Pontificia Universidad
JAVERIANA
Cali

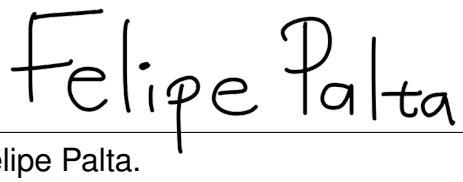
Santiago de Cali, 2025

Doctor
Gerardo Mauricio Sarria M.
Director Carrera Ingeniería de Sistemas y Computación
Pontificia Universidad Javeriana Cali

Cordial Saludo:

Por medio de la presente me permito informarle que he revisado el proyecto de grado ANÁLISIS DE FATIGA EN CONDUCTORES MEDIANTE PROCESAMIENTO DE IMÁGENES: UNA APROXIMACIÓN PARA LA SEGURIDAD VIAL de los estudiantes Miguel Ángel Cumbalaza Garcia y Johann Emilson Ruano Perez del cual soy Director y lo considero finalizado y listo para sustentación.

Atentamente,



Felipe Palta.
Director de Trabajo de Grado
Pontificia Universidad Javeriana Cali

Pontificia Universidad JAVERIANA Cali

Santiago de Cali, 2025

Doctor
Gerardo Mauricio Sarria M.
Director Carrera Ingeniería de Sistemas y Computación
Pontificia Universidad Javeriana Cali

Cordial Saludo:

Me permito presentar a su consideración el proyecto de grado denominado:ANÁLISIS DE FATIGA EN CONDUCTORES MEDIANTE PROCESAMIENTO DE IMÁGENES: UNA APROXIMACIÓN PARA LA SEGURIDAD VIAL, con el fin de cumplir con los requisitos exigidos por la Universidad para llevar a cabo el término de este y posteriormente optar por el título de Ingeniero de Sistemas y Computación.

Atentamente,

Miguel Angel Cumbalaza Garcia
Código: 8956850

Johann Emilson Ruano Perez
Código: 8953288

ANÁLISIS DE FATIGA EN CONDUCTORES MEDIANTE PROCESAMIENTO DE IMÁGENES: UNA APROXIMACIÓN PARA LA SEGURIDAD VIAL

MIGUEL ANGEL CUMBALAZA GARCIA
JOHANN EMILSON RUANO PEREZ

UNIVERSIDAD PONTIFICIA JAVERIANA CALI
FACULTAD DE INGENIERÍA Y CIENCIAS, DEPARTAMENTO
CALI
2025

ANÁLISIS DE FATIGA EN CONDUCTORES MEDIANTE PROCESAMIENTO DE IMÁGENES: UNA APROXIMACIÓN PARA LA SEGURIDAD VIAL

MIGUEL ANGEL CUMBALAZA GARCIA
JOHANN EMILSON RUANO PEREZ

TRABAJO DE GRADO

FELIPE PALTA MsC.

UNIVERSIDAD PONTIFICIA JAVERIANA CALI
FACULTAD DE INGENIERÍA Y CIENCIAS, DEPARTAMENTO
CALI
2025

Dedico esta tesis a mi familia, por ser mi mayor inspiración y apoyo a lo largo de este camino. Su amor, sacrificio y confianza en mí me han dado la fuerza para superar cada obstáculo y seguir adelante con determinación.

Gracias por estar presentes en cada etapa de mi vida, brindándome palabras de aliento y un hogar lleno de cariño. Han sido una base firme sobre la cual he construido mis sueños.

Agradezco profundamente a mis compañeros de universidad. Aunque no puedo mencionarlos a todos por nombre, valoro enormemente cada momento compartido, cada conversación, cada trabajo en equipo y cada gesto de apoyo. Su presencia hizo de esta experiencia algo más humano, más llevadero y lleno de aprendizajes que van más allá del aula.

Gracias a todos por acompañarme en este trayecto.

Miguel Angel Cumbalaza.

Dedico esta tesis a mis padres, Ana Lidia Perez Samboní y Emilson Ruano Caicedo. Gracias a su apoyo y esfuerzo, he logrado llegar hasta aquí. A mis tíos, tías, abuelas y demás familiares, les agradezco por ser un pilar importante en mi vida. Es por ellos que siempre doy lo mejor de mí.

Agradezco especialmente a mis compañeros de universidad. Aunque no puedo mencionarlos a todos por nombre, quiero expresar mi profunda gratitud por su apoyo incondicional, tanto en el ámbito académico como en el emocional. Gracias a ellos, este camino fue más llevadero y enriquecedor, y cada uno de ellos dejó una huella imborrable en esta etapa de mi vida.

Un agradecimiento especial al Presbítero Yoan José Pinzón Ardila, quien ha sido mi maestro en este camino. Su apoyo no solo ha sido académico, sino también emocional y espiritual. Gracias a él, he entendido que todo el conocimiento adquirido y el que aún me falta por aprender, tiene un propósito mayor: ponerlo al servicio de los demás.

Johann Emilson Ruano Perez.

0. Resumen

En este trabajo de grado se desarrolló un sistema automático no invasivo para la detección de fatiga en conductores, empleando exclusivamente imágenes faciales. El estudio se basó en el conjunto de datos UTA Real-Life Drowsiness Dataset, del cual se extrajeron y normalizaron regiones de interés (ojos, boca y mejillas) para calcular métricas geométricas y de color, así como para entrenar modelos de clasificación.

Se exploraron dos enfoques complementarios: (1) **aprendizaje automático superficial**, para el cual se diseñó un protocolo de extracción de características geométricas faciales (como apertura ocular, forma de la boca y tonalidad de mejillas). Estas métricas constituyeron el insumo para el entrenamiento de clasificadores tradicionales, incluyendo SVM, Random Forest, Decision Tree, KNN, Naive Bayes y MLP; y (2) **aprendizaje profundo**, mediante la arquitectura YOLO, que procesa imágenes completas y aprende representaciones directamente de los datos.

Ambos enfoques abordaron la clasificación en tres niveles de somnolencia: alerta (clase 0), baja vigilancia (clase 5) y somnolencia (clase 10).

Los resultados mostraron que los modelos superficiales alcanzaron exactitudes superiores al 95 % en varios casos, mientras que YOLO obtuvo un desempeño cercano al 100 %, con solo cuatro errores en casi 36 000 imágenes. Además, técnicas de interpretabilidad como Grad-CAM y mapas de oclusión confirmaron que las predicciones se basan en regiones faciales relevantes.

Estos hallazgos validan la viabilidad de sistemas de visión por computador para la detección de fatiga en tiempo real, con aplicaciones potenciales en seguridad vial, operación de flotas y desarrollo de sistemas avanzados de asistencia al conductor. Se identifican, no obstante, limitaciones relacionadas con la generalización a condiciones distintas de las del entrenamiento, lo que abre la puerta a futuras investigaciones en escenarios reales y con arquitecturas ligeras para implementación embebida.

0. Abstract

This thesis presents the development of a non-invasive automatic system for driver fatigue detection, based exclusively on facial image analysis. The study was conducted using the UTA Real-Life Drowsiness Dataset, from which regions of interest (eyes, mouth, and cheeks) were extracted and normalized to compute geometric and color-based metrics, as well as to train classification models.

Two complementary approaches were explored: (1) **shallow machine learning**, for which a protocol of geometric facial feature extraction (such as eye aperture, mouth shape, and cheek coloration) was designed. These metrics served as input for the training of traditional classifiers, including SVM, Random Forest, Decision Tree, KNN, Naive Bayes, and MLP; and (2) **deep learning**, through the YOLO architecture, which processes complete images and learns representations directly from the data.

Both approaches addressed a three-class problem: alert (class 0), low vigilance (class 5), and drowsy (class 10).

The results showed that shallow models achieved accuracies above 95 % in several cases, while YOLO reached nearly 100 % performance, with only four errors in almost 36,000 images. Furthermore, interpretability techniques such as Grad-CAM and occlusion maps confirmed that predictions were based on relevant facial regions.

These findings validate the feasibility of computer vision systems for real-time fatigue detection, with potential applications in road safety, fleet management, and the development of advanced driver assistance systems. Nevertheless, limitations related to generalization under conditions different from those in training were identified, highlighting the need for further research in real-world scenarios and the exploration of lightweight architectures for embedded deployment.

0. Glosario

API Siglas de *Application Programming Interface* (Interfaz de Programación de Aplicaciones). Conjunto de definiciones, protocolos y herramientas que permiten que diferentes programas o componentes de software se comuniquen entre sí. Una API expone funciones y servicios de forma controlada, facilitando la integración y automatización de procesos sin necesidad de conocer los detalles internos de su implementación. IV, 87

AUC *Area Under the Curve*. Métrica asociada a la curva ROC que cuantifica el rendimiento global de un clasificador. Un valor de 1 indica una clasificación perfecta, mientras que 0.5 corresponde a un rendimiento equivalente al azar.. IV, 119

CM *Coloración de Mejillas*. Métrica cromática obtenida a partir del promedio del componente *Hue* (H) del modelo de color HSV en regiones de interés ubicadas en las mejillas. Permite cuantificar variaciones en el tono de la piel, potencialmente asociadas a cambios fisiológicos como el enrojecimiento, y se utiliza como característica complementaria a las métricas geométricas faciales.. IV, 2, 8, 26, 35, 39, 46, 47, 65

CNN Convolutional Neural Network, red neuronal profunda diseñada para procesar datos con estructura en forma de rejilla, como imágenes, mediante el uso de convoluciones que permiten extraer características jerárquicas. IV, 2, 10–12, 16, 17, 19–23, 41, 44, 45

CSV *Comma-Separated Values*. Formato de archivo de texto plano utilizado para almacenar datos tabulares, donde cada línea representa un registro y los valores de cada campo están separados por comas (u otros delimitadores, como punto y coma o tabulador). Es ampliamente utilizado por su simplicidad y compatibilidad con hojas de cálculo, bases de datos y lenguajes de programación.. IV

Decision Tree Modelo de clasificación basado en reglas jerárquicas, fácil de interpretar. IV, 2, 3, 12, 47, 48

EAR Eye Aspect Ratio, métrica geométrica que mide la apertura ocular. IV, 2, 16, 17, 22, 23, 35–39, 46, 47, 64, 65, 69, 72, 73

ECG Siglas de *Electrocardiography* (electrocardiografía). Método para medir la actividad eléctrica del corazón mediante electrodos colocados en la piel. Permite evaluar el ritmo cardíaco y detectar alteraciones fisiológicas asociadas a fatiga o estrés.. IV, 7

EEG Siglas de *Electroencephalography* (electroencefalografía). Técnica de registro de la actividad eléctrica cerebral mediante electrodos colocados sobre el cuero cabelludo. Se utiliza para analizar patrones de ondas cerebrales y detectar estados como somnolencia, alerta o fatiga.. IV, 7

EOG Siglas de *Electrooculography* (electrooculografía). Técnica que registra los movimientos oculares midiendo las diferencias de potencial eléctrico entre la córnea y la retina, a través de electrodos ubicados alrededor de los ojos. Es útil para detectar parpadeo, cierre ocular y patrones de mirada.. IV, 7

F1-score Media armónica entre precisión y recall. IV, 13, 48

Grad-CAM Gradient-weighted Class Activation Mapping, técnica de interpretabilidad visual. III, IV, 50, 118, 119

GSR Siglas de *Galvanic Skin Response* (respuesta galvánica de la piel). Técnica que mide cambios en la conductancia eléctrica de la piel, asociados a la actividad de las glándulas sudoríparas y al nivel de activación del sistema nervioso autónomo. Es sensible a factores como humedad, temperatura y estado emocional.. IV, 7

Haar Cascada de clasificadores basada en características Haar, utilizada para la detección rápida de rostros en imágenes. IV, 47, 50

KNN K-Nearest Neighbors, clasificador basado en los vecinos más cercanos. IV, 2, 3, 12, 14, 16, 19–23, 27, 32, 46–48

KSS Escala de Somnolencia de Karolinska (*Karolinska Sleepiness Scale*), instrumento subjetivo ampliamente utilizado para medir el nivel de somnolencia percibida por un individuo. Utiliza una escala de 9 puntos, donde 1 indica un estado de alerta muy alto y 9 corresponde a un estado de somnolencia extrema. Es empleada en estudios de fatiga, investigación del sueño y evaluación de la vigilancia en tareas críticas como la conducción. IV, 54, 58

LDA Linear Discriminant Analysis, técnica supervisada que maximiza la separabilidad entre clases. IV, 12, 16, 40, 41, 72, 74, 75, 78

mAP Mean Average Precision, métrica que promedia la precisión en distintos umbrales. IV

MAR Mouth Aspect Ratio, métrica que evalúa la apertura de la boca para detectar somnolencia. IV, 2, 16, 17, 22, 23, 35–37, 39, 46, 47, 64, 65, 69, 71–73

MLP Multilayer Perceptron, red neuronal con múltiples capas ocultas para clasificación binaria. IV, 2, 3, 11, 12, 15, 20, 21, 27, 31, 46–48

MOE Measure of Eye Occlusion, métrica derivada del EAR que representa el grado de cierre ocular. IV, 2, 35, 38, 39, 46, 47, 64, 65, 69, 72, 73

Naive Bayes Clasificador probabilístico basado en el teorema de Bayes con independencia entre variables. IV, 2, 3, 13, 14, 27, 32, 33, 46–48

PCA Principal Component Analysis, técnica de reducción de dimensionalidad. IV, 12, 19, 22, 23, 40, 41, 72, 74, 75, 78

Precision Proporción de verdaderos positivos sobre el total de predicciones positivas. IV

PUC Percentage of Unit Contact, variante del EAR más sensible a cambios sutiles. IV, 2, 19, 22, 23, 35, 37–39, 46, 47, 64, 65, 69, 72, 73

Random Forest Bosque de árboles de decisión que mejora la precisión mediante votación entre modelos. IV, 2, 3, 14, 16, 19–22, 27, 31, 46–48

Recall Proporción de verdaderos positivos detectados sobre el total de casos reales. IV

RGB Modelo de color aditivo basado en la combinación de los tres colores primarios: **Rojo** (Red), **Verde** (Green) y **Azul** (Blue). Cada píxel en una imagen RGB se representa mediante una tripleta de valores que indican la intensidad de cada componente en un rango típico de 0 a 255. Este modelo es ampliamente utilizado en procesamiento de imágenes, visión por computadora y sistemas de visualización digital. IV, 53, 54

ROC *Receiver Operating Characteristic*. Curva que representa la relación entre la tasa de verdaderos positivos (*TPR*) y la tasa de falsos positivos (*FPR*) para diferentes umbrales de decisión. Se utiliza para evaluar el rendimiento de clasificadores binarios y multiclase.. IV, V, 119

ROI *Region of Interest* o Región de Interés. Área específica de una imagen seleccionada para análisis, en la que se aplican cálculos o extracciones de características. En este trabajo, las ROIs corresponden a zonas faciales como ojos, boca y mejillas.. IV

StratifiedKFold Técnica de validación cruzada que preserva la proporción de clases en cada partición. IV, 44

SVM Support Vector Machine, algoritmo que busca el hiperplano óptimo para separar clases. IV, 2, 3, 10, 13, 14, 16, 19, 21–23, 27, 30, 31, 46–48

YOLO Algoritmo de detección de objetos en tiempo real que realiza predicciones en una sola pasada por la red neuronal. III, IV, 2, 3, 41–43, 45, 46, 50, 52, 54, 82–85, 87, 90, 118, 119

0. Índice general

1. Introducción

En las últimas décadas, la seguridad vial ha ganado protagonismo como uno de los principales desafíos en el ámbito del transporte. La fatiga y la somnolencia al volante son factores recurrentes en la generación de accidentes graves, ya que reducen la atención y los reflejos del conductor, incrementando el riesgo de siniestros con consecuencias potencialmente letales. En Estados Unidos, la Administración Nacional de Seguridad del Tráfico en las Carreteras (*NHTSA*) reportó que en 2021 se registraron al menos **684 muertes** relacionadas con conducción somnolienta, y se estima que ocurren más de **100,000 accidentes** anuales por esta causa [15]. En Colombia, según el Anuario Estadístico de la Agencia Nacional de Seguridad Vial, más de **1,000 siniestros fatales** fueron clasificados como “otros” en 2023, una categoría que incluye incidentes vinculados a microsueños y fatiga, aunque no siempre se reportan explícitamente [16].

La Figura 1.1 ilustra la distribución de siniestros con lesionados y fallecidos según el tipo de incidente en Colombia, destacando la persistencia de esta categoría en la estadística nacional. Para dimensionar la magnitud del problema en un contexto más amplio, la Tabla 1.1 presenta una comparación entre las cifras nacionales, las reportadas en Estados Unidos y una estimación global basada en estudios de la Fundación AAA, que indican que aproximadamente el **20 %** de las muertes en carretera están relacionadas con la somnolencia al volante [17].

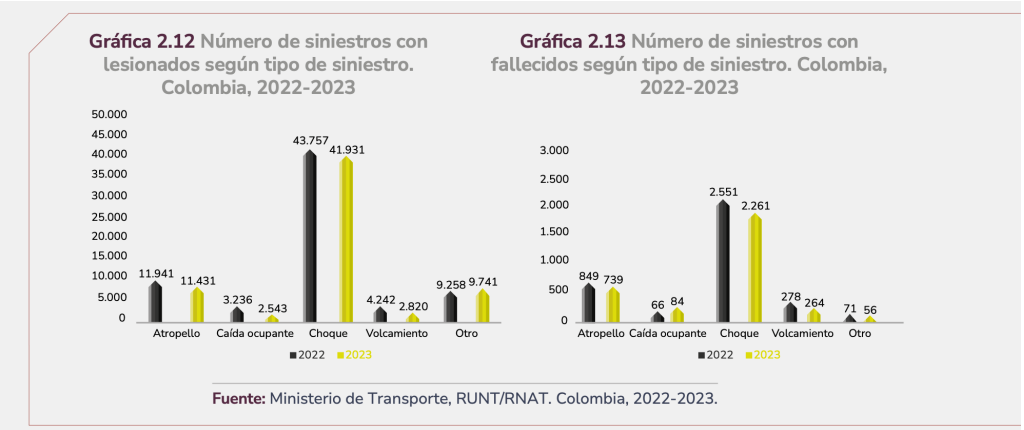


Figura 1.1: Número de siniestros con lesionados y fallecidos según tipo de siniestro en Colombia (2022–2023). Fuente: ANSV [16].

Región	Muertes por somnolencia	Fuente
Colombia (2023)	>1,000	ANSV [16]
Estados Unidos (2021)	684	NHTSA [15]
Estimación global	$\sim$20,000	Fundación AAA [17]

Cuadro 1.1: Comparación de muertes por conducción somnolienta en Colombia, EE.UU. y estimación global.

Adicionalmente, la Fundación AAA alerta que conducir tras dormir menos de cinco horas produce en el cerebro efectos comparables a una concentración de alcohol en sangre de 0.10

Los métodos tradicionales de detección de fatiga se basan en sensores fisiológicos invasivos o en el análisis de parámetros vehiculares, como patrones de giro de volante o variaciones de velocidad. Estas aproximaciones, aunque útiles, presentan limitaciones prácticas: requieren equipamiento costoso, resultan incómodas para el usuario o pierden precisión ante cambios en las condiciones de iluminación y el entorno vial. Por ello, surge la necesidad de soluciones no invasivas, económicas y capaces de operar en tiempo real.

La visión por computadora ofrece una alternativa prometedora al aprovechar señales visuales del rostro para inferir estados de alerta y somnolencia. En este trabajo se emplean dos estrategias complementarias para clasificar la somnolencia en tres niveles discretos (0: alerta; 5: somnolencia leve; 10: somnolencia profunda):

1. Extracción manual de características faciales - Cálculo de métricas como la proporción de apertura de ojos (EAR), la apertura de boca (MAR), el porcentaje de unidad de contacto (PUC), la oclusión ocular (MOE) y la coloración de mejillas (CM). - Entrenamiento de clasificadores supervisados clásicos: MLP, SVM, Random Forest, Decision Tree, KNN y Naive Bayes.

2. Aprendizaje profundo con detector YOLO - Uso de una arquitectura CNN (YOLO) que autoextrae características faciales y clasifica simultáneamente los mismos tres niveles.

En este proyecto, se encuentra un desarrollo basado en los siguientes lineamientos genéricos, que serán explicados con mayor detalle en el Capítulo de Metodología.

- Desarrollo de un flujo de preprocesamiento que, a partir de los fotogramas de video, extrae y normaliza automáticamente las características faciales clave, las cuales constituyen el insumo para el posterior entrenamiento de modelos de aprendi-

zaje automático superficial, con el objetivo de clasificar los niveles de somnolencia.

- Entrenamiento de modelos de aprendizaje automático basado en dos enfoques: algoritmos de aprendizaje supervisado superficial (MLP, SVM, Random Forest, Decision Tree, KNN, Naive Bayes) y aprendizaje profundo supervisado utilizando YOLOv8 para la clasificación de los tres niveles de somnolencia.
- Evaluación del desempeño de los modelos entrenados y determinación de su alcance y limitaciones.

2. Descripción del Problema

Planteamiento del Problema

En las últimas décadas, la seguridad vial ha enfrentado múltiples desafíos, entre los cuales la fatiga y la somnolencia al volante destacan por ser estados silenciosos y difíciles de detectar. Estos fenómenos deterioran de forma progresiva la atención, alargan los tiempos de reacción y merman el control motor, elementos críticos para mantener la trayectoria y la estabilidad del vehículo.

Particularmente preocupantes son los episodios de microsueño, que pueden durar tan solo unos segundos pero bastan para que el conductor pierda el dominio del volante y se desplace fuera de su carril. La naturaleza fugaz de estos lapsos de inconsciencia complica su detección en tiempo real y los convierte en una amenaza "invisible" tanto para el propio conductor como para el resto de los usuarios de la vía.

Aunque en los últimos años se han desarrollado sistemas de alerta basados en parámetros fisiológicos y señales vehiculares, su adopción masiva sigue limitada por barreras de comodidad, costos y sensibilidad a las condiciones ambientales. En este contexto, la visión por computadora emerge como una vía prometedora para detectar cambios sutiles en la expresión facial y los movimientos oculares que preceden a la pérdida de alerta, sin requerir contacto físico ni equipamiento especializado.

2.1.1. Contexto Estadístico

La *NHTSA* de EE. UU. estimó más de **100 000 accidentes** y **684 muertes** por conducción somnolienta en 2021 [15]. En Colombia, la *ANSV* reportó más de **1 000 siniestros fatales** relacionados con fatiga en 2023 [16]. A nivel global, la Fundación AAA sitúa en el **20 %** la proporción de muertes viales atribuibles a la somnolencia [17]. Estas cifras ponen de relieve la magnitud del problema y la urgencia de contar con sistemas de detección temprana y alerta oportuna.

2.1.2. Árbol de Problemas

El fenómeno de los **altos índices de accidentalidad vial en conductores** constituye una problemática de gran impacto humano, social y económico. Este tipo de siniestros responde a un conjunto de causas que actúan en distintos niveles: desde factores estructurales del sistema de transporte y la regulación, hasta condiciones operativas y estados inmediatos del conductor. Entre estas causas, la somnolencia al volante se identifica como un factor de riesgo relevante por su carácter silencioso y difícil detección, capaz de derivar en episodios de microsueño y pérdida de control del vehículo. La Figura 2.1 presenta el árbol de problemas elaborado, en el que se organizan las causas en tres niveles —estructurales, intermedias e inmediatas— y se muestran los principales efectos que la problemática central genera en diferentes ámbitos.

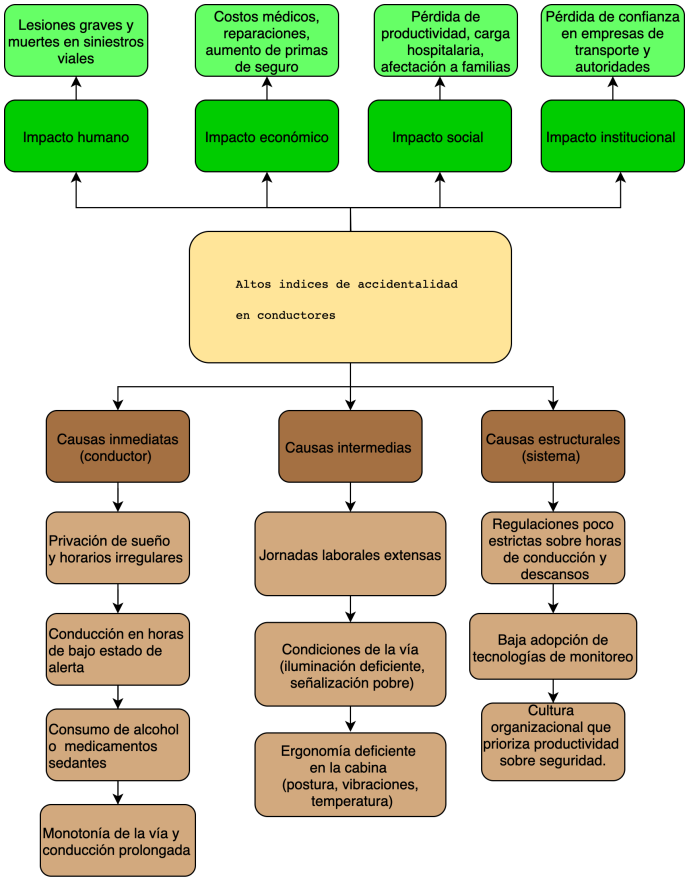


Figura 2.1: Árbol de problemas: causas, problema central y efectos asociados a la somnolencia y otros factores humanos en la conducción.

Las causas identificadas se agrupan en tres niveles:

1. **Causas inmediatas (conductor):** privación de sueño, ritmos circadianos altera-

dos, consumo de sedantes o alcohol, monotonía y distracciones en la vía.

2. **Causas intermedias (operativas):** jornadas laborales extensas, condiciones de iluminación deficiente, ergonomía inadecuada del asiento y estímulos externos constantes.

3. **Causas estructurales (sistema):** regulaciones insuficientes sobre tiempos de conducción y descanso, baja adopción de tecnologías de monitoreo y cultura organizacional proclive a la sobrecarga de turnos.

Los efectos derivados de esta problemática abarcan el impacto humano (lesiones y muertes), económico (costos médicos, reparación de bienes, seguros), social (pérdida de productividad, carga hospitalaria, afectación a familias) e institucional (pérdida de confianza en empresas de transporte y autoridades).

Del análisis emergen barreras adicionales, como la falta de protocolos de reporte estandarizados, umbrales de alerta inadecuados y disparidad normativa entre regiones.

2.1.3. Limitaciones de los Métodos Actuales

En la literatura se han explorado diversos enfoques para la detección de fatiga y somnolencia en conductores. Cada uno de ellos presenta ventajas particulares, pero también limitaciones que dificultan su adopción masiva en entornos reales de conducción. En la Tabla 2.1 se resumen los principales métodos reportados, junto con sus aspectos positivos y negativos. El análisis comparativo permite observar que, si bien algunos enfoques ofrecen alta precisión en condiciones controladas, suelen ser invasivos, costosos o poco adaptables a diferentes contextos. En este escenario, la visión por computador destaca como una alternativa no invasiva, capaz de operar con equipamiento estándar y sin interferir con la comodidad del conductor.

Cuadro 2.1: Comparativa de métodos para la detección de fatiga en conductores

Método	**Pros**	**Contras**
Electrofisiología (EEG, ECG, EOG)	Alta precisión en la medición de señales fisiológicas; detección directa de cambios en el estado de alerta.	Invasivos; requieren contacto continuo con el conductor; alto costo de implementación y mantenimiento; incomodidad prolongada.
Respuesta galvánica de la piel (GSR)	Equipos relativamente simples; detección de cambios en la actividad del sistema nervioso autónomo.	Sensible a la humedad y temperatura; provoca irritación cutánea; variabilidad alta según el entorno.
Señales vehiculares	No requieren contacto físico con el conductor; pueden integrarse a sistemas del vehículo.	Sensibles al estilo de conducción, tipo de vía y condiciones ambientales; alta tasa de falsas alarmas; no detectan la causa real de la fatiga.
Enfoques híbridos	Combinan múltiples fuentes de datos para mejorar la cobertura y fiabilidad.	Mayor complejidad técnica; costos elevados; mantenimiento frecuente; menor aceptación por parte de los usuarios.

En síntesis, aunque cada método tiene fortalezas, las limitaciones en términos de invasividad, costo, adaptabilidad y aceptación del usuario han motivado la búsqueda de alternativas más prácticas. La visión por computador se posiciona como una opción prometedora al ser no invasiva, escalable y compatible con hardware accesible, lo que facilita su implementación en contextos reales de conducción.

2.1.4. Necesidad y Enunciado del Problema

El análisis del árbol de problemas y la revisión de las limitaciones de los métodos actuales evidencian la necesidad de soluciones no invasivas, precisas y adaptables para la detección temprana de somnolencia en conductores. La visión por computador se presenta como una alternativa viable, al permitir inferir el estado de alerta a partir de señales faciales —como apertura de párpados, parpadeo, gestos de la boca y variaciones en la CM— sin contacto físico ni equipamiento especializado.

En este proyecto se plantea la necesidad de desarrollar y evaluar, en un entorno controlado y a partir de un conjunto de imágenes previamente capturadas, un sistema de visión por computador para la detección no invasiva y oportuna de somnolencia en conductores.

En este contexto, el trabajo se orienta a responder la siguiente cuestión: *¿cómo detectar la fatiga en conductores mediante el análisis de imágenes faciales?*

A partir de esta pregunta, el objetivo central consiste en diseñar y validar un sistema que, empleando técnicas de aprendizaje automático superficial y profundo, permita explorar la viabilidad de la clasificación de distintos niveles de somnolencia en condiciones controladas, como paso previo hacia futuras aplicaciones en escenarios reales.

Objetivos

2.2.1. Objetivo General

Desarrollar y validar un modelo de aprendizaje automático avanzado que sea capaz de detectar la fatiga en conductores mediante el análisis de imágenes.

2.2.2. Objetivos Específicos

1. Investigar técnicas de captura y procesamiento de imágenes óptimas para la detección de fatiga en conductores.

2. Analizar características visuales asociadas con la fatiga en imágenes de conductores, como la apariencia facial, los movimientos oculares y la postura.

3. Recopilar un conjunto de datos representativo que incluya imágenes de conductores en diversos estados de fatiga.

4. Implementar varios modelos de aprendizaje automático, como redes neuronales convolucionales (CNN), para detectar fatiga en imágenes.

5. Evaluar el rendimiento de los modelos utilizando métricas de precisión, sensibilidad y especificidad en conjuntos de datos de prueba.

Justificación

La siniestralidad asociada a somnolencia en Colombia y a nivel global evidencia una necesidad concreta: detectar a tiempo signos faciales de fatiga que preceden episodios de microsueño y habilitar una alerta oportuna en el vehículo. Los métodos predominantes presentan barreras de adopción: los fisiológicos son invasivos y costosos; los basados en señales vehiculares son indirectos y sensibles al estilo de conducción y al contexto vial. Esta brecha abre espacio a soluciones no invasivas, de bajo costo marginal y escalables.

Con base en este panorama, la justificación del proyecto se organiza en cuatro ejes: pertinencia del enfoque, aportes específicos, impacto esperado y viabilidad técnica.

Por qué este enfoque

La elección de la visión por computador como base de este trabajo responde a su alineación directa con la naturaleza del problema: la somnolencia se manifiesta principalmente a través de señales faciales como la dinámica de los párpados, la apertura bucal y ciertas microexpresiones, las cuales pueden ser observadas y analizadas en el mismo punto donde ocurren. Este enfoque, además, presenta la ventaja de ser no invasivo y escalable, ya que puede implementarse utilizando cámaras que suelen estar disponibles en cabinas o entornos de monitoreo, junto con hardware de fácil acceso, evitando así el contacto físico y la necesidad de instrumentación adicional compleja. A ello se suma que, en pruebas controladas, el procesamiento de cada fotograma se realizó con latencias del orden de milisegundos, lo que evidencia un potencial significativo para generar alertas con baja demora, siempre dentro del alcance experimental de este proyecto y con la posibilidad de evolucionar hacia aplicaciones en escenarios operativos reales.

Alcances y Límites

Este proyecto se centra en el desarrollo y validación de un sistema de detección de fatiga en conductores, empleando tecnologías de visión por computadora y aprendizaje automático. El trabajo incluirá el entrenamiento de al menos tres modelos diferentes, con especial atención en las redes neuronales convolucionales (CNN), utilizando una base de datos específica de conductores distraídos que se encuentra disponible públicamente en Kaggle (UTA Real-Life Drowsiness Dataset). La selección del modelo más efectivo se realizará mediante criterios rigurosos de precisión, sensibilidad y especificidad para garantizar que el sistema pueda identificar correctamente los signos de fatiga.

El alcance del proyecto comprende la construcción de un prototipo funcional que pueda emitir alertas en tiempo real. Sin embargo, no se abordará el desarrollo de software para usuarios finales ni la implementación completa del sistema en vehículos en entornos operativos reales durante esta fase. Además, aunque el sistema estará diseñado para poder funcionar bajo diferentes condiciones de iluminación y variadas posturas de los conductores, las pruebas del modelo se limitarán al uso de datos de la base de Kaggle, sin incluir ensayos en carreteras o pruebas con conductores en condiciones reales de manejo, dejando espacio para futuras investigaciones y desarrollos basados en los resultados obtenidos y las evaluaciones preliminares.

Trabajos relacionados

A Real-time Fatigue Detection System using Multi-Task Cascaded CNN Model El artículo titulado *A Real-time Fatigue Detection System using Multi-Task Cascaded CNN Model* aborda la problemática de la somnolencia al volante, una de las principales causas de accidentes automovilísticos con graves consecuencias para la seguridad vial. Según la Organización Mundial de la Salud (OMS), se estima que ocurren 1.22 millones de accidentes de tráfico cada día debido a la fatiga de los conductores [1].

El estudio propone un nuevo enfoque para analizar y detectar la somnolencia del conductor utilizando técnicas de aprendizaje profundo basadas en Redes Neuronales Convolucionales (CNN). A diferencia de otros métodos que utilizan Máquinas de Vectores de Soporte (SVM) para monitorear los niveles de fatiga, este enfoque se centra en el uso de movimientos faciales para obtener características más precisas, como el porcentaje de cierre de párpados sobre la pupila a lo largo del tiempo (PERCLOS), el bostezo y el

movimiento de la cabeza.

El modelo propuesto, denominado MTCNN (Multi-Task Cascaded Convolutional Neural Network), ha demostrado ser capaz de alcanzar una tasa de precisión del 97.5 %. Este alto nivel de precisión sugiere que el MTCNN es una herramienta eficaz para detectar la somnolencia en tiempo real y alertar a los conductores, lo que podría ayudar a prevenir numerosos accidentes de tráfico [1].

Real Time Drowsiness Detection System based on ResNet-50 El artículo titulado *Real Time Drowsiness Detection System based on ResNet-50* aborda la alta frecuencia de accidentes de tráfico causados por la somnolencia al volante. Según el Consejo Nacional de Seguridad (NSC), la conducción somnolienta es una de las principales causas de accidentes, representando aproximadamente el 9.5 % de todos los accidentes, es decir, alrededor de 100,000 colisiones [2].

La fatiga del conductor puede tener múltiples causas, como la falta de sueño, viajes largos, inquietud, consumo de alcohol y presión mental. Estudios previos en este dominio incluyen la detección de somnolencia utilizando aprendizaje automático y perceptrón multicapa (MLP). Este artículo propone un modelo con mayor precisión, donde se utilizan cámaras para capturar los puntos faciales. Para facilitar la predicción de la somnolencia, estas imágenes se procesan a través de una Red Neuronal Convolucional (CNN).

Las imágenes se recopilan y analizan minuciosamente para detectar las expresiones faciales y la posición de la cabeza del conductor en condiciones de somnolencia. Luego, utilizando el modelo de red neuronal de aprendizaje por transferencia ResNet-50, se construyen conjuntos de datos y se desarrolla un sistema para detectar y clasificar la somnolencia. Finalmente, la conclusión del experimento valida la eficacia del modelo propuesto, mostrando una precisión del 98.60 % [2].

Driver drowsiness detection using Convolutional Neural Networks-inspired features and Principal component analysis with K-Nearest Neighbors El artículo titulado *Driver drowsiness detection using Convolutional Neural Networks-inspired features and Principal component analysis with K-Nearest Neighbors* aborda la problemática de la somnolencia y la fatiga como principales causas de accidentes de tráfico graves en Zimbabue y en todo el mundo. Los avances tecnológicos recientes han proporcionado apoyo a los conductores mediante sistemas automovilísticos inteligentes.

Varios estudios de investigación han utilizado conjuntos de datos de un solo conductor para el entrenamiento y la prueba, y en muchos casos, se han utilizado principalmente imágenes diurnas. Por lo tanto, la fatiga y la somnolencia son campos de estudio clave para prevenir numerosos accidentes inducidos por el sueño.

En este artículo, se propusieron dos métodos de extracción de características: el Perceptrón Multicapa (MLP) y la Red Neuronal Convolucional (CNN). Además, se utilizó el método de Análisis de Componentes Principales (PCA) para la reducción de dimensionalidad. Basándose en estos métodos, se emplearon cinco clasificadores para detectar la somnolencia del conductor: LDA, XGBoost, LR, Árbol de Decisión (Decision Tree) y K-Nearest Neighbors (KNN).

Se realizaron experimentos para examinar la capacidad y utilidad de los enfoques en comparación con otras técnicas. Los resultados experimentales demostraron que la técnica de extracción de características basada en CNN proporcionó alta precisión en los cinco clasificadores. El KNN fue el mejor clasificador promedio con una precisión del 100 %. Los resultados experimentales indicaron que el PCA mejoró los clasificadores. Este estudio ofrece respuestas importantes y significativas en la práctica para reducir los accidentes de vehículos debido a la somnolencia [3].

Driver Mental Fatigue Detection Based on Head Posture Using New Modified reLU-BiLSTM Deep Neural Network El artículo titulado *Driver Mental Fatigue Detection Based on Head Posture Using New Modified reLU-BiLSTM Deep Neural Network* se centra en la detección temprana de la fatiga mental del conductor, un área de investigación activa en vehículos inteligentes. Existen varios métodos basados en la medición de las características fisiológicas del conductor utilizando sensores y visión por computadora.

En general, el comportamiento del conductor es impredecible y puede cambiar repentinamente bajo fatiga mental, lo que resulta en variaciones súbitas en la postura corporal y el movimiento de la cabeza, con un comportamiento inatento que puede terminar en accidentes fatales. Este artículo contribuye a avanzar en los enfoques de medición directa al proponer un nuevo método para medir la fatiga mental y la somnolencia del conductor mediante el monitoreo de los movimientos de la postura de la cabeza utilizando el sistema de captura de movimiento XSENS.

Los experimentos se realizaron en 15 sujetos sanos utilizando un simulador de con-

ducción de MATHWORKS (DIL) integrado con Unreal Engine 4. Se diseñó, entrenó y probó una nueva red neuronal profunda bidireccional de memoria a largo plazo modificada, basada en una capa de unidad lineal rectificada, en datos de aceleración angular de la cabeza en series temporales 3D para la clasificación secuencial. Los resultados mostraron que el clasificador propuesto superó a los enfoques de última generación y a las herramientas convencionales de aprendizaje automático, reconociendo con éxito los estados activos, de fatiga y de transición del conductor, con una precisión de entrenamiento del 99.2 %, sensibilidad del 97.54 %, precisión y puntuaciones F1-score del 97.38 % y 97.46 %, respectivamente [4].

Intelligent Driver Drowsiness Detection for Traffic Safety Based on Multi CNN Deep Model and Facial Subsampling El artículo titulado *Intelligent Driver Drowsiness Detection for Traffic Safety Based on Multi CNN Deep Model and Facial Subsampling* revela que numerosos accidentes de tráfico en todo el mundo ocurren debido a la fatiga, somnolencia y distracción mientras se conduce. Algunos trabajos sobre la detección automatizada de somnolencia proponen extraer señales fisiológicas del conductor, como ECG, EEG, tasa de variabilidad cardíaca, presión arterial, etc., lo que hace que esas soluciones no sean ideales. Mientras que otros trabajos recientes proponen soluciones basadas en visión por computadora, pero muestran un rendimiento limitado al usar características hechas a mano con técnicas convencionales como Naive Bayes y SVM, o modelos de aprendizaje profundo excesivamente voluminosos que aún tienen un rendimiento bajo.

En este trabajo, se propone una arquitectura de aprendizaje profundo en conjunto que opera sobre características incorporadas de submuestras de ojos y boca junto con una estructura de decisión para determinar la aptitud del conductor. El modelo propuesto en conjunto consta de solo dos módulos InceptionV3 que ayudan a contener el espacio de parámetros de la red. Estos dos módulos realizan respectivamente y exclusivamente la extracción de características de las submuestras de ojos y boca extraídas utilizando el MTCNN de las imágenes faciales. Su salida respectiva se pasa al límite del conjunto utilizando el método de promedio ponderado cuyos pesos se ajustan utilizando el algoritmo de conjunto. La salida de este sistema determina si el conductor está somnoliento o no.

El conjunto de datos de video de referencia NTHU-DDD se utilizó para el entrena-

miento y la evaluación efectivos del modelo propuesto. El modelo estableció una precisión de entrenamiento y validación del 99.65 % y 98.5 % respectivamente, con una precisión del 97.1 % en el conjunto de datos de evaluación, lo cual es significativamente mayor que las precisiones logradas por los modelos propuestos en trabajos recientes sobre este conjunto de datos [5].

A robust and efficient EEG-based drowsiness detection system using different machine learning algorithms El artículo titulado *A robust and efficient EEG-based drowsiness detection system using different machine learning algorithms* aborda la problemática de los accidentes de vehículos en rutas largas, frecuentemente causados por conductores somnolientos. Esto se debe principalmente a la falta de un sistema que mida el estado de alerta del conductor. Un sistema de detección de fatiga preciso y eficiente podría notificar al conductor para interrumpir su viaje, ayudando a evitar accidentes y a tomar decisiones correctas.

Este trabajo tiene como objetivo detectar la somnolencia de los conductores utilizando una herramienta de software potente, desarrollada inicialmente capturando y procesando señales de electroencefalografía (EEG). En esta investigación, se aplicaron diferentes algoritmos de aprendizaje automático a las señales EEG de doce sujetos para medir su rendimiento. En el primer paso, todos los datos registrados para todos los sujetos se segmentaron en épocas de un segundo. Las señales cerebrales se etiquetaron como alerta o somnoliento para cada época.

Antes de aplicar los algoritmos de aprendizaje automático a la señal segmentada, se introdujo un paso de preprocesamiento para extraer las características relevantes. Los algoritmos aplicados fueron: Naive Bayes (Análisis Discriminante Lineal Diagonal), SVM (Funciones Lineales y de Base Radial), KNN y Random Forest. Al capturar señales de solo tres electrodos, se encontró que utilizar más de un clasificador llevó a la mayor precisión del 100 % para todos los sujetos considerados en este estudio.

En general, este sistema basado en EEG desarrollado detecta la somnolencia y la pérdida de enfoque de los conductores en tiempo real con alta precisión, lo que lo convierte en una opción práctica y confiable para aplicaciones en tiempo real [6].

Machine Learning Trained Drowsiness Detection using Gyroscope on a Microcontroller El artículo titulado *Machine Learning Trained Drowsiness Detection using*

Gyroscope on a Microcontroller aborda la problemática de la conducción somnolienta, que en los últimos años ha resultado en 328,000 incidentes anuales. La somnolencia es generalmente un signo de fatiga, y los conductores que la experimentan pueden tener episodios temporales de microsueño que pueden ser fatales mientras conducen.

Aunque algunos fabricantes de automóviles han implementado sistemas de detección de somnolencia, estos solo se han aplicado a vehículos de gama alta. Por lo tanto, es necesario crear un dispositivo de detección de somnolencia que sea portátil y accesible para cualquier persona, independientemente de su origen.

El sistema propuesto está diseñado utilizando un microcontrolador e implementado como una MLP de pequeño tamaño y alta precisión, portado a un archivo de encabezado utilizando Tensorflow Lite. Con esto, el dispositivo es capaz de detectar la somnolencia del usuario utilizando el giroscopio para monitorear el movimiento de la cabeza y utiliza un LED y un zumbador para alertar al usuario o a otros sobre la somnolencia en tiempo real [7].

Machine learning based driver monitoring system: A case study for the Kayoola EVS El artículo titulado *Machine learning based driver monitoring system: A case study for the Kayoola EVS* aborda el creciente problema de los accidentes de tráfico debido a la alta densidad de tráfico. Encontrar soluciones para reducir los accidentes de tráfico y mejorar la seguridad vial se ha convertido en una prioridad para Kiira Motors Corporation, una empresa automotriz estatal de Uganda. La compañía busca desarrollar sistemas inteligentes de asistencia al conductor para su producto de entrada al mercado, el autobús Kayoola EVS.

Se ha desarrollado un sistema de monitoreo del conductor basado en aprendizaje automático que monitorea la somnolencia del conductor y envía una alarma en caso de detectar somnolencia, con el objetivo de reducir los accidentes relacionados con la somnolencia. El sistema consiste en una cámara posicionada de tal manera que rastrea el rostro del conductor. La cámara está conectada a una minicomputadora Raspberry Pi que realiza los cálculos y análisis, y cuando se detecta somnolencia, se activa una alarma.

El comportamiento peligroso del conductor, incluida la distracción y la fatiga, ha sido reconocido durante mucho tiempo como el principal factor contribuyente en los accidentes de tráfico. Este artículo presenta el desarrollo de un sistema de monitoreo del

conductor para el autobús eléctrico Kayoola para abordar el aumento de los accidentes de tráfico. El sistema de monitoreo del conductor basado en aprendizaje automático está diseñado para ser no intrusivo y operar en tiempo real de manera continua [8].

Role of Deep Learning in Improving the Performance of Driver Fatigue Alert System El artículo titulado *Role of Deep Learning in Improving the Performance of Driver Fatigue Alert System* tiene como objetivo monitorear el estado del conductor. Cuando se detecta fatiga causada por actitudes diferentes a las del hábito normal de conducción, el sistema advierte al conductor que debe interrumpir su viaje, ayudando así a evitar accidentes de tráfico.

El sistema analiza en tiempo real cualquier cambio en las características de los ojos y la boca del conductor y emite una advertencia cuando es necesario. El sistema propuesto contiene varias etapas para detectar la fatiga del conductor. Primero, en la etapa de preprocesamiento, se mejoran los fotogramas, se determina el rostro y se recortan los ojos y la boca del conductor. Luego, en la etapa de extracción de características, se procesan las características de cada fotograma utilizando métricas como EAR y MAR.

Finalmente, se presentaron dos enfoques de clasificación y se realizó una comparación entre ellos. En el primer enfoque, se aplicaron cuatro clasificadores tradicionales: Análisis Discriminante Lineal Diagonal (LDA), Máquina de Vectores de Soporte Lineal (SVM), KNN y Random Forest. Los resultados muestran que dos clasificadores, KNN y Random Forest, obtuvieron la mayor precisión promedio del 91.94 % para todos los sujetos presentados en este artículo.

En el segundo enfoque, se aplicó un modelo de CNN de aprendizaje profundo; el modelo Resnet-50". Los resultados también muestran que el modelo de aprendizaje profundo propuesto obtuvo una alta precisión promedio del 96.3889 % para los mismos datos. En general, la somnolencia y la pérdida de enfoque de los conductores se detectaron con alta precisión utilizando el sistema basado en procesamiento de imágenes desarrollado, lo que lo hace práctico y confiable para aplicaciones en tiempo real [9].

Deep convolutional network based real time fatigue detection and drowsiness alertness system El artículo titulado *Deep convolutional network based real time fatigue detection and drowsiness alertness system* discute las técnicas innovadoras, métodos eficientes y avances recientes en el campo de la detección de somnolencia y

fatiga. En este modelo propuesto, se planea una amplia aplicación en el campo de la inteligencia artificial, definiendo los fundamentos de la interacción humano-computadora, el reconocimiento de expresiones faciales y la determinación de la fatiga-somnolencia del conductor.

Esta investigación describe una estrategia eficiente y efectiva de tres fases para detectar la somnolencia. En estas tres fases, se utiliza el algoritmo de Viola-Jones para detectar rasgos faciales. Una vez identificada la cara, se realiza la detección del bostezo y el seguimiento, segmentando la piel para que el sistema se vuelva invariante a la iluminación, enfocándose en los componentes cromáticos basados en la piel y rechazando la mayoría de los fondos de imágenes que no son rostros.

El seguimiento del color de los ojos y la detección del bostezo se llevan a cabo mediante la coincidencia de plantillas con el coeficiente de correlación. Los vectores de características basados en cada una de las fases anteriores se concatenan y se obtiene un resultado binario. El análisis de los fotogramas sucesivos se clasifica en estados de fatiga y no fatiga utilizando un modelo de CNN. Si el tiempo en estado de fatiga supera el umbral, el sistema emitirá una alarma. Además, métricas como EAR y MAR permiten cuantificar la apertura de ojos y boca para mejorar la detección de somnolencia [10].

A Hybrid Driver Fatigue and Distraction Detection Model Using AlexNet Based on Facial Features El artículo titulado *A Hybrid Driver Fatigue and Distraction Detection Model Using AlexNet Based on Facial Features* aborda el problema del aumento de accidentes de tráfico debido a la fatiga y la falta de sueño de los conductores, especialmente en ciudades modernas con un estilo de vida acelerado. Estos accidentes se han convertido en una de las principales causas de lesiones y muertes entre jóvenes y niños. La detección temprana de los síntomas de fatiga puede prevenir estos accidentes.

Por esta razón, los autores proponen y comparan dos modelos basados en AlexNet (CNN) para detectar comportamientos de fatiga en los conductores, utilizando la posición de la cabeza y los movimientos de la boca como medidas de comportamiento. Se utilizaron dos enfoques diferentes. El primer enfoque es el aprendizaje por transferencia, específicamente, el ajuste fino de AlexNet, lo que permitió aprovechar lo que el modelo ya había aprendido sin desarrollarlo desde cero. El nuevo modelo entrenado fue capaz de predecir los comportamientos de somnolencia de los conductores.

El segundo enfoque utiliza AlexNet para extraer características entrenando las capas superiores de la red. Estas características se redujeron utilizando la factorización de matrices no negativas (NMF) y se clasificaron con un clasificador de máquina de vectores de soporte (SVM). Los experimentos mostraron que el modelo de aprendizaje por transferencia propuesto alcanzó una precisión del 95.7 %, mientras que el modelo basado en la extracción de características y SVM tuvo un mejor rendimiento, con una precisión del 99.65 %. Ambos modelos se entrenaron en un conjunto de datos simulado de detección de somnolencia del conductor de NTHU [11].

Estudio	Metodología	Fortalezas	Debilidades
A Real-time Fatigue Detection System using Multi-Task Cascaded CNN Model	Uso de CNN para detectar somnolencia mediante movimientos faciales (PUC, bostezo, movimiento de cabeza)	Alta precisión (97.5 %), no invasivo, detección en tiempo real	Afectado por condiciones de iluminación y uso de gafas
Real Time Drowsiness Detection System based on ResNet-50	Uso de ResNet-50 (CNN) para analizar expresiones faciales y posición de la cabeza	Alta precisión (98.60 %), uso de aprendizaje por transferencia	Requiere cámaras de alta calidad, puede ser costoso
Driver drowsiness detection using CNN-inspired features and PCA with KNN	Uso de CNN para extracción de características y PCA para reducción de dimensionalidad, clasificador KNN	Alta precisión (100 %), mejora con PCA	Dependencia de imágenes diurnas, puede no ser preciso en condiciones nocturnas
Driver Mental Fatigue Detection Based on Head Posture Using reLU-BiLSTM	Monitoreo de movimientos de la cabeza con XSENS y red neuronal BiLSTM modificada	Alta precisión (99.2 %), sensibilidad (97.54 %), precisión (97.38 %)	Requiere equipo especializado (XSENS), puede ser costoso y complejo de implementar
Intelligent Driver Drowsiness Detection for Traffic Safety Based on Multi CNN Deep Model and Facial Subsampling	Uso de InceptionV3 (CNN) para extracción de características de ojos y boca, promedio ponderado para decisión final	Alta precisión (97.1 %), uso eficiente de parámetros	Complejidad en la implementación, puede requerir ajuste fino de parámetros
A robust and efficient EEG-based drowsiness detection system	Uso de señales EEG y varios algoritmos de aprendizaje automático (SVM, Random Forest)	Alta precisión (100 %), uso de múltiples clasificadores	Requiere equipo EEG, invasivo, puede no ser práctico para uso en vehículos

Cuadro 2.2: Comparación de estudios relacionados sobre detección de fatiga en conductores (Parte 1)

Estudio	Metodología	Fortalezas	Debilidades
Machine Learning Trained Drowsiness Detection using Gyroscope on a Microcontroller	Uso de giroscopio y microcontrolador con MLP	Portátil, accesible, bajo costo	Menor precisión comparado con otros métodos, limitado a movimientos de la cabeza
Machine learning based driver monitoring system: A case study for the Kayoola EVS	Monitoreo de somnolencia con cámara y Raspberry Pi	No intrusivo, operación en tiempo real	Dependencia de calidad de cámara, puede ser afectado por condiciones de iluminación
Role of Deep Learning in Improving the Performance of Driver Fatigue Alert System	Análisis de características de ojos y boca, uso de clasificadores tradicionales (KNN, Random Forest) y CNN	Alta precisión (96.39 %), uso de múltiples clasificadores	Requiere procesamiento de imágenes en tiempo real, puede ser costoso
Deep convolutional network based real time fatigue detection and drowsiness alertness system	Uso de CNN para detección de fatiga mediante análisis de expresiones faciales y detección de bostezo	Alta precisión (96.54 %), uso de técnicas avanzadas de procesamiento de imágenes	Complejidad en la implementación, puede requerir ajuste fino de parámetros
A Hybrid Driver Fatigue and Distraction Detection Model Using AlexNet Based on Facial Features	Uso de AlexNet (CNN) para detección de fatiga y distracción mediante características faciales	Alta precisión (99.65 %), uso de aprendizaje por transferencia	Requiere procesamiento de imágenes en tiempo real, puede ser costoso

Cuadro 2.3: Comparación de estudios relacionados sobre detección de fatiga en conductores (Parte 2)

Discusión

En esta sección se discuten las metodologías implementadas en los estudios relacionados, agrupándolas según los tipos de parámetros que monitorean para la detección de fatiga en conductores.

- **Parámetros de comportamiento:** Estos métodos se basan en la observación de señales visuales y expresiones faciales del conductor. Estudios como *A Real-time*

Fatigue Detection System using Multi-Task Cascaded CNN Model implementan una CNN en cascada para extraer características de movimientos faciales como PERCLOS, bostezos y movimientos de cabeza, logrando una detección en tiempo real con alta precisión. De manera similar, *Real Time Drowsiness Detection System based on ResNet-50* utiliza aprendizaje por transferencia con ResNet-50 para analizar imágenes faciales y posición de la cabeza, también alcanzando resultados precisos. Estos métodos son no invasivos, pero su desempeño puede verse afectado por condiciones de iluminación y el uso de gafas.

- **Parámetros vehiculares:** Este enfoque monitorea el comportamiento del conductor a través de sensores integrados en el vehículo. Por ejemplo, *Machine Learning Trained Drowsiness Detection using Gyroscope on a Microcontroller* utiliza un giroscopio conectado a un microcontrolador y una MLP para detectar movimientos de la cabeza asociados a somnolencia. Los sistemas basados en parámetros vehiculares son portátiles y de bajo costo, pero su precisión puede ser menor y limitada a ciertos tipos de movimiento.

- **Parámetros fisiológicos:** Estas metodologías miden cambios físicos directos en el conductor. El estudio *A robust and efficient EEG-based drowsiness detection system* emplea señales EEG procesadas mediante varios algoritmos de aprendizaje automático (SVM, KNN, Random Forest) para clasificar el estado de alerta, logrando precisiones cercanas al 100 %. Aunque altamente precisos, requieren equipo especializado y pueden ser invasivos, dificultando su uso práctico en vehículos.

Por otro lado, se puede observar que, independientemente de la metodología utilizada, la mayoría de los estudios comparten etapas fundamentales para la detección de fatiga, que son determinantes para garantizar la precisión y confiabilidad del sistema:

- **Preprocesamiento:** Captura y limpieza de imágenes o señales; segmentación de regiones de interés (rostro, ojos, boca, cabeza, ventanas de EEG o giroscopio).

- **Detección de rostro:**

  - **Haar Cascade:** Algoritmo clásico basado en características de Haar y clasificadores en cascada, rápido para rostros frontales.
  - **ResNet-50:** Red profunda usada como detector de rostros, más precisa en distintas condiciones de iluminación y ángulos.

- **Extracción de características:** Métricas visuales (EAR, MAR, PUC, movimientos de cabeza, expresiones); fisiológicas (EEG, frecuencia cardíaca); vehiculares (patrones de conducción).
- **Selección/reducción de características:** Técnicas como PCA o NMF para conservar solo las más relevantes.
- **Clasificación/modelado:** Modelos de aprendizaje automático y profundo: CNN, ResNet, BiLSTM, SVM, KNN, Random Forest; aprendizaje por transferencia en algunos casos.

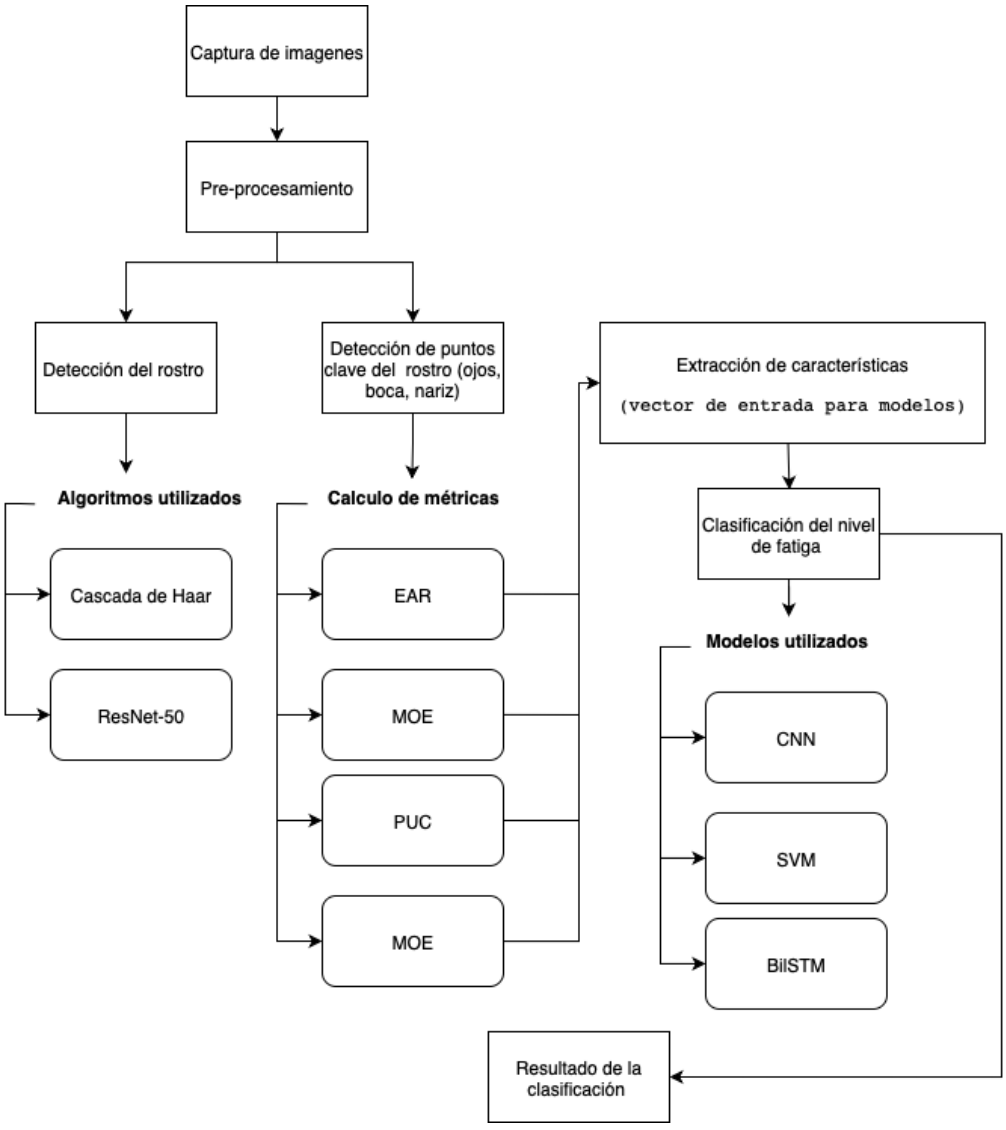


Figura 2.2: Diagrama de flujo general de los métodos de detección de fatiga.

2.6.1. Hallazgos de la literatura revisada

- La identificación precisa del rostro es esencial para la extracción confiable de métricas faciales como EAR, MAR y PUC, lo que permite detectar signos de fatiga con alta precisión.
- Algoritmos clásicos como Haar son rápidos y eficientes para rostros frontales, mientras que modelos profundos como ResNet-50 ofrecen mayor tolerancia a variaciones de iluminación y ángulo.
- Las métricas visuales, combinadas con modelos de aprendizaje profundo (CNN, BiLSTM) o tradicional (SVM, KNN), permiten clasificaciones precisas de fatiga en tiempo real.
- La reducción de características mediante técnicas como PCA o NMF optimiza el procesamiento sin perder información relevante.
- Aunque existen métodos alternativos basados en parámetros fisiológicos o vehiculares, los sistemas visuales ofrecen un equilibrio entre precisión, no invasividad y aplicabilidad práctica.
- La integración de aprendizaje por transferencia y redes profundas facilita la adaptación del sistema a distintos conductores y condiciones de conducción.

3. Marco Teórico

Fatiga

Conducir un vehículo es una actividad que exige la integración simultánea de procesos sensoriales, cognitivos y motores. El conductor debe percibir y procesar información del entorno —como la señalización, el comportamiento de otros vehículos y las condiciones de la vía—, tomar decisiones rápidas y ejecutar acciones precisas para mantener el control del vehículo. Este proceso continuo requiere un nivel sostenido de atención, coordinación y tiempo de reacción, apoyado por un estado fisiológico y mental óptimo. Sin embargo, factores como la privación de sueño, la alteración de los ritmos circadianos o la exposición prolongada a condiciones monótonas pueden afectar estas capacidades, generando un deterioro progresivo del rendimiento cognitivo y físico. Cuando estas alteraciones alcanzan cierto umbral, se manifiestan en forma de fatiga, la cual constituye un factor de riesgo crítico en la seguridad vial.

La fatiga es un estado fisiológico y/o psicológico caracterizado por la disminución de la capacidad para mantener un nivel óptimo de rendimiento, ya sea físico o mental. En el contexto de la conducción, este estado compromete la atención, el tiempo de reacción y la capacidad de toma de decisiones, aumentando significativamente el riesgo de siniestros.

3.1.1. Características de la fatiga

La fatiga puede clasificarse en dos grandes categorías:

- **No patológica:** De carácter transitorio, provocada por causas comunes como la falta de sueño, el esfuerzo físico prolongado o el estrés. Suele revertirse con descanso adecuado.
- **Patológica:** De carácter crónico, vinculada a enfermedades, trastornos psicológicos o condiciones médicas que afectan de manera sostenida el rendimiento y la percepción de energía.

Desde un enfoque subjetivo, la literatura distingue tres dimensiones principales:

- **Fatiga I:** Asociada a somnolencia y enlentecimiento general de las respuestas.

- **Fatiga II:** Relacionada con la dificultad para mantener la concentración y la atención sostenida.
- **Fatiga III:** Vinculada a una sensación de desconexión o desintegración personal.

En el ámbito de la seguridad vial, la **Fatiga I** es la más relevante, ya que sus manifestaciones pueden ser evaluadas mediante indicadores físicos y tecnológicos, lo que permite su detección y monitoreo.

3.1.2. Temas biológicos asociados

La fatiga, especialmente la relacionada con la conducción, está influenciada por diversos procesos biológicos y fisiológicos:

Ritmos circadianos

El ciclo biológico de vigilia y sueño regula la alerta y el rendimiento cognitivo. Conducir en horas de baja activación circadiana (por ejemplo, de madrugada) incrementa la probabilidad de somnolencia.

Privación de sueño

Dormir menos de las horas necesarias provoca acumulación de fatiga y deterioro progresivo de las funciones cognitivas y motoras. Estudios equiparan la conducción con privación severa de sueño a conducir bajo los efectos del alcohol.

Microsueños

Episodios breves de pérdida de conciencia (de 1 a 10 segundos) que pueden ocurrir sin previo aviso. Durante un microsueño, el conductor no procesa estímulos visuales ni controla el vehículo.

Procesos neurofisiológicos

La fatiga afecta la actividad de áreas cerebrales como la corteza prefrontal (toma de decisiones) y el tálamo (procesamiento sensorial), reduciendo la capacidad de respuesta ante estímulos imprevistos.

Factores fisiológicos

Cambios en la frecuencia cardíaca, la respiración y la temperatura corporal pueden acompañar la fatiga y servir como indicadores indirectos de su presencia.

Manifestaciones visuales

Incluyen parpadeo lento, cierre ocular parcial o total, apertura bucal frecuente (bostezos) y cambios en la CM, todos ellos observables mediante visión por computador.

Aprendizaje Automático

El aprendizaje automático (*Machine Learning*, ML) es una rama de la inteligencia artificial que desarrolla algoritmos capaces de aprender patrones a partir de datos y realizar predicciones o clasificaciones sin ser programados explícitamente para cada caso [18]. En el ámbito de la detección de fatiga, el ML se ha aplicado tanto a variables visuales —como indicadores derivados de la apertura ocular, el movimiento de la boca o el parpadeo— como a señales fisiológicas, incluyendo el EEG y el ECG [19].

Diversos estudios han demostrado que los algoritmos de ML pueden alcanzar altos niveles de precisión en la clasificación de estados de fatiga cuando se entrenan con datos relevantes y bien etiquetados. Por ejemplo, se han reportado exactitudes superiores al 90 % en la identificación de fatiga mental a partir de señales fisiológicas utilizando clasificadores como SVM, XGBoost y arquitecturas de aprendizaje profundo [19]. De forma similar, revisiones recientes destacan que técnicas como máquinas de vectores de soporte, redes neuronales artificiales y métodos de ensamble son ampliamente utilizadas en este campo, tanto con datos fisiológicos como con características visuales [18].

Tipos de aprendizaje automático

El aprendizaje automático se clasifica comúnmente en tres categorías principales:

- **Aprendizaje supervisado (Supervised Learning):** El modelo se entrena con un conjunto de datos etiquetados, donde cada ejemplo incluye las características de entrada y la salida deseada. El objetivo es aprender una función que relacione ambas para predecir la salida de nuevos datos. Ejemplos: clasificación de imágenes, predicción de series temporales.

- **Aprendizaje no supervisado (Unsupervised Learning):** El modelo trabaja con datos no etiquetados y busca descubrir patrones o estructuras ocultas, como agrupamientos o relaciones entre variables. Ejemplos: *clustering*, reducción de dimensionalidad.

- **Aprendizaje por refuerzo (Reinforcement Learning):** Un agente aprende a tomar decisiones mediante la interacción con un entorno, recibiendo recompensas o penalizaciones según sus acciones, con el objetivo de maximizar la recompensa acumulada. Ejemplos: control de robots, juegos.

Dentro del aprendizaje supervisado se incluyen algoritmos ampliamente utilizados en tareas de clasificación y regresión, como las **máquinas de vectores de soporte** (SVM), los **perceptrones multicapa** (MLP), los **bosques aleatorios** (Random Forest), el algoritmo de **k-vecinos más cercanos** (KNN), el clasificador **Naïve Bayes** (Naive Bayes) y los **árboles de decisión**.

Antes de profundizar en los enfoques de aprendizaje automático, es útil presentar los fundamentos de la visión por computador y la representación de imágenes, dado que gran parte de las características consideradas provienen de señales visuales.

Visión por computador

La visión por computador (*Computer Vision*) es una rama de la inteligencia artificial que estudia cómo dotar a las máquinas de la capacidad de interpretar y comprender el contenido de imágenes y secuencias de video. Su objetivo es automatizar tareas que el sistema visual humano realiza de forma natural, como identificar objetos, reconocer personas, interpretar expresiones faciales o seguir el movimiento de elementos en una escena.

En el contexto de la seguridad vial, la visión por computador se ha aplicado en sistemas para la detección de peatones y carriles, el reconocimiento de señales de tránsito y el monitoreo del estado de los conductores. Este último caso resulta especialmente relevante para la detección de fatiga, ya que permite analizar regiones faciales como los ojos, la boca y la frente para inferir el nivel de alerta a partir de indicadores visuales.

Desde un punto de vista técnico, una **imagen digital** puede entenderse como una matriz de tamaño $H \times W$ compuesta por píxeles, donde cada píxel almacena información sobre la intensidad luminosa y, en el caso de imágenes en color, su composición

cromática. La **resolución** (número de píxeles en ancho y alto) y la **profundidad de color** (bits por canal) determinan el nivel de detalle y el rango dinámico de la imagen.

Esta representación matricial permite aplicar transformaciones y extraer características relevantes para el aprendizaje automático. En particular, el análisis de imágenes en diferentes **espacios de color** —como RGB o HSV— facilita la separación de información de tono, saturación y brillo, lo que resulta útil para estudiar variaciones cromáticas en regiones específicas, como la coloración de las mejillas. De esta manera, la visión por computador actúa como el puente entre la captura de información visual y el procesamiento mediante algoritmos de aprendizaje automático, proporcionando las características necesarias para que los modelos, tanto superficiales como profundos, puedan clasificar el estado de somnolencia.

Modelo RGB. El espacio **RGB** representa el color como combinación aditiva de rojo (R), verde (G) y azul (B). Es el modelo nativo de la mayoría de sensores y pantallas, pero no separa explícitamente color e iluminación, por lo que cambios de luz afectan simultáneamente a los tres canales.

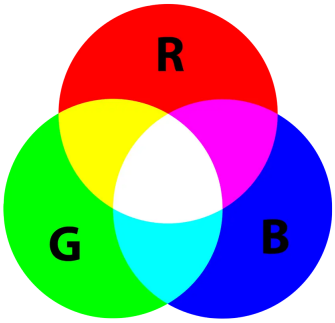


Figura 3.1: Representación esquemática del modelo de color RGB y la formación de colores secundarios mediante mezcla aditiva.

Espacios de color alternativos. Para desacoplar mejor color y brillo se emplean transformaciones como:

- **HSV (Hue, Saturation, Value): H** (tono) codifica el color percibido (p. ej., rojizo, verdoso), **S** la saturación y **V** la luminosidad. La separación de tono y brillo facilita comparar tonalidades bajo variaciones moderadas de iluminación.

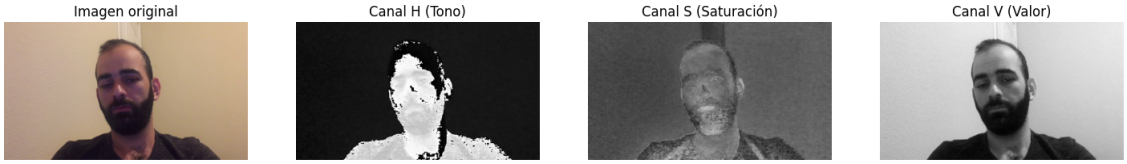


Figura 3.2: Representación del modelo de color HSV y sus tres canales: H (tono), S (saturación) y V (valor o luminosidad).

- **YCbCr / Lab:** Separan luminancia de crominancia (YCbCr) o aproximan componentes perceptuales (Lab), favoreciendo un análisis más robusto del color.

Figura 3.3: Separación de canales en los espacios de color YCbCr y CIELab: luminancia (Y o L) y crominancia (Cb/Cr o a/b), lo que permite analizar el color de forma independiente a la iluminación.

Relevancia del canal H. El canal **H** concentra el *tono* y tiende a ser menos sensible a la iluminación global que **V**, lo que lo hace conveniente para analizar variaciones cromáticas relativas en regiones de piel.

Operaciones básicas. De forma general, el procesamiento incluye: (i) *conversión de espacio de color* (p. ej., RGB→HSV), (ii) *selección de regiones de interés*, (iii) *suavizado y normalización* y (iv) *cálculo de descriptores* (estadísticas de canales, histogramas, proporciones) que actuarán como características para los modelos.

Estos fundamentos de representación y análisis de imágenes permiten distinguir con mayor claridad entre modelos que dependen de características previamente diseñadas y aquellos que aprenden representaciones directamente de los datos crudos.

Enfoques en aprendizaje automático

El aprendizaje automático puede dividirse en dos enfoques [18]:

- **Métodos superficiales:** Modelos que operan sobre características previamente definidas (ingeniería de características), como métricas geométricas y cromáticas calculadas a partir de los datos.
- **Métodos profundos:** Modelos que aprenden representaciones jerárquicas directamente de los datos crudos (por ejemplo, píxeles de una imagen), como las redes neuronales convolucionales.

Dado que en el enfoque superficial el desempeño depende de la calidad de las características diseñadas, a continuación se describen los modelos supervisados más utilizados y los esquemas que ilustran su funcionamiento.

3.2.1. Aprendizaje Automático Superficial

En este enfoque, el rendimiento del modelo depende en gran medida de la calidad y relevancia de las características de entrada. Tales características se derivan de transformaciones de las imágenes (p. ej., mediciones geométricas y cromáticas en RGB/HSV) u otros resúmenes estadísticos de los datos originales.

A continuación, se presenta una descripción general de cada algoritmo supervisado considerado, junto con un esquema ilustrativo de su funcionamiento.

Máquinas de Vectores de Soporte (SVM)

Las SVM son clasificadores que buscan el hiperplano que mejor separa las clases en el espacio de características, maximizando el margen entre los datos de distintas clases. Son eficaces en espacios de alta dimensión y pueden utilizar funciones *kernel* para manejar relaciones no lineales.

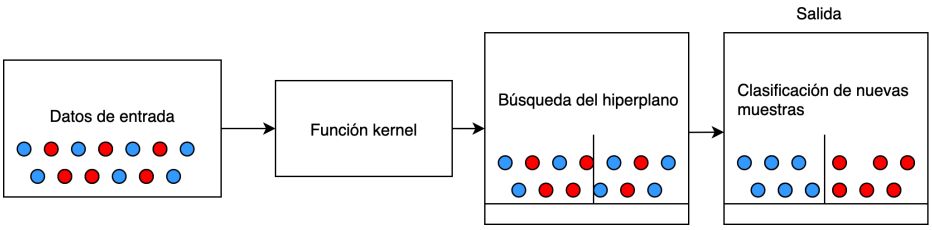


Figura 3.4: Esquema simplificado del funcionamiento de una SVM: transformación de los datos mediante una función *kernel*, búsqueda del hiperplano óptimo y clasificación de nuevas muestras.

Perceptrón Multicapa (MLP)

El MLP es un tipo de red neuronal artificial organizada en capas de nodos interconectados: una capa de entrada que recibe las variables iniciales, una o varias capas ocultas que procesan la información mediante combinaciones lineales y funciones de activación no lineales, y una capa de salida que entrega la predicción o clasificación final. Este modelo es capaz de aprender relaciones complejas entre las variables de entrada y la salida deseada, lo que lo hace aplicable a una amplia variedad de tareas de clasificación y regresión.

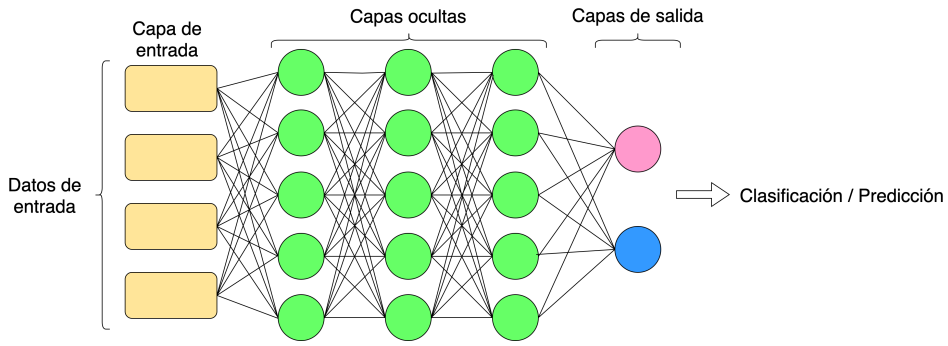


Figura 3.5: Esquema de un perceptrón multicapa: los datos ingresan por la capa de entrada, se transforman en las capas ocultas mediante pesos y funciones de activación, y la capa de salida genera la predicción o clasificación final.

Bosques Aleatorios (Random Forest)

El Random Forest es un método de ensamble que combina múltiples árboles de decisión entrenados sobre subconjuntos aleatorios de datos y características. Cada árbol genera una predicción individual y, posteriormente, los resultados se combinan mediante un mecanismo de votación (para clasificación) o promedio (para regresión). Este enfoque

permite reducir el sobreajuste y mejorar la capacidad de generalización del modelo.

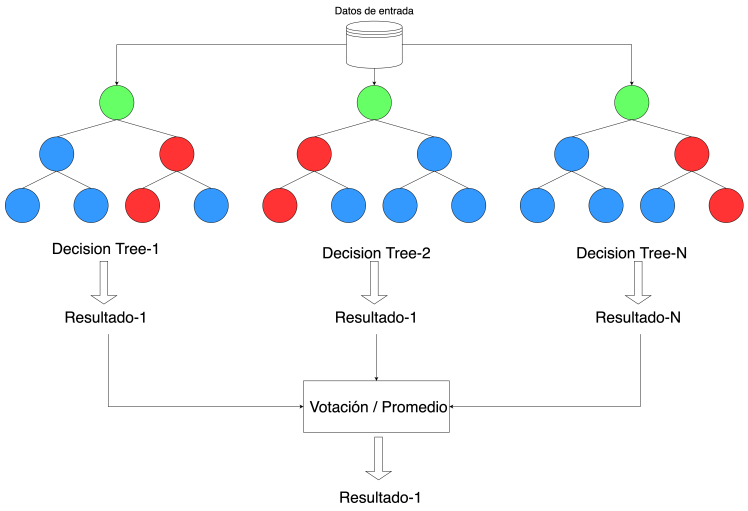


Figura 3.6: Esquema de un bosque aleatorio: los datos de entrada se utilizan para entrenar múltiples árboles de decisión, cuyos resultados se combinan mediante votación o promedio para obtener la predicción final.

K-Vecinos Más Cercanos (KNN)

El KNN es un algoritmo de clasificación que asigna una muestra a la clase mayoritaria entre sus k vecinos más cercanos en el espacio de características. Su simplicidad y efectividad lo hacen útil en una amplia variedad de problemas, siempre que las características estén bien definidas y normalizadas.

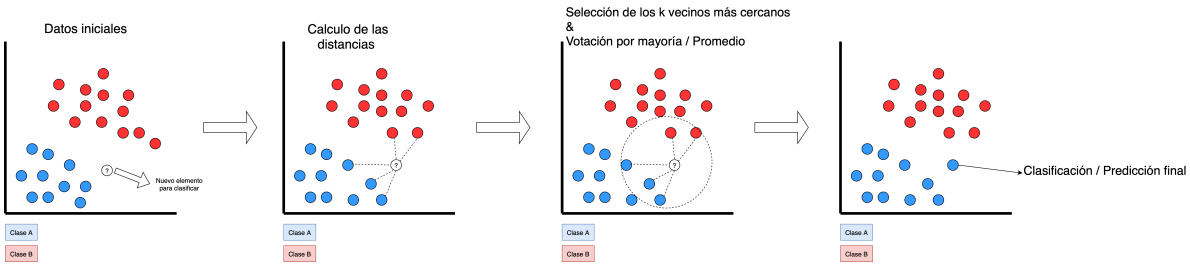


Figura 3.7: Esquema del algoritmo KNN: a partir de un nuevo elemento, se calculan las distancias a los puntos existentes, se seleccionan los k vecinos más cercanos y se asigna la clase por mayoría de votos o promedio.

Naïve Bayes

El Naive Bayes es un clasificador probabilístico que aplica el teorema de Bayes para estimar la probabilidad de que una muestra pertenezca a cada clase posible, asumien-

do independencia condicional entre las características. El proceso parte de los datos de entrada, a partir de los cuales se calculan las probabilidades de observar esas características dado cada clase. Luego, mediante el teorema de Bayes, se combinan estas probabilidades con la probabilidad previa de cada clase para obtener la probabilidad posterior. Finalmente, la muestra se asigna a la clase con la probabilidad posterior más alta. A pesar de la simplificación que supone la independencia condicional, este método suele ofrecer buenos resultados en problemas con datos limitados y presencia moderada de ruido.

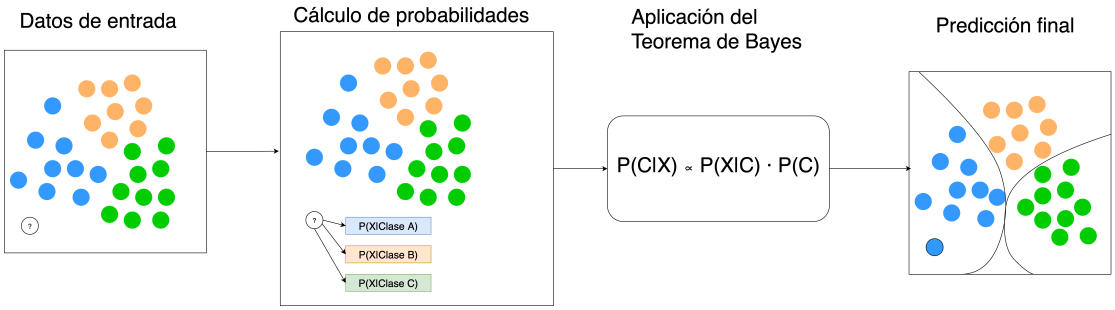


Figura 3.8: Esquema del clasificador Naive Bayes: a partir de los datos de entrada, se calculan las probabilidades condicionales, se aplica el teorema de Bayes y se determina la clase final según la mayor probabilidad posterior.

Árboles de Decisión

Un árbol de decisión es un modelo de aprendizaje supervisado que representa un conjunto de reglas de decisión en forma jerárquica. A partir de los datos de entrada, el modelo divide el espacio de características en regiones más pequeñas mediante preguntas o condiciones secuenciales sobre los atributos. Cada nodo interno corresponde a una prueba sobre una característica, cada rama representa el resultado de esa prueba y cada nodo hoja asigna una clase o un valor de salida. Su estructura intuitiva facilita la interpretación y permite manejar tanto variables numéricas como categóricas.

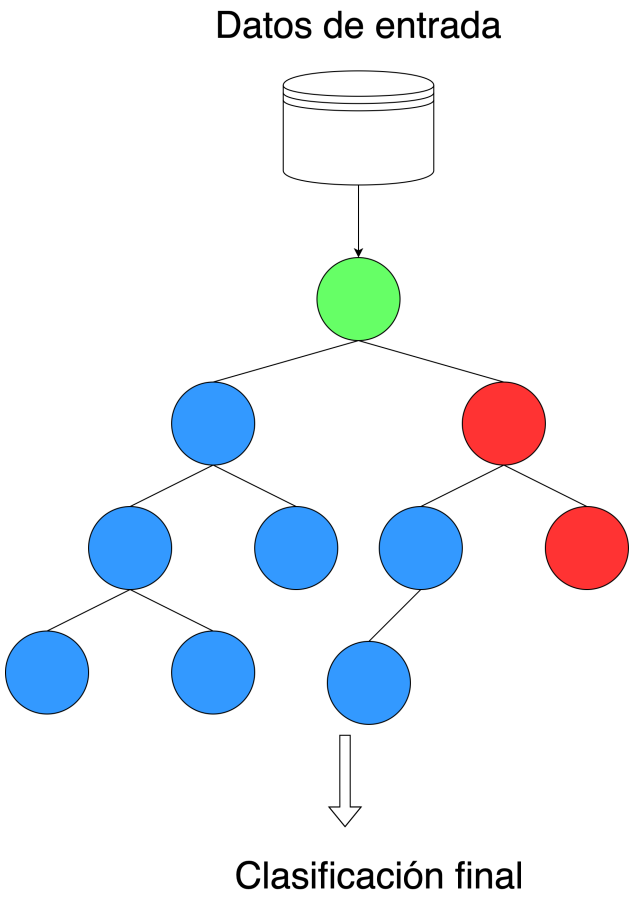


Figura 3.9: Esquema de un árbol de decisión: los datos de entrada se procesan a través de nodos de decisión que dividen el espacio de características, hasta llegar a nodos hoja que determinan la clasificación final.

Estos modelos superficiales permiten evaluar la capacidad discriminativa de las métricas faciales diseñadas, sirviendo como referencia frente a los métodos profundos que se describen en la siguiente sección.

Extracción de Características: EAR, MAR, PUC, MOE y Análisis de las Mejillas

En el contexto del aprendizaje automático, una **característica** (*feature*) es una variable medible que describe un aspecto relevante de los datos de entrada. En el aprendizaje supervisado, las características constituyen la base sobre la cual los modelos aprenden a establecer relaciones entre las entradas y las salidas esperadas. La calidad, relevancia y representatividad de estas características influyen directamente en el rendimiento del modelo, ya que determinan la cantidad de información útil disponible para la tarea de

clasificación o regresión.

La **extracción de características** es el proceso mediante el cual se identifican, calculan o transforman atributos relevantes a partir de los datos originales, con el fin de representar la información de manera compacta y discriminativa. Este paso es crucial porque permite reducir el ruido, disminuir la dimensionalidad y resaltar patrones que faciliten el aprendizaje del modelo.

En la literatura, las características se pueden clasificar de forma genérica en varias categorías, entre las que destacan:

- **Características estadísticas:** derivadas de medidas como media, varianza, desviación estándar, curtosis o asimetría.
- **Características geométricas:** relacionadas con formas, distancias, ángulos o proporciones.
- **Características cromáticas:** basadas en la distribución y variación de colores.
- **Características de textura:** que describen patrones repetitivos o rugosidad en una superficie.
- **Características en el dominio de la frecuencia:** obtenidas mediante transformadas como Fourier o Wavelet.

En el estudio de la detección de fatiga mediante visión por computadora, un enfoque común consiste en definir y analizar métricas geométricas y cromáticas que permitan cuantificar movimientos y variaciones faciales relevantes. Entre estas métricas se encuentran el EAR, MAR, PUC, MOE y la CM, las cuales se han propuesto para describir patrones asociados al parpadeo, la apertura bucal y la oclusión ocular, así como cambios en la tonalidad de la piel. Dichos indicadores se consideran potencialmente útiles para estimar el nivel de somnolencia de un individuo a partir de señales visuales.

A continuación, se presenta una descripción detallada de cada una de estas métricas, incluyendo su definición, fundamento teórico y relevancia en el análisis de la fatiga.

3.3.1. EAR: Eye Aspect Ratio

El EAR mide la apertura de los ojos a partir de proporciones geométricas entre puntos faciales clave. Su cálculo utiliza seis puntos por ojo: - Ojo izquierdo: P_{36} a P_{41}. - Ojo derecho: P_{42} a P_{47}.

$$EAR_{izq} = \frac{||P_{37} - P_{41}|| + ||P_{38} - P_{40}||}{2 \cdot ||P_{36} - P_{39}||} \tag{3.1}$$

$$EAR_{der} = \frac{||P_{43} - P_{47}|| + ||P_{44} - P_{46}||}{2 \cdot ||P_{42} - P_{45}||} \tag{3.2}$$

donde $|| \cdot ||$ representa la distancia euclidiana entre dos puntos.

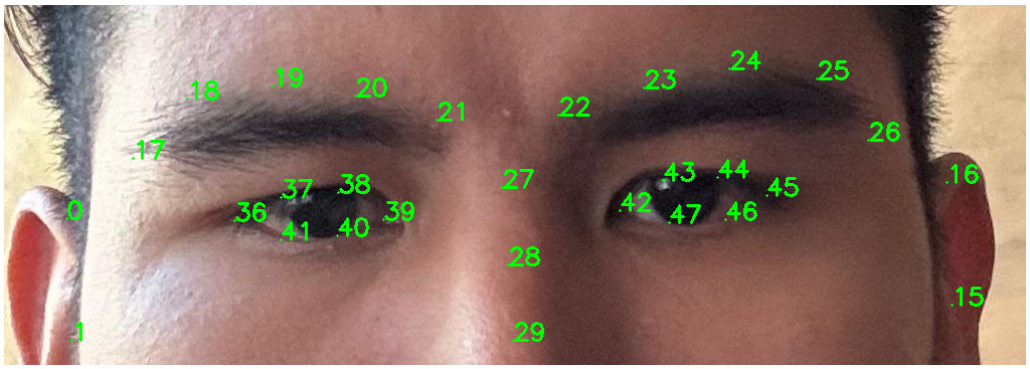


Figura 3.10: Puntos faciales utilizados para el cálculo del EAR.

El EAR se emplea principalmente en dos tareas:

- **Detección de parpadeos:** un descenso rápido y temporal del valor indica un cierre momentáneo de los ojos.
- **Medición de somnolencia:** valores bajos y sostenidos reflejan cierre ocular prolongado, asociado a fatiga.

3.3.2. MAR: Mouth Aspect Ratio

El MAR cuantifica la apertura de la boca mediante proporciones geométricas de puntos faciales clave. Su cálculo emplea 12 puntos del modelo de 68 puntos de Dlib, distribuidos alrededor de la boca: P_{48} a P_{60}.

$$MAR = \frac{||P_{51} - P_{59}|| + ||P_{53} - P_{57}|| + ||P_{52} - P_{58}||}{3 \cdot ||P_{48} - P_{54}||} \tag{3.3}$$

donde P_i representa la coordenada del punto facial i, y $|| \cdot ||$ denota la distancia euclidiana entre dos puntos. El numerador promedia tres distancias verticales (51–59, 53–57, 52–58), mientras que el denominador corresponde a la distancia horizontal entre los puntos 48 y 54.

El MAR se utiliza principalmente para:

Figura 3.11: Puntos faciales utilizados para el cálculo del MAR.

- **Detección de bostezos:** un incremento significativo refleja apertura amplia de la boca, asociado a somnolencia.
- **Medición de fatiga:** valores elevados y sostenidos indican apertura prolongada de la boca, útil como señal complementaria al EAR.

3.3.3. PUC: Percentage of Unit Contact

El PUC mide la apertura ocular a partir de la relación entre la suma de las distancias verticales del ojo y la distancia horizontal entre los extremos. Se diferencia del EAR en que no promedia las distancias verticales en el numerador, lo que lo hace más sensible a variaciones pequeñas en la apertura del ojo.

$$PUC = \frac{||P_{37} - P_{41}|| + ||P_{38} - P_{40}||}{||P_{36} - P_{39}||} \tag{3.4}$$

Para el ojo derecho se utilizan los puntos 43–47, 44–46 y 42–45, y el valor final puede calcularse como el promedio de ambos ojos.

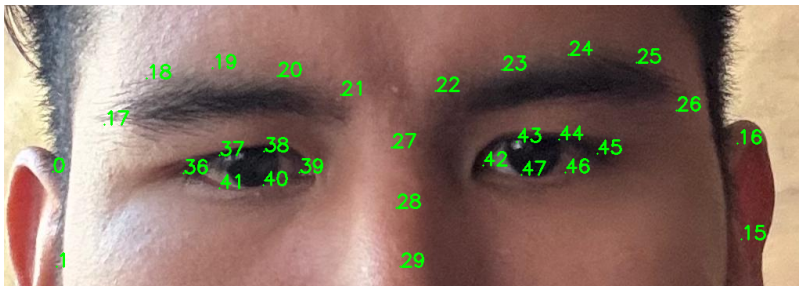


Figura 3.12: Puntos faciales utilizados para calcular el PUC.

El PUC se utiliza principalmente para:

- **Detección de parpadeos:** al ser más sensible que el EAR, permite capturar cambios rápidos en la apertura ocular.
- **Evaluación de somnolencia:** valores bajos y sostenidos reflejan cierres prolongados de ojos, asociados a fatiga o distracción.

3.3.4. MOE: Measure of Eye Occlusion

El MOE es una métrica derivada del EAR que cuantifica el grado de oclusión ocular. Su valor es inversamente proporcional a la apertura del ojo, por lo que permite resaltar cierres parciales o totales.

$$MOE = \frac{1}{EAR}, \quad \text{cuando } EAR > 0 \tag{3.5}$$

El MOE se utiliza principalmente para:

- **Identificar somnolencia severa:** resalta momentos de cierre ocular prolongado, en los que el EAR presenta valores cercanos a cero.
- **Complementar otras métricas:** combinado con EAR y PUC, permite caracterizar mejor los parpadeos lentos o micro-sueños.

3.3.5. Coloración de mejillas

El análisis de las mejillas se basa en la extracción de información cromática a partir del modelo de color HSV (*Hue-Saturation-Value*), utilizando específicamente el componente *Hue* (H), que representa el tono de color. Este enfoque permite detectar variaciones sutiles en la coloración de la piel, potencialmente asociadas a cambios fisiológicos como el enrojecimiento.

Para este análisis, se definen dos regiones de interés (ROIs) en el rostro: una en la mejilla izquierda y otra en la mejilla derecha, ubicadas en el cuadrante inferior de cada lado de la cara (Figura 3.13). Estas zonas son sensibles a cambios en el flujo sanguíneo y enrojecimiento, lo que las convierte en áreas relevantes para estudios de variaciones fisiológicas.

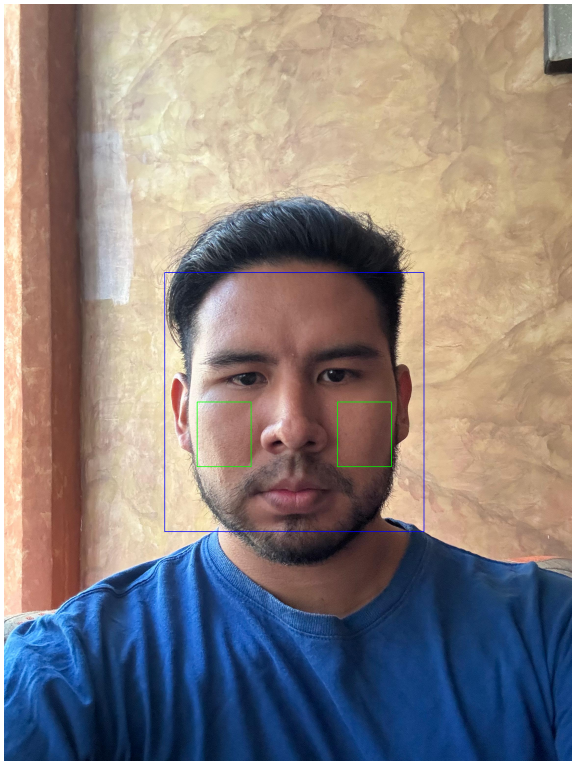


Figura 3.13: Regiones de interés (ROIs) en las mejillas para el análisis de color.

El cálculo consiste en extraer los valores del componente H de cada píxel dentro de la ROI y promediarlos para obtener un valor representativo de tono:

$$H_{\text{mejilla}} = \frac{1}{N} \sum_{i=1}^{N} H_i \tag{3.6}$$

donde H_i es el valor del componente Hue del píxel i y N es el número total de píxeles en la región.

El análisis de las mejillas permite identificar cambios en la coloración cutánea que pueden estar relacionados con variaciones en el flujo sanguíneo, la temperatura o el estado fisiológico general, constituyendo así una métrica complementaria a las basadas en geometría facial previamente descritas.

En conjunto, las métricas EAR, MAR, PUC, MOE y CM constituyen un conjunto de características derivadas de la imagen facial, diseñadas para cuantificar de forma objetiva indicadores asociados a la fatiga y la somnolencia. Estas variables numéricas, obtenidas a partir de la detección de puntos faciales y del análisis cromático de regiones específicas, condensan información relevante sobre la apertura ocular, la oclusión parcial o total de los ojos, la apertura bucal y las variaciones en la CM. Su cálculo permite transformar

datos visuales en entradas estructuradas para modelos de aprendizaje automático, facilitando así el análisis automatizado del estado de alerta.

Reducción de características

Cuando se trabaja con múltiples métricas faciales, es común que algunas de ellas estén correlacionadas entre sí o que aporten información redundante. Esto puede dificultar la interpretación de los resultados y aumentar la complejidad de los modelos de clasificación. Para enfrentar este desafío, se emplean técnicas de reducción de dimensionalidad, cuyo propósito es transformar el conjunto original de variables en un espacio más compacto que conserve la información más relevante para el análisis.

Entre las técnicas más utilizadas se encuentran las siguientes:

Análisis de Componentes Principales (PCA) El PCA es una técnica estadística ampliamente utilizada para reducir la dimensionalidad de un conjunto de datos. Su funcionamiento consiste en transformar las variables originales en un nuevo conjunto de variables llamadas *componentes principales*, que son combinaciones lineales de las variables iniciales. Estas componentes se ordenan de manera que la primera captura la mayor cantidad de variabilidad presente en los datos, la segunda la siguiente mayor, y así sucesivamente.

En términos prácticos, PCA identifica las direcciones en las que los datos presentan mayor dispersión y proyecta la información hacia esos ejes. De esta forma, es posible representar los datos en menos dimensiones sin perder gran parte de la información contenida en el conjunto original. Esto facilita la visualización de patrones, la eliminación de redundancias y la simplificación de los modelos posteriores, al trabajar con un número reducido de variables que concentran la esencia de los datos.

Análisis Discriminante Lineal (LDA) El LDA es una técnica estadística utilizada en contextos de aprendizaje supervisado. A diferencia de PCA, que no considera las etiquetas de clase, LDA aprovecha la información de las categorías a las que pertenecen los datos. Su objetivo principal es encontrar combinaciones lineales de características que maximicen la separación entre clases, de modo que los ejemplos de una misma categoría queden lo más agrupados posible y, al mismo tiempo, se distancien de los ejemplos de otras categorías.

En términos sencillos, LDA proyecta los datos en un espacio de menor dimensión donde las diferencias entre grupos sean más evidentes. Esto no solo facilita la visualización de las clases en problemas multiclase, sino que también puede mejorar el desempeño de los algoritmos de clasificación al proporcionar representaciones más discriminativas. En un problema con tres clases, por ejemplo, LDA puede generar hasta dos dimensiones, lo que permite representar gráficamente la relación entre categorías de manera clara y comprensible.

En conjunto, tanto PCA como LDA permiten simplificar el espacio de variables, ya sea priorizando la variabilidad general de los datos o la separación entre clases. Estas técnicas se entienden aquí como herramientas conceptuales que ayudan a representar la información de manera más clara y manejable, sin necesidad de conservar todas las variables originales.

3.3.6. Aprendizaje Automático Profundo

El aprendizaje profundo (*Deep Learning*) es una rama del aprendizaje automático que utiliza arquitecturas de redes neuronales con múltiples capas para aprender representaciones jerárquicas de los datos. A diferencia de los métodos superficiales, que dependen de características previamente definidas, los modelos profundos pueden extraer automáticamente patrones relevantes directamente de los datos crudos, como los píxeles de una imagen.

En el ámbito de la visión por computador, las redes neuronales convolucionales (CNN) son la base de muchos modelos de detección y reconocimiento, ya que permiten identificar patrones espaciales y jerárquicos en las imágenes.

YOLO

YOLO (*You Only Look Once*) es un algoritmo de detección de objetos que procesa imágenes de forma rápida, identificando simultáneamente la ubicación y la clase de los elementos presentes. Su arquitectura divide la imagen en regiones y predice, para cada una, las probabilidades de pertenencia a distintas clases junto con las coordenadas de los cuadros delimitadores.

En el contexto de la detección de signos de fatiga, la literatura reporta que YOLO puede entrenarse para reconocer patrones faciales asociados a distintos niveles de som-

nolencia (por ejemplo, ojos abiertos, semicerrados o cerrados). Para ello, se requiere un conjunto de imágenes anotadas, en el que cada región de interés esté delimitada y etiquetada con la clase correspondiente.

La anotación de imágenes puede realizarse de forma manual, utilizando herramientas como *LabelImg* o *Roboflow*, o de manera automática mediante algoritmos de detección de rostros como la cascada de Haar, lo cual resulta especialmente útil cuando se dispone de un gran volumen de imágenes. Este proceso de anotación proporciona la información supervisada que el modelo necesita para aprender a identificar dichas señales.

Estudios previos han demostrado que YOLO ofrece un buen equilibrio entre velocidad y precisión, lo que lo hace adecuado para aplicaciones que requieren respuesta rápida, como el monitoreo del estado de alerta en conductores.

Flujo general de entrenamiento En la Figura 3.14 se presenta un flujo general descrito en la literatura para la preparación y entrenamiento de un modelo YOLO. Este diagrama resume las etapas desde la captura de imágenes hasta la exportación del modelo entrenado, incluyendo tanto el etiquetado manual como el etiquetado automático mediante Haar.

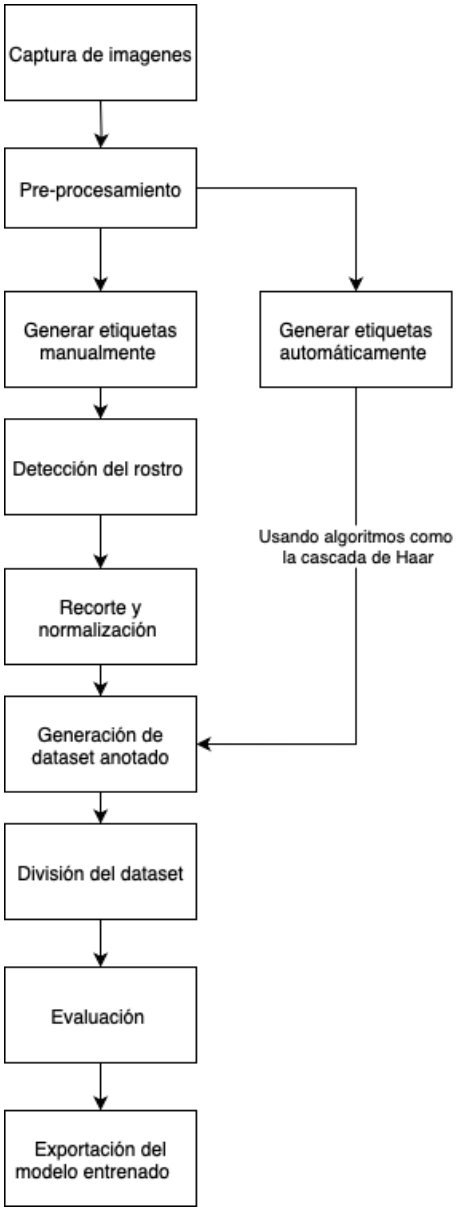


Figura 3.14: Flujo general de preparación y entrenamiento del modelo YOLO.

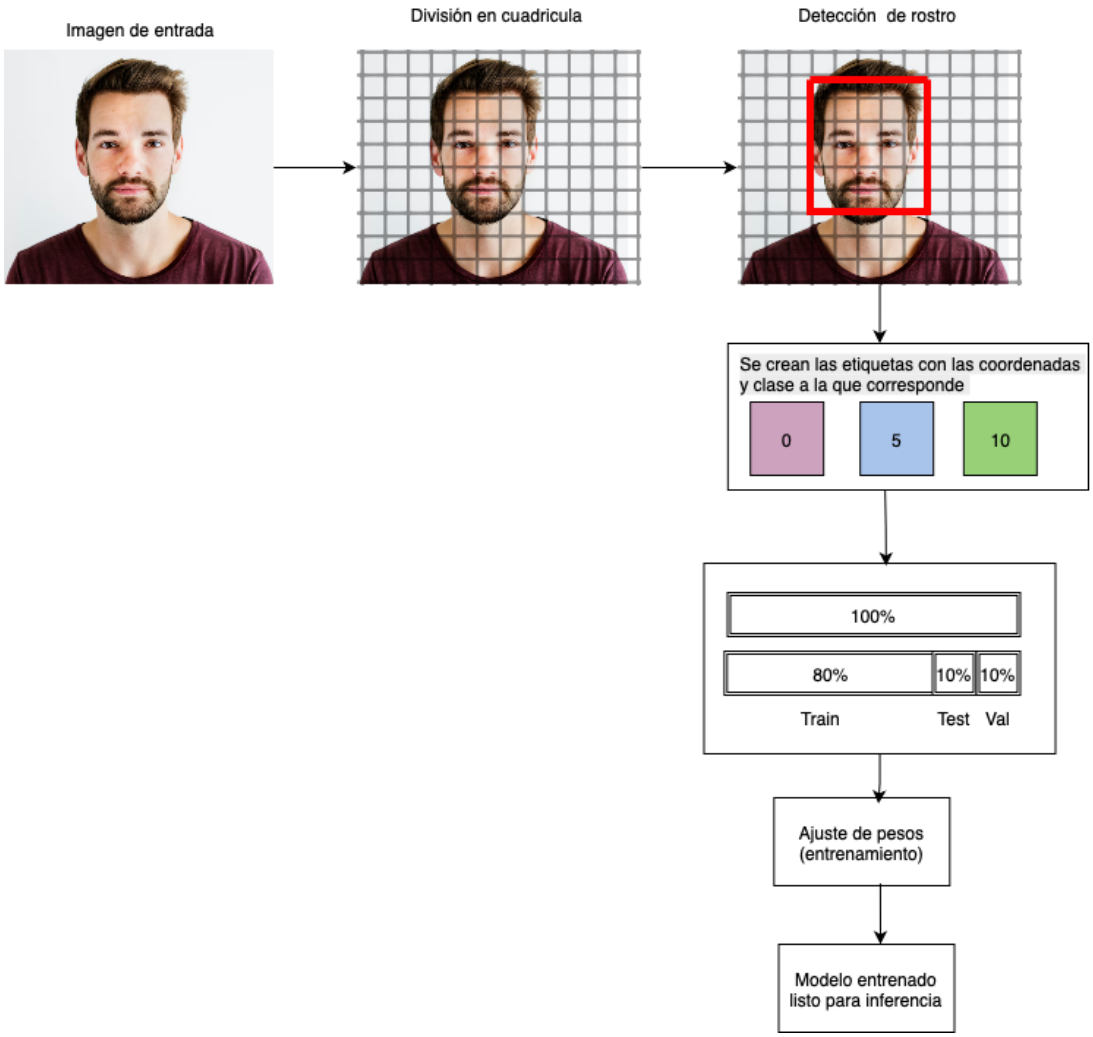


Figura 3.15: Proceso conceptual de preparación y entrenamiento de un modelo de reconocimiento facial.

En la Figura 3.15 se ilustra un flujo conceptual para el entrenamiento de un modelo de reconocimiento facial basado en CNN. El proceso inicia con la **imagen de entrada**, que es segmentada en una cuadrícula para facilitar la detección de rostro. Posteriormente, se generan las etiquetas con las coordenadas y la clase correspondiente, lo que permite estructurar el conjunto de datos. Este conjunto se divide en *train*, *test* y *val*, siguiendo buenas prácticas de validación como StratifiedKFold cuando es necesario. Finalmente, el modelo ajusta sus pesos durante el entrenamiento y produce un modelo listo para la *inferencia*.

Otros enfoques profundos

Además de YOLO, existen otros modelos de detección y clasificación basados en aprendizaje profundo que han sido aplicados en la literatura para el análisis de fatiga y somnolencia:

- **Faster R-CNN:** combina una red de propuestas de regiones con una CNN para lograr detección precisa, aunque con mayor coste computacional.
- **SSD (Single Shot MultiBox Detector):** similar a YOLO en su enfoque de detección en una sola pasada, pero con diferencias en la generación de cajas y escalas.
- **MTCNN (Multi-task Cascaded Convolutional Networks):** especializado en la detección de rostros y puntos faciales, útil como etapa previa a la extracción de características.

Estos enfoques ofrecen alternativas con distintos compromisos entre precisión, velocidad y complejidad, y su elección depende de las necesidades específicas de la aplicación.

4. Metodología

En este capítulo se describe la metodología seguida para el desarrollo de un sistema de clasificación automática de niveles de somnolencia a partir de imágenes faciales. El diseño metodológico se estructura en dos enfoques complementarios, alineados con lo expuesto en el marco teórico:

- **Aprendizaje automático superficial:** basado en la extracción manual de características geométricas y cromáticas del rostro, que posteriormente se utilizan como entrada para algoritmos clásicos de clasificación.
- **Aprendizaje automático profundo:** fundamentado en técnicas de visión por computador que emplean modelos de detección y clasificación basados en redes neuronales convolucionales, específicamente YOLOv8.

Ambos enfoques comparten una base común compuesta por tres etapas iniciales:

1. Selección del conjunto de datos.
2. Extracción de fotogramas a partir de secuencias de video.
3. Preprocesamiento de las imágenes.

A partir de esta base, cada enfoque sigue un flujo de trabajo específico adaptado a su naturaleza:

- En el **aprendizaje automático superficial**, se realiza la detección de puntos faciales clave, el cálculo de métricas (EAR, MAR, PUC, MOE, CM) y la conformación de vectores de características para su procesamiento mediante clasificadores como SVM, MLP, Random Forest, KNN y Naive Bayes.
- En el **aprendizaje automático profundo**, se lleva a cabo la anotación de datos, la división del conjunto en entrenamiento, validación y prueba, y el entrenamiento de un modelo YOLO para la detección y clasificación de patrones faciales asociados a distintos niveles de somnolencia.

En las siguientes secciones se detallan las etapas particulares de cada enfoque, describiendo tanto los procedimientos comunes como las diferencias metodológicas entre el aprendizaje superficial y el profundo.

Aprendizaje Automático Superficial

Este enfoque se basa en la extracción manual de características geométricas y cromáticas del rostro —como EAR, MAR, PUC, MOE y CM— que posteriormente se utilizan como insumo de entrada para algoritmos de aprendizaje automático superficial orientados a tareas de clasificación, tales como SVM, MLP, Decision Tree, Random Forest, KNN y Naive Bayes.

1. **Selección del conjunto de datos:** Se selecciona un conjunto de videos faciales previamente anotados con niveles de somnolencia. Este conjunto se utiliza de forma idéntica en ambos enfoques metodológicos.

2. **Extracción de fotogramas:** Se extraen imágenes individuales (fotogramas) de los videos seleccionados, utilizando una frecuencia de muestreo de 1 fotograma cada 5 (*frame skipping*). Esta tasa permite capturar variaciones faciales relevantes sin perder información valiosa, al tiempo que evita la generación excesiva de datos redundantes.

3. **Preprocesamiento:** Los fotogramas extraídos se someten a un proceso de detección facial mediante el algoritmo de detección por *cascada de Haar* (Haar), con el fin de localizar la región del rostro en cada imagen. Esta etapa permite centrar el análisis en zonas relevantes como la boca, la nariz y los ojos.

4. **Detección de puntos clave:** Sobre las regiones faciales detectadas, se emplea el modelo `shape_predictor_68_face_landmarks.dat` de *dlib* para localizar 68 puntos faciales de referencia, que permiten identificar estructuras específicas como ojos, boca y cejas.

5. **Cálculo de métricas geométricas:** A partir de los puntos clave, se calculan métricas como el EAR, MAR, PUC , MOE y la coloración de las mejillas, que permiten cuantificar patrones faciales asociados al nivel de somnolencia.

6. **Extracción y almacenamiento de características:** Las métricas calculadas se organizan en un archivo `.csv`, estructurado por fotograma y etiquetado con el nivel de somnolencia correspondiente. Este archivo constituye la base para el entrenamiento de modelos.

7. **Entrenamiento de modelos clásicos:** Se entrenan distintos algoritmos de clasificación, incluyendo Random Forest, Decision Tree, KNN, SVM, Naive Bayes y MLP, utilizando las características geométricas como variables de entrada. Se evalúa el rendimiento de cada modelo mediante métricas como precisión, sensibilidad y F1-score.

8. **Evaluación e inferencia:** Se realiza la inferencia sobre el conjunto de prueba y se analizan los resultados obtenidos, estableciendo líneas base de rendimiento para comparar con el enfoque basado en visión por computador.

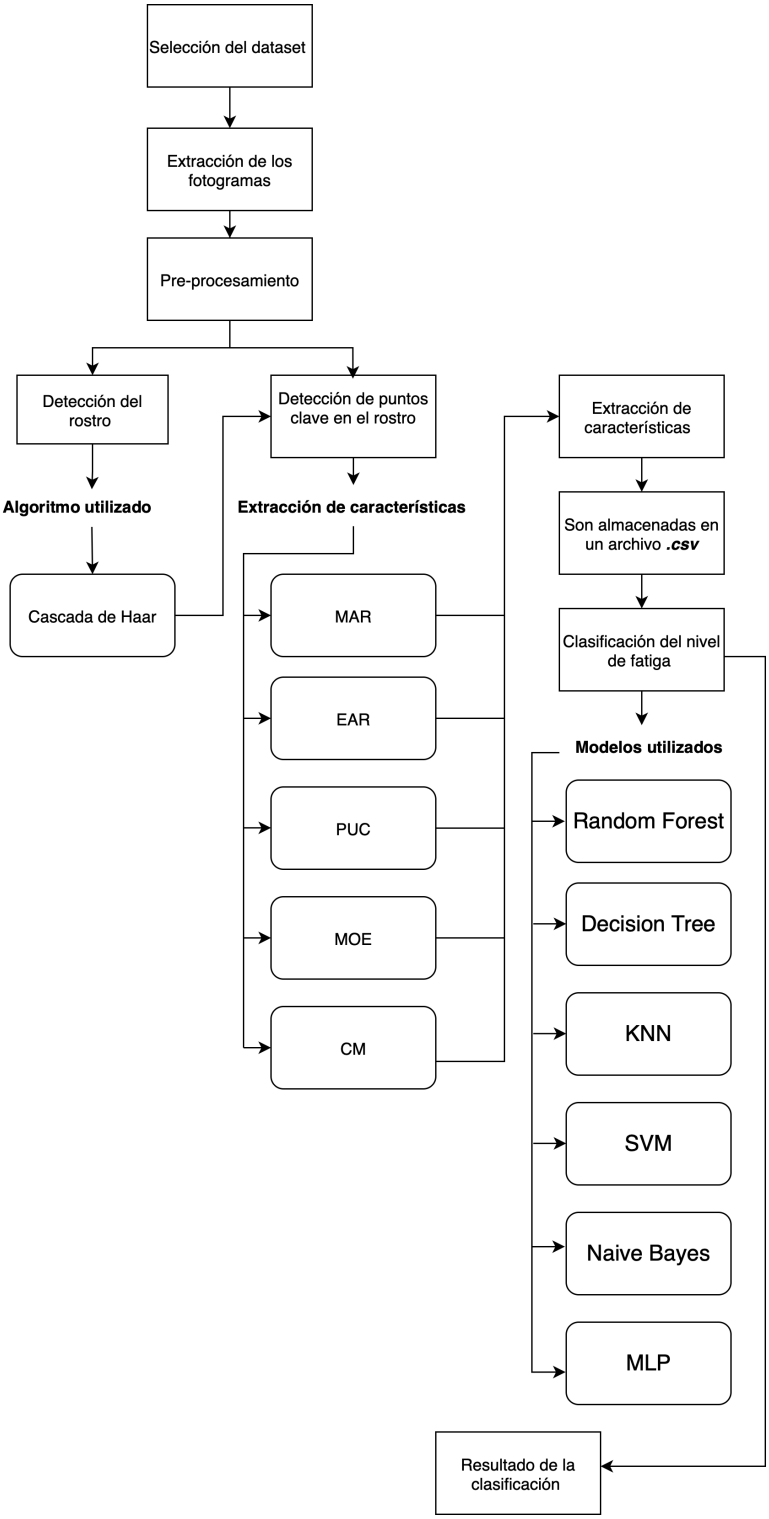


Figura 4.1: Flujo de procesamiento en el enfoque de aprendizaje automático superficial. Se representan las etapas desde la selección del conjunto de datos hasta la clasificación final, incluyendo la detección facial mediante cascada de Haar, el cálculo de métricas geométricas y cromáticas, y el entrenamiento de clasificadores tradicionales.

Aprendizaje Automático Profundo

Este enfoque se fundamenta en técnicas de visión por computador que emplean modelos de detección y clasificación basados en redes neuronales convolucionales. En particular, se utiliza el modelo YOLOv8 para identificar patrones faciales asociados a distintos niveles de somnolencia a partir de imágenes faciales anotadas.

1. **Selección del conjunto de datos:** Se utiliza el mismo conjunto de videos faciales anotados que en el enfoque anterior.

2. **Extracción de fotogramas:** Se extraen imágenes individuales de los videos, siguiendo una frecuencia de muestreo que preserve la variabilidad facial sin generar redundancia.

3. **Preprocesamiento:** Se aplica la cascada de Haar para localizar el rostro en cada fotograma.

4. **Anotación y almacenamiento:** Las coordenadas de las regiones faciales detectadas se anotan siguiendo el formato requerido por YOLO y se almacenan en archivos `.txt`, manteniendo la correspondencia con cada imagen. Esta estructura permite definir explícitamente el *Ground Truth* de cada instancia, indicando no solo la clase, sino también la ubicación espacial del patrón facial asociado a la somnolencia. La decisión de utilizar anotaciones completas en lugar de etiquetas globales se justifica más adelante en función de la precisión requerida por el modelo YOLO.

5. **División del conjunto de datos:** Las imágenes y sus anotaciones se dividen en subconjuntos de entrenamiento (*train*), validación (*val*) y prueba (*test*) en una proporción 80 %-10 %-10 %.

6. **Entrenamiento del modelo YOLOv8:** Se entrena el modelo YOLOv8 para realizar detección y clasificación de niveles de somnolencia directamente sobre las imágenes faciales, aprendiendo patrones visuales relevantes sin extracción manual de características.

7. **Evaluación e inferencia:** Se realiza la inferencia sobre el conjunto de prueba y se aplican técnicas de interpretación como Grad-CAM para validar que el modelo esté tomando decisiones coherentes basadas en regiones faciales significativas.

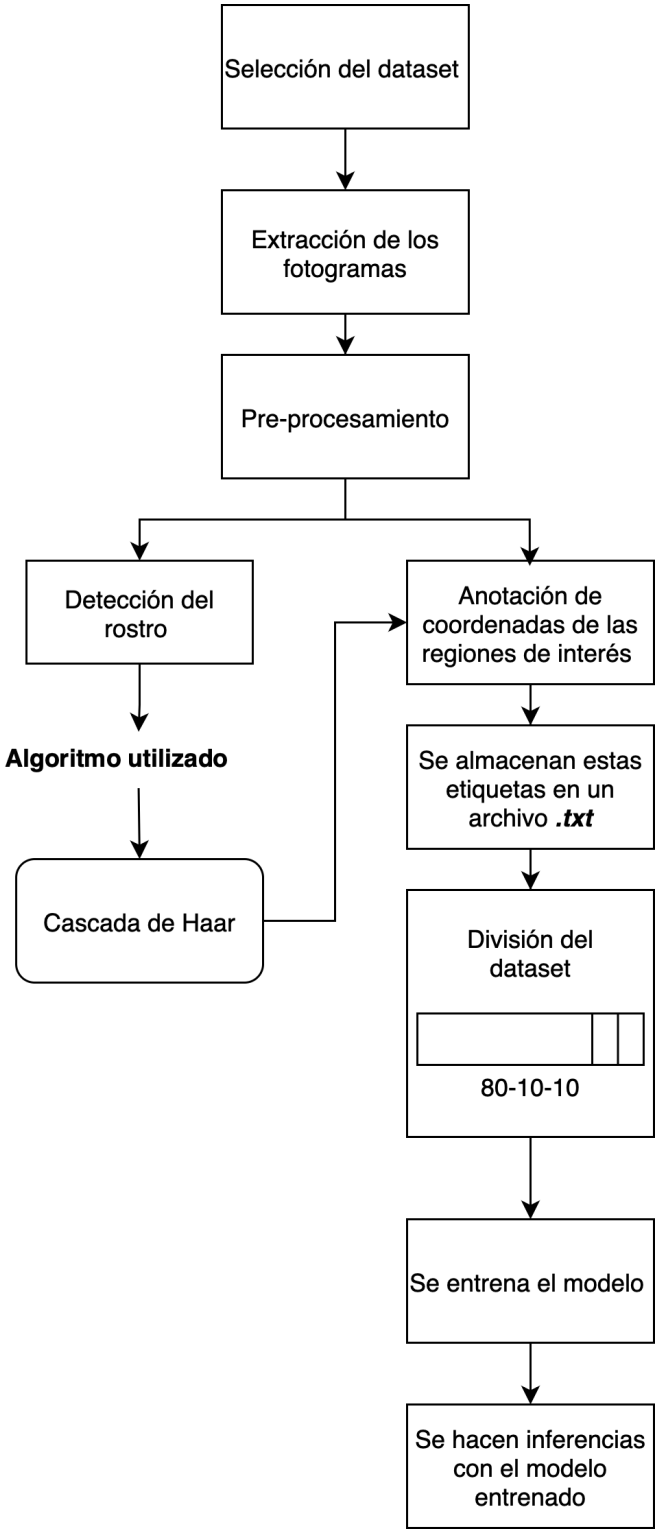


Figura 4.2: Flujo de trabajo del enfoque basado en visión por computador utilizando YOLOv8. Se muestran las etapas desde la selección del conjunto de datos hasta la inferencia final, incluyendo la detección facial, anotación y entrenamiento supervisado.

Nota: Las etapas de *selección del conjunto de datos*, *extracción de fotogramas* y *prepro-*

cesamiento se realizan una sola vez y son comunes a ambos enfoques. Esto asegura que ambos métodos partan de la misma base de datos procesada, garantizando condiciones de comparación justas y evitando redundancia en el flujo de trabajo.

Selección del Conjunto de Datos

En el desarrollo de sistemas de visión artificial orientados a la detección de somnolencia, existen múltiples conjuntos de datos disponibles en la literatura. Sin embargo, muchos de ellos presentan restricciones de acceso, ya que suelen ser creados por grupos de investigación en colaboración con entidades médicas o industriales, lo que limita su disponibilidad para la comunidad científica. Esta situación ha motivado la creación de bases de datos públicas que permitan avanzar en el estudio de la somnolencia en condiciones reales, especialmente en contextos como la conducción vehicular.

Entre los conjuntos de datos abiertos más relevantes se encuentran:

- **MRL Drowsiness Detection Dataset** [12]: Conjunto de imágenes faciales clasificadas por estado ocular (abierto/cerrado), generado a partir del MRL y CEW dataset. Incluye versiones con hasta 10,000 imágenes tomadas bajo diversas condiciones de iluminación, ángulo y resolución, lo que lo hace útil para tareas de detección ocular y somnolencia.

- **Driver Drowsiness Detection Dataset (DDD)** [13]: Contiene más de 41,000 imágenes faciales etiquetadas para detección de somnolencia. Aunque la página presenta problemas de acceso ocasionales, el volumen y variedad de datos lo convierten en una fuente valiosa para entrenamiento supervisado.

- **Roboflow Drowsiness Detection Dataset** [14]: Dataset anotado para detección en tiempo real con YOLO, compuesto por 1,230 imágenes divididas en subconjuntos de entrenamiento, validación y prueba. Incluye anotaciones en múltiples formatos (YOLOv5–v8, COCO, Pascal VOC, entre otros) y técnicas de aumento como rotación, zoom y ajuste de brillo.

Para este proyecto se utilizó el **UTA Real-Life Drowsiness Dataset (UTA-RLDD)**, un conjunto de datos público desarrollado por la Universidad de Texas en Arlington. Este dataset fue diseñado específicamente para abordar el problema de la detección de somnolencia en múltiples niveles, desde casos extremos hasta expresiones sutiles apenas perceptibles, pero clave para distinguir entre estados de alerta y fatiga.

El conjunto de datos contiene aproximadamente 30 horas de video RGB de 60 participantes, cada uno grabando tres videos que representan diferentes niveles de somnolencia según la **Escala de Somnolencia de Karolinska (KSS)**, un instrumento subjetivo ampliamente utilizado que califica el nivel de somnolencia percibida en una escala del 1 (muy alerta) al 9 (muy somnoliento). Las clases incluidas son:

- **Alerta:** Alta concentración y consciencia (niveles 1–3 en KSS).
- **Baja vigilancia:** Presencia de signos de somnolencia sin esfuerzo consciente por mantenerse despierto (niveles 6–7 en KSS).
- **Somnoliento:** Estado en el que el sujeto lucha activamente por no quedarse dormido.

En total, el dataset incluye 180 videos con una duración promedio de 10 minutos cada uno. Los participantes presentan diversidad en edad, género y características físicas (por ejemplo, uso de gafas o presencia de barba), y las grabaciones fueron realizadas en ambientes reales con variaciones en ángulos, iluminación y fondo, lo que lo convierte en un recurso valioso para aplicaciones prácticas como la detección de somnolencia en conductores.

4.3.1. Justificación de la Elección

El UTA-RLDD fue seleccionado para este proyecto por las siguientes razones:

- **Relevancia para el problema:** Está orientado a detectar somnolencia a partir de señales faciales sutiles, lo cual es esencial para sistemas preventivos en contextos críticos como la conducción.
- **Realismo:** Las condiciones de grabación reflejan escenarios del mundo real, lo que favorece la generalización de los modelos entrenados.
- **Diversidad:** Incluye participantes con distintas características demográficas y físicas, lo que contribuye a reducir sesgos en el entrenamiento.
- **Volumen y accesibilidad:** Con 30 horas de video y tres clases bien definidas, el dataset ofrece suficiente información para entrenar modelos complejos.

Este conjunto de datos constituye la base común para ambos enfoques metodológicos desarrollados en este trabajo: el enfoque tradicional basado en características geométricas y el enfoque de clasificación directa mediante YOLOv8.

4.3.2. Resumen del Dataset UTA-RLDD

Cuadro 4.1: Resumen del conjunto de datos UTA-RLDD

Atributo	Valor
Participantes	60
Videos por participante	3
Total de videos	180
Duración promedio por video	10 minutos
Total de horas de video	$\sim$30 horas
Formato de video	RGB
Clases	Alerta, Baja vigilancia, Somnoliento
Etiquetado	Basado en la escala KSS (1–9)
Entorno de grabación	Ambientes reales con variaciones de iluminación y ángulo
Diversidad	Edad, género, etnia, gafas, barba

4.3.3. Visualización de la estructura y etiquetado del dataset

Para complementar la descripción anterior, en esta sección se ilustra la organización interna del conjunto de datos UTA-RLDD y su esquema de etiquetado. Cada participante cuenta con tres videos etiquetados como `0`, `5` y `10`, que corresponden respectivamente a las clases **Alerta**, **Baja vigilancia** y **Somnoliento**, correspondientes a los niveles de somnolencia definidos por la escala KSS.

Esta nomenclatura facilita la identificación automática de la clase a partir del nombre del archivo y permite un preprocesamiento más eficiente.

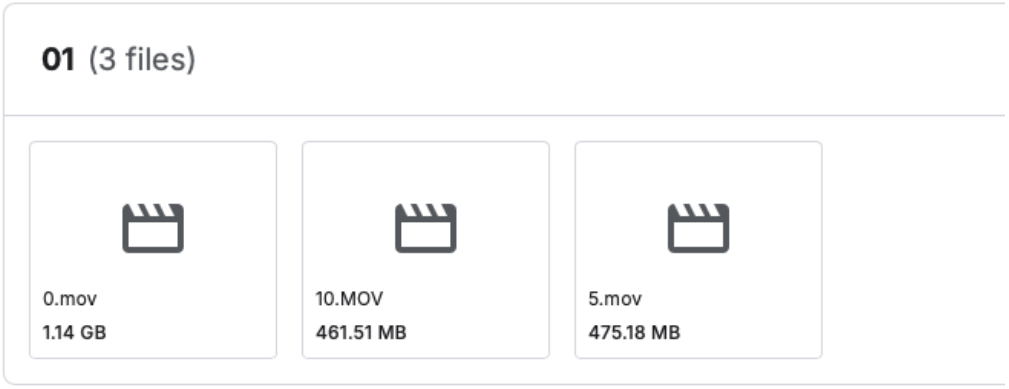


Figura 4.3: Vista general de la estructura de carpetas del UTA-RLDD. Cada carpeta corresponde a un participante y contiene tres videos etiquetados como 0, 5 y 10.

Figura 4.4: Distribución de videos por clase en el dataset, evidenciando la presencia de las tres categorías.

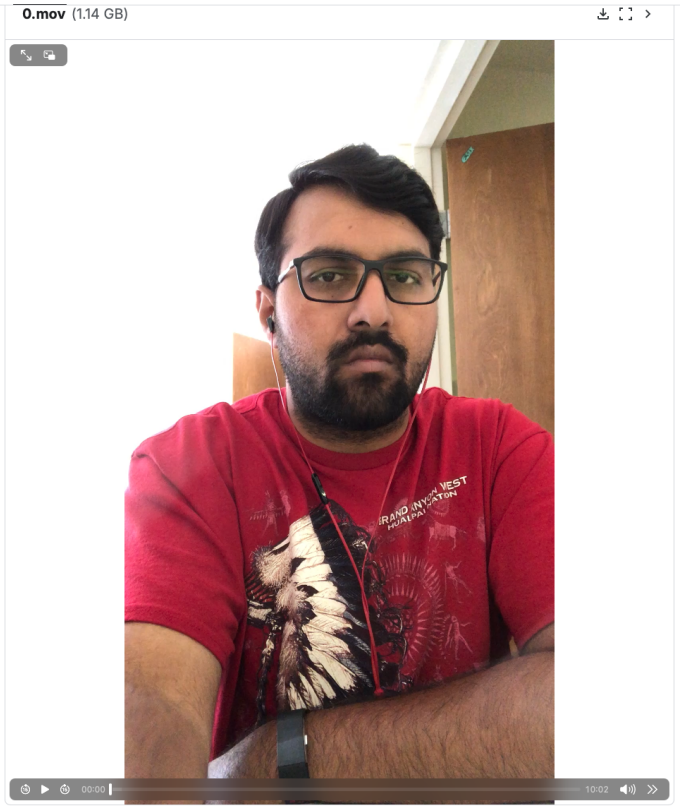


Figura 4.5: Ejemplo de fotograma extraído de un video etiquetado como `0` (clase Alerta).

A partir de esta organización, el siguiente paso consiste en la extracción sistemática de fotogramas, lo que permitirá disponer de representaciones visuales estáticas para el entrenamiento y validación de los modelos.

Extracción de fotogramas

La extracción de fotogramas consiste en descomponer un video en una secuencia de imágenes estáticas que representan instantes específicos de la grabación. Este procedimiento permite transformar datos audiovisuales en representaciones visuales discretas, facilitando su análisis mediante técnicas de visión por computadora y aprendizaje automático.

En el contexto del presente trabajo, la extracción sistemática de fotogramas a partir de los videos del conjunto de datos UTA-RLDD constituye el primer paso para obtener un conjunto de imágenes etiquetadas, que posteriormente se utilizarán para el entrenamiento y la evaluación de los modelos de detección de somnolencia.

El conjunto de datos está organizado en múltiples carpetas, cada una correspondien-

te a un participante. En cada carpeta se encuentran tres videos, nombrados como `0`, `5` y `10`, que representan respectivamente las clases **Alerta**, **Baja vigilancia** y **Somnoliento**, de acuerdo con la escala KSS.

Para el preprocesamiento, se desarrolló un algoritmo personalizado denominado *FrameExtractor*, implementado en Python, que automatiza la lectura de cada video, aplica un muestreo sistemático de un fotograma cada cinco (*frame skipping*) y asigna automáticamente la etiqueta correspondiente según el nombre del archivo de origen. Este enfoque permite capturar variaciones faciales relevantes sin generar redundancia excesiva y garantiza la consistencia en la organización de los datos.

Los fotogramas extraídos se almacenan en carpetas separadas según su clase, de manera que todas las imágenes correspondientes a la clase `0` se guardan en una carpeta `0`, las de la clase `5` en una carpeta `5`, y las de la clase `10` en una carpeta `10`.

Para ilustrar este procedimiento, la Figura 4.6 resume gráficamente el flujo de trabajo implementado, desde la lectura de los videos originales hasta la organización final de los fotogramas por clase.

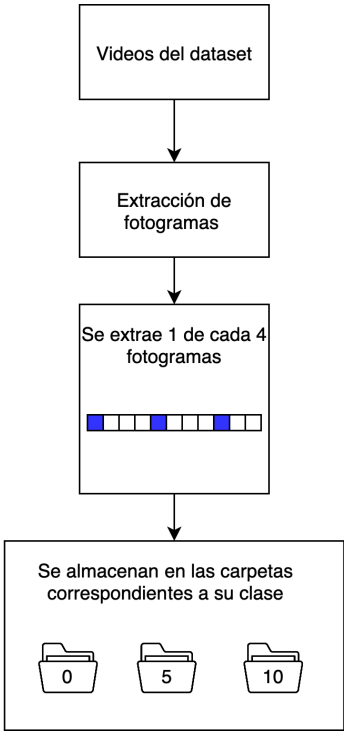


Figura 4.6: Flujo del proceso de extracción de fotogramas: a partir de los videos del dataset, se selecciona 1 de cada 4 fotogramas y se almacenan en las carpetas correspondientes a su clase (`0`, `5` o `10`).

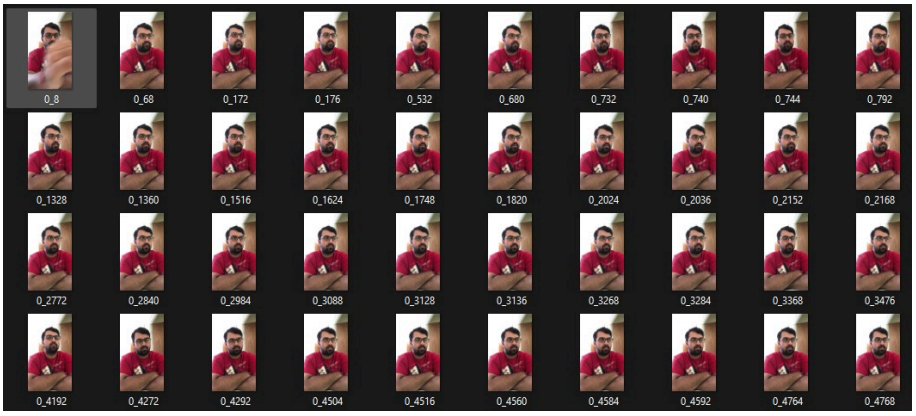


Figura 4.7: Ejemplo de la organización de carpetas con los fotogramas extraídos a partir de los videos del UTA-RLDD.

Fundamentos de la estrategia de extracción

- **Reducción del volumen de datos:** Procesar todos los fotogramas sería computacionalmente costoso y redundante, ya que muchos cuadros consecutivos contienen información similar.
- **Preservación de la información relevante:** Reducir el número de fotogramas no implica una pérdida significativa de información, pues los cambios en expresiones faciales relacionadas con la somnolencia ocurren en una escala temporal suficientemente amplia.
- **Eficiencia en el procesamiento:** Se disminuye el tiempo necesario para almacenar, procesar y entrenar modelos con los datos, optimizando el uso de recursos computacionales.
- **Consistencia y generalización:** Este método asegura un equilibrio entre el volumen de datos y la variabilidad en las imágenes, proporcionando un conjunto representativo para el entrenamiento.

Una vez definida la estrategia de extracción de fotogramas, se procedió a analizar la distribución de las clases en el conjunto de datos resultante, con el fin de verificar su balance y evaluar su idoneidad para el entrenamiento de modelos de detección.

Distribución de las Clases

El análisis de la distribución de las clases permite identificar posibles desbalances en el conjunto de datos que puedan afectar el rendimiento de los modelos de clasifica-

ción. Un conjunto de datos equilibrado contribuye a que el modelo aprenda de manera uniforme las características de cada clase, evitando sesgos hacia las categorías más representadas.

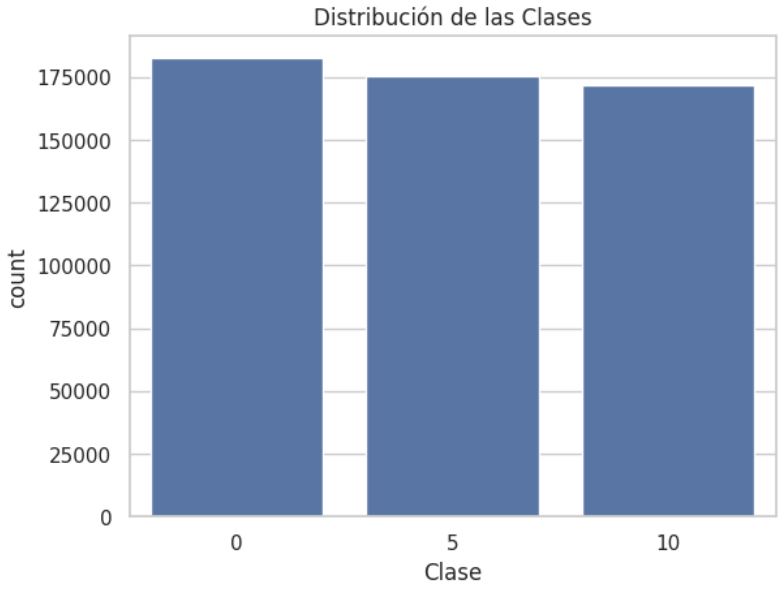


Figura 4.8: Distribución de las Clases

Análisis de la distribución. La Figura 4.8 muestra la frecuencia de instancias para las tres clases presentes en el conjunto de datos: 0, 5 y 10. Se observa que las tres categorías presentan un número de muestras muy similar, con ligeras variaciones. La clase 0 presenta la mayor cantidad de registros, seguida de cerca por la clase 5, mientras que la clase 10 tiene una frecuencia ligeramente menor. Esta distribución balanceada es favorable para el entrenamiento, ya que reduce el riesgo de sesgo hacia una clase específica y facilita que el modelo aprenda patrones representativos de cada categoría.

Preprocesamiento

En esta etapa se aplicó un método de detección facial basado en **cascadas de Haar** en lugar de una arquitectura más compleja como **ResNet-50**. La elección se fundamenta en que las cascadas de Haar presentan un *tiempo de inferencia* promedio de aproximadamente **25–30 ms por fotograma en CPU**, lo que permite procesar decenas de imágenes por segundo sin necesidad de aceleración por GPU. Esta característica no solo reduce los requerimientos de hardware, sino que también disminuye los costos de implementación, ya que el procesamiento puede realizarse en equipos de menor potencia sin sacrificar la velocidad. En contraste, una red convolucional profunda como ResNet-50,

incluso optimizada, puede tardar entre **200 y 300 ms por fotograma en CPU**, es decir, entre 8 y 10 veces más, lo que incrementaría significativamente el tiempo total de preprocesamiento y demandaría recursos computacionales más costosos.

Cascadas de Haar

El algoritmo de cascadas de Haar, propuesto por Viola y Jones, utiliza **características Haar-like** para identificar patrones visuales simples (bordes, líneas, cambios de intensidad) en regiones específicas de la imagen. Estas características se calculan mediante diferencias de sumas de píxeles en áreas rectangulares, y se combinan en una estructura en cascada que permite descartar rápidamente regiones no relevantes, evaluando con mayor detalle solo aquellas que superan las primeras etapas.

Justificación de la elección

En este proyecto, el objetivo del preprocesamiento no es realizar una *clasificación profunda* ni extraer representaciones complejas, sino simplemente **localizar y recortar el rostro** para su análisis posterior en dos flujos metodológicos distintos. Por ello, el uso de cascadas de Haar resulta suficiente, eficiente y coherente con los requerimientos de la tarea, evitando el sobrecoste computacional de modelos más pesados.

Detección y delimitación del rostro

En la Figura 4.9 se muestra un ejemplo del resultado de la detección facial, donde el algoritmo genera un **rectángulo delimitador** alrededor de la región identificada como rostro. Este recorte se utiliza posteriormente para las fases de extracción de características o anotación, dependiendo de la rama metodológica.

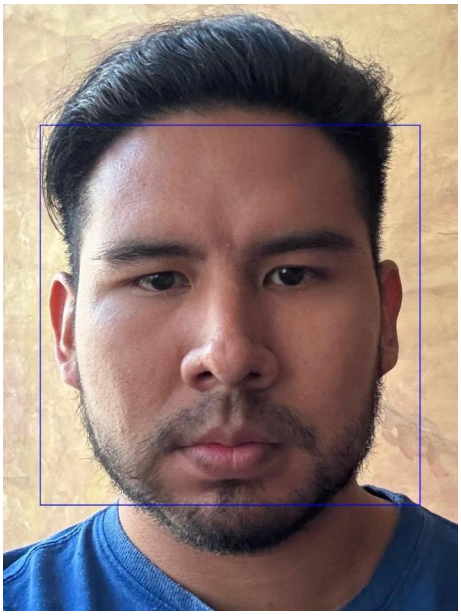


Figura 4.9: Ejemplo de detección facial mediante cascadas de Haar, mostrando el rectángulo delimitador alrededor del rostro.

Transición a los enfoques metodológicos

El preprocesamiento constituye la última etapa común a ambos enfoques. A partir de aquí, el trabajo continúa por dos rutas metodológicas independientes:

- **Enfoque de aprendizaje automático superficial basado en características geométricas**
- **Enfoque de aprendizaje profundo basado en *computer vision* con modelo YOLO**

En la Figura 4.10 se presenta un diagrama comparativo que resume estas rutas. En las siguientes secciones se desarrollará en primer lugar el **enfoque de características aprendizaje automático superficial**.

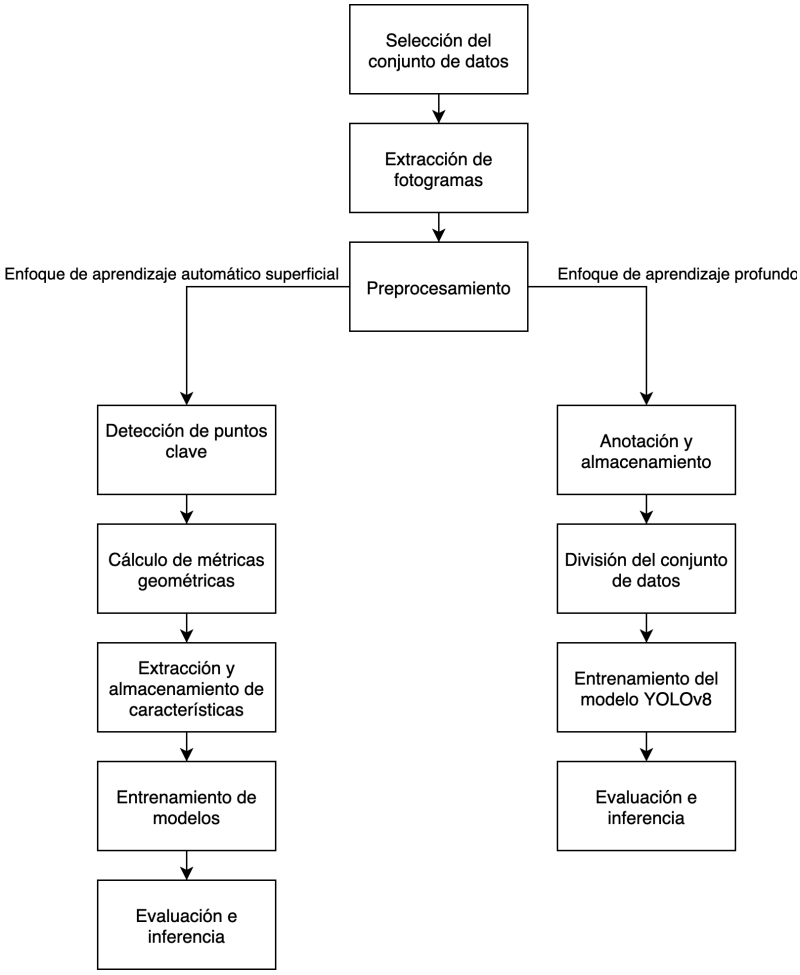


Figura 4.10: Comparación de los flujos metodológicos: características geométricas vs. *computer vision* (YOLO).

Detección de puntos clave

Tras haber localizado el rostro en la etapa de preprocesamiento, el siguiente paso consiste en analizar con mayor detalle su estructura interna. Para ello, se recurre a la **detección de puntos faciales de referencia** o *landmarks*, un procedimiento ampliamente utilizado en visión por computador para mapear la geometría del rostro mediante un conjunto de coordenadas bien definidas. Cada uno de estos puntos actúa como un marcador anatómico virtual que permite identificar la posición exacta de rasgos como ojos, cejas, nariz, boca y contorno facial. Este mapeo no solo facilita la descripción cuantitativa de la forma y disposición de los elementos faciales, sino que también constituye la base para el cálculo de métricas especializadas que, en etapas posteriores, servirán para estimar el nivel de somnolencia.

En el contexto de este proyecto, la detección de *landmarks* es un paso importante

dentro del enfoque de características geométricas, ya que a partir de ellos se obtendrán las métricas EAR, MAR, PUC y MOE. Aunque dichas métricas aún no se calculan en esta fase, la precisión en la localización de los puntos es determinante para garantizar la fiabilidad de los valores que se obtendrán más adelante.

Para esta tarea se empleó el modelo `shape_predictor_68_face_landmarks.dat` de `dlib`, capaz de estimar 68 puntos faciales siguiendo la numeración estándar de la biblioteca. La detección se aplica exclusivamente sobre el área delimitada en el paso anterior, asegurando que el análisis se centre en la zona de interés y evitando procesar información irrelevante.

Procedimiento

El algoritmo implementado realiza las siguientes operaciones:

1. Utiliza la región facial detectada previamente para aplicar el predictor de `dlib`.
2. Estima las coordenadas de los 68 puntos faciales.
3. Dibuja el recuadro delimitador y marca cada punto con su índice correspondiente.
4. Almacena las coordenadas de los puntos en un archivo `.csv` para su posterior análisis.

En esta etapa únicamente se realiza la localización y registro de los puntos faciales. El cálculo de las métricas EAR, MAR, PUC y MOE se desarrolla en la siguiente sección.

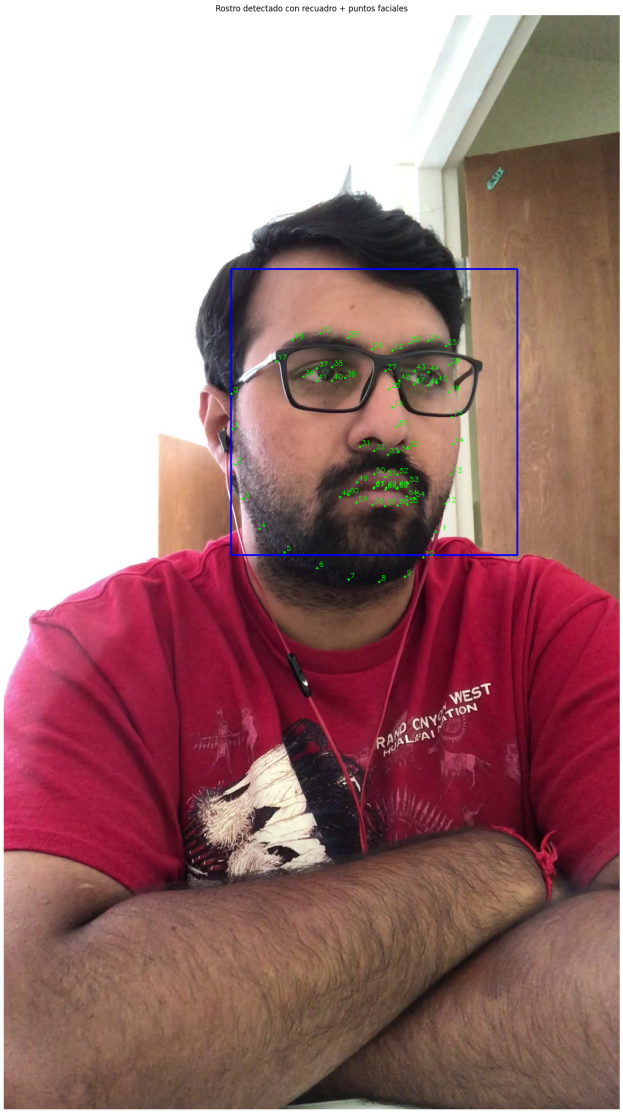


Figura 4.11: Ejemplo de detección de puntos faciales: recuadro delimitador y numeración de los 68 *landmarks*.

Cálculo de métricas geométricas

A partir de las coordenadas de los 68 puntos faciales detectados, se implementaron en `Python` funciones para calcular automáticamente las métricas EAR, MAR, PUC, MOE y CM. Cada función recibe como entrada un arreglo `shape` con las coordenadas (x, y) de los *landmarks* y devuelve un valor numérico que describe la característica geométrica correspondiente.

En particular, se emplearon las siguientes definiciones:

- **EAR (Eye Aspect Ratio):** mide la apertura ocular a partir de proporciones geométricas, ver Sección 3.3.1.

- **MAR (Mouth Aspect Ratio):** cuantifica la apertura de la boca mediante distancias verticales y horizontales, ver Sección 3.3.2.
- **PUC (Percentage of Unit Contact):** alternativa al EAR, más sensible a variaciones pequeñas en la apertura ocular, ver Sección 3.3.3.
- **MOE (Measure of Eye Occlusion):** métrica inversa al EAR que resalta cierres parciales o totales de los ojos, ver Sección 3.3.4.
- **Coloración de mejillas:** analiza variaciones cromáticas en el rostro asociadas a cambios fisiológicos, ver Sección 3.3.5.

De esta manera, se obtuvo un conjunto de métricas que sirvieron como variables de entrada para los modelos de aprendizaje automático superficial.

Extracción

El proceso de extracción consiste en tomar los valores numéricos generados por el módulo de cálculo y asociarlos a la imagen de origen. Para cada archivo procesado, se obtiene un registro compuesto por:

```
{nombre_imagen, clase, EAR, MAR, PUC, MOE, MEJILLA_IZQUIERDA,
                MEJILLA_DERECHA, id_video}
```

donde `nombre_imagen` identifica de forma única el fotograma analizado y `clase` corresponde a la etiqueta asignada según el estado de fatiga. Las métricas `MEJILLA_IZQUIERDA` y `MEJILLA_DERECHA` representan el valor promedio del componente *Hue* en las regiones de interés de cada mejilla. Finalmente, el campo `id_video` se incluye únicamente con fines organizativos, ya que permite mantener la trazabilidad entre cada fotograma y el video de origen, así como la clase a la que pertenece. Este identificador no se emplea como insumo en el entrenamiento de los modelos de clasificación, pero resulta esencial para garantizar orden, control y reproducibilidad en la gestión del conjunto de datos.

Los registros generados en la etapa de extracción se almacenan en un archivo llamado`metricas.csv`, lo que permite organizar de manera estructurada las características de cada imagen y facilita su posterior manipulación con bibliotecas de análisis de datos y aprendizaje automático (como `pandas` o `scikit-learn`). Este archivo constituye el conjunto de entrada para los modelos de aprendizaje automático superficial, asegurando

que cada instancia esté correctamente etiquetada con sus valores geométricos y cromáticos.

Construcción del Conjunto de Datos por Secuencias

Una vez consolidado el archivo `metricas.csv` con todas las métricas faciales extraídas, se obtuvo un total de 530,830 registros correspondientes a fotogramas individuales. Sin embargo, evaluar cada frame de manera independiente no resulta adecuado, ya que la fatiga no se manifiesta en un instante aislado, sino a lo largo de una secuencia temporal de imágenes. Por esta razón, en lugar de considerar cada fotograma como una instancia separada, se optó por agruparlos en segmentos que representen aproximadamente un segundo de video.

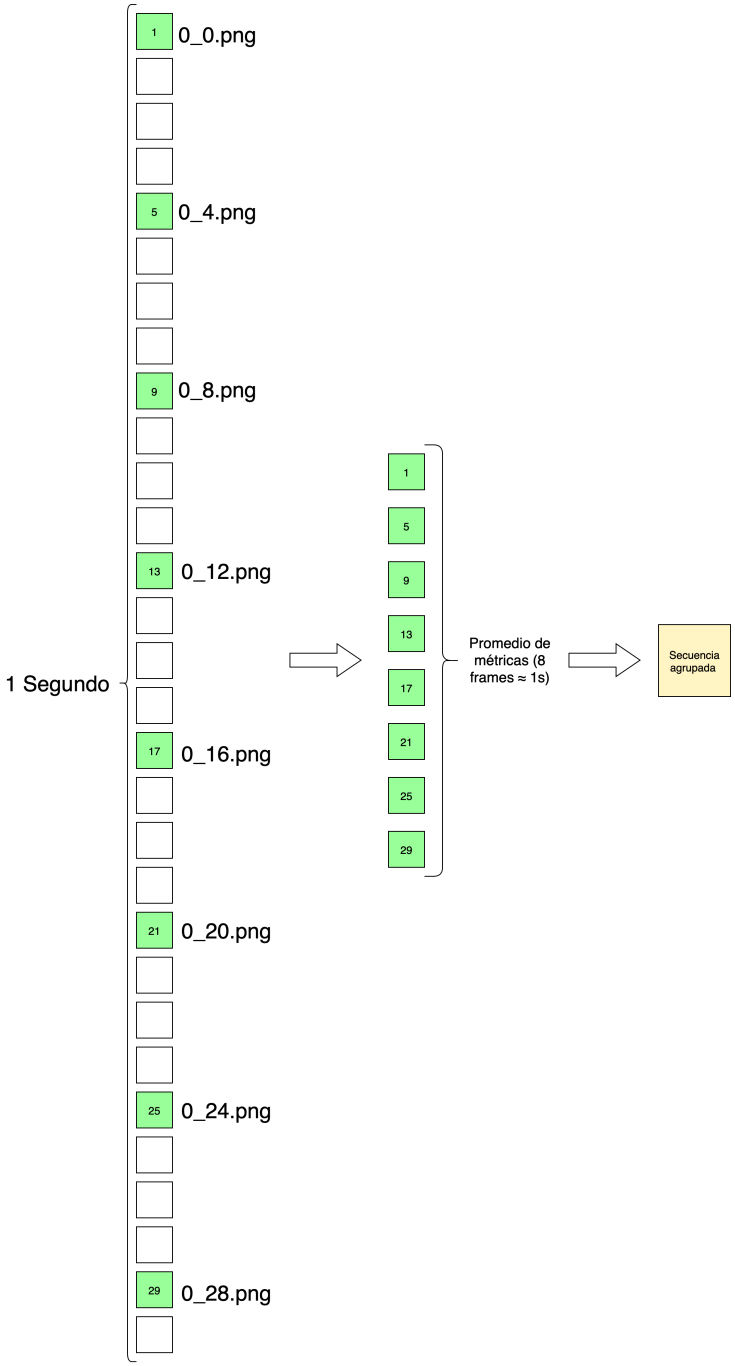


Figura 4.12: Agrupación de frames en secuencias temporales.

Dado que en la etapa de extracción se seleccionó un fotograma de cada cuatro, la frecuencia efectiva resultante fue de aproximadamente 7.5 fps. Bajo esta tasa de muestreo, un segundo de video queda representado por 8 frames consecutivos. Para construir instancias más estables y reducir la variabilidad propia de cada imagen individual, se calcularon los promedios de las métricas faciales en ventanas de 8 frames. De esta manera, cada secuencia temporal de un segundo se resume en un único registro que refleja el

comportamiento promedio de las variables durante ese intervalo.

Este procedimiento permite capturar patrones asociados a la fatiga —como parpadeos prolongados o aperturas bucales sostenidas— que difícilmente podrían identificarse en un solo fotograma aislado. Para asegurar la correcta formación de las secuencias, se empleó la columna `id_video` como criterio de agrupación, de modo que los promedios se calcularan únicamente entre frames pertenecientes al mismo video. Con ello se evita la mezcla de instancias de diferentes secuencias, lo que podría introducir valores erróneos y comprometer la coherencia temporal del conjunto de datos.

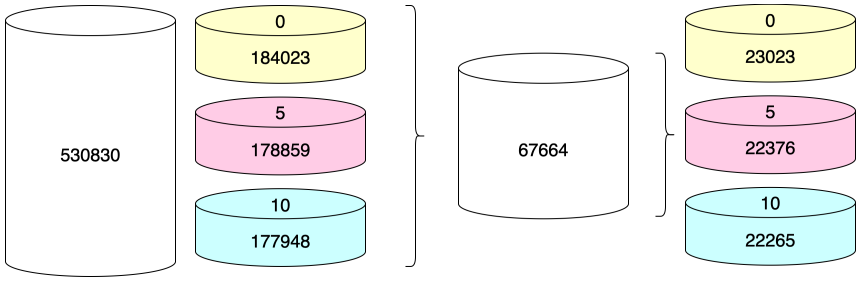


Figura 4.13: Reducción del conjunto de datos tras la agrupación por secuencias.

La Figura 4.13 muestra cómo el conjunto de datos pasó de 530,830 frames individuales a 67,664 instancias representativas de un segundo de video, manteniendo la proporción entre las tres clases. Este procedimiento evita el análisis de fotogramas aislados y refleja mejor la naturaleza temporal de la fatiga.

Introducción a los histogramas

Una vez conformado el conjunto de instancias representativas, se procedió a realizar un análisis exploratorio de las métricas faciales mediante histogramas, con el fin de examinar su distribución y posibles diferencias entre clases.

Un histograma es una representación gráfica que muestra la distribución de una variable continua dividiendo su rango en intervalos (*bins*) y contando cuántos valores caen dentro de cada uno. Esta herramienta permite identificar patrones, concentraciones de valores, asimetrías y posibles *outliers* en los datos.

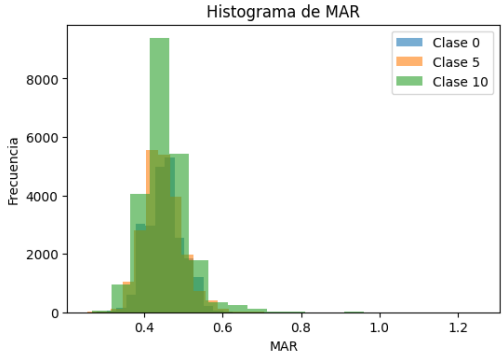
En este caso, se utilizaron histogramas para visualizar cómo varían las métricas faciales MAR, EAR, PUC y MOE, junto con las tonalidades promedio en `MEJILLA_IZQUIERDA` y `MEJILLA_DERECHA`, entre las tres clases objetivo (0, 5 y 10). Estas visualizaciones permiten interpretar el comportamiento facial característico de cada nivel de fatiga y evaluar

si existen diferencias estructurales que puedan ser aprovechadas por los modelos de clasificación. Al comparar las distribuciones por clase, se identifican patrones que orientan el diseño de algoritmos más sensibles y precisos para la detección automática de fatiga.

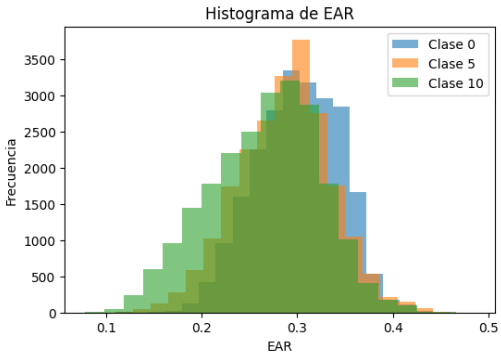
4.8.1. Análisis de distribuciones mediante histogramas

Una vez conformado el conjunto de instancias representativas, se procedió a realizar un análisis exploratorio de las métricas faciales mediante histogramas, con el fin de examinar su distribución y posibles diferencias entre clases.

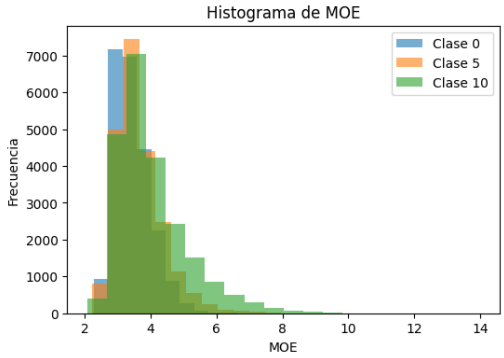
Un histograma es una representación gráfica que muestra la distribución de una variable continua dividiendo su rango en intervalos (*bins*) y contando cuántos valores caen dentro de cada uno. Esta herramienta permite identificar patrones, concentraciones de valores, asimetrías y posibles *outliers* en los datos.



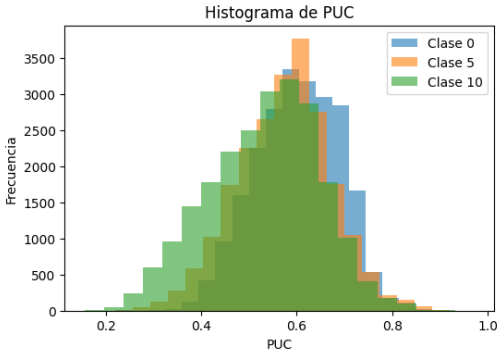
(a) Métrica MAR



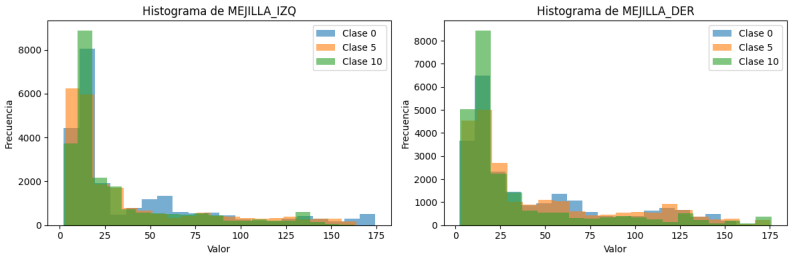
(b) Métrica EAR



(c) Métrica MOE



(d) Métrica PUC



(e) Tonalidades en mejillas (Hue promedio)

Figura 4.14: Distribución de las principales métricas faciales y cromáticas por clase (0, 5 y 10).

Métrica MAR (*Mouth Aspect Ratio*) La métrica MAR cuantifica la apertura relativa de la boca. Todas las clases presentan una asimetría positiva, concentrando la mayor parte de los valores entre 0.3 y 0.4. La clase 10 muestra el pico más pronunciado, lo que indica aperturas bucales reducidas en estados de fatiga avanzada. La superposición entre clases evidencia que MAR no es separable por sí sola, aunque aporta información útil al combinarse con otras métricas.

Métrica EAR (*Eye Aspect Ratio*) EAR mide la relación entre la altura y el ancho del ojo, actuando como indicador de parpadeo o cierre ocular. Las distribuciones son aproximadamente simétricas, con un pico alrededor de 0.3. A medida que aumenta la fatiga, la variabilidad disminuye, lo que refleja aperturas oculares más limitadas. Aunque el solapamiento entre clases es considerable, EAR refuerza el análisis al combinarse con variables oculares y cromáticas.

Métrica MOE (*Measure of Eye Occlusion*) Esta métrica evalúa el grado de oclusión ocular. Las tres clases presentan concentraciones en valores bajos, con la clase 10 destacando por un pico más pronunciado cerca de cero, lo que sugiere cierres oculares más frecuentes o prolongados. La clase 0 presenta valores más dispersos, lo que refleja mayor variabilidad y apertura ocular en sujetos en estado de alerta.

Métrica PUC (*Percentage of Unit Contact*) PUC representa el porcentaje de tiempo en que los párpados permanecen en contacto. La clase 0 concentra valores bajos (ojos abiertos), mientras que la clase 10 se desplaza hacia valores altos (ojos cerrados con mayor frecuencia). Esta métrica muestra la mayor separación entre clases, indicando un poder discriminativo destacado.

Tonalidades en mejillas Las métricas `MEJILLA_IZQUIERDA` y `MEJILLA_DERECHA` reflejan el valor promedio del componente *Hue* del modelo HSV. La mayoría de los valores se concentran en rangos bajos, con una disminución progresiva hacia los altos. Se observan acumulaciones adicionales en torno a ≈ 150, más notorias en la clase 10, lo que sugiere variaciones cromáticas específicas bajo fatiga avanzada.

Síntesis del análisis. En conjunto, el análisis individual de los histogramas revela que PUC presenta la mayor capacidad de separación entre clases, mientras que MAR y MOE muestran diferencias intermedias. Por su parte, EAR y las métricas cromáticas de mejillas exhiben un mayor grado de solapamiento entre clases.

Estas observaciones evidencian que ninguna métrica es completamente discriminativa por sí sola. Por tanto, la combinación de variables oculares, geométricas y cromáticas resulta esencial para capturar los patrones asociados a los distintos niveles de fatiga. Este hallazgo justifica la aplicación de técnicas de reducción de dimensionalidad, como PCA y LDA, con el fin de explorar representaciones más compactas y discriminativas, que se describen en la siguiente sección.

4.8.2. Análisis conjunto y matriz de correlación

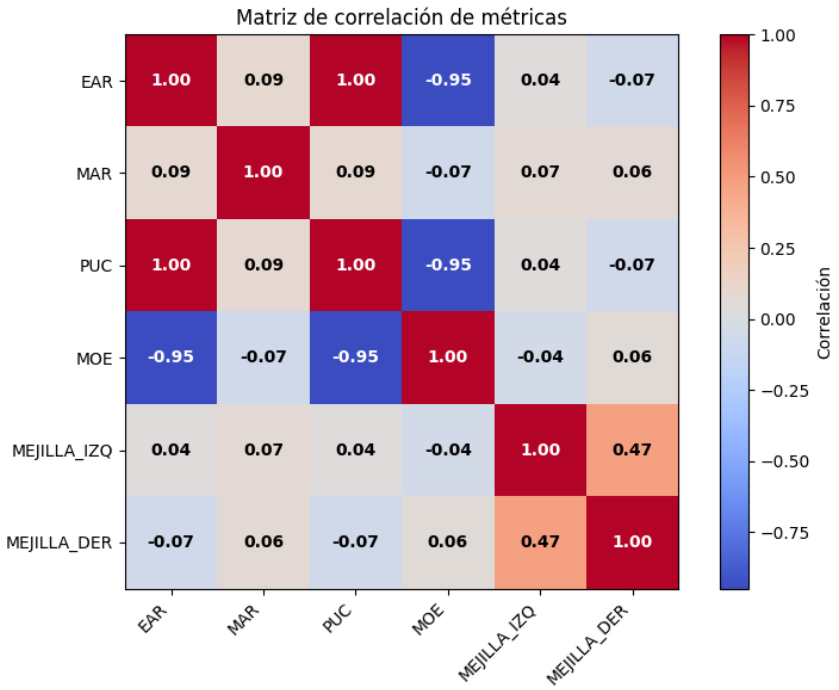


Figura 4.15: Matriz de correlación entre métricas faciales y la variable de clase.

El análisis individual de histogramas permitió identificar patrones característicos en cada métrica facial: PUC mostró la mayor separación entre clases, MAR y MOE evidenciaron diferencias intermedias, mientras que EAR y las métricas cromáticas de mejillas presentaron distribuciones con mayor solapamiento. Sin embargo, la capacidad discriminativa de cada variable por sí sola es limitada, lo que refuerza la necesidad de un enfoque multivariado.

La Figura 4.15 presenta la matriz de correlación entre todas las métricas y la variable de clase. Se observa que:

- PUC y MOE mantienen correlaciones fuertes y opuestas con la clase, coherentes con su relación directa con la dinámica de cierres oculares.
- EAR presenta correlación negativa, en línea con la reducción de apertura ocular a medida que aumenta la fatiga.
- MAR muestra correlación positiva baja, lo que indica un aporte complementario más que determinante.

- Las métricas cromáticas de mejillas (izquierda y derecha) exhiben correlación moderada entre sí, pero baja con la clase, lo que sugiere redundancia interna y un papel de refuerzo más que de discriminación principal.

En conjunto, la matriz confirma que no existe una única métrica con poder predictivo absoluto, sino que la combinación de variables oculares, geométricas y cromáticas es la que potencialmente maximiza la capacidad de clasificación. Este hallazgo respalda el uso de modelos multivariados que integren información complementaria para la detección de distintos niveles de fatiga.

El patrón de correlaciones observado también evidencia redundancia entre ciertas métricas y diferencias en su relación con la variable objetivo. Estas observaciones motivaron la aplicación de técnicas de reducción de dimensionalidad, como PCA y LDA, con el propósito de evaluar si una representación más compacta de las características podía mejorar la separabilidad entre clases y el rendimiento de los modelos de clasificación.

4.8.3. Reducción de características y selección de hiperparámetros

En esta etapa se buscó optimizar el desempeño de los modelos tradicionales mediante dos estrategias complementarias: la reducción de características y la selección de hiperparámetros. Aunque el conjunto de datos estaba compuesto únicamente por seis variables, se incluyeron técnicas de reducción de dimensionalidad con fines exploratorios y comparativos, con el propósito de identificar posibles redundancias entre variables, mejorar la separabilidad de las clases y analizar el impacto de estas transformaciones en el rendimiento de los clasificadores.

4.8.4. Preparación y dimensionalidad de características

Para encontrar la configuración más adecuada en cada caso, se diseñó una estrategia de evaluación que integró tanto la reducción de características como la optimización de hiperparámetros. Este proceso se implementó mediante `pipelines` que combinan, de forma ordenada, la etapa de transformación de variables y el modelo de clasificación.

Reducción de características

Se evaluaron dos enfoques para transformar el espacio original de variables:

- **PCA**: es una técnica no supervisada que transforma el conjunto original de variables en un nuevo espacio de menor dimensión, priorizando aquellas direcciones en las que los datos presentan mayor variabilidad. De esta manera, se logra condensar la información más relevante en un número reducido de componentes principales, lo que facilita tanto la visualización como la detección de patrones subyacentes. En este trabajo se evaluaron configuraciones con 2, 3, 4 y 5 componentes principales, con el fin de analizar cómo la reducción progresiva del espacio de variables influía en la capacidad de los modelos para discriminar entre los distintos niveles de somnolencia.
- **LDA**: a diferencia de PCA, esta técnica es supervisada y utiliza la información de las clases para encontrar combinaciones lineales de variables que maximicen la separación entre categorías. Su objetivo es proyectar los datos en un espacio de menor dimensión en el que los ejemplos de una misma clase queden más agrupados y, al mismo tiempo, se distancien de los de clases diferentes. En el caso particular de un problema con tres clases, LDA permite obtener hasta dos dimensiones, lo que no solo facilita la representación gráfica de los datos, sino que también ofrece una base más clara para evaluar el efecto de esta transformación en el rendimiento de los clasificadores.

Pipelines y Búsqueda de Hiperparámetros

Con el objetivo de evaluar distintas configuraciones de modelos, se construyeron múltiples `pipelines` que combinan procesos de estandarización, reducción de dimensionalidad y clasificación. Para cada modelo base se implementaron versiones con y sin reducción de características, aplicando tanto PCA (con entre 2 y 5 componentes principales) como LDA (con 1 o 2 componentes, según el caso). Cada pipeline fue optimizado mediante una búsqueda sistemática de hiperparámetros con `GridSearchCV`, utilizando `accuracy` como métrica principal de evaluación.

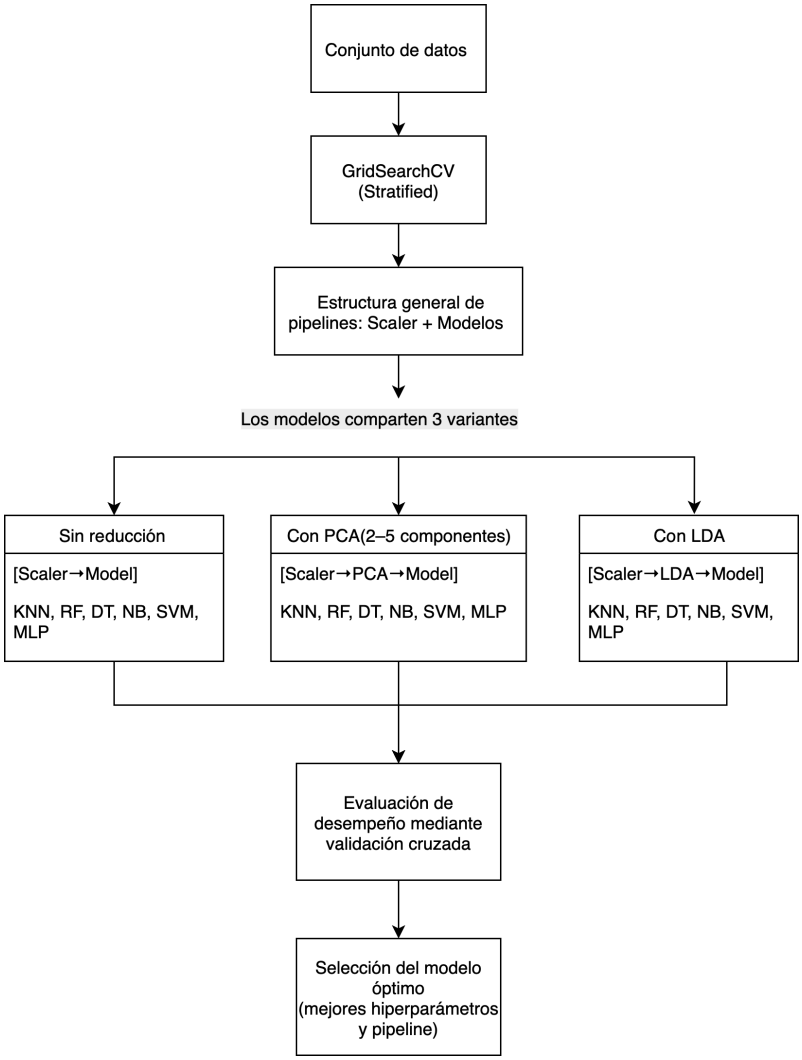


Figura 4.16: Esquema general de los pipelines evaluados, con y sin reducción de características.

Cuadro 4.2: Espacios de búsqueda de hiperparámetros definidos por modelo.

Modelo	**Hiperparámetros explorados y descripción**
KNN	***n_neighbors*** = (3, 5, 7): número de vecinos considerados. ***weights*** = (uniform, distance): ponderación de los vecinos. ***metric*** = (minkowski, euclidean, manhattan): forma de medir la distancia.
Random Forest	***n_estimators*** = (50–150): número de árboles del bosque. ***max_depth*** = (None, 10, 20): controla la profundidad máxima de cada árbol.
Decision Tree	***max_depth*** = (None, 10, 20): complejidad del árbol. ***criterion*** = (gini, entropy): criterio para evaluar las divisiones.
Naive Bayes	***var_smoothing*** = (1e-9 a 1e-7): factor de suavizado para evitar problemas con varianzas muy pequeñas.
SVM	***C*** = (0.1, 1, 10): parámetro de regularización. Valores pequeños implican mayor penalización a errores, mientras que valores grandes permiten mayor flexibilidad.
MLP	***hidden_layer_sizes*** = ((128,64), (256,128)): arquitectura de las capas ocultas. ***activation*** = (relu): función de activación. ***solver*** = (adam): método de optimización. ***learning_rate*** = (adaptive): tasa de aprendizaje ajustada dinámicamente. ***alpha*** = (1e-4, 1e-3): regularización L2. ***max_iter*** = (100): número máximo de iteraciones.

Para cada algoritmo se implementaron distintas variantes de `pipeline`, incluyendo versiones sin reducción de dimensionalidad (*No_PCA*), con PCA de 2 a 5 componentes y con LDA. En todos los casos se aplicó una estandarización previa mediante `StandardScaler`, asegurando que las variables tuvieran media cero y desviación estándar unitaria antes del entrenamiento.

La validación se realizó utilizando un esquema de `StratifiedKFold` con cinco particiones, manteniendo la proporción de clases en cada subdivisión. Para este procedimiento, se empleó únicamente el 80 % del conjunto total de datos, destinado al entrenamiento y validación interna mediante `GridSearchCV`.

Es importante subrayar que el 20 % restante del conjunto total fue reservado de antemano como **evaluación independiente**, sin participar en el entrenamiento ni en la validación interna. Dicho subconjunto se utilizó exclusivamente al final, garantizando que los modelos fueran probados sobre datos completamente no vistos.

De esta manera, se reduce el riesgo de sobreajuste y se obtiene una estimación más realista del desempeño de los modelos en condiciones de generalización. Finalmente, los mejores parámetros obtenidos mediante `GridSearchCV` fueron registrados junto con las métricas de rendimiento, permitiendo comparar el desempeño entre modelos y configuraciones de reducción de características.

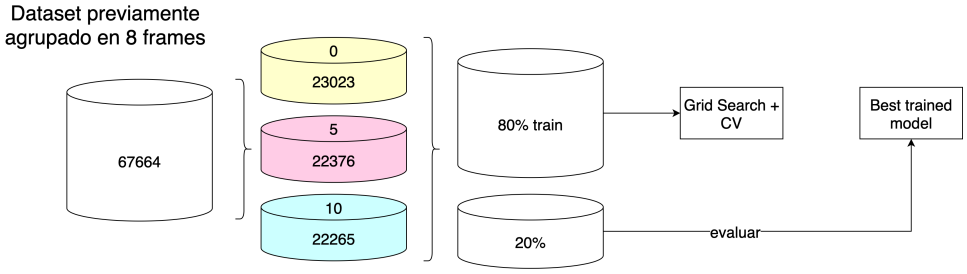


Figura 4.17: Esquema del protocolo de entrenamiento experimental.

La Figura 4.17 resume gráficamente este procedimiento, mostrando la división del conjunto de datos reducido (tras la agrupación de 8 frames) y su utilización en las etapas de entrenamiento, validación interna y evaluación final independiente.

Entrenamiento de Modelos

Se evaluaron seis clasificadores de aprendizaje automático superficial: *Multilayer Perceptron (MLP)*, *Random Forest (RF)*, *Support Vector Machine (SVM)*, *Decision Tree (DT)*, *K-Nearest Neighbors (KNN)* y *Naive Bayes (NB)*. Todos fueron entrenados siguiendo un

`pipeline` común que incluyó, cuando correspondía, estandarización de variables, reducción de dimensionalidad, validación cruzada estratificada de cinco pliegues y optimización de hiperparámetros mediante `GridSearchCV`. A continuación, se resumen las particularidades de cada modelo.

4.9.1. MLP

El entrenamiento del *Multilayer Perceptron (MLP)* se realizó mediante un `pipeline` que incluyó estandarización de variables, reducción de dimensionalidad (cuando aplicaba) y clasificación. Se utilizó validación cruzada estratificada de cinco pliegues sobre el 80 % del conjunto de entrenamiento, lo que permitió mantener la proporción de clases y obtener una estimación más estable del rendimiento.

La optimización de hiperparámetros se llevó a cabo con `GridSearchCV`, explorando número y tamaño de capas ocultas, función de activación, optimizador, tasa de aprendizaje y regularización. Dado que el modelo es sensible a la escala, se aplicó `StandardScaler` para garantizar media cero y varianza unitaria en todas las variables. Los modelos resultantes fueron almacenados para su posterior evaluación en el conjunto de prueba independiente.

4.9.2. Random Forest

El entrenamiento del clasificador *Random Forest* se realizó mediante un `pipeline` que incluyó validación cruzada estratificada de cinco pliegues sobre el 80 % del conjunto de entrenamiento. Este esquema permitió mantener la proporción de clases y obtener una estimación más estable del rendimiento.

La optimización de hiperparámetros se llevó a cabo con `GridSearchCV`, explorando principalmente el número de árboles (`n_estimators`) y la profundidad máxima de cada árbol (`max_depth`). Dado que los algoritmos basados en árboles no dependen de la escala de las variables, no fue necesario aplicar estandarización previa.

Los parámetros óptimos y las métricas de validación interna fueron registrados, y los modelos resultantes se almacenaron para su posterior evaluación en el conjunto de prueba independiente (20 %).

4.9.3. SVM

El clasificador *Support Vector Machine (SVM)* se entrenó mediante un `pipeline` que incluyó estandarización de variables, reducción de dimensionalidad (cuando aplicaba) y clasificación. Se utilizó validación cruzada estratificada de cinco pliegues sobre el 80 % del conjunto de entrenamiento, lo que permitió mantener la proporción de clases y obtener una estimación más estable del rendimiento.

La optimización de hiperparámetros se realizó con `GridSearchCV`, evaluando combinaciones de `kernel`, parámetro de regularización C y coeficiente γ. Para cada configuración se calcularon métricas de validación interna (exactitud promedio y desviación estándar). Los parámetros óptimos fueron registrados y los modelos resultantes almacenados para su posterior evaluación en el conjunto de prueba independiente (20 %).

4.9.4. Decision Tree

El clasificador *Decision Tree* se entrenó mediante un `pipeline` con validación cruzada estratificada de cinco pliegues sobre el 80 % del conjunto de entrenamiento. Este esquema permitió mantener la proporción de clases y obtener una estimación más estable del rendimiento.

La optimización de hiperparámetros se realizó con `GridSearchCV`, explorando el criterio de impureza (`gini` o `entropy`) y la profundidad máxima del árbol (`max_depth`). Al no depender de la escala de las variables, no fue necesario aplicar estandarización previa.

Los parámetros óptimos y las métricas de validación interna fueron registrados, y los modelos resultantes se almacenaron para su posterior evaluación en el conjunto de prueba independiente (20 %).

4.9.5. K-Nearest Neighbors (KNN)

El clasificador *KNN* se entrenó mediante un `pipeline` con validación cruzada estratificada de cinco pliegues sobre el 80 % del conjunto de entrenamiento. Este esquema permitió mantener la proporción de clases y obtener una estimación más estable del rendimiento.

La optimización de hiperparámetros se realizó con `GridSearchCV`, explorando el número de vecinos (`n_neighbors`), la métrica de distancia (`metric`) y el esquema de ponde-

ración (`weights`). Dado que el modelo es sensible a la escala de las variables, se incluyó una etapa de estandarización previa para garantizar comparabilidad entre características.

Los parámetros óptimos y las métricas de validación interna fueron registrados, y los modelos resultantes se almacenaron para su posterior evaluación en el conjunto de prueba independiente (20 %).

4.9.6. Naive Bayes

El modelo *Naive Bayes* se entrenó mediante un `pipeline` con validación cruzada estratificada de cinco pliegues sobre el 80 % del conjunto de entrenamiento. Se utilizó la variante *GaussianNB*, adecuada para variables continuas bajo el supuesto de distribución normal.

La optimización de hiperparámetros se realizó con `GridSearchCV`, ajustando el parámetro `var_smoothing` para mejorar la estabilidad numérica frente a varianzas muy pequeñas.

Los parámetros óptimos y las métricas de validación interna (exactitud promedio y desviación estándar) fueron registrados, y los modelos resultantes se almacenaron para su posterior evaluación en el conjunto de prueba independiente (20 %).

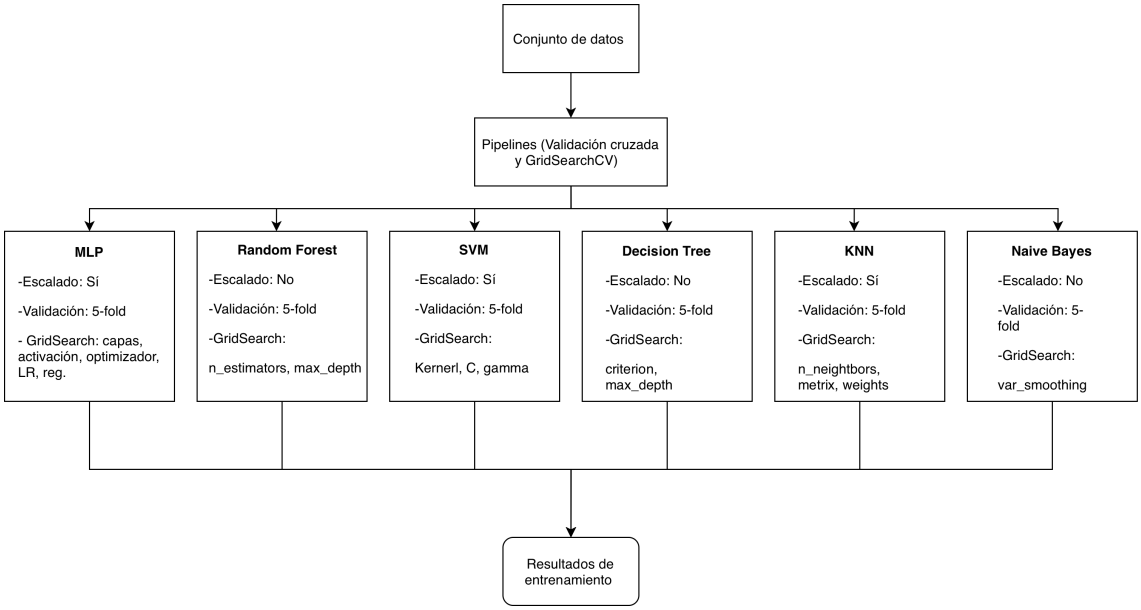


Figura 4.18: Esquema de entrenamiento y validación de los modelos.

En la Figura 4.18 se resume el procedimiento aplicado a cada clasificador. Todos los

modelos fueron entrenados bajo un `pipeline` común que incluyó validación cruzada estratificada de cinco pliegues y optimización de hiperparámetros mediante `GridSearchCV`. Las diferencias principales se encuentran en la necesidad de aplicar estandarización previa y en los parámetros específicos ajustados en cada caso. Finalmente, los modelos seleccionados fueron almacenados para su posterior evaluación en el conjunto de prueba independiente (20 %).

Enfoque basado en visión por computador

Como se indicó en la sección de metodología, ambos enfoques comparten las tres primeras etapas del flujo de trabajo: (1) selección del conjunto de datos, (2) extracción de fotogramas desde los videos y (3) preprocesamiento de las imágenes.

A partir de este punto, el presente enfoque adopta una estrategia distinta, prescindiendo del cálculo explícito de métricas geométricas y empleando en su lugar un detector de objetos de última generación (YOLO). Este modelo es capaz de aprender directamente a partir de imágenes anotadas, identificando y localizando patrones visuales asociados a la somnolencia sin necesidad de ingeniería manual de características.

4.10.1. Anotación automática y almacenamiento de datos

Tras completar las tres etapas comunes a ambos enfoques —selección del conjunto de datos, extracción de fotogramas y preprocesamiento—, el flujo de trabajo del enfoque basado en visión por computador incorpora una fase específica de **anotación automática y almacenamiento** de las regiones de interés.

Para automatizar este proceso, se desarrolló un script que emplea el clasificador *Haar Cascade* de OpenCV para detectar la región facial en cada fotograma correspondiente a las distintas clases de somnolencia (0, 5 y 10). Una vez detectado el rostro, se extraen las coordenadas de la *bounding box* que lo encierra y se almacenan inicialmente en un único archivo de texto por clase, donde se registran todas las detecciones correspondientes a esa categoría. Este formato intermedio facilita la revisión y depuración de las anotaciones antes de su conversión.

Este procedimiento elimina la necesidad de un etiquetado manual exhaustivo —lo cual sería inviable dada la gran cantidad de imágenes procesadas— y automatiza significativamente el flujo de trabajo, garantizando anotaciones consistentes y precisas. De esta manera, el modelo YOLO puede concentrar su análisis en la región facial, donde se

manifiestan las características relevantes para la detección de somnolencia.

En la Figura 4.19 se ilustra este proceso: a la izquierda se observa la imagen original, mientras que a la derecha se muestra la misma imagen con la detección facial resaltada mediante una caja delimitadora. Esta comparación evidencia la importancia de indicar explícitamente la zona de interés; sin estas coordenadas, el modelo analizaría la imagen completa, lo que podría introducir ruido y dificultar el aprendizaje. En cambio, al focalizarse en el rostro, el modelo optimiza su capacidad para identificar patrones asociados a los diferentes niveles de somnolencia.

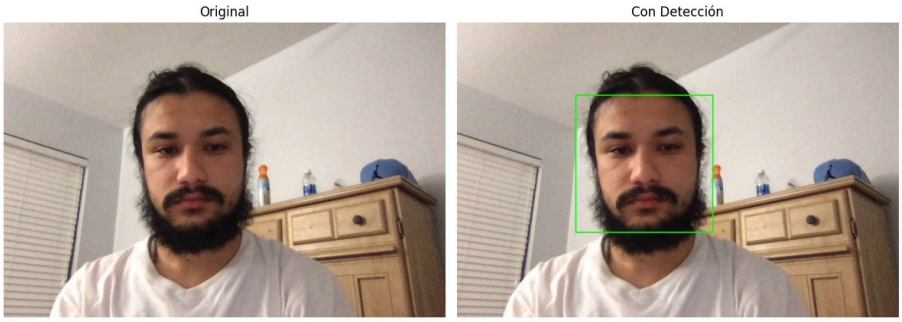


Figura 4.19: Ejemplo de imagen original (izquierda) y detección facial (derecha) utilizada para generar anotaciones de entrenamiento.

Conversión de anotaciones al formato YOLO

Una vez generados los archivos de anotaciones globales por clase, fue necesario transformarlos al formato requerido por YOLO. A diferencia del formato intermedio, YOLO exige que cada imagen disponga de su propio archivo de etiquetas, con el mismo nombre que la imagen y extensión `.txt`, conteniendo exclusivamente las anotaciones correspondientes a esa imagen. Además, este formato utiliza coordenadas normalizadas en lugar de píxeles absolutos, y cada línea debe seguir la estructura:

```
<clase><x_center><y_center><width><height>
```

donde todos los valores están normalizados respecto al ancho y alto de la imagen.

Para ilustrar la transformación, a continuación se presenta un ejemplo del contenido original generado tras la detección con la cascada de Haar, seguido de su correspondiente anotación en formato YOLO:

- **Formato original (coordenadas absolutas, archivo global por clase)**:

```
0_8224.png 385 460 857 932
```

(donde los valores representan: `<nombre_imagen><x_min><y_min><x_max><y_max>`)

- **Formato YOLO (coordenadas normalizadas, archivo individual por imagen)**:

  `0 0.575 0.3625 0.43703703703703706 0.24583333333333332`

  (donde los valores representan: `<clase><x_center><y_center><width><height>`)

4.10.2. División del conjunto de datos

Una vez generadas todas las anotaciones en formato YOLO —con un archivo `.txt` individual por imagen—, el material se encontraba almacenado en una ubicación única, sin una estructura específica para el entrenamiento. Para preparar el conjunto de datos de forma óptima, se implementó un algoritmo que organizó y dividió el contenido en tres subconjuntos: 80 % para entrenamiento, 10 % para validación y 10 % para prueba.

Este proceso no solo distribuyó las imágenes de manera proporcional, sino que también garantizó que cada imagen estuviera correctamente emparejada con su archivo de etiquetas correspondiente. Para ello, el script creó la carpeta raíz `dataset_yolo/`, dentro de la cual se generaron dos subcarpetas principales:

- `images/`: Contiene todas las imágenes, organizadas en `train/`, `val/` y `test/`.
- `labels/`: Contiene los archivos de anotaciones en formato YOLO, con la misma estructura de subdivisión que las imágenes.

Esta organización sigue la convención estándar utilizada por los entrenadores de YOLO, lo que facilita la carga directa del conjunto de datos en el modelo sin pasos adicionales de preprocesamiento. Además, asegura la reproducibilidad del experimento y la trazabilidad de cada imagen y su etiqueta.

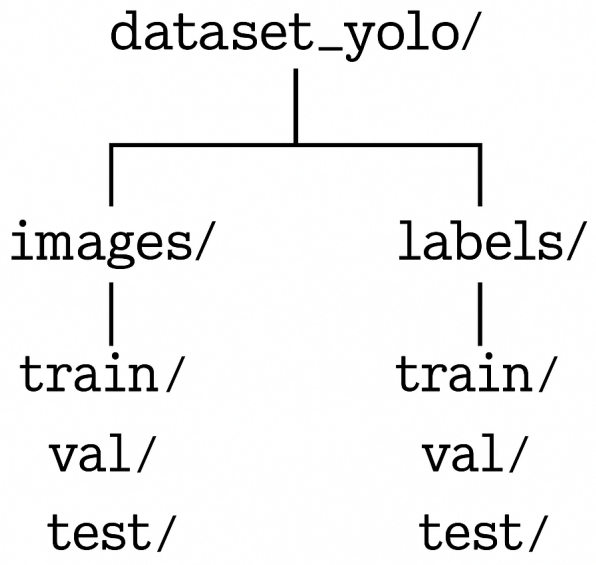


Figura 4.20: Estructura final del conjunto de datos en formato YOLO, con división en entrenamiento (80 %), validación (10 %) y prueba (10 %).

Generación del archivo `data.yaml`

El archivo `data.yaml` es un componente esencial en la configuración de un modelo de detección de objetos con YOLO, ya que define la estructura del conjunto de datos y permite al sistema de entrenamiento localizar correctamente las imágenes y sus anotaciones.

Este archivo actúa como un descriptor que especifica:

- La ruta base donde se encuentran las carpetas de imágenes y etiquetas.
- Las subrutas correspondientes a los subconjuntos de entrenamiento y validación.
- La lista de clases presentes en el dataset, junto con su identificador numérico.

YOLO utiliza este archivo al momento de cargar los datos, verificando automáticamente la existencia de imágenes y etiquetas, asociando correctamente cada imagen con su archivo de anotación correspondiente, y reconociendo las clases que deben ser aprendidas. Sin este archivo, el sistema no sabría cuántas clases hay, cómo se llaman, ni en qué ubicación buscar los recursos necesarios para el entrenamiento y validación.

A continuación se muestra un ejemplo del contenido del archivo `data.yaml`:

```
path: F:/Yolo/dataset_definitivo

train: images/train
val: images/val

names:
  0: "0"
  1: "5"
  2: "10"
```

Donde:

- `path` indica la ruta raíz que contiene las carpetas `images/` y `labels/`.
- `train` y `val` son las rutas relativas a los subconjuntos de entrenamiento y validación, respectivamente.
- `names` define la relación entre los identificadores numéricos de clase (usados en los archivos de etiquetas) y su representación textual.

Este archivo puede generarse de manera automática mediante un breve script en Python:

```
yaml_path = r"F:\Yolo\data.yaml"

yaml_content = """
path: F:/Yolo/dataset_definitivo

train: images/train
val: images/val

names:
  0: "0"
  1: "5"
  2: "10"
"""
```

```
with open(yaml_path, "w") as f:
    f.write(yaml_content.strip())

print(f"Archivo data.yaml sobrescrito en: {yaml_path}")
```

Una vez creado el archivo `data.yaml`, el conjunto de datos queda completamente listo para ser utilizado por cualquier modelo basado en YOLO, tanto en tareas de entrenamiento como de validación o inferencia.

4.10.3. Entrenamiento del Modelo YOLO

Una vez preparado el conjunto de datos y generado el archivo `data.yaml`, se procedió al entrenamiento del modelo utilizando la implementación oficial de `YOLOv8` proporcionada por la librería `Ultralytics`. Esta biblioteca de alto nivel simplifica el uso de modelos YOLO al ofrecer una API intuitiva para entrenamiento, validación e inferencia. Para el manejo de la GPU y la detección automática de dispositivos, se empleó la librería `torch` (PyTorch), que proporciona las herramientas necesarias para computación acelerada y entrenamiento de redes neuronales profundas.

El modelo base seleccionado fue `yolov8s.pt`, una versión ligera de YOLOv8 diseñada para lograr un equilibrio entre velocidad y precisión. El entrenamiento se realizó en una tarjeta gráfica **NVIDIA RTX 4080 Super**, aprovechando la aceleración por GPU para reducir significativamente los tiempos de procesamiento.

El siguiente script resume los parámetros utilizados:

```
from ultralytics import YOLO
import torch

# Verificación de GPU
print("¿GPU disponible?:", torch.cuda.is_available())
print("Nombre de la GPU:", torch.cuda.get_device_name(0))

# Cargar modelo YOLOv8 preentrenado (versión Small)
model = YOLO("yolov8s.pt")
```

```
# Entrenamiento
model.train(
    data="F:/Yolo/data.yaml",
    epochs=50,
    imgsz=640,
    batch=32,
    device=0,
    project="F:/Yolo",
    name="deteccion_somnolencia_v1",
    exist_ok=True,
    workers=12,
    amp=True,
    patience=20
)
```

Durante el entrenamiento se empleó una resolución de entrada de **640 píxeles**, un **tamaño de lote de 32** y un máximo de **50 épocas**, habilitando además el uso de **precisión mixta (AMP)** para optimizar memoria y rendimiento. Se implementó un esquema de **early stopping** con una paciencia de 20 épocas para prevenir el sobre-entrenamiento.

Los resultados de cada época —incluyendo métricas como **precisión**, **recall** y **mAP@0.5**— se almacenaron automáticamente en la ruta `F:/Yolo/runs/detect/deteccion_somnolencia_v1`, permitiendo disponer de un modelo entrenado de forma eficiente para la detección de regiones faciales asociadas a distintos niveles de somnolencia, cuyo desempeño se analizará en las siguientes secciones.

Resumen del Entrenamiento

El modelo YOLOv8s se entrenó durante **50 épocas**, sin que se activara el mecanismo de `early stopping`, ya que las métricas continuaron mejorando de forma sostenida. Se observó una reducción progresiva de las pérdidas (*box*, *class* y *DFL*), como se aprecia en la Figura 4.21 (arriba izquierda), lo que evidencia una buena capacidad de generalización.

La métrica `mAP@0.5` alcanzó un valor final cercano a **0.96**, mientras que `mAP@0.5:0.95` se estabilizó alrededor de **0.93**, indicando alta precisión en las predicciones. Las curvas

de **precisión** y **recall** también mostraron un comportamiento consistente a lo largo de las épocas, con valores finales superiores a **0.92**.

En conjunto, estos resultados confirman que el modelo aprendió de forma estable y efectiva las características relevantes de las clases presentes en el conjunto de datos, sin evidencias de sobreajuste ni degradación del rendimiento.

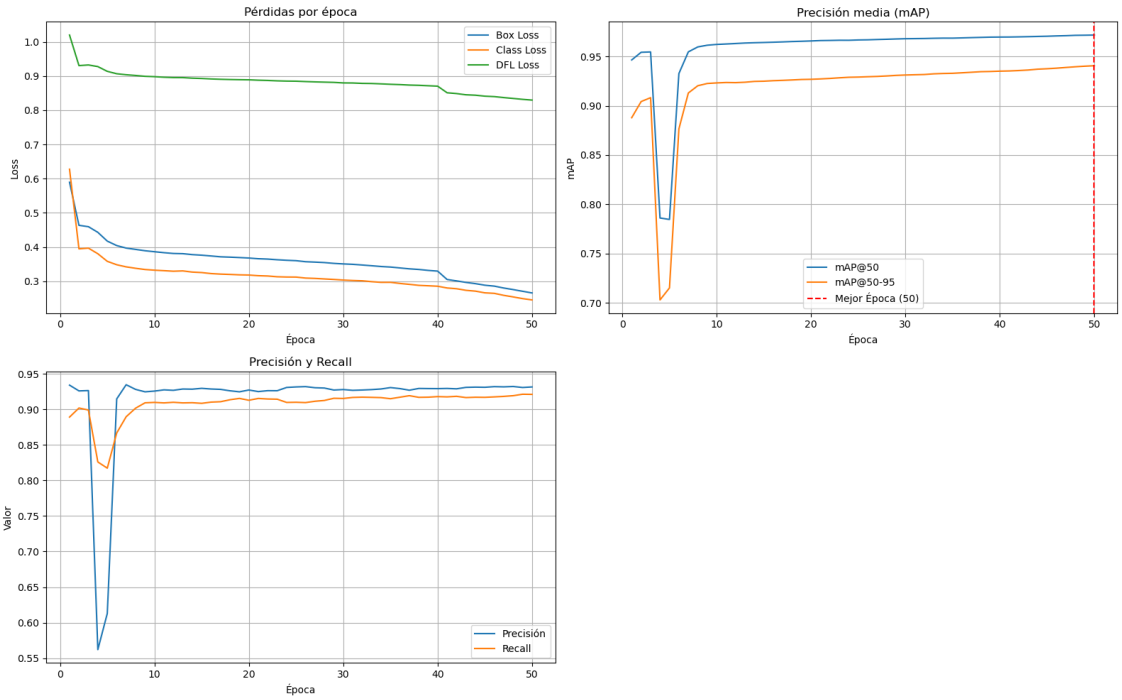


Figura 4.21: Evolución de las métricas durante el entrenamiento: pérdidas (arriba izquierda), precisión media mAP (arriba derecha) y precisión/recall (abajo).

5. Resultados

En este capítulo se presentan los resultados obtenidos tras el entrenamiento de los modelos diseñados para clasificar niveles de somnolencia a partir de imágenes faciales. Se comparan dos enfoques: (1) modelos de **aprendizaje automático superficial**, y (2) un modelo de **aprendizaje profundo** basado en visión por computador (YOLO). El objetivo es analizar el desempeño de cada enfoque bajo un mismo esquema de clasificación multiclase, considerando las tres categorías definidas en el dataset UTA-RLDD: 0 (alerta), 5 (baja vigilancia) y 10 (somnolencia).

Modelos de Aprendizaje Automático Superficial

En esta sección se presentan los resultados obtenidos con los clasificadores tradicionales de aprendizaje automático. Cada modelo fue entrenado y validado bajo el esquema metodológico descrito previamente, incorporando variantes con y sin reducción de dimensionalidad.

El objetivo de este apartado no es reiterar el procedimiento, sino mostrar de manera comparativa el desempeño alcanzado por cada clasificador en las distintas configuraciones evaluadas. De esta forma, se busca identificar patrones comunes, contrastar fortalezas y limitaciones, y establecer qué combinaciones de técnicas ofrecieron un mejor equilibrio entre exactitud y capacidad de generalización.

Resultados Comparativos y Evaluación Final del MLP

Los resultados obtenidos para cada variante del clasificador *Multilayer Perceptron (MLP)* se resumen en la Tabla 5.1. Se incluyen tanto los valores de exactitud promedio en validación cruzada (80 % de los datos) como los resultados finales en el conjunto de prueba independiente (20 %).

Cuadro 5.1: Resultados de exactitud en validación cruzada y en el conjunto de prueba independiente para las variantes del MLP.

Variante	Precisión en Validación Cruzada	Precisión en Conjunto de Prueba (20 %)	Observaciones
MLP_No_PCA	99.98	99.93	Desempeño casi perfecto; posible sobreajuste.
MLP_PCA_2	54.33	53.69	Pérdida significativa de información; bajo rendimiento.
MLP_PCA_3	71.75	70.89	Mejora respecto a PCA_2, pero aún limitado.
MLP_PCA_4	98.38	98.21	Muy alto desempeño; resultados poco realistas en generalización.
MLP_PCA_5	99.95	99.89	Resultados casi perfectos; riesgo de sobreajuste similar a No_PCA.
MLP_LDA	93.12	93.12	Rendimiento sólido y equilibrado; más representativo de la generalización.

En términos generales, las variantes **MLP_No_PCA** y **MLP_PCA_5** alcanzaron exactitudes cercanas al 100 %, aunque estos valores tan elevados podrían reflejar un sobreajuste al conjunto de entrenamiento. Por el contrario, la variante **MLP_LDA** mostró un desempeño más moderado (93.12 %), pero con métricas equilibradas entre clases, lo que la convierte en la opción más representativa y defendible en términos de capacidad de generalización.

5.1.1. Random Forest

Resultados Comparativos

Se evaluaron distintas variantes del clasificador *Random Forest*, incorporando reducciones de dimensionalidad mediante PCA (con 2 a 5 componentes) y LDA. Los resultados de validación cruzada mostraron un rango amplio de desempeños, desde configuraciones con pérdida significativa de información hasta variantes con valores cercanos a la perfección.

Evaluación Final en el Conjunto de Prueba (20 %)

Una vez identificadas las configuraciones óptimas mediante validación cruzada, los modelos de *Random Forest* fueron evaluados en el 20 % del conjunto de datos reservado de manera independiente. Los resultados obtenidos se resumen en la Tabla 5.2.

Cuadro 5.2: Resultados de precisión en validación cruzada y en el conjunto de prueba independiente para las variantes de Random Forest.

Variante	Precisión en Validación Cruzada	Precisión en Conjunto de Prueba (20 %)	Observaciones
RF_No_PCA	100.00	99.98	Desempeño casi perfecto; posible sobreajuste o sensibilidad excesiva a los datos.
RF_PCA_2	58.80	54.97	Pérdida significativa de información; bajo rendimiento.
RF_PCA_3	97.49	72.10	Mejora respecto a PCA_2, pero con errores en clases minoritarias.
RF_PCA_4	100.00	95.51	Muy alto desempeño; resultados más consistentes y defendibles.
RF_PCA_5	100.00	96.69	Rendimiento sobresaliente, aunque menos estable que PCA_4.
RF_LDA	90.08	89.61	Buen desempeño general, pero inferior a PCA con mayor dimensionalidad.

En términos generales, aunque la variante **RF_No_PCA** alcanzó valores cercanos a la perfección, este comportamiento puede reflejar un sobreajuste al conjunto de entrenamiento o una excesiva dependencia de las características originales. Por el contrario, la variante **RF_PCA_4** ofreció un equilibrio más sólido entre precisión y capacidad de generalización, consolidándose como la configuración más adecuada dentro de las evaluadas.

5.1.2. Support Vector Machine (SVM)

Resultados Comparativos y Evaluación Final

Se evaluaron distintas variantes del clasificador *Support Vector Machine (SVM)*, incorporando reducciones de dimensionalidad mediante PCA (con 2 a 5 componentes) y LDA. Los resultados de validación cruzada y de prueba independiente mostraron diferencias notables en el desempeño, resumidas en la Tabla 5.3.

Cuadro 5.3: Resultados de precisión en validación cruzada y en el conjunto de prueba independiente para las variantes de SVM.

Variante	Precisión en Validación Cruzada (80 %)	Precisión en Conjunto de Prueba (20 %)	Observaciones
SVM_No_PCA	73.98	73.86	Configuración sólida; desempeño alto, aunque con ligero desequilibrio en clases intermedias.
SVM_PCA_2	47.06	46.62	Pérdida considerable de información; desempeño muy limitado.
SVM_PCA_3	51.25	51.61	Mejora respecto a PCA_2, pero aún insuficiente para clasificación confiable.
SVM_PCA_4	67.69	67.94	Rendimiento intermedio; buen equilibrio, aunque con variaciones entre clases.
SVM_PCA_5	74.63	74.46	Mejor resultado global; desempeño consistente y competitivo frente a No_PCA.
SVM_LDA	69.03	68.96	Rendimiento aceptable, aunque inferior a PCA_5 y No_PCA.

Tras comparar los resultados obtenidos, las variantes **SVM_No_PCA** y **SVM_PCA_5** alcanzaron los mejores resultados, con desempeños muy similares. Sin embargo, se selecciona **SVM_PCA_5** como la configuración más representativa, al ofrecer un mejor compromiso entre precisión global y estabilidad en la clasificación de las distintas clases.

5.1.3. Decision Tree

Resultados Comparativos y Evaluación Final

Se entrenaron distintas variantes del clasificador *Decision Tree*, incluyendo configuraciones sin reducción de dimensionalidad, con PCA (2 a 5 componentes) y con LDA. Los resultados de validación cruzada y de prueba independiente se resumen en la Tabla 5.4.

Cuadro 5.4: Resultados de precisión en validación cruzada y en el conjunto de prueba independiente para las variantes de Decision Tree.

Variante	Precisión en Validación Cruzada (80 %)	Precisión en Conjunto de Prueba (20 %)	Observaciones
DT_No_PCA	100.00	100.00	Desempeño perfecto; resultado extremo que puede reflejar sobreajuste.
DT_PCA_2	57.25	54.27	Pérdida significativa de información; bajo rendimiento.
DT_PCA_3	71.40	67.96	Mejora respecto a PCA_2, aunque con errores notables en varias clases.
DT_PCA_4	100.00	93.37	Muy alto desempeño; métricas equilibradas y rendimiento sólido.
DT_PCA_5	100.00	95.12	Mejor resultado defendible; precisión alta y consistente en todas las clases.
DT_LDA	89.66	89.44	Desempeño sólido y balanceado, aunque inferior a PCA con mayor dimensionalidad.

Los resultados muestran que las variantes **DT_PCA_4** y **DT_PCA_5** ofrecieron un rendimiento competitivo y estable, mientras que **DT_No_PCA**, a pesar de alcanzar valores perfectos, refleja un comportamiento poco realista en escenarios de generalización. La configuración más representativa y defendible corresponde a **DT_PCA_5**, al ofrecer un equilibrio adecuado entre precisión global y estabilidad en la clasificación de las distintas clases.

5.1.4. K-Nearest Neighbors (KNN)

Resultados Comparativos y Evaluación Final

Se entrenaron diferentes configuraciones del modelo *K-Nearest Neighbors* (KNN), incluyendo variantes sin reducción de dimensionalidad, con PCA (2 a 5 componentes) y con LDA. Los resultados de validación cruzada y de prueba independiente se resumen en la Tabla 5.5.

Cuadro 5.5: Resultados de precisión en validación cruzada y en el conjunto de prueba independiente para las variantes de KNN.

Variante	Precisión en Validación Cruzada (80 %)	Precisión en Conjunto de Prueba (20 %)	Observaciones
KNN_No_PCA	100.00	97.38	Mejor desempeño global; métricas macro cercanas al 97.4 %; posible sobreajuste.
KNN_PCA_2	63.34	51.59	Pérdida considerable de información; desempeño muy limitado.
KNN_PCA_3	77.27	70.39	Mejora respecto a PCA_2, aunque aún con errores notables en varias clases.
KNN_PCA_4	100.00	95.00	Alto rendimiento y estabilidad; métricas equilibradas; variante seleccionada como representativa.
KNN_PCA_5	100.00	96.68	Resultados muy altos y consistentes; competitivo frente a No_PCA.
KNN_LDA	90.45	88.78	Desempeño sólido y equilibrado, aunque inferior a PCA con mayor dimensionalidad.

Los resultados muestran que **KNN_No_PCA**, **KNN_PCA_4** y **KNN_PCA_5** alcanzaron los valores más altos de precisión, mientras que **KNN_PCA_2** y **KNN_PCA_3** evidenciaron un desempeño limitado debido a la pérdida de información en la reducción de dimensionalidad. La configuración más representativa corresponde a **KNN_PCA_4**, al ofrecer un equilibrio adecuado entre precisión global, estabilidad y capacidad de generalización en escenarios prácticos.

5.1.5. Naive Bayes

Resultados Comparativos y Evaluación Final

Se compararon distintas variantes del modelo *Naive Bayes* (GaussianNB), incluyendo configuraciones sin reducción de dimensionalidad, con PCA (2 a 5 componentes) y con LDA. Los resultados de validación cruzada y de prueba independiente se resumen en la Tabla 5.6.

Cuadro 5.6: Resultados de precisión en validación cruzada y en el conjunto de prueba independiente para las variantes de Naive Bayes.

Variante	Precisión en Validación Cruzada (80 %)	Precisión en Conjunto de Prueba (20 %)	Observaciones
NB_No_PCA	83.18	83.46	Configuración sólida; métricas equilibradas en torno al 83 %.
NB_PCA_2	46.63	46.46	Pérdida considerable de información; desempeño muy limitado.
NB_PCA_3	51.18	51.08	Ligera mejora respecto a PCA_2, aunque aún insuficiente.
NB_PCA_4	80.58	80.90	Buen rendimiento; métricas consistentes y estables.
NB_PCA_5	77.72	78.26	Resultados aceptables; mejora frente a PCA_2 y PCA_3.
NB_LDA	85.36	85.81	Mejor desempeño global; métricas macro elevadas y balanceadas.

Los resultados muestran que **NB_LDA** alcanzó el mejor desempeño general, con métricas macro elevadas y un comportamiento equilibrado entre clases. Las variantes **NB_No_PCA** y **NB_PCA_4** también ofrecieron resultados sólidos, mientras que **NB_PCA_2** y **NB_PCA_3** evidenciaron un rendimiento claramente insuficiente debido a la pérdida de información en la reducción de dimensionalidad.

Visualización de los Mejores Modelos

Con el fin de consolidar la selección de los clasificadores más representativos, se presentan a continuación los resultados de los mejores modelos de cada familia. La elección de estas configuraciones se basó en el desempeño obtenido en el conjunto de prueba independiente (20 %), ya que este escenario refleja de manera más realista la capacidad de generalización de cada clasificador.

En la Tabla 5.7 se resumen los modelos seleccionados, lo que permite contrastar de manera directa su rendimiento final.

Cuadro 5.7: Resumen comparativo de los mejores modelos de aprendizaje superficial en el conjunto de prueba (20 %).

Modelo	**Precisión en Conjunto de Prueba (20 %)**	**AUC Promedio**
MLP_LDA	93.12 %	0.98–0.99
RF_PCA_4	95.51 %	0.99–1.00
SVM_PCA_5	74.46 %	0.56–1.00
DT_PCA_5	95.12 %	0.96–0.98
KNN_PCA_4	95.00 %	0.98–0.99
NB_LDA	85.81 %	0.91–0.95

Matriz de Confusión y Curvas ROC de los Mejores Modelos

Además de las métricas globales de precisión y AUC, resulta fundamental analizar de manera gráfica el comportamiento de los clasificadores seleccionados. Para ello, se presentan las matrices de confusión y las curvas ROC de los mejores modelos de cada familia.

Las matrices de confusión permiten identificar con claridad en qué clases se concentran los aciertos y dónde se producen los errores de clasificación, revelando patrones de confusión específicos entre categorías. Por su parte, las curvas ROC, junto con el valor del área bajo la curva (AUC), ofrecen una medida visual de la capacidad discriminativa de cada modelo, mostrando el equilibrio alcanzado entre verdaderos positivos y falsos positivos.

Estas representaciones gráficas complementan los resultados numéricos y proporcionan una visión más completa del desempeño de los modelos en el conjunto de prueba independiente (20 %).

Para cada modelo seleccionado se incluyen dos elementos clave de análisis:

- **Matriz de confusión:** Muestra de manera directa los aciertos y errores en la clasificación de cada categoría, permitiendo identificar patrones de confusión entre clases específicas.
- **Curva ROC y AUC:** Representa gráficamente la capacidad discriminativa del modelo, evaluando el equilibrio entre la tasa de verdaderos positivos y la tasa de

falsos positivos. El área bajo la curva (AUC) se utiliza como indicador global de desempeño.

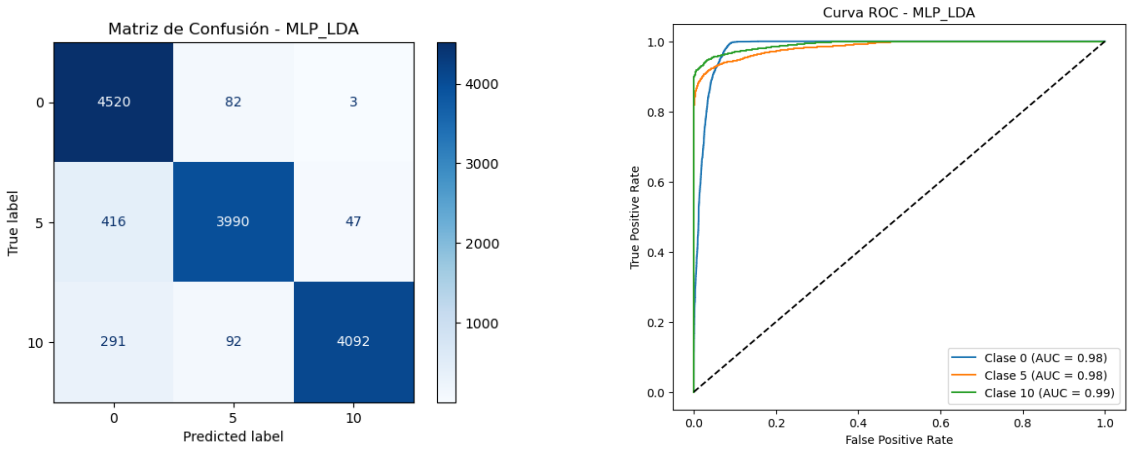
Estas representaciones gráficas refuerzan la justificación de la elección de los modelos seleccionados, al evidenciar no solo su precisión global, sino también su comportamiento frente a las distintas clases del problema de clasificación.

MLP_LDA (MLP) – Evaluación Final

El modelo *Multilayer Perceptron* con reducción de dimensionalidad mediante *Linear Discriminant Analysis* (MLP_LDA) alcanzó una precisión del 93.12 % en el conjunto de prueba independiente, consolidándose como la variante más representativa dentro de la familia MLP.

La **matriz de confusión** (Figura 5.1a) muestra un alto nivel de aciertos con pocos errores entre clases adyacentes, mientras que la **curva ROC** (Figura 5.1b) refleja una capacidad discriminativa elevada, con valores de AUC cercanos a 1.0 en todas las categorías.

Estos resultados confirman el buen equilibrio del modelo entre precisión global y capacidad de generalización.



(a) Matriz de confusión.

(b) Curva ROC.

Figura 5.1: Resultados del modelo **MLP_LDA** en el conjunto de prueba (20 %).

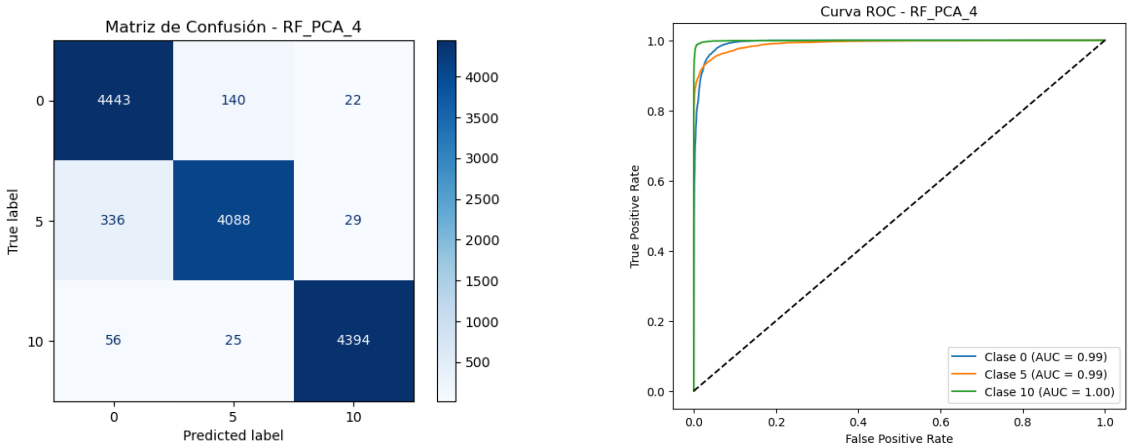
RF_PCA_4 (RF) – Evaluación Final

El modelo *Random Forest* con reducción de dimensionalidad mediante *PCA* (4 componentes) alcanzó una precisión del 95.51 % en el conjunto de prueba independiente,

consolidándose como la variante más representativa dentro de la familia RF.

La **matriz de confusión** (Figura 5.2a) muestra un alto nivel de aciertos con pocos errores entre clases adyacentes, mientras que la **curva ROC** (Figura 5.2b) refleja una capacidad discriminativa sobresaliente, con valores de AUC cercanos a 1.0 en todas las categorías.

Estos resultados confirman la solidez del modelo, combinando precisión global elevada con estabilidad en la clasificación.



(a) Matriz de confusión.

(b) Curva ROC.

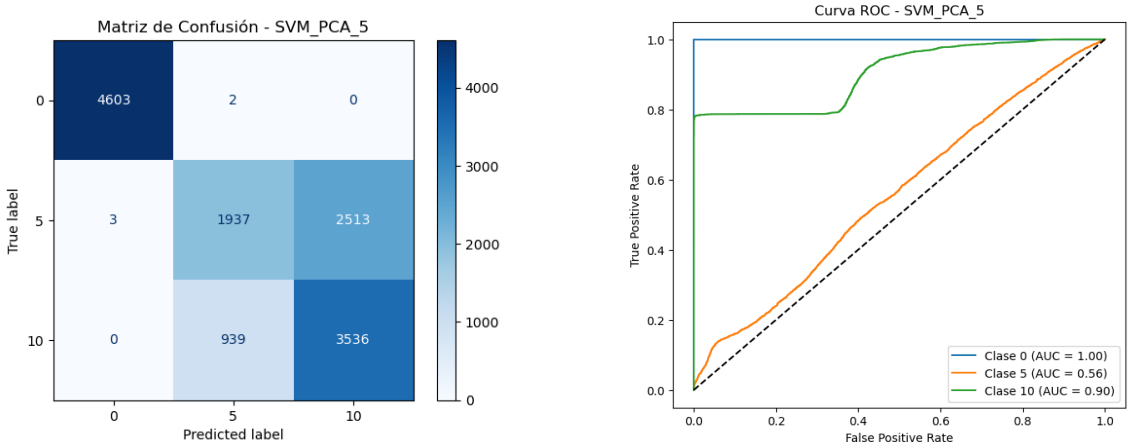
Figura 5.2: Resultados del modelo **RF_PCA_4** en el conjunto de prueba (20 %).

SVM_PCA_5 (SVM) – Evaluación Final

El modelo *Support Vector Machine* con reducción de dimensionalidad mediante *PCA* (5 componentes) alcanzó una precisión del 74.46 % en el conjunto de prueba independiente, consolidándose como la variante más representativa dentro de la familia SVM.

La **matriz de confusión** (Figura 5.3a) muestra un buen desempeño en la clase 0, aunque con errores frecuentes entre las clases 5 y 10, lo que explica la menor precisión global. La **curva ROC** (Figura 5.3b) refleja un comportamiento desigual: AUC=1.00 para la clase 0, AUC=0.90 para la clase 10 y un valor más bajo (AUC=0.56) en la clase 5, evidenciando mayores dificultades en esta categoría.

En conjunto, los resultados posicionan a **SVM_PCA_5** como la opción más adecuada dentro de la familia SVM, al ofrecer el mejor equilibrio posible entre precisión global y estabilidad en la clasificación.



(a) Matriz de confusión.

(b) Curva ROC.

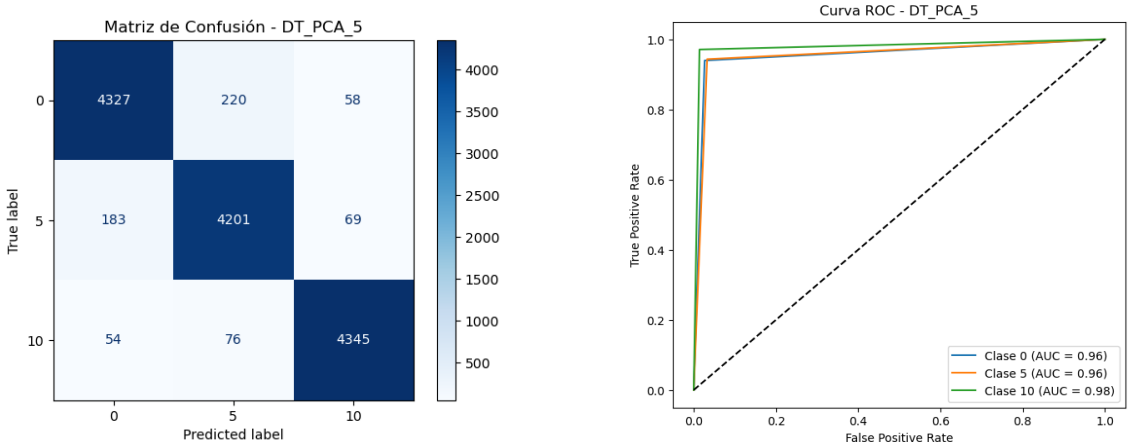
Figura 5.3: Resultados del modelo **SVM_PCA_5** en el conjunto de prueba (20 %).

DT_PCA_5 (DT) – Evaluación Final

El modelo *Decision Tree* con reducción de dimensionalidad mediante *PCA* (5 componentes) alcanzó una precisión del 95.12 % en el conjunto de prueba independiente, consolidándose como la variante más representativa dentro de la familia DT.

La **matriz de confusión** (Figura 5.4a) muestra un alto nivel de aciertos con pocos errores entre clases adyacentes, mientras que la **curva ROC** (Figura 5.4b) refleja una capacidad discriminativa elevada, con valores de AUC entre 0.96 y 0.98 en todas las categorías.

Estos resultados confirman la solidez del modelo, al combinar precisión global elevada con estabilidad en la clasificación.



(a) Matriz de confusión. (b) Curva ROC.

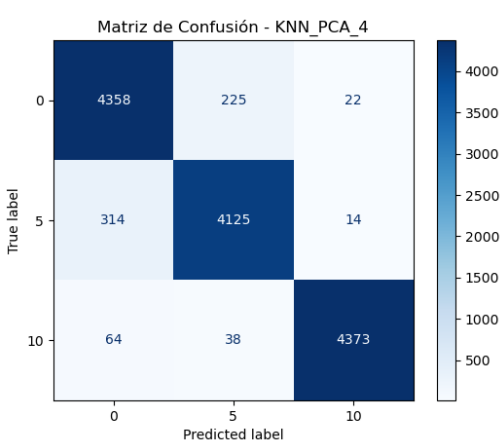
Figura 5.4: Resultados del modelo **DT_PCA_5** en el conjunto de prueba (20 %).

KNN_PCA_4 (KNN) – Evaluación Final

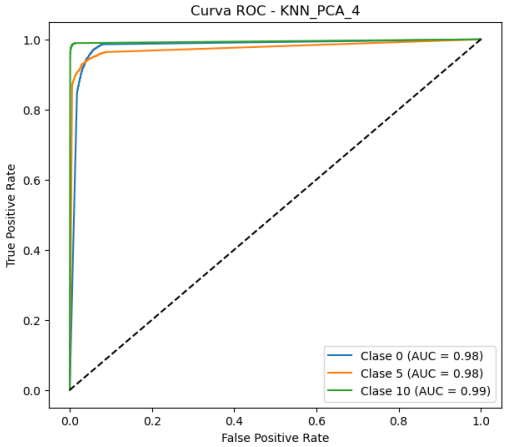
El modelo *K-Nearest Neighbors* con reducción de dimensionalidad mediante *PCA* (4 componentes) alcanzó una precisión del 95.00 % en el conjunto de prueba independiente, consolidándose como la variante más representativa dentro de la familia KNN.

La **matriz de confusión** (Figura 5.5a) muestra un alto nivel de aciertos con pocos errores entre clases adyacentes, mientras que la **curva ROC** (Figura 5.5b) refleja una capacidad discriminativa elevada, con valores de AUC entre 0.98 y 0.99 en todas las categorías.

Estos resultados confirman la solidez del modelo, al combinar precisión global elevada con estabilidad en la clasificación.



(a) Matriz de confusión.



(b) Curva ROC.

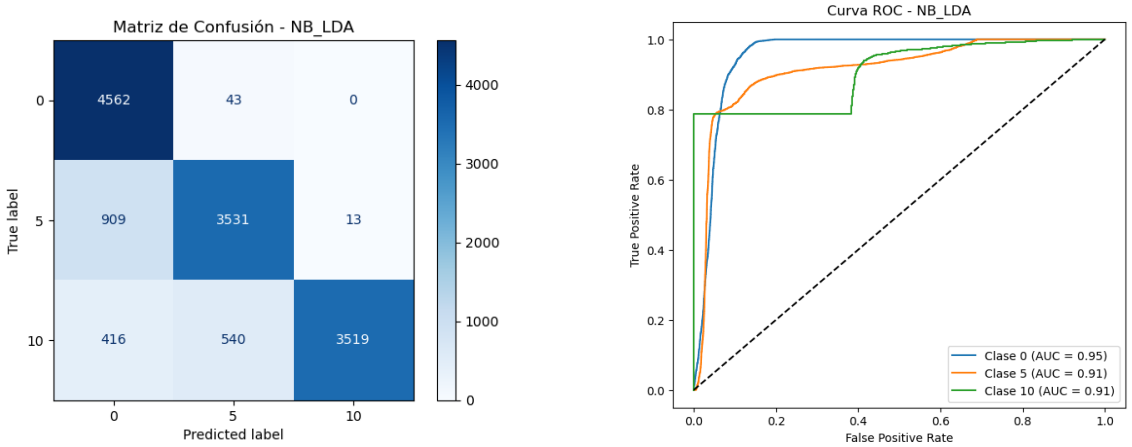
Figura 5.5: Resultados del modelo **KNN_PCA_4** en el conjunto de prueba (20 %).

NB_LDA (NB) – Evaluación Final

El modelo *Naive Bayes* con reducción de dimensionalidad mediante *Linear Discriminant Analysis* (NB_LDA) alcanzó una precisión del 85.81 % en el conjunto de prueba independiente, consolidándose como la variante más representativa dentro de la familia NB.

La **matriz de confusión** (Figura 5.6a) muestra un buen nivel de aciertos, aunque con errores entre las clases 0 y 5, así como entre 5 y 10. La **curva ROC** (Figura 5.6b) refleja una capacidad discriminativa adecuada, con AUC de 0.95 para la clase 0 y de 0.91 para las clases 5 y 10.

En conjunto, los resultados confirman que **NB_LDA** ofrece un desempeño competitivo y equilibrado dentro de su familia de modelos.



(a) Matriz de confusión.

(b) Curva ROC.

Figura 5.6: Resultados del modelo **NB_LDA** en el conjunto de prueba (20 %).

En síntesis, los modelos de aprendizaje automático superficial demostraron un desempeño competitivo, alcanzando precisiones superiores al 95 % en varias configuraciones. Sin embargo, sus limitaciones en la representación de patrones complejos motivan la exploración de un enfoque más robusto.

A continuación, se presentan los **resultados obtenidos con YOLOv8s**, organizados en dos apartados: primero se describen los hallazgos durante el entrenamiento y posteriormente se analizan los resultados finales junto con técnicas de interpretabilidad.

Resultados con YOLOv8s: Análisis e Interpretabilidad

Con el objetivo de comprender en mayor profundidad el comportamiento del modelo YOLOv8s y validar que sus predicciones se basan en características visuales relevantes, se realizaron diversos análisis de interpretabilidad y diagnóstico. Estas técnicas permiten identificar posibles debilidades, confirmar la coherencia de las decisiones internas y reforzar la confianza en el uso del modelo en entornos reales.

Errores Durante el Entrenamiento

Durante el entrenamiento, el modelo generó matrices de confusión internas que evidenciaron ciertas ambigüedades entre clases cercanas, especialmente entre las clases 0 y 5. Las Figuras 5.7 y 5.8 muestran las versiones normalizada y no normalizada, respectivamente, reflejando el proceso progresivo de aprendizaje.

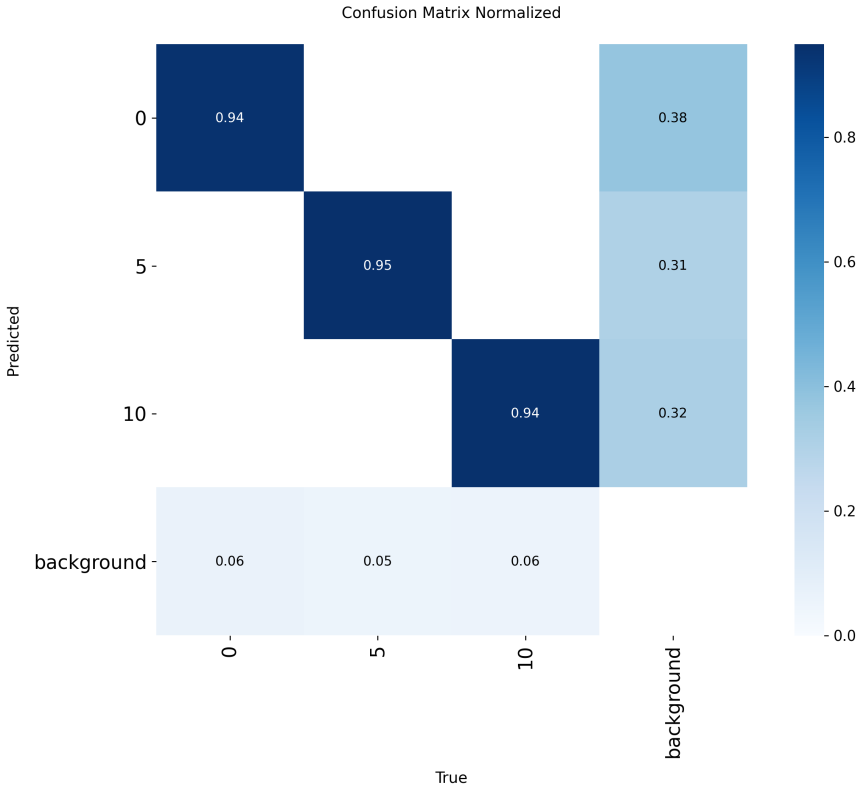


Figura 5.7: Matriz de confusión normalizada generada automáticamente durante el entrenamiento.

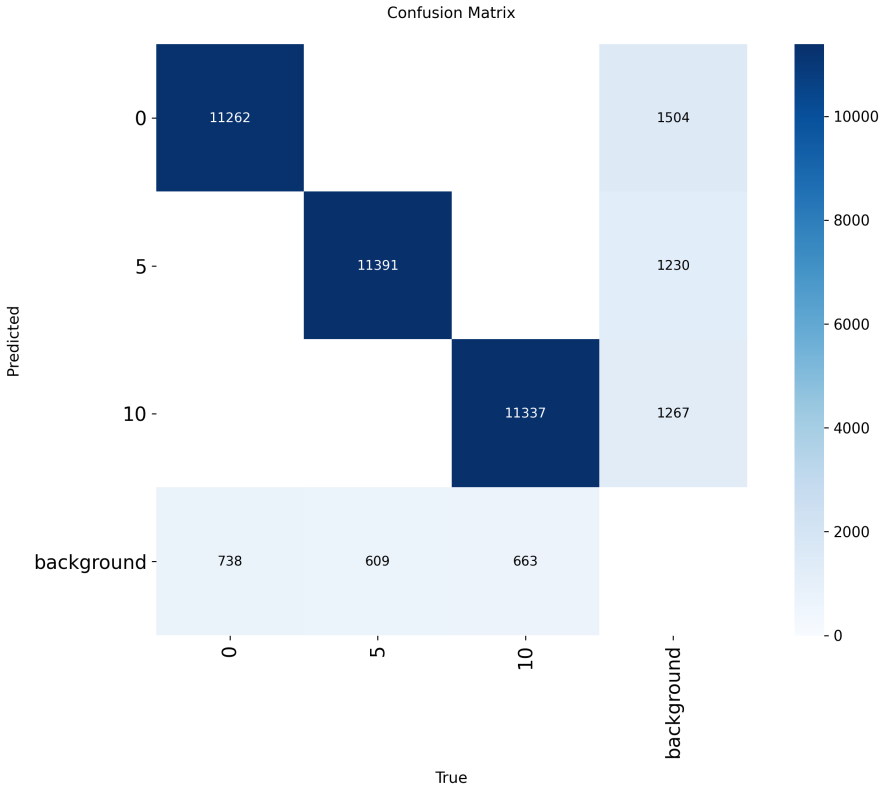


Figura 5.8: Matriz de confusión no normalizada del entrenamiento.

Aunque el rendimiento final fue cercano al óptimo, estas confusiones iniciales evidencian que el modelo logró superar dichas dificultades y alcanzar un desempeño altamente confiable sobre datos nuevos.

Visualización de Activaciones con Grad-CAM

Para verificar que el modelo toma decisiones basadas en regiones faciales relevantes, se aplicó la técnica **Grad-CAM** (*Gradient-weighted Class Activation Mapping*) sobre una imagen del conjunto de validación. Esta técnica genera un mapa de calor que resalta las zonas que más influyen en la predicción.

En la Figura 5.9 se observa que las activaciones se concentran principalmente en la región facial, aunque también existe cierta respuesta en el fondo, lo que sugiere que el modelo considera información contextual sin perder el foco en los rasgos clave.

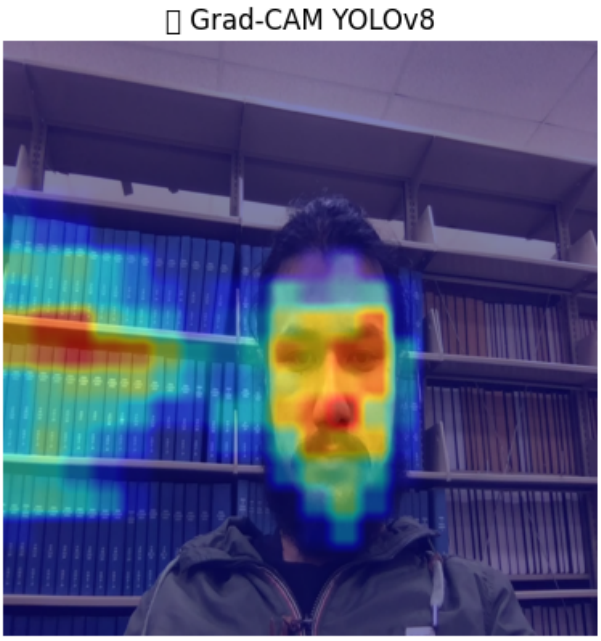


Figura 5.9: Visualización Grad-CAM del modelo YOLOv8 sobre una imagen de validación.

Análisis de Sensibilidad por Oclusión

Como complemento a Grad-CAM, se implementó un análisis de sensibilidad por oclusión (*Occlusion Sensitivity Map*) para evaluar el impacto de cada región de la imagen en la predicción final. El método consiste en oscurecer pequeñas áreas de la imagen y medir la variación en la confianza del modelo.

La Figura 5.10 muestra, a la izquierda, la imagen original y, a la derecha, el mapa de sensibilidad. Las zonas en rojo indican regiones cuya ocultación provoca una mayor disminución en la confianza, destacando la importancia de ojos, cejas y boca.

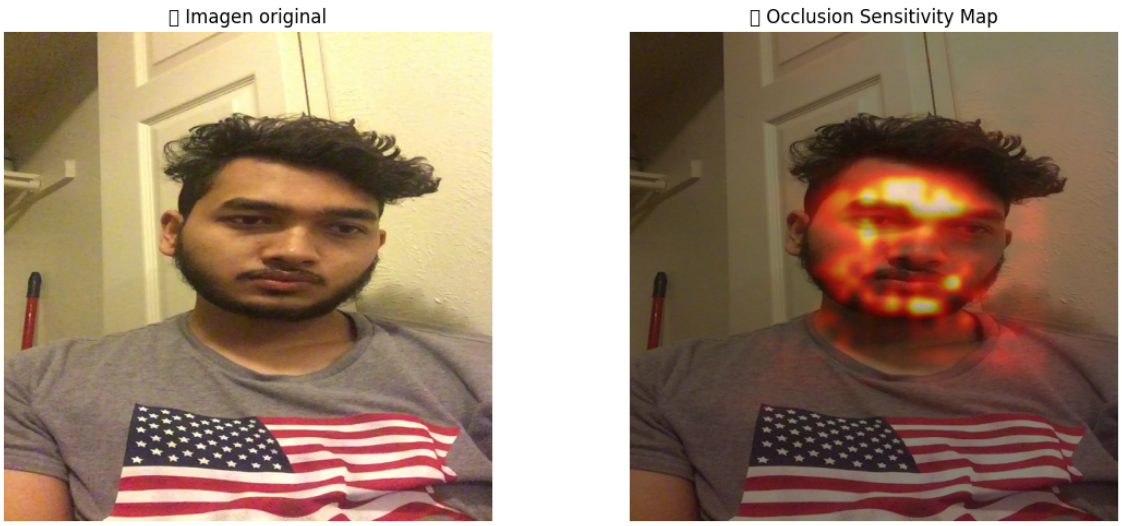


Figura 5.10: Análisis de sensibilidad por oclusión. La intensidad del color rojo indica regiones donde la eliminación causó mayor disminución en la confianza del modelo.

Visualización de Mapas de Activación

Finalmente, se exploraron las representaciones internas aprendidas por el modelo mediante la extracción de mapas de activación de tres capas seleccionadas de la arquitectura YOLOv8, correspondientes a etapas tempranas, intermedias y profundas del procesamiento.

En la Figura 5.11 se presenta, para cada capa, la imagen original y los mapas de activación de distintos canales. Estos canales responden a patrones visuales específicos, desde bordes y texturas hasta estructuras faciales más complejas como ojos, nariz o boca.

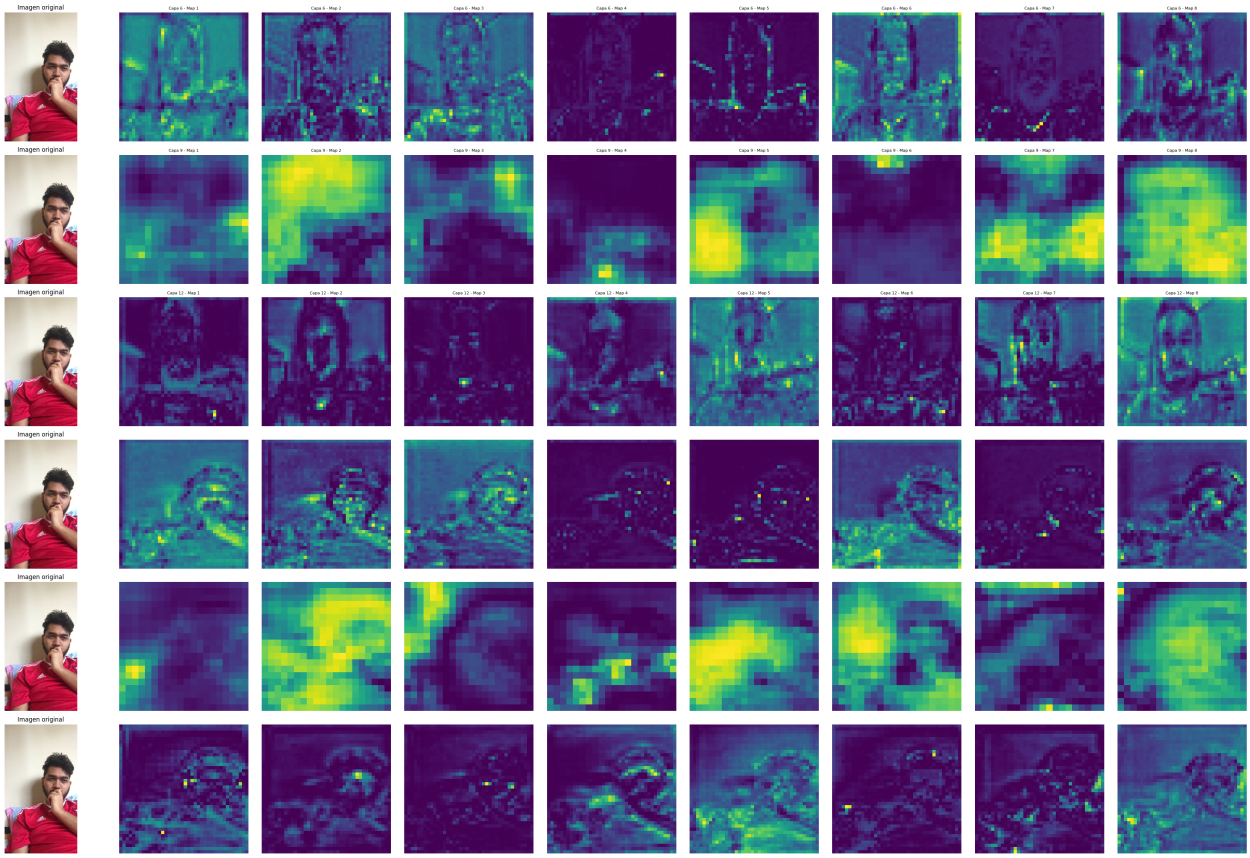


Figura 5.11: Visualización de activaciones internas del modelo YOLOv8 para distintas capas. Cada canal responde a patrones visuales específicos extraídos de la imagen.

Este análisis confirma que el modelo concentra su atención en las regiones faciales más relevantes para la estimación del nivel de somnolencia, lo que respalda la idoneidad del preprocesamiento centrado en el rostro y la coherencia de las predicciones obtenidas.

En síntesis, los modelos de aprendizaje automático superficial demostraron un desempeño competitivo, alcanzando precisiones superiores al 95 % en varias configuraciones y evidenciando un buen equilibrio entre exactitud global y capacidad discriminativa. No obstante, estas aproximaciones presentan limitaciones inherentes a la reducción de dimensionalidad y a la dependencia de características predefinidas, lo que restringe su capacidad para capturar patrones más complejos en las imágenes faciales.

Con el fin de superar estas limitaciones y explorar un enfoque más sólido y flexible, se recurre a técnicas de **aprendizaje profundo**, específicamente a la arquitectura **YOLOv8s**, diseñada para la detección y localización de objetos en imágenes.

Inferencia del Modelo YOLO

Una vez finalizado el entrenamiento y verificado el rendimiento del modelo, se procedió a la fase de **inferencia**, en la cual el modelo YOLOv8s entrenado se emplea para realizar predicciones sobre nuevas imágenes o secuencias de vídeo no vistas previamente. Esta etapa permite evaluar su capacidad para detectar y localizar, en tiempo real, las regiones faciales asociadas a distintos niveles de somnolencia en condiciones similares o diferentes a las del conjunto de entrenamiento.

La inferencia constituye el paso final del flujo de trabajo, ya que traduce el conocimiento adquirido durante el entrenamiento en resultados prácticos y medibles, listos para su integración en aplicaciones o sistemas de monitoreo.

Ejemplos de Inferencia

A continuación se presentan cuatro ejemplos de inferencia sobre imágenes seleccionadas aleatoriamente del conjunto de validación. Cada caso incluye la imagen procesada por el modelo, la clase verdadera (obtenida del nombre del archivo), la predicción realizada y una breve interpretación de los resultados.

Archivo: `0_645948.png` **(Clase verdadera: 0)**

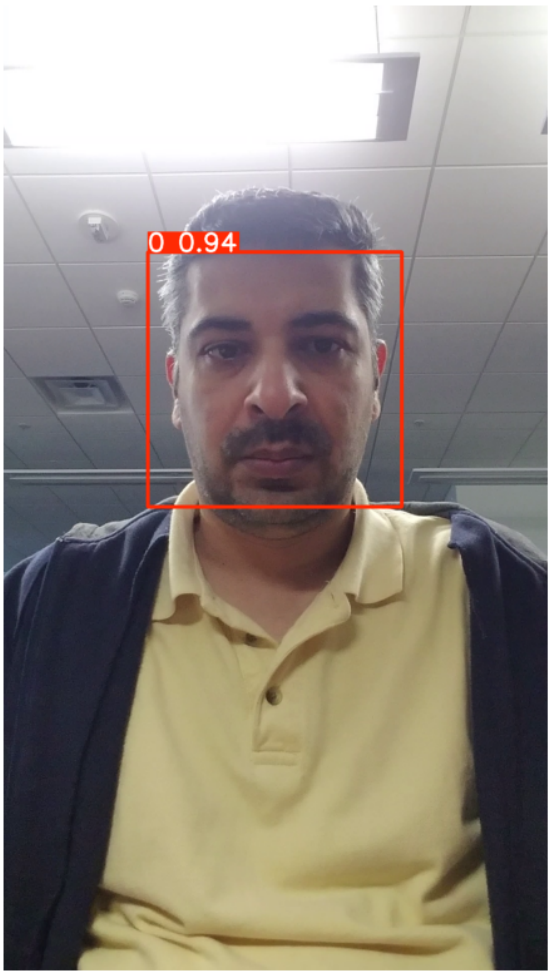


Figura 5.12: Predicción sobre una imagen de clase 0.

Predicción: Clase 0 (confianza: 0.94)

Interpretación: El modelo identificó correctamente la clase 0 con una alta confianza del 94 %, lo que indica una detección clara del rostro correspondiente a este nivel de somnolencia.

Archivo: `5_146044.png` **(Clase verdadera: 5)**

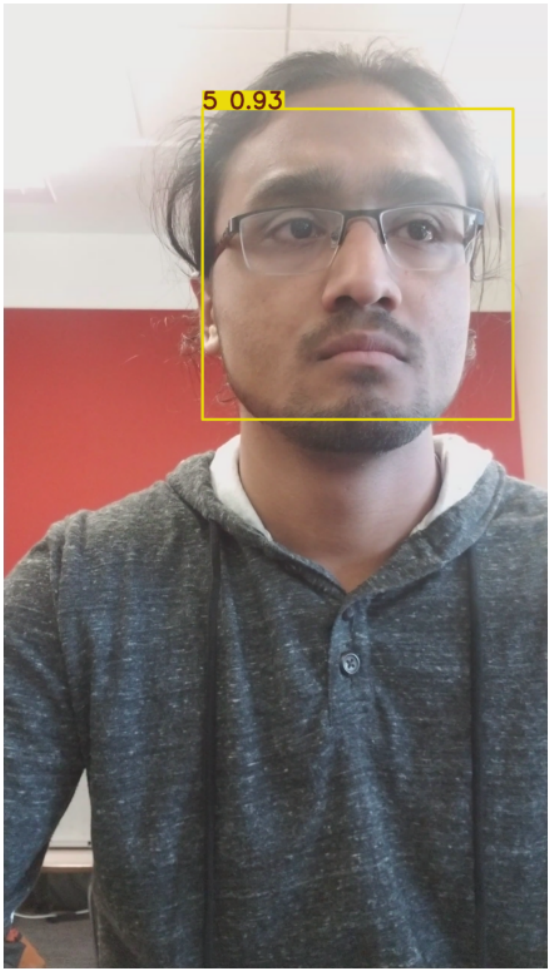


Figura 5.13: Predicción sobre una imagen de clase 5.

Predicción: Clase 5 (confianza: 0.93)

Interpretación: En este caso también se obtuvo una predicción correcta con alta certeza. La región facial fue clasificada como clase 5, correspondiente a un nivel intermedio de somnolencia.

Archivo: `10_572820.png` **(Clase verdadera: 10)**

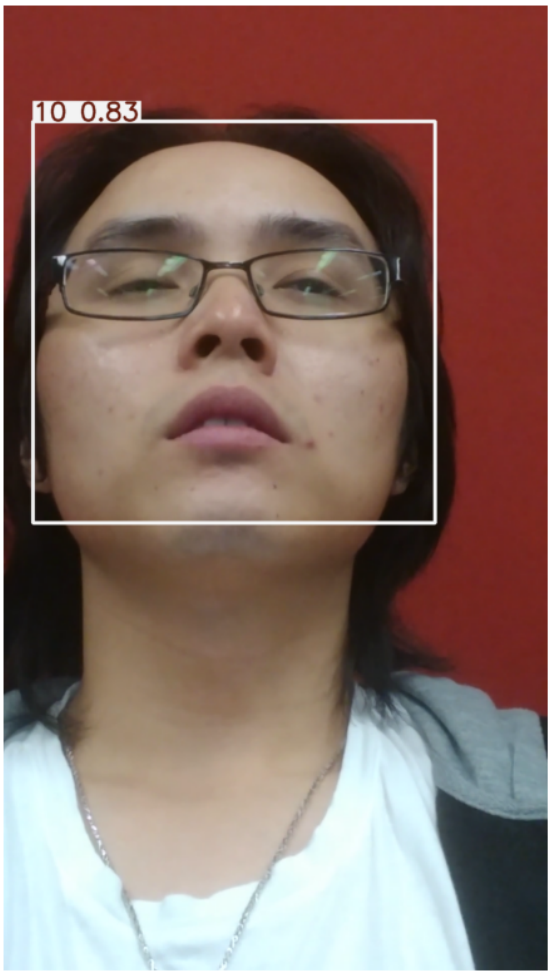


Figura 5.14: Predicción sobre una imagen de clase 10.

Predicción: Clase 10 (confianza: 0.83)

Interpretación: Aunque la confianza fue ligeramente menor que en los ejemplos anteriores, el modelo clasificó correctamente la imagen como clase 10, lo que evidencia su capacidad para reconocer patrones faciales asociados a somnolencia severa.

Archivo: `5_634152.png` **(Clase verdadera: 5)**

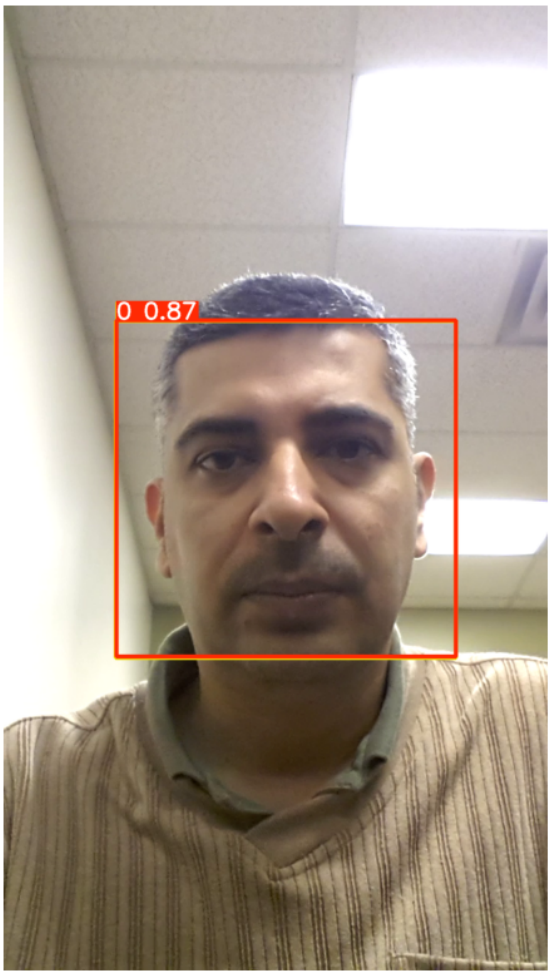


Figura 5.15: Predicción con ambigüedad (clase verdadera: 5).

Predicciones:

- Clase 0 (confianza: 0.87)
- Clase 5 (confianza: 0.45)

Interpretación: En esta imagen se registraron dos predicciones distintas, lo que refleja ambigüedad. Aunque se detectó la clase verdadera (5), esta tuvo menor confianza que una predicción incorrecta (clase 0). Esto sugiere que existen similitudes visuales entre las clases 0 y 5, lo que puede dificultar la clasificación precisa en ciertos casos.

Entrenamiento y Validación Interna

El conjunto de datos empleado para YOLOv8s estuvo conformado por un total de 360 000 imágenes (120 000 por clase). Este se dividió en tres subconjuntos: 80 % para entrenamiento, 10 % para validación y 10 % para prueba.

Durante el entrenamiento, el modelo utilizó únicamente los subconjuntos de entrenamiento y validación. El primero permitió el ajuste de parámetros, mientras que el segundo sirvió para monitorear el proceso de aprendizaje y prevenir sobreajuste. Es importante destacar que el conjunto de prueba se mantuvo completamente independiente y no fue utilizado en esta etapa, reservándose exclusivamente para la evaluación final mediante inferencia.

Evaluación en Conjunto de Validación

Durante el entrenamiento, el modelo fue evaluado de manera periódica utilizando el **conjunto de validación** (10 % del total de datos), el cual permitió monitorear el proceso de aprendizaje y ajustar los parámetros para evitar sobreajuste.

La matriz de confusión obtenida (Figura 5.16) y el correspondiente reporte de clasificación muestran un desempeño sobresaliente, con una precisión global cercana al 100 %.

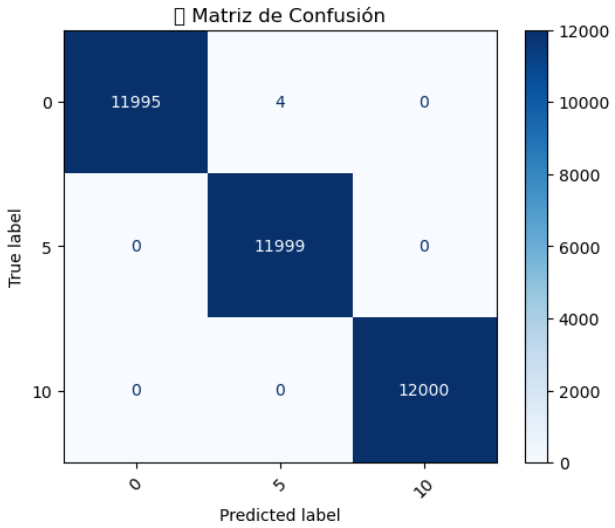


Figura 5.16: Matriz de confusión generada sobre el conjunto de validación.

Reporte de clasificación (validación):

	precision	recall	f1-score	support
0	1.000	1.000	1.000	11999
5	1.000	1.000	1.000	11999
10	1.000	1.000	1.000	12000

accuracy			1.000	35998
macro avg	1.000	1.000	1.000	35998
weighted avg	1.000	1.000	1.000	35998

El modelo cometió solo cuatro errores en más de treinta y cinco mil predicciones, todas entre clases adyacentes. Este comportamiento evidencia una excelente capacidad de aprendizaje y una adecuada sensibilidad frente a diferencias sutiles entre niveles de somnolencia.

Este rendimiento puede explicarse por diversos factores:

- El conjunto de datos fue extenso, balanceado y diverso, lo que permitió al modelo aprender de forma sólida las características propias de cada clase.
- Las anotaciones se centraron exclusivamente en la región facial, eliminando información irrelevante del fondo y mejorando la consistencia del entrenamiento.
- Las clases representan estados visualmente diferenciables (por ejemplo, ojos abiertos vs. cerrados), lo que favorece una clasificación más clara.
- Las condiciones visuales del conjunto de validación (iluminación, ángulos, calidad de imagen) son similares a las del conjunto de entrenamiento, reduciendo el riesgo de desajustes por cambio de dominio.

Evaluación Cuantitativa en el Conjunto de Prueba

Tras finalizar el entrenamiento, se evaluó el rendimiento del modelo **YOLOv8s** utilizando el **conjunto de prueba independiente** (10 % del total de datos, aproximadamente 36 000 imágenes). Este conjunto no fue expuesto durante el entrenamiento ni en la validación, por lo que sus resultados reflejan de manera más realista la capacidad de generalización del modelo.

Figura 5.17 muestra la matriz de confusión correspondiente a esta evaluación. La Figura 5.18 presenta la curva ROC multiclase, mientras que la mientras que la Figura 5.17 muestra la matriz de confusión correspondiente a esta evaluación.

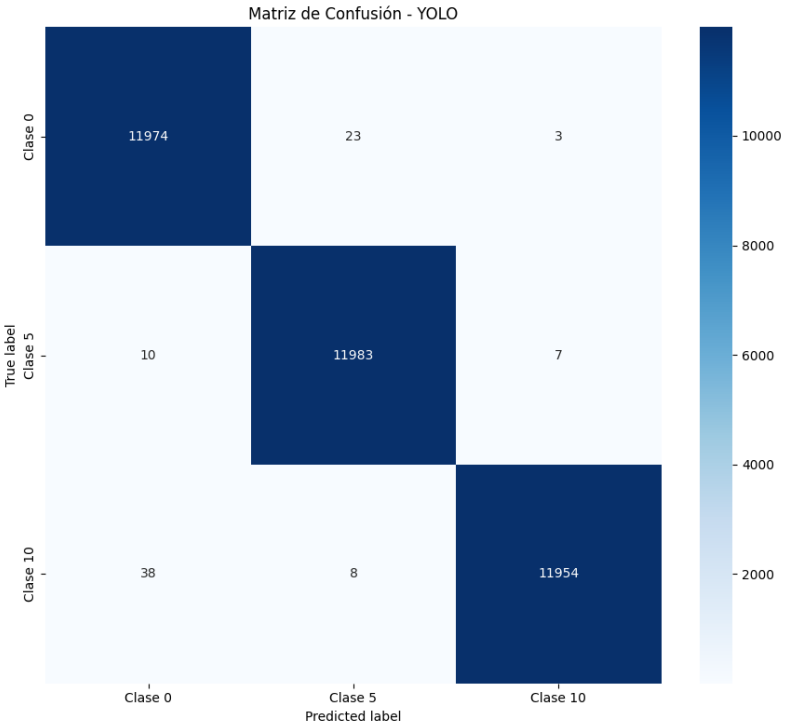


Figura 5.17: Matriz de confusión multiclase en el conjunto de prueba.

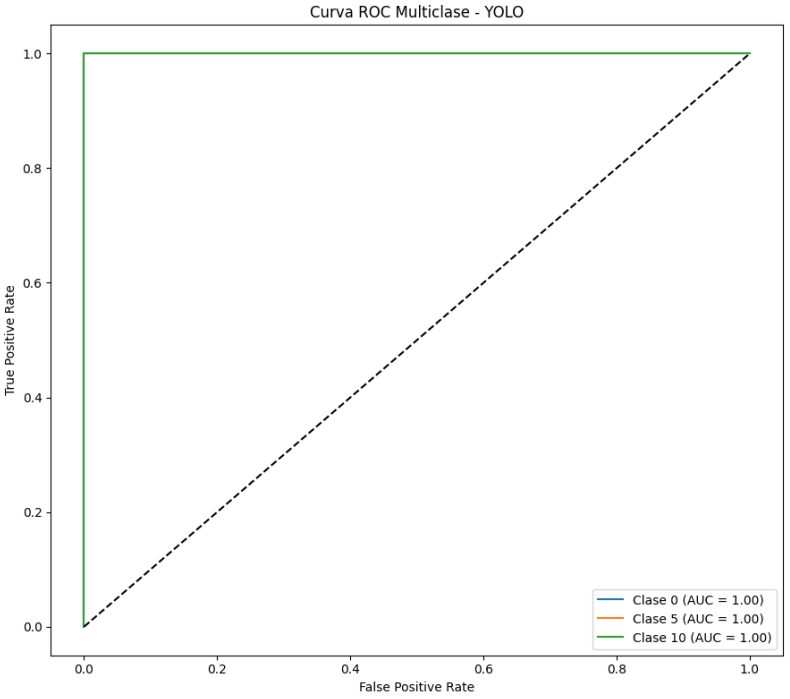


Figura 5.18: Curva ROC multiclase obtenida sobre el conjunto de prueba.

Total de imágenes en prueba: 36 000

Accuracy global: 99.75 %

Los resultados muestran un desempeño sobresaliente: las tres clases alcanzaron valores de AUC de 1.00 en la curva ROC, lo que indica una capacidad discriminativa perfecta. La matriz de confusión confirma este comportamiento, con un número de errores extremadamente reducido (menos de un centenar de casos en más de 36 000 predicciones), concentrados únicamente entre clases adyacentes.

Interpretación: Estos resultados evidencian que el modelo YOLOv8s no solo logró un ajuste adecuado durante el entrenamiento, sino que también mantiene una excelente capacidad de generalización frente a datos completamente nuevos dentro del mismo dominio. Sin embargo, dado que tanto el conjunto de entrenamiento como el de prueba provienen del mismo dataset, es importante considerar que en un **entorno real** podrían presentarse condiciones distintas (iluminación, ángulos, calidad de imagen, variabilidad de sujetos), lo que podría afectar el rendimiento.

En consecuencia, aunque la **accuracy global del 99.75 %** respalda la idoneidad del modelo para la tarea de clasificación de niveles de somnolencia, se recomienda validar su desempeño en escenarios más diversos para confirmar su correcto funcionamiento fuera del laboratorio.

6. Conclusiones

Este trabajo de grado presentó un sistema automático no invasivo para la detección de fatiga en conductores, basado en procesamiento de imágenes faciales y evaluado con el *UTA Real-Life Drowsiness Dataset*. Se exploraron dos enfoques: (i) modelos de aprendizaje automático superficial con métricas geométricas y (ii) un modelo de aprendizaje profundo con YOLO.

En la Tabla 5.7 se resumieron los mejores modelos de aprendizaje superficial, los cuales alcanzaron precisiones entre 74.46 % y 95.51 %, con valores de AUC promedio superiores a 0.95 en la mayoría de los casos. Estos resultados confirman que las métricas geométricas extraídas del rostro (EAR, MAR, PUC, MOE) contienen información suficiente para discriminar distintos niveles de somnolencia con alta fiabilidad.

Los principales hallazgos fueron:

- El enfoque con YOLO, evaluado sobre el conjunto de **prueba** (36 000 imágenes), alcanzó una precisión global del 99.75 %, con **89 errores** concentrados entre clases adyacentes, y curvas ROC con AUC de 1.00 en todas las clases.
- El mejor desempeño entre los modelos superficiales se obtuvo con **RF_PCA_4** (95.51 % de precisión y AUC cercano a 1.0), seguido de **DT_PCA_5** y **KNN_PCA_4**, todos con resultados competitivos.
- Técnicas de interpretabilidad (Grad-CAM, mapas de oclusión, activaciones internas) confirmaron que las predicciones se basan en regiones faciales clave, principalmente ojos y boca.

Estos resultados demuestran que tanto las métricas geométricas como las imágenes completas permiten discriminar niveles de somnolencia con alta precisión. Sin embargo, el uso de YOLO simplificó la arquitectura al integrar en una sola etapa la detección facial y la clasificación del nivel de somnolencia, alcanzando un rendimiento superior.

Como limitación, la evaluación se realizó en condiciones similares a las del entrenamiento, lo que restringe la generalización. Para avanzar hacia aplicaciones reales se recomienda:

- Probar el sistema en escenarios más diversos (iluminación, ángulos, cámaras distintas).

- Incorporar señales adicionales (patrones de conducción, audio o parámetros fisiológicos).
- Desarrollar un prototipo embebido en un vehículo real, evaluando latencia, consumo de recursos y desempeño en tiempo real.

En conclusión, este trabajo valida la viabilidad de una solución basada en visión por computadora para la detección de fatiga en conductores. Los resultados obtenidos evidencian el potencial de aplicar esta tecnología en sistemas avanzados de asistencia al conductor, contribuyendo a una movilidad más segura e inteligente.

Aportes del proyecto

Este trabajo aporta varios elementos relevantes al campo de la detección automática de somnolencia en conductores. En primer lugar, se implementó un esquema de **clasificación en tres niveles** (clases 0, 5 y 10), lo que mejora la granularidad frente a aproximaciones binarias y permite calibrar umbrales de alerta según el nivel de riesgo.

Asimismo, se exploraron **dos enfoques complementarios**: el uso de *características geométricas*, que ofrece un pipeline ligero y portable, y el análisis de *imágenes completas* mediante YOLO, capaz de capturar patrones sutiles sin necesidad de ingeniería manual de rasgos.

Finalmente, se garantizó una **validación reproducible**, con división de datos por dominio, métricas estandarizadas (precisión, *recall*, F1, ROC/AUC), ejemplos de inferencia y análisis de interpretabilidad mediante Grad-CAM y pruebas de oclusión.

Impacto esperado

Los resultados de este proyecto tienen un impacto potencial en distintos ámbitos. En el plano de la **seguridad vial**, pueden contribuir a reducir incidentes asociados a la somnolencia en trayectos prolongados, al habilitar alertas tempranas en cabina. En la **operación de flotas**, se proyecta una disminución de tiempos fuera de servicio y de costos por siniestros, además de servir como base para políticas de turnos y descanso mejor informadas. Finalmente, en el ámbito de la **política pública y la estandarización**, este trabajo ofrece evidencia técnica que puede respaldar la adopción de sistemas no invasivos en el transporte de carga y pasajeros.

Viabilidad técnica

Desde el punto de vista técnico, el sistema se apoya en un **dataset curado**, un preprocesamiento automatizado y pipelines replicables. Los experimentos mostraron un **desempeño elevado** en pruebas controladas, con tiempos de procesamiento por fotograma que sugieren un potencial de respuesta rápida. Además, la **implementación** es compatible con cámaras estándar de cabina y puede beneficiarse de aceleración por GPU cuando esté disponible.

Alcance y límites

El alcance de este trabajo se centra en la **detección no invasiva** basada en el análisis del rostro, con clasificación en tres niveles y operación en condiciones similares al dominio de entrenamiento. Entre los **límites actuales** se identifican la sensibilidad a cambios de dominio (iluminación extrema, oclusiones) y la necesidad de validación en escenarios reales con sujetos no vistos. Como **próximos pasos**, se plantea la realización de pruebas en campo, el ajuste de umbrales según contexto, y la evaluación de falsas alarmas y latencia extremo a extremo (desde la captura de la cámara hasta la emisión de la alerta).

6. Bibliografía

[1] Raja Mohana S P, Manu Vidhya S, Reshma D. *A Real-time Fatigue Detection System using Multi-Task Cascaded CNN Model*. 2021.

[2] Ankit Kumar Yadav, Ankit, Abhilasha Sharma. *Real Time Drowsiness Detection System based on ResNet-50*. 2022.

[3] Marvelous Alexander Panganai, Leslie Kudzai Nyandoro, Kudakwashe Zvarevashe. *Driver drowsiness detection using Convolutional Neural Networks-inspired features and Principal component analysis with K-Nearest Neighbors*. 2022.

[4] Shahzeb Ansari, Fazel Naghdy, Haiping Du, Yasmeen Naz Pahnwar. *Driver Mental Fatigue Detection Based on Head Posture Using New Modified reLU-BiLSTM Deep Neural Network*. 2021.

[5] Muneeb Ahmed, Sarfaraz Masood, Musheer Ahmad, Ahmed A. Abd El-Latif. *Intelligent Driver Drowsiness Detection for Traffic Safety Based on Multi CNN Deep Model and Facial Subsampling*. 2021.

[6] Islam A. Fouad. *A robust and efficient EEG-based drowsiness detection system using different machine learning algorithms*. 2023.

[7] Bagas Aditya Putra, Valerius Owen, Antoni Wibowo. *Machine Learning Trained Drowsiness Detection using Gyroscope on a Microcontroller*. 2023.

[8] Ali Ziryawulawo, Melissa Kirabo, Cosmas Mwikirize, Jonathan Serugunda, Edwin Mugume, Simon Peter Miyingo. *Machine learning based driver monitoring system: A case study for the Kayoola EVS*. 2023.

[9] Islam A. Fouad, Fatma El-Zahraa M. Labib. *Role of Deep Learning in Improving the Performance of Driver Fatigue Alert System*. 2022.

[10] Jitendra Singh Yadav, Vijay Prakash Sharma, Vivek Sharma. *Deep convolutional network based real time fatigue detection and drowsiness alertness system*. 2022.

[11] Salma Anber, Wafaa Alsaggaf, Wafaa Shalash. *A Hybrid Driver Fatigue and Distraction Detection Model Using AlexNet Based on Facial Features*. 2022.

[12] Prasad V. Patil. *MRL Drowsiness Detection Dataset*. Kaggle. 2020. Disponible en: https://www.kaggle.com/datasets/prasadvpatil/mrl-dataset

[13] Ismail Nasri. *Driver Drowsiness Detection Dataset (DDD)*. Kaggle. 2021. Disponible en: https://www.kaggle.com/datasets/ismailnasri20/driver-drowsiness-dataset-ddd

[14] Augmented Startups. *Roboflow Drowsiness Detection Dataset*. Roboflow Universe. 2021. Disponible en: https://universe.roboflow.com/augmented-startups/drowsiness-detection-cntmz/dataset/2

[15] National Highway Traffic Safety Administration. *Drowsy Driving: Avoid Falling Asleep Behind the Wheel*. 2023. Disponible en: https://www.nhtsa.gov/risky-driving/drowsy-driving

[16] Agencia Nacional de Seguridad Vial. *Anuario Estadístico de Seguridad Vial 2024*. 2024. Disponible en: https://ansv.gov.co/sites/default/files/2024-11/Matriz%20Anuario%202024%20-%204.pdf (ver página 30).

[17] Fundación AAA para la Seguridad del Tráfico. *20 % de accidentes mortales en carretera están relacionados con conducir somnolientos*. Macronews. 2023. Disponible en: https://macronews.mx/desarrollo-humano/salud-y-belleza/20-de-accidentes-mortales-en-carretera-estan-relacionados-conducir-somnolientos-

[18] Rohit Hooda, Vedant Joshi, Manan Shah. *A comprehensive review of approaches to detect fatigue using machine learning techniques*. Clean Technologies and Environmental Policy, 23(2), 495–513, 2021. doi:10.1007/s10098-020-01968-1.

[19] Ildar Rakhmatulin. *Review of algorithms for predicting fatigue using EEG*. arXiv preprint arXiv:2402.09443, 2024. Disponible en: https://arxiv.org/abs/2402.09443