

Detección de somnolencia y síncope en conductores mediante visión artificial

José Manuel Jiménez Berlanga

Máster Universitario en Ingeniería de Telecomunicación
Electrónica

Juan Antonio Ortega Redondo

Carlos Monzo Sánchez

10/06/2021



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Detección de somnolencia mediante visión por computador</i>
Nombre del autor:	<i>José Manuel Jiménez Berlanga</i>
Nombre del consultor/a:	<i>Juan Antonio Ortega Redondo</i>
Nombre del PRA:	<i>Carlos Monzo Sánchez</i>
Fecha de entrega (mm/aaaa):	06/2021
Titulación:	<i>Máster Universitario en Ingeniería de Telecomunicación</i>
Área del Trabajo Final:	<i>Electrónica</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Detección facial, visión por computador, detección de somnolencia, sistemas avanzados de asistencia a la conducción basados en visión, aprendizaje automático</i> <i>Face detection, computer vision, detection of drowsiness, vision-based advanced driver assistance systems, machine learning</i>

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

Los accidentes de tráfico son una de las principales causas de mortalidad entre la población, con 1,25 millones de fallecidos por año siendo además la principal causa de muerte en jóvenes (5 a 29 años). De ellos, el 8% de los accidentes de tráfico se deben a somnolencia y fatiga [1]. Esta situación no ha pasado desapercibida para las autoridades que llevan años implementando medidas mitigadoras y correctoras para reducir el número de siniestros de manera que una de las principales novedades para los próximos años será la obligatoriedad de sistemas automáticos de detección de fatiga del conductor a partir del 2022 [2].

Es por ello que el presente trabajo pretende implementar un demostrador de tecnología software capaz de detectar/predecir el estado de fatiga de un conductor mediante la adquisición y procesado de imágenes para determinar la tasa de parpadeo e interpretar el resultado. El sistema permitirá que el usuario conozca el resultado del análisis bajo requisitos de tiempo real *soft*.

El sistema se plantea como una aplicación de visión artificial lo cual garantiza la ausencia de problemas derivados de la interoperabilidad con sistemas HW-SW propietarios del vehículo.

Abstract (in English, 250 words or less):

Traffic accidents are one of the main causes of mortality among the population, with 1.25 million deaths per year being also the main cause of death in young people (5 to 29 years). Of these, 8% of traffic accidents are due to drowsiness and fatigue [1]. This situation has not gone unnoticed by the authorities, who have been implementing mitigating and corrective measures for years to reduce the number of accidents, so that one of the main novelties for the coming years will be the obligatory nature of automatic systems for detecting driver fatigue from the 2022 [2].

That is why the present work aims to implement a software technology demonstrator capable of detecting / predicting the fatigue state of a driver by acquiring and processing images to determine the blink rate and interpret the result. The system will allow the user to know the result of the analysis under soft real-time requirements.

The system is proposed as an artificial vision application, which guarantees the absence of problems derived from interoperability with the vehicle's proprietary HW-SW systems.

Índice

1.	Introducción	1
1.1	Contexto y justificación del Trabajo	1
1.2	Objetivos del Trabajo.....	2
1.3	Enfoque y método seguido.....	3
1.4	Planificación del Trabajo	4
1.4.1	Recursos	4
1.4.2	Planificación de tareas	4
1.5	Breve resumen de productos obtenidos	6
1.6	Breve descripción de los otros capítulos de la memoria.....	6
2.	Estado del arte	9
2.1	Introducción.....	9
2.2	Técnicas y estrategias actuales para detectar somnolencia	10
2.2.1	Técnicas basadas en el comportamiento del conductor	10
2.2.1.1	Detección de ojos y tasa de parpadeo	10
2.2.1.2	Detección de boca y bostezos.....	11
2.2.1.3	Detección de cara e identificación de expresiones faciales.....	11
2.2.1.4	Detección y estimación de la postura de la cabeza.....	13
2.2.1.5	Ventajas e inconvenientes.....	15
2.2.2	Técnicas basadas en parámetros del vehículo	15
2.2.2.1	Detección basada en análisis de la dirección.....	15
2.2.2.2	Detección de carril.....	16
2.2.2.3	Ventajas e inconvenientes.....	17
2.2.3	Técnicas basadas en parámetros fisiológicos	17
2.2.3.1	Análisis de electroencefalografía (EEG).....	17
2.2.3.2	Análisis de pulso cardíaco (ECG).....	18
2.2.3.3	Ventajas e inconvenientes.....	18
2.3	Soluciones implantadas en la industria de la automoción	18
2.3.1	Basadas en la dirección del vehículo	18
2.3.2	Basadas en el conductor	19
2.3.3	Observaciones relativas a sistemas comerciales	19
2.4	Patentes	19
2.5	Otras aplicaciones.....	20
3.	Fundamentos y diseño	22
3.1	Introducción.....	22
3.2	Descripción de etapas.....	22
3.2.1	Adquisición de imágenes.....	23
3.2.2	Detección de caras.....	24
3.2.2.1	Introducción al descriptor de característica HOG	25
3.2.3	Detección ojos y boca	26
3.2.4	Discriminador de somnolencia	27
3.2.4.1	Métricas por escena	27
3.2.4.2	Métricas temporales	31
3.3	Parámetros y umbrales	39
3.3.1	Definición.....	39
3.3.2	Consideraciones.....	40

3.4	Discriminación de somnolencia: alarma	41
3.4.1	Activación	41
3.4.2	Desactivación	41
4	Implementación	43
4.1	Diseño detallado de etapas	43
4.2	Especificación.....	43
4.2.1	Requisitos no funcionales.....	43
4.2.2	Requisitos funcionales.....	44
4.3	Librerías	45
4.4	Lenguaje de programación	45
4.5	Entorno de desarrollo	45
4.6	Control de versiones.....	46
4.7	Configuración e instalación	47
4.7.1	Librerías y dependencias	47
4.8	Desarrollo software.....	47
4.8.1	Red neuronal MLP: implementación.....	48
4.8.2	Red neuronal MLP: entrenamiento.....	50
4.8.3	Solución.....	53
4.8.3.1	Adquisición de imágenes.....	53
4.8.3.2	Detección de cara.....	53
4.8.3.3	Estimador de puntos de referencias faciales.....	53
4.8.3.4	Calcular excentricidades y ángulos	54
4.8.3.5	Almacenamiento de métricas (BD)	55
4.8.3.6	Discriminador de somnolencia	55
4.8.3.7	Alarma: Avisador acústico y visual	56
4.9	Despliegue en hardware específico.....	57
4.9.1	Plataforma hardware	57
4.9.2	Interfaz de salida	58
4.9.2.1	Diagrama de interconexión de elementos	59
4.9.3	Configuración de NVIDIA Jetson Nano:	62
4.9.4	Desarrollo software extra para control de interfaz de salida.....	63
5	Verificación y validación	65
5.1	Resultados	65
5.1.1	Configuración de pruebas	67
5.1.1.1	Parámetros de pruebas	67
5.1.1.2	Datos de prueba	67
5.1.2	Sesión 1: configuración con alta sensibilidad	68
5.1.2.1	Grupo A	68
5.1.2.2	Grupo B.....	72
5.1.3	Sesión 2: configuración con sensibilidad media	76
5.1.3.1	Grupo A	76
5.1.3.2	Grupo B.....	79
5.1.4	Comparativa entre sesiones de pruebas	82
6	Conclusiones y trabajo futuro	84
6.1	Conclusiones.....	84
6.2	Cumplimiento de objetivos	85
6.3	Trabajo futuro	86
7	Glosario	88
8	Bibliografía	89

Lista de figuras

Figura 1. Accidentes de tráfico mortales en 2018 por millón de habitantes [3] ..	1
Figura 2. Diagrama Gantt para entregables PEC1 y PEC2.....	6
Figura 3. Diagrama Gantt para entregables PEC3, PEC4 y PEC5	6
Figura 4. Categorización de técnicas de detección de somnolencia. Elaboración propia en base a [11].....	10
Figura 5. Índices de correlación entre ondas cerebrales y otros métodos de detección según S. Hachisuka [14]	12
Figura 6. Detección de cejas y ojos mediante análisis de histograma horizontal según [17].....	13
Figura 7. Postura de la cabeza según los tres ejes (fuente [15]).....	14
Figura 8. Configuración de sistema de multi-cámara de estimación de pose de cabeza según [16].....	14
Figura 9. Comparativa entre movimiento angular del volante en curva y tras la eliminación del efecto de la geometría según [18]	16
Figura 10. Posición angular del volante (en azul), velocidad del vehículo (en verde) y en rojo partes seleccionadas de la posición igual, según [19].....	16
Figura 11. Detección de carril mediante la Transformada de Hough, como se propone en [20]	16
Figura 12. Relación entre actividad cerebral y apertura/cierre de ojos [21].....	17
Figura 13. Etapas básicas de detección-predicción somnolencia	22
Figura 14. Comparativa de una imagen antes y después de ecualizar el histograma.....	23
Figura 15. Sistema de adquisición: webcam papalook 1080P	24
Figura 16. Imagen procesada mediante un descriptor de bordes (a. Canny b. Sobel).....	25
Figura 17. Proceso de extracción de características con HOG, según [29]	26
Figura 18. A) Facial landmarks de [27] con 194 puntos. B) Implementación de [27] en Dlib, con 68 puntos.....	26
Figura 19. Aplicación de versión implementada en Dlib del algoritmo [27] One Millisecond Face Alignment with an Ensemble of Regression Trees	27
Figura 20. Métrica de inclinación de cabeza	28
Figura 21: Distancia euclídea entre dos puntos P_1 y P_2	29
Figura 22. Ejemplo de puntos de referencia para estimar elipse que envuelve el ojo	29
Figura 23. Comparativa entre elipses con diferente grado de excentricidad	30
Figura 24. Puntos extraídos de la comisura de la boca mediante facial landmarks	31
Figura 25. Tasa de PERCLOS a lo largo del tiempo	32
Figura 26. Contador de ojos cerrados durante 3 frames consecutivos	32
Figura 27. Ángulo de pose en diferentes posiciones.....	33
Figura 28. Variación temporal de la excentricidad de los ojos	34
Figura 29. Comparativa de excentricidad en diferentes estados del sujeto	34
Figura 30. Comparativa de los histogramas de la boca en diferentes estados: cerrada, conversación, bostezos.....	35
Figura 31. Esquema de MLP para $N = 3$ frames	37
Figura 32. Esquema de perceptrón multicapa	38

Figura 33. Arquitectura de red MLP	38
Figura 34. Proceso de cálculo de la regla de la cadena según	39
Figura 35. Umbral de fatiga determinado en [31]	41
Figura 36. Lógica de activación de alarmas	42
Figura 37. Flujo y etapas de procesado en la solución propuesta.....	43
Figura 38. Microsoft Visual Community 2017	46
Figura 39. Vista de registro de commits de SVN.....	46
Figura 40. Curva de la función sigmoide	50
Figura 41 Muestra de valores x de excentricidad del conjunto de datos de entrenamiento. Salida “despierto”	51
Figura 42. Muestra de valores x de excentricidad del conjunto de datos de entrenamiento. Salida “somnoliento”	51
Figura 43. Lógica de activación de alarmas	56
Figura 44. Salida sujeto con somnolencia: “DROWSINESS”	57
Figura 45. Salida sujeto en alerta: “AWAKE”.....	57
Figura 46. Esquema de interconexión dispositivos hardware y NVIDIA Jetson Nano.....	60
Figura 47. Esquema de pines GPIO NVIDIA Jetson Nano.....	61
Figura 48. Vista de NVIDIA Jetson Nano con interfaz HW ejecutando la solución propuesta	61
Figura 49. Acceso por escritorio remoto VNC Viewer a Jetson Nano	62
Figura 50. Instalación de dlib en Jetson Nano.....	63
Figura 51. a) Agrupamiento en tres categoría de la escala de somnolencia de Karolinska b) Configuración de cámara recomendada por UTA-RLDD	66
Figura 52. Muestra de sjetos bajo estudio: grupo A y grupo B	66
Figura 53. Gráfica resultados en caso somnolencia. Sesión 1 – Grupo A	70
Figura 54. Gráfica estadísticas caso alerta. Sesión 1 – Grupo A	71
Figura 55. Gráfica de estadísticas caso somnolencia. Sesión 1 – grupo B.....	74
Figura 56. Gráfica Estadísticas caso alerta. Sesión 1 – grupo B.....	75
Figura 57. Estadísticas caso somnolencia. Sesión 2 – grupo A.....	77
Figura 58. Gráfica de estadísticas caso alerta. Sesión 2 – grupo A.....	78
Figura 59. Gráfica estadísticas caso somnolencia. Sesión 2 – grupo B.....	81
Figura 60. Gráfica estadísticas caso alerta. Sesión 2 – grupo B.....	82

Lista de tablas

Tabla 1. Planificación de tareas y temporización prevista	5
Tabla 2. Técnicas de detección de somnolencia según [10] Development of drowsiness detection system.....	9
Tabla 3. Niveles de somnolencia según la propuesta de S. Hachisuka [14]	12
Tabla 4. Características de las señales en el dominio temporal para detección de somnolencia en base al posicionamiento angular del volante según [18]	16
Tabla 5. Relación de puntos seleccionad para contorno de ojos y boca, según Dlib	28
Tabla 6. Escala de somnolencia de Karolinska	40
Tabla 7. Requisitos no funcionales.....	44
Tabla 8. Requisitos funcionales.....	45
Tabla 9. Lista de librerías de Python	47
Tabla 10. Diagrama UML de la Clase que implementa la red neuronal MLP ...	49
Tabla 11. Tasa de acierto de red neuronal.....	52
Tabla 12. Lista de elementos hardware para interfaz de salida	59
Tabla 13. Librerías requeridas y no incluidas en la imagen de instalación de NVIDIA Jetson Nano	63
Tabla 14. Lista de ficheros para pruebas - grupo A.....	67
Tabla 15. Lista de ficheros para pruebas - grupo B.....	68
Tabla 16. Resultados en bruto. Sesión 1 – Grupo A	69
Tabla 17. Resultados estadísticos. Sesión 1 – Grupo A	69
Tabla 18. Resultados estadísticos caso somnolencia. Sesión 1 – Grupo A.....	70
Tabla 19. Estadísticas en caso alerta. Sesión 1 – Grupo A	70
Tabla 20. Datos en bruto. Sesión 1 – grupo B.....	72
Tabla 21. Estadísticas. Sesión 1 – grupo B.....	73
Tabla 22. Estadísticas caso somnolencia. Sesión 1 – grupo B	73
Tabla 23. Estadísticas caso alerta. Sesión 1 – grupo B	74
Tabla 24. Datos en bruto. Sesión 2 – grupo A.....	76
Tabla 25. Estadísticas totales. Sesión 2 – grupo A	77
Tabla 26. Estadísticas caso somnolencia. Sesión 2 – grupo A	77
Tabla 27. Estadísticas caso alerta. Sesión 2 – grupo A	78
Tabla 28. Datos en bruto Sesión 2 – grupo B.....	79
Tabla 29. Estadísticas. Sesión 2 – grupo B.....	80
Tabla 30. Estadísticas caso somnolencia. Sesión 2 – grupo B	80
Tabla 31. Estadísticas caso alerta. Sesión 2 – grupo B	81

1. Introducción

1.1 Contexto y justificación del Trabajo

Según datos de la Dirección General de Tráfico, los accidentes de carretera suponen una de las principales causas de mortalidad entre la población, con 1,25 millones de fallecidos por año siendo además la principal causa de muerte en jóvenes (5 a 29 años). Además se estima que hasta el 8% de los accidentes de tráfico está originados por la somnolencia y fatiga del conductor [1]. Así, en la Figura 1 se muestran los datos recogidos por la Agencia Europea de Estadística, Eurostat, para el año 2018 y relativos al número de accidentes mortales por millón de habitantes, segregados por regiones.

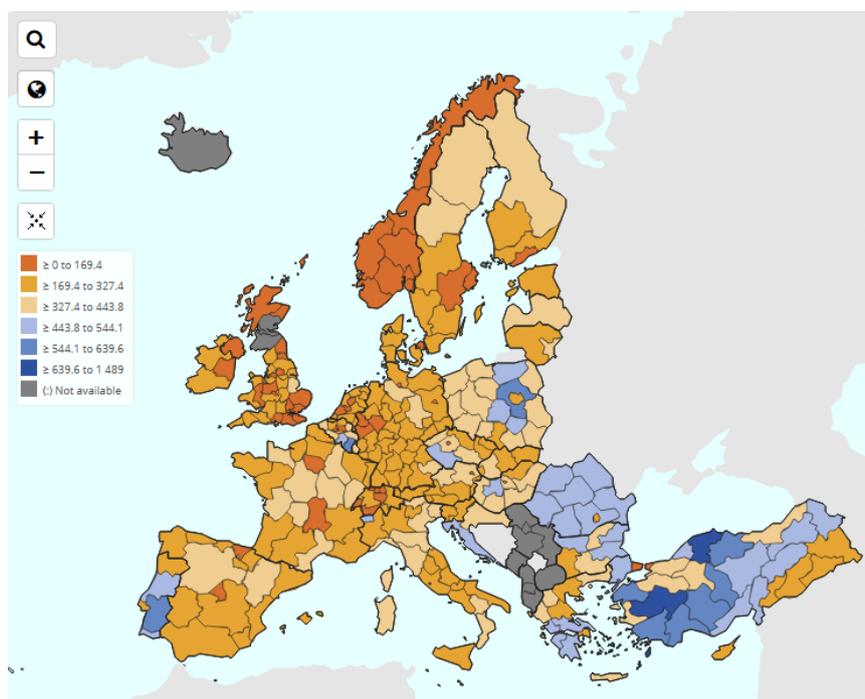


Figura 1. Accidentes de tráfico mortales en 2018 por millón de habitantes [3]

Esta situación no ha pasado desapercibida para las autoridades, que llevan años implementando medidas cada vez más exigentes para mitigar y reducir el número de siniestros, tanto en materia de seguridad activa como pasiva, de manera que una de las principales novedades para los próximos años será la obligatoriedad, a partir del 2022, de la incorporación de sistemas automáticos de detección de fatiga del conductor, tal como se establece en la ya aprobada regulación europea 2018/0145 (COD) PE-CONS 82/19 [2].

Es por ello que la industria de la automoción y los propios fabricantes están desarrollando nuevos sistemas y tecnologías que serán integrados en los modelos a comercializar en los próximos años. En este sentido hay diferentes estrategias posibles para abordar el desarrollo de sistemas que permitan la detección y predicción de síntomas de fatiga, somnolencia o desvanecimientos del conductor, tales como:

- Supervisión del patrón de dirección: usando como datos de entrada la posición de la dirección del vehículo.
- Evaluación de la posición del vehículo con seguimiento de carril: mediante una cámara se monitoriza el posicionamiento del vehículo dentro del carril
- Monitorización ocular y/o facial del conductor para interpretar gestos o parámetros biométricos.
- Análisis de parámetros relevantes de conducción tales como la velocidad, aceleración, presión/posición en los pedales, que ya son proporcionados por el propio vehículo.

Sirva para ilustrar los esfuerzos de la industria para satisfacer esta nueva demanda con el ejemplo concreto del fabricante francés DS, que ha desarrollado el sistema *DS Driver Attention Monitoring* mediante la combinación de dos cámaras infrarrojas que enfocan al conductor con el seguimiento continuo de la posición del vehículo [6].

En conclusión, y dentro del marco regulatorio de obligado cumplimiento por la industria en los próximos años, el presente TFM pretende obtener datos biométricos del conductor mediante el uso de tecnologías, pertenecientes al campo de la visión artificial, que permitan determinar si un conductor tiene síntomas de fatiga o somnolencia notificando esta circunstancia a través de una interfaz de salida.

1.2 Objetivos del Trabajo

Los objetivos del trabajo se enumeran a continuación:

- Análisis y estudio de factores fisiológicos y biométricos que permitan determinar situaciones de somnolencia y fatiga.
- Estudio y uso de librerías de visión por computador ampliamente extendidas en la industria (OpenCV).
- Estudio y uso de librerías que implementan redes neuronales ya entrenadas para la clasificación de caras y ojos (TensorFlow, PyTorch, Caffe y MXNet).

- Desarrollo de una aplicación software sobre Linux que:
 - Capture imágenes en tiempo real de forma cíclica.
 - Realice la detección y clasificación de ojos del usuario y determine si los párpados están abiertos o cerrados.
 - Determine, en base al análisis de secuencias de vídeo, el tiempo que los párpados permanecen cerrados y abiertos, de manera que se pueda calcular la duración del parpadeo.
 - Guarde un histórico con los resultados del punto anterior (párpados cerrados o abiertos) para determinar el patrón predominante en escena: usuario despierto o con signos de somnolencia o fatiga.
 - Controle una cámara para la adquisición de imágenes.
- Diseñar la arquitectura software que permita encapsular las etapas y funcionalidades descritas anteriormente.
- Diseñar interfaz de salida que permita visualizar la imagen capturada con los resultados del procesado y aquellos valores estadísticos notables sobrepuestos en la imagen con avisador y/o notificación de posible situación de somnolencia/fatiga.
- Se plantea el análisis de diferentes sensores hardware que se pudieran integrar con el sistema software para la adquisición de información adicional relevante y en adición a la captura de vídeo procedente de la cámara. El objetivo es complementar la información biométrica a procesar por el sistema con el fin de otorgarle una mayor fiabilidad. Algunos posibles sensores bajo estudio serían sensores ópticos de ritmo cardíaco o sensores de temperatura, ya que hay estudios que relacionan estos parámetros con situaciones de somnolencia, tal como se expone en las publicaciones médicas [8] y [9]
- Explorar posibles aplicaciones adicionales en otros ámbitos que requieren alta precisión o atención del operador, aplicaciones biomédicas o marketing digital.

1.3 Enfoque y método seguido

El enfoque a seguir en el presente trabajo consiste en el desarrollo de un producto nuevo en base a productos y tecnologías ya existentes, como es el caso de algunas de las librerías de Inteligencia Artificial mencionadas en el epígrafe anterior, de manera que estaríamos ante un proyecto que

integra estas tecnologías para construir un nuevo producto con capacidades específicas para resolver un problema concreto de la industria.

Se opta por esta estrategia porque la pretensión es construir una aplicación que se enfrentará a importantes desafíos técnicos (requisitos de tiempo real y computación bajo paradigmas de visión artificial e inteligencia artificial) por lo que se hará uso de librerías que resuelven ciertos problemas clásicos del contexto, como puede ser la configuración y entrenamiento de redes neuronales para tareas concretas como la detección de ojos: algunas de estas librerías han demostrado una acreditada solvencia dentro de la industria, de manera que el proceso de desarrollo podrá enfocarse más en la implementación de funcionalidades a más alto nivel para la consecución del objetivo propuesto.

1.4 Planificación del Trabajo

Para la planificación del trabajo se consideran, en primer lugar, los recursos mínimos en cuanto a software, hardware y herramientas que serán necesarios. En segundo lugar se ha establecido la planificación de tareas e hitos con su respectiva temporización.

1.4.1 Recursos

Para realizar este proyecto se hará uso de las siguientes herramientas y tecnologías:

- Equipo con S.O. Linux Ubuntu 18.04
- IDE Visual Studio Community / Visual Studio Code
- Python 3.x
- Control de versiones: Subversion (SVN) en repositorio local
- Suite Ofimática Microsoft Office 2016
- Cámara

1.4.2 Planificación de tareas

En la Tabla 1 se detalla la planificación inicial de tareas y subtareas necesarias para la consecución de los objetivos trazados. Considerando una carga teórica de 12 créditos ECTS (300h) se realiza una asignación de tiempos considerando la complejidad y relevancia de cada uno de los hitos previstos.

Así mismo, se establecen cinco hitos que conllevan sus respectivos entregables oficiales de manera que la planificación se divide en un primer nivel de tareas:

1. Definición
2. Estado del arte
3. Análisis, diseño e implementación
4. Memoria
5. Presentación

Además, para cada uno de ellos se realiza un desglose de tareas en hasta dos subniveles más, considerando en el primer subnivel los hitos internos (no requieren entregable parcial) más relevantes y en un segundo subnivel las subtareas necesarias para la consecución de éstos.

Milestone	Task Name	Duration	Start	Finish	Regular Work
No	TFM	162 days	Wed 17/02/21	Thu 10/06/21	313,63 hrs
No	Definición	14 days	Wed 17/02/21	Sun 28/02/21	49,8 hrs
No	Brainstorming	4,5 days	Wed 17/02/21	Fri 19/02/21	9 hrs
No	Contexto y justificación del trabajo	0 days	Fri 19/02/21	Sat 20/02/21	6 hrs
No	Investigar marco legal y siniestralidad	3 days	Fri 19/02/21	Sat 20/02/21	6 hrs
No	Planificación del trabajo	10 days	Mon 22/02/21	Sun 28/02/21	34,8 hrs
No	Objetivos del Trabajo	3 days	Mon 22/02/21	Mon 22/02/21	6 hrs
No	Enfoque y método seguido	3 days	Mon 22/02/21	Tue 23/02/21	4,8 hrs
No	Breve resumen de productos obtenidos	3 days	Wed 24/02/21	Thu 25/02/21	3 hrs
No	Elección de herramientas de planificación	0,5 days	Thu 25/02/21	Thu 25/02/21	2 hrs
No	Elaboración de planificación y diagramas de Gantt	3 days	Fri 26/02/21	Sat 27/02/21	12 hrs
No	Resumen	5 days	Fri 26/02/21	Sun 28/02/21	7 hrs
Yes	Entrega: PEC 1 - Definición	0 days	Sun 28/02/21	Sun 28/02/21	0 hrs
No	Estado del arte	18 days	Mon 01/03/21	Sun 14/03/21	42 hrs
No	Recopilar información	17 days	Mon 01/03/21	Fri 12/03/21	20 hrs
No	Patentes y sistemas en la industria	11 days	Mon 01/03/21	Sun 07/03/21	8,8 hrs
No	Patentes y artículos en el ámbito académico	7 days	Mon 08/03/21	Fri 12/03/21	11,2 hrs
No	Revisión de consideraciones previas	5 days	Wed 10/03/21	Fri 12/03/21	10 hrs
No	Preparar documentación	3 days	Sat 13/03/21	Sun 14/03/21	12 hrs
Yes	Entrega: PEC 2 - Estado del arte	0 days	Mon 15/03/21	Mon 15/03/21	0 hrs
No	Análisis, diseño e implementación	78 days	Tue 16/03/21	Fri 07/05/21	169 hrs
No	Especificación de requisitos	4 days	Tue 16/03/21	Wed 17/03/21	8 hrs
No	Definición de requisitos de sistema	1 day	Tue 16/03/21	Tue 16/03/21	4 hrs
No	Definición de requisitos SW (bajo nivel)	2 days	Tue 16/03/21	Wed 17/03/21	4 hrs
No	Diseño	8 days	Tue 16/03/21	Sat 20/03/21	26 hrs
No	Elección de HW a integrar	5 days	Tue 16/03/21	Wed 17/03/21	8 hrs
No	Elección de librerías a integrar	5 days	Thu 18/03/21	Fri 19/03/21	11 hrs
No	Definición de arquitectura HW-SW	5 days	Fri 19/03/21	Sat 20/03/21	7 hrs
No	Implementación	64 days	Tue 23/03/21	Wed 05/05/21	123,8 hrs
No	SWC_1: adquisición de imágenes	11 days	Tue 23/03/21	Sun 28/03/21	15,4 hrs
No	SWC_2: procesado	48 days	Tue 30/03/21	Sun 02/05/21	61,6 hrs
No	SWC_2.1: Detector	15 days	Tue 30/03/21	Wed 07/04/21	21 hrs
No	SWC_2.2: Clasificador	23 days	Thu 08/04/21	Fri 23/04/21	27,6 hrs
No	SWC_2.3: Cálculo de estadísticos notables	13 days	Sat 24/04/21	Sun 02/05/21	13 hrs
No	SWC_3: salida de resultados	13 days	Mon 26/04/21	Wed 05/05/21	15,6 hrs
No	Pruebas funcionales y validación	15 days	Mon 26/04/21	Wed 05/05/21	18 hrs
No	Resolución de bugs y refactorización	11 days	Wed 28/04/21	Wed 05/05/21	13,2 hrs
No	Documentación	7 days	Wed 05/05/21	Fri 07/05/21	11,2 hrs
Yes	Entrega: PEC 3 - Diseño e implementación	0 days	Fri 07/05/21	Fri 07/05/21	0 hrs
No	Memoria	32 days	Mon 10/05/21	Mon 31/05/21	37,23 hrs
No	Documentación	24 days	Mon 10/05/21	Tue 25/05/21	33,03 hrs
No	Revisión y actualización de documentación	25 days	Mon 10/05/21	Tue 25/05/21	16 hrs
No	Resultados y conclusiones	7 days	Thu 13/05/21	Sun 16/05/21	8,4 hrs
No	Manual de operación	7 days	Tue 18/05/21	Thu 20/05/21	5,03 hrs
No	Guía de usuario	3 days	Fri 21/05/21	Tue 25/05/21	3,6 hrs
No	Despliegue y puesta en servicio	7 days	Wed 26/05/21	Mon 31/05/21	4,2 hrs
Yes	Entrega: PEC 4 - Memoria	0 days	Mon 31/05/21	Mon 31/05/21	0 hrs
No	Presentación	12 days	Wed 02/06/21	Wed 09/06/21	15,6 hrs
No	Preparación de la presentación	13 days	Wed 02/06/21	Wed 09/06/21	15,6 hrs
Yes	Entrega: PEC 5 - Presentación	0 days	Thu 10/06/21	Thu 10/06/21	0 hrs

Tabla 1. Planificación de tareas y temporización prevista

Como resultado de esta planificación se obtiene el siguiente diagrama de Gantt (Figura 2 y Figura 3), con la dependencia entre tareas e hitos que se detalla en la Tabla 1.

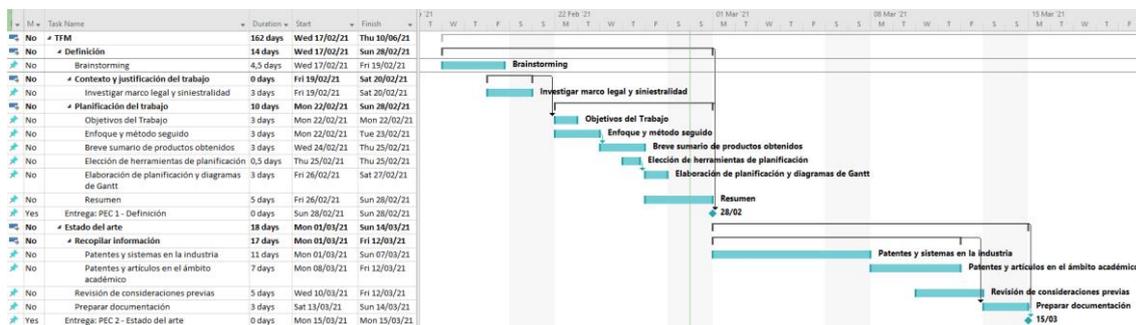


Figura 2. Diagrama Gantt para entregables PEC1 y PEC2

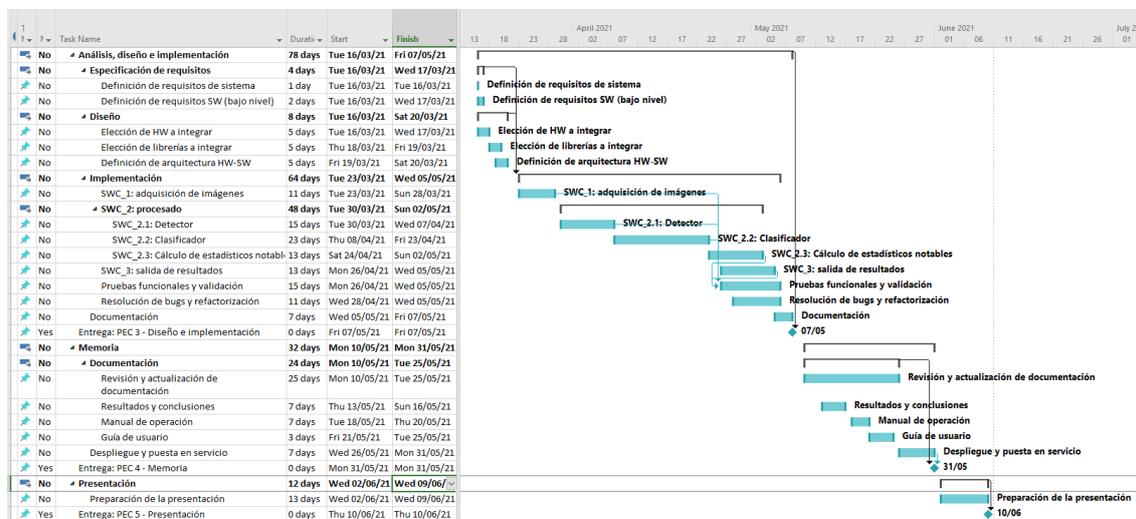


Figura 3. Diagrama Gantt para entregables PEC3, PEC4 y PEC5

1.5 Breve resumen de productos obtenidos

A más alto nivel se obtendrá como producto un sistema con capacidades para la adquisición de imágenes y que serán procesados por una aplicación software que correrá sobre Linux en tiempo real.

Otro tipo de entregables derivados de esta aplicación serán:

- Documentación: memoria y presentación
- Software: Código fuente y binarios

1.6 Breve descripción de los otros capítulos de la memoria

La memoria contará con los siguientes capítulos:

- **Estado del arte:**
Análisis y estudio de las soluciones similares existentes, tanto en el ámbito industrial como académico, caso de haberlas.
- **Análisis previo y fundamentos técnicos**
Se realiza un estudio inicial de las tecnologías, fundamentos matemáticos y computacionales relevantes para el desarrollo del proyecto en las disciplinas técnicas bajo las que se realiza el proyecto: visión artificial e inteligencia artificial.
- **Especificación de requisitos**
Desglose de requisitos funcionales y no funcionales desde un nivel de sistema con trazabilidad a más bajo nivel de detalle: atómicos y autocontenidos.
- **Diseño**
Se aborda el diseño funcional de la aplicación a más alto nivel y haciendo énfasis en el modelado matemático sobre el que se sustentará
- **Implementación**
Versa sobre el desarrollo de la solución y los diferentes componentes software implementados.
- **Resultados: verificación y validación**
Mediante una metodología analítica y en base a la especificación de requisitos definida se ha de determinar la matriz de trazabilidad entre requisitos, así como el método de prueba propuesto y el respectivo grado de consecución para cada uno de ellos, en términos absolutos (si/no) y/o mediante KPIs estadísticos relevantes.
- **Conclusiones y trabajo futuro:**
Se realiza un análisis de los resultados obtenidos, valorando las limitaciones, fiabilidad y posibles errores existentes en la solución. En base a ellos se desarrolla una propuesta de funcionalidades y mejoras a conseguir en posibles futuras iteraciones en el desarrollo de la solución.
- **Glosario:**
Lista de términos y acrónimos más relevantes
- **Bibliografía**
Lista de referencias bibliográficas
- **Anexos:**
Documentación extra del proyecto que por su extensión no pueda ser incluida en el resto de capítulos como:

- Anexo A: Documentación software:
- Anexo B: Pruebas funcionales
- Anexo C: Manuales

2. Estado del arte

2.1 Introducción

El estudio y desarrollo de sistemas de detección y prevención de síntomas de somnolencia es, desde hace varias décadas, un campo de gran interés en entornos académicos como para diferentes industrias, que promueven su estudio e investigación así como el desarrollo de nuevos productos a través de diferentes centros de I+D. Este es el caso de la industria de la automoción, ya que este tipo de sistemas tienen una más que evidente utilidad como nuevo elemento de seguridad activa dentro del vehículo, contribuyendo por ende a reducir el número de accidentes de tráfico.

En este sentido, uno de los primeros estudios de referencia que abordan la temática del presente trabajo es el paper “Development of drowsiness detection system” [10] presentado por investigadores del centro de I+D de Nissan Motor en el año 1994 con motivo de la conferencia *Vehicle Navigation and Information Systems Conference*.

Este documento supone un importante punto de partida dentro del estudio del estado de arte del presente trabajo por dos motivos:

1. La fecha de publicación, 1994, lo convierte en uno de los primeros estudios que abordan posibles soluciones a la problemática objeto de estudio.
2. Realiza una categorización y comparativa de diferentes técnicas para abordar el problema según criterios de precisión, practicidad y extensibilidad, tal como se ilustra en la siguiente tabla:

Detection Techniques		Description	Detection Accuracy	Practicality	Extensibility
Sensing of Human Physiological Phenomena	Physiological Signals	Detection by Changes in Brain Waves, Blinking, Heart Rate, Pulse Rate, Skin Electric Potential, etc.	◎	×	△
	Physical Reactions	Detection by Changes in Inclination Driver's Head, Sagging posture, Frequency at Which Eyes Close, Gripping force on Steering Wheel, etc.	◎	○	◎
Sensing of Driving Operation		Detection by Changes in Driving Operations (Steering, Accelerator, Braking, Shift Lever, etc.)	○	◎	×
Sensing of Vehicle Behavior		Detection by Changes in Vehicle Behavior (Speed, Lateral G, Yaw Rate, Lateral Position, etc.)	○	◎	×
Response of Driver		Detection by Periodic Request for Response	△	×	◎
Traveling Conditions		Detection by Measurement of Traveling Time and Conditions (Daytime or Nighttime, Speed, etc.)	×	○	◎

◎ : Very Good ○ : Good △ : Average × : Poor

Tabla 2. Técnicas de detección de somnolencia según [10] Development of drowsiness detection system

2.2 Técnicas y estrategias actuales para detectar somnolencia

Las diferentes estrategias planteadas en la propuesta de los investigadores de Nissan Motor se pueden agrupar en tres categorías, que son las que siguen en la actualidad la mayoría de los diferentes grupos de investigación y desarrollo en el área, tal como se plantea en un paper bastante más reciente realizado por investigadores de la Universidad COMSATS de Islamabad [11], es decir:

- A. Técnicas basadas en parámetros de comportamiento del conductor.
- B. Técnicas basadas en parámetros del vehículo.
- C. Técnicas basadas en parámetros fisiológicos.

Para cada una estas categorías se pueden seguir diferentes estrategias en función de los parámetros de entrada a analizar, tal como es expone en la Figura 4.

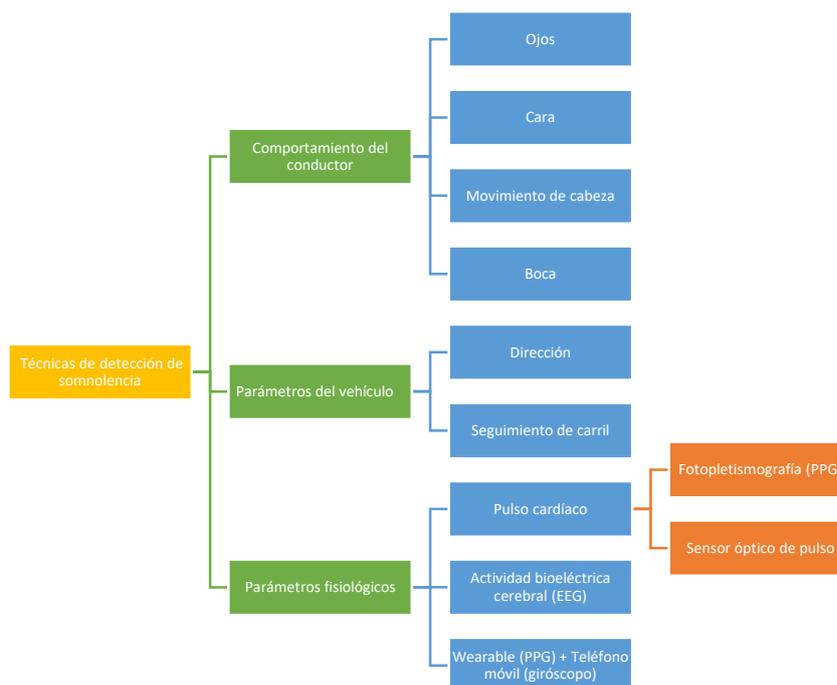


Figura 4. Categorización de técnicas de detección de somnolencia. Elaboración propia en base a [11]

2.2.1 Técnicas basadas en el comportamiento del conductor

Las técnicas de detección de somnolencia basadas en el comportamiento del conductor se catalogan como no invasivas y emplean tecnologías del ámbito de la visión artificial para observar y detectar parámetros que indiquen el estado del sujeto, tales como la tasa de cierre de ojos o de bostezos.

2.2.1.1 Detección de ojos y tasa de parpadeo

El objetivo que persigue esta estrategia es la detección de los ojos para determinar la tasa de ciclos en los que estos están cerrados. Existen diferentes

métricas para determinar el nivel de somnolencia o fatiga, si bien el más extendido actualmente en los sistemas de monitorización es la tasa PERCLOS, que contiene una buena cantidad de información sobre el estado de fatiga de un sujeto e indica el porcentaje de tiempo que el párpado ocluye la pupila (> 80%) dentro de una ventana móvil de 60 segundos, tal como se concluye en el estudio de U. Trutschel [12] y que se define como:

$$PERCLOS = \frac{N^{\circ} \text{ de frames con párpados cerrados}}{N^{\circ} \text{ frames capturados en 60s}} \quad (1)$$

2.2.1.2 Detección de boca y bostezos

Los sistemas de detección basados en bostezos realizan una detección de la boca para posteriormente analizar la amplitud de apertura de la misma de manera que si el sistema encuentra un número elevado de fotogramas consecutivos donde la boca está muy abierta, el sistema genera la alerta. Estos sistemas consideran que las aperturas pequeñas se producen cuando el conductor está hablando. Por tanto, hace uso de la relación geométrica entre el ancho y alto de la boca, de manera que se puede considerar apertura por bostezo aquella en la que el cociente de ambas medidas es mayor a 0.5, tal como propone L. Lingling en [13].

2.2.1.3 Detección de cara e identificación de expresiones faciales

Existen otras propuestas que persiguen determinar el comportamiento y estado del sujeto en base a aquellas expresiones faciales que pueden caracterizar a un sujeto bajo síntomas de fatiga. Éstas técnicas se basan en el análisis de otras características como el posicionamiento y factor de forma de las cejas, las pestañas, los labios y los ojos.

En base a la combinación de estos parámetros la propuesta realizada por S. Hachisuka [14] de los Research Laboratories de DENSO Corporation desarrolla un método para la detección de la somnolencia del conductor utilizando toda la expresión facial, concluyendo que los resultados de este análisis revelaron que aparecen rasgos indicadores de somnolencia en las cejas, mejillas y boca, además de en los ojos.

En dicho estudio se realiza una comparativa entre diferentes estrategias de detección de somnolencia con respecto al método que analiza las ondas alfa cerebrales¹ (puesto que este método es considerado el más efectivo para detectar estados de somnolencia de forma efectiva). Así, se concluye que el método propuesto en el paper es el que arroja un valor de correlación más alto respecto a las citadas ondas alfa, tal como se observa en figuras de correlación expuestas en la Figura 5.

¹ Las ondas alfa son oscilaciones electromagnéticas en el rango de frecuencias de 8-13 Hz que surgen de la actividad eléctrica sincrónica y coherente.

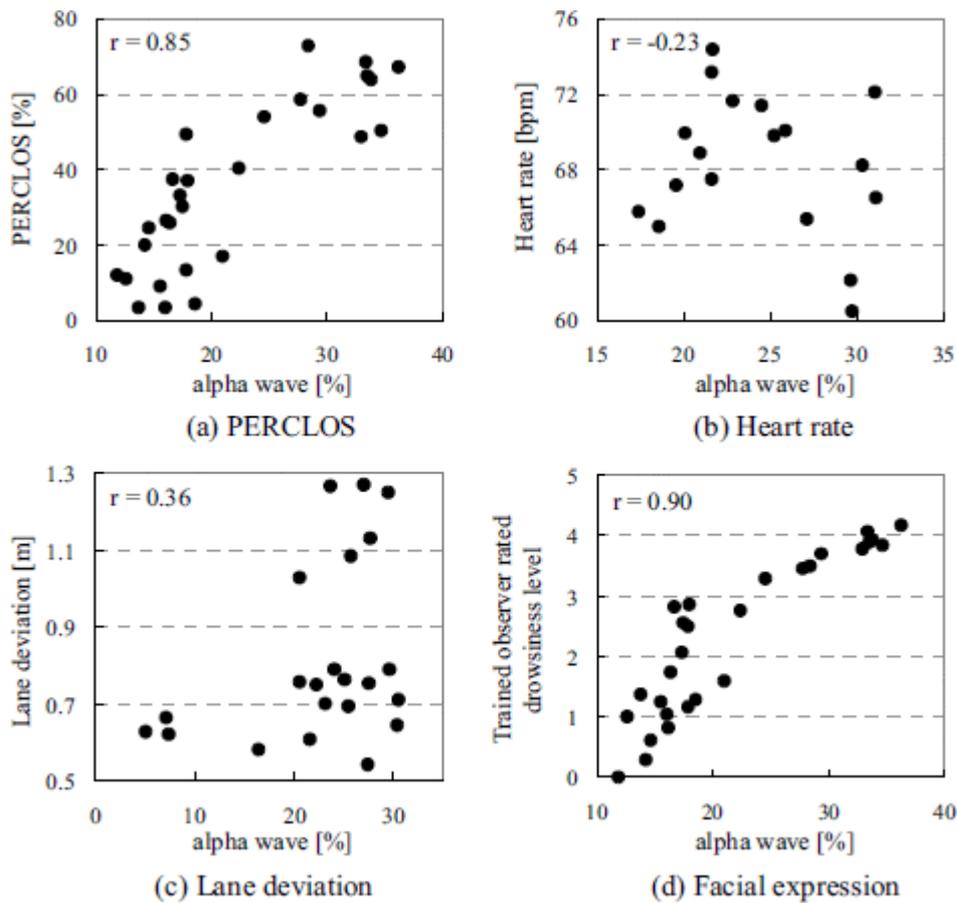


Figura 5. Índices de correlación entre ondas cerebrales y otros métodos de detección según S. Hachisuka [14]

Con ello llegan a identificar hasta seis niveles de somnolencia diferentes tal como se observa en la Tabla 3:

Categoría	Nivel de somnolencia
0	sin sueño
1	Ligeramente somnoliento
2	somnoliento
3	bastante somnoliento
4	muy somnoliento
5	durmiendo

Tabla 3. Niveles de somnolencia según la propuesta de S. Hachisuka [14]

En el trabajo de M. A. Assari y M. Rahmati [17] también se propone monitorizar y analizar cejas, ojos y boca sin embargo, y en aras de incrementar la robustez del sistema cuando las condiciones de luz son pobres, se añaden diodos emisores de infrarrojos (IR) al sistema de adquisición de imágenes, de manera que en las capturas se procesa el posicionamiento de los ojos y cejas en escena

en base al histograma vertical de los niveles de brillo. De esta manera las cejas y los ojos abiertos pueden ser identificados a través de los valles que provocan en el histograma, dado su menor nivel de intensidad, tal como se muestra en la Figura 6.

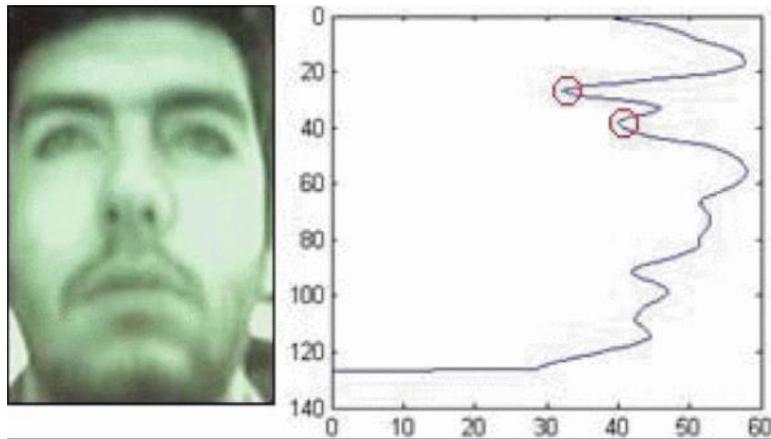


Figura 6. Detección de cejas y ojos mediante análisis de histograma horizontal según [17]

2.2.1.4 Detección y estimación de la postura de la cabeza

Un problema que presentan los sistemas que pretenden monitorizar el estado de diferentes características faciales, como los ojos o la boca, reside en las oclusiones que con frecuencia se producen debido a la perspectiva de una sólo cámara que habitualmente se posiciona frontalmente al sujeto. No obstante, cuando el conductor realiza movimientos de cabeza fuera de la pose frontal se producen eventos de especial interés que no pueden ser capturados por los sistemas anteriormente mencionados.

Precisamente esta limitación de los sistemas anteriores es la que pretenden solucionar los sistemas de estimación de posición de la cabeza, y por ende de la mirada, que habitualmente optan por una configuración de varias cámaras distribuidas de manera que se pueda determinar la posición a lo largo de tres ejes:

- a. Eje lateral (cabeceo)
- b. Eje longitudinal (alabeo o balanceo)
- c. Eje vertical (guiñada)

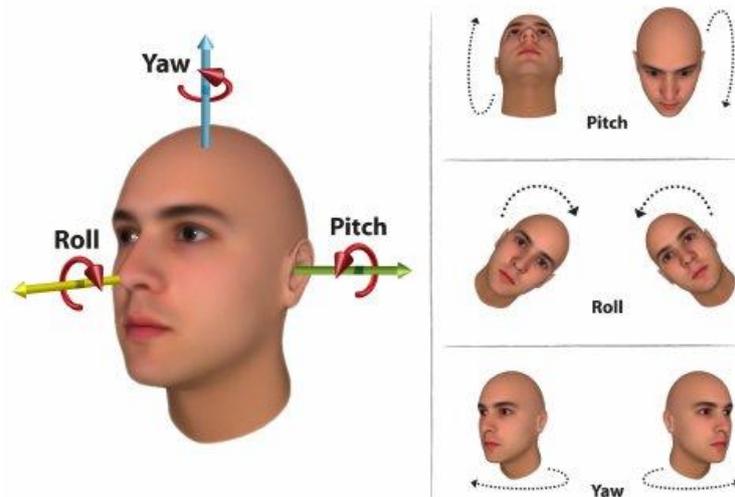


Figura 7. Postura de la cabeza según los tres ejes (fuente [15])

Una propuesta especialmente relevante siguiendo esta estrategia es la presentada por A. Tawari y M. M. Trivedi en [15], que combina dos cámaras frontales para la estimación de la posición de la cabeza y la dirección de la mirada mientras que una tercera cámara trasera se usa para la monitorización del entorno del conductor. Con ello se pueden identificar posibles focos de atención, tal como se ilustra en la Figura 8.

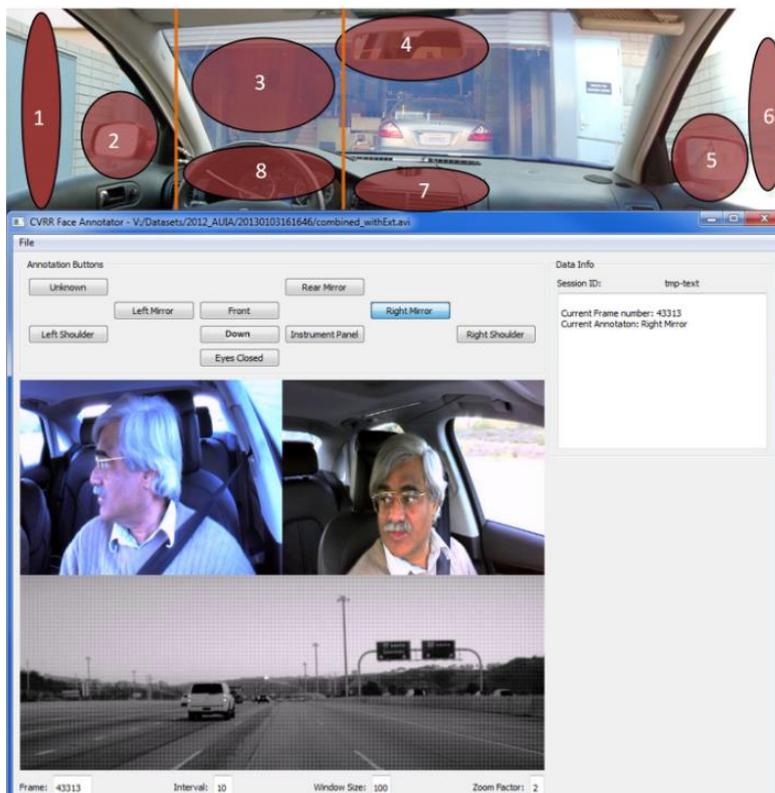


Figura 8. Configuración de sistema de multi-cámara de estimación de pose de cabeza según [16]

2.2.1.5 Ventajas e inconvenientes

Entre las principales ventajas de este tipo de técnicas está la universalidad, ya que las detecciones serían igualmente válidas en otro tipos de contextos industriales o comerciales. Además, su desarrollo resulta más sencillo puesto que su experimentación y validación depende de los sujetos y no del vehículo.

En cuanto a sus principales desventajas está el hecho de que pueden suponer una violación de la privacidad del sujeto bajo análisis, ya que en la mayoría de los casos se emplean cámaras de visión. Además, y especialmente cuando las cámaras trabajan en el espectro visible, las condiciones de luminosidad o el uso de accesorios como gafas impactan negativamente en los resultados.

Por último y no menos importante, en el momento actual este tipo de técnicas no consideran las implicaciones que podrían tener en los resultados el estado de salud del sujeto.

2.2.2 Técnicas basadas en parámetros del vehículo

Dentro de este grupo de técnicas se engloban todas aquellas que hacen uso de parámetros propios de conducción siendo el análisis de la dirección el más extendido.

2.2.2.1 Detección basada en análisis de la dirección

Este método recopila y analiza datos de posicionamiento angular del volante de dirección en combinación con datos de velocidad de manera que persigue identificar patrones de somnolencia que influyen en una conducción errática.

En esta línea se enmarcan la propuesta desarrolladas en [18] y [19], que analizan el ángulo del volante, el cual a su vez depende de la geometría y la curvatura de la carretera. Los cambios y patrones que se pretenden identificar se basan en las siguientes dos hipótesis:

1. En las secciones rectas de la carretera, las señales del volante solo consisten en ajustes de la dirección para mantener el carril por parte de los conductores.
2. Por el contrario, en las secciones curvas las señales de dirección reflejan los efectos de geometría de la carretera, que debe ser eliminado para poder detectar también los posibles efectos de la somnolencia, tal como se muestra en la Figura 9.

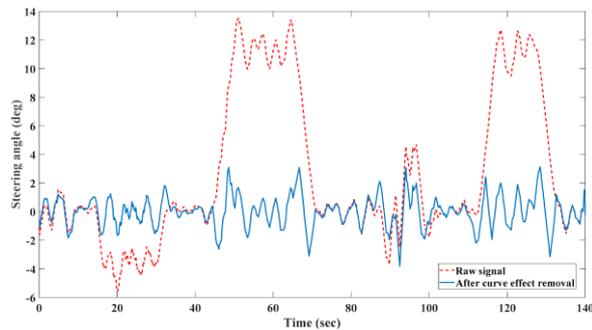


Figura 9. Comparativa entre movimiento angular del volante en curva y tras la eliminación del efecto de la geometría según [18]

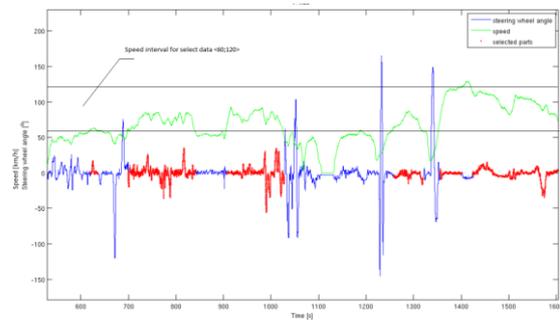


Figura 10. Posición angular del volante (en azul), velocidad del vehículo (en verde) y en rojo partes seleccionadas de la posición angular, según [19]

Así, correlando los datos de posición angular y del volante se obtienen diferentes métricas tales como rango, desviación estándar o coeficiente de oblicuidad para determinar situaciones de fatiga: en el trabajo [18] se propone la siguiente lista de calculados, tal como se expone en la Tabla 4.

Index	Time Domain Features	Descriptions
I_1^a, I_1^v	Range	Difference between minimum and maximum of signal
I_2^a, I_2^v	Standard Deviation	Dispersion of the data around mean value
I_3^a, I_3^v	Energy	Sum of the square of signal magnitude
I_4^a, I_4^v	Zero Crossing Rate (ZCR)	Number of steering or steering velocity direction changes per second
I_5^a, I_5^v	First Quartile	Middle number between the smallest number and the median of the signal in sliding window
I_6^a, I_6^v	Second Quartile	Median of the signal in the sliding window
I_7^a, I_7^v	Third Quartile	Middle value between the median and the highest value of the signal in sliding window
I_8^a, I_8^v	Katz Fractal Dimension (KFD)	An index for characterizing fractal patterns or sets by quantifying their complexity as a ratio of the change in detail to the change in scale.
I_9^a, I_9^v	Skewness	A measure for signal similarity
I_{10}^a, I_{10}^v	Kurtosis	Measure of tailedness of the probability distribution of a random variable
I_{11}^a, I_{11}^v	Sample Entropy (SamEn)	Complexity of signal in time domain based on distance in embedding dimension
I_{12}^a, I_{12}^v	Shannon Entropy (ShEn)	Complexity of signal in time domain based on probability function

Tabla 4. Características de las señales en el dominio temporal para detección de somnolencia en base al posicionamiento angular del volante según [18]

2.2.2.2 Detección de carril

Existen otras técnicas menos extendidas que proponen alternativas como la detección y el análisis de la posición del vehículo respecto al carril de circulación. Es el caso de la propuesta de Y. Katyal, S Alur y S. Dwivedi en [20] de manera que si el vehículo sale de la pista delimitada por las líneas el sistema envía una alerta.

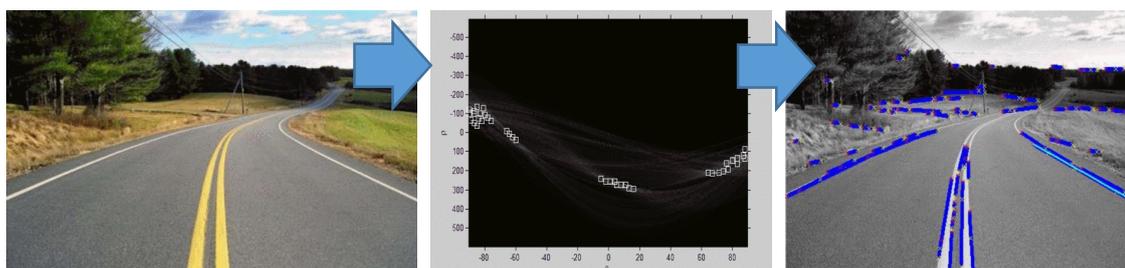


Figura 11. Detección de carril mediante la Transformada de Hough, como se propone en [20]

2.2.2.3 Ventajas e inconvenientes

Nótese que este tipo de técnicas habitualmente obtienen los datos objeto de análisis desde el bus CAN del propio vehículo a través del puerto OBD².

Entre las ventajas de este tipo de estrategias está el hecho de que no sean intrusivas y tengan poca dependencia del tipo de conductor, lo que les otorga alta robustez. Sin embargo, y entre sus principales desventajas, adolecen de una alta dependencia de los fabricantes por lo que su aplicabilidad real dependería de éstos. Así mismo, las condiciones de contorno suponen una problemática adicional para comparar los datos de las series temporales, ya que tienen una alta dependencia de las condiciones del tráfico o de las condiciones de la carretera.

2.2.3 Técnicas basadas en parámetros fisiológicos

El tercer grupo de estrategias se basa en parámetros fisiológicos como el análisis de la actividad cerebral, el registro y estudio de las variaciones del pulso cardíaco.

2.2.3.1 Análisis de electroencefalografía (EEG)

El análisis de la actividad cerebral mediante la adquisición de señales de un electroencefalograma (EEG) constituye un excelente método para determinar si un sujeto se encuentra bajo síntomas de somnolencia, tal como se expone en el artículo [21].

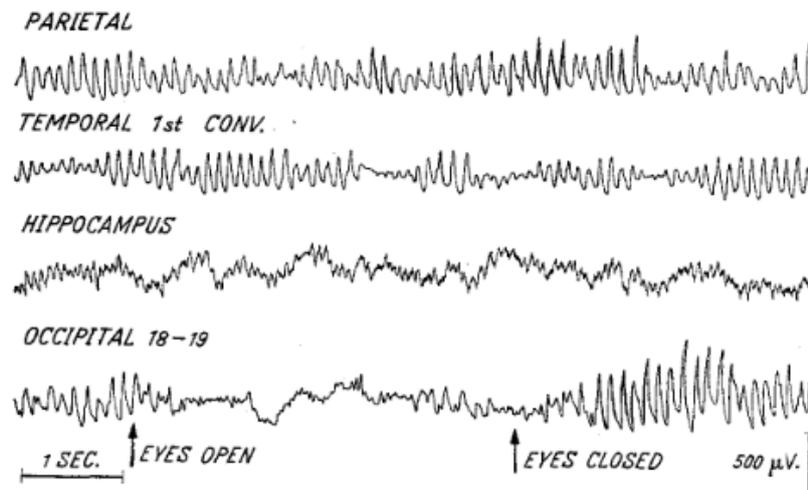


Figura 12. Relación entre actividad cerebral y apertura/cierre de ojos [21]

² OBD (On Board Diagnostics) es un sistema de diagnóstico de a bordo en vehículos comerciales e industriales (coches y camiones). Actualmente se emplean los estándares OBD-2 (Estados Unidos), EOBD (Europa) y JOBD (Japón) y reporta numerosos parámetros de conducción y monitorización de sistemas en tiempo real.

2.2.3.2 Análisis de pulso cardíaco (ECG)

Otra técnica bastante extendida dada su precisión consiste en analizar el ritmo cardíaco del sujeto bajo estudio, ya que se ha demostrado que el sistema nervioso presenta alteraciones ante episodios de estrés, fatiga y somnolencia. Por tanto, y mediante el análisis de la Variabilidad del Ritmo Cardíaco (HRV), se pretende determinar inicios de episodios de somnolencia.

Así, en el detector que se propone en [22] se ensaya un método basado en el cociente de la potencia cardíaca en banda LF (0.04–0.15Hz), que se considera una medida de la actividad simpática³, y la potencia cardíaca en banda HF (0,15-0,4 Hz) característica de actividad parasimpática³. El equilibrio entre los sistemas simpático y parasimpático se mide mediante la relación LF / HF.

El predominio del sistema simpático que caracteriza los estados de vigilia disminuye durante el sueño no REM y vuelve a aumentar hasta niveles cercanos a la vigilia durante el sueño REM. Se ha observado una disminución de la frecuencia cardíaca (FC) al inicio del sueño. El período de transición se caracteriza por una disminución en la oscilación de frecuencia muy baja (VLF) de la frecuencia cardíaca que anticipa un cambio en la relación LF / HF a un predominio parasimpático.

2.2.3.3 Ventajas e inconvenientes

Las técnicas basadas en parámetros fisiológicos, especialmente aquellas que se basan en el análisis de las señales alfa del cerebro, tienen una alta precisión y fiabilidad sin embargo su principal desventaja reside en que son altamente intrusivas y por ende tienen bajas posibilidades de ser implantadas en vehículos comerciales, ya que requieren, en muchos casos, del uso de sensores adheridos al sujeto bajo estudio.

Por otro lado, el hecho de usar este tipo de técnicas en entornos de conducción en tiempo real también acarrea problemas de adición de ruido que deben ser abordados en el procesado de dichas señales.

2.3 Soluciones implantadas en la industria de la automoción

2.3.1 Basadas en la dirección del vehículo

- **Bosh driver drowsiness detection**

Solución basada en el análisis de la posición de la dirección más el análisis de cerca de setenta señales⁴.

³ El sistema nervioso autónomo (SNA) controla los actos involuntarios y está formado por el sistema nervioso simpático y parasimpático. El sistema simpático regula las respuestas corporales de activación mientras que el parasimpático es responsable de volver al estado de equilibrio tras la activación del sistema simpático. (Fuente: Psicología-online.com)

⁴ Bosh driver drowsiness detection (Fuente: bosch-mobility-solutions.com)

- **Honda Driver Attention Monitor**

El CR-V es el primer Honda que incorpora un monitor de atención al conductor: esta función utiliza un sensor de ángulo para medir el grado de corrección del volante por parte del conductor para mantener una posición correcta en el carril. Si detecta demasiada actividad de corrección, notificará al conductor que se tome un descanso⁵.

Algunos otros ejemplos de soluciones de otros de fabricantes que optan por la misma tecnología son:

- **Nissan Driver Attention Alert (DAA)**
- **Volkswagen: Fatigue detection system**
- **Renault/Dacia: Tiredness Detection Warning (TDW)**
- **Audi Driver Attention Detection**
- **Mercedes-Benz Attention Assist**

2.3.2 Basadas en el conductor

- **DS Driver Attention Monitoring**

Sistema que combina dos cámaras en el espectro de infrarrojos para la monitorización del los ojos (movimiento y parpadeo) y el cuello. Esta información se combina con el seguimiento continuo de la posición del vehículo [6].

El sistema se comercializa ya en el modelo DS7 Crossback.

2.3.3 Observaciones relativas a sistemas comerciales

A la vista de la prospección de soluciones comerciales existentes actualmente en la industria resulta evidente que se opta por opciones más consolidadas como son las basadas en el posicionamiento de la dirección del vehículo, frente a soluciones basadas en visión artificial y el análisis de parámetros del conductor. Por tanto, para la muestra analizada se observa que la relación de soluciones que optan por la primera tecnología respecto a la segunda es de 7 a 1, lo que posiblemente denote la falta de madurez tecnológica de este tipo de propuestas.

2.4 Patentes

En el siguiente punto se establece una relación de patentes según la tecnología o técnica empleada para detectar la somnolencia, observándose que tras realizar una prospección usando el término genérico “*Drowsiness Detection*” se obtiene que para la muestra extraída existe un mayor número de patentes registradas en el ámbito del análisis del conductor en lugar de las que analizan parámetros de conducción, como parece que es actualmente predominante en la industria. Así, se observa una relación de 7 a 1, lo que podría ser un indicador

⁵ Honda Driver Attention Monitor: hondainfocenter.com

de que existe un gran interés por seguir desarrollando este tipo de tecnologías, si bien aún tienen retos que resolver para que puedan ser implantadas de forma masiva en la industria. No obstante no se puede obviar que este tipo de tecnologías tienen aplicaciones adicionales en otros campos e industrias lo cual también influye en el significativo número de patentes que acaban registrándose.

Basadas en el análisis de la dirección

- Eskandarian, Azim; Mortazavi, Ali, "Unobtrusive driver drowsiness detection system and method"
Patent No. 8519853

Basadas en el análisis del conductor

- Edwards, Timothy; Heinzmann, Jochen; Rougeaux, Sebastien; Zelinsky, Alex, "FACIAL IMAGE PROCESSING SYSTEM"
Patent No. 7043056
- Zhang, Harry; Witt, Gerald J.; Smith, Matthew R., "Drowsiness detection system and method"
Patent No. EP1418082
- Pirim, Patrick; Binford, Thomas, "METHOD AND APPARATUS FOR DETECTION OF DROWSINESS"
Patent No. 6717518
- HULTZSCH, Hagen, "METHOD AND DEVICE FOR IDENTIFYING THE DROWSINESS OF DRIVERS OF MOVING OBJECTS"
Patent No. EP1397268
- Pirim, Patrick, "METHOD AND DEVICE FOR DETECTING DROWSINESS AND PREVENTING A DRIVER OF A MOTOR VEHICLE FROM FALLING ASLEEP"
Patent No. 6304187
- Galiana, Henrietta L.; Hunter, Ian W.; Jones, Lynette A.; Tangorra, James L., "DROWSINESS ALERTNESS MONITOR"
Patent No. 6091334
- Suenaga, Nobumasa; Ogawa, Kenji, "FACIAL IMAGE PROCESSING SYSTEM"
Patent No. 5805720

2.5 Otras aplicaciones

El conocimiento del nivel de fatiga o cansancio de un sujeto resulta de especial interés, y no sólo para su aplicación en vehículos para detectar la somnolencia en el conductor. Así, aquellas actividades que requieren una alta concentración y conllevan altos riesgos para la seguridad propia del sujeto y/o terceros, serían claras candidatas a usar un sistema de detección de fatiga, como podrían ser:

- Cirujanos
- Pilotos de aeronaves
- Controladores de tráfico aéreo
- Artificieros y desactivación de explosivos
- Trabajos con maquinaria pesada

Por otro lado, podría tener especial interés en el ámbito sanitario para el diagnóstico de posibles patologías, como por ejemplo aquellas que se estudian en la unidades del sueño existentes en numerosos centros hospitalarios.

3. Fundamentos y diseño

3.1 Introducción

El presente proyecto pretende abordar un sistema de detección y análisis del nivel de somnolencia de un sujeto bajo estudio usando técnicas basadas en su comportamiento, tal como se explica en el epígrafe 2.2.1. Así, en el trabajo objeto de análisis se persigue analizar el estado de ojos y boca, de manera que combinando los datos obtenidos y procesados se pueda discernir si el sujeto está bajo efectos de somnolencia y/o fatiga.

En el siguiente diagrama de flujo se observa, a más alto nivel, las diferentes etapas que deberán conformar el sistema software de detección de fatiga.

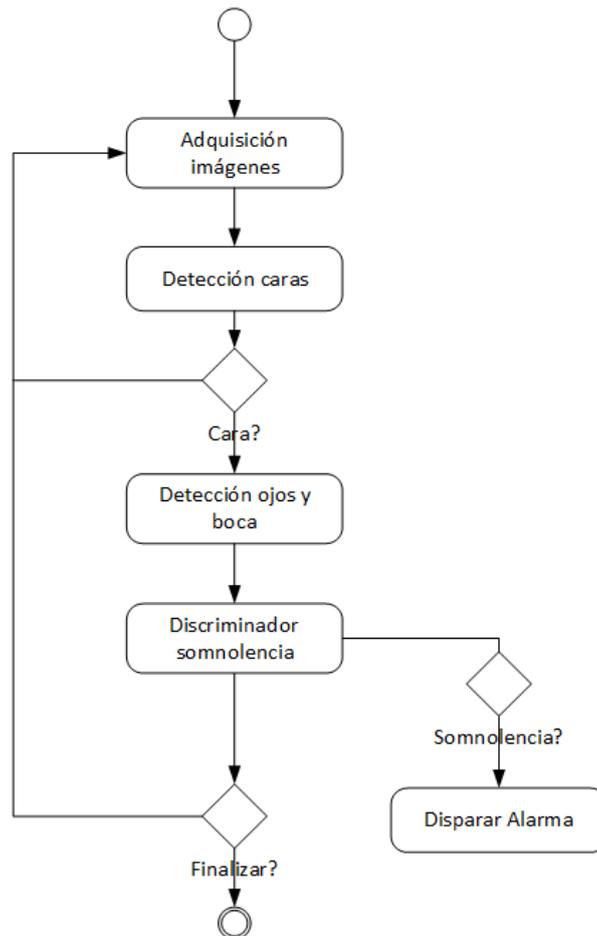


Figura 13. Etapas básicas de detección-predicción somnolencia

3.2 Descripción de etapas

En los siguientes epígrafes se hará una descripción detallada de los algoritmos implementados y el modelo matemático en el que se sustentan siguiendo las etapas descritas en la Figura 13.

3.2.1 Adquisición de imágenes

Como sistema de adquisición de imágenes, y dado el carácter de prototipo demostrador del presente desarrollo, se plantea el uso de una webcam comercial con las siguientes características:

- Resolución mínima: 640x480 píxeles
- Velocidad mínima de captura de imágenes: 10 fps ya que se plantea un tiempo de ciclo máximo de 100ms.
- Ángulo de visión horizontal FOVh mínimo: 45°. La aplicación está diseñada para monitorizar, inicialmente, a un solo sujeto por lo que no será necesario un gran campo de visión horizontal.

Teniendo en cuenta las anteriores restricciones de mínimos, la mayoría de modelos comerciales cumplirán con estos requisitos, por lo que finalmente se opta por el modelo **papalook Webcam 1080P, AF925** que además incorpora capacidad de auto enfoque, especialmente interesante para corregir escenas en las que el sujeto varíe su posición, (aunque las variaciones deberían ser mínimas dado el contexto de trabajo en un entorno estático).

Por otro lado, el modelo seleccionado tiene capacidad de adquirir imágenes a una tasa de hasta 30fps, FOV de 65° y con una resolución de hasta 1920x1080 píxeles: esto permitiría, si la algoritmia y el hardware sobre el que se ejecute la aplicación lo permiten, mejorar las capacidades siempre y cuando se pudiera asegurar el tiempo de ciclo máximo exigido.

Finalmente, el fabricante indica que el modelo incorpora la capacidad de corrección automática del color (típicamente se realiza mediante la ecualización del histograma) lo que permitiría obtener mejores resultados en escena con condiciones de iluminación extremas, tal como se puede apreciar en el ejemplo siguiente:

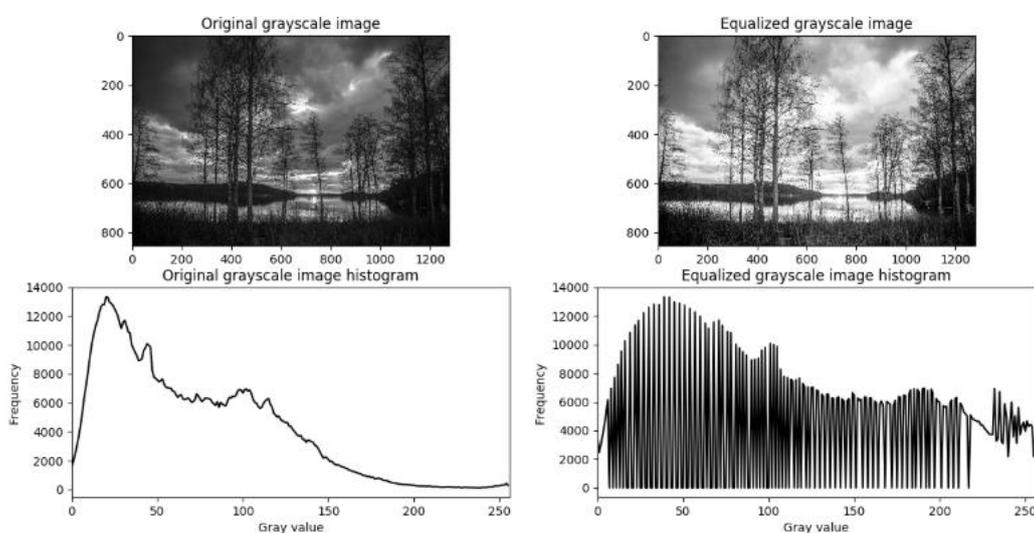


Figura 14. Comparativa de una imagen antes y después de ecualizar el histograma⁶

⁶ Ecualización del histograma: <https://medium.com/@rmdayala/image-histograms-in-opencv-40ee5969a3b7>

Por último, el modelo elegido permite también acoplar un pequeño trípode de mesa, lo cual resulta interesante para conseguir una superficie firme y por ende evitar posibles vibraciones que perjudicarían al procesado de imagen.

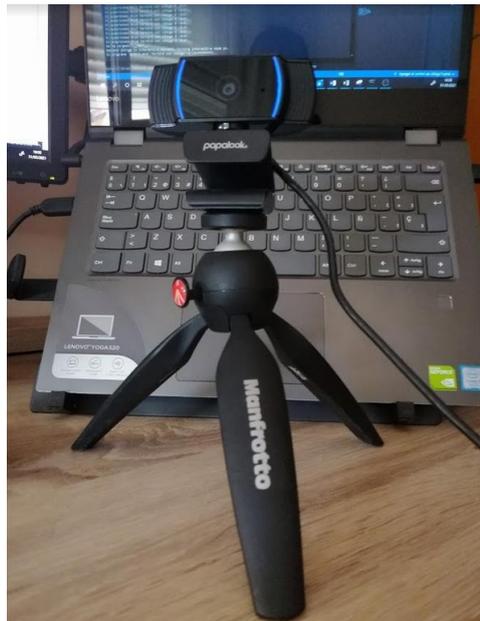


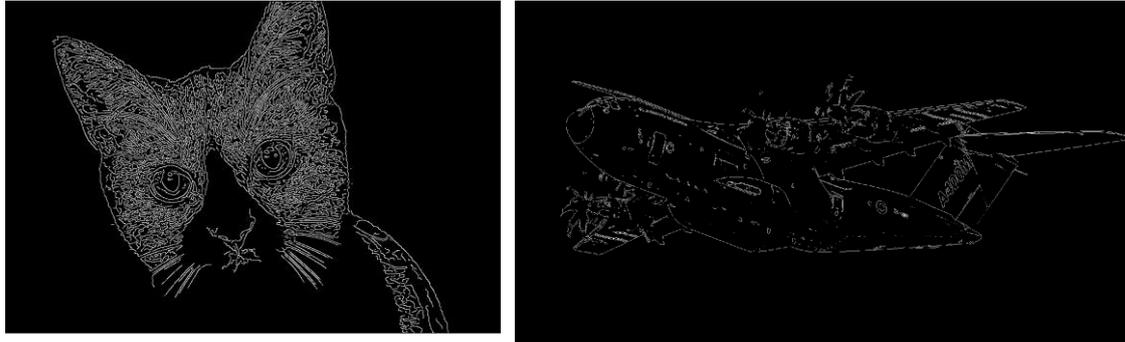
Figura 15. Sistema de adquisición: webcam papalook 1080P

3.2.2 Detección de caras

Para la detección facial, una vez conocidos los diferentes métodos existentes, se ha optado por la técnica del Histograma de los Gradientes Orientados (en inglés Histogram of Oriented Gradients (HOG)) que consiste en la extracción de características en base a descriptores de bordes. Concretamente se establecen ventanas de análisis que se deslizan sobre toda la imagen, y por cada una de ellas se construye un histograma de gradientes y orientación de los bordes, de manera que esto permite conocer cómo de abruptos son los bordes así como su dirección.

Descriptores: Definimos los descriptores como una representación simplificada de la imagen que contiene sólo la información más importante. El concepto se puede apreciar visualmente en la siguiente ilustración de ejemplo.





- a. Detección perimetral usando el método de detección de bordes de Canny (bordes suaves y fuertes)
- b. Detección perimetral usando el método de detección de bordes de Sobel

Figura 16. Imagen procesada mediante un descriptor de bordes (a. Canny b. Sobel)

La idea detrás del uso de descriptores, tal como se deduce de la comparativa expuesta en la Figura 16, radica en que las imágenes inferiores, (después de aplicar los descriptores) contienen menos información que las originales, pero sin embargo siguen conteniendo suficiente información para reconocer el objeto presente en ellas. Esto simplifica el problema considerablemente, además de reducir la carga computacional asociada.

Algunos descriptores típicos:

- HOG: Histogram of Oriented Gradients
- SIFT: Scale Invariant Feature Transform
- SURF: Speeded-Up Robust Feature

3.2.2.1 Introducción al descriptor de característica HOG

El descriptor HOG se centra en la estructura o forma de un objeto cuya principal diferencia con las características de borde de una imagen radica en que no solo se extraen las entidades de borde (si una región es borde o no), sino que HOG también puede proporcionar la dirección del borde. Esto se hace extrayendo el gradiente y la orientación (es decir, magnitud y dirección) de los bordes.

Además, estas orientaciones se calculan en porciones localizadas. Esto significa que la imagen completa se divide en regiones más pequeñas y, para cada región, se calculan los gradientes y la orientación tal como se aprecia en la siguiente figura.

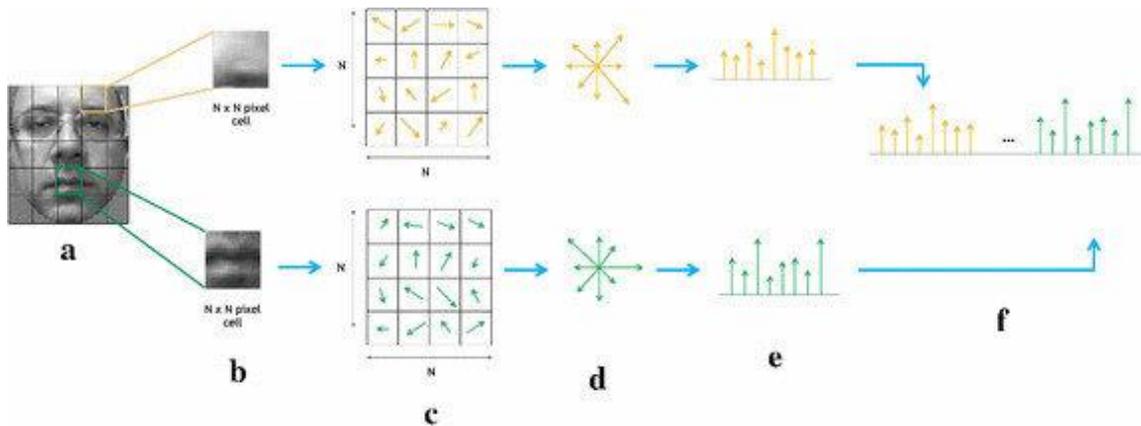


Figura 17. Proceso de extracción de características con HOG, según [29]

En el proceso de extracción de características mediante HOG se observa que la imagen se divide en celdas de tamaño $N \times N$ píxeles. La orientación de todos los píxeles se calcula y se acumula en un histograma de orientaciones de M -bins. Finalmente, todos los histogramas de celda se concatenan para construir el vector de características finales. En el ejemplo de la Figura 17 se usa un tamaño de celda de 4 píxeles y 8 bins de orientación para los histogramas de celda.

3.2.3 Detección ojos y boca

Para la estimación de ojos y boca se hará uso del algoritmo propuesto por Kazemi y Sullivan en 2014, [27] One Millisecond Face Alignment with an Ensemble of Regression Trees de manera que una vez tenemos identificada la región de la imagen que contiene la cara, se empleará esta ROI como entrada para la estimación de marcas faciales, que estimará hasta 194 puntos (x,y) que definen el contorno de la cara, cejas, ojos, nariz y labios, tal como se ilustra en la siguiente figura:

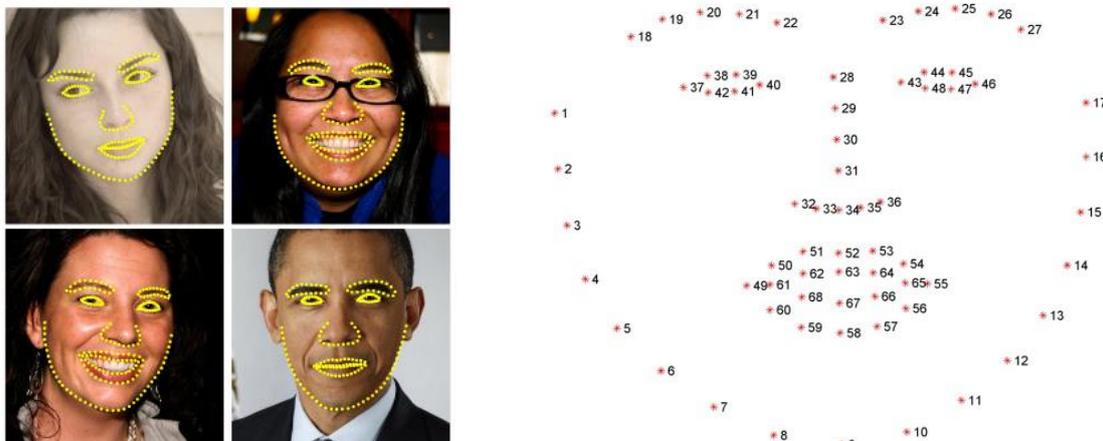


Figura 18. A) Facial landmarks de [27] con 194 puntos. B) Implementación de [27] en Dlib, con 68 puntos.

En este sentido se opta por el uso de la librería Dlib.

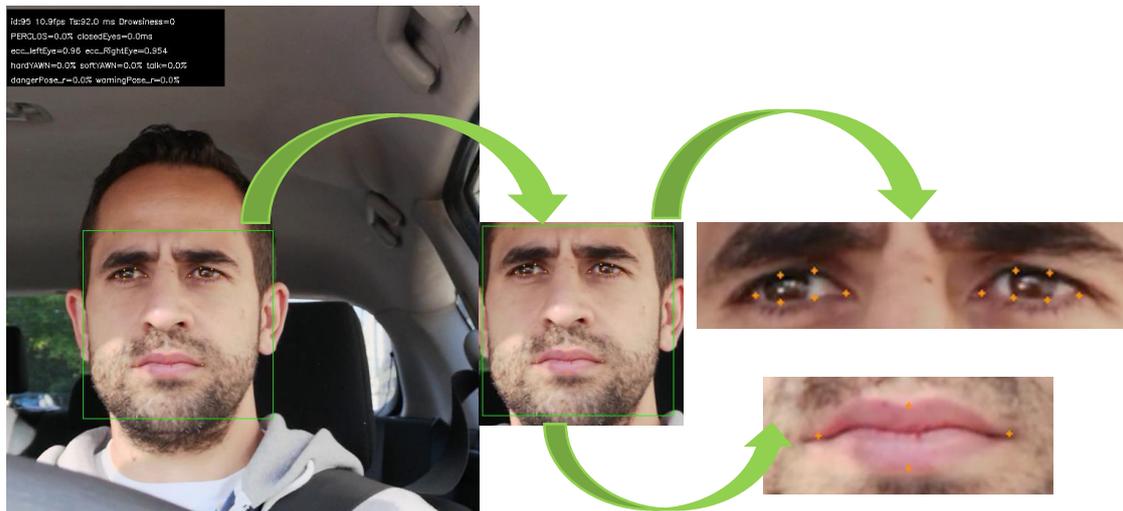


Figura 19. Aplicación de versión implementada en Dlib del algoritmo [27] One Millisecond Face Alignment with an Ensemble of Regression Trees

3.2.4 Discriminador de somnolencia

Una de las partes fundamentales del trabajo será la implementación de un discriminador de somnolencia para lo cual se tendrá que hacer uso de diferentes métricas. A continuación se enumeran las más relevantes y el motivo de su necesidad.

3.2.4.1 Métricas por escena

Por cada escena se calcularán las siguiente medidas:

- **Grado de apertura de los ojos:** tal como se explica a lo largo del capítulo 2, conocer si los ojos se encuentran cerrados o abiertos resulta fundamental para conocer el estado en el que se encuentra un sujeto. Por tanto, habrá que medir dicho grado de apertura para clasificarlos como cerrados o abiertos. Para ello se usará la medida geométrica de la excentricidad.
- **Grado de apertura de la boca:** conocer el grado de apertura de la boca permite clasificar si está cerrada o abierta. Así mismo, si el grado de apertura es muy amplio, se puede distinguir entre escenas de conversación y bostezos. Así se puede llegar a determinar si el sujeto puede estar sintiendo fatiga, cansancio o sueño. Se hará uso también del valor de excentricidad de la boca.
- **Grado de inclinación de la cabeza:** conocer el nivel de inclinación de la cabeza permitirá saber si se desvía de la vertical, lo que permite determinar si se produce cabeceo, lo que denota somnolencia y/o falta atención.

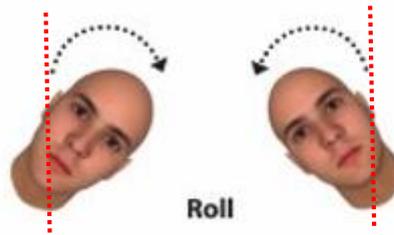


Figura 20. Métrica de inclinación de cabeza

Una vez se obtienen los puntos que definen los contornos, se discriminarán aquellos de interés y que definen el contorno de ojos (izquierdo y derecho) y boca, generando un subvector con los siguientes puntos:

Ojo Derecho	Puntos 37 a 42
Ojo Izquierdo	Puntos 43 a 48
Boca	Puntos [49, 52, 55, 58]*

Tabla 5. Relación de puntos seleccionados para contorno de ojos y boca, según Dlib

* En el caso de la boca, y dentro de la nube de puntos que definen ambos labios, se opta por elegir los 4 puntos cardinales que delimitan los máximos del contorno exterior de los labios, en aras de facilitar el cálculo de la relación de aspecto, tal como se explicara en los siguientes apartados.

3.2.4.1.1 Propiedades geométricas para obtener las métricas por escena

Para el diseño de la presente aplicación se va a recurrir a dos métricas de carácter geométrico, como son la excentricidad y ángulo de la elipse que envuelve cada elemento, por lo que se procede a hacer un breve repaso del modelo matemático en el que se sustentan.

3.2.4.1.1.1 Relación de aspecto: Excentricidad

Una de las bases fundamentales para detectar/predecir signos de fatiga o somnolencia es el análisis del estado de ojos y boca: se trata de identificar cuándo están cerrados o abiertos. Para ello, y conociendo los puntos que delimitan su contorno, se puede recurrir a diferentes cálculos geométricos, tal como apuntan Soukupová y Čech en [28] **Real-Time Eye Blink Detection using Facial Landmarks**.

Sin embargo en el presente trabajo se optará por el cálculo geométrico de la excentricidad de la elipse que contiene cada uno de los objetos detectados.

Cálculo de ejes

En primer lugar, y partiendo de los puntos definidos para cada contorno, obtenemos la distancia euclídea (Figura 21) de separación entre cada par de

objetos medidos en píxeles, calculada tal como se expone en la siguiente ecuación.

$$d(P1, P2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} [px] \quad (2)$$

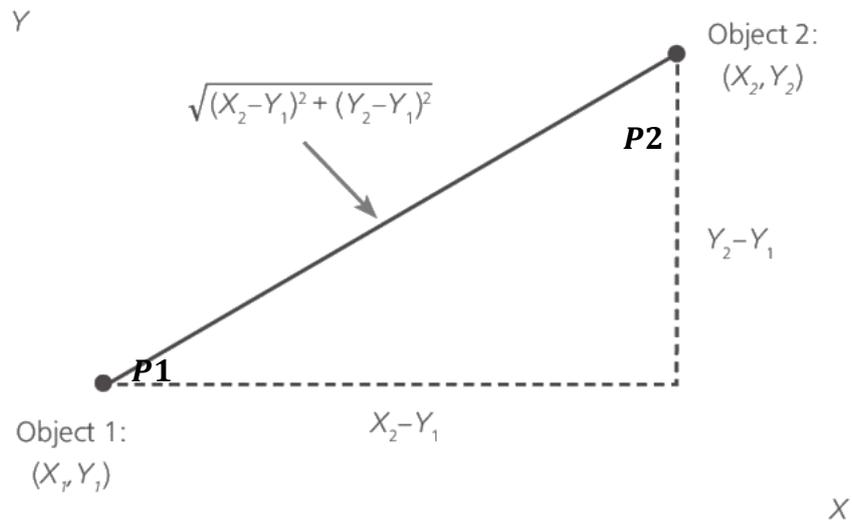


Figura 21: Distancia euclídea entre dos puntos **P1** y **P2**

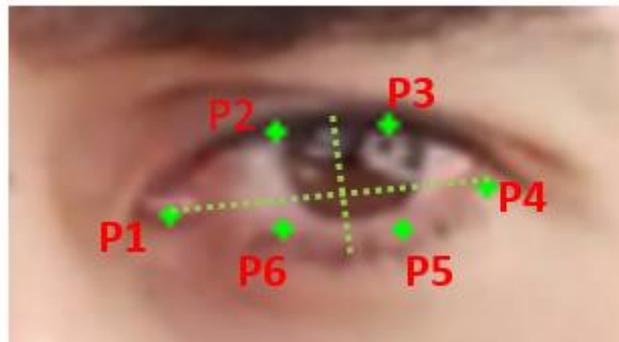


Figura 22. Ejemplo de puntos de referencia para estimar elipse que envuelve el ojo

Según se observa en la Figura 22, en el algoritmo propuesto se calculará el eje mayor como la distancia euclídea entre P1-P4 y el eje menor como el máximo entre las distancias euclídeas de P2-P6 y P3-P5. Esto es:

$$a = \sqrt{(x_4 - x_1)^2 + (y_4 - y_1)^2} [px]$$

$$b = \max(\sqrt{(x_2 - x_6)^2 + (y_2 - y_6)^2}, \sqrt{(x_3 - x_5)^2 + (y_3 - y_5)^2} [px])$$

Cálculo de excentricidad de una elipse

Si bien existen diferentes técnicas y/o ecuaciones que podrían posibilitar la medición de grado de apertura de ojos y boca, con el objetivo de discriminar entre abiertos y cerrados, en el presente trabajo se ha optado por técnica de marcas faciales (que devuelve diferentes puntos de interés) de la posición de los diferentes contornos faciales. Por tanto se hará uso del parámetro geométrico de la excentricidad para determinar el grado de apertura de ojos y boca. La idea básica consiste en asumir que una vez se obtienen los puntos que delimitan el

contorno de ojos y boca, se puede estimar fácilmente el eje mayor y menor de la elipse que contiene dicho objeto. En base a ello, y tal como se expone en la Figura 23, se asumirá que excentricidades cercanas a 1 corresponden con objetos cerrados mientras que valores menores tendrán la consideración de aperturas. El ajuste fino de umbrales requerirá un proceso empírico con reiteradas iteraciones y muestras.

Así, la ecuación que permite obtener el factor de excentricidad de una elipse viene dada por:

$$e = \frac{\sqrt{a^2 - b^2}}{a} \quad (3)$$

En la siguiente figura se observa la relación de aspecto entre elipses con el mismo eje mayor pero con diferente tamaño para el menor: el primer caso, elipse azul, representa la elipse con mismas dimensiones para su eje mayor y menor, esto es, un círculo. Por tanto, y en base a la ecuación (3), obtenemos que la excentricidad es 0. El otro caso extremo, y que se ilustra en rojo, representa aquella elipse cuyo eje menor b vale 0, por tanto, la excentricidad será 0. Geométricamente representa una línea recta.

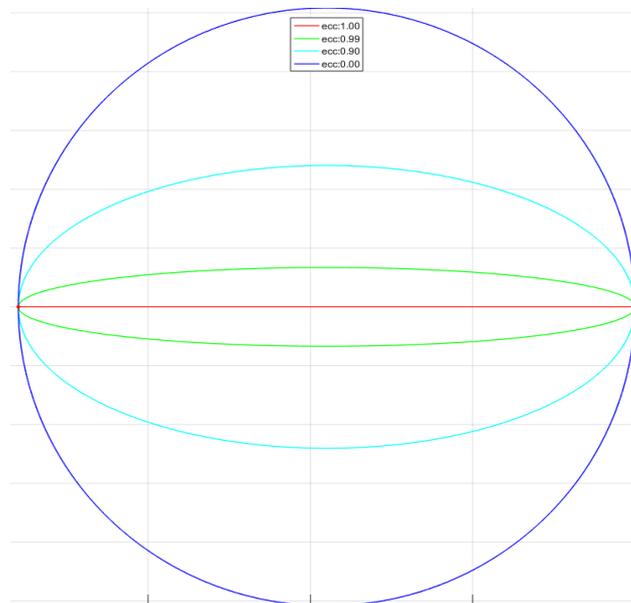


Figura 23. Comparativa entre elipses con diferente grado de excentricidad

3.2.4.1.1.2 Ángulo

Para calcular el ángulo de la cabeza nos bastan dos puntos del contorno de los labios, concretamente aquellos puntos que ubican las comisuras, esto es los dos puntos extremos de los contornos de los labios, lo que nos permite determinar los ejes de la elipse que los envolvería. Una vez conocidos dichos ejes, se puede obtener la inclinación que tiene dicho eje aplicando la arco tangente.

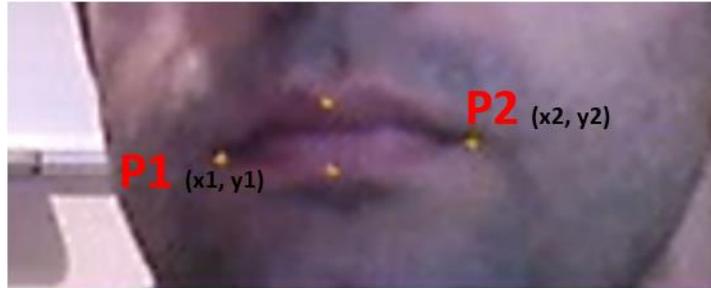


Figura 24. Puntos extraídos de la comisura de la boca mediante facial landmarks

$$\alpha = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$$

3.2.4.2 Métricas temporales

Dado el carácter de procesamiento en tiempo real o psuedo tiempo real que ha de tener un sistema que pretenda detectar síntomas de somnolencia, resulta necesario realizar un estudio de la evolución temporal de las diferentes métricas señaladas en el epígrafe 3.2.4 de manera que se puedan obtener unas métricas más robustas. Se usarán las siguientes:

- Tasa de parpadeos o **PERCLOS**: permite determinar el porcentaje de frames en los que se han detectado los ojos cerrados. Es habitual usar una ventana temporal de 60 segundos. Por tanto, tasas de PERCLOS por encima de un determinado umbral dispararán una alarma.
- **Contador de frames consecutivos con ojos cerrados**. Métrica necesaria puesto que un sujeto que permanezca con los ojos cerrados por un período largo denotaría claramente una situación de somnolencia. El grado de apertura se medirá usando la excentricidad, donde valores cercanos a 1 indican que la boca estará cerrada, mientras que valores inferiores indicarán diferente grado de apertura (mayor cuanto menor sea la excentricidad). Así, si dicho contador excede un determinado umbral se disparará la alamar asociada.
- **Clasificación de una red neuronal** tipo perceptrón multicapa (MLP) con tres neuronas de entrada con las métricas de excentricidad de ojos y boca. Para conferir cierto histórico a los resultados de clasificación se añadirán como entradas las métricas del frame actual y dos anteriores, es decir, nueve neuronas de entrada. Se definirán dos clases para la salida: somnolencia y alerta.
- **Tasa de pose**: esta métrica temporal evaluará el porcentaje de frames en los que el sujeto aparecía en escena con la cabeza inclinada respecto al eje vertical, de manera que si se excede un determinado umbral temporal se disparará la alarma correspondiente.

3.2.4.2.1 Tasa de parpadeos: PERCLOS

Una de las métricas cualitativas más importantes empleadas en el presente desarrollo para determinar el nivel de somnolencia es el PERCLOS, que como se describe en 2.2.1.1 Detección de ojos y tasa de parpadeo, consiste en evaluar la tasa de parpadeos en una ventana temporal, típicamente 60s, si bien en la aplicación propuesta se parametrizará de manera que pueda ser fácilmente modificable. De esta forma, la ecuación quedará como:

$$PERCLOS = \frac{N^{\circ} \text{ de frames con párpados cerrados}}{FRAMES_TO_CALCULATE_PERCLOS} \quad (4)$$

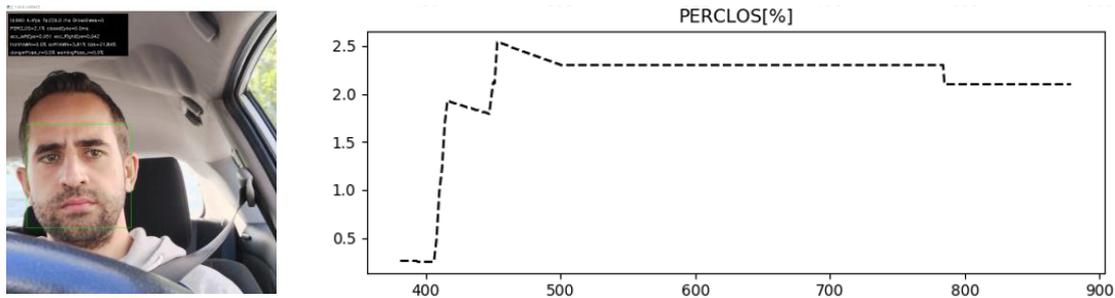


Figura 25. Tasa de PERCLOS a lo largo del tiempo

3.2.4.2.2 Contador de ojos cerrados

Esta métrica se basa en un contador del número de frames consecutivos en los que el usuario permanece con los ojos cerrados. En la siguiente figura se puede ver, de forma ampliada, el contador sobre un sujeto que aparece en escena con los ojos cerrados en varios frames consecutivos.



Figura 26. Contador de ojos cerrados durante 3 frames consecutivos

3.2.4.2.3 Detección de pose

Para la detección del ángulo de pose se realiza el cálculo del ángulo del eje mayor de la boca, es decir el ángulo de la recta que une los dos puntos obtenidos para la comisura de la boca, tal como se explica en el apartado 3.2.4.1.1.2.

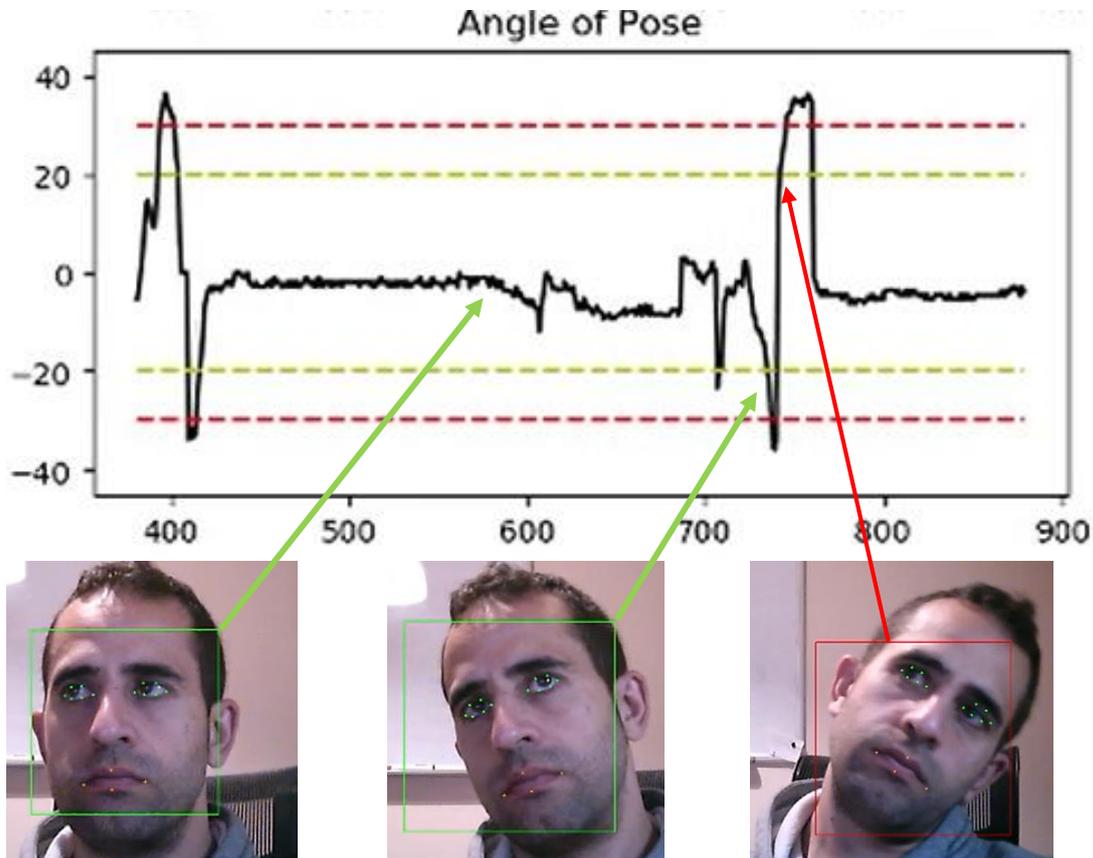


Figura 27. Ángulo de pose en diferentes posiciones

En la Figura 27 se realiza un estudio temporal de su evolución, de manera que se puede obtener la tasa de posiciones comprometidas y críticas para la atención del sujeto y por ende un buen patrón para determinar si se pudiera estar ante un escenario de somnolencia o incluso síncope. Este evolutivo se puede observar en la siguiente figura, donde nuevamente se comparan diferentes ángulos de posición de la cabeza y como el sistema es capaz medirlos e identificarlos.

3.2.4.2.4 Clasificación de estado con red neuronal MLP

En primer lugar hay que indicar que el principal motivo por el que se implementa una red neuronal reside en la necesidad de combinar las diferentes métricas existentes de manera que la red, mediante su adecuado entrenamiento, pueda encontrar el patrón adecuado para discriminar situaciones de somnolencia a partir de los valores medidos. Esta necesidad surge dada la dificultad de lograr un ajuste fino y robusto mediante el uso de umbrales por medidas.

Así, y tal como se ha mencionado anteriormente, la red que se diseña contará con tres neuronas por frame y analizará los datos del frame actual y los dos

anteriores. En este sentido, la justificación de las métricas elegidas se basa en el hecho de que el estado de apertura de boca y ojos ofrece una buena idea del estado del sujeto, si bien éstas métricas no pueden ser consideradas exclusivamente para un frame al objeto de evitar que la red clasifique el estado del sujeto como somnoliento cuando se produce un parpadeo o hay cierta apertura de la boca. A continuación se presenta la variación temporal que sufre el coeficiente de excentricidad en un sujeto de prueba

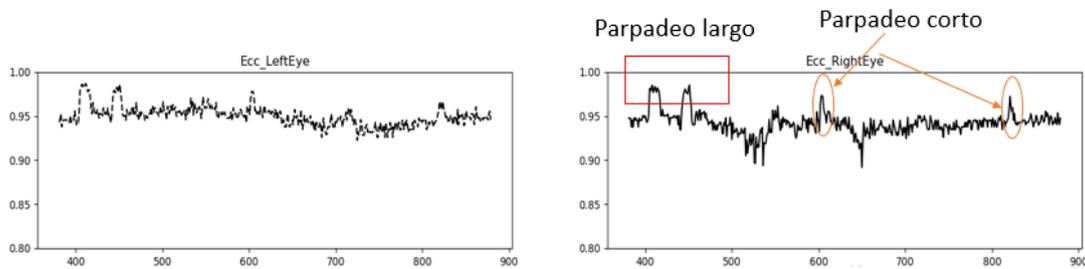


Figura 28. Variación temporal de la excentricidad de los ojos

Por otro lado, y respecto a la segunda métrica de entrada de la red diseñada, el estudio temporal de la evolución del coeficiente de excentricidad de la boca arroja unas métricas especialmente interesantes para determinar el estado de somnolencia del sujeto. En este sentido se pueden llegar a identificar hasta tres escenarios diferentes en base a la evolución de la excentricidad de ésta: boca cerrada, conversación y bostezos, con diferente grado de profundidad, tal como se expone en la siguiente figura.

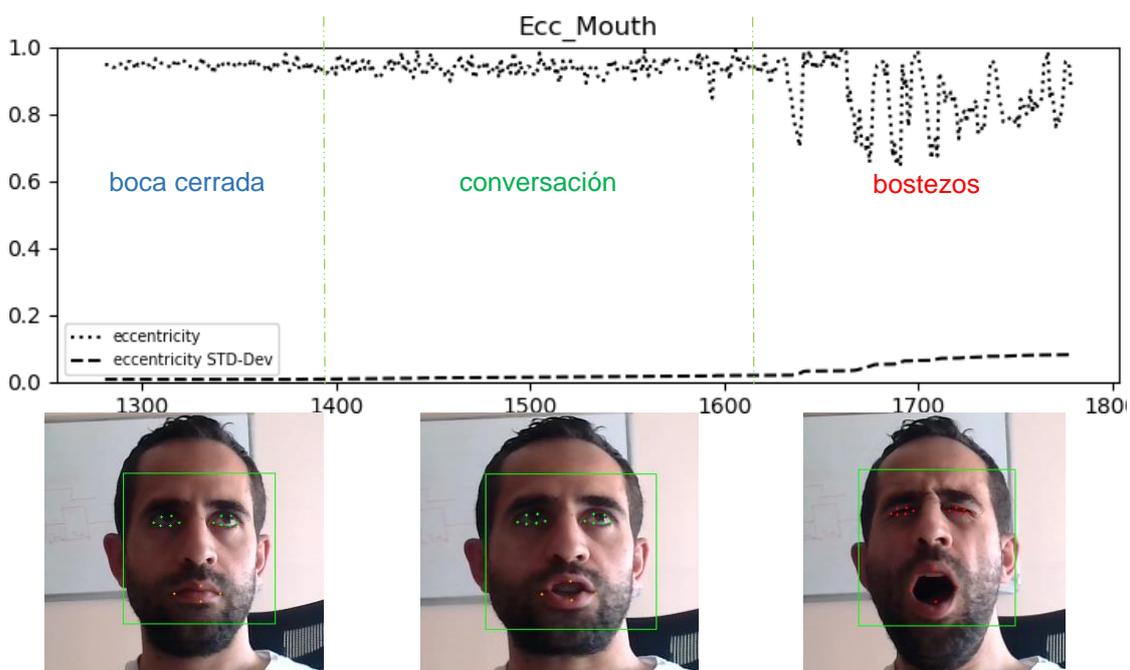


Figura 29. Compartiva de excentricidad en diferentes estados del sujeto

Un estudio a través del histograma permite ver claramente las diferencias de excentricidad para los tres casos expuestos:

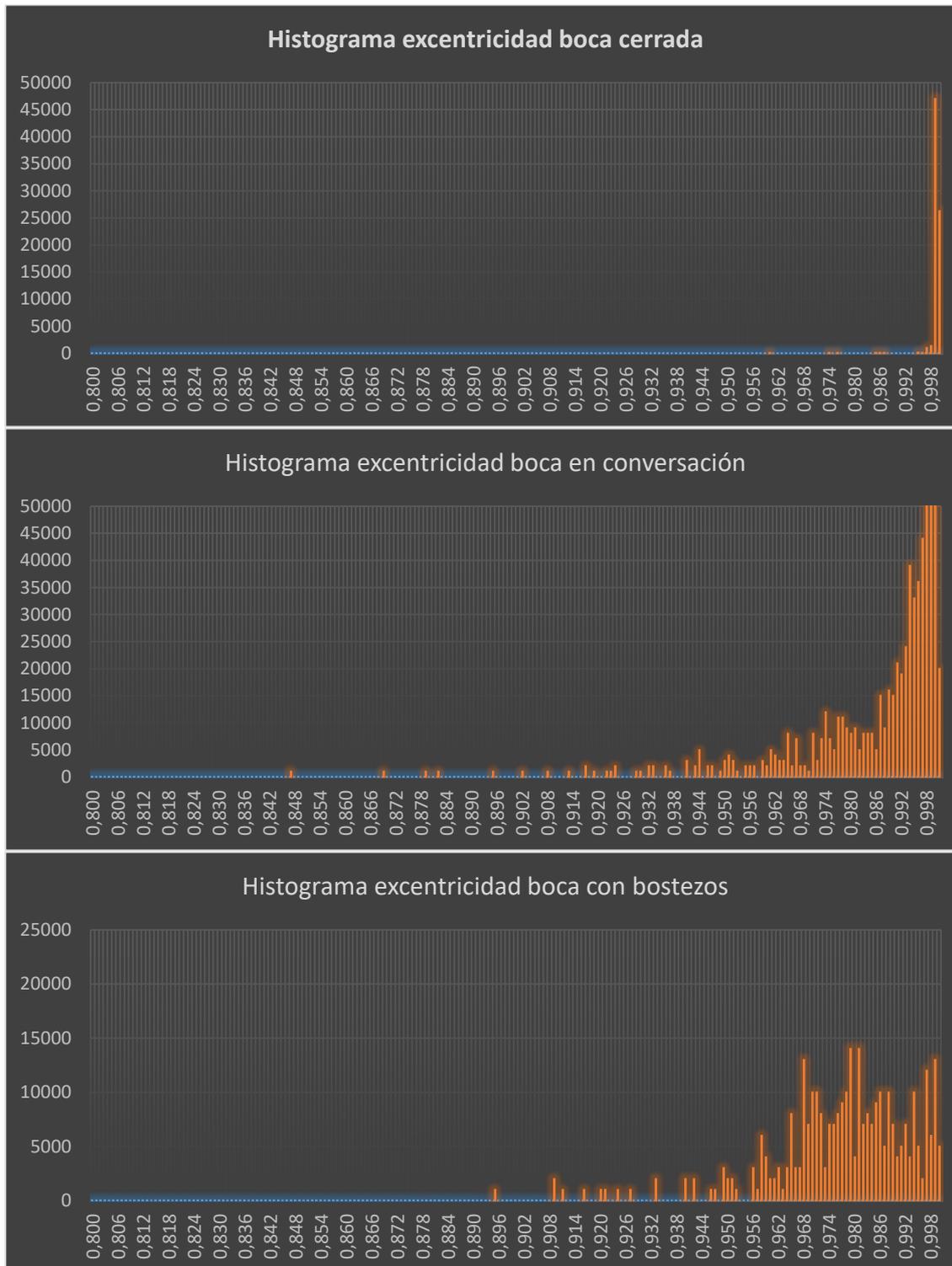


Figura 30. Comparativa de los histogramas de la boca en diferentes estados: cerrada, conversación, bostezos

Para una mejor identificación y comprensión de las variaciones, y por ende de los posibles patrones existentes en la señal que mide la excentricidad para

determinar los estados en los que se encuentra el sujeto, se realiza un estudio del estadístico de la desviación estándar:

$$std_dev = \sqrt{\frac{\sum_{i=1}^N (ecc_i - \overline{ecc})^2}{N - 1}} \quad (5)$$

Donde:

ecc_i : *excentricidad instantánea*

\overline{ecc} : *excentricidad media de la ventana temporal*

Así se observa que, en la primera secuencia de la Figura 29, con el sujeto con la boca cerrada, el valor de excentricidad tiene a ser estable en valores cercanos a 1, de manera que la desviación estándar tiende a 0.

Por otro lado, y como se observa en la ventana marcada como *conversación*, los valores de excentricidad fluctúan suavemente y la desviación estándar gana en amplitud.

Finalmente, y cuando el sujeto bosteza, los valores de excentricidad caen abruptamente, observándose fluctuaciones más profundas mientras que la desviación estándar comienza a crecer.

Por tanto, el modelado de la red neuronal considerando las métricas de excentricidades de ojos y boca, para 3 frames consecutivos tendría el siguiente aspecto a nivel de sistema:

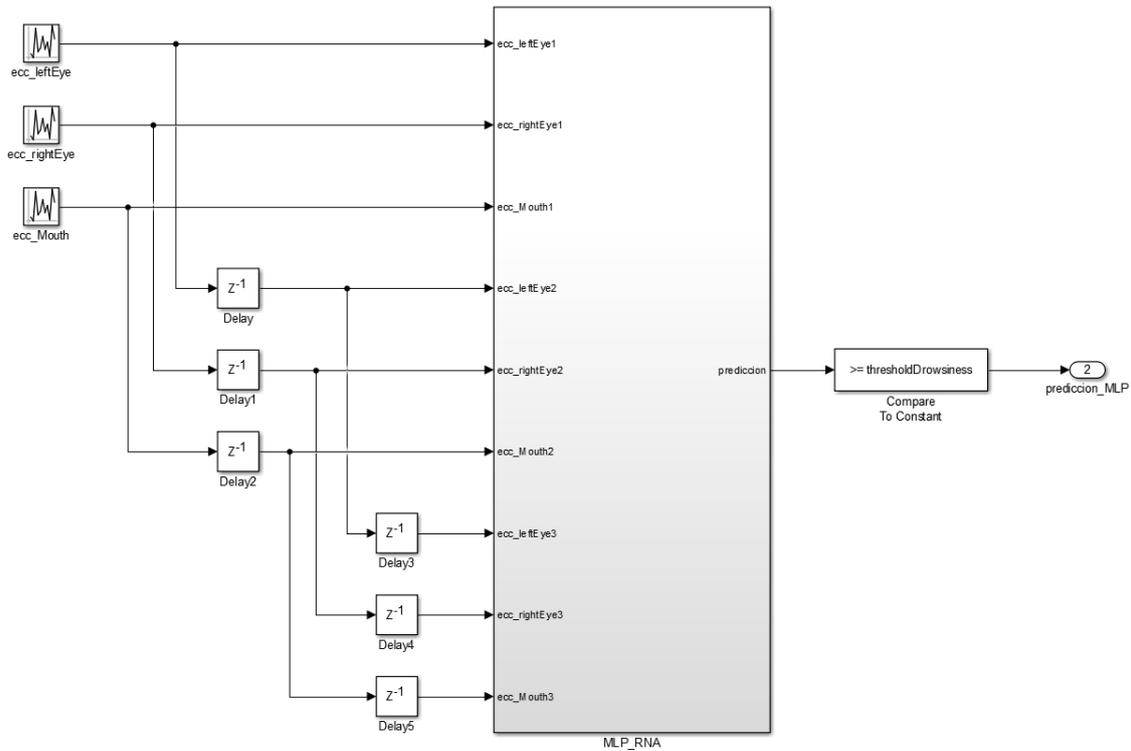


Figura 31. Esquema de MLP para N = 3 frames

3.2.4.2.4.1 Red neuronal Perceptrón Multicapa (MLP): fundamentos

En primer lugar es necesario definir que un perceptrón es un clasificador lineal: un algoritmo que clasifica la entrada separando dos categorías con una línea recta. La entrada es típicamente un vector de características, que se puede definir como x multiplicado por pesos w y agregado a un sesgo, comúnmente denominado $bias$ tal que $y = w * x + bias$.

Un perceptrón produce una salida única basada en varias entradas de valor real al formar una combinación lineal usando sus pesos de entrada, y siendo opcional pasar la salida a través de una función de activación no lineal. Se modela matemáticamente como:

$$y = \varphi \left(\sum_{i=1}^N w_i x_i + bias \right) \quad (6)$$

Donde w_i denota el vector de pesos, x_i es el vector de entradas, $bias$ es el sesgo y φ es la función de activación no lineal.

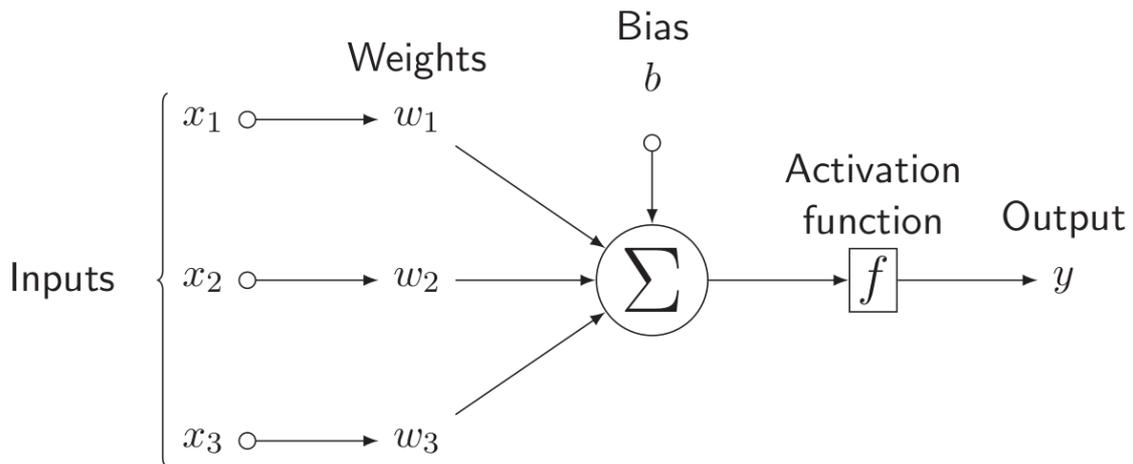


Figura 32. Esquema de perceptrón multicapa ⁷

Por tanto, un perceptrón multicapa (MLP) es una red neuronal artificial que está compuesta por más de un perceptrón. Está formada por una capa de entrada para recibir la señal, una capa de salida que toma una decisión o predicción sobre la entrada, y entre esas dos, un número arbitrario de capas ocultas que son el verdadero motor computacional del MLP. Los MLP con una capa oculta son capaces de aproximarse a cualquier función continua.

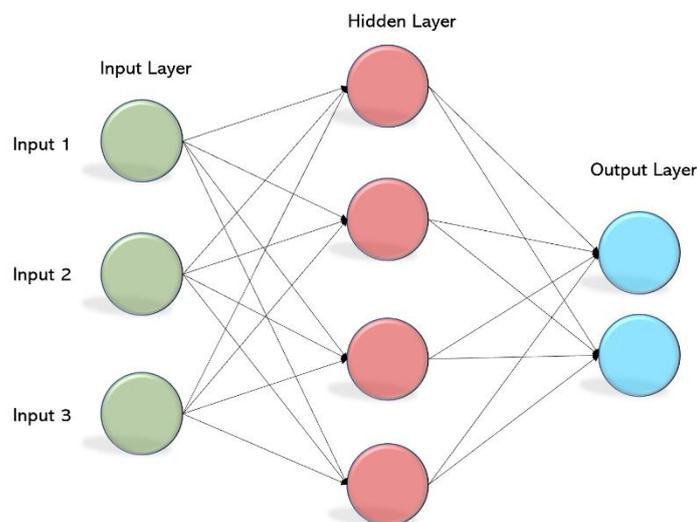


Figura 33. Arquitectura de red MLP

Los perceptrones multicapa se aplican a menudo a problemas de aprendizaje supervisados: se entrenan en un conjunto de pares de entrada-salida y aprenden a modelar la correlación (o dependencias) entre esas entradas y salidas. El entrenamiento implica ajustar los parámetros, o los pesos y sesgos, del modelo para minimizar el error. La retropropagación se utiliza para realizar esos ajustes de ponderación y sesgo en relación con el error, y el error en sí se puede medir de diversas formas, incluido el error cuadrático medio (RMSE).

⁷ Perceptrón multicapa: <https://medium.com/biaslyai/pytorch-introduction-to-neural-network-feedforward-neural-network-model-e7231cff47cb>

En el paso directo, el flujo de señal se mueve desde la capa de entrada a través de las capas ocultas hasta la capa de salida, y la decisión de la capa de salida se mide comparando con la salida real para evaluar el error cometido.

En el paso hacia atrás, utilizando la propagación hacia atrás y la regla de la cadena del cálculo, las derivadas parciales de la función de error con respecto a los diversos pesos y sesgos se propagan hacia atrás a través del MLP. Ese acto de diferenciación nos da un gradiente, o una pendiente de error, a lo largo del cual los parámetros pueden ajustarse a medida que mueven el MLP un paso más cerca del mínimo de error. Este proceso se repite tantas veces hasta que se consiga minimizar el error, lo que se conoce como alcanzar la convergencia. El proceso que se sigue al aplicar la regla de la cadena se puede observar en el siguiente flujo:

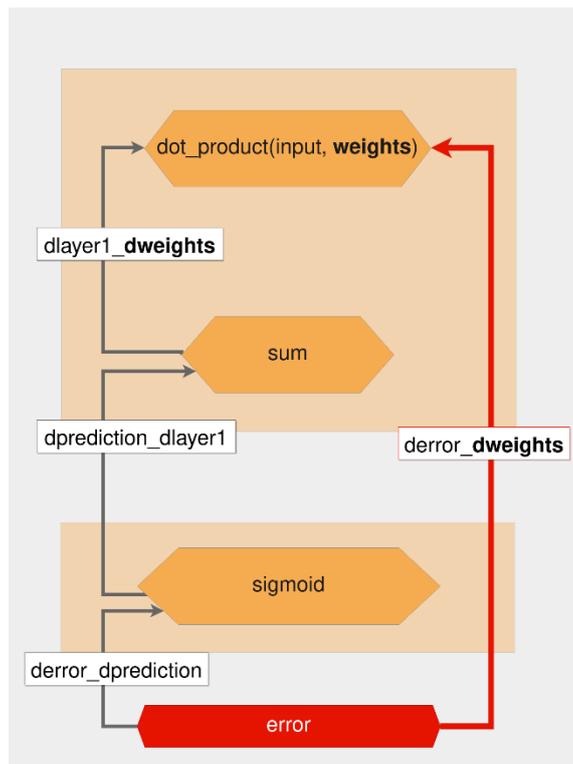


Figura 34. Proceso de cálculo de la regla de la cadena según ⁸

3.3 Parámetros y umbrales

3.3.1 Definición

A continuación se enumeran los parámetros más importantes definidos:

1. Umbrales instantáneos:
 - 1.1. Umbral de excentricidad para considerar ojo cerrado
 - 1.2. Umbral de excentricidad para considerar boca cerrada
 - 1.3. Umbral de salida de la red MLP para clasificar como somnoliento

⁸ Regla de la cadena: <https://realpython.com/python-ai-neural-network>

2. Ventana temporal sobre la que realizar los cálculos
 - 2.1. Número de frames para calcular PERCLOS
 - 2.2. Número de frames para calcular porcentajes de estados
3. Umbrales temporales:
 - 3.1. Porcentaje umbral de PERCLOS para alerta por fatiga
 - 3.2. Porcentaje umbral de alerta por ángulo de pose de riesgo
 - 3.3. Máximo número de frames consecutivos con ojos cerrados
 - 3.4. Umbral de retardo de tiempo para activar alarma por MLP
 - 3.5. Umbral de retardo de tiempo para desactivar alarma por MLP

3.3.2 Consideraciones

Respecto a los valores de excentricidad establecidos para determinar ojos cerrados o con cierre parcial así como los que determinan bostezos se ha realizado un ajuste empírico, dado que otros autores optan por otro tipo de métricas diferentes a la excentricidad, como es el caso de Soukupová y Čech en [28] **Real-Time Eye Blink Detection using Facial Landmarks**.

Resultado también de un proceso empírico se ha llegado a determinar los porcentajes temporales de alertas por ángulo de pose de riesgo. En este caso el motivo se debe a que esta métrica es poco habitual tal como se exponen en el capítulo Estado del arte.

Sin embargo, y para establecer el porcentaje de PERCLOS umbral que denota fatiga, se han usado los umbrales recomendados por los investigadores de la Universidad de Salerno en [31], que consideran la escala de somnolencia Karolinska tal como se ilustra en la Tabla 6:

Karolinska Sleepiness Scale			Post-Processed Sleepiness Scale	
Level	Verbal Description	Vigilance Stage	Level	Verbal Description
1	extremely alert	alertness	0	
2	very alert		1	
3	alert		2	negligible signs of fatigue
			3	
			4	
4	rather alert	hypovigilance	5	
5	neither alert nor sleepy		6	slight fatigue
6	some signs of sleepiness		7	
			8	
7	sleepy, but no effort to keep alert	drowsiness	9	increasing fatigue and drowsiness
8	sleepy, some effort to keep alert		10	
			11	
9	extremely sleepy, fighting sleep		12	

Tabla 6. Escala de somnolencia de Karolinska

Así, en dicha investigación concluyen que un nivel de PERCLOS que denote somnolencia podría ser fijado en una tasa cercana al 33%, tal como se observa en la siguiente figura y por tanto este será el umbral por defecto que se usará en la presenta aplicación.



Figura 35. Umbral de fatiga determinado en [31]

3.4 Discriminación de somnolencia: alarma

3.4.1 Activación

Finalmente, y de acuerdo a las métricas temporal definidas anteriormente el sistema disparará una alarma si cualquiera de las siguientes condiciones se produce:

- A. El contador de ojos cerrados excede su umbral de consigna.
- B. La tasa de PERCLOS excede su porcentaje umbral de consigna.
- C. La tasa de pose peligrosa sobrepasan sus respectivos umbrales de consigna.
- D. El sistema de clasificación de estado implementado mediante la red neuronal arroja una tasa mayor a $thresholdMLP$ durante $thresholdTurnOnDelayMLP$ frames consecutivos.

3.4.2 Desactivación

La desactivación de alarma para los criterios A, B y C se producirá de forma inmediata una vez sus respectivas condiciones de de activación se extingan.

La desactivación de alarma para el criterio D sólo se producirá si la red neuronal arroja una menor o igual a $thresholdMLP$ durante $thresholdTurnOffDelayMLP$ frames consecutivos.

A nivel gráfico, la implementación del sistema de gestión de alarmas puede verse en el siguiente modelo:

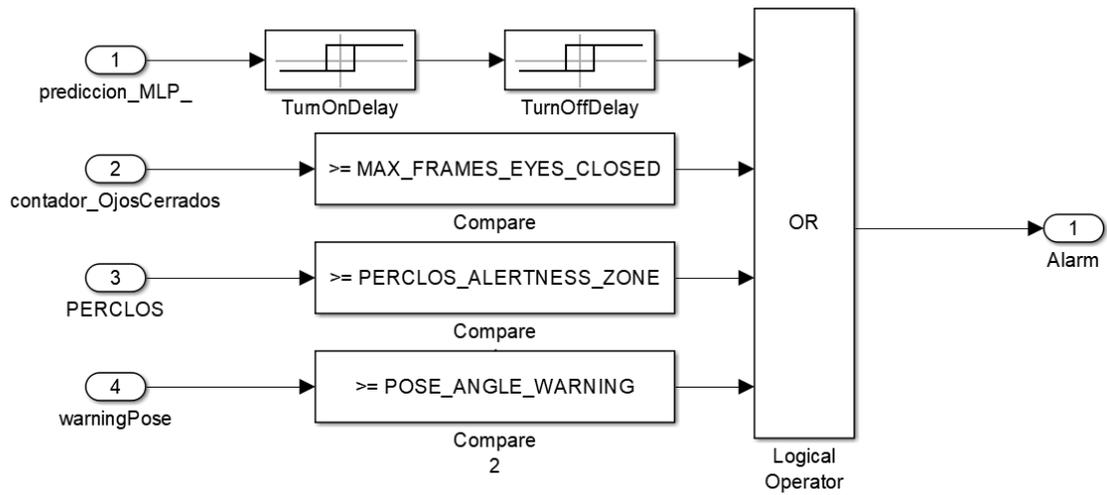


Figura 36. Lógica de activación de alarmas

4 Implementación

4.1 Diseño detallado de etapas

Una vez realizado un diseño con las premisas y consideraciones establecidas en el capítulo anterior se puede crear un flujo de procesado más detallado, tal como se expone en la siguiente figura:

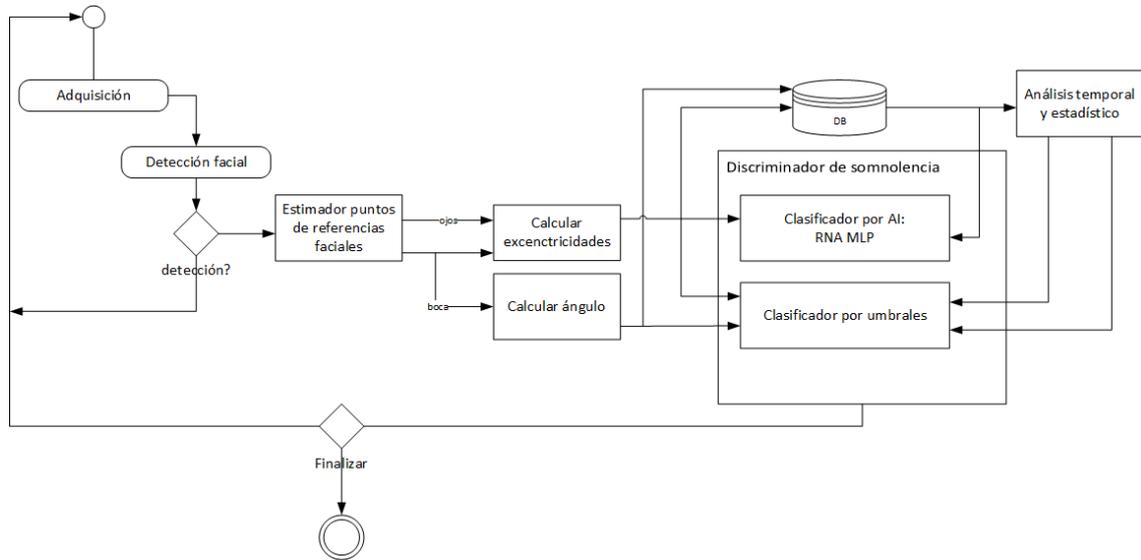


Figura 37. Flujo y etapas de procesado en la solución propuesta

4.2 Especificación

A continuación se detallan una serie de requisitos, funcionales y no funcionales, que ha de cumplir la aplicación.

4.2.1 Requisitos no funcionales

Identificador	Descripción
Req_NF-1	El tiempo de ciclo completo será inferior a 100ms
Req_NF-2	El sistema podrá reducir la imagen a procesar mediante parámetro de entrada
Req_NF-3	El sistema hará detección de cara en un mínimo de un frame por cada dos frames consecutivos, en cuyo caso reutilizará las coordenadas de la detección anterior como argumento de entrada del estimador de coordenadas faciales

Req_NF-4	El sistema implementará una red neuronal tipo perceptrón multicapa con Nx3 neuronas. Donde N será el número de frames. Los valores de entrada corresponderán con las excentricidades de los ojos y la boca
Req_NF-5	El sistema mostrará datos actualizados cada, al menos, M frames sobre: <ul style="list-style-type: none"> • Valores de excentricidad de ambos ojos • Valor de excentricidad de la boca • Ángulo de la pose
Req_NF-6	El sistema calculará la tasa de frames por segundos a la que completa cada ciclo

Tabla 7. Requisitos no funcionales

4.2.2 Requisitos funcionales

Identificador	Descripción
Req_F-7	El sistema hará adquisición de imágenes en cada ciclo
Req_F-8	El sistema convertirá las imágenes a procesar escala de grises
Req_F-9	El sistema debe detectar caras en escena usando imágenes en escala de grises devolviendo una ROI identificada por coordenadas de origen y tamaño
Req_F-10	El sistema determinará las coordenadas faciales de contorno de ojos y boca por cada ROI en la que se haya detectado una cara
Req_F-11	El sistema calculará las excentricidades de ambos ojos usando 6 puntos de contorno por cada ojo
Req_F-12	El sistema calculará la excentricidad de la boca usando dos puntos que indique la posición de la comisura
Req_F-13	El sistema almacenará los valores de excentricidades de ojos de hasta FRAMES_TO_CALCULATE_PERCLOS anteriores al frame actual
Req_F-14	El sistema almacenará los valores de ángulo de boca de hasta FRAMES_TO_CALCULATE_RATES anteriores al frame actual
Req_F-15	El sistema almacenará los valores de excentricidad de boca de hasta FRAMES_TO_CALCULATE_RATES anteriores al frame actual
Req_F-16	El sistema considerará ojo cerrado si se supera el umbral de excentricidad EYES_ECC_THRESHOLD_CLOSED
Req_F-17	El sistema contará el número de frames consecutivos en los que se detecta ojo cerrado
Req_F-18	El sistema activará una señal de alarma si se supera el umbral de MAX_FRAMES_EYES_CLOSED frames con ojos cerrados

Req_F-19	El sistema calculará la tasa de PERCLOS como el cociente entre el número de frames con los ojos cerrados y <code>FRAMES_TO_CALCULATE_PERCLOS</code> evaluado para los últimos <code>FRAMES_TO_CALCULATE_PERCLOS</code> frames
Req_F-20	El sistema activará una señal de alarma si la tasa PERCLOS supera el umbral de <code>PERCLOS_ALERTNESS_ZONE</code>
Req_F-21	El sistema activará una señal de alarma si la red neuronal clasifica los valores de entrada como somnolencia
Req_F-22	El sistema activará un aviso acústico si la señal de alarma se activa
Req_F-23	El sistema activará un mensaje de alarma si se activa la señal de alarma

Tabla 8. Requisitos funcionales

4.3 Librerías

En base al diseño propuesto y los algoritmos más complejos de los que será necesario hacer uso, se decide usar las siguientes librerías:

- OpenCv: Librería de visión por computador que permite la adquisición de imágenes desde vídeos en disco o desde una cámara.
- Dlib: Implementa el algoritmo de estimación de contornos faciales [27] y un detector de caras basado en la técnica de histograma de gradientes orientados (HOG)

El resto de librerías auxiliares se detallarán en posteriores epígrafes dado que no tienen un peso relevante en la elección del lenguaje de desarrollo.

4.4 Lenguaje de programación

Una vez definidas las dos principales librerías de las que habrá que hacer uso, existen dos posibles alternativas en cuanto al lenguaje de programación a usar:

- C++
- Python

No obstante, y aunque C++ otorga un mejor desempeño y eficiencia al ser un lenguaje compilado y no interpretado como Python, se decide apostar por este último dada su fácil sintaxis y por el hecho de no requerir ser compilado, lo que le otorga ventaja en un entorno de prototipado y validación conceptual. No obstante, si la solución tuviera que ser desplegada en un entorno de producción probablemente se planteara una implementación en C++.

4.5 Entorno de desarrollo

Una vez conocido el lenguaje de desarrollo, se decide implementar la solución sobre un IDE como Microsoft Visual Studio Community 2017 con la instalación

del intérprete de Python 3.7. Inicialmente el desarrollo se realizará sobre el sistema operativo Windows10.

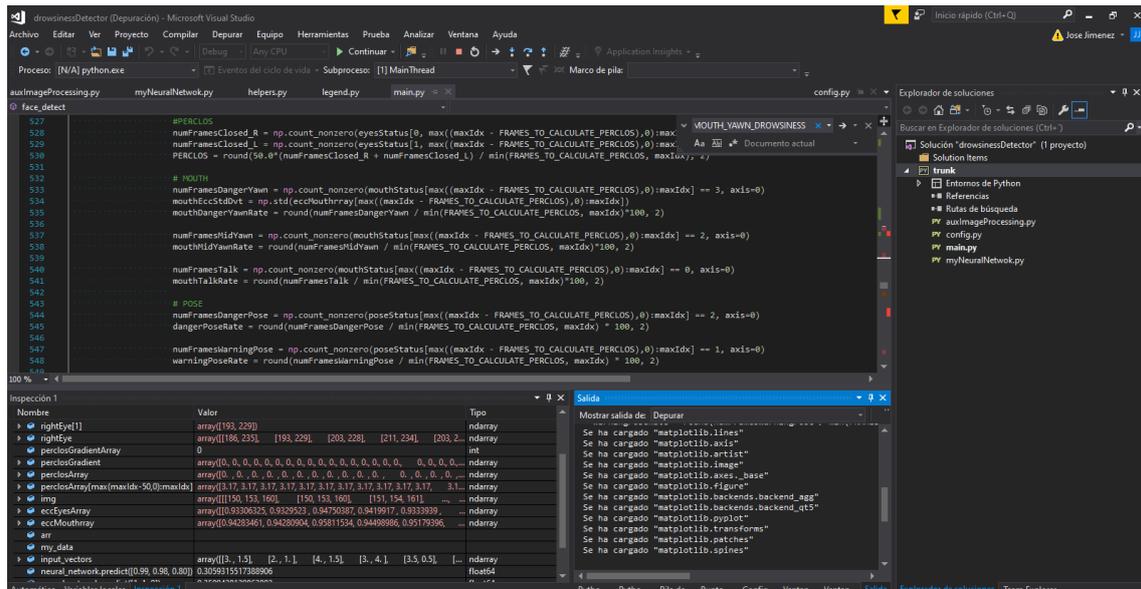
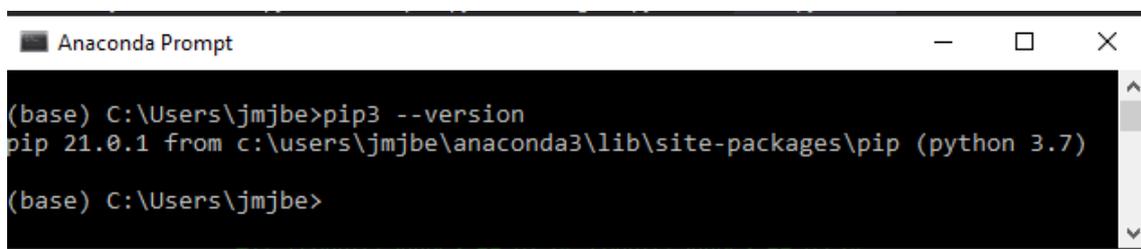


Figura 38. Microsoft Visual Community 2017



4.6 Control de versiones

Para la gestión y control de cambios y configuración se hará uso de una instalación local de Subversion (SVN), dada su facilidad de uso y de que el desarrollo será llevado a cabo por un único desarrollador.

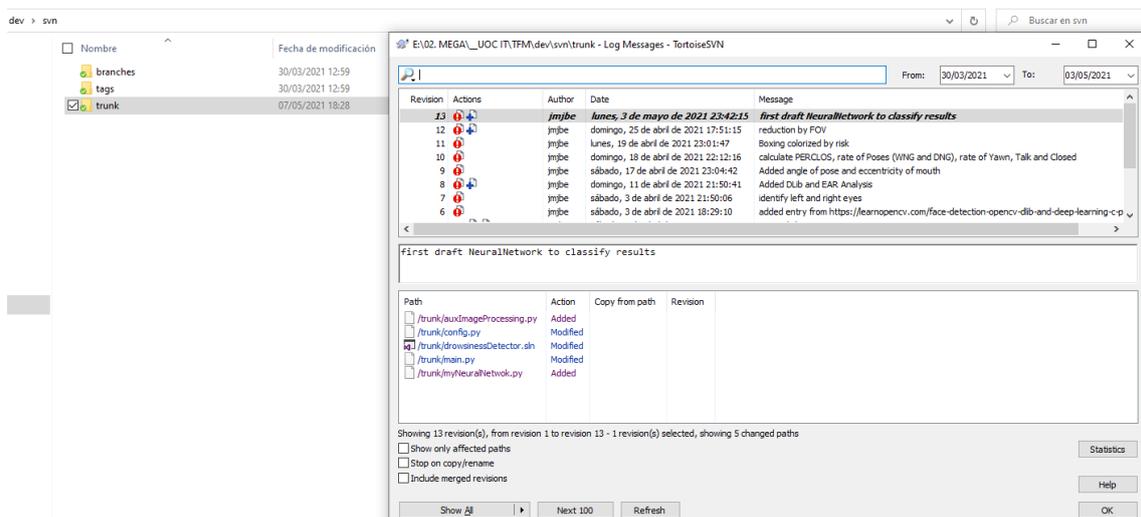


Figura 39. Vista de registro de commits de SVN

4.7 Configuración e instalación

La instalación de librería resulta bastante sencilla usando el intérprete de Python haciendo uso de siguiente comando:

```
pip install <nombreLibrería>
```

Donde las diferentes librerías que deben ser instaladas se enumeran a continuación.

4.7.1 Librerías y dependencias

Para el desarrollo de la solución propuesta se requiere la instalación de las siguientes librería de Python, cuya aplicación se describe a continuación:

<i><nombreLibrería></i>	Aplicación
numpy	Cálculo numérico y uso de vectores y matrices así como múltiples operaciones matemáticas
math	Funciones matemáticas auxiliares como el arcotangente
scipy.spatial	Cálculo de distancias euclídeas y otras operaciones matemáticas auxiliares
cv2	OpenCV: librería de visión por computador ampliamente extendida. Necesaria para la adquisición desde la cámara y procesado de las imágenes
matplotlib	Permite mostrar gráficas, especialmente útiles para la fase de depuración de la aplicación
threading	Permite crear y gestionar diferentes hilos de ejecución. Necesario para disparar la alarma sonora sin necesidad de bloquear el procesado de sucesivos frames
playsound	Permite reproducir ficheros de audio
dlib	Librería de IA que implementa [27] One Millisecond Face Alignment with an Ensemble of Regression Trees

Tabla 9. Lista de librerías de Python

4.8 Desarrollo software

Dentro del desarrollo software se definen dos etapas, dada la singularidad de las redes neuronales y la necesidad de entrenarlas para establecer los pesos adecuados al problema a resolver. Por tanto, en una primera fase hay entrenar y ajustar la red siguiendo una metodología concreta y una preparación previa de datos previos, que tiene una alta importancia en el éxito de la red.

En una segunda parte se abordarán los detalles propios de la solución propuesta.

4.8.1 Red neuronal MLP: implementación

Para la implementación de la red neuronal existen numerosas librerías de código abierto, tales como *scikit-learn*⁹, *PyTorch*¹⁰ o *TensorFlow*¹¹ y que podrían integrarse dentro de la solución propuesta, si bien se ha optado por realizar la implementación desde cero, usando exclusivamente la librería de cálculo matemático *NumPy* por las siguientes razones:

- El uso de alguna librería específica para aprendizaje automático como las mencionadas facilitan la labor de implementación, si bien requieren un mínimo conocimiento sobre los fundamentos matemáticos sobre los que se construyen, ya que permite abstraerse de la implementación a más bajo nivel, pero no por ello se puede obviar la necesidad de entender lo que sucede matemática y computacionalmente a dicho nivel.
- Por otro lado, el uso de alguna de estas librerías requerirá necesariamente un mínimo estudio de la documentación técnica y de usuario, con la consiguiente inversión de tiempo y esfuerzo de desarrollo.
- La ventaja que inicialmente supone disponer de varias librerías de código abierto también tiene el inconveniente de que habría que realizar un estudio/análisis comparativo de las fortalezas e inconvenientes de cada una de ellas para afrontar el problema a resolver. Además, y en el caso de que el desarrollador se inicie en esta disciplina técnica, la realización este análisis no sería una tarea trivial.

Así, se ha definido una clase *myNeuralNetworkMLP* que tendrá los atributos y métodos que se especifican en el siguiente diagrama de clase UML.

⁹ *scikit-learn*: <https://scikit-learn.org>

¹⁰ *PyTorch*: <https://pytorch.org>

¹¹ *TensorFlow*: <https://www.tensorflow.org>

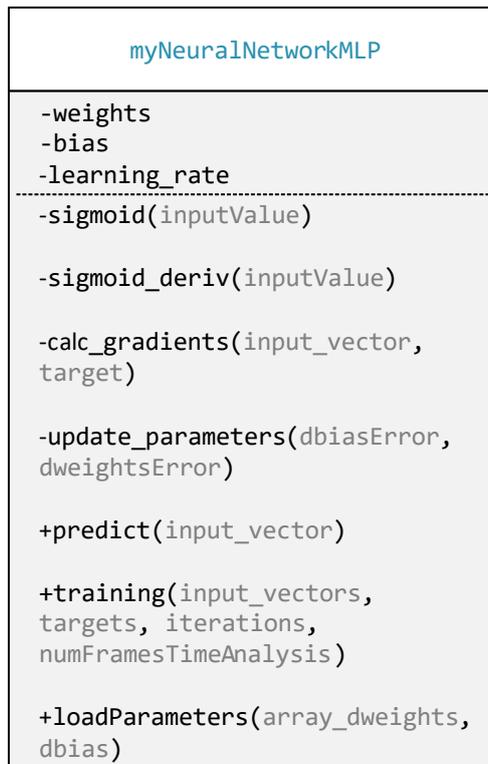


Tabla 10. Diagrama UML de la Clase que implementa la red neuronal MLP

Siguiendo el esquema de red representado en la Figura 33. Arquitectura de red MLP y el diagrama de clase UML especificado, se definen unos miembros con diferentes niveles de acceso (“+” públicos y “-“ privados). A continuación, se hará una breve introducción de las características y finalidad de cada uno de ellos:

- Los atributos, tanto los pesos como el parámetro de sesgo *bias*. Cada uno de los pesos va asociado a una neurona de entrada (medida de entrada). Inicialmente, y una vez se instancia un objeto de la clase, tanto los pesos como el sesgo serán inicializados con un valor aleatorio.
- Método *predict()* realiza la estimación de la clase a la que pertenece cada juego de datos de entrada, asignado un valor entre 0 y 1, donde 0 representa despierto y 1 dormido. Este calculado lo realiza en dos pasos:

1. Capa 1: calcular el producto escalar entre el vector de entrada con las medidas y el vector de pesos a la cual se le suma el parámetro de sesgo. Como los vectores de entrada se expresan según sus coordenadas cartesianas rectangulares, el producto escalar se realiza como un producto matricial:

$$\begin{aligned}
 \text{capa}_1 &= A \cdot W = [a_1 \quad a_2 \quad \dots \quad a_n] \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_3 \end{bmatrix} + \text{bias} \rightarrow \\
 \text{capa}_1 &= a_1w_1 + a_2w_2 + \dots + a_nw_n + \text{bias}
 \end{aligned}$$

2. Capa 2: calcular el valor de la predicción usando el valor obtenido en la capa 1 como entrada de la función de activación, que en este caso se calculará con el método auxiliar *sigmoid()*.

$$predicción = \frac{1}{1 + e^{-capa_1}}$$

- Método *sigmoid()* representa la segunda capa de la red e implementa la función de activación, que para el presente trabajo se ha elegido de tipo sigmoide¹², y que tal como se aprecia en la siguiente gráfica devuelve como salida un valor acotado entre 0 y 1.

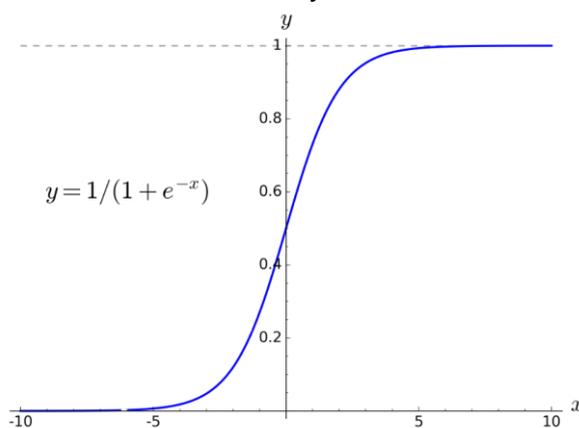


Figura 40. Curva de la función sigmoide

El resto de métodos son necesarios, exclusivamente, para la etapa de entrenamiento, por lo que serán explicados en el siguiente apartado.

4.8.2 Red neuronal MLP: entrenamiento

Los pasos a seguir para conseguir un correcto entrenamiento son:

- Crear un conjunto de datos de entrada con las métricas obtenidas tras procesar vídeo con sujeto en alerta y con las que se pretende estimular la red una vez forme parte de la solución.
- Crear un conjunto de datos de entrada con las métricas obtenidas tras procesar vídeo con sujeto con somnolencia.
- Establecer las salidas para cada uno de los set de datos.

A continuación se muestran algunos ejemplos de los datos usados para realizar el entrenamiento, de manera que se puede apreciar la diferencia entre valores de entrada de un sujeto despierto y en alerta y un segundo sujeto con claros signos de somnolencia.

Ecc_IE	Ecc_rE	Ecc_boca	Y
0.942616	0.953939	0.850333	0
0.942616	0.955925	0.850586	0
0.949155	0.948087	0.858130	0
0.943156	0.956210	0.833607	0
0.945698	0.956210	0.819209	0
0.946026	0.955925	0.799716	0

¹² Funciones sigmoideas: funciones que se definen en el cuerpo de los números reales con forma de "S" <https://core.ac.uk/download/pdf/328836064.pdf>

0.946026	0.950892	0.815827	0
0.951435	0.963337	0.802540	0
0.942809	0.955898	0.796552	0
0.943156	0.958043	0.790387	0
0.936784	0.955898	0.845754	0
0.951616	0.953641	0.846504	0
0.954057	0.963051	0.925291	0
0.953969	0.958019	0.931508	0
0.951272	0.945714	0.938359	0

Figura 41 Muestra de valores x de excentricidad del conjunto de datos de entrenamiento. Salida “despierto”

Ecc_IE	Ecc_rE	Ecc_boca	Y
0.991769	0.993448	0.943215	1
0.991667	0.993037	0.943215	1
0.988209	0.993453	0.949615	1
0.987979	0.992620	0.948540	1
0.994885	0.994143	0.957172	1
0.994885	0.994143	0.957172	1
0.984871	0.986774	0.961116	1
0.986013	0.986774	0.952070	1
0.965011	0.971373	0.956305	1
0.962848	0.964791	0.957888	1
0.978570	0.980656	0.952537	1
0.983262	0.989926	0.953881	1
0.991071	0.993448	0.958370	1
0.992212	0.996522	0.952070	1
0.992212	0.994139	0.944908	1

Figura 42. Muestra de valores x de excentricidad del conjunto de datos de entrenamiento. Salida “somnoliento”

Con ellos ya se habría definido el conjunto de datos para una aprendizaje supervisado de la red neuronal en base a unos vectores x, y conocidos.

Para el entrenamiento de la red se ha construido un dataset de datos de entrenamiento que se ha obtenido mediante la extracción de las métricas de interés de un juego de vídeos etiquetados como “somnolencia” y “alerta”. De esta manera, las métricas obtenidas están etiquetadas, lo que permite iniciar el aprendizaje supervisado ya que se conoce el resultado deseado o esperado. El total de datos de entrenamiento, considerando que cada juego de datos consta de tres métricas ha sido:

- 1976 juegos de datos para sujeto en alerta
- 1438 juegos de datos para situación de somnolencia

Así mismo, el proceso de validación de entrenamiento de la red se ha realizado sobre dos vídeos etiquetados previamente, de manera que se obtiene una tasa

de acierto del 87.10% si bien la red demuestra mejor compartamiento para la detección de somnolencia (98.58%) que para clasificar situaciones de alerta (75.64%), tal como puede verse en la siguiente tabla:

Sujeto bajo análisis	frames analizados	despierto	somnolencia	Tasa acierto
sujeto despierto	4729	3577	1152	75,64%
sujeto somnoliento	4723	67	4656	98,58%
TOTAL	9452	3644	5808	87,10%

Tabla 11. Tasa de acierto de red neuronal

Finalmente, y para realizar este proceso de diseño, se define una clase que implementa la red neuronal y en la que se crea un método encargado de realizar el entrenamiento, tal como se muestra a continuación:

```
class NeuralNetwork:
...
def training(self, input_vectors, targets, iterations):
    cumulative_errors = []
    for current_iteration in range(iterations):
        # Pick a data instance at random
        random_data_index = np.random.randint(len(input_vectors))
        input_vector = input_vectors[random_data_index]
        target = targets[random_data_index]
        # Compute the gradients and update the weights
        derror_dbias, derror_dweights = self._compute_gradients(
            input_vector, target
        )
        self._update_parameters(derror_dbias, derror_dweights)
        # Measure the cumulative error for all the instances
        if current_iteration % 100 == 0:
            cumulative_error = 0
            # Loop through all the instances to measure the error
            for data_instance_index in range(len(input_vectors)):
                data_point = input_vectors[data_instance_index]
                target = targets[data_instance_index]
                prediction = self.predict(data_point)
```

```
error = np.square(prediction - target)

cumulative_error = cumulative_error + error

cumulative_errors.append(cumulative_error)

return cumulative_errors
```

4.8.3 Solución

4.8.3.1 Adquisición de imágenes

Para la adquisición de imágenes, se hará uso de la librería OpenCV, instanciando un objeto de la clase VideoCapture(n) donde n es un número que indica el número del dispositivo. Por defecto, y cuando sólo existe una cámara conectada al sistema, será 0 si se usa un portátil con cámara integrada o 1, como en el presente trabajo, si se conecta una cámara a través del puerto USB.

```
cap = cv2.VideoCapture(1)
```

Una vez tenemos correctamente instanciado el objeto que controla la cámara, tan solo será necesario hacer lecturas periódicas con el método read()

```
ret, img = cap.read()
```

4.8.3.2 Detección de cara

Para la detección de cara instanciamos un objeto de la clase incluida en DLib como frontal_face_detector():

```
detector = dlib.get_frontal_face_detector()
```

4.8.3.3 Estimador de puntos de referencias faciales

Para la estimación de los puntos faciales instanciamos un objeto de la clase, también de Dlib, shape_predictor():

```
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
```

Una vez se adquieren imágenes, es necesario convertirlas a escala de grises usando la clase de OpenCV, cvtColor()

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Cada imagen en escala de grises será el parámetro que habrá que pasar al detector de dlib instanciado previamente:

```
facesDetected = detector(gray, 1)
```

Que devolverá un vector con una lista de ROIs en las que se ha detectado una cara. Así, si hay detecciones, esa ROI será la región de la imagen de entrada que se pasará como argumento de entrada al predictor de marcas faciales:

```
shape = predictor(gray, rect)
```

Obteniéndose un vector con 68 puntos que determinan los contornos faciales, tal como se muestra en la Figura 18.

Aquellos puntos de interés obtenidos, serán los argumentos de entrada de las funciones auxiliares definidas para el cálculo de las excentricidades y a través de las cuales se podrán obtener el ángulo y excentricidad de los objetos bajo estudio.

4.8.3.4 Calcular excentricidades y ángulos

Se define una clase auxiliar en la que se definen los métodos para el cálculo de las propiedades geométricas principales que se han implementado: excentricidad y ángulo:

- calculateEyesEccentricity
- calculateEccentricity
- calculateAngle

```
from scipy.spatial import distance as dist
import math

RAD_TO_DEGREE = 2*90 / math.pi

def calculateEyesEccentricity(p1, p2, p3, p4, p5, p6):

    # horizontal distance
    h = dist.euclidean(p1, p4)

    # calculates eccentricity of eyes using the min distance between p2-p6 and
    # p3-p5
    if h > 0.0:
        # vertical distance
        v1 = dist.euclidean(p2, p6)
        v2 = dist.euclidean(p3, p5)
        v = min(v1, v2)
        ecc = math.sqrt(h*h - v*v)/h
    else:
        ecc = 1.0

    return ecc

def calculateEccentricity(p1, p2, p3, p4):

    # horizontal distance
    h = dist.euclidean(p1, p3)

    # compute eccentricity of ellipse
    if h > 0.0:
```

```

# vertical distance
v = dist.euclidean(p2, p4)
if v > 0.0:
    ecc = math.sqrt(abs(h*h - v*v))/max(h, v)
    if v > h:
        ecc = -1.0 * ecc

    else:
        ecc = 0.0
else:
    ecc = 1.0

return ecc

def calculateAngle(p1, p2):

    angleP1_P2 = RAD_TO_DEGREE * math.atan2( (p2[1]-p1[1]), (p2[0]-p1[0]) )

    return angleP1_P2

```

4.8.3.5 Almacenamiento de métricas (BD)

Para los cálculos de las tasas y métricas temporales será necesario construir estructuras con la suficiente longitud para que permitan obtener los estadísticos de interés. Por tanto, por simplicidad se hará uso de arrays:

```

eccEyesArray = np.zeros((2, nData))
eccMouthrray = np.zeros(nData)
faceAnglerray = np.zeros(nData)
perclosArray = np.zeros(nData)
...
eccEyesArray[0, maxIdx] = leftEyEcc
eccEyesArray[1, maxIdx] = rightEyEcc
eccMouthrray[maxIdx] = mouthEccentricity
faceAnglerray[maxIdx] = angleMouth
perclosArray[maxIdx] = PERCLOS

```

4.8.3.6 Discriminador de somnolencia

La implementación del discriminador de somnolencia, encargado de la activación de la alarma sigue el diseño expuesto anteriormente en la fase de diseño a través de la Figura 36.

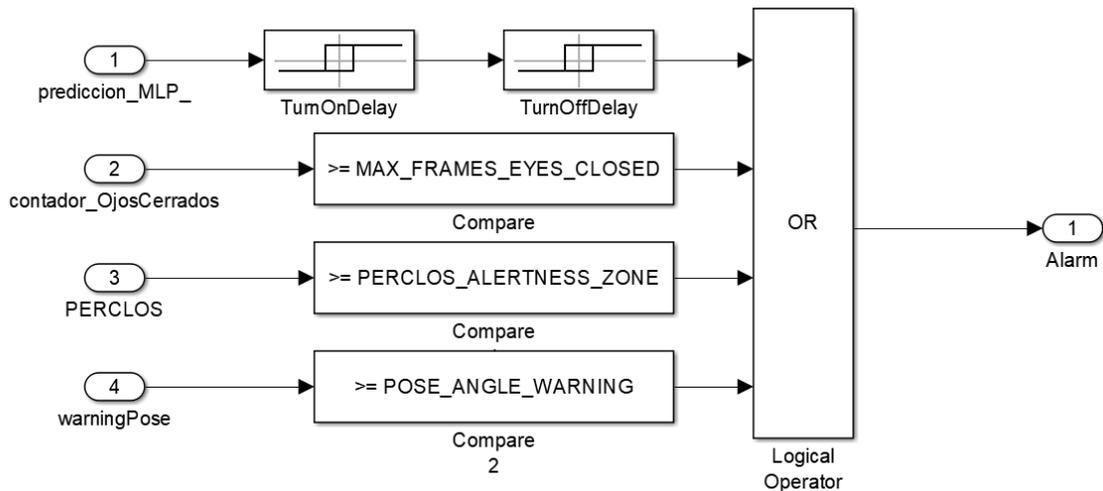


Figura 43. Lógica de activación de alarmas

4.8.3.7 Alarma: Avisador acústico y visual

La gestión de la alarma requiere, como principal consideración, el uso de un segundo hilo de ejecución para no bloquear la ejecución principal al reproducir un sonido. Esto se implementa creando el hilo por primera vez una vez se detecta alarma y no volviéndolo a crear si el anterior no fue finalizado, tal como se muestra en el siguiente trozo de código:

```

if numberOfAlarmsDetected > 0:
    if not T.is_alive():
        T = Thread(target=playAlarm) # create thread
        T.start() # Launch created thread and play alarm
    else:
        T = Thread(target=playAlarm) # create thread
        T.start() # Launch created thread and play alarm
  
```

Adicionalmente el sistema mostrará en todo momento el resultado de la ejecución en curso con los siguientes posibles resultados:

AWAKE. Sujeto en alerta.

DROWSINESS. En este caso además mostrará los motivos por los que se ha disparado la alarma:

- MLP
- PERCLOS
- Closed Eyes
- Pose

En las Figura 44 y Figura 45 se muestran dos ejemplos de la salida que se obtiene cuando se detecta somnolencia y cuando se declara sujeto en alerta respectivamente.

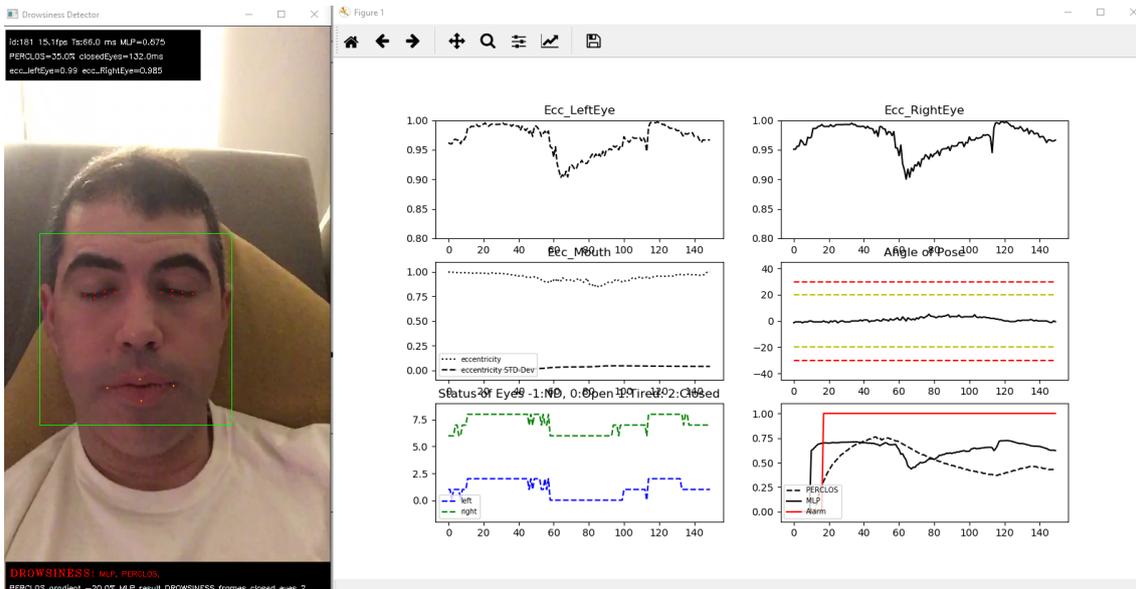


Figura 44. Salida sujeto con somnolencia: "DROWSINESS"

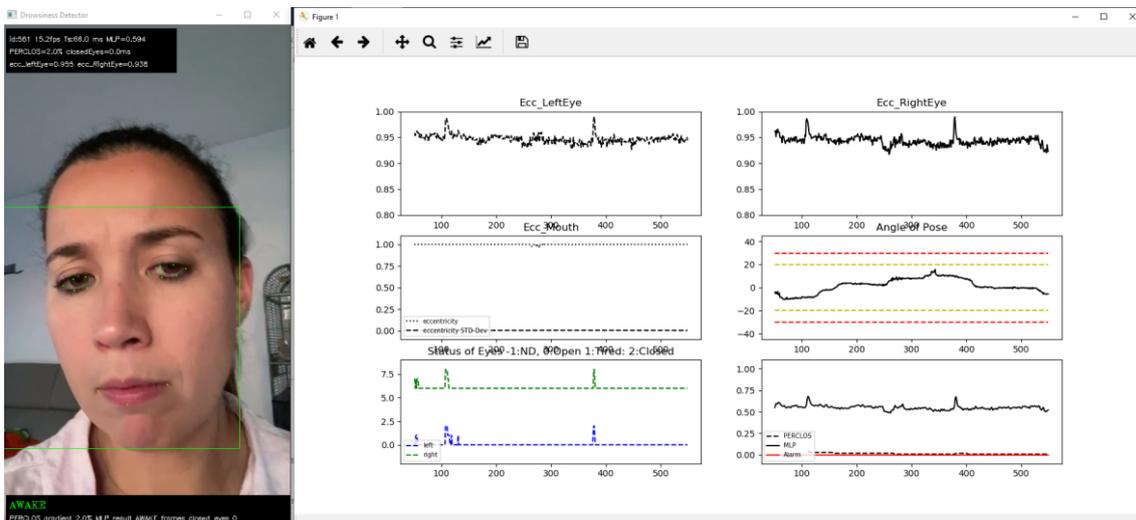
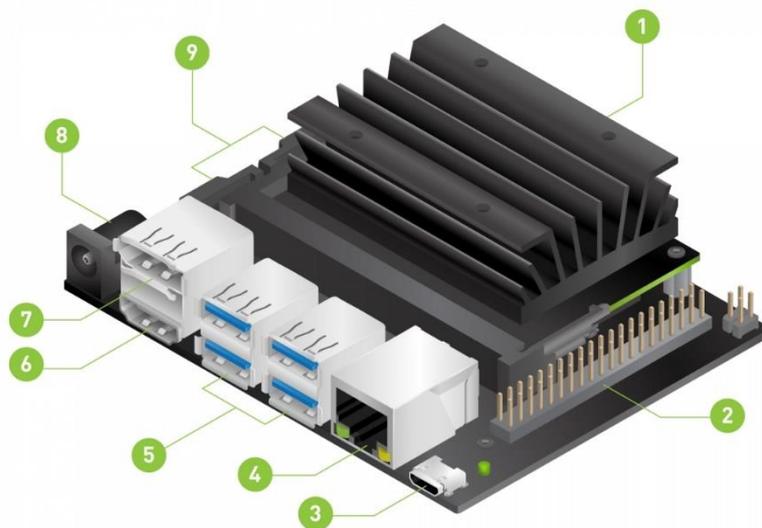


Figura 45. Salida sujeto en alerta: "AWAKE"

4.9 Despliegue en hardware específico

4.9.1 Plataforma hardware

Para desplegar la solución propuesta en un entorno de hardware embebido y dedicado se plantea la plataforma de desarrollo de NVIDIA Jetson Nano Developer Kit.



Con la siguiente configuración E/S:

- | | |
|---|-------------------------------------|
| 1 microSD card slot for main storage | 5 USB 3.0 ports (x4) |
| 2 40-pin expansion header | 6 HDMI output port |
| 3 Micro-USB port for 5V power input, or for Device Mode | 7 DisplayPort connector |
| 4 Gigabit Ethernet port | 8 DC Barrel jack for 5V power input |
| | 9 MIPI CSI-2 camera connectors |

4.9.2 Interfaz de salida

Además, y como interfaz de salida se realiza la integración de un par de luminarias led (rojo y verde) para indicar el estado del proceso en ejecución con la siguiente simbología:

Led verde encendido: la aplicación se encuentra en ejecución.

Led rojo parpadeando: la aplicación ha activado una alarma por somnolencia. El parpadeo se producirá a la velocidad de procesado o tiempo de ciclo.

Adicionalmente, y en caso de activación de alarma, la aplicación emitirá un aviso sonoro mediante la integración de un zumbador.

Por tanto, serán necesarios los siguientes componetes harware:

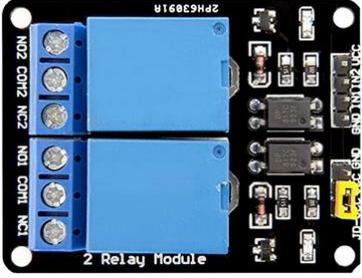
ID	Elemento HW	Captura/Esquema
1	<p>Pareja de relés integrados en la placa de desarrollo AZDelivery 2 Canales Módulo Relé 5V con Optoacoplador Low-Level-Trigger.</p> <p>Se elige esta placa puesto que ya incorpora un fototransistor (DPC-817C¹³) por relé de manera que éstos pueden ser excitados (5V) pesar de que la salida de GPIO de la Jetson Nano sólo ofrece una tensión de 3.3V</p>	
2	<p>Señal de Luz Industrial Columna Torre, BEM-50-2T-D 12-24V DC</p>	
3	<p>Zumbador piezoeléctrico de tono electrónico 85 a 95db DC 3-24V</p>	

Tabla 12. Lista de elementos hardware para interfaz de salida

4.9.2.1 Diagrama de interconexión de elementos

La interconexión de los elementos hardware enumerados en la Tabla 12 deberá realizarse de acuerdo al siguiente esquema gráfico:

¹³ DPC-817C Data sheet: <http://www.datasheet.es/PDF/952198/DPC-817C-pdf.html>

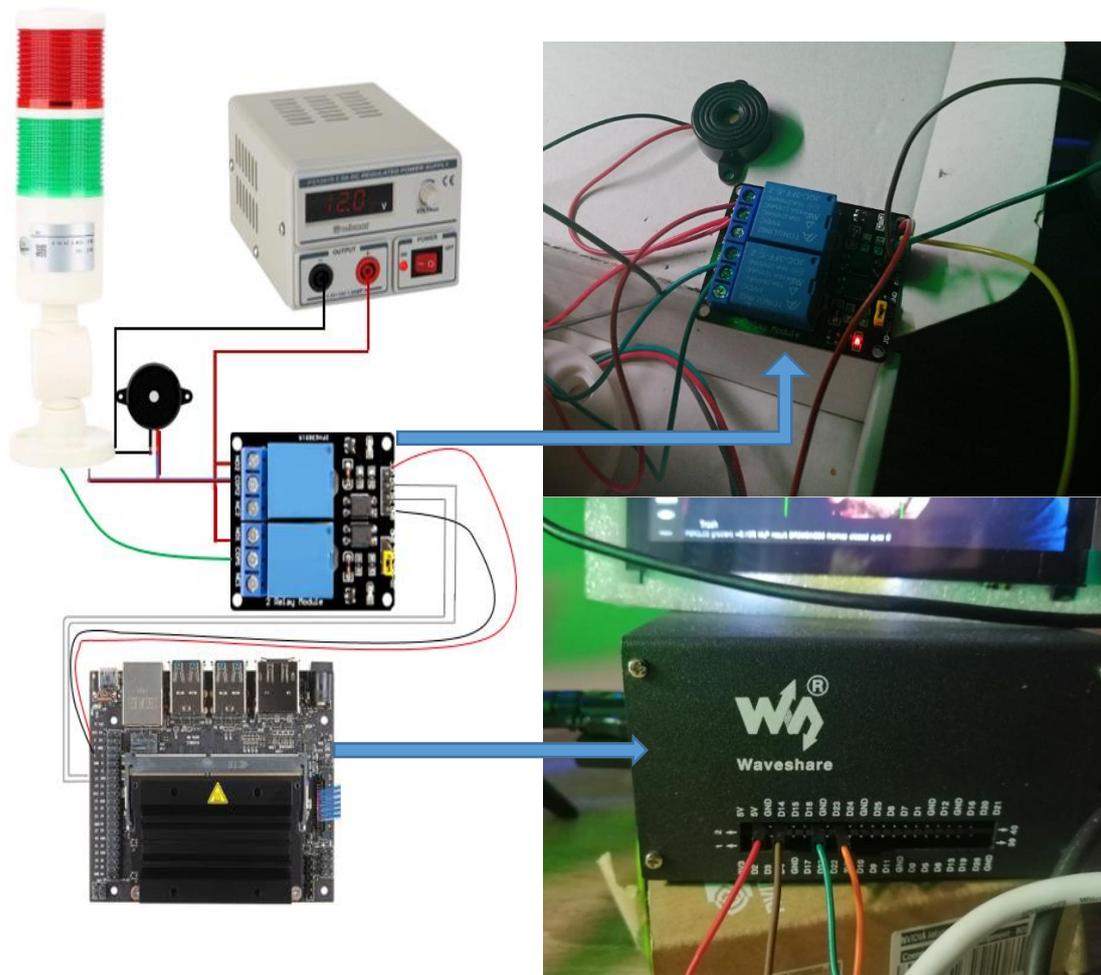


Figura 46. Esquema de interconexión dispositivos hardware y NVIDIA Jetson Nano

Por otro lado, y en base al esquema de pines del GPIO provisto por la plataforma de desarrollo Jetson Nano y adjunto en la Figura 47, se necesitarán dos pines de propósito general para la activación de sendos relés, de manera que se seleccionarán los pines 18 y 23. Por otro lado, la placa que contiene los relés necesita ser alimentada a +5VDC, por lo que usaremos los pines de alimentación provistos en el GPIO con dicha tensión de salida.

Para alimentar el zumbador y el sistema de avisos lumínicos se usará una fuente de alimentación externa regulable *HQ Power PS12015*.

SoC GPIO	Linux GPIO #	Alternate Function	Default Function			Default Function	Alternate Function	Linux GPIO #	SoC GPIO
			3.3 VDC	①	②	5 VDC			
PJ.03	75	GPIO	I2C1_SDA	③	④	5 VDC			
PJ.02	74	GPIO	I2C1_SCL	⑤	⑥	GND			
PBB.00	216	AUD_CLK	GPIO	⑦	⑧	UART1_TXD	GPIO	48	PG.00
			GND	⑨	⑩	UART1_RXD	GPIO	49	PG.01
PG.02	50	UART1_RTS	GPIO	⑪	⑫	GPIO	I2S0_SCLK	79	FJ.07
PB.06	14	SPI1_SCK	GPIO	⑬	⑭	GND			
PY.02	194		GPIO	⑮	⑯	GPIO	SPI1_CS1	232	PDD.00
			3.3 VDC	⑰	⑱	GPIO	SPI1_CS0	15	PB.07
PC.00	16	SPI0_MOSI	GPIO	⑲	⑳	GND			
PC.01	17	SPI0_MISO	GPIO	㉑	㉒	GPIO	SPI1_MISO	13	PB.05
PC.02	18	SPI0_SCK	GPIO	㉓	㉔	GPIO	SPI0_CS0	19	PC.03
			GND	㉕	㉖	GPIO	SPI0_CS1	20	PC.04
PB.05	13	GPIO	I2C0_SDA	㉗	㉘	I2C0_CLK	GPIO	18	PC.02
PS.05	149	CAM_MCLK	GPIO	㉙	㉚	GND			
PZ.00	200	CAM_MCLK	GPIO	㉛	㉜	GPIO	PWM	168	PV.00
PE.06	38	PWM	GPIO	㉝	㉞	GND			
PJ.04	76	I2S0_FS	GPIO	㉟	㊱	GPIO	UART1_CTS	51	PG.03
PB.04	12	SPI1_MOSI	GPIO	㊲	㊳	GPIO	I2S0_DIN	77	PJ.05
			GND	㊴	㊵	GPIO	I2S0_DOUT	78	FJ.06

Figura 47. Esquema de pines GPIO NVIDIA Jetson Nano

Así la configuración final queda tal como se puede apreciar en la siguiente figura:

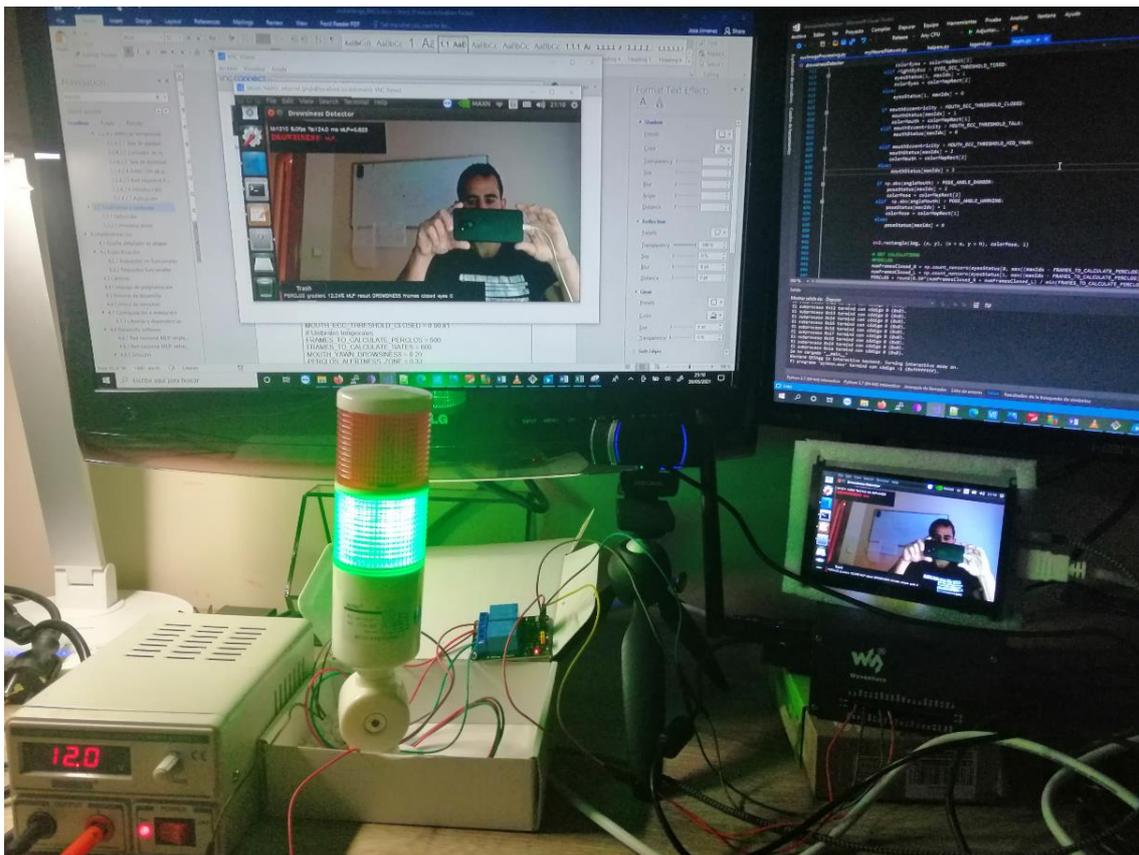


Figura 48. Vista de NVIDIA Jetson Nano con interfaz HW ejecutando la solución propuesta

4.9.3 Configuración de NVIDIA Jetson Nano:

La plataforma de desarrollo está diseñada para ejecutar un sistema operativo Linux de manera que el fabricante suministra una imagen de la distribución Ubuntu 18.04 con un buen número de librerías de visión artificial e inteligencia artificial ya instaladas, lo que facilitará enormemente el trabajo de despliegue de la solución. Por tanto, habrá que seguir los siguientes pasos:

1. Escribir la imagen en la tarjeta microSD tal como se especifica en la [web](#) del fabricante.
2. Insertar la tarjeta en la ranura provista y seguir los pasos que aparecen por pantalla. Al final el proceso de instalación habrá que una instalación fresca de la distribución de Linux Ubuntu 18.04.
3. Todas las librerías empleadas en este proyecto vienen preinstaladas en la imagen instalada, excepto Dlib que deberá ser instalada manualmente.

Para acceder a la plataforma de desarrollo Jetson Nano, sin necesidad de conectar un monitor auxiliar y se ésta se encuentra dentro de la red de trabajo, se podrá establecer una conexión a través de la aplicación de escritorio remotor VNC Viewer, tal como se ilustra a continuación.

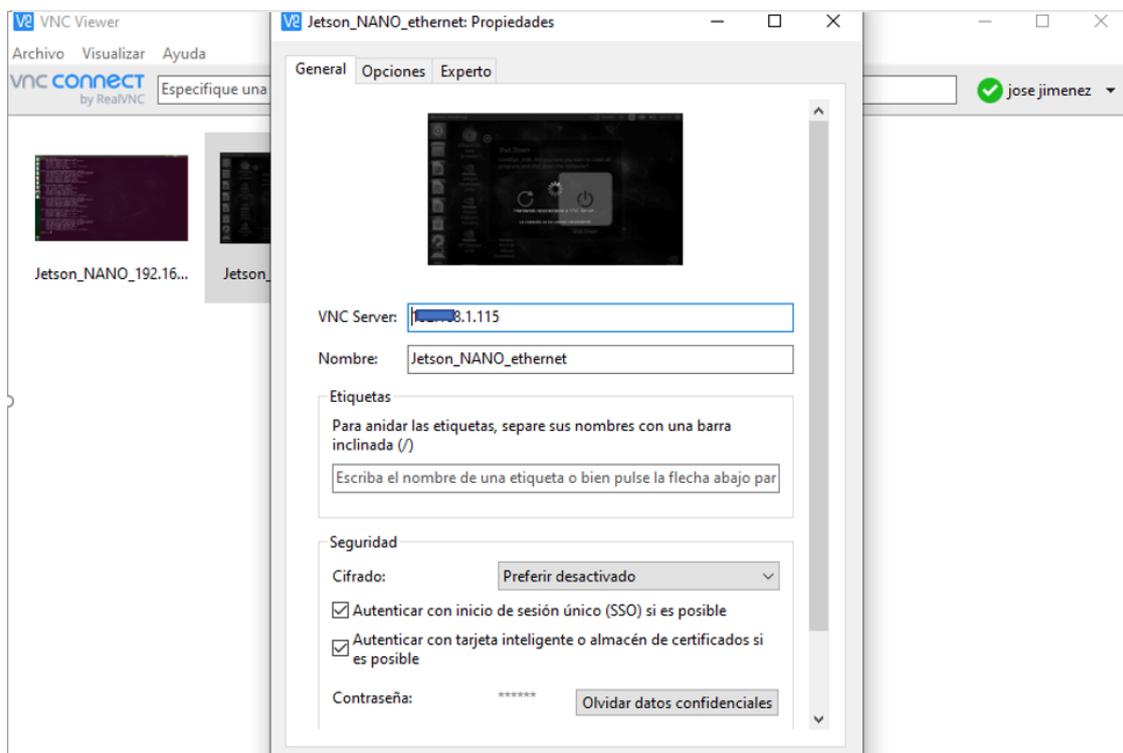


Figura 49. Acceso por escritorio remoto VNC Viewer a Jetson Nano

Una vez se realizada la configuración inicial, habrá que instalar algunas librerías adicionales mediante el comando:

```
pip install <nombreLibrería>
```

Las librerías adicionales que serán necesarias se enumeran en la siguiente tabla:

<nombreLibrería>	Aplicación
numpy	Cálculo numérico y uso de vectores y matrices así como múltiples operaciones matemáticas
math	Funciones matemáticas auxiliares como el arcotangente
scipy.spatial	Cálculo de distancias euclídeas y otras operaciones matemáticas auxiliares
dlib	Librería de IA que imlementa [27] One Millisecond Face Alignment with an Ensemble of Regression Trees
Jetson.GPIO	Permitirá controlar la interfaz GPIO.

Tabla 13. Librerías requeridas y no incluidas en la imagen de instalación de NVIDIA Jetson Nano

El proceso de instalación de las librerías adicionales se realizaría tal como se ilustra en la siguiente imagen para el caso de la instalación de Dlib.

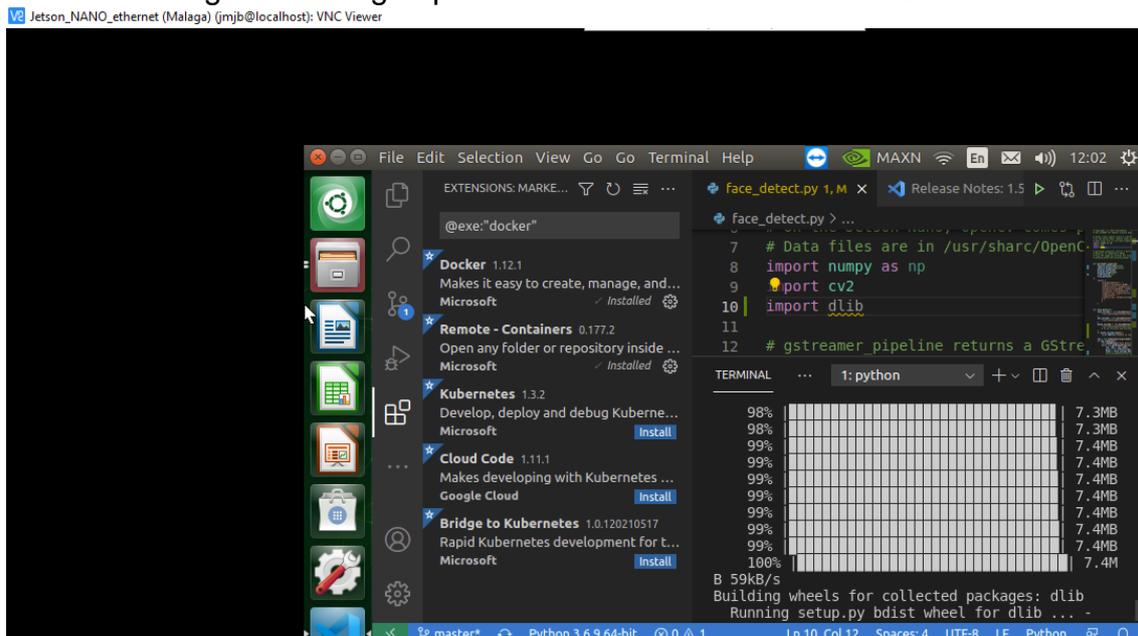


Figura 50. Instalación de dlib en Jetson Nano.

4.9.4 Desarrollo software extra para control de interfaz de salida

Por último será necesario introducir un pequeño desarrollo extra para controlar y activar las salidas que se han conectado como salidas del GPIO. En el presenta trabajo se ha usado la salida 18 para activar la luz verde mientras que la salida 23 será la encargada de activar la luz rojo y el zumbador en caso de producirse una alarma por somnolencia.

```

drowsiness_pin = 23
working_pin = 18
# configuracion de la tarjeta: BCM o BOARD
GPIO.setmode(GPIO.BCM)
# Configurar pines de control
GPIO.setup(drowsiness_pin, GPIO.OUT, initial=GPIO.HIGH)
GPIO.setup(working_pin, GPIO.OUT, initial=GPIO.LOW)

```

```
# valor de conmutación
curr_value = GPIO.LOW
GPIO.output(working_pin, GPIO.LOW)

if bAlarm:
    # conmutar salida de alarma por cada ciclo
    GPIO.output(drowsiness_pin, curr_value)
    # curr_value XOR GPIO.HIGH
    curr_value ^= GPIO.HIGH

else:
    # apagar salida de alarma
    GPIO.output(drowsiness_pin, GPIO.HIGH)
```

5 Verificación y validación

5.1 Resultados

Datos de análisis:

Para validar y medir la efectividad de la solución propuesta se ha realizado el procesado de una serie de grabaciones de diferentes sujetos, en condiciones controladas de manera que cada una de las grabaciones se encuentra clasificada entre resultado de somnolencia y de alerta. De esta manera, se podrán comparar con los resultados que arroja el sistema y además obtener estadísticas de acierto (falsos positivos y falsos negativos). Para este estudio se ha considerando dos grupos de datos:

- A. Análisis de un grupo de sujetos bajo situaciones extremas, que permitirá realizar una validación ante situaciones de clasificación básicas:
 - 1. Situaciones de somnolencia muy marcadas (alto régimen de parpadeos y bostezos) o inclinaciones de cabeza continuadas.
 - 2. Situaciones con alto nivel de alerta y baja somnolencia: la tasa de parpadeos es baja y de corta duración.

- B. Grupo de sujetos en situaciones de somnolencia y alerta menos marcadas y aparentes, incluso para el ojo humano. La muestra extraída para el estudio proviene del dataset preparado por el grupo de investigación *Vision-Learning Mining* de la Universidad de Texas: UTA Real-Life Drowsiness Dataset [32]. Se trata de una muestra más amplia que el primer grupo y permitirá validar el algoritmo en situaciones de clasificación más complejas.

El dataset generado por UTA-RLDD contiene una muestra de 60 sujetos voluntarios, a los que se indicó que tomaran tres videos de sí mismos con su teléfono /cámara web (de cualquier modelo o tipo) en tres estados de somnolencia diferentes con tres vídeos por cada uno según su grado de somnolencia de acuerdo a la Tabla 6. Escala de somnolencia de Karolinska agrupándose los niveles en tres categorías que se usan diferente identificador:

- **ID 0** Alerta: Uno de los tres primeros estados resaltados en la Tabla 6. Se les dijo a los sujetos que estar alerta significaba que estaban completamente conscientes para que pudieran conducir fácilmente durante largas horas.

- **ID 5** Vigilancia baja: Como se indica en los niveles 6 y 7 de la Tabla 6, este estado corresponde a casos sutiles en los que aparecen algunos signos de somnolencia o hay somnolencia pero no se requiere ningún esfuerzo para mantenerse alerta. Si bien los sujetos posiblemente podrían conducir en este estado, se desaconsejaría conducir. Este grupo será descartado en el

presente análisis, dado que se trata de casos frontera entre situaciones de somnolencia y alerta.

- **ID 10** Somnolencia: este estado significa que el sujeto debe intentar activamente no quedarse dormido (nivel 8 y 9 en la Tabla 6).

Por tanto, gráficamente la escala de somnolencia de Karolinska quedaría simplificada en sólo tres categorías tal como se aprecia en la Figura 51a.

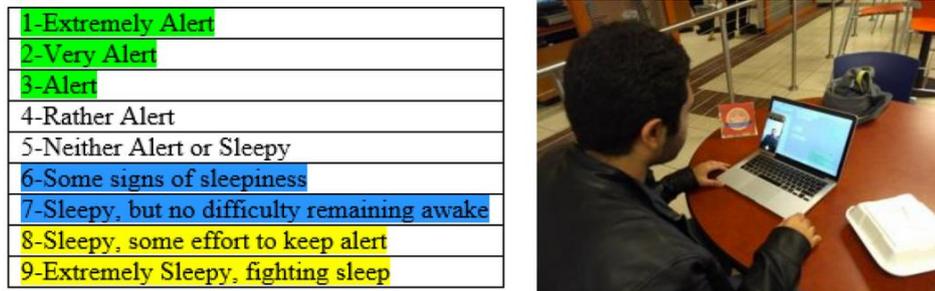


Figura 51. a) Agrupamiento en tres categorías de la escala de somnolencia de Karolinska b) Configuración de cámara recomendada por UTA-RLDD

Además, a los sujetos se les dio un tiempo suficiente (20 días) para producir los tres videos y la libertad de grabar los vídeos en casa o en la universidad, en cualquier momento que se sintieran alerta, poco vigilantes o somnolientos mientras mantenían la cámara configurada (ángulo y distancia) aproximadamente igual. Todos los videos se grabaron en un ángulo tal que ambos ojos eran visibles y la cámara se colocó a un brazo de distancia del sujeto. Estas instrucciones se utilizaron para hacer que los videos sean similares a los videos que se obtendrían en un automóvil, tal como se parecía en la Figura 51b

Por tanto, y una vez quedan definidas ambas muestras objeto de estudio y análisis, tal como se observa en la siguiente figura, se presentarán los resultados:

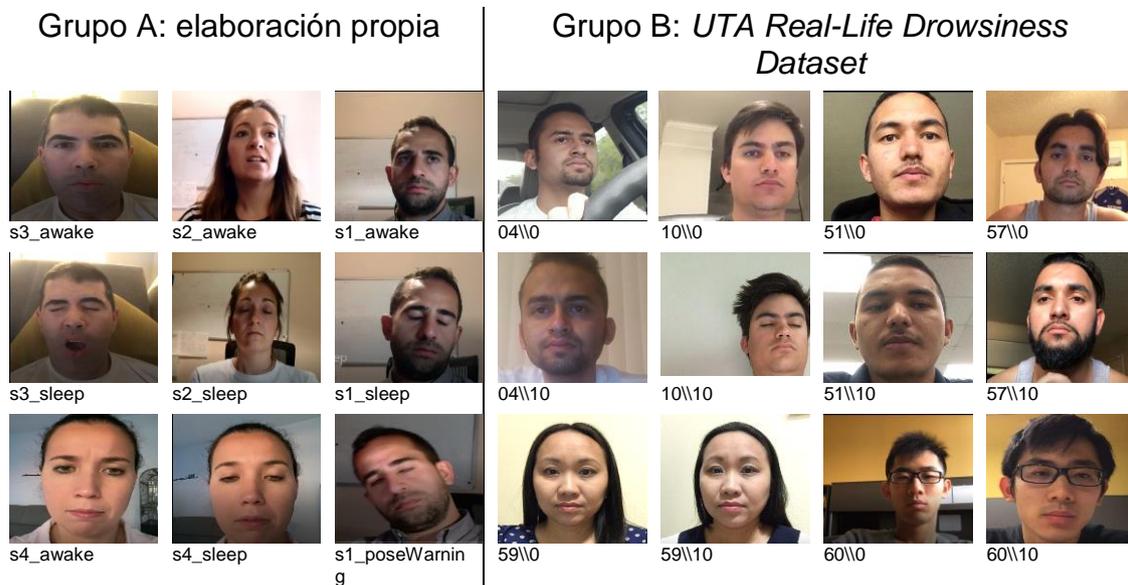


Figura 52. Muestra de sujetos bajo estudio: grupo A y grupo B

5.1.1 Configuración de pruebas

5.1.1.1 Parámetros de pruebas

Los parámetros sobre los que se establecerán dos configuraciones diferentes son:

- **thresholdMLP**: parámetro umbral a partir del cual se determina si la salida de la función de la activación pertenece a la clase “somnolencia” (mayor a thresholdMLP) o “despierto” (menor igual a thresholdMLP).
- **thresholdTurnOnDelayMLP**: número de frames consecutivos en los que la red MLP debe arrojar como resultado la clase “somnolencia” para activar alarma. En tiempo, equivale a thresholdTurnOnDelayMLP/FPS donde FPS es habitualmente 30 capturas por segundo.
- **thresholdTurnOffDelayMLP**: número de frames consecutivos en los que la red MLP debe arrojar como resultado la clase “despierto” para desactivar la alarma (si está activa). En tiempo, equivale a thresholdTurnOnDelayMLP/FPS donde FPS es habitualmente 30 capturas por segundo.

5.1.1.2 Datos de prueba

Para la realización de las pruebas se ha definido un fichero que contendrá la lista de vídeos a procesar:

sujeto ID	somnolencia	fichero
S1	NO	drowsiness_dataSet\\S1_awake.mp4
S1	SI	drowsiness_dataSet\\S1_sleep.mp4
S2	NO	drowsiness_dataSet\\S2_awake.mp4
S2	SI	drowsiness_dataSet\\S2_sleep.mp4
S3	NO	drowsiness_dataSet\\S3_awake.mp4
S3	SI	drowsiness_dataSet\\S3_sleep.mp4
S4	NO	drowsiness_dataSet\\S4_awake.mp4
S4	SI	drowsiness_dataSet\\S4_sleep.mp4
S1	SI	drowsiness_dataSet\\S1_poseWarning.mp4

Tabla 14. Lista de ficheros para pruebas - grupo A

sujeto ID	somnolencia	fichero
04	NO	drowsiness_dataSet\\UTA-RLDD\\04\\0.mp4

04	SI	drowsiness_dataSet\\UTA-RLDD\\04\\10.mp4
10	NO	drowsiness_dataSet\\UTA-RLDD\\10\\0.MOV
10	SI	drowsiness_dataSet\\UTA-RLDD\\10\\10.MOV
51	NO	drowsiness_dataSet\\UTA-RLDD\\51\\0.MOV
51	SI	drowsiness_dataSet\\UTA-RLDD\\51\\10.MOV
57	NO	drowsiness_dataSet\\UTA-RLDD\\57\\0.MOV
57	SI	drowsiness_dataSet\\UTA-RLDD\\57\\10.MOV
59	NO	drowsiness_dataSet\\UTA-RLDD\\59\\0.MOV
59	SI	drowsiness_dataSet\\UTA-RLDD\\59\\10.MOV
60	NO	drowsiness_dataSet\\UTA-RLDD\\60\\0.MOV
60	SI	drowsiness_dataSet\\UTA-RLDD\\60\\10.MOV

Tabla 15. Lista de ficheros para pruebas - grupo B

Debido a la cantidad de parámetros de configuración existentes, y su influencia en los posibles resultados, se han realizado dos baterías de pruebas con diferente configuraciones del sistema, al objeto de comparar el comportamiento de éste.

Para realizar un análisis estadístico de los resultados por cada sesión se almacenarán los siguientes datos:

- Número de frames procesados
- Número de frames en los que se disparó alarma por somnolencia
- Número de veces que la alarma es disparada por la red MLP
- Número de veces que la alarma es disparada por tasa de PERCLOS elevada
- Número de veces que la alarma es disparada por ojos cerrados de forma consecutiva
- Número de veces que la alarma es disparada por una tasa de inclinación de cabeza (pose) alejada de la vertical.

5.1.2 Sesión 1: configuración con alta sensibilidad

La primera configuración, podría ser calificada de alta sensibilidad a las detecciones, lo que permitirá detectar un mayor número de casos de somnolencia. Como contrapartida, se obtendrá un mayor número de falsos positivos, es decir: el sistema indicará erróneamente situaciones de somnolencia para determinadas escenas que no lo son.

thresholdMLP	0.625
thresholdTurnOnDelayMLP	7
thresholdTurnOffDelayMLP	60

5.1.2.1 Grupo A

Los resultados en bruto que se obtienen son:

Sujeto bajo análisis	Somnolencia	Framas	Alarmas	por MLP	por Ojos cerrados	por PERCLOS	por Pose	Falsos negativos	Falsos positivos
S1_sleep.mp4	Y	1825	1794	1794	613	1751	0	31	0
S1_awake.mp4	N	1813	0	0	0	0	0	0	0
S2_sleep.mp4	Y	1814	1567	1567	515	871	0	247	0
S2_awake.mp4	N	1818	5	0	0	5	0	0	5
S3_sleep.mp4	Y	1561	1545	1523	386	1463	0	16	0
S3_awake.mp4	N	1562	0	0	0	0	0	0	0
S4_sleep.mp4	Y	2008	1925	1925	100	92	0	83	0
S4_awake.mp4	N	1496	0	0	0	0	0	0	0
S1_poseWarning.mp4	Y	1241	1170	603	38	0	1024	71	0
TOTAL (somnolencia)	Y	8449	8001	7412	1652	4177	1024	448	0
TOTAL (despierto)	N	6689	5	0	0	5	0	0	5

Tabla 16. Resultados en bruto. Sesión 1 – Grupo A

El análisis estadístico que se obtiene analizando los datos en bruto arroja la tasa de acierto del sistema:

- Caso ideal si somnolencia (Y): Alarmas = 100%
- Caso ideal si no hay somnolencia (N): Alarmas = 0%

Por tanto la columna [Alarmas] determina el resultado global de la batería de pruebas:

Sujeto bajo análisis	Somnolencia	Alarmas	por MLP	pos Ojos cerrados	por PERCLOS	Pose	Falsos negativos	Falsos positivos
S1_sleep.mp4	Y	98,30%	98,30%	33,59%	95,95%	0,00%	1,70%	0,00%
S1_awake.mp4	N	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
S2_sleep.mp4	Y	86,38%	86,38%	28,39%	48,02%	0,00%	13,62%	0,00%
S2_awake.mp4	N	0,28%	0,00%	0,00%	0,28%	0,00%	0,00%	0,28%
S3_sleep.mp4	Y	98,98%	97,57%	24,73%	93,72%	0,00%	1,02%	0,00%
S3_awake.mp4	N	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
S4_sleep.mp4	Y	95,87%	95,87%	4,98%	4,58%	0,00%	4,13%	0,00%
S4_awake.mp4	N	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
S1_poseWarning.mp4	Y	94,28%	48,59%	3,06%	0,00%	82,51%	5,72%	0,00%
TOTAL [somnolencia]	Y	94,70%	87,73%	19,55%	49,44%	12,12%	5,30%	0,00%
TOTAL [despierto]	N	0,07%	0,00%	0,00%	0,07%	0,00%	0,00%	0,07%

Tabla 17. Resultados estadísticos. Sesión 1 – Grupo A

Para un mejor análisis de los resultados, se agrupan los casos de somnolencia:

Sujeto	Alarmas [%]	MLP [%]	PERCLOS [%]	Ojos cerrados [%]	Pose [%]
S1_poseWarning.mp4	94,28%	48,59%	0,00%	3,06%	87,19%
S1_sleep.mp4	98,30%	98,30%	95,95%	33,59%	0,00%
S2_sleep.mp4	86,38%	86,38%	48,02%	28,39%	0,00%
S3_sleep.mp4	98,98%	97,57%	93,72%	24,73%	0,00%

S4_sleep.mp4	95,87%	95,87%	4,58%	4,98%	0,00%
TOTAL [somnolencia]	94,70%	87,73%	49,44%	19,55%	12,81%

Tabla 18. Resultados estadísticos caso somnolencia. Sesión 1 – Grupo A
Por último, los resultados obtenidos se representan gráficamente:

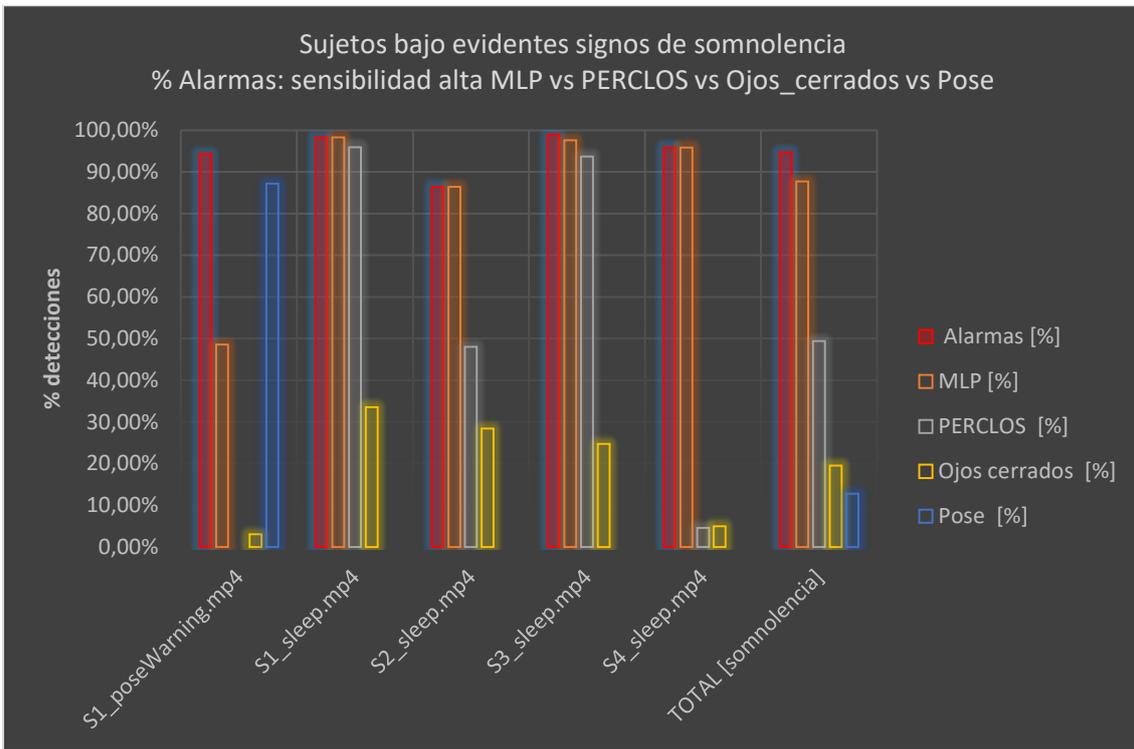


Figura 53. Gráfica resultados en caso somnolencia. Sesión 1 – Grupo A

De la misma forma, se realiza el agrupamiento de los casos donde no existe somnolencia así como su representación gráfica:

Sujeto	Alarmas [%]	MLP [%]	PERCLOS [%]	Ojos cerrados [%]	Pose [%]
S1_awake.mp4	0,00%	0,00%	0,00%	0,00%	0,00%
S2_awake.mp4	0,28%	0,00%	0,28%	0,00%	0,00%
S3_awake.mp4	0,00%	0,00%	0,00%	0,00%	0,00%
S4_awake.mp4	0,00%	0,00%	0,00%	0,00%	0,00%
TOTAL [despierto]	0,07%	0,00%	0,07%	0,00%	0,00%

Tabla 19. Estadísticas en caso alerta. Sesión 1 – Grupo A

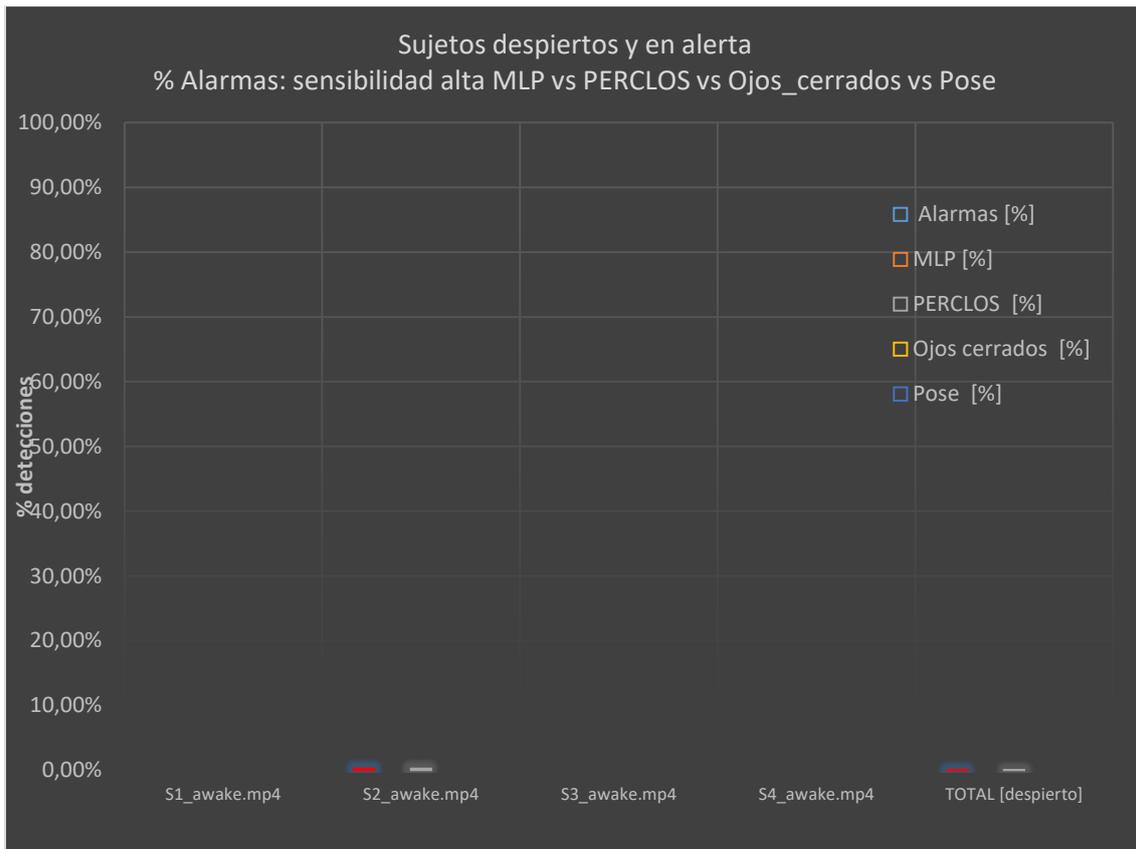


Figura 54. Gráfica estadísticas caso alerta. Sesión 1 – Grupo A

El grupo A arroja unos resultados altamente satisfactorios, ya que el sistema arroja una **tasa de acierto del 94,70%** para **casos de somnolencia**, con las siguientes conclusiones parciales:

- Valorando los resultados globales se observa que la predicción de la red neuronal (MLP) es la que genera una mayor tasa de aciertos (87,73%) frente al 49,44% que arroja el sistema de alarma por umbral de PERCLOS. En tercera posición aparece la tasa de ojos cerrados de forma consecutiva con un 19,55%.
- No obstante hay un matiz relevante: en dos de los cinco casos bajo estudio, la tasa de alarma por MLP y PERCLOS son bastante similares mientras que en los otros tres casos MLP arroja mejores resultados que PERCLOS, S1 (caso de parpadeos y caso de inclinación de cabeza) y S3. Estos resultados parecen consecuencia de que los sujetos S1 y S3 muestran signos más evidentes de fatiga. Curiosamente, y sin que se puedan extraer conclusiones sobre este hecho, los sujetos S1 y S3 son varones mientras que S2 y S4 son mujeres. Nótese que la red neuronal fue entrenada con datos obtenidos de un varón lo que podría suponer un indicio que debería ser estudiado en futuras reimplementaciones.
- El caso de especial de S1, donde el sujeto muestra signos de somnolencia por inclinación de la cabeza sobre la vertical, tal como era previsible arroja la mayor tasa de aciertos por la tasa de pose (87,19%). No obstante la red MLP consigue un 48,59% mientras que la tasa de

PERCLOS no detecta ningún caso (0%). Finalmente, la tasa de ojos cerrados arroja un residual 3,06%. Sobre este hecho conviene señalar que la red neuronal no ha sido diseñada para considerar métricas de inclinación de la cabeza, por lo que podría deducirse que en una nueva versión que sí lo evaluara los resultados serían mejores.

Respecto a los casos de **alerta o usencia de somnolencia** se obtiene una tasa global de **falsas detecciones del 0.07%**:

- El sujeto 2 (S2) es el único que arroja falsas detecciones, si bien el valor es residual (0.28%) sobre el total. Extrapolando estas falsas detecciones al total de muestras analizadas, la tasa de falsas detecciones se situaría en un valor despreciable del 0.07%. En este caso, las falsas detecciones provienen exclusivamente de la tasa de PERCLOS, lo que podría explicarse por el hecho de que el sujeto 2 realizó la grabación mientras leía un texto sobre una pantalla, lo que podría haber provocado un mayor número de parpadeos.

5.1.2.2 Grupo B

Los resultados en bruto que se obtienen para el segundo grupo, muestra de la Universidad de Texas, son:

Sujeto bajo análisis	Somnolencia	Frames	Alarmas	por MLP	por Ojos cerrados	por PERCLOS	por Pose	Falsos negativos	Falsos positivos
04\\0.mp4	N	13065	6648	6648	34	0	0	0	6648
04\\10.mp4	Y	8275	1552	1552	1	0	0	6723	0
10\\0.MOV	N	16309	5864	5864	17	0	0	0	5864
10\\10.MOV	Y	11125	8902	8842	156	455	0	2223	0
51\\0.MOV	N	10545	3661	3648	429	886	0	0	3661
51\\10.MOV	Y	11629	11569	11569	917	3333	0	60	0
57\\0.MOV	N	1525	881	881	54	0	0	0	881
57\\10.MOV	Y	11017	11001	11001	3503	8219	0	16	0
59\\0.MOV	N	13519	2125	2125	0	0	0	0	2125
59\\10.MOV	Y	9189	7458	7458	37	0	0	1731	0
60\\0.MOV	N	3337	261	261	0	0	0	0	261
60\\10.MOV	Y	11239	4891	4891	2	0	0	6348	0
TOTAL (somnolencia)		42046	33024	32964	4577	12007	0	9022	0
TOTAL (despierto)		58300	19440	19427	534	886	0	0	19179

Tabla 20. Datos en bruto. Sesión 1 – grupo B

El análisis estadístico que se obtiene analizando los datos en bruto arroja la tasa de acierto del sistema:

- Caso ideal si somnolencia (Y): Alarmas = 100%
- Caso idea si no hay somnolencia (N): Alarmas = 0%

Por tanto la columna [Alarmas] determina el resultado global de la batería de pruebas:

Sujeto bajo análisis	Somnolencia	Alarmas	por MLP	pos Ojos cerrados	por PERCLOS	Pose	Falsos negativos	Falsos positivos
04\\0.mp4	N	50,88%	50,88%	0,26%	0,00%	0,00%	0,00%	50,88%
04\\10.mp4	Y	18,76%	18,76%	0,01%	0,00%	0,00%	81,24%	0,00%
10\\0.MOV	N	35,96%	35,96%	0,10%	0,00%	0,00%	0,00%	35,96%
10\\10.MOV	Y	80,02%	79,48%	1,40%	4,09%	0,00%	19,98%	0,00%
51\\0.MOV	N	34,72%	34,59%	4,07%	8,40%	0,00%	0,00%	34,72%
51\\10.MOV	Y	99,48%	99,48%	7,89%	28,66%	0,00%	0,52%	0,00%
57\\0.MOV	N	57,77%	57,77%	3,54%	0,00%	0,00%	0,00%	57,77%
57\\10.MOV	Y	99,85%	99,85%	31,80%	74,60%	0,00%	0,15%	0,00%
59\\0.MOV	N	15,72%	15,72%	0,00%	0,00%	0,00%	0,00%	15,72%
59\\10.MOV	Y	81,16%	81,16%	0,40%	0,00%	0,00%	18,84%	0,00%
60\\0.MOV	N	7,82%	7,82%	0,00%	0,00%	0,00%	0,00%	7,82%
60\\10.MOV	Y	43,52%	43,52%	0,02%	0,00%	0,00%	56,48%	0,00%
TOTAL [somnolencia]	Y	78,54%	78,40%	10,89%	28,56%	0,00%	21,46%	0,00%
TOTAL [despierto]	N	33,34%	33,32%	0,92%	1,52%	0,00%	0,00%	32,90%

Tabla 21. Estadísticas. Sesión 1 – grupo B

Para un mejor análisis de los resultados, se agrupan los casos de somnolencia:

Sujeto	Alarmas [%]	MLP [%]	PERCLOS [%]	Ojos cerrados [%]	Pose [%]
04\\10.mp4	18,76%	18,76%	0,00%	0,01%	0,00%
10\\10.MOV	80,02%	79,48%	4,09%	1,40%	0,00%
51\\10.MOV	99,48%	99,48%	28,66%	7,89%	0,00%
57\\10.MOV	99,85%	99,85%	74,60%	31,80%	0,00%
59\\10.MOV	81,16%	81,16%	0,00%	0,40%	0,00%
60\\10.MOV	43,52%	43,52%	0,00%	0,02%	0,00%
TOTAL [somnolencia]	78,54%	78,40%	28,56%	10,89%	0,00%

Tabla 22. Estadísticas caso somnolencia. Sesión 1 – grupo B

Por último, los resultados obtenidos se representan gráficamente:

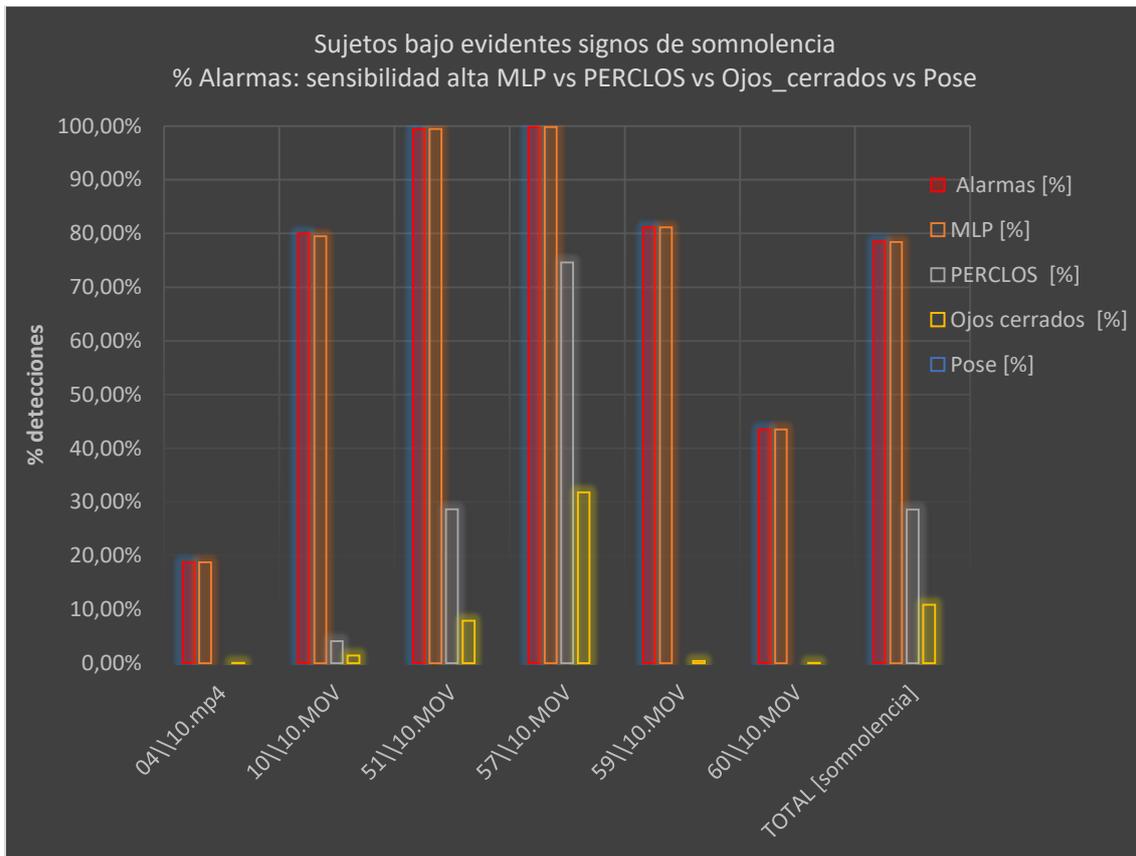


Figura 55. Gráfica de estadísticas caso somnolencia. Sesión 1 – grupo B

De la misma forma, se realiza el agrupamiento de los casos donde no existe somnolencia así como su representación gráfica:

Sujeto	Alarmas [%]	MLP [%]	PERCLOS [%]	Ojos cerrados [%]	Pose [%]
04\\0.mp4	50,88%	50,88%	0,00%	0,26%	0,00%
10\\0.MOV	35,96%	35,96%	0,00%	0,10%	0,00%
51\\0.MOV	34,72%	34,59%	8,40%	4,07%	0,00%
57\\0.MOV	57,77%	57,77%	0,00%	3,54%	0,00%
59\\0.MOV	15,72%	15,72%	0,00%	0,00%	0,00%
60\\0.MOV	7,82%	7,82%	0,00%	0,00%	0,00%
TOTAL [despierto]	33,34%	33,32%	1,52%	0,92%	0,00%

Tabla 23 Estadísticas caso alerta. Sesión 1 – grupo B

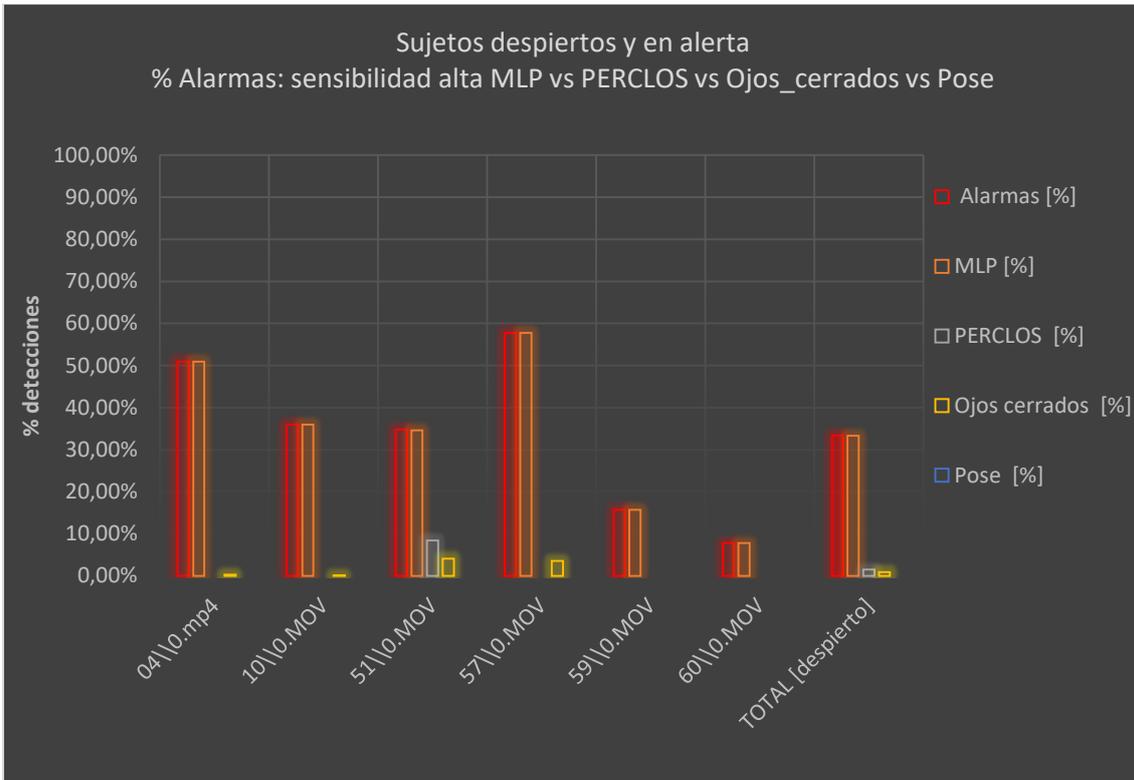


Figura 56. Gráfica Estadísticas caso alerta. Sesión 1 – grupo B

El grupo B arroja unos resultados peores que el grupo A, ya que los sujetos de bajo estudio ofrecen métricas menos evidentes de síntomas de somnolencia lo que resulta en una tasa del **78.54% para detecciones verdaderas** y del **33.34% de falsas detecciones**, cuando el usuario declaró no encontrarse bajo síntomas de somnolencia:

- Valorando los resultados globales se observa que la predicción de la red neuronal (MLP) nuevamente genera la mayor tasa de aciertos global (78.04%), seguida por la tasa de PERCLOS, con un 28.56%, si bien sólo resulta relevante en un sujeto (57) donde arrojó un 74.06% mientras que para el resto de sujetos bajo estudio no supera el 29% (con un 0% para los sujetos 04, 59, 60).
- El caso de especial de S1, donde el sujeto muestra signos de somnolencia por inclinación de la cabeza sobre la vertical, tal como era previsible arroja la mayor tasa de aciertos por la tasa de pose (87,19%) y en este caso la la red MLP reduce considerablemente su desempeño: 9,11% frente al 48,59% conseguido con la configuración de alta sensibilidad.

Respecto a los casos de **alerta o usencia de somnolencia** se obtiene una tasa global de **falsas detecciones del 33.34%**, lo cual sería un nivel excesivamente alto. Nótese que el grueso de las falsas detecciones se origina por la red MLP (33.32%).

5.1.3 Sesión 2: configuración con sensibilidad media

En este caso, el sistema se configura de forma que resulta menos sensible a situaciones de somnolencia, lo cual significa que los estímulos han de tener un mayor énfasis y/o duración para que el sistema acabe disparando la alarma.

thresholdMLP	0.690
thresholdTurnOnDelayMLP	9
thresholdTurnOffDelayMLP	90

5.1.3.1 Grupo A

Los resultados en bruto que se obtienen son:

Sujeto bajo análisis	Somnolencia	Frames	Alarmas	por MLP	por Ojos cerrados	por PERCLOS	por Pose	Falsos negativos	Falsos positivos
S1_sleep.mp4	Y	1825	1760	940	619	1751	0	65	0
S1_awake.mp4	N	1813	0	0	0	0	0	0	0
S2_sleep.mp4	Y	1814	1647	1539	515	869	0	167	0
S2_awake.mp4	N	1818	5	0	0	5	0	0	5
S3_sleep.mp4	Y	1561	1545	1382	387	1464	0	16	0
S3_awake.mp4	N	1562	0	0	0	0	0	0	0
S4_sleep.mp4	Y	2008	1359	1352	100	74	0	649	0
S4_awake.mp4	N	1496	0	0	0	0	0	0	0
S1_poseWarning.mp4	Y	1241	1024	113	38	0	1024	217	0
TOTAL (somnolencia)	Y	8449	7335	5326	1659	4158	1024	1114	0
TOTAL (despierto)	N	6689	5	0	0	5	0	0	5

Tabla 24. Datos en bruto. Sesión 2 – grupo A

El análisis estadístico que se obtiene analizando los datos en bruto arroja la tasa de acierto del sistema:

- Caso ideal si somnolencia (Y): Alarmas = 100%
- Caso ideal si no hay somnolencia (N): Alarmas = 0%

Por tanto la columna [Alarmas] determina el resultado global de la batería de pruebas:

Sujeto bajo análisis	Somnolencia	Alarmas	por MLP	por Ojos cerrados	por PERCLOS	Pose	Falsos negativos	Falsos positivos
S1_sleep.mp4	Y	96,44%	51,51%	33,92%	95,95%	0,00%	3,56%	0,00%
S1_awake.mp4	N	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
S2_sleep.mp4	Y	90,79%	84,84%	28,39%	47,91%	0,00%	9,21%	0,00%
S2_awake.mp4	N	0,28%	0,00%	0,00%	0,28%	0,00%	0,00%	0,28%

S3_sleep.mp4	Y	98,98%	88,53%	24,79%	93,79%	0,00%	1,02%	0,00%
S3_awake.mp4	N	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
S4_sleep.mp4	Y	67,68%	67,33%	4,98%	3,69%	0,00%	32,32%	0,00%
S4_awake.mp4	N	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
S1_poseWarnin g.mp4	Y	82,51%	9,11%	3,06%	0,00%	82,51%	17,49%	0,00%
TOTAL [somnolencia]	Y	86,82%	63,04%	19,64%	49,21%	12,12%	13,18%	0,00%
TOTAL [despierto]	N	0,07%	0,00%	0,00%	0,07%	0,00%	0,00%	0,07%

Tabla 25. Estadísticas totales. Sesión 2 – grupo A

Para un mejor análisis de los resultados, se agrupan los casos de somnolencia:

Sujeto	Alarmas [%]	MLP [%]	PERCLOS [%]	Ojos cerrados [%]	Pose [%]
S1_poseWarning.mp4	82,51%	9,11%	0,00%	3,06%	87,19%
S1_sleep.mp4	96,44%	51,51%	95,95%	33,92%	0,00%
S2_sleep.mp4	90,79%	84,84%	47,91%	28,39%	0,00%
S3_sleep.mp4	98,98%	88,53%	93,79%	24,79%	0,00%
S4_sleep.mp4	67,68%	67,33%	3,69%	4,98%	0,00%
TOTAL [somnolencia]	86,82%	63,04%	49,21%	19,64%	12,81%

Tabla 26. Estadísticas caso somnolencia. Sesión 2 – grupo A

Por último, los resultados obtenidos se representan gráficamente:

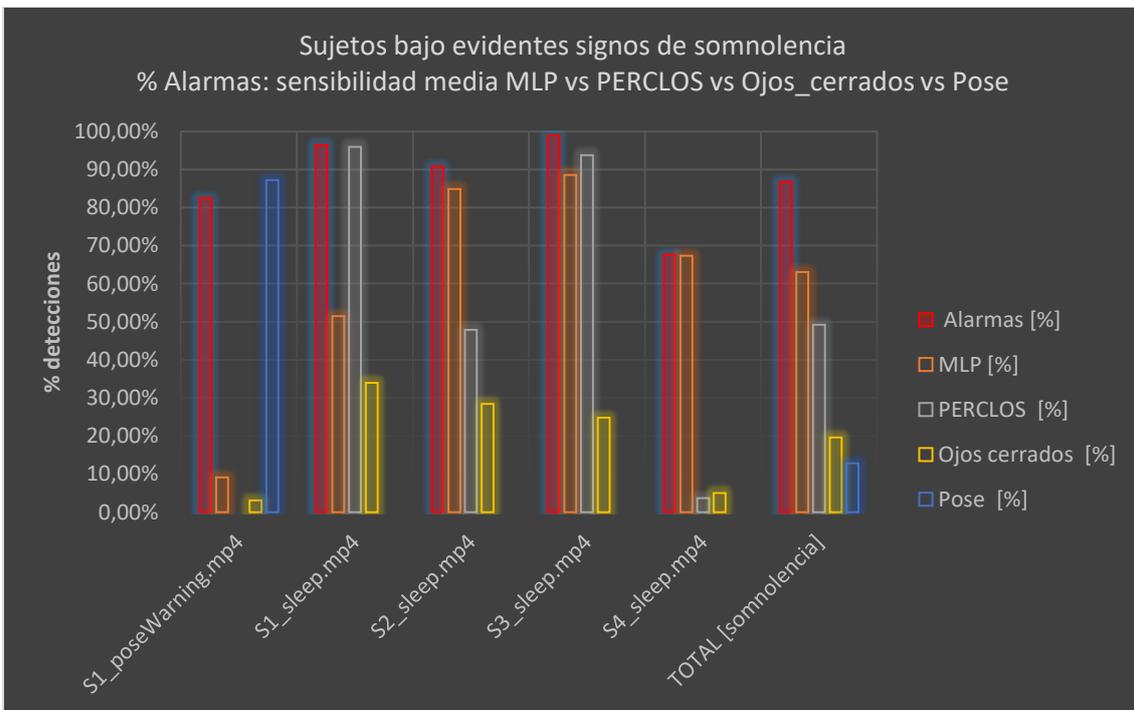


Figura 57. Estadísticas caso somnolencia. Sesión 2 – grupo A

De la misma forma, se realiza el agrupamiento de los casos donde no existe somnolencia así como su representación gráfica:

Sujeto	Alarmas [%]	MLP [%]	PERCLOS [%]	Ojos cerrados [%]	Pose [%]
S1_awake.mp4	0,00%	0,00%	0,00%	0,00%	0,00%

S2_awake.mp4	0,28%	0,00%	0,28%	0,00%	0,00%
S3_awake.mp4	0,00%	0,00%	0,00%	0,00%	0,00%
S4_awake.mp4	0,00%	0,00%	0,00%	0,00%	0,00%
TOTAL [despierto]	0,07%	0,00%	0,07%	0,00%	0,00%

Tabla 27. Estadísticas caso alerta. Sesión 2 – grupo A

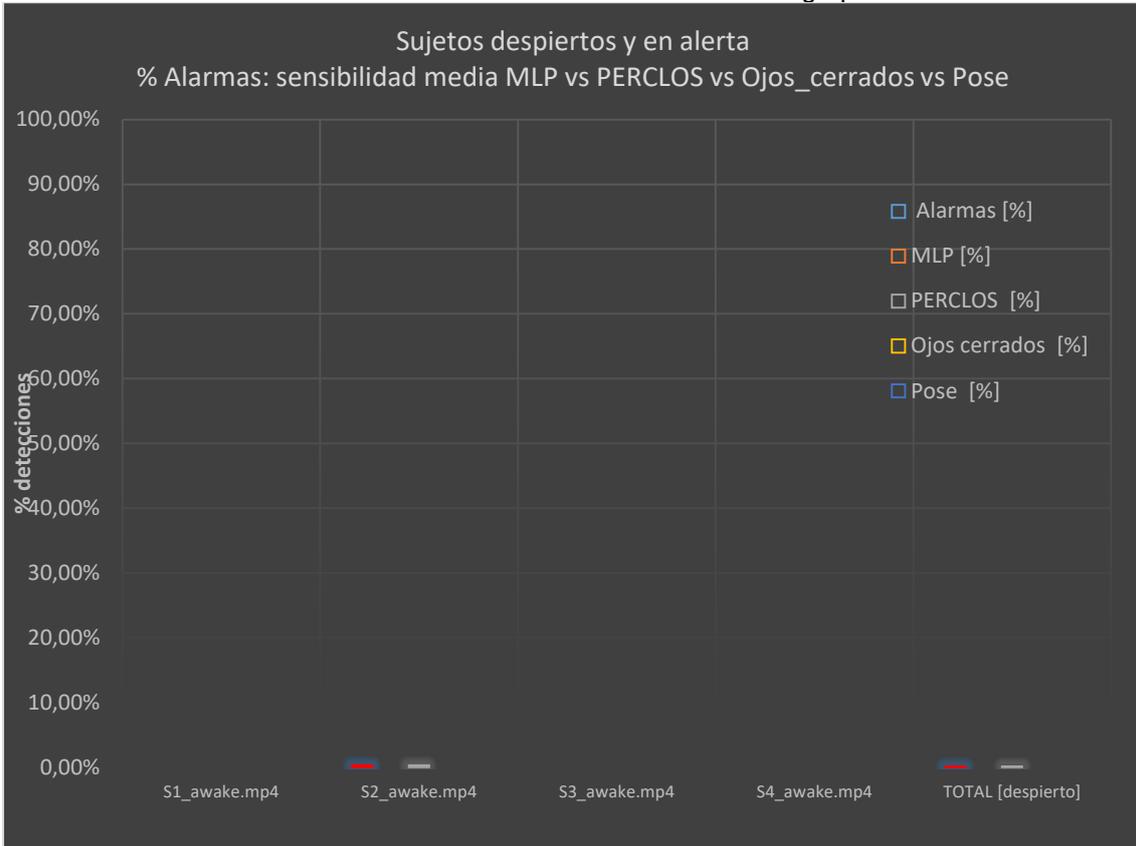


Figura 58. Gráfica de estadísticas caso alerta. Sesión 2 – grupo A

El grupo A arroja unos resultados nuevamente satisfactorios, si bien resultan peores que en el caso de configuración con alta sensibilidad (94.70%) ya que el sistema arroja una **tasa de acierto del 86,82%** para **casos de somnolencia**, con las siguientes conclusiones parciales:

- Valorando los resultados globales se observa que la predicción de la red neuronal (MLP) sigue generando la mayor tasa de aciertos global (63,04%), si bien ya no es la que siempre genera una mayor tasa de aciertos: S1 y S3 ofrecen mejores resultados por la tasa de PERCLOS (51,51% vs. 95.95% y 88.53% vs. 93.79% respectivamente).
- El caso de especial de S1, donde el sujeto muestra signos de somnolencia por inclinación de la cabeza sobre la vertical, tal como era previsible arroja la mayor tasa de aciertos por la tasa de pose (87,19%) y en este caso la red MLP reduce considerablemente su desempeño: 9,11% frente al 48,59% conseguido con la configuración de alta sensibilidad.

Respecto a los casos de **alerta o usencia de somnolencia** se obtiene una tasa global de **falsas detecciones del 0.07%**, idéntica a los resultados obtenidos con

la configuración de alta sensibilidad, lo cual era previsible porque el sistema es más “exigente” a la hora de declarar una alarma, por lo que la tasa no podría haber empeorado.

5.1.3.2 Grupo B

Los resultados en bruto que se obtienen son:

Sujeto bajo análisis	Somnolencia	Frames	Alarmas	por MLP	por Ojos cerrados	por PERCLOS	por Pose	Falsos negativos	Falsos positivos
04\\10.mp4	Y	15394	3	0	3	0	0	15391	0
10\\10.MOV	Y	18866	871	595	159	441	0	17995	0
51\\10.MOV	Y	18494	7399	1469	1583	5455	926	11095	0
57\\10.MOV	Y	18367	13068	2322	4664	12261	0	5299	0
59\\10.MOV	Y	18095	109	0	109	0	0	17986	0
60\\10.MOV	Y	18581	4	0	4	0	0	18577	0
04\\0.mp4	N	15364	118	90	34	0	0	0	118
10\\0.MOV	N	16309	19	0	19	0	0	0	19
51\\0.MOV	N	18179	1068	0	509	886	0	0	1068
57\\0.MOV	N	10863	59	0	59	0	0	0	59
59\\0.MOV	N	13120	0	0	0	0	0	0	0
60\\0.MOV	N	10897	0	0	0	0	0	0	0
TOTAL (somnolencia)		107797	21454	4386	6522	18157	926	86343	0
TOTAL (despierto)		84732	1264	90	621	886	0	0	1205

Tabla 28. Datos en bruto Sesión 2 – grupo B

El análisis estadístico que se obtiene analizando los datos en bruto arroja la tasa de acierto del sistema:

- Caso ideal si somnolencia (Y): Alarmas = 100%
- Caso ideal si no hay somnolencia (N): Alarmas = 0%

Por tanto la columna [Alarmas] determina el resultado global de la batería de pruebas:

Sujeto bajo análisis	Somnolencia	Alarmas	por MLP	por Ojos cerrados	por PERCLOS	Pose	Falsos negativos	Falsos positivos
04\\10.mp4	Y	0,02%	0,00%	0,02%	0,00%	0,00%	99,98%	0,00%
10\\10.MOV	Y	4,62%	3,15%	0,84%	2,34%	0,00%	95,38%	0,00%
51\\10.MOV	Y	40,01%	7,94%	8,56%	29,50%	5,01%	59,99%	0,00%

57\\10.MOV	Y	71,15%	12,64%	25,39%	66,76%	0,00%	28,85%	0,00%
59\\10.MOV	Y	0,60%	0,00%	0,60%	0,00%	0,00%	99,40%	0,00%
60\\10.MOV	Y	0,02%	0,00%	0,02%	0,00%	0,00%	99,98%	0,00%
04\\0.mp4	N	0,77%	0,59%	0,22%	0,00%	0,00%	0,00%	0,77%
10\\0.MOV	N	0,12%	0,00%	0,12%	0,00%	0,00%	0,00%	0,12%
51\\0.MOV	N	5,87%	0,00%	2,80%	4,87%	0,00%	0,00%	5,87%
57\\0.MOV	N	0,54%	0,00%	0,54%	0,00%	0,00%	0,00%	0,54%
59\\0.MOV	N	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
60\\0.MOV	N	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
TOTAL [somnolencia]	Y	19,90%	4,07%	6,05%	16,84%	0,86%	80,10%	0,00%
TOTAL [despierto]	N	1,49%	0,11%	0,73%	1,05%	0,00%	0,00%	1,42%

Tabla 29. Estadísticas. Sesión 2 – grupo B

Para un mejor análisis de los resultados, se agrupan los casos de somnolencia:

Sujeto	Alarmas [%]	MLP [%]	PERCLOS [%]	Ojos cerrados [%]	Sum of Pose
60\\10.MOV	0,02%	0,00%	0,00%	0,02%	0,00%
59\\10.MOV	0,60%	0,00%	0,00%	0,60%	0,00%
57\\10.MOV	71,15%	12,64%	66,76%	25,39%	0,00%
51\\10.MOV	40,01%	7,94%	29,50%	8,56%	5,01%
10\\10.MOV	4,62%	3,15%	2,34%	0,84%	0,00%
04\\10.mp4	0,02%	0,00%	0,00%	0,02%	0,00%
TOTAL [somnolencia]	19,90%	4,07%	16,84%	6,05%	0,86%

Tabla 30. Estadísticas caso somnolencia. Sesión 2 – grupo B

Por último, los resultados obtenidos se representan gráficamente:

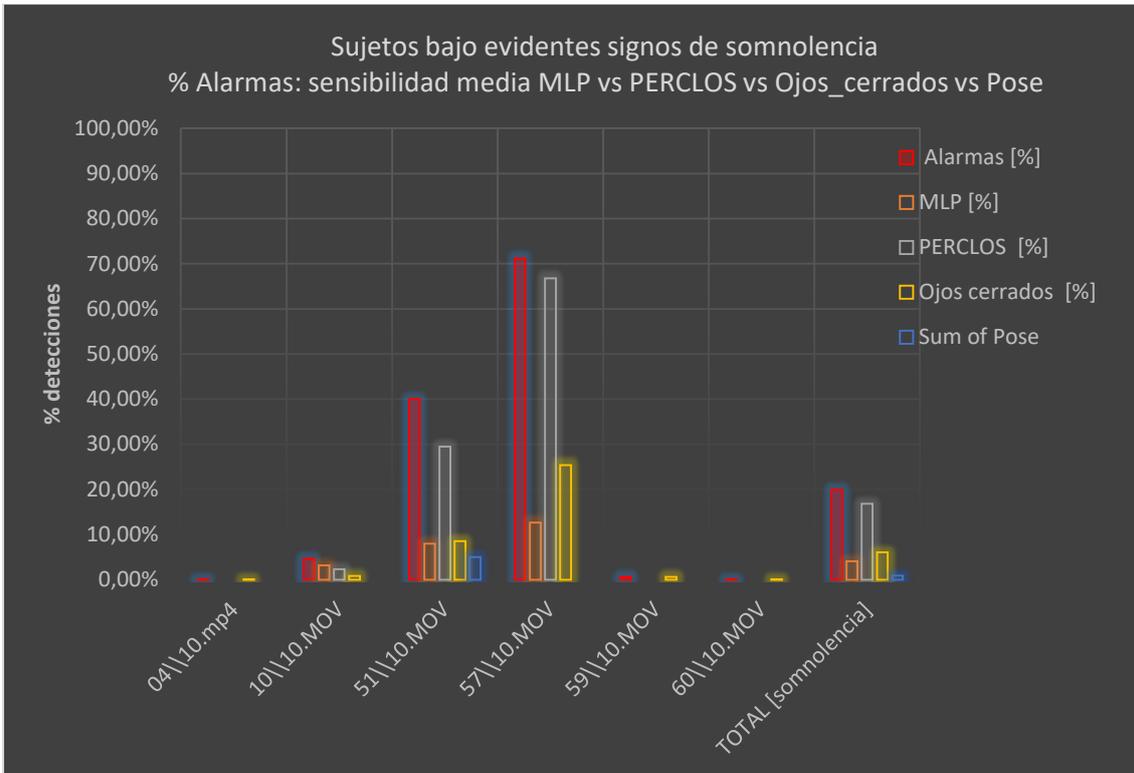


Figura 59. Gráfica estadísticas caso somnolencia. Sesión 2 – grupo B

De la misma forma, se realiza el agrupamiento de los casos donde no existe somnolencia así como su representación gráfica:

Sujeto	Alarmas [%]	MLP [%]	PERCLOS [%]	Ojos cerrados [%]	Pose [%]
60\\0.MOV	0,00%	0,00%	0,00%	0,00%	0,00%
59\\0.MOV	0,00%	0,00%	0,00%	0,00%	0,00%
57\\0.MOV	0,54%	0,00%	0,00%	0,54%	0,00%
51\\0.MOV	5,87%	0,00%	4,87%	2,80%	0,00%
10\\0.MOV	0,12%	0,00%	0,00%	0,12%	0,00%
04\\0.mp4	0,77%	0,59%	0,00%	0,22%	0,00%
TOTAL [despierto]	1,49%	0,11%	1,05%	0,73%	0,00%

Tabla 31. Estadísticas caso alerta. Sesión 2 – grupo B

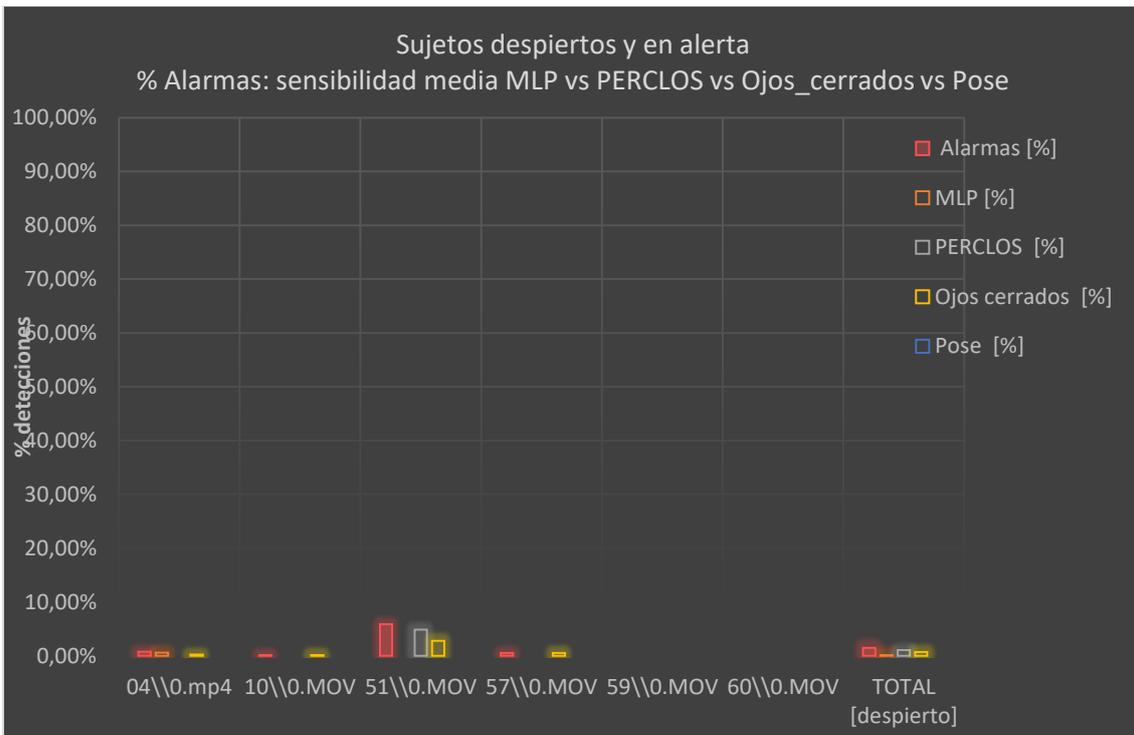


Figura 60. Gráfica estadísticas caso alerta. Sesión 2 – grupo B

El grupo B arroja unos resultados más pobres en cuanto a **detecciones**, resultando un total del **19.90%** para detecciones de somnolencia, si bien con comportamiento dispar entre sujetos: 71.15%, 40.01% para 51 y 57 respectivamente. El resto ofrecen tasas muy bajas o despreciables.

- Valorando los resultados globales se observa que la predicción de la red neuronal (MLP), se vuelve bastante inmune a los estímulos, que no son tan prolongados como se exige con la nueva configuración, de tal manera que el sólo ofrece detecciones para el 4.07% de los frames analizados mientras que la tasa de PERCLOS ofrece detecciones en el 16.84%, que si resultar un valor notable, es considerablemente mejor que el que ofrece la MLP.

Respecto a los casos de **alerta o usencia de somnolencia** se obtiene una tasa global de **falsas detecciones del 1.49%**, que supone un resultado considerablemente mejor que el obtenido con una configuración de alta sensibilidad para el mismo grupo de sujetos (33.34%) .

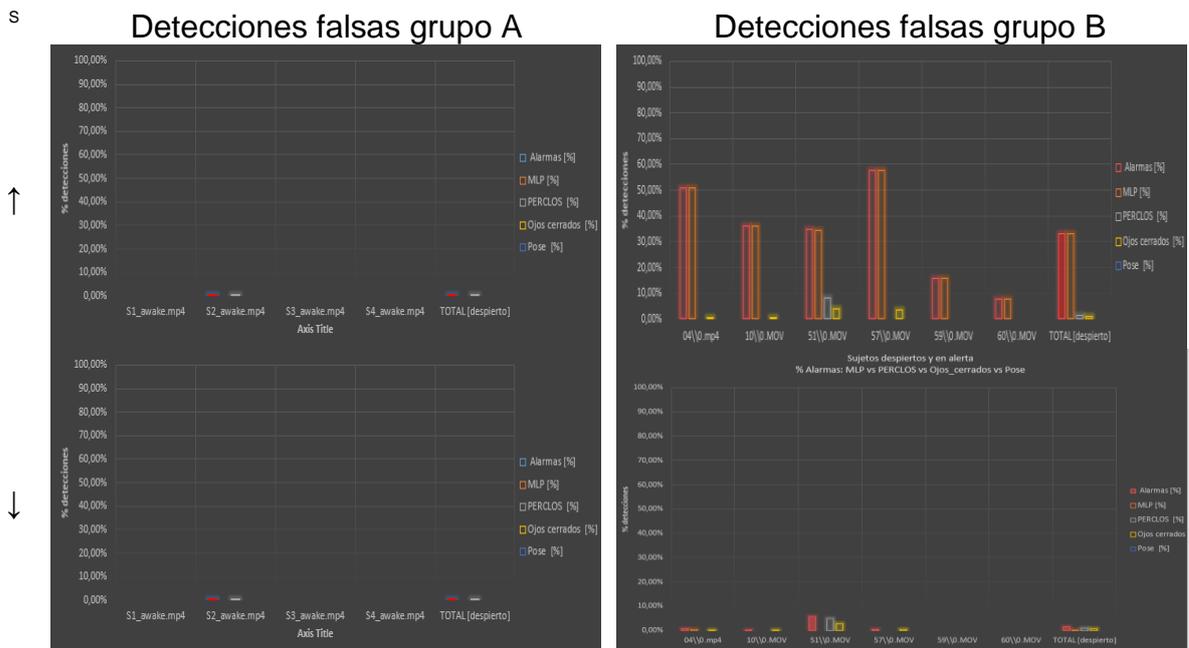
5.1.4 Comparativa entre sesiones de pruebas

A continuación se exponen dos matrices de gráficas de resultados. La fila 1 corresponde a la configuración de alta sensibilidad (↑) mientras que la fila 2 corresponde a la configuración con sensibilidad media (↓).

La primera muestra la tasa de éxito para sujetos que presentan síntomas de somnolencia.



En esta segunda matriz se muestra la tasa de error por falsas detecciones para para sujetos que no presentan síntomas.



6 Conclusiones y trabajo futuro

6.1 Conclusiones

Los sistemas de detección de fatiga, dadas sus claras ventajas a la hora de prevenir accidentes de tráfico representan una de las áreas de mayor interés de desarrollo por parte de la industria y la administración. No obstante, el elevado nivel de complejidad técnica se refleja en el hecho de que a fecha actual no existen aún sistemas lo suficientemente maduros y robustos como para suponer un estándar de facto.

Por tanto, el desarrollo y futura integración y despliegue de este tipo de sistemas en vehículos comerciales va a seguir necesitando grandes esfuerzos de investigación e ingeniería, tanto en el ámbito industrial como en el académico.

Por ello, el presente trabajo surge con la pretensión de, por un lado, investigar el prolífico estado del arte existente en la materia, y por otro realizar una modesta contribución con una propuesta de posible solución parcial: a tenor del estado del arte y de la experiencia empírica obtenida tras el desarrollo realizado, todo hace indicar que una solución lo suficientemente confiable y robusta debería combinar diferentes técnicas, tanto para monitorizar el estado de sujeto bajo estudio como para obtener parámetros de conducción del vehículo.

En otro orden, y respecto al campo de la inteligencia artificial y el uso de redes neuronales para encontrar patrones complejos como es el caso que se ha abordado en el presente trabajo, se concluye también que la solución propuesta requeriría un mayor esfuerzo en la recolección de datos de calidad para el entrenamiento de la red: dado lo ajustado de los plazos este trabajo ha tenido que realizarse construyendo un dataset de datos de entrenamiento *ad-hoc* con bajo volumen de datos y escasa variabilidad de éstos, lo cual necesariamente repercute en un peor rendimiento de la red implementada. En este sentido la construcción de un dataset de la máxima calidad y objetividad en su etiquetado, debería pasar por la captura de vídeos de sujetos mientras se someten a un análisis de electroencefalografía (EEG) de manera que se pueda tener verdadera fiabilidad.

Además, y dada la necesidad de encontrar patrones temporales sobre las series de datos a evaluar a lo largo del tiempo, sería necesario plantear arquitecturas de red que contemplaran el análisis de un mayor número de frames, por lo que posiblemente resultaría más adecuado usar redes de tipo long short-term memory (LSTM).

Finalmente, y a la vista de los resultados obtenidos en el capítulo Verificación y validación resulta evidente que otro de los grandes problemas en el ajuste de este tipo de sistemas reside en la dificultad de configurarlos y parametrizarlos de forma adecuada para evitar que una alta sensibilidad, que genera alto número de detecciones positivas, genere también de forma indeseada un elevado número de falsos positivos. Por contra, ajustar el sistema para hacerlo más inmune a falsos positivos conlleva también la reducción de la tasa de detecciones verdaderas.

6.2 Cumplimiento de objetivos

A continuación se muestra la lista de requisitos, funcionales y no funcionales, definidos al inicio y la marca OK y NOK si se han cumplido o no respectivamente.

En este sentido el único requisito que no se ha logrado tal como estaba definido es el Req_F-21: la red neuronal implementada, si bien es capaz de clasificar entre las clases somnolencia o alerta, hace uso de un histórico de datos relativamente pequeño (tres frames) de manera que en determinadas escenas ofrece resultados espúreos. Es por ello que, para aportarle mayor confiabilidad, se le ha añadido sendos retardadores de activación y desactivación.

Identificador	Descripción	Estado
Req_NF-1	El tiempo de ciclo completo será inferior a 100ms	OK
Req_NF-2	El sistema podrá reducir la imagen a procesar mediante parámetro de entrada	OK
Req_NF-3	El sistema hará detección de cara en un mínimo de un frame por cada dos frames consecutivos, en cuyo caso reutilizará las coordenadas de la detección anterior como argumento de entrada del estimador de coordenadas faciales	OK
Req_NF-4	El sistema implementará una red neuronal tipo perceptrón multicapa con Nx3 neuronas. Donde N será el número de frames. Los valores de entrada corresponderán con las excentricidades de los ojos y la boca	OK
Req_NF-5	El sistema mostrará datos actualizados cada, al menos, M frames sobre: <ul style="list-style-type: none"> • Valores de excentricidad de ambos ojos • Valor de excentricidad de la boca • Ángulo de la pose 	OK
Req_NF-6	El sistema calculará la tasa de frames por segundos a la que completa cada ciclo	OK

Identificador	Descripción	Estado
Req_F-7	El sistema hará adquisición de imágenes en cada ciclo	OK
Req_F-8	El sistema convertirá las imágenes a procesar escala de grises	OK
Req_F-9	El sistema debe detectar caras en escena usando imágenes en escala de grises devolviendo una ROI identificada por coordenadas de origen y tamaño	OK
Req_F-10	El sistema determinará las coordenadas faciales de contorno de ojos y boca por cada ROI en la que se haya detectado una cara	OK
Req_F-11	El sistema calculará las excentricidades de ambos ojos usando 6 puntos de contorno por cada ojo	OK

Req_F-12	El sistema calculará la excentricidad de la boca usando dos puntos que indiquen la posición de la comisura	OK
Req_F-13	El sistema almacenará los valores de excentricidades de ojos de hasta FRAMES_TO_CALCULATE_PERCLOS anteriores al frame actual	OK
Req_F-14	El sistema almacenará los valores de ángulo de boca de hasta FRAMES_TO_CALCULATE_RATES anteriores al frame actual	OK
Req_F-15	El sistema almacenará los valores de excentricidad de boca de hasta FRAMES_TO_CALCULATE_RATES anteriores al frame actual	OK
Req_F-16	El sistema considerará ojo cerrado si se supera el umbral de excentricidad EYES_ECC_THRESHOLD_CLOSED	OK
Req_F-17	El sistema contará el número de frames consecutivos en los que se detecta ojo cerrado	OK
Req_F-18	El sistema activará una señal de alarma si se supera el umbral de MAX_FRAMES_EYES_CLOSED frames con ojos cerrados	OK
Req_F-19	El sistema calculará la tasa de PERCLOS como el cociente entre el número de frames con los ojos cerrados y FRAMES_TO_CALCULATE_PERCLOS evaluado para los últimos FRAMES_TO_CALCULATE_PERCLOS frames	OK
Req_F-20	El sistema activará una señal de alarma si la tasa PERCLOS supera el umbral de PERCLOS_ALERTNESS_ZONE	OK
Req_F-21	El sistema activará una señal de alarma si la red neuronal clasifica los valores de entrada como somnolencia	NOK
Req_F-22	El sistema activará un aviso acústico si la señal de alarma se activa	OK
Req_F-23	El sistema activará un mensaje de alarma si se activa la señal de alarma	OK

En cuanto a la planificación, el grueso de los hitos contemplados ha encajado temporalmente y en esfuerzos si bien la estimación de éstos para la realización de los diferentes SWC no ha sido la más adecuada: en los inicios se pensaba que el esfuerzo de desarrollo vendría en la implementación del SWC1 y SWC2, sin embargo el punto que mayor esfuerzo ha requerido en tiempo y dedicación ha sido el desarrollo (preparación de datos, entrenamiento y validación) de la red neuronal, que se podría encuadrar dentro del SWC3.

6.3 Trabajo futuro

Como líneas de trabajo futuro se plantean las siguientes posibilidades:

- Reemplazo de la actual red neuronal por una red LSTM que permita un mejor desempeño
- La red neuronal debería manejar nuevos datos de entrada, como es el caso del ángulo de pose de la cabeza.

- Al objeto de mejorar el desempeño del algoritmo, éste debería ser migrado a C++.
- Para la solución empotrada sobre la plataforma de desarrollo NVIDIA Jetson Nano habría que aprovechar las posibilidades de procesamiento en paralelo que brinda su GPU. En este sentido no se ha profundizado en esta posibilidad porque la principal librería empleada (Dlib) no soporta aún esta posibilidad.
- Combinar las métricas obtenidas mediante visión artificial con medidas de un sensor de ángulo de giro, que pudiera simular la dirección del vehículo.

7 Glosario

ROI	Region of Interest
RNA	Red Neuronal Artificial
MLP	Multilayer Perceptron
HOG	Histogramm Oriented Gradient
FOV	Field of View
FPS	Frames per Second
SWC	Software Component
PERCLOS	Percentage of eye Closure

8 Bibliografía

- [1] Los accidentes de tráfico, principal causa de muerte en jóvenes
Autor: Dirección General de Tráfico (DGT)
Resumen: Artículo en el que se abordan con datos y estadísticas el impacto que tienen los accidentes de tráfico en el número de decesos anuales. Así mismo se aportan datos sobre cuántos de ellos tienen un origen en la somnolencia y/o fatiga del conductor.
URL: <http://revista.dgt.es/es/noticias/internacional/2018/1218oms-informe-mundial-accidentes-trafico.shtml>
Fecha de visita: 22/02/2021
- [2] Regulación 2018/0145 (COD) PE-CONS 82/19 del Consejo de la Unión Europea relativo a los requisitos de homologación para la seguridad general de los vehículos de motor.
Autor: Consejo de la Unión Europea
Fecha: 18 octubre 2019
URL: <https://data.consilium.europa.eu/doc/document/PE-82-2019-INIT/en/pdf>
Fecha de visita: 24/02/2021
- [3] Accidentes de tráfico mortales en carretera por países para el año 2018
Autor: Eurostat
URL: <https://ec.europa.eu/eurostat/cache/RCI/?vis=nuts2.transport&lang=en#?vis=nuts2.transport&lang=en>
Fecha de visita: 24/02/2021
- [4] Causas y síntomas del presíncope y del síncope.
Autor: Redacción médica
URL: <https://www.redaccionmedica.com/recursos-salud/diccionario-enfermedades/sincope>
Fecha de visita: 21/02/2021
- [5] Diferencias entre sistemas de seguridad activa y pasiva
Resumen: Artículo en el que se diferencian y enumeran algunos de los sistemas de seguridad activa y pasiva que se han ido incorporando en los últimos años, entre ellos los sistemas de asistencia a la conducción.
Autor: Real Automóvil Club de España
Fecha: 29 octubre 2019
URL: <https://www.race.es/sistemas-seguridad-pasiva-y-activa>
Fecha de visita: 21/02/2021
- [6] DS Driver Attention Monitoring
Resumen: Artículo en el que se explica la solución comercial que proporciona el fabricante francés DS para detectar signos de fatiga en el conductor.
Autor: Car and Driver
Fecha: 8 junio 2020
URL: <https://www.caranddriver.com/es/coches/planeta-motor/a32796345/deteccion-somnolencia-conductor-ds/>
Fecha de visita: 20/02/2021
- [7] Eye detection based on the Viola-Jones method and corners points
Resumen: Paper en el que se propone un método robusto de detección de ojos basado en el algoritmo de Viola & Jones y corner points (método Shi-Tomasi detector y K-means)
Autores: S. El Kaddouhi, A. Saaidi, M. Abarkan
Fecha: 1 marzo 2017
URL: https://www.researchgate.net/publication/314168190_Eye_detection_based_on_the_Viola-Jones_method_and_corners_points
Fecha de visita: 18/02/2021

- [8] Heart Rate Change While Drowsy Driving
Resumen: Estudio sobre la relación entre el ritmo cardíaco y la fatiga durante la conducción
Autores: Sang-Ho Jo, Jin-Myung Kim, and Dong Kyoo Kim
Fecha: 4 marzo 2019
URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6393761>
Fecha de visita: 13/03/2021
- [9] Changes in sleepiness and body temperature precede nocturnal sleep onset: evidence from a polysomnographic study in young men
Resumen: Estudio sobre los cambios de temperatura corporal durante la fase de inicio del sueño.
Autores: C. J. van den Heuvel, J. T. Noone, K. Lushington, D. Dawson
Fecha: 5 enero 2002
URL: <https://pubmed.ncbi.nlm.nih.gov/9785270/>
Fecha de visita: 13/03/2021
- [10] Development of drowsiness detection system
Resumen: Se abordan diferentes estrategias para detectar la somnolencia del conductor y se propone un sistema basado en detección de parpadeo.
Autores: H. Ueno, M. Kaneda, M. Tsukino
Fecha: 31 agosto – 2 septiembre 1994
URL: <https://ieeexplore.ieee.org/abstract/document/396873>
Fecha de visita: 10/03/2021
- [11] A Survey on State-of-the-Art Drowsiness Detection Techniques
Resumen: Estudio y análisis de las diferentes estrategias y técnicas para detectar la somnolencia del conductor agrupadas por comportamiento del conductor, parámetros de conducción registrados por el vehículo y parámetros fisiológicos.
Autores: Muhammad Ramzan, Hikmat Ullah Khan, Shahid Mahmood Awan, Amina Ismail, Mahwish Ilyas and Ahsan Mahmood
Fecha: 1 mayo 2019
URL: <https://ieeexplore.ieee.org/abstract/document/8704263>
Fecha de visita: 10/03/2021
- [12] PERCLOS: An Alertness Measure of the Past
Resumen: Estudio de diferentes métricas de la fatiga del conductor, entre ellas la tasa PERCLOS.
Autores: Udo Trutschel, Bill Sirois, David Sommer, Martin Golz, and Dave Edwards
Fecha: 28 junio 2011
URL: https://www.researchgate.net/publication/321841702_PERCLOS_An_Alertness_Measure_of_the_Past
Fecha de visita: 06/03/2021
- [13] Yawning detection for monitoring driver fatigue based on two cameras
Resumen: Sistema de detección de somnolencia basado en dos cámaras para analizar posición de la cabeza y tasa de bostezos respectivamente.
Autores: Lingling Li; Yangzhou Chen; Zhenlong Li
Fecha: 6 noviembre 2009
URL: <https://ieeexplore.ieee.org/abstract/document/5309841>
Fecha de visita: 07/03/2021
- [14] Human and Vehicle-Driver Drowsiness Detection by Facial Expression
Resumen: Algoritmo de detección de somnolencia basado reconocimiento de expresiones faciales utilizando toda la expresión facial: ojos, cejas, mejillas y boca.
Autores: Satori Hachisuka - Research Laboratories, DENSO Corporation, Aichi, Japan
Fecha: 19 septiembre 2013
URL: <https://ieeexplore.ieee.org/abstract/document/6603526>
Fecha de visita: 12/03/2021

- [15] Enhanced real-time head pose estimation system for mobile device
 Resumen: Técnica para la implementación en dispositivos móviles de un estimador de la postura de la cabeza en tiempo real.
 Autores: Euclides N. Arcoverde Neto, Rafael M. Duarte, Rafael M. Barreto, João Paulo Magalhães Carlos C. M. Bastos, Tsang Ing Ren, and George D. C. Cavalcanti
 Fecha: Abril 2014
 URL: https://www.researchgate.net/publication/279291928_Enhanced_real-time_head_pose_estimation_system_for_mobile_device
 Fecha de visita: 12/03/2021
- [16] Robust and Continuous Estimation of Driver Gaze Zone by Dynamic Analysis of Multiple Face Videos
 Resumen: Estimación de posición de la cabeza mediante la combinación de dos cámaras frontales y una trasera y la fusión de datos.
 Autores: Ashish Tawari y Mohan M. Trivedi
 Fecha: 8-11 junio 2014
 URL: <https://ieeexplore.ieee.org/abstract/document/6856607>
 Fecha de visita: 12/03/2021
- [17] Driver drowsiness detection using face expression recognition
 Resumen: Detección de cejas, ojos y boca mediante cámara en espectro visible con diodos IR y análisis de histograma horizontal.
 Autores: M. A. Assari; M. Rahmati
 Fecha: 16-18 november 2011
 URL: <https://ieeexplore.ieee.org/abstract/document/6144162>
 Fecha de visita: 13/03/2021
- [18] Driver Drowsiness Detection Based on Steering Wheel Data Applying Adaptive Neuro-Fuzzy Feature Selection
 Resumen: Sistema de adquisición y análisis de posición angular del volante en función de la geometría y curvatura de la carretera.
 Autores: Sadeqh Arefnezhad, Sajjad Samiee, Arno Eichberger, and Ali Nahvi
 Fecha: 15 enero 2019
 URL: <https://www.mdpi.com/1424-8220/19/4/943/htm>
 Fecha de visita: 13/03/2021
- [19] Steering wheel motion analysis for detection of the driver's drowsiness
 Resumen: Sistema de adquisición y análisis de posición angular del volante en función de la correlado con velocidad
 Autores: Daniel Haupt, Petr Honzík, Petr Honzík, Peter Raso, Ondrej Hyncica
 Fecha: diciembre 2011
 URL: https://www.researchgate.net/publication/261849547_Steering_wheel_motion_analysis_for_detection_of_the_driver's_drowsiness
 Fecha de visita: 13/03/2021
- [20] Safe driving by detecting lane discipline and driver drowsiness
 Resumen: Sistema de detección de somnolencia por análisis de seguimiento de carril
 Autores: Yashika Katyayal, Suhas Alur, Shipra Dwivedi
 Fecha: 8-10 mayo 2014
 URL: <https://ieeexplore.ieee.org/document/7019248>
 Fecha de visita: 08/03/2021
- [21] Electroencephalograms in man: Effect of voluntary movement upon the electrical activity of the precentral gyrus
 Resumen: Sistema de detección de somnolencia por análisis de seguimiento de carril
 Autores: Herbert Jasper D. es Sc., M. D. & Wilder Penfield M. D., F. R. S.
 Fecha: 1949
 URL: <https://link.springer.com/article/10.1007/BF01062488>

- [22] Detection of driver's drowsiness by means of HRV analysis
Autores: José Vicente; Pablo Laguna; Ariadna Bartra; Raquel.
Fecha: 18-21 septiembre 2011
URL: <https://ieeexplore.ieee.org/abstract/document/6164509>
Fecha de visita: 10/03/2021
- [23] Rapid object detection using a boosted cascade of simple features
Autores: Paul Viola, Michael Jones
Fecha: 8-14 diciembre 2001
URL: <https://ieeexplore.ieee.org/document/990517>
Fecha de visita: 26/03/2021
- [24] Detecting Faces in Images: A Survey
Autores: M. Yang, D.J. Kriegman, N. Ahuja
Fecha: enero 2002
URL: <https://ieeexplore.ieee.org/document/982883>
Fecha de visita: 27/03/2021
- [25] Diagnostic cluster analysis of mathematics skills
Autores: Yoon Soo Park, Young-Sun Lee
Fecha: enero 2011
URL: <https://www.researchgate.net/publication/263889770>
Fecha de visita: 27/03/2021
- [26] YOLOv3: An Incremental Improvement
Resumen: Red neuronal para detección en tiempo de real de objetos.
Autores: Joseph Redmon, Ali Farhadi
Fecha: abril 2018
URL: <https://www.researchgate.net/publication/324387691>
Fecha de visita: 27/03/2021
- [27] One Millisecond Face Alignment with an Ensemble of Regression Trees
Resumen: Algoritmo presentado en 2014 en la IEEE Conference on Computer Vision and Pattern Recognition que permite estimar la posición de de marcas faciales que definen los contornos de cara, cejas, ojos, nariz y boca
Autores: Vahid Kazemi, Josephine Sullivan
Fecha: 23-28 junio 2014
URL: <https://ieeexplore.ieee.org/document/6909637>
Fecha de visita: 03/04/2021
- [28] Real-Time Eye Blink Detection using Facial Landmarks
Resumen: Algoritmo presentado en 2016 para identificar parpadeos en tiempo real, usando fórmula que calcula EA
Autores: Tereza Soukupová y Jan Cech
Fecha: 3-5 febrero 2016
URL: <http://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf>
Fecha de visita: 03/04/2021
- [29] Facial expression recognition and histograms of oriented gradients: a comprehensive study
Resumen: Estudio del histograma de gradientes orientados HOG
Autores: Pierluigi Carcagnì, Marco Del Coco, Marco Leo, Cosimo Distanto
Fecha: noviembre 2015
URL: <https://www.researchgate.net/publication/283543555>
Fecha de visita: 22/04/2021
- [30] Feature Engineering for Images: A Valuable Introduction to the HOG Feature Descriptor
Resumen: Estudio del histograma de gradientes orientados HOG
Autores: Aishwarya Singh, revista Analytics Vidhya
Fecha: septiembre 2019

URL: <https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/>
Fecha de visita: 22/04/2021

[31] On-Road Detection of Driver Fatigue and Drowsiness during Medium-Distance Journeys
Autores: Luca Salvati,, Matteo d'Amore, Anita Fiorentino, Arcangelo Pellegrino, Pasquale Sena, Francesco Villecco
Fecha: 19 enero 2021
URL: <https://www.mdpi.com/1099-4300/23/2/135>
Fecha de visita: 24/04/2021

[32] A Realistic Dataset and Baseline Temporal Model for Early Drowsiness Detection
Autores: Reza Ghoddoosian, Marnim Galib, Vassilis Athitsos
Presentado: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops
Fecha: 15 abril 2021
URL: <https://arxiv.org/abs/1904.07312>
Fecha de visita: 27/04/2021