto de características seleccionado en la fase anterior. Para comparar los modelos, se han tenido en cuenta características como la calidad de las respuestas, el tiempo invertido en el entrenamiento y en las predicciones y la explicabilidad del modelo. Los resultados de este apartado se pueden ver con más detalle en la sección 6.2.

4.8.4 Búsqueda de los hiperparámetros óptimos

Una vez se ha determinado el tipo de clasificador a valorar, se ha llevado a cabo la fase final del proceso de entrenamiento, donde se ha llevado a cabo un *fine-tuning* de los hiperparámetros para encontrar los valores que optimizan el rendimiento del clasificador.

Este proceso consiste esencialmente en una búsqueda exhaustiva de los hiperparámetros que proporcionan la mejor respuesta de cada clasificador, dentro de un rango de valores establecido por el usuario.

Una vez se han optimizado los hiperparámetros, se lleva a cabo una comparación de los resultados obtenidos y se determina si ha habido una mejora significativa. El modelo seleccionado en esta fase ha sido utilizado para realizar el reconocimiento del cansancio en la solución propuesta. Los resultados de este apartado están descritos con más detalle en la sección 6.2.

Implementación

5.1 Consideraciones planteadas para el sistema

Se ha decidido acotar los escenarios en los que se va a valorar el rendimiento del sistema para simplificar el desarrollo del proyecto y así poder garantizar un mejor tratamiento de este conjunto limitado de escenarios.

Más concretamente, se ha decidido que la detección de las regiones de interés en escenarios donde los usuarios llevan gafas o donde las condiciones de iluminación son muy malas quedan fuera del alcance de este proyecto. Por lo tanto, el desarrollo se ha centrado en aquellos escenarios donde la detección de las regiones de interés resulta más sencilla para poder dedicar más tiempo a la parte más analítica del proyecto.

El proyecto se ha diseñado con el objetivo de ser ejecutado sobre secuencias de imágenes con una tasa de refresco de 30 *frames* por segundo. La resolución mínima que espera el sistema es de **640x480 píxeles** y se espera que las imágenes presenten un enfoque frontal de la cara de los conductores.

5.2 Lenguaje de programación utilizado

Se ha decidido desarrollar el proyecto en el lenguaje de programación de *Python* en lugar de Matlab, como se había planteado en un primer momento, puesto que el lenguaje ofrece un abanico bastante más amplio de soluciones y librerías de código abierto que han resultado de mucha utilidad para el desarrollo del proyecto.

Por otro lado, cabe también mencionar que se ha descartado el uso de lenguajes de más bajo nivel, como por ejemplo C++, puesto que, pese a que estos proporcionan un mejor rendimiento y ofrecen bastantes funcionalidades, el desarrollo resulta mucho más lento y la *performance* del sistema no se ha valorado como un objetivo prioritario en esta etapa del proyecto.

Por lo tanto, se ha decidido llevar a cabo el proceso experimental y el desarrollo inicial en *Python*, y la implementación del proyecto en un lenguaje compilado junto con las optimizaciones de rendimiento quedan fuera del alcance de este TFG.

5.3 Procesado de imagen

5.3.1 Librerías utilizadas

OpenCV

Para llevar a cabo el tratamiento de las imágenes se ha decidido utilizar la librería de OpenCV[7], puesto que ofrece un abanico muy grande de funcionalidades, desde la lectura y escritura de estas hasta la aplicación de técnicas más avanzadas como una ecualización adaptativa del histograma.

5.4 Segmentación de las regiones de interés

Para implementar las técnicas de segmentación descritas en la sección 4.5, se ha decidido utilizar la librería de Mediapipe[27].

5.4.1 Librerías utilizadas

Mediapipe

Mediapipe[27] es una librería que ofrece múltiples soluciones de *machine learning* al nivel del Estado del Arte para varios problemas en el ámbito de Visión por Computador. Destaca por ser fácilmente integrable y proporcionar soluciones rápidas y ligeras, además de ser capaz de funcionar en cualquier sistema operativo, incluyendo Android e IoT.

ML solutions in MediaPipe

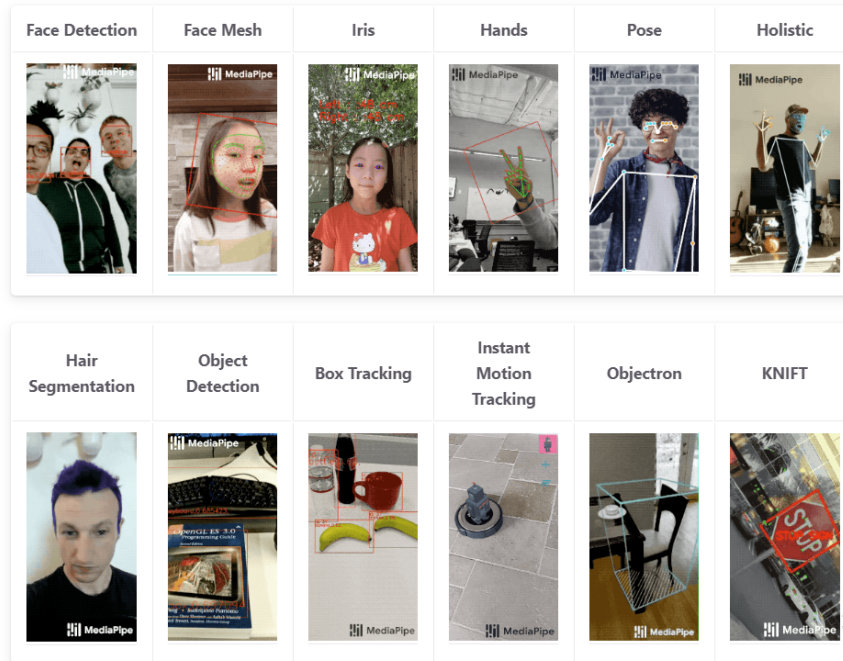


Figura 34: Soluciones ofrecidas por Mediapipe

Fuente: Imagen extraída de [17]

Dentro de las múltiples prestaciones que la librería ofrece, como se ha presentado en la sección 4.5, para el objetivo de este proyecto se destaca el modelo de detección de caras, y, de manera más ambiciosa, el modelo que genera una malla facial.

Mediapipe Face Detector

El detector de caras de Mediapipe es un modelo que destaca por la velocidad y la precisión de sus respuestas. Está basado en *BlazeFace*[5], basado en el funcionamiento de un *Single Shot Multibox Detector*[26], que consiste en un detector de caras ligero y preciso, diseñado para tener resultados satisfactorios sobre las targetas gráficas de los *smartphones* actuales.

No obstante, cabe destacar que el detector presenta un rendimiento algo inferior cuando las caras a detectar se encuentran en un ángulo muy horizontal, y es cierto que existen soluciones alternativas disponibles más robustas para tratar estos escenarios.

Sin embargo, el módulo de detección de mallas faciales de Mediapipe solo puede ser ejecutado sobre su detector de caras, de manera que el detector de Mediapipe resulta de más valor para el desarrollo del proyecto.

Mediapipe Face mesh

Mediapipe Face Mesh[27] consiste en una solución que estima con mucha precisión 468 puntos faciales en 3D a tiempo real requiriendo el uso de una sola cámara, sin necesidad de un sensor de profundidad.

El modelo proporciona respuestas en un tiempo de ejecución muy reducido, de manera que resulta muy valioso para poder llevar a cabo una segmentación sofisticada de las regiones de interés.

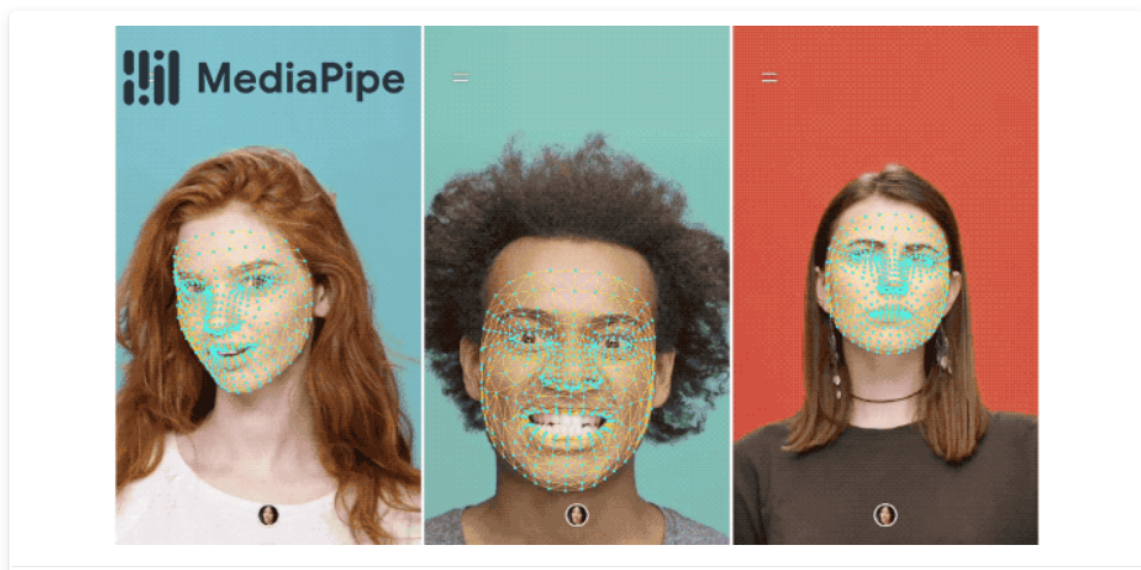


Figura 35: Mallas faciales generadas por el modelo

Fuente: [18]

El módulo de *face mesh* almacena los puntos faciales en una lista, y permite accederla mediante índices, de manera que la selección de los puntos faciales ha consistido simplemente en un conjunto de accesos a la lista generada por la librería.

5.5 Entrenamiento de los modelos predictivos

A continuación se describirán los diferentes pasos que se han llevado a cabo para realizar el entrenamiento de los diferentes modelos predictivos.

5.5.1 Librerías utilizadas

Pandas

Pandas[28] es una librería de Python especializada en la manipulación y análisis de datos. Se trata de un *framework* de código abierto rápido, eficiente, flexible y fácil de utilizar.

La librería introduce nuevas estructuras de datos muy útiles para tratar con *datasets* de una volumetría elevada, compatibles con la mayoría de librerías de Machine Learning que Python ofrece. Además, permite leer y escribir de manera sencilla ficheros en formato CSV y bases de datos SQL.

Debido a esto, el *framework* ha resultado de gran utilidad para el proyecto, puesto que ha permitido crear, manipular y analizar grandes conjuntos de datos, que se han utilizado para entrenar los modelos de Machine Learning utilizados para dar respuestas al problema de la detección del cansancio en conductores.

Scikit-Learn

Scikit-Learn[35] es una librería dedicada al Machine Learning para Python. El *framework* incluye una gran variedad de algoritmos de clasificación, regresión y análisis de grupos muy sencillos de utilizar e integrar en proyectos, como por ejemplo las máquinas de vectores de soporte (SVM) o el algoritmo de k-nearest-neighbours (KNN).

La librería ofrece también una gran variedad de funciones útiles para la valoración del rendimiento de los modelos de Machine Learning, así como para el preprocesado de los conjuntos de datos de entrada.

Gracias a esto, la librería ha resultado de gran utilidad, dando la posibilidad de probar una gran variedad de algoritmos de clasificación de manera muy sencilla y rápida.

LightGBM

LightGBM[25] es una librería de *gradient boosting* basada en el uso de árboles de decisión. Se trata de un *framework* muy competitivo que destaca por las siguientes características.

- **Alta velocidad de entrenamiento y eficiencia** — Los modelos de la librería consiguen buenos resultados en un espacio de entrenamiento muy reducido.
- **Buen rendimiento sin *fine-tuning*** — Los modelos presentan un buen rendimiento sin necesidad de hacer *fine-tuning* de los hiperparámetros. Esto es realmente útil para las fases de experimentación de un proyecto, donde se busca comparar distintos modelos en un espacio de tiempo razonable.
- **Conjunto de parámetros extenso** — Pese a que los modelos tengan un buen rendimiento sin mucha personalización, la librería ofrece un conjunto extenso de parámetros para modificar el comportamiento de estos.
- **Rendimiento elevado**
- **Bajo consumo de memoria**
- **Explicabilidad** — La librería incorpora funciones para obtener información acerca de las variables más importantes para un modelo. Además, presenta compatibilidad con librerías que tratan de explicar las decisiones de los modelos, como por ejemplo SHAP.
- **Librería fácil de utilizar y de integrar en proyectos**

Optuna

Optuna[3] es una librería dedicada a la optimización automática de hiperparámetros, desarrollada principalmente para modelos de Machine Learning. El *framework* realiza búsquedas eficientes, al nivel del Estado del Arte, sobre grandes espacios de valores definidos por el usuario y lleva a cabo podas de las ramificaciones que no prometen buenos resultados.

De esta manera, la librería ha habilitado un *fine-tuning* eficiente, fácil y sencillo de personalizar, permitiendo así elevar el rendimiento de los clasificadores desarrollados durante el transcurso del proyecto.

5.5.2 Dataset utilizado: NTHUDDD

5.5.2.1 Introducción

Para llevar a cabo el entrenamiento de los modelos predictores se ha decidido utilizar un *dataset* dedicado a la detección del cansancio en conductores, proporcionado por el departamento de Visión por Computador de la Universidad Nacional de Tsing Hua (NTHU)[8]. El conjunto de datos es utilizado como *benchmark* para una gran variedad de soluciones del Estado del Arte.

El *dataset* presenta un conjunto muy diverso de vídeos, donde se pueden encontrar sujetos masculinos y femeninos de diversas etnicidades, en un conjunto de escenarios muy variado y que supone un reto tanto para la extracción correcta de las características de interés como para el correcto análisis de la somnolencia.

Los vídeos presentan una resolución relativamente baja, de 640x480 píxeles, y la mayoría presentan una tasa de refresco de 30 fotogramas por segundo. Cabe destacar también que estos se encuentran en escala de grises.

El contenido del *dataset* se divide en tres conjuntos.

Conjunto de entrenamiento

El conjunto de entrenamiento tiene como propósito proporcionar un conjunto de vídeos y anotaciones que permitan generar un *dataset* apto para el correcto entrenamiento de los algoritmos de clasificación basados en Machine Learning.

Este presenta un conjunto de vídeos pertenecientes a una combinación de 18 posibles sujetos, que se pueden encontrar en los siguientes escenarios y estados.

Escenarios Dentro del *dataset* de entrenamiento se pueden encontrar 5 escenarios distintos.

- **noglasses** — Los vídeos están grabados de día y los sujetos no llevan gafas.
- **glasses** — Los vídeos están grabados de día y los sujetos llevan gafas.
- **sunglasses** — Los vídeos están grabados de día y los sujetos llevan gafas de sol.
- **night-noglasses** — Los vídeos están grabados de noche y los sujetos no llevan gafas.
- **night-glasses** — Los vídeos están grabados de noche y los sujetos llevan gafas.

Estados del sujeto Dentro de cada escenario se pueden encontrar 4 vídeos correspondientes a los diferentes estados en los que se pueden encontrar los sujetos.

- **nonsleepyCombination** — Consiste en una combinación de comportamientos propios de personas que no padecen síntomas de fatiga. Por ejemplo, se puede ver a los sujetos hablando, riendo, mirando a un lado, etc.
- **sleepyCombination** — Consiste en un repertorio de acciones o indicios propios de personas que sufren de somnolencia. Por ejemplo, bostezos, cabeceos u ojos cansados.
- **yawning** — En este tipo de vídeos se puede ver a los sujetos bostezando y con ojos cansados.
- **slowBlinkWithNodding** — En este tipo de vídeos se puede ver a los sujetos con los ojos cansados y dando cabeceos.

Anotaciones Por cada vídeo correspondiente a uno de los estados de los sujetos se pueden encontrar 4 ficheros de anotaciones.

Los ficheros presentan un dígito por cada *frame* del vídeo a analizar, que describe el estatus de este respecto a la característica que se esté valorando.

- **Drowsiness.txt** — Representa el estado de cansancio del sujeto en cada *frame* del vídeo. Un valor de 0 representa alerta, mientras que un 1 representa fatiga.
- **Eye.txt** — Representa el estado de los ojos del sujeto. Un valor de 0 indica que los ojos se encuentran alerta, mientras que un valor de 1 representa unos ojos cansados.
- **Head.txt** — Representa el comportamiento del sujeto referente a la cabeza. Un valor de 0 indica que está quieta, un 1 representa que se está dando un cabeceo y un 2 indica movimientos laterales de la cabeza.
- **Mouth.txt** — Representa el comportamiento del sujeto referente a la boca. Un valor de 0 indica que está cerrada, un 1 representa que el sujeto está bostezando y un 2 indica que el sujeto está hablando o riendo.

Dentro de los 4 ficheros de anotaciones que presenta cada vídeo, el entrenamiento de los modelos se ha centrado únicamente en los ficheros de **drowsiness.txt**, puesto que son los que presentan las respuestas que los modelos deberían proporcionar.

Los ficheros de **mouth.txt** y **head.txt** han sido utilizados para la valoración la calidad de extracción de las características, mientras que el fichero de **eye.txt** no ha resultado de mucha utilidad para el proyecto, puesto que las definiciones de ojos cansados y alerta no se alinean con la manera en la que se han definido las características extraídas de los ojos.

Conjuntos de evaluación y test

Los conjuntos de evaluación y test tienen como propósito evaluar la calidad de los modelos que se han entrenado con los datos del conjunto de entrenamiento.

Su estructura es muy similar a la del conjunto de entrenamiento, es decir, los escenarios se mantienen pero los estados se han combinado en un único vídeo, que contiene una combinación de fragmentos donde se pueden encontrar diferentes tipos de comportamiento. Por otro lado, los vídeos de evaluación y *test* únicamente presentan el fichero de anotación de **drowsiness.txt**, puesto que lo que se quiere evaluar es la somnolencia de los sujetos.

5.5.2.2 Tratamiento

En este apartado se presentarán las decisiones que se han tomado respecto a la utilización del *dataset* de NTHUDDD, así como el tratamiento que se ha realizado para obtener un conjunto de datos que los modelos de Machine Learning puedan procesar.

Como se ha explicado en la sección 5.1, se ha decidido no trabajar con los escenarios donde los sujetos llevan gafas o en los que las condiciones de iluminación son muy malas, dada la elevada dificultad que supone su tratamiento.

Debido a esto, se ha decidido trabajar únicamente con los escenarios de *noglasses*, puesto que son los únicos que cumplen con estas condiciones.

Tratamiento del fichero de anotaciones

El fichero de anotaciones presenta la información correspondiente a su respectivo vídeo mediante una secuencia de dígitos, donde cada uno de estos representa el estatus del frame correspondiente a su posición en la secuencia.

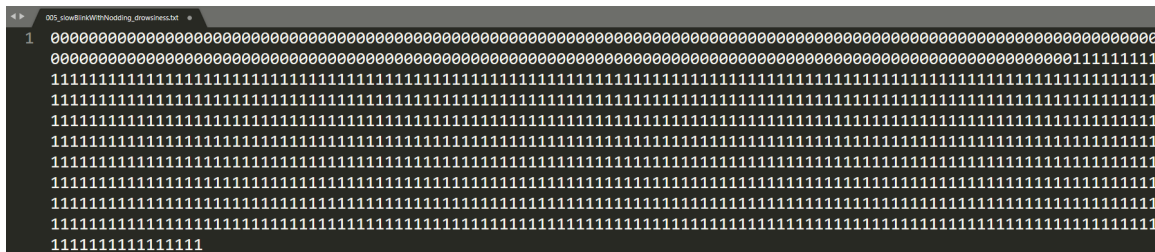


Figura 36: Contenido de los ficheros de anotaciones

Fuente: Elaboración propia

Como se puede ver en la figura 36, puesto que todos los dígitos se encuentran en la misma línea del fichero, resulta muy sencillo leer la información y almacenarla en una lista, donde cada posición de esta corresponde a un *frame* del vídeo original.

Tratamiento del vídeo

El tratamiento del vídeo se ha llevado a cabo mediante el módulo de obtención de características que se ha presentado en la sección 4.6. Por cada *frame* del vídeo, se ha ejecutado la extracción de características y se ha obtenido un diccionario donde se almacenan los valores de cada indicador.

Este diccionario ha sido añadido a una lista, donde cada posición corresponde a un *frame* del vídeo y contiene la información que se ha extraído de este. En la figura 38 se muestra el contenido del diccionario.

Finalmente, una vez se ha iterado sobre todos los *frames* del vídeo, se convierte la lista resultante a un *DataFrame* de **Pandas**, se le integra la lista de anotaciones y se almacena como fichero csv.

En la figura 39 se presenta una parte del fichero csv generado.

Este proceso se ha llevado a cabo por cada vídeo de los tres conjuntos y, finalmente, se ha realizado una fusión de todos los ficheros csv en un único archivo, apto para el entrenamiento de los modelos, que contiene información acerca del conjunto, sujeto, escenario y estado de los datos que lo conforman.

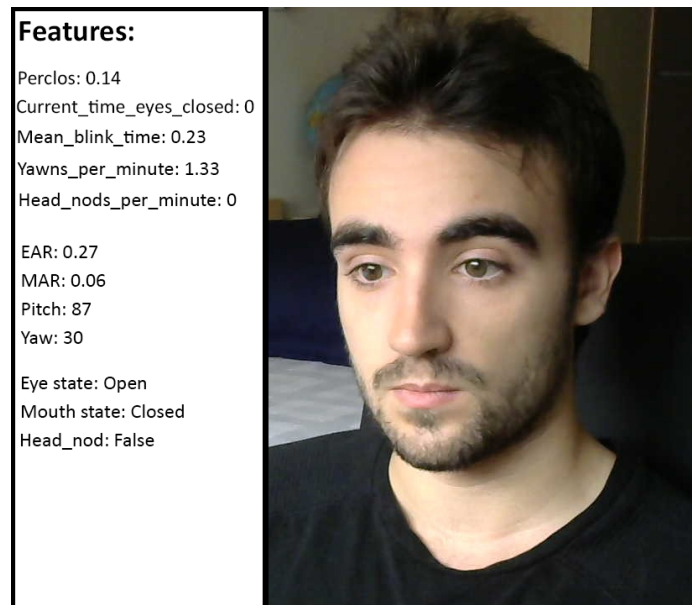


Figura 37: Características extraídas del *frame* actual y de la secuencia temporal

Fuente: Elaboración propia

| Frame | EAR | MAR | Pitch | Yaw | Eye_state | Mouth_state | Head_nod | ... | Yawns_per_minute | Head_nods_per_minute |
|-------|------|------|-------|-----|-----------|-------------|----------|-----|------------------|----------------------|
| 163 | 0.27 | 0.06 | 87 | 32 | Open | Closed | False | ... | 1.33 | 0 |

Figura 38: Características del *frame* puntual en formato de diccionario

Fuente: Elaboración propia

| Frame | EAR | MAR | Pitch | Yaw | Eye_state | Mouth_state | Head_nod | ... | Yawns_per_min | Head_nods_per | Drowsiness |
|-------|------|------|-------|-----|-----------|-------------|----------|-----|---------------|---------------|------------|
| ... | | | | | | | | | | | |
| 163 | 0.27 | 0.06 | 87 | 32 | Open | Closed | False | ... | 1.33 | 0 | 1 |
| 164 | 0.21 | 0.03 | 89 | 29 | Open | Closed | False | ... | 1.33 | 0 | 1 |
| 165 | 0.14 | 0.02 | 91 | 36 | Closed | Closed | False | ... | 1.33 | 0 | 1 |
| 166 | 0.06 | 0.09 | 90 | 37 | Closed | Closed | False | ... | 1.33 | 0 | 1 |
| 167 | 0.09 | 0.04 | 91 | 33 | Closed | Closed | False | ... | 1.33 | 0 | 1 |
| 168 | 0.13 | 0.05 | 87 | 34 | Closed | Closed | False | ... | 1.32 | 0 | 1 |
| 169 | 0.2 | 0.03 | 88 | 32 | Open | Closed | False | ... | 1.32 | 0 | 1 |
| 170 | 0.24 | 0.07 | 89 | 28 | Open | Closed | False | ... | 1.32 | 0 | 1 |
| ... | | | | | | | | | | | |

Figura 39: Extracto del fichero csv que contiene las características del vídeo tratado

Fuente: Elaboración propia

5.5.3 Entrenamiento de los modelos

Una vez se ha generado el fichero csv que contiene las características de cada *frame* y la respuesta que los modelos tienen que dar, se puede iniciar el proceso de entrenamiento de los modelos de Machine Learning.

El proceso de entrenamiento de un clasificador se estructura de la siguiente manera.

1. **Lectura del fichero csv** – Se carga el fichero csv generado durante el tratamiento del *dataset* y se almacena en un *DataFrame* de **Pandas**.
2. **Procesado de los datos** – Se eliminan las filas correspondientes a los *frames* donde no han habido detecciones, puesto que no se ha podido extraer las características. También se divide el *dataset* en los conjuntos de datos de entrenamiento, validación y *test*.
3. **Entrenamiento del clasificador** – Se lleva a cabo el entrenamiento del clasificador, mediante los conjuntos de entrenamiento y validación, que serán pasados como argumentos junto con los valores de los hiperparámetros.
4. **Validación de los resultados** – Se ejecuta el clasificador sobre el conjunto de *test* y se comparan las predicciones con las respuestas esperadas. Para valorar los resultados, se computa una tabla de confusión y las métricas de exactitud, precisión, exhaustividad y *f1-score*.

5.6 Mecanismo de inferencia

El mecanismo de inferencia representa el funcionamiento del producto final. Se trata de un programa que recibe una secuencia de imágenes como entrada, computa las características que describen el estado del conductor y monitoriza la fatiga del mismo, realizando predicciones de manera periódica.

5.6.1 Entrada

El programa ha sido diseñado para funcionar de la misma manera sobre vídeos y cámaras *web*. El programa funciona mediante el uso de cuatro parámetros.

- **Vídeo (v)** – Define si la secuencia de entrada se trata de un vídeo cuando el valor del parámetro es 1, cuya localización se tendrá que especificar, o si se trata de una cámara *web*, cuando el valor del parámetro es 0, en cuyo caso se tendrá que especificar el índice de la cámara a utilizar. El parámetro tiene la opción de vídeo como valor por defecto.
- **Localización del vídeo (path)** – Este parámetro únicamente será utilizado si se ha especificado que la secuencia de entrada se trata de un vídeo, y representa la localización de dicha secuencia en el sistema de ficheros.
- **Índice de la cámara (ind)** – Puesto que puede haber más de una cámara disponible en un mismo ordenador o dispositivo, este parámetro permite seleccionar entre las distintas opciones disponibles. Solo será utilizado si se ha especificado el uso de la cámara *web* como secuencia de entrada.
- **Complejidad de la salida (c)** – Este parámetro permite establecer la cantidad de información que se quiere generar al finalizar la ejecución del programa. Un valor de 0 especifica una salida simple, donde se proporciona la mínima información necesaria para describir la somnolencia del conductor, mientras que un valor de 1 proporciona toda la información que se ha utilizado para la toma de decisiones del sistema.

5.6.2 Salida

El programa genera la misma salida para los dos tipos de secuencia disponibles.

Salida reducida

Si el programa se ejecuta con la salida reducida, este únicamente proporcionará una lista con las predicciones acerca del estado de somnolencia para cada *frame* de la secuencia de imágenes de entrada a través de la salida estándar.

Salida completa

En caso de ejecutar el programa con la salida completa, este almacenará además la siguiente información.

- **Vídeo con las respuestas** – Consiste en una secuencia de imágenes en formato *.avi* donde se han escrito las características y las predicciones realizadas a tiempo real.
- **Fichero csv** – Consiste en un fichero similar a los que se han utilizado para llevar a cabo los entrenamientos, donde se especifican las características obtenidas y las predicciones realizadas para cada *frame* de la secuencia.

5.6.3 Ejecución del programa

El programa se puede ejecutar a través de la línea de comandos, seleccionando la secuencia de imágenes de entrada y las diferentes configuraciones del sistema mediante el uso de los parámetros ya presentados.

```
python main.py -v 1 -path "path_to_video.mp4" -c 0
```

Este comando representa la ejecución del programa sobre un vídeo y con una salida reducida.

5.7 Tiempo de ejecución del sistema

Pese a que el desarrollo del proyecto se ha enfocado más en el diseño del producto final y en los resultados obtenidos, resulta de mucho interés realizar un análisis del tiempo de ejecución del sistema para valorar la viabilidad futura del proyecto.

Este se ha analizado computando el tiempo medio que tarda el sistema en procesar 30 *frames*, correspondientes a un segundo de grabación mediante las cámaras que se especifican como target del proyecto.

El programa tarda alrededor de 2,5 segundos en procesar completamente 30 *frames*. Esto corresponde a un rendimiento de alrededor de **12 fps** y resulta un rendimiento insuficiente para el objetivo final del proyecto, puesto que se quiere poder ejecutar el sistema en dispositivos *hardware* pequeños.

No obstante, cabe destacar que el sistema actual está implementado en *python*, un lenguaje de programación que no tiene como objetivo optimizar el rendimiento de los programas, sino más bien permitir un desarrollo rápido, puesto que se trata de un lenguaje interpretado. La implementación del sistema en un lenguaje compilado, como puede ser C++, puede llegar a comportar un **speed-up** de alrededor de 5[45].

Por otro lado, también existen muchas optimizaciones que se pueden llevar a cabo para reducir el consumo de memoria y el tiempo de ejecución global del sistema. Cabe destacar que el modelo utilizado para calcular los ángulos de navegación, descrito en la sección 4.6.1.3, se ejecuta durante

cada *frame* de las secuencias a analizar, y esto actualmente supone la mitad del tiempo de ejecución del sistema. Puesto que la detección de cabeceos no requiere que se realicen mediciones durante cada *frame* debido a que la duración de estos es bastante prolongada, una optimización muy relevante que se puede llevar a cabo consiste en reducir la frecuencia de ejecución de este modelo.

En conclusión, las sensaciones acerca del potencial rendimiento del sistema son muy positivas, puesto que únicamente se ha de alcanzar un **speed-up** de 2.5 para alcanzar los requisitos de *performance* que se han establecido y existen muchas vías de mejora para alcanzar este objetivo.

Resultados

En esta sección se exponen los diferentes experimentos realizados durante el desarrollo del proyecto con el objetivo de valorar el rendimiento de los diferentes mecanismos que forman parte del sistema que se ha desarrollado.

6.1 Experimentos relacionados con los métodos de obtención de características

6.1.1 Detección de parpadeos sobre secuencias de vídeos

Como ya se ha expuesto previamente, las características obtenidas a partir de los ojos son cruciales para un buen análisis de la fatiga de una persona. Debido a esto, es necesario llevar a cabo experimentos que permitan, no solo garantizar la calidad de obtención de dichas características, sino que también posibiliten la comparación de los diferentes métodos implementados con el objetivo de seleccionar el más robusto.

En este apartado se presentarán los diferentes experimentos que se han llevado a cabo para valorar los resultados de las diferentes etapas del sistema presentadas en la sección 4.6.2.1.

El objetivo de los experimentos es comparar la calidad de un sistema base respecto a uno que introduce un cambio significativo en el tratamiento de la información, para así determinar si el nuevo cambio incrementa la calidad y robustez de las respuestas.

Una vez se han llevado a cabo todos los experimentos y decidido los mejores mecanismos para el tratamiento de la información, se realizará una comparación entre los diferentes métodos de medición y se optará por el que obtenga mejores resultados, teniendo en cuenta las desventajas que presentan.

Dataset utilizado

El conjunto de datos utilizado en este experimento es el *dataset* de **Eyeblink8**[14], que presenta un conjunto de vídeos grabados en entornos domésticos, donde los sujetos actúan con naturalidad delante de la cámara.

Los vídeos modelan las problemáticas que los algoritmos se pueden encontrar sobre sujetos en casos reales, como oclusiones, cambios de iluminación o un desenfoque por movimiento, además de presentar resoluciones relativamente bajas.

El *dataset* contiene 8 vídeos pertenecientes a 4 sujetos, donde se puede destacar un escenario en el que un sujeto lleva gafas. Los vídeos presentan un total de 408 parpadeos en 70.992 frames con una resolución de 640x480 píxeles.

Target

El *target* del *dataset* son los parpadeos, presentados como intervalos de *frames* donde los ojos están cerrados de manera consecutiva.

Algoritmo de valoración

Como método de valoración de los resultados, se computarán los intervalos de parpadeo derivados de las anotaciones del *dataset*, así como los obtenidos a partir de la aplicación de los diferentes métodos que se quieren valorar.

Una vez se tienen los intervalos, se valorarán como aciertos o positivos reales los intervalos generados que presenten una intersección no vacía con los parpadeos reales. Si la intersección de estos es vacía, el intervalo se valorará como un falso positivo si este termina antes que el intervalo real, mientras que será determinado como falso negativo en el caso contrario. El funcionamiento queda detallado en el algoritmo 1.

Algorithm 1 Valoración de los intervalos detectados

```
input true_intervals, predicted_intervals
  true_positives ← 0
  false_positives ← 0
  false_negatives ← 0
  i1 ← 0
  i2 ← 0
  num_true_intervals ← len(true_intervals)
  num_predicted_intervals ← len(predicted_intervals)
while i1 < num_real_intervals AND i2 < num_method_intervals do
  true_blink ← true_intervals[i1]
  predicted_blink ← predicted_intervals[i2]
  if true_blink ∩ predicted_blink ≠ ∅ then
    true_positives ← true_positives + 1
    i1 ← i1 + 1
    i2 ← i2 + 1
  else if true_blink[-1] < predicted_blink[-1] then
    false_negatives ← false_negatives + 1
    i1 ← i1 + 1
  else
    false_positives ← false_positives + 1
    i2 ← i2 + 1
  end if
end while
false_negatives ← false_negatives + num_real_intervals - i1
false_positives ← false_positives + num_method_intervals - i2
```

Métodos a valorar

Los métodos a valorar se corresponden con los presentados en la sección 4.6.1.1.

- **Método 1:** Método basado en el EAR no euclidiano
- **Método 2:** Método basado en el EAR euclidiano
- **Método 3:** Método basado en la apertura respecto al diámetro del iris

Experimento 1: Sistema inicial

El primer experimento a realizar tiene como objetivo valorar la calidad del sistema inicial, presentado en la sección 4.6.2.1.

Este sistema utiliza un *threshold* estático como único parámetro para determinar el estado del ojo.

Procedimiento

Para cada uno de los métodos de medición a valorar, se llevará a cabo una exploración exhaustiva del parámetro que optimiza los resultados. Una vez se hayan encontrado estos valores, se podrá llevar a cabo una comparación justa para determinar el método que proporciona las mejores respuestas.

Para cada método, se ha llevado a cabo una exploración del rango de valores [0, 1] para el parámetro de *threshold* estático (th).

Resultados

En la figura 40 se puede ver la evolución de la *f1-score* respecto al valor del *threshold* estático para los métodos de medición planteados y en la figura 41 se puede ver una comparación del rendimiento de los métodos para cada sujeto.

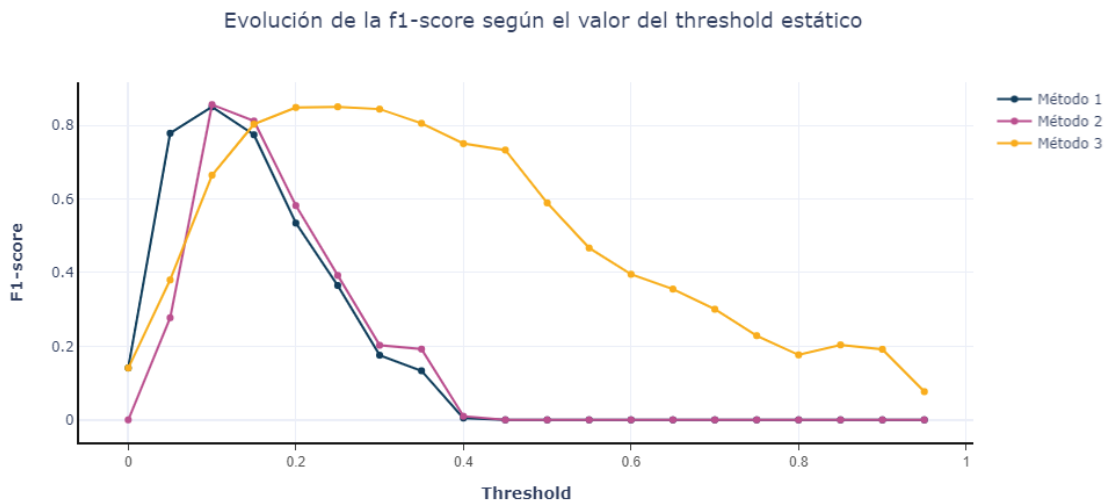


Figura 40: Evolución de la calidad de las respuestas según el valor del *threshold* estático

Fuente: Elaboración propia.

Por otro lado, en la tabla 5 se pueden ver los parámetros que optimizan los resultados de cada método, junto con las métricas que describen la calidad de sus respuestas.

Conclusiones

Como se puede observar en la tabla 5, los tres métodos proporcionan resultados muy similares, con una *f1-score* alrededor del 85 %.

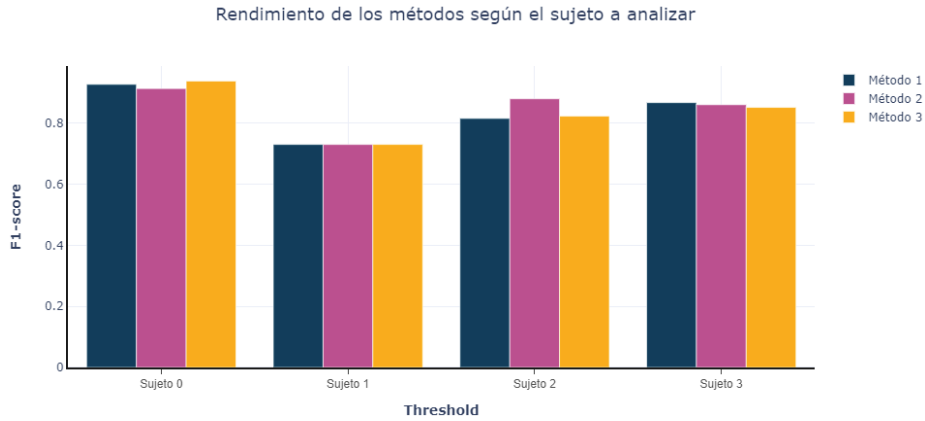


Figura 41: Comparación del rendimiento de los métodos para los distintos sujetos

Fuente: Elaboración propia.

| Método | Parámetro óptimo | Precisión | Exhaustividad | F1-Score |
|--------|------------------|-----------|---------------|----------|
| 1 | Th = 0.1 | 0.83 | 0.87 | 0.85 |
| 2 | Th = 0.1 | 0.88 | 0.83 | 0.86 |
| 3 | Th = 0.25 | 0.85 | 0.85 | 0.85 |

Tabla 5: Rendimiento de los diferentes métodos implementados

Fuente: Elaboración propia.

Sorprendentemente, las métricas obtenidas son bastante altas, teniendo en cuenta que el sistema únicamente implementa un *threshold* estático. Estos resultados pueden deberse a que todos los sujetos del *dataset* son de la misma raza y la fisonomía de sus ojos es bastante similar.

No obstante, en la figura 41 se puede observar que el rendimiento de los métodos depende bastante del sujeto que se está analizando, y por lo tanto, es probable que un sistema más sofisticado, capaz de tener en cuenta la fisonomía del ojo del sujeto, pueda obtener resultados más robustos.

Experimento 2: *Threshold* dinámico

El siguiente experimento tiene como objetivo valorar la calidad del sistema una vez se han integrado los mecanismos que computan un *threshold* dinámico, descritos en la sección 4.6.2.1, para así determinar si estos mecanismos proporcionan una mejora de rendimiento significativa.

Procedimiento

De manera análoga al primer experimento, se llevará a cabo una exploración exhaustiva de los parámetros que optimizan los resultados y, una vez se hayan encontrado estos valores, se realizará una comparación de los diferentes métodos para determinar el que proporciona mejores respuestas.

Para el parámetro de *threshold_perc*, se explorará el rango de valores de [0.2, 0.8] y para el parámetro *n* se explorará el rango [0, 1000].

Resultados

En la figura 42 se puede ver la evolución de la *f1-score* de los métodos de acuerdo con los dos nuevos parámetros implementados y en la figura 43 una comparación del rendimiento de los métodos para los diferentes sujetos.

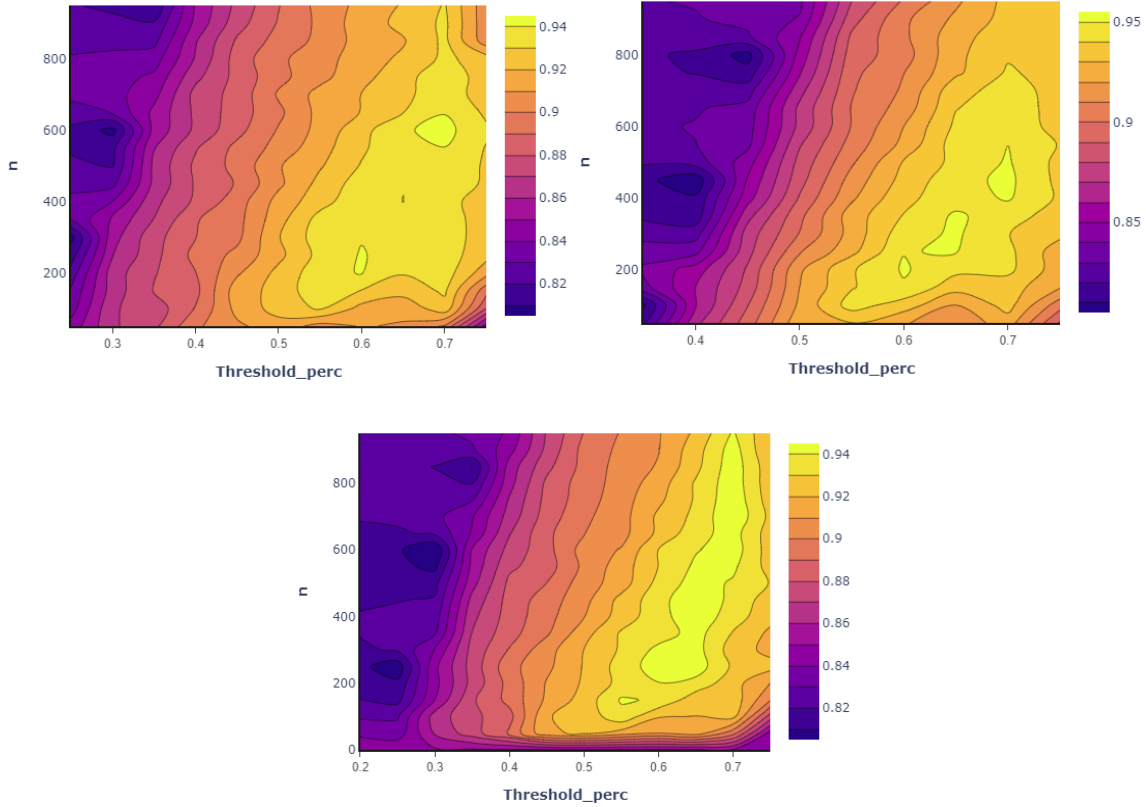


Figura 42: Evolución de los resultados según los dos parámetros a valorar para los métodos 1, 2 y 3, respectivamente.

Fuente: Elaboración propia.

Por otro lado, en la tabla 6 se presentan los parámetros que optimizan los resultados de cada método, junto con las métricas que describen la calidad de sus respuestas.

| Método | Parámetros óptimos | Precisión | Exhaustividad | F1-Score |
|--------|-------------------------|-----------|---------------|----------|
| 1 | th_perc = 0.7, n = 600 | 0.91 | 0.98 | 0.94 |
| 2 | th_perc = 0.7, n = 450 | 0.93 | 0.98 | 0.95 |
| 3 | th_perc = 0.65, n = 450 | 0.93 | 0.97 | 0.95 |

Tabla 6: Comparativa del rendimiento de los métodos implementados

Fuente: Elaboración propia.

Conclusiones

Como se puede ver en la tabla 6, los métodos siguen presentando resultados muy similares.

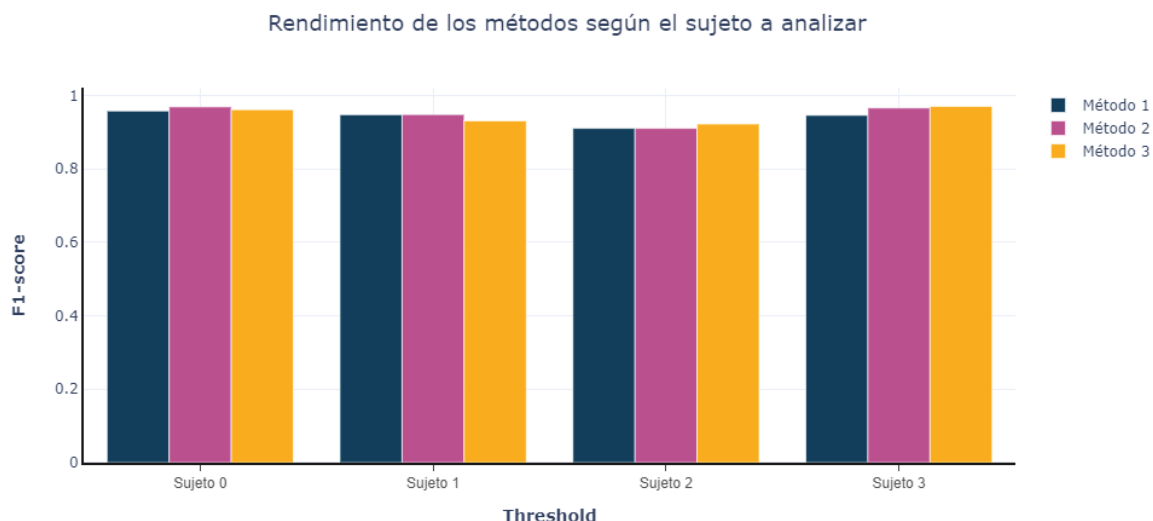


Figura 43: Comparativa del rendimiento de los métodos implementados para los distintos sujetos

Fuente: Elaboración propia.

Cabe destacar que la introducción del *threshold* dinámico ha comportado una mejora general de un 10 % en la métrica de *f1-score*, de modo que se puede afirmar que la mejora introducida es significativa.

También cabe mencionar que todos los métodos presentan una exhaustividad mayor que su precisión, lo que se traduce en un número de falsos positivos mayor que de falsos negativos. Es decir, los métodos detectan casi todos los parpadeos, pero también realizan detecciones adicionales que no son correctas.

Una de las posibles causas de estas detecciones es la inmediatez en el paso de uno de los estados al otro.

Por otro lado, como se puede ver en la figura 43, se confirma que la falta de precisión en algunos de los sujetos se debía a que el sistema de *threshold* estático no tenía en cuenta las fisonomía de cada individuo, puesto que ahora los resultados son mucho más equilibrados, y el sistema presenta un comportamiento mucho más robusto.

Experimento 3: Introducción de un intervalo de incertidumbre

Finalmente, en este experimento se valorará el impacto del mecanismo de zona gris descrito en la sección 4.6.2.1, diseñado para tratar de reducir las detecciones falsas de parpadeos.

Procedimiento

De manera análoga al primer experimento, se llevará a cabo una exploración exhaustiva de los parámetros que optimizan los resultados y, una vez se hayan encontrado estos valores, se realizará una comparación de los diferentes métodos para determinar el que proporciona mejores respuestas.

Para cada método, se explorará el intervalo de valores [0.2, 0.8] para el *threshold_perc*, el rango [0, 0.25] para la zona gris y el intervalo [0, 1000] para el parámetro *n*.

Resultados

En la tabla 7 se presentan los parámetros que optimizan los resultados de cada método, junto con las métricas que describen la calidad de sus respuestas.

| Método | Parámetros óptimos | Precisión | Exhaustividad | F1-Score |
|--------|------------------------------------|-----------|---------------|----------|
| 1 | th_perc = 0.75, gz = 0.05, n = 600 | 0.94 | 0.97 | 0.95 |
| 2 | th_perc = 0.75, gz = 0.05, n = 450 | 0.94 | 0.98 | 0.96 |
| 3 | th_perc = 0.7, gz = 0.05, n = 450 | 0.95 | 0.97 | 0.96 |

Tabla 7: Rendimiento de los diferentes métodos implementados

Fuente: Elaboración propia.

Conclusiones

En la tabla 7 se puede observar que la precisión de los métodos ha aumentado ligeramente, mientras que su exhaustividad se ha mantenido. Por lo tanto, se puede afirmar que la introducción de la zona gris ha comportado una reducción de las falsas detecciones, pese a que sigue habiendo margen de mejora.

Respecto a la calidad general de los métodos, esta ha mejorado muy ligeramente, puesto que las métricas ya se encuentran bastante cerca de la perfección. Hay que tener en cuenta que la precisión de los puntos faciales detectados por el sistema, a partir de los que se calculan las mediciones, también introduce un cierto grado de error en las detecciones, y por lo tanto, es improbable alcanzar una detección perfecta.

Respecto a la elección del método de detección, los diferentes experimentos han mostrado que sus resultados son bastante similares. Esto se debe posiblemente a que las mediciones son bastante similares entre sí, con pequeñas variaciones acerca de la invariabilidad cuando los individuos giran la cara, y este escenario no es muy frecuente en los vídeos del *dataset*.

Por lo tanto, se ha decidido escoger el método 2, basado en el EAR euclidiano, puesto que resulta sencillo de computar, es más invariable que el método 1 a cambios en el ángulo de *Roll*, y no requiere tener que computar una media del diámetro del iris.

En conclusión, la calidad general de los métodos es satisfactoria para el objetivo del proyecto y, la introducción de los diferentes mecanismos planteados ha comportado una mejora significativa en el rendimiento del sistema.

6.1.2 Detección de bostezos sobre secuencias de vídeo

Dataset utilizado

Como ya se ha presentado en la sección 5.5.2, se utilizará el dataset NTHUDDD para la valoración de los resultados, puesto que contiene anotaciones que indican los intervalos temporales donde los sujetos se encuentran bostezando, hablando y riendo, o simplemente con la boca cerrada.

Target

El *target* del *dataset* son los bostezos, presentados como intervalos de *frames* donde la boca se encuentra muy abierta de manera prolongada.

Algoritmo de valoración

El algoritmo de valoración utilizado es el mismo que se introduce en la sección 6.1.1 para los parpadeos, aplicado a los intervalos de bostezo.

Métodos a valorar

Los métodos a valorar se corresponden con los presentados en la sección 4.6.1.2

- **Método 1:** Método basado en el EAR no euclidiano
- **Método 2:** Método basado en el EAR euclidiano

Experimento 1: Sistema inicial

El primer experimento a realizar tiene como objetivo valorar la calidad del sistema inicial, presentado en la sección 4.6.2.2.

De manera similar al sistema implementado para la detección del estado del ojo, este sistema utiliza únicamente un *threshold* estático como parámetro.

Procedimiento

Para cada uno de los métodos de medición a valorar, se llevará a cabo una exploración exhaustiva del parámetro que optimiza los resultados. Una vez se han determinado estos valores óptimos, se llevará a cabo una comparación de la calidad de los diferentes métodos implementados.

Para cada método, se ha llevado a cabo una exploración del rango de valores $[0, 1]$ para el parámetro de *threshold* estático (th).

Resultados

En la figura 44 se puede ver la evolución de la *f1-score* respecto al valor del *threshold* estático para los métodos de medición planteados.

Por otro lado, en la tabla 8 se puede ver un resumen de los parámetros que optimizan los resultados de cada método, junto con las métricas que describen la calidad de sus respuestas.

Evolución de la f1-score según el valor del threshold estático

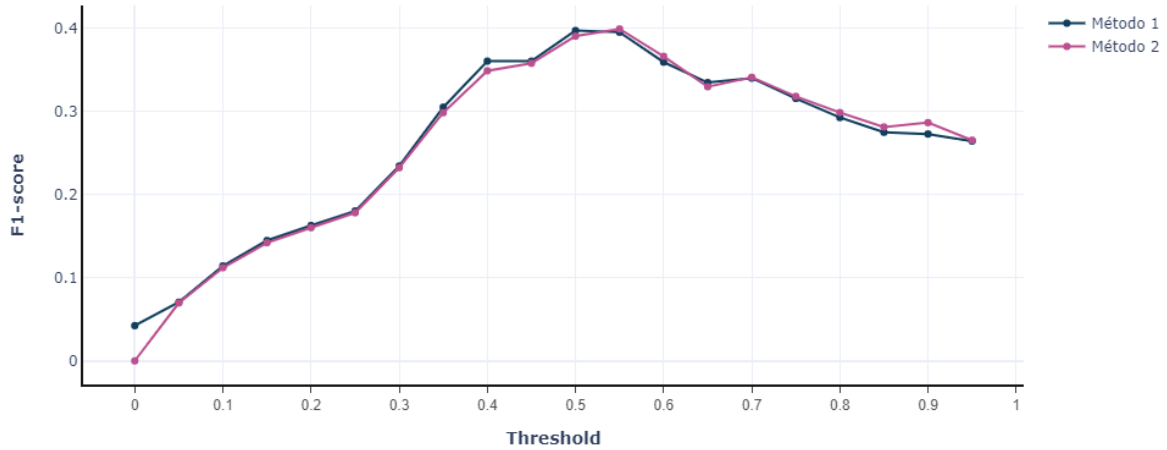


Figura 44: Evolución del rendimiento de los métodos según el valor del *threshold* estático

Fuente: Elaboración propia.

| Método | Parámetro óptimo | Precisión | Exhaustividad | F1-Score |
|--------|------------------|-----------|---------------|----------|
| 1 | Th = 0.5 | 0.25 | 0.96 | 0.4 |
| 2 | Th = 0.55 | 0.25 | 0.95 | 0.4 |

Tabla 8: Rendimiento de los diferentes métodos implementados

Fuente: Elaboración propia.

Conclusiones

Como se puede ver en la tabla 8, la calidad general de las respuestas de los métodos planteados es muy baja, a causa de que los sistemas presentan una precisión muy mala.

Puesto que la exhaustividad es muy alta, los sistemas tienen un número de falsos negativos muy bajo, mientras que los falsos positivos son muy abundantes. Esto quiere decir que el sistema detecta casi todos los bostezos reales, pero se realizan muchísimas detecciones de bostezos que no son correctas.

Tras analizar las respuestas de los sistemas obtenidos, se ha llegado a la conclusión de que existen dos causas para la abundancia de falsas detecciones.

- **Falsas detecciones del habla o de la risa** — En los escenarios donde las personas se encuentran hablando o riendo se están realizando detecciones de bostezos, a causa de que el único criterio utilizado es un *threshold* estático que solo tiene en cuenta información del *frame* actual.
- **Segmentación de los bostezos** — En los escenarios donde se realizan detecciones correctas de los bostezos con frecuencia se detecta momentáneamente que la boca está cerrada y el intervalo que representa el bostezo queda segmentado en múltiples detecciones falsas. También puede ocurrir el caso contrario, donde se detecta momentáneamente que la boca está abierta y se contabiliza un nuevo bostezo.

Experimento 2: Zona gris

El siguiente experimento tiene como objetivo valorar la calidad del sistema una vez se ha integrado el mecanismo que introduce una zona de incertidumbre para las mediciones, descrito en la sección 4.6.2.2.

Se pondrá especial atención en el impacto que este nuevo parámetro tiene sobre las falsas detecciones.

Procedimiento

Para cada uno de los métodos de medición a valorar, se llevará a cabo una exploración exhaustiva del parámetro que optimiza los resultados. Una vez se han determinado estos valores óptimos, se llevará a cabo una comparación de la calidad de los diferentes métodos implementados.

Para cada método, se explorará el intervalo de valores $[0, 1]$ para el *threshold* estático (th) y el rango $[0, 0.4]$ para la zona gris (gz).

Resultados

En las figuras 45 y 46 se puede observar la evolución de la calidad de las respuestas según los valores de los dos parámetros a explorar. Las gráficas de la izquierda presentan la evolución de la *f1-score* para todo el espacio de valores de parámetros que se han explorado, mientras que las gráficas de la derecha muestran una versión ampliada de las regiones donde la *f1-score* es más alta.

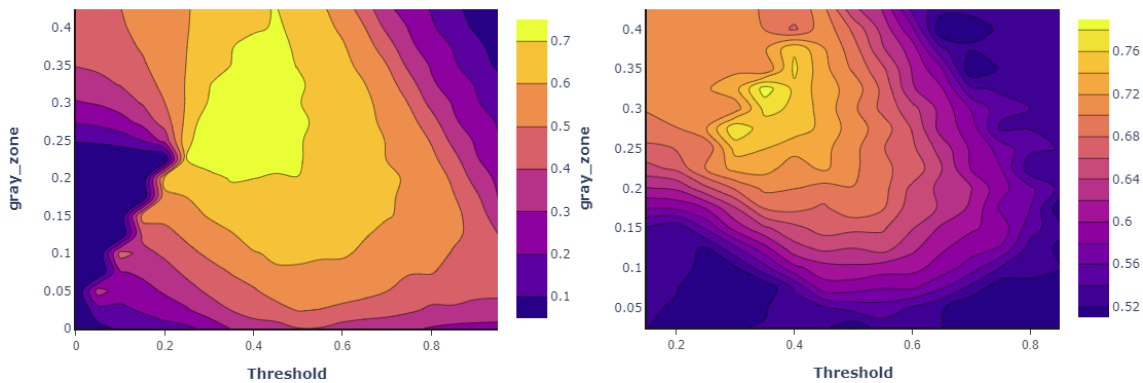


Figura 45: Evolución de las respuestas para el método 1

Fuente: Elaboración propia

Por otro lado, en la tabla 9 se pueden ver los parámetros que optimizan los resultados de cada método, junto con las métricas que describen la calidad de sus respuestas.

Conclusiones

Observando los resultados de la tabla 9, se puede llegar a la conclusión de que la introducción de la zona gris comporta una mejora significativa en la precisión de los modelos, y por lo tanto, en la calidad general de sus respuestas.

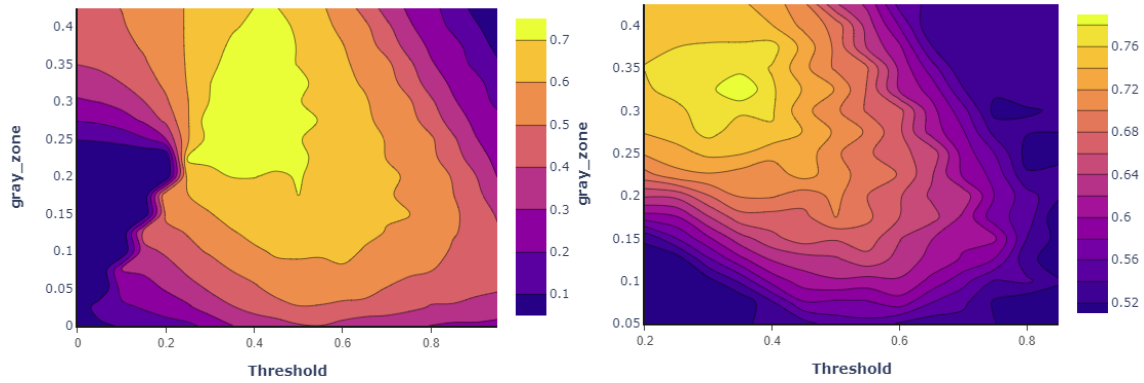


Figura 46: Evolución de las respuestas para el método 2

Fuente: Elaboración propia

| Método | Parámetros óptimos | Precisión | Exhaustividad | F1-Score |
|--------|-----------------------|-----------|---------------|----------|
| 1 | th = 0.35, gz = 0.325 | 0.76 | 0.82 | 0.79 |
| 2 | th = 0.35, gz = 0.325 | 0.78 | 0.82 | 0.8 |

Tabla 9: Rendimiento de los diferentes métodos implementados

Fuente: Elaboración propia

No obstante, pese a que la reducción de las falsas detecciones ha sido muy significativa, esto ha comportado una reducción de la exhaustividad y, por lo tanto, de los falsos negativos. Es decir, a causa de este nuevo parámetro se han dejado de detectar algunos bostezos reales.

Por lo tanto, sería interesante explorar nuevos mecanismos para tratar de reducir las falsas detecciones junto con valores más bajos de zona gris, donde la exhaustividad se vea menos afectada.

Experimento 3: Suavizado de las mediciones

El siguiente experimento tiene como objetivo valorar el impacto que tiene la introducción de un suavizado de las mediciones en el sistema, descrito con más detalle en la sección 4.6.2.2.

De nuevo, se pondrá especial atención a la evolución de la precisión de los modelos.

Procedimiento

Para cada uno de los métodos de medición a valorar, se llevará a cabo una exploración exhaustiva del parámetro que optimiza los resultados. Una vez se han determinado estos valores óptimos, se llevará a cabo una comparación de la calidad de los diferentes métodos implementados.

Para cada método, se explorará el intervalo de valores [0, 1] para el *threshold* estático (th), el rango [0, 0.4] para la zona gris (gz) y el intervalo [0, 100] para los *frames* de suavizado (m).

Resultados

En la tabla 10 se presentan los parámetros que optimizan los resultados de cada método, junto con las métricas que describen la calidad de sus respuestas.

| Método | Parámetros óptimos | Precisión | Exhaustividad | F1-Score |
|--------|-------------------------------|-----------|---------------|----------|
| 1 | th = 0.35, gz = 0.075, m = 60 | 0.97 | 0.89 | 0.92 |
| 2 | th = 0.35, gz = 0.15, m = 40 | 0.94 | 0.91 | 0.93 |

Tabla 10: Rendimiento de los diferentes métodos implementados

Fuente: Elaboración propia

Conclusiones

En la tabla 10 se puede observar que la precisión de los métodos ha vuelto a aumentar de manera significativa, junto con el valor de la exhaustividad.

Viendo estos resultados, se puede afirmar con seguridad que la introducción del suavizado de las mediciones ha resuelto las dos fuentes del problema de las falsas detecciones que el sistema inicial presentaba, sin perjudicar la detección de los bostezos reales que el sistema inicial ya detectaba correctamente.

Por un lado, la introducción de la zona gris junto con el suavizado ha reducido las segmentaciones, que estaban siendo causadas por falsas detecciones momentáneas. Por otro lado, el suavizado también ha tenido impacto sobre las detecciones de bostezos en escenarios donde los sujetos están hablando, puesto que la detección ahora requiere un intervalo prolongado donde el MAR de la boca sea alto, y en el caso del habla, la boca se abre y cierra de manera intermitente.

No obstante, el sistema sigue siendo incapaz de discernir entre las carcajadas y los bostezos, puesto que no se ha dado con ningún mecanismo que pueda ayudar a su distinción.

Sin embargo, pese a que estas detecciones son evidentemente erróneas, se puede argumentar que una persona riendo a carcajadas no se encuentra atenta a la carretera y, por lo tanto, no es del todo negativo que estas detecciones contribuyan a determinar que el sujeto necesita una alerta para volver a centrar su atención en la vía.

En conclusión, los sistemas implementados presentan unos resultados satisfactorios para el objetivo del sistema y, puesto que ambos métodos presentan un comportamiento muy similar, se ha decidido escoger el método 2, basado en el EAR euclidiano, ya que presenta la ventaja de que es invariante a las variaciones en el ángulo de *Roll*.

6.1.3 Detección de cabeceos sobre secuencias de vídeo

Dataset utilizado

De manera análoga al experimento que se ha realizado para la detección de los bostezos, se utilizará el dataset NTHUDDD para la valoración de los resultados, puesto que contiene anotaciones que indican los intervalos temporales donde los sujetos se encuentran dando cabeceos, haciendo movimientos laterales de cabeza o simplemente mirando al frente.

Target

El *target* del *dataset* son los cabeceos, presentados como intervalos de *frames* donde la boca se encuentra muy abierta de manera prolongada.

Algoritmo de valoración

El algoritmo de valoración utilizado es el mismo que se introduce en la sección 6.1.1 para los parpadeos, aplicado a los intervalos de cabeceo.

Métodos a valorar

Los métodos a valorar se corresponden con los que se han presentado en la sección 4.6.1.3

- **Método 1:** Método basado en *deep learning*
- **Método 2:** Método basado en el enfoque geométrico

Experimento 1: Sistema inicial

Procedimiento

Para cada uno de los métodos de medición a valorar, se llevará a cabo una exploración exhaustiva del parámetro que optimiza los resultados. Una vez se han determinado estos valores óptimos, se llevará a cabo una comparación de la calidad de los diferentes métodos implementados.

Para cada método, Para cada método, se explorará el rango de valores [0, 100] para el *threshold* estático (th).

Resultados

En la figura 47 se puede observar la evolución de la *f1-score* respecto al valor del *threshold* estático para los métodos de medición planteados.

Por otro lado, en la tabla 11 se pueden ver los parámetros que optimizan los resultados de cada método, junto con las métricas que describen la calidad de sus respuestas.

| Método | Parámetros óptimos | Precisión | Exhaustividad | F1-Score |
|--------|--------------------|-----------|---------------|----------|
| 1 | Th = 70 | 0.42 | 0.86 | 0.56 |
| 2 | Th = 45 | 0.22 | 0.74 | 0.34 |

Tabla 11: Rendimiento de los diferentes métodos implementados

Fuente: Elaboración propia

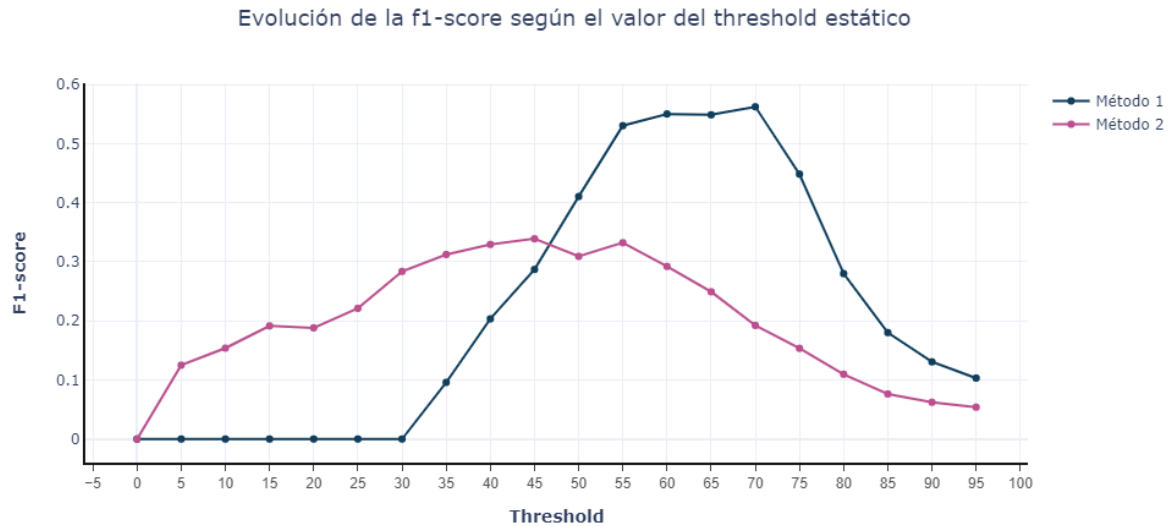


Figura 47: Evolución de las respuestas según el valor del *threshold* estático

Fuente: Elaboración propia

Conclusiones

Como se puede ver en la tabla 11, los sistemas iniciales proporcionan unos resultados bastante mediocres a causa de la baja precisión de la que sufren.

Viendo que la exhaustividad es bastante más alta, los sistemas presentan un comportamiento muy similar al que tenían los algoritmos de detección de bostezos, es decir, se realizan muchas detecciones de cabeceos que no son correctas.

Teniendo en cuenta los resultados vistos en los experimentos para la detección de los bostezos y, tras analizar las respuestas de los sistemas actuales, se ha llegado a la conclusión de que la causa principal de las falsas detecciones vuelve a ser la segmentación de los intervalos de detección.

Cabe destacar también que el método 2 presenta un rendimiento significativamente inferior al del método 1, debido a que realiza muchas más falsas detecciones. La causa de esta diferencia es, posiblemente, que las mediciones del segundo método son mucho más inestables, dando lugar a más detecciones erróneas del estado de la inclinación de la cabeza.

Experimento 2: Zona gris

Procedimiento

Para cada uno de los métodos de medición a valorar, se llevará a cabo una exploración exhaustiva del parámetro que optimiza los resultados. Una vez se han determinado estos valores óptimos, se llevará a cabo una comparación de la calidad de los diferentes métodos implementados.

Para cada método, se explorará el intervalo de valores $[0, 100]$ para el *threshold* estático (th) y el intervalo $[0, 25]$ para la zona gris (gz).

Resultados

En las figuras 48 y 49, se puede observar la evolución de la *f1-score* según los parámetros descritos en el apartado anterior. De nuevo, las gráficas situadas a la derecha presentan una versión ampliada de las regiones donde la *f1-score* es más alta.

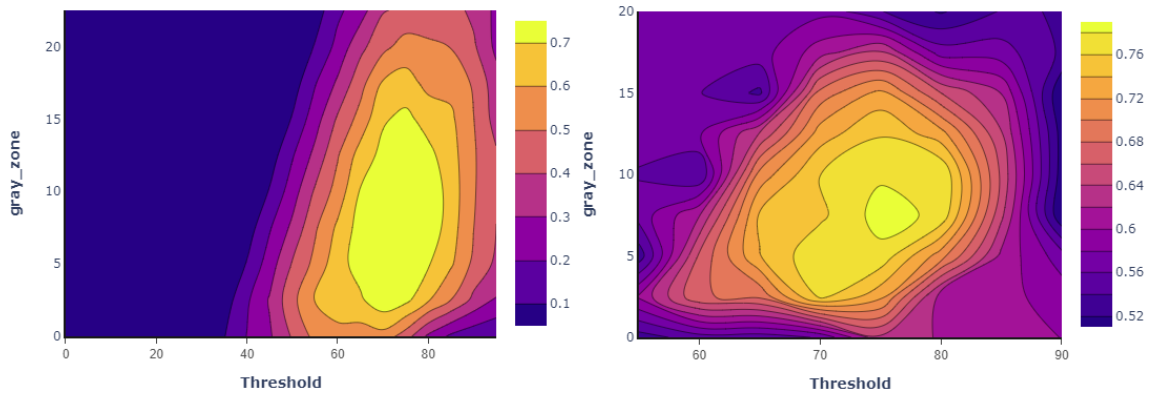


Figura 48: Evolución de los parámetros para el método 1

Fuente: Elaboración propia

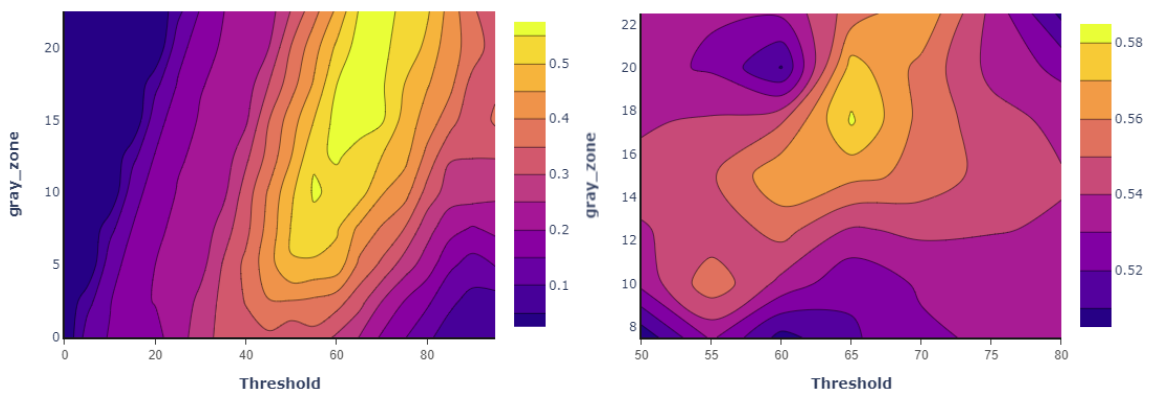


Figura 49: Evolución de los parámetros para el método 2

Fuente: Elaboración propia

Por otro lado, en la tabla 12 se presentan los parámetros que optimizan los resultados de cada método, junto con las métricas que describen la calidad de sus respuestas.

| Método | Parámetros óptimos | Precisión | Exhaustividad | F1-Score |
|--------|--------------------|-----------|---------------|----------|
| 1 | Th = 75, gz = 7.5 | 0.81 | 0.77 | 0.79 |
| 2 | Th = 65, gz = 17.5 | 0.47 | 0.76 | 0.58 |

Tabla 12: Rendimiento de los diferentes métodos implementados

Fuente: Elaboración propia

Conclusiones

Observando los resultados de la tabla 12, se puede afirmar que la introducción de la zona gris ha impactado a los sistemas actuales de una manera muy similar a como se ha podido ver en los experimentos relacionados con la detección de bostezos.

La precisión de ambos métodos ha mejorado significativamente, y cabe mencionar que la diferencia de rendimiento entre los dos se ha mantenido. Por un lado, se puede ver que el primer método tiene un comportamiento equilibrado, mientras que el segundo sigue presentando muchas falsas detecciones, lo cual induce a pensar que la causa de esta diferencia es, en efecto, la inestabilidad del método.

Experimento 3: Suavizado de las mediciones

El siguiente experimento tiene como objetivo valorar el impacto que tiene la introducción de un suavizado de las mediciones en el sistema, descrito con más detalle en la sección 4.6.2.2.

Procedimiento

Para cada uno de los métodos de medición a valorar, se llevará a cabo una exploración exhaustiva del parámetro que optimiza los resultados. Una vez se han determinado estos valores óptimos, se llevará a cabo una comparación de la calidad de los diferentes métodos implementados.

Para cada método, se explorará el intervalo de valores $[0, 100]$ para el *threshold* estático (*th*), el rango $[0, 25]$ para la zona gris (*gz*) y el intervalo $[0, 100]$ para los *frames* de suavizado (*m*).

Resultados

En la tabla 13 se presentan los parámetros que optimizan los resultados de cada método, junto con las métricas que describen la calidad de sus respuestas.

| Método | Parámetros óptimos | Precisión | Exhaustividad | F1-Score |
|--------|---------------------------|-----------|---------------|----------|
| 1 | Th = 70, gz = 2.5, m = 5 | 0.88 | 0.76 | 0.82 |
| 2 | Th = 70, gz = 7.5, m = 15 | 0.71 | 0.77 | 0.74 |

Tabla 13: Rendimiento de los diferentes métodos implementados

Fuente: Elaboración propia

Conclusiones

En la tabla 13 se puede observar que la precisión de los métodos ha vuelto a aumentar de manera significativa, destacando esta mejora principalmente para el método 2.

Esto tiene sentido, puesto que las mediciones proporcionadas por el método 2 son mucho más inestables que las del método 1, y por lo tanto, un suavizado permite reducir el impacto de las mediciones erróneas que ocurren de manera momentánea.

Por otro lado, la exhaustividad del método 1 es bastante baja, y esto puede deberse a que los mecanismos introducidos para reducir las falsas detecciones estén haciendo más difícil la detección de dos cabeceos consecutivos.

Experimento 4: *Threshold* dinámico

El siguiente experimento tiene como objetivo valorar la calidad del sistema una vez se han integrado los mecanismos que computan un *threshold* dinámico, descritos en la sección 4.6.2.1, para así determinar si estos mecanismos proporcionan una mejora de rendimiento significativa.

Procedimiento

Para cada uno de los métodos a valorar, se llevará a cabo una exploración exhaustiva de los parámetros que optimizan los resultados y, una vez se hayan encontrado estos valores, se realizará una comparación de los diferentes métodos para determinar el que proporciona mejores respuestas.

Para parámetro de *threshold_perc*, se explorará el rango de valores de [0.2, 0.8], para el parámetro *n* se explorará el rango [0, 1800], para la zona gris (*gz*) el intervalo [0, 0.25] y para el parámetro *m* el rango [0, 100].

Resultados

En la tabla 14 se presentan los parámetros que optimizan los resultados de cada método, junto con las métricas que describen la calidad de sus respuestas.

| Método | Parámetros óptimos | Precisión | Exhaustividad | F1-Score |
|--------|--|-----------|---------------|----------|
| 1 | th_perc=0.8, gz = 0.08, m = 5, n = 1200 | 0.95 | 0.77 | 0.85 |
| 2 | th_perc=0.8, gz = 0.13, m = 10, n = 1200 | 0.73 | 0.84 | 0.78 |

Tabla 14: Comparativa del rendimiento de los métodos implementados

Fuente: Elaboración propia.

Conclusiones

Finalmente, en la tabla 14 se puede observar que la calidad de detección de los métodos ha aumentado ligeramente. Esto tiene sentido, puesto que el nuevo sistema de *threshold* dinámico se adapta al ángulo de *pitch* que suelen tener los conductores y permite detectar más fácilmente las variaciones en la inclinación de la cabeza.

Respecto a la elección del método, durante el transcurso de los experimentos el primer método ha superado con creces los resultados del segundo, puesto que las mediciones que proporciona son mucho más estables. Debido a esto, se ha decidido utilizar este método para llevar a cabo la detección de los cabeceos en el sistema.

Por otro lado, el método 1 presenta una exhaustividad bastante baja. Tras analizar los vídeos en los que se producían las falsas detecciones, se ha llegado a la conclusión de que la mayoría de estos falsos negativos se producen porque no se ha podido realizar una detección de la cara de los sujetos, a causa de una inclinación muy exagerada en los cabeceos.

Los resultados obtenidos resultan satisfactorios para los objetivos del proyecto, pese a que existen vías de trabajo que recorrer para tratar de optimizar el funcionamiento de los algoritmos.

6.2 Experimentos relacionados con los modelos de detección del cansancio

En este apartado se presentan los diferentes experimentos que se han llevado a cabo para determinar la combinación de parámetros, características y arquitectura de los modelos que optimiza la resolución del problema definido por el proyecto.

Dataset utilizado

Se utilizará el *dataset* de NTHUDDD[8] para valorar la calidad de los modelos a implementar, cuya estructura y tratamiento quedan descritos de manera detallada en la sección 5.5.2.

Target

El *target* de los experimentos es el estado del conductor, presentado como una variable *booleana* que describe el estado de este en cada *frame* de los diferentes vídeos que conforman el *dataset*.

El estado de los sujetos a analizar tiene dos valores posibles, **alerta** o **cansado**.

Por lo tanto, se quiere tratar un problema de clasificación binaria, donde el estado de cansancio representa la clase positiva y el estado de alerta la negativa.

Algoritmo de valoración

Para valorar los resultados de los clasificadores, se realizará una comparación *frame a frame* de las predicciones de los modelos con la *ground-truth* que propone el *dataset*. Cada *frame* se corresponde con una predicción del clasificador y un valor de la *ground-truth*.

A partir de esta comparación se extraerá la matriz de confusión de las predicciones y, a partir de esta, las métricas de **exactitud, precisión, exhaustividad y *f1-score***.

Este conjunto de métricas no solo permite determinar la calidad general de las predicciones que realizan los modelos, sino que también permite realizar una valoración del rendimiento de los clasificadores para cada uno de los valores del *target*.

Clasificadores a valorar

Los algoritmos de clasificación a valorar se han presentado en la sección 4.7.

- **LightGBM** – Se trata del modelo que implementa el *gradient boosting* de árboles.
- **Regresión logística**
- **KNN**
- **SVM lineal**

Consideraciones previas

Antes de presentar los experimentos, es importante entender las consecuencias que pueden acarrear las predicciones erróneas para cada una de las clases del problema a tratar.

En este caso el problema es de clasificación binaria, y, por lo tanto, existen dos escenarios donde las predicciones son erróneas. En el primero de ellos, la persona a analizar se encuentra alerta,

pero el modelo predice que padece síntomas de somnolencia, mientras que en el segundo sucede lo contrario, el conductor se encuentra cansado y supone un peligro para la carretera, pero el modelo determina que no lo está y no se lleva a cabo ninguna acción.

Asumiendo que el predictor forma parte de un sistema que alerta al conductor de manera periódica cuando se detecta que está cansado, las consecuencias de que el primer escenario ocurra son bastante leves, resultando en una leve distracción del conductor. Sin embargo, las consecuencias de que el segundo escenario suceda son bastante más graves, puesto que no se está tomando ninguna acción para tratar de alertar a un conductor muy susceptible a tener un accidente.

Por lo tanto, se valorarán de manera más positiva aquellos clasificadores que maximicen las detecciones correctas del estado de cansancio. Es decir, los clasificadores que resuelven mejor el problema son aquellos que tienen una **exhaustividad del estado del cansancio alta** y que maximizan la exactitud general de las predicciones.

Experimento 1: Selección del mejor conjunto de características

Este experimento tiene como objetivo aportar visibilidad acerca del impacto que tiene cada variable sobre las respuestas del modelo, para así determinar si existe alguna característica prescindible o que perjudica el rendimiento de este.

Procedimiento

Se ha decidido utilizar el modelo de **LightGBM** para poder valorar el impacto de las variables del modelo, puesto que proporciona métodos que permiten obtener una visión global de este impacto.

Para iniciar el experimento, se entrenará un clasificador con todas las características que se han presentado en la sección 4.2, sin modificar los valores predeterminados de sus hiperparámetros y se analizará el impacto de cada variable sobre los resultados.

Finalmente, se eliminarán las características que no parezcan relevantes y se valorará la evolución del rendimiento del modelo resultante para determinar si realmente tienen importancia.

Resultados

El rendimiento del clasificador inicial se presenta en la tabla 15.

| Clasificador | Exactitud | Precisión | Exhaustividad | F1-Score |
|--------------|-----------|-----------|---------------|----------|
| LightGBM | 0.83 | 0.75 | 0.92 | 0.83 |

Tabla 15: Calidad de las respuestas del clasificador inicial

Fuente: Elaboración propia

En la figura 50 se presenta el impacto que tienen las diferentes variables sobre los resultados del modelo. En esta figura se pueden ver los distintos valores de cada variable para cada una de las observaciones del *dataset* de *test* que se ha utilizado. Los puntos que se encuentran a la izquierda influyen al modelo para que realice una predicción positiva, mientras que los puntos que se encuentran a la derecha hacen lo mismo para una predicción negativa.

En la tabla 16 se presenta el rendimiento del nuevo clasificador, una vez se han eliminado las variables que no parecen tener impacto sobre el modelo.

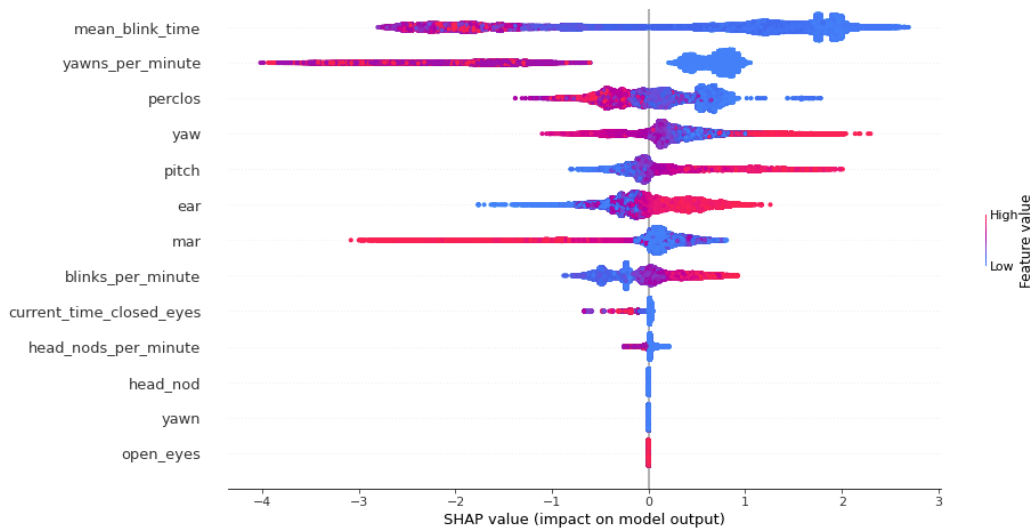


Figura 50: Importancia de las variables sobre las respuestas del modelo.

Fuente: Elaboración propia

| Clasificador | Exactitud | Precisión | Exhaustividad | F1-Score |
|---------------------|------------------|------------------|----------------------|-----------------|
| LightGBM | 0.83 | 0.75 | 0.92 | 0.83 |

Tabla 16: Calidad de las respuestas del clasificador tras eliminar las variables irrelevantes

Fuente: Elaboración propia

Conclusiones

En la figura 50 se puede observar que las variables que informan acerca del estado de las regiones de interés tienen un impacto nulo en las respuestas que proporciona el modelo. Lo más probable es que estas características estén siendo eclipsadas por el resto de variables y la información que aportan resulte redundante para el modelo.

En la tabla 16 se puede comprobar que el rendimiento del clasificador no se ha visto alterado al eliminar estas variables, y, por lo tanto, se ha decidido descartarlas.

Por otro lado, los resultados presentados en la figura 50 confirman que las características se han obtenido correctamente y que el *dataset* muestra una distinción clara de los comportamientos de los sujetos entre las dos clases que se modelan, puesto que en la mayoría de las predicciones positivas del modelo, características como la frecuencia de los bostezos o cabeceos y el MAR presentan valores altos, mientras que en las predicciones negativas predominan los valores más bajos. Lo mismo sucede para variables como el EAR en el sentido contrario.

De la figura 50 también se pueden extraer las conclusiones de que las variables más importantes para el clasificador son el **tiempo medio de parpadeo**, la **frecuencia de bostezo** y el **perclos**.

Experimento 2: Comparación de los diferentes tipos de clasificadores

El siguiente experimento tiene como objetivo valorar la calidad de las respuestas de los algoritmos, así como el tiempo de entrenamiento y predicción que estos ofrecen.

Procedimiento

Se compararán los modelos presentados en la sección 6.2, utilizando el conjunto de características por el que se ha optado en el experimento anterior.

Para realizar una medición más precisa de los tiempos de predicción y entrenamiento se llevarán a cabo 5 ejecuciones y se computará la media, además de la desviación estándar.

Resultados

En las tablas 17 y 18 se puede ver una comparación de la calidad de las respuestas de los diferentes modelos, así como una comparación de los tiempos de entrenamiento y de predicción sobre una única observación.

| Clasificador | Exactitud | Precisión | Exhaustividad | F1-Score |
|---------------------|-----------|-----------|---------------|----------|
| Regresión Logística | 0.84 | 0.78 | 0.94 | 0.84 |
| KNN | 0.81 | 0.73 | 0.90 | 0.80 |
| SVM | 0.84 | 0.75 | 0.94 | 0.83 |
| LightGBM | 0.84 | 0.75 | 0.93 | 0.83 |

Tabla 17: Calidad de las respuestas de cada clasificador

Fuente: Elaboración propia

| Clasificador | Tiempo entrenamiento (s) | Tiempo predicción (ms) |
|---------------------|--------------------------|------------------------|
| Regresión Logística | 2.2 ± 1 | 0.4 ± 0.06 |
| KNN | 37 ± 4.3 | 0.6 ± 0.12 |
| SVM | 158 ± 12.6 | 1 ± 0.09 |
| LightGBM | 2.5 ± 1.2 | 0.3 ± 0.05 |

Tabla 18: Tiempo de entrenamiento y predicción de cada clasificador

Fuente: Elaboración propia

Conclusiones

Como se ha podido observar en las tablas presentadas en la sección de resultados, todos los modelos presentan un rendimiento bastante similar, alrededor de un 84 % de exactitud, junto con una exhaustividad bastante alta.

Esta diferencia mínima de rendimiento se debe a que los datos presentan distribuciones muy distinguidas dependiendo de la clase a la que pertenecen, lo cual comporta que el problema de clasificación resulta sencillo para los algoritmos.

Por otro lado, resulta sorprendente ver que el tiempo de entrenamiento de los modelos de regresión logística y de LightGBM es muy reducido. Esto refuerza la teoría de que los datos son fáciles de clasificar.

Puesto que el rendimiento de los modelos es muy similar, se ha utilizado el tiempo de predicción y la explicabilidad de los modelos como criterios para decidir cual resulta más conveniente para los

intereses del proyecto. Por estas razones se ha decidido escoger el modelo de **LightGBM**, ya que ofrece métodos que proporcionan información acerca de la importancia de las características y que permiten explicar como se realizan las predicciones.

Experimento 3: Impacto del proceso de fine-tuning

Este experimento tiene como objetivo llevar a cabo una búsqueda exhaustiva de los hiperparámetros del modelo para optimizar los resultados que proporciona. En la sección 4.8.4 se ha realizado una descripción más detallada del proceso.

Procedimiento

Para llevar a cabo el experimento, se establecerán unos rangos de valores para cada uno de los hiperparámetros que se quieran valorar y se escogerá la combinación que proporcione la mayor *performance*. El *fine-tuning* se lleva a cabo mediante la librería de *Optuna*.

Finalmente, se realizará una comparación de los resultados presentados por el modelo obtenido en el experimento 2 y el modelo óptimo que se ha encontrado.

Resultados

En la tabla 19 se puede ver una comparación de la diferencia de rendimiento que presentan el modelo base del experimento anterior y el modelo que se ha tratado de optimizar mediante la exploración de hiperparámetros.

| Clasificador | Exactitud | Precisión | Exhaustividad | F1-Score |
|---------------|-----------|-----------|---------------|----------|
| Baseline | 0.84 | 0.75 | 0.93 | 0.83 |
| Modelo óptimo | 0.84 | 0.75 | 0.92 | 0.83 |

Tabla 19: Comparativa del modelo base con el modelo optimizado

Fuente: Elaboración propia

Conclusiones

Tras llevar a cabo la exploración y analizar los nuevos resultados, se ha llegado a la conclusión de que el proceso de *fine-tuning* no ha aportado una mejora significativa al rendimiento del modelo.

Esto refuerza la teoría de que el problema de clasificación planteado por el *dataset* no resulta muy complicado y, por lo tanto, no resulta necesario realizar un *fine-tuning*. Así mismo, el hecho de que todos los modelos presenten un rendimiento similar y que el *fine-tuning* no haya resultado efectivo parece indicar que la causa de este error se encuentra en la naturaleza de los datos, puesto que los sistemas de obtención de características presentan un cierto grado de error.

Valoración final de los resultados

En este apartado se llevará a cabo una valoración del rendimiento del método escogido y se tratará de explicar las posibles causas del error que presenta, así como de generar un entendimiento de la naturaleza de sus respuestas.

En primer lugar, es necesario presentar un conjunto de visualizaciones que describen la naturaleza de las predicciones presentadas por el modelo.

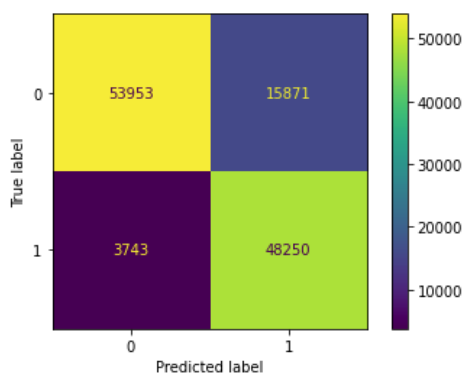


Figura 51: Matriz de confusión de las predicciones del modelo sobre el *dataset* de *test*

Fuente: Elaboración propia

| Clasificador | Exactitud | Precisión | Exhaustividad | F1-Score |
|--------------|-----------|-----------|---------------|----------|
| Modelo final | 0.84 | 0.75 | 0.93 | 0.83 |

Tabla 20: Calidad de las respuestas del modelo que se ha escogido

Fuente: Elaboración propia

En la figura 51 se puede observar la matriz de confusión de las predicciones del modelo sobre el *dataset* de *test*. Se puede destacar que el modelo se inclina más a predecir que los sujetos padecen síntomas de fatiga, llegando a presentar una exhaustividad del 93 %.

Como ya se ha planteado en la introducción de los experimentos, es preferible que los modelos tengan una exhaustividad alta, puesto que eso supone que existen pocos escenarios donde el cansancio de los conductores no se consigue detectar. Esto supone una pérdida de precisión en los escenarios donde los sujetos no están cansados, pero la diferencia significativa entre la penalización de los fallos para cada una de las clases justifica este desequilibrio en las predicciones.

Respecto a las causas del error en las predicciones, en la figura 53 se puede observar un análisis del error que presenta el modelo para cada sujeto del *dataset* de *test*. Tras identificar un rendimiento muy inferior en algunos de los sujetos, y realizar un análisis profundo de los vídeos, se ha concluido que la fuente de error reside en el proceso de extracción de las características.

Más concretamente, en algunos de los sujetos la detección de los puntos faciales no es nada precisa, mientras que en otros casos los métodos de detección no realizan una extracción correcta de la información, identificando carcajadas como bostezos, por ejemplo.

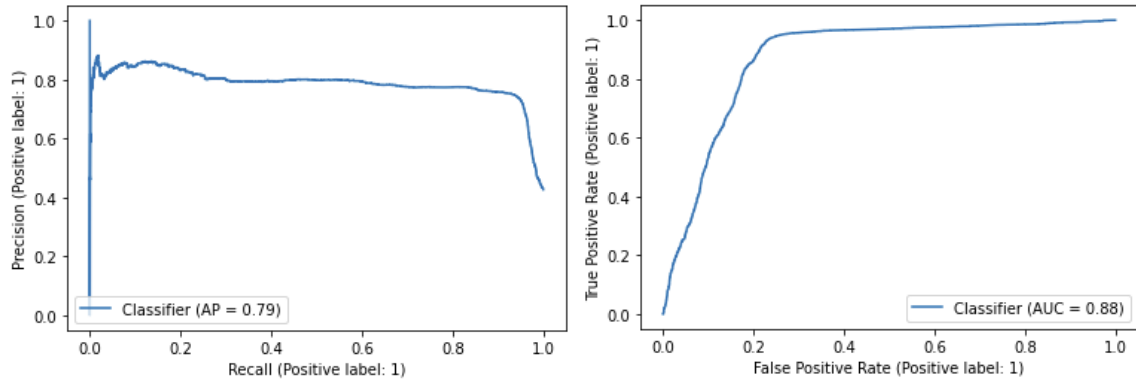


Figura 52: Curvas de precision-exhaustividad y ROC

Fuente: Elaboración propia

Evolución de la exactitud según el sujeto a analizar

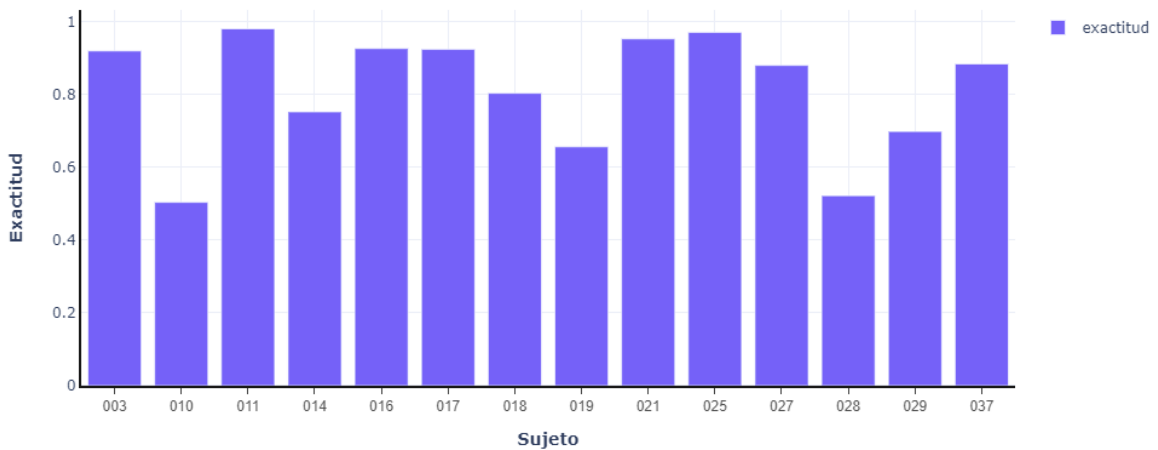


Figura 53: Rendimiento del clasificador final para los diferentes sujetos del *dataset* de *test*

Fuente: Elaboración propia

Teniendo todo esto en cuenta, el sistema que se ha desarrollado tiene mucho potencial de mejora, y su rendimiento se encuentra limitado por los mecanismos de extracción de características y la naturaleza de los datos actuales.

Informe de Sostenibilidad

Es importante realizar un análisis de sostenibilidad para poder justificar y demostrar la viabilidad del proyecto. De nada servirá plantear una idea cuyo impacto medioambiental o social es demasiado grave o cuya propuesta de negocio resulta demasiado arriesgada como para poder ser llevada a la práctica. Este análisis se realizará sobre tres dimensiones: económica, ambiental y social.

Para realizar este análisis se han tenido en cuenta las preguntas propuestas en la matriz de sostenibilidad, además de un conjunto de cuestiones adicionales que permiten definir mejor el impacto del proyecto en los tres ámbitos a analizar.

7.1 Estudio del impacto Ambiental

En este apartado se realizará una valoración del impacto ambiental que ha tenido el desarrollo del proyecto, así como una estimación de los recursos que se utilizarán durante la vida útil del mismo y de los riesgos que pueden provocar un aumento de la huella ecológica.

Desarrollo del Proyecto

Como se presenta en la sección 2.2, el único recurso material que se ha necesitado para llevar a cabo el desarrollo del proyecto ha sido un ordenador de prestaciones medias. Teniendo en cuenta que finalmente se han dedicado 399 horas de desarrollo mediante el uso del ordenador, el consumo de la realización de este TFG ha sido de **87.78 kWh**.

Cabe destacar que para tratar de reducir este impacto, se ha utilizado el ordenador única y exclusivamente cuando era necesario para realizar el desarrollo y las tareas de diseño y planificación se han realizado minimizando el uso de este. Resulta complicado determinar la reducción que se ha conseguido llevar a cabo, pero se estima que se ha disminuido el uso del ordenador en una cantidad alrededor de las 51 horas, suponiendo una reducción de **11.22 kWh**.

Teniendo en cuenta esto, si tuviera que empezar de nuevo el desarrollo del proyecto utilizaría un portátil como herramienta de trabajo en lugar de un ordenador de escritorio, puesto que el consumo energético es bastante inferior.

Vida Útil

Cabe destacar que el producto final del proyecto se ha diseñado con el objetivo de ser integrado en un dispositivo *hardware*, situado en el interior de los vehículos. No obstante, el diseño de este dispositivo queda fuera del alcance de este TFG y, por lo tanto, resulta difícil realizar una estimación de la huella ecológica que puede presentar.

A pesar de todo esto, resulta muy sencillo identificar las fases de la vida útil del proyecto que tienen impacto sobre el medio ambiente. Estas son la manufacturación del dispositivo, que requiere de recursos materiales, energía y genera residuos; y el final de su vida útil, donde el propio dispositivo se convierte en un residuo que ha de ser desechado.

Además de esto, también se tiene que considerar el impacto ambiental que supondrá el uso de ordenadores para realizar el mantenimiento o las actualizaciones de la aplicación por parte de los programadores contratados para ello.

Por otro lado, se tendrán que realizar muchas pruebas controladas en circuitos para valorar la efectividad de los prototipos desarrollados, lo que comportará un impacto ambiental difícil de medir en estas etapas del proyecto, y que se deberá, principalmente, a las emisiones de los vehículos utilizados y a los costes de manufacturar los diferentes prototipos.

Globalmente, el uso del proyecto empeorará la huella ecológica, puesto que se propone la incorporación de un dispositivo adicional en el conjunto de las prestaciones de los vehículos. Así mismo, esta incorporación no supone una reducción en el uso de otros recursos, más bien todo lo contrario, puesto que el consumo de la batería de los coches se verá aumentado, pese a que posiblemente este aumento no sea significativo.

Riesgos

El principal riesgo que se ha identificado y que podría comportar un aumento de la huella ecológica del proyecto es la ocurrencia de averías en los nuevos dispositivos, que provocarían la necesidad de tener que realizar una reparación de estos o, en el peor de los casos, una nueva manufacturación de sistemas de repuesto, que comportaría la generación de residuos adicionales.

7.2 Estudio del impacto Económico

En este apartado se realizará una valoración del coste total que ha supuesto el desarrollo del proyecto, así como una estimación de los costes que el producto final puede tener durante su vida útil y los riesgos que pueden perjudicar la viabilidad económica del proyecto.

Desarrollo del Proyecto

Respecto al desarrollo del proyecto, se realizará una valoración del coste real que ha supuesto la realización de la solución propuesta en este TFG, así como una descripción de las desviaciones que han habido respecto a la planificación inicial.

7.2.1 Costes de personal

Para poder determinar los costes de personal es necesario primero identificar los perfiles que participarán en el desarrollo del proyecto:

- **Líder del proyecto:** Rol compartido entre el autor del TFG y el director del mismo. Se ha encargado de llevar a cabo la gestión del proyecto.
- **Programador:** Papel llevado a cabo individualmente por el autor del TFG. Se ha encargado de desarrollar los algoritmos capaces de determinar la fatiga de los conductores.

En la tabla 21 se presentan los sueldos brutos por hora de cada rol del proyecto y la aportación a la seguridad social, que supone alrededor de un 30 % de dicha cantidad [16]. Por último, también se puede ver el coste total por hora que supone cada papel del proyecto. Estos datos han sido obtenidos de la página web *tecnoempleo* [42].

| Código | Rol | Sueldo Bruto | SS | Coste Total |
|---------------|-------------------|---------------------|-----------|--------------------|
| LP | Líder de proyecto | 21€/h | 6.30€/h | 27.30€/h |
| P | Programador | 15€/h | 4.50€/h | 19.50€/h |

Tabla 21: Costes de los diferentes roles del proyecto

Fuente: Elaboración Propia.

7.2.2 Costes por actividad

Una vez se han definido los costes por hora de cada rol que participa en la realización del proyecto, se han calculado los costes por actividad, realizando una estimación del tiempo que dedicará cada uno de los integrantes del proyecto. En la tabla 22 se puede ver un resumen de las tareas, el tiempo de dedicación de cada rol y su coste resultante.

Cabe destacar que las tareas en las que más de un rol se encuentra implicado se ha considerado que todos los participantes dedican el tiempo total de la tarea para su cumplimiento. Por lo tanto, el coste total de dicha tarea se calcula como la suma de los costes por hora del personal implicado por la duración de la tarea.

| Código | Tarea | Tiempo | Roles | Coste |
|---------------|------------------------------|---------------|--------------|-------------------|
| GP | Gestión del Proyecto | 130h | - | 3.978,00€ |
| GP_1 | Contexto y Alcance | 15h | LP | 409,50€ |
| GP_2 | Planificación temporal | 10h | LP | 273,00€ |
| GP_3 | Presupuesto y Sostenibilidad | 10h | LP | 273,00€ |
| GP_4 | Reuniones | 15h | LP, P | 702,00€ |
| GP_5 | Documentación | 75h | LP | 2047,50€ |
| GP_6 | Presentación | 10h | LP | 273,00€ |
| EP | Estudio Previo | 15h | - | 409,50€ |
| EP_1 | Estudio del estado del arte | 15h | P | 409,50€ |
| MVP | Desarrollo de un MVP | 65h | - | 2.242,50€ |
| MVP_1 | Planificación | 15h | LP, P | 1170,00€ |
| MVP_2 | Diseño | 10h | P | 292,50€ |
| MVP_3 | Implementación | 30h | P | 585,00€ |
| MVP_4 | Validación | 10h | P | 195,00€ |
| PC | Producto completo | 240h | - | 5.557,50€ |
| PC_2 | Sprint 2 | 40h | - | 1248,00€ |
| PC_2_1 | Planificación | 5h | LP, P | 468,00€ |
| PC_2_2 | Diseño | 5h | P | 156,00€ |
| PC_2_3 | Implementación | 20h | P | 429,00€ |
| PC_2_4 | Validación | 10h | P | 195,00€ |
| PC_3 | Sprint 3 | 40h | - | 916,50€ |
| PC_3_1 | Planificación | 5h | LP, P | 234,00€ |
| PC_3_2 | Diseño | 5h | P | 97,50€ |
| PC_3_3 | Implementación | 20h | P | 390,00€ |
| PC_3_4 | Validación | 10h | P | 195,00€ |
| PC_4 | Sprint 4 | 40h | - | 682,50€ |
| PC_4_1 | Planificación | 5h | LP, P | 234,00€ |
| PC_4_2 | Diseño | 5h | P | 39,00€ |
| PC_4_3 | Implementación | 20h | P | 253,50€ |
| PC_4_4 | Validación | 10h | P | 156,00€ |
| PC_5 | Sprint 5 | 40h | - | 604,50€ |
| PC_5_1 | Planificación | 5h | LP, P | 234,00€ |
| PC_5_2 | Diseño | 5h | P | 39,00€ |
| PC_5_3 | Implementación | 20h | P | 175,50€ |
| PC_5_4 | Validación | 10h | P | 156,00€ |
| PC_6 | Sprint 6 | 40h | - | 994,50€ |
| PC_6_1 | Planificación | 5h | LP, P | 234,00€ |
| PC_6_2 | Diseño | 5h | P | 97,50€ |
| PC_6_3 | Implementación | 20h | P | 468,00€ |
| PC_6_4 | Validación | 10h | P | 195,00€ |
| PC_7 | Sprint 7 | 40h | - | 1111,50€ |
| PC_7_1 | Planificación | 5h | LP, P | 234,00€ |
| PC_7_2 | Diseño | 5h | P | 97,50€ |
| PC_7_3 | Implementación | 20h | P | 507,00€ |
| PC_7_4 | Validación | 10h | P | 273,00€ |
| - | Total | 471h | - | 12.187,50€ |

Tabla 22: Resumen de los costes por actividad del proyecto.

Fuente: Elaboración Propia.

7.2.3 Costes genéricos

Estos costes son calculados de manera genérica puesto que no dependen de las tareas del proyecto.

Recursos Software

El coste de los recursos *Software* utilizados en el proyecto ha sido el siguiente:

- *Overleaf* – Herramienta gratuita.
- *Python* – Puesto que se trata de un lenguaje de programación de código abierto, no ha sido necesario pagar ninguna licencia.
- *git* – Herramienta gratuita.
- *Gantter* – Coste de 5 euros al mes, será necesaria una única suscripción para el líder del proyecto.

| Recurso Software | Coste/mes | Tiempo de uso | Coste total |
|------------------|-----------|---------------|-------------|
| <i>Overleaf</i> | 0€ | 5 meses | 0€ |
| <i>Python</i> | 0€ | 5 meses | 0€ |
| <i>git</i> | 0€ | 5 meses | 0€ |
| <i>Gantter</i> | 5€ | 5 meses | 25€ |

Tabla 23: Costes de los recursos *Software* del proyecto.

Fuente: Elaboración Propia.

Recursos Hardware

Por otro lado, se han calculado las amortizaciones de los dispositivos *Hardware* utilizados, dividiendo el tiempo de uso de los dispositivos entre su tiempo de vida útil.

La amortización de los recursos *Hardware* ha sido la siguiente:

- *Ordenador con prestaciones de gama media-baja* – Puesto que el tiempo de vida útil de un ordenador suele ser de 5 años (60 meses) y supone un coste de 800 euros, la amortización de este recurso ha sido de $(5 * 800€) / 60 = 67€$

| Recurso Hardware | Precio | Vida útil | Tiempo de uso | Amortización |
|------------------|--------|-----------|---------------|--------------|
| <i>Ordenador</i> | 800€ | 60 meses | 5 meses | 67€ |

Tabla 24: Costes de los recursos *Hardware* del proyecto.

Fuente: Elaboración Propia.

Consumo eléctrico

Puesto que un elemento clave en el desarrollo del proyecto ha sido el ordenador, se ha tenido en cuenta el coste de mantenerlo encendido durante las horas en las que se ha llevado a cabo el desarrollo del proyecto.

Dado que el consumo eléctrico del ordenador ha sido de **87.79 kWh**, como se ha presentado en el estudio del impacto ambiental, y que el coste actual de la electricidad es de 0.35861€/kWh [41], el coste total de su uso ha sido de **31,50€**.

Factura de internet

Paralelamente al consumo eléctrico, también se ha tenido en cuenta la factura de internet, puesto que ha sido necesario investigar activamente nuevas estrategias durante la realización del trabajo. Según las facturas del hogar del autor del TFG, el coste de la factura de internet ha sido de unos **50€ mensuales**.

Suma de los costes genéricos

En la tabla 25 se puede ver un resumen de los diferentes costes genéricos descritos en los anteriores apartados, juntamente con la suma de todos estos agregados en un valor total.

| Tipo de Gasto | Coste |
|--------------------------|----------------|
| Recursos <i>Software</i> | 25,00€ |
| Recursos <i>Hardware</i> | 67,00€ |
| Consumo eléctrico | 31,50€ |
| Factura de internet | 50,00€ |
| Total | 173,50€ |

Tabla 25: Resumen de los costes genéricos estimados.

Fuente: Elaboración Propia.

7.3 Coste Total del Proyecto

Una vez identificados todos los costes que ha supuesto el desarrollo del proyecto, se ha realizado una suma agregada de todas las partidas calculadas y se ha determinado que el coste total del proyecto ha sido de **12.361,00€**, como se puede ver en la tabla 26.

| Tipo de Gasto | Coste |
|----------------------|-------------------|
| Costes por actividad | 12.187,50€ |
| Costes genéricos | 173,50€ |
| Total | 12.361,00€ |

Tabla 26: Suma de los costes totales del proyecto.

Fuente: Elaboración Propia.

7.4 Control de Gestión

Para controlar los costes reales del desarrollo del proyecto e identificar los desvíos que han ido surgiendo, se ha realizado un seguimiento mediante los siguientes descriptores numéricos.

- **Desviación de horas dedicadas:**

$$\text{Horas_dedicadas_reales} - \text{Horas_dedicadas_estimadas}$$

- **Desviación de los costes:**

$$\text{Costes_totales_reales} - \text{Costes_totales_estimados}$$

- **Desviación de costes por actividad:**

$$\text{Desviación_de_horas_dedicadas} * \text{coste_real_personal}$$

- **Desviación de costes genéricos:**

$$\text{Costes_genéricos_reales} - \text{Costes_genéricos_estimados}$$

- **Desviación de costes por imprevistos:**

$$\text{Costes_por_imprevistos_reales} - \text{Costes_por_imprevistos_estimados}$$

Al final de cada *sprint* se han calculado los descriptores descritos y se ha valorado el grado de la desviación respecto a los valores estimados inicialmente.

Cabe destacar que el control de las horas dedicadas y de los costes del proyecto se ha llevado a cabo mediante un documento de *Excel*.

7.5 Desviaciones respecto a la planificación inicial

Como ya se ha planteado en la sección 2.5, han habido desvíos entre el tiempo real que se ha invertido para cada tarea y las duraciones que se habían estimado inicialmente. Debido a esto, también se han producido desvíos en los costes por actividad del proyecto.

En la figura 27 se presentan las tareas que han tenido un desvío temporal y la diferencia entre los costes reales y los costes que se habían estimado inicialmente.

Por otro lado, puesto que se ha tenido en cuenta la sostenibilidad económica y ambiental del proyecto, se ha minimizado el uso del ordenador en las tareas de planificación y diseño, en las que se ha utilizado únicamente para llevar a cabo búsquedas de información que resultaban esenciales para poder llevar a cabo las tareas. En estas tareas se ha dedicado, aproximadamente, un 30 % del tiempo a utilizar el ordenador.

Esta reducción del tiempo de uso del dispositivo ha comportado un ahorro de **4€**.

Cabe destacar que el coste que se ha estimado inicialmente se ha acercado bastante al coste final del proyecto. El desvío ha sido de **1204,53€** a causa de que no se han materializado muchos riesgos, es decir, este ha sido causado principalmente por una sobreestimación de los costes de contingencia. El coste final del desarrollo del proyecto ha sido de **12.361,00€**.

| Código | Tarea | Tiempo | Roles Implicados | Coste estimado | Coste real |
|--------|----------------|-------------|------------------|-------------------|-----------------|
| GP_4 | Reuniones | 15h | LP, P | 936,00€ | 702,00€ |
| GP_5 | Documentación | 75h | LP | 1638,00€ | 2047,50€ |
| MVP_1 | Planificación | 25h | LP, P | 702,00€ | 1170,00€ |
| MVP_2 | Diseño | 15h | P | 195,00€ | 292,50€ |
| PC_2_1 | Planificación | 10h | LP, P | 234,00€ | 468,00€ |
| PC_2_2 | Diseño | 8h | P | 97,50€ | 156,00€ |
| PC_2_3 | Implementación | 22h | P | 390,00€ | 429,00€ |
| PC_4_2 | Diseño | 2h | P | 97,50€ | 39€ |
| PC_4_3 | Implementación | 13h | P | 390,00€ | 253,5€ |
| PC_4_4 | Validación | 8h | P | 195,00€ | 156,00€ |
| PC_5_2 | Diseño | 2h | P | 97,50€ | 39€ |
| PC_5_3 | Implementación | 9h | P | 390,00€ | 175,50€ |
| PC_5_4 | Validación | 8h | P | 195,00€ | 156,00€ |
| PC_6_3 | Implementación | 24h | P | 390,00€ | 468,00€ |
| PC_7_3 | Implementación | 26h | P | 390,00€ | 507,00€ |
| PC_7_4 | Validación | 14h | P | 195,00€ | 273,00€ |
| - | Total | 471h | - | 11.388,00€ | +799,50€ |

Tabla 27: Resumen de los desvíos en los costes por actividad.

Fuente: Elaboración Propia.

| Tipo de Gasto | Coste estimado | Coste real |
|----------------------|-------------------|-------------------|
| Costes por actividad | 11.388,00€ | 12.187,50€ |
| Costes genéricos | 177,50€ | 173,50€ |
| Contingencias | 1734,83€ | - |
| Imprevistos | 265,20€ | - |
| Total | 13.565,53€ | 12.361,00€ |

Tabla 28: Comparación de los costes estimados y los costes reales del proyecto.

Fuente: Elaboración Propia.

Vida Útil

De manera análoga a la estimación del impacto ambiental que puede tener el proyecto durante su vida útil, resulta difícil realizar una estimación de los costes económicos puesto que el diseño del dispositivo en el que se integraría el producto final del proyecto queda fuera del alcance de este TFG.

No obstante, resulta muy sencillo identificar los recursos principales que conforman el coste económico total del proyecto durante su vida útil. Estos son los siguientes.

- **Programadores y gestores del proyecto** — Se trata de las personas que se encargarán de realizar el mantenimiento del producto desarrollado durante este proyecto, proporcionando actualizaciones y solucionando los diferentes problemas de funcionamiento a nivel *software* que puedan ocurrir.

Puesto que el proyecto se ha podido llevar a cabo con un programador y un líder de proyecto y se quieren minimizar los costes económicos, se estima que es posible realizar el mantenimiento

del mismo con los mismos perfiles y una dedicación diaria de 4 horas, con la posibilidad de ampliarse en caso de que ocurran imprevistos. Por lo tanto, teniendo en cuenta los costes totales por hora de cada perfil, presentados en la tabla 21, se estima que el coste por hora del mantenimiento es de **46.80€** y el coste mensual es de **4119,40€**

- **Mantenimiento del proyecto en un repositorio** — Se trata del coste que supone mantener el código en un repositorio para poder mantener un control de versiones. Los costes pueden variar dependiendo de las prestaciones contratadas, desde un importe gratuito hasta costes que se encuentran alrededor de los 100€ al mes.

Para tratar de reducir costes, se propone partir de un repositorio gratuito, que se verá sustituido si las circunstancias del proyecto lo exigen.

- **Reparaciones y actualizaciones** — Consiste en el coste derivado de las posibles reparaciones que se tengan que realizar a causa de actualizaciones o averías en el funcionamiento del dispositivo *hardware*.
- **Manufacturación del dispositivo *hardware*** — Se trata del coste que supone la manufacturación inicial del dispositivo.

Estas dos últimas fuentes de costes resultan muy difíciles de cuantificar, puesto que el diseño del dispositivo no se ha valorado en profundidad.

Riesgos

Tras llevar a cabo una reflexión meditada, se han identificado los siguientes escenarios potencialmente perjudiciales para la viabilidad del proyecto.

- **Falta de interés por parte de los potenciales clientes** — Es posible que el producto no despierte el suficiente interés en los potenciales clientes o que se considere que su precio es demasiado elevado para los beneficios que ofrece el producto, lo que podría provocar una diferencia significativa entre las ventas estimadas y las reales. Esto, a su vez, podría suponer un retraso en el tiempo necesario para alcanzar la rentabilidad económica, o en el peor de los casos, esta podría no llegar a alcanzarse.

Para tratar de reducir el riesgo de que este escenario ocurra, se propone llevar a cabo campañas de publicidad que conciencien acerca de las consecuencias terribles que acarrearán los accidentes y que presenten las bondades que comporta la integración de este dispositivo en los vehículos.

- **Presencia de competencia** — Es posible que surjan iniciativas similares si el producto resulta innovador en el sector y despierta interés en los potenciales clientes. Estas iniciativas podrían proponer soluciones *software* más sofisticadas o plantear un producto más asequible económicamente, que podrían implementar otras empresas del sector automovilístico.

De manera similar a la solución planteada para el escenario anterior, se propone llevar a cabo campañas publicitarias que destaquen los beneficios que aporta el proyecto respecto a las soluciones implementadas por la competencia.

7.6 Estudio del impacto Social

Desarrollo del Proyecto

El desarrollo del proyecto ha tenido un impacto muy positivo sobre el autor del mismo en una gran variedad de ámbitos.

En primer lugar, desde un punto de vista profesional, la realización del proyecto ha supuesto la adquisición de conocimiento en muchas disciplinas distintas, de las cuales se puede destacar los ámbitos de Visión por Computador y *Machine Learning*, junto con las áreas de algorítmia y estadística. Finalmente, también se ha ganado mucha experiencia acerca de la gestión de los proyectos, puesto que ha sido necesario mantener un cierto nivel de organización para poder llevar a cabo el desarrollo del trabajo.

Por otro lado, también se han tenido que realizar consideraciones acerca del funcionamiento de los hipotéticos dispositivos *hardware* en los que el sistema sería integrado si la propuesta del proyecto fuera aceptada y puesta en funcionamiento.

En segundo lugar, se ha tenido que realizar un conjunto de reflexiones éticas muy meditadas acerca del impacto que tendría sobre la sociedad la implementación del proyecto en un dispositivo hipotético, integrado en el interior de los vehículos. Se han valorado las bondades que tendría la implementación, pero también se han identificado posibles riesgos o escenarios donde el impacto del proyecto podría ser negativo.

Esta etapa ha consistido en un proceso de crecimiento tanto personal como profesional para el autor del proyecto, en la que se han tenido que tomar decisiones de diseño relevantes de manera muy meditada y afrontar obstáculos de una naturaleza distinta a los que se han encontrado durante el transcurso del grado.

Vida Útil

Respecto al impacto del proyecto sobre la sociedad durante su vida útil, cabe definir primero los colectivos que se ven implicados.

El colectivo que resultaría más beneficiado de que el proyecto se implementara con éxito es toda la sociedad como conjunto, puesto que cualquier individuo que la compone puede verse envuelto en un accidente de tráfico.

Una reducción significativa en el número de accidentes supondría, por un lado, una disminución del número de fallecidos y heridos, reduciendo así potenciales consecuencias psicológicas sobre las víctimas o sus círculos cercanos, el número de situaciones familiares comprometidas, el grado de ocupación en hospitales, posibles daños materiales, etc.

Por otro lado, la congestión del tráfico se vería reducida, dando lugar a una circulación más fluida y causando menos perjuicio a todos los usuarios de la vía.

El potencial impacto social es inmenso, puesto que la vía es un recurso compartido y las consecuencias de los accidentes no solo afectan a las víctimas directas, sino a todos los usuarios de esta y, en general, a un conjunto muy extenso de individuos.

Otro colectivo más reducido que se vería beneficiado de la disminución de los accidentes de tráfico son las compañías de seguros, puesto que tendrían que cubrir menos accidentes.

Pese a que se ha reflexionado sobre ello, no se ha podido identificar ningún colectivo que se vea perjudicado por el proyecto, puesto que su implementación no supone la pérdida de puestos de trabajo ni la vulneración de la privacidad de ningún individuo.

No obstante, cabe admitir que la implementación de este podría llegar a contribuir a la desigualdad social. Esto es debido a que el sistema se ha diseñado con el objetivo de ser manufacturado por las empresas del sector automotriz, y por lo tanto, lo más probable es que el dispositivo hipotético resultante tuviera un precio y únicamente fuera integrado en los vehículos de los individuos dispuestos a pagar por él. Esto comportaría que el proyecto fuera accesible únicamente a las personas con los recursos económicos suficientes.

Riesgos

Pese a todas las bondades que la implementación del proyecto acarrearía sobre la sociedad, cabe admitir que existe un conjunto de riesgos que podrían agravar el impacto negativo del producto final.

En primer lugar, el producto final del proyecto se ha diseñado con el objetivo de alertar a los conductores en caso de detectar indicios de fatiga o somnolencia. Una decisión muy importante que se tendría que llevar a cabo es la naturaleza de esta alerta, puesto que ha de ser lo suficientemente notable como para poder despertar o alertar a una persona, pero tampoco debería suponer una distracción para los conductores en caso de detección errónea.

El primer riesgo a tener en cuenta consiste precisamente en este escenario. Existe la posibilidad de que las detecciones erróneas del sistema puedan suponer una distracción significativa para los conductores y acabar comportando accidentes de tráfico adicionales.

Para reducir el potencial impacto de este riesgo, durante el periodo de diseño del producto sería necesario llevar a cabo una exploración exhaustiva de las distintas posibles estrategias para tratar de alertar a los conductores, minimizando las distracciones que estos pueden sufrir en caso de que las detecciones sean erróneas.

En este proceso de diseño también habría que tener en cuenta la usabilidad del producto para las personas que pueden resultar más sensibles a las alertas, como pueden ser, por ejemplo, las personas mayores, o para aquellos individuos que padecen de alguna discapacidad, como por ejemplo la auditiva.

En segundo lugar, en el caso hipotético de que el producto final fuera capaz de detectar con mucha fiabilidad los indicios de fatiga en las personas y de alertarlos correctamente, los usuarios podrían desarrollar una dependencia hacia el producto que los liberara de la responsabilidad de tener que mantenerse despiertos y que fomentara el mal uso del dispositivo. Como consecuencia de esto, la presencia de conductores con síntomas de somnolencia se podría ver aumentada y la implementación del proyecto tendría el impacto contrario al que se esperaba conseguir inicialmente.

Una respuesta plausible para este escenario hipotético sería la realización de campañas para concienciar a los conductores acerca de los peligros de conducir con síntomas de fatiga y de las consecuencias que comporta el uso irresponsable de los vehículos. También sería necesario educar acerca del uso correcto del producto, aclarando que se trata únicamente de un mecanismo de apoyo y que no pretende influir en la toma de decisiones de sus usuarios.

Conclusiones y trabajo futuro

8.1 Conclusiones

El objetivo principal de este proyecto consistía en la implementación de un sistema capaz de detectar los síntomas de somnolencia y fatiga en los conductores con el fin de reducir los accidentes de tráfico. Tras analizar los resultados que ha proporcionado el modelo final, presentados en la sección 6.2, se puede afirmar con confianza que se ha cumplido con el objetivo propuesto inicialmente, puesto que la precisión de las respuestas es bastante alta para un conjunto de sujetos muy diverso y el modelo presenta una detección muy fiable de los escenarios donde los sujetos padecen de somnolencia.

Pese a esto, cabe destacar que existe una diferencia significativa entre los escenarios representados por el *dataset* utilizado y las situaciones reales en la carretera. Es decir, el conjunto de datos utilizado modela únicamente dos posibles estados para el conductor que presentan comportamientos muy diferenciados; un estado de **alerta**, donde no es necesario que el sistema actúe, y un estado de **somnolencia**, donde el conductor representa un peligro para la carretera. Esta división es muy abrupta, puesto que existen muchos estados intermedios del cansancio para los que debería existir una clase propia en el *dataset*.

Puesto que el rendimiento del sistema desarrollado en este proyecto está limitado a la calidad del *dataset* utilizado, se puede afirmar que el sistema detecta con éxito los casos donde el cansancio de los conductores es muy pronunciado. Sin embargo, es cierto que no se tiene la certeza de que el sistema sea tan acertado en los escenarios donde los síntomas de fatiga son más sutiles, puesto que el *dataset* no los modela.

También cabe destacar que los vídeos del *dataset* presentan una calidad de imagen baja, lo cual ha supuesto una limitación para el rendimiento de los algoritmos de detección, puesto que la librería de Mediapipe tiene un rendimiento muy superior cuando las condiciones de iluminación son buenas y no hay reflejos en estas. Esto ha introducido ruido en los datos de entrenamiento, y por lo tanto, una reducción en el rendimiento final del sistema. Debido a esto, el sistema de extracción de las características es uno de los grandes puntos de mejora del sistema.

A pesar de todo esto, el proyecto desarrollado presenta un potencial inmenso y las sensaciones finales son muy positivas, puesto que sus principales limitaciones actuales provienen del *dataset* utilizado, y existen muchas vías de trabajo futuro que se han quedado en el tintero y que pueden dotar al sistema implementado de una naturaleza más sofisticada.

8.2 Trabajo futuro

En este apartado se exponen las principales vías de desarrollo que se podrían seguir en un futuro para reducir las limitaciones expuestas anteriormente o simplemente dotar al producto final de un mejor rendimiento, que no han podido implementarse en el proyecto por falta de tiempo.

Mejorar el conjunto de características que describe el comportamiento del conductor

Uno de los principales puntos de mejora para el proyecto consiste en la ampliación del conjunto de características utilizadas para modelar el comportamiento de los conductores. Dentro de la infinidad de indicadores que se pueden explorar, los que pueden tener un mayor impacto en el rendimiento del sistema son las siguientes.

- **Características relacionadas con la risa y el habla del conductor** — Consisten en una ampliación del algoritmo de detección de los bostezos y permitirían realizar un modelaje más descriptivo de la actividad de la persona cuando se encuentra alerta.
- **Información acerca de la mirada del conductor** — Esta característica representa la dirección de la mirada del conductor y resultaría de utilidad para determinar si las variaciones en la dirección de esta son frecuentes o no.
- **Información acerca de los movimientos laterales de cabeza** — Consiste en una ampliación del algoritmo de detección de los cabeceos y permite modelar con más detalle la actividad de la persona. Puesto que los conductores han de tener controlados los alrededores del vehículo, este grupo de características permitiría modelar muy bien el estado de alerta.
- **Pulso del conductor** — Como se ha presentado en la sección 4.6.1.4, durante el desarrollo del proyecto se trató de extraer el pulso de los conductores mediante una variedad de algoritmos pero se acabó descartando esta opción por falta de tiempo de implementación. Esta característica resultaría muy útil para modelar desde otra perspectiva la actividad de los conductores.

Por otro lado, el sistema implementado actualmente para la detección de los bostezos únicamente tiene en cuenta la boca, y sería interesante tener en cuenta otros factores o implementar nuevas soluciones capaces de discernir entre los bostezos, la risa y el habla.

Mejorar el procesamiento inicial de las imágenes

Como se ha concluido en la sección 6.2, uno de los principales motivos por los que el producto final del proyecto no alcanza un mejor rendimiento sobre el *dataset* de *test* utilizado es la falta de precisión en la generación de la malla facial sobre algunos de los sujetos.

Esta falta de precisión se debe a que la resolución de estos vídeos es relativamente baja y a que presentan una mala iluminación en las regiones de interés, destacando sobre todo los ojos.

Durante la realización del proyecto se han probado diferentes técnicas de ecualización del histograma de las imágenes con el objetivo de mejorar el contraste y reducir los artefactos causados por la baja calidad de estas. No obstante, no se han obtenido resultados concluyentes.

Se plantea como posible vía de trabajo la valoración de más técnicas que puedan reducir los artefactos presentes en las imágenes del *dataset* y mejorar la calidad de detección de Mediapipe.

Valorar otros *datasets*

Como se ha comentado anteriormente, el rendimiento del sistema actual depende principalmente de la calidad del *dataset* utilizado en el entrenamiento de los clasificadores.

Debido a que el conjunto de datos utilizado actualmente presenta una división muy sencilla de los estados de cansancio y no modela estados intermedios de la fatiga, representa uno de los puntos de mejora más grandes del proyecto. Por lo tanto, puede resultar interesante la creación de un *dataset* propio o la búsqueda de algún conjunto de datos más cercano a la realidad, que permita trabajar con escalas más sofisticadas del cansancio, como por ejemplo la escala de Karolinska[39, Capítulo 47] y que presente vídeos con una resolución más alta.

Encontrar un *dataset* de estas características puede resultar una tarea difícil, pese a que se puede valorar la opción de negociar la creación de este *dataset* mediante una colaboración con alguna empresa del sector automovilístico.

Implementar un filtro de Kalman

Otra iniciativa que puede resultar muy interesante es la aplicación de filtros para incrementar la robustez de las mediciones que proporcionan los diferentes modelos utilizados durante la ejecución del sistema. Esto aportaría una pequeña mejora al rendimiento del modelo que genera las mallas faciales, reduciendo la inestabilidad que a veces presentan las mediciones entre *frames* consecutivos.

Se ha valorado implementar los filtros de Kalman[24], puesto que presentan muchas bondades, pero su complejidad y la escasez de tiempo de implementación no lo han permitido.

Referencias

- [1] What is agile? what is scrum? *En línea*, <https://www.cprime.com/resources/what-is-agile-what-is-scrum/>. [Consulta: 25-02-2022].
- [2] 20minutos.es. Cada muerto en un accidente de tráfico supone unos 860.000 euros para la sociedad. *En línea*, https://imagenes.20minutos.es/files/image_656_370/uploads/imagenes/2008/06/12/827522a.jpg. [Consulta: 25-02-2022].
- [3] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.
- [4] Dyah Anggraeni. Cascade classifier illustration. *En línea*, https://www.researchgate.net/figure/Cascade-classifier-illustration-2_fig2_323057610. [Consulta: 26-02-2022].
- [5] Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov, Karthik Raveendran, and Matthias Grundmann. BlazeFace: Sub-millisecond neural face detection on mobile gpus. *CoRR*, abs/1907.05047, 2019.
- [6] BenMauss. Haar-like features: Seeing in black and white. *En línea*, <https://levelup.gitconnected.com/haar-like-features-seeing-in-black-and-white-1a240caaf1e3>. [Consulta: 26-02-2022].
- [7] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [8] Shang-Hong Lai Ching-Hua Weng, Ying-Hsiu Lai. Driver drowsiness detection via a hierarchical temporal deep belief network. In *Asian Conference on Computer Vision Workshop on Driver Drowsiness Detection from Video*, Taipei, Taiwan. 20–24 November 2016. URL: https://link.springer.com/chapter/10.1007/978-3-319-54526-4_9.
- [9] Wikimedia Commons. k vecinos más próximos. URL: https://es.wikipedia.org/wiki/K_vecinos_m%C3%A1s_pr%C3%B3ximos#/media/Archivo:KnnClassification.svg.
- [10] Wikimedia Commons. Support-vector machine. URL: https://en.wikipedia.org/wiki/File:SVM_margin.png.
- [11] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [12] Pádraig Cunningham and Sarah Jane Delany. k-nearest neighbour classifiers: 2nd edition (with python examples). *CoRR*, abs/2004.04523, 2020.
- [13] United States. Federal Motor Carrier Safety Administration. Technology Division. Perclos: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance. 1998. URL: <https://doi.org/10.21949/1502740>.
- [14] T. Drutarovsky and A. Fogelton. European conference on computer vision workshops. pages 436–448, 2016.

- [15] EBF. Ventajas y desventajas de las metodologías agile (ágiles). *En línea*, <https://ebf.com.es/blog/wp-content/uploads/2019/09/Ventajas-del-uso-de-metodolog%C3%ADas-%C3%A1giles.jpg>. [Consulta: 25-02-2022].
- [16] Expansión.com. ¿cómo se calculan las cotizaciones a la seguridad social? *En línea*, <https://www.expansion.com/economia-para-todos/impuestos/como-se-calculan-las-cotizaciones-a-la-seguridad-social.html>. [Consulta: 13-02-2022].
- [17] Google. Mediapipe solutions. *En línea*, <https://google.github.io/mediapipe/>. [Consulta: 22-05-2022].
- [18] Google. Mediapipe face mesh ar effects. *En línea*, https://google.github.io/mediapipe/images/face_mesh_ar_effects.gif. [Consulta: 22-05-2022].
- [19] Eugene Shih John Guttag Frédo Durand William T. Freeman Hao-Yu Wu, Michael Rubinstein. Eulerian video magnification for revealing subtle changes in the world. *En línea*, <http://people.csail.mit.edu/mrub/evm/>. [Consulta: 27-02-2022].
- [20] Thorsten Hempel, Ahmed A. Abdelrahman, and Ayoub Al-Hamadi. 6d rotation representation for unconstrained head pose estimation, 2022. URL: <https://arxiv.org/abs/2202.12555>, doi:10.48550/ARXIV.2202.12555.
- [21] MICHAEL INGRE, TORBJÖRN ÅKERSTEDT, BJÖRN PETERS, ANNA ANUND, and GÖRAN KECKLUND. Subjective sleepiness, simulated driving performance and blink duration: examining individual differences. *Journal of Sleep Research*, 15(1):47–53, 2006. doi:<https://doi.org/10.1111/j.1365-2869.2006.00504.x>.
- [22] D.J. Schroeder J.A. Stern, D. Boyer. Blink rate as a measure of fatigue: A review. August, 1994. URL: <https://rosap.ntl.bts.gov/view/dot/21415>.
- [23] H. Nasir S. Safie Z. Bakti K. Kadir, M. K. Kamaruddin. A comparative study between lbp and haar-like features for face detection using opencv. *En línea*, <https://www.semanticscholar.org/paper/A-comparative-study-between-LBP-and-Haar-like-for-Kadir-Kamaruddin/3ec98dabbcf440db746d4baf6550e44e12c5ee76>. [Consulta: 26-02-2022].
- [24] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [25] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. 30, 2017. URL: <https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf>.
- [26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.
- [27] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. Mediapipe: A framework for building perception pipelines. *CoRR*, abs/1906.08172, 2019. URL: <http://arxiv.org/abs/1906.08172>, arXiv:1906.08172.

- [28] Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.
- [29] Ioannis Mollas, Grigorios Tsoumakas, and Nick Bassiliades. Lionforests: Local interpretation of random forests through path selection. 11 2019.
- [30] Shahid Mahmood Awan Amina Ismail Mahwish Ilyas; Ahsan Mahmood Muhammad Ramzan, Hikmat Ullah Khan. A survey on state-of-the-art drowsiness detection techniques. *En línea*, <https://ieeexplore.ieee.org/document/8704263>. [Consulta: 28-02-2022].
- [31] NASA. Aircraft rotations. *En línea*, <https://web.archive.org/web/20080704135659/http://www.grc.nasa.gov/WWW/K-12/airplane/rotations.html>. [Consulta: 10-05-2022].
- [32] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Frontiers in neurobotics*, 7:21, 12 2013. doi:10.3389/fnbot.2013.00021.
- [33] OMS. Objetivo 3: Garantizar una vida sana y promover el bienestar para todos en todas las edades. *En línea*, <https://www.un.org/sustainabledevelopment/es/health/#tab-3bee2d0d15e1c8b1c99>. [Consulta: 28-02-2022].
- [34] OpenCV.org. Face detection using haar cascades. *En línea*, https://docs.opencv.org/4.x/haar_features.jpg. [Consulta: 26-02-2022].
- [35] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.
- [36] Joanne Peng, Kuk Lee, and Gary Ingersoll. An introduction to logistic regression analysis and reporting. *Journal of Educational Research - J EDUC RES*, 96:3–14, 09 2002. doi:10.1080/00220670209598786.
- [37] Serena Raju. yawn_eye_dataset_new. *En línea*, <https://www.kaggle.com/serenaraju/yawn-eye-dataset-new>. [Consulta: 25-02-2022].
- [38] Lior Rokach and Oded Maimon. *Decision Trees*, volume 6, pages 165–192. 01 2005. doi:10.1007/0-387-25465-X_9.
- [39] Azmeh Shahid, Kate Wilkinson, Shai Marcu, and Colin Shapiro. *STOP, THAT and one hundred other sleep scales*. 01 2012. doi:10.1007/978-1-4419-9893-4.
- [40] Statista.com. Evolución del número total de accidentes de tráfico en españa entre 2011 y 20202. *En línea*, <https://es.statista.com/estadisticas/892169/numero-de-accidentes-de-carretera-espana/>. [Consulta: 18-03-2022].
- [41] Tarifaluzhora.es. ¿cuánto cuesta el kilovatio hora de luz (kwh) en españa? *En línea*, <https://tarifaluzhora.es/info/precio-kwh>. [Consulta: 13-02-2022].
- [42] Tecnoempleo.com. Informe empleo informática - marzo 2022. *En línea*, <https://www.tecnoempleo.com/informe-empleo-informatica.php>. [Consulta: 13-03-2022].

- [43] Euclides Arcoverde *et al.* Enhanced real-time head pose estimation system for mobile device. *En línea*. [Consulta: 01-06-2022]. URL: https://www.researchgate.net/figure/Orientation-of-the-head-in-terms-of-pitch-roll-and-yaw-movements-describing-the-three_fig1_279291928.
- [44] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. *En línea*, <https://www.merl.com/publications/docs/TR2004-043.pdf>. [Consulta: 26-02-2022].
- [45] Farzeen Zehra, Maha Javed, Darakhshan Khan, and Maria Pasha. Comparative analysis of c++ and python in terms of memory and time. 12 2020. doi:10.20944/preprints202012.0516.v1.