



UNIVERSIDAD TECNOLÓGICA ISRAEL
ESCUELA DE POSGRADOS “ESPOG”

MAESTRÍA EN
ELECTRÓNICA Y AUTOMATIZACIÓN
Resolución: RPC-SO-09-No.265-2021

PROYECTO DE TITULACIÓN EN OPCIÓN AL GRADO DE MAGÍSTER

Título del proyecto:
Sistema de alerta de somnolencia por visión artificial para la cooperativa de transporte pesado Gran Colombiana de Ambato.
Línea de Investigación:
Ciencias de la Ingeniería aplicadas a la producción, sociedad y desarrollo sustentable
Campo amplio de conocimiento:
Sistemas Informáticos, desarrollo de software.
Autor/a:
Lalaleo Quispe Edison Mauricio
Tutor/a:
Mg. Wilmer Fabian Albarracín Guarochico

Quito – Ecuador

2025

APROBACIÓN DEL TUTOR



Yo, **Mg. Paúl Francisco Baldeón Egas** con C.I: **100280781-4** en mi calidad de Tutor metodológico del proyecto de investigación titulado: **Sistema de alerta de somnolencia por visión artificial para la cooperativa de transporte pesado Gran Colombiana de Ambato.**

Elaborado por: **Lalaleo Quispe Edison Mauricio**, de C.I: **1804593588**, estudiante de la Maestría: **Electrónica Y Automatización** de la **UNIVERSIDAD TECNOLÓGICA ISRAEL (UISRAEL)**, como parte de los requisitos sustanciales con fines de obtener el Título de Magister, me permito declarar que luego de haber orientado, analizado y revisado el trabajo de titulación, lo apruebo en todas sus partes.

Quito D.M., 10 de septiembre de 2025



Firma

APROBACIÓN DEL TUTOR



Yo, **Mg. Wilmer Fabian Albarracín Guarochico** con C.I: **171334115-2** en mi calidad de Tutor del proyecto de investigación titulado: **Sistema de alerta de somnolencia por visión artificial para la cooperativa de transporte pesado Gran Colombiana de Ambato.**

Elaborado por: **Lalaleo Quispe Edison Mauricio**, de C.I: **1804593588**, estudiante de la Maestría: **Electrónica Y Automatización** de la **UNIVERSIDAD TECNOLÓGICA ISRAEL (UISRAEL)**, como parte de los requisitos sustanciales con fines de obtener el Título de Magister, me permito declarar que luego de haber orientado, analizado y revisado el trabajo de titulación, lo apruebo en todas sus partes.

Quito D.M., 10 de septiembre de 2025

Firma

DECLARACIÓN DE AUTORIZACIÓN POR PARTE DEL ESTUDIANTE



Yo, **Lalaleo Quispe Edison Mauricio** con C.I: **1804593588**, autor/a del proyecto de titulación denominado **Sistema de alerta de somnolencia por visión artificial para la cooperativa de transporte pesado Gran Colombiana de Ambato**. Previo a la obtención del título de Magister en **ELECTRÓNICA Y AUTOMATIZACIÓN**.

1. Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar el respectivo trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Manifiesto mi voluntad de ceder a la Universidad Tecnológica Israel los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor@ del trabajo de titulación, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital como parte del acervo bibliográfico de la Universidad Tecnológica Israel.
3. Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de prosperidad intelectual vigentes.

Quito D.M., 15 de septiembre de 2025

Firma

Tabla de contenidos

APROBACIÓN DEL TUTOR	ii
DECLARACIÓN DE AUTORIZACIÓN POR PARTE DEL ESTUDIANTE	iii
INFORMACIÓN GENERAL.....	1
Contextualización del tema	1
Problema de investigación	3
Objetivo general	6
Objetivos específicos.	6
Vinculación con la sociedad y beneficiarios directos:	6
CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO	8
1.1. Contextualización general del estado del arte	8
1.2. Proceso investigativo metodológico	12
CAPÍTULO II: PROPUESTA	14
2.1 Fundamentos teóricos aplicados.....	14
2.1.2 Hardware del sistema	17
2.1.2.1 Raspberry Pi 5	17
2.1.2.2 Cámara module v2	18
Cámara module v2	19
2.1.2.3 Módulo Buzzer:	19
2.1.3 Software y algoritmos	20

2.1.3.1	Configuración Inicial de la Raspberry Pi 5	20
2.1.3.2	Cargar el Sistema Operativo en la Tarjeta MicroSD	20
2.1.3.3	Configuración Inicial.....	20
2.1.3.4	Acceso remoto y Actualización del Sistema Operativo	21
2.1.3.5	Configuración de la Cámara	21
2.1.3.6	Instalación de Python y Librerías Necesarias	22
2.1.3.7	Verificación de la Versión de Python.....	22
2.1.3.8	Instalación y Verificación de OpenCV.....	22
2.1.3.9	Código para Detección de Rostros y Ojos LANDMARKS	23
2.1.3.10	Cálculos Matemáticos.	24
2.2	Descripción de la propuesta	28
2.3	Validación de la propuesta	33
2.4	Matriz de articulación de la propuesta.....	37
2.5	Análisis de resultados. Presentación y discusión.....	39
CONCLUSIONES		42
RECOMENDACIONES		43
BIBLIOGRAFÍA		44
ANEXOS		47

Índice de Figuras

<i>Figura 1</i> Ojos abiertos y cerrados con puntos de referencia faciales.	14
Figura 2 Placa Raspberry Pi	18
Figura 3 Placa de cámara oficial Raspberry Pi V2	18
Figura 4 Módulo Buzzer.....	19
Figura 5 Suavizado de imagen	25
Figura 6 Puntos faciales.....	27
Figura 7 Diagrama del sistema	28
Figura 8 Diagrama de flujo del sistema	29
Figura 9 Bostezos capturados.....	39
Figura 10 Estado de los ojos	39
Figura 11 Bostezo frecuente.....	40

Índice de Ecuaciones

Ecuación 1	Fórmula de luminancia	24
Ecuación 2	Fórmula Matemática del Filtro Gaussiano	24
Ecuación 3	Proporción de Píxeles Blancos.....	26
Ecuación 4	Proporción de Píxeles Negros.....	26
Ecuación 5	Índice de Relación de Aspecto del Ojo	27
Ecuación 6	Distancia Euclidiana.....	27

Índice de tablas

Tabla 1	Características técnicas cámara v2	19
Tabla 2	Características Técnicas Módulo Buzzer	19

INFORMACIÓN GENERAL

Contextualización del tema

La visión artificial es una parte que abarca la inteligencia artificial desarrollando algoritmos y procesos que las computadoras y otros dispositivos puedan analizar, comprender y procesar información visual, como imágenes y videos, y se usa métodos mediante proceso de imágenes y algoritmos para que los sistemas computacionales puedan "ver" y reconocer patrones, objetos, rostros, textos y otras características en imágenes y videos, tiene diversas aplicaciones útiles en varias áreas entre ellas la detección de objetos, el seguimiento de movimiento, la identificación facial, el análisis de imágenes, la vigilancia y seguridad, entre otra, en resumen, la visión artificial permite a las máquinas "ver" y comprender el mundo visual de una manera similar a como lo hace el ser humano (International Business Machines, 2021).

La Cooperativa de Transporte Pesado Gran Colombiana fue fundada en 1973 por un grupo de 9 socios con el objetivo de trabajar de manera legal en Ecuador. A lo largo de los años la cooperativa fue creciendo de manera significativa y dando el servicio de transporte en la sierra costa y oriente ecuatoriano. La empresa actualmente cuenta con una base sólida para seguir extendiéndose en las diferentes rutas del Ecuador, dando una estabilidad laboral y legal a los socios. Esto se logra gracias al esfuerzo, perseverancia y firmeza de cada uno de los socios con un transporte seguro y responsable que ha contribuido al bienestar de nuestro país (Veronica, 2023).

Para el año 2025 los socios cuentan con su directiva conformada por el Gerente, presidente, tesorero y secretaria quienes promueven el mejoramiento para poder seguir creciendo, optando por incorporar el primer prototipo de alerta de somnolencia por visión artificial esto aporta significativamente a la seguridad de cada transportista de la cooperativa y prevención de accidentes en las diferentes rutas del país. La escalabilidad de este prototipo es de largo plazo para lograr implementar en todos los vehículos este sistema, logrando un mejor control en la seguridad de los

conductores y acompañantes. Con este sistema innovador la cooperativa demuestra su compromiso por el bienestar del sector del transporte pesado.

La toma de decisiones es importante para cambios significativos en la cooperativa, su historia es importante demostrando su compromiso en el bienestar de sus socios con un transporte responsable y seguro. Su perseverancia de crecimiento contribuye al bienestar de los ecuatorianos.

El uso de inteligencia artificial en un sistema de somnolencia resalta en diferentes ámbitos, mejorando la seguridad integral de los conductores y reduciendo accidentes, esto permite realizar los diferentes estudios que permite tener datos en tiempo real con el uso de la I.A. y este tipo de sistemas están en auge e innovadoras esto abre nuevas tecnologías al transporte reduciendo los niveles de siniestro (Flores et al, 2023).

La seguridad del conductor es importante a nivel de todo el mundo debido a la somnolencia sin embargo existen sistemas con limitaciones en su robustez y precisión este campo combina sensores faciales no invasivos y control en tiempo real, en la actualidad existen varios componentes para mejorar los diferentes sistemas de somnolencia y mejorar la seguridad vial y reduciendo los accidentes (Shuyan & Gangtie, 2020)

Los informes de los últimos años realizados en EE. UU. indica que alrededor del 30 por ciento de los accidentes de tránsito son por somnolencia reduciendo su atención y toma de decisiones al momento de conducir, se dan varias propuestas para evitar este tipo de tragedias con el análisis facial en tiempo real de los conductores, siendo esto es una gran innovación tecnológica y teniendo una movilidad inteligente y disminuyendo la siniestralidad en las vías (Siddiqui et al, 2021).

La Organización Internacional del Trabajo indica que alrededor de cada 15 segundos ocurre una muerte por accidentes laborales y esto implica al sector del transporte. Ecuador tiene carencias en seguridad vial los datos demuestran que en un total de 26,291 accidentes la tasa de mortalidad es del 8.48% los diferentes países del mundo implementan espacios adecuados y seguros para el descanso de los conductores, el uso de test de Yoshitake validan las mediciones fisiológicas pero este no es realizado en tiempo real (Medina et al, 2021).

El sector del transporte pesado y el caso de la cooperativa de transporte pesado Gran Colombiana es un área crítica porque trabajan en condiciones de riesgo dando el servicio durante las 24 horas del día con rutas muy extensas de más de horas en diferentes rutas del país esto puede llegar a una depresión psicológica y estrés laboral a esto se suma la mala calidad de las vías climas variables que producen fatiga ocular y esto llevando a gastos adicionales en su economía, para ello se desea implementar un prototipo para alertar al conductor en caso de presentar somnolencia mediante visión artificial el equipo directivo apuesta a esta propuesta para mejorar la integridad siendo los conductores las personas quien ayudará en este plan piloto.

Problema de investigación

La cooperativa de transporte pesado gran colombiana se dedica a transportar carga pesada en parte del Ecuador y se destaca por su eficiencia en sus servicios a lo largo de los años.

Gran parte de los conductores son forzados a laborar jornadas muy extensas para evitar ser víctimas de robos en las carreteras en ciudades con alta peligrosidad delictiva teniendo un descanso de tres horas durante una jornada de trabajo de 24 horas el poco tiempo que tienen para dormir provoca en los conductores fatiga, cansancio y estrés, por lo general siempre tienen la misma rutina de descanso antes de volver a la carretera. Esta práctica es preocupante afectando la capacidad de manejar prudentemente y aumentando el riesgo de accidentes de tránsito.

El poco descanso produce somnolencia en cualquier hora del día a los conductores de transporte pesado, poniendo en peligro la seguridad de cada uno de ellos incluso de otros usuarios de las vías. Con este problema la cooperativa de transporte pesado Gran Colombiana de Ambato propone la implementación de sistema de alerta de somnolencia y reducir el riesgo de accidentes de y salvaguardar la seguridad de los socios y sus unidades.

Cada accidente de tránsito provocado por somnolencia representa un costo muy elevado, provocando un gasto sumamente alto para los socios incluso para las aseguradoras y afectando directamente la economía de las familias. La reparación del vehículo es un proceso largo que también afecta el ingreso económico a esto se suman gastos médicos y produce un descenso

económico por un largo tiempo para sus familias. Para prevenir los accidentes de tránsito por somnolencia se requiere medidas que permitan evitar este tipo de acontecimientos para proteger a los conductores, la economía de las familias y a terceros.

Lo que se destaca de la cooperativa de transporte pesado Gran Colombiana es la calidad de sus servicios y mejoras constantes en tecnologías innovadoras en los diferentes ámbitos del transporte pesado. Sin embargo, saben que el alto costo de inversión puede ser un obstáculo para el progreso y la mejora continua de la empresa. Es por ello que han tomado la decisión de apostar por una propuesta seria y efectiva: el Sistema de Alerta de Somnolencia para conductores mediante Visión Artificial. Esta herramienta es una muestra más de su interés en garantizar la seguridad y el bienestar de sus conductores y pasajeros, al mismo tiempo que les permite optimizar sus operaciones y mejorar su rentabilidad.

El déficit de sueño en conductores constituye un factor en los accidentes de tránsito, cuando los choferes no tienen un adecuado descanso después de una larga jornada laboral, el cerebro no trabaja adecuadamente sobre todo cuando se requiere concentración y esto hace que afecte a los conductores, causando accidentes de tránsito (Gómez Montoya & Cuartas Arias, 2020). Los conductores de la Cooperativa de Transporte Pesado Gran Colombiana no son una excepción. Según los datos de la empresa, al menos un conductor tiene un accidente cada año debido a la somnolencia al volante, lo que pone en peligro no solo al conductor, sino también a terceros, sino que también causa pérdidas económicas para la cooperativa. Por esta razón, la implementación del sistema de alerta de somnolencia para conductores mediante visión artificial es una propuesta necesaria que atiende esta problemática y que, a su vez, puede mejorar la seguridad en carretera.

Si el personal que opera los vehículos de carga continúa laborando sin el apoyo de tecnologías adecuadas, es probable que la siniestralidad vial continúe en aumento, traduciendo esto en significativas pérdidas económicas para esta línea de negocio. Y es que, por un lado, resulta urgente que esta línea de negocio implemente tecnologías que ayuden a proteger la seguridad vial, así como a la reducción de accidentes que, a la postre, no solo trae un alto costo para el operador,

sino para su familia, la empresa, y aseguradoras. De esta manera, es posible proteger y dar seguridad a las personas, conseguir un equilibrio entre la seguridad y la economía en la oferta del transporte pesado.

Se sugiere el diseño de un sistema de alerta que notifique al usuario de la somnolencia basado en visión artificial para la solución de este problema. Este sistema podrá determinar patrones de comportamiento no adecuado de un conductor somnoliento, por ejemplo, parpadeo intermitente y movimientos de la cabeza. Si estos signos son detectados, el sistema emitirá una alerta sonora para que el conductor aplique acciones para prevenir un eventual accidente.

El proyecto prevé la puesta en marcha de un primer prototipo del sistema en la cooperativa Gran Colombiana de Ambato, con una capacidad de expansión a largo plazo. Para su evolución, se emplearán métodos de inteligencia artificial y aprendizaje automático, y se colaborará con conductores expertos para garantizar la eficacia y factibilidad del sistema.

Al implementar este sistema incrementará la seguridad de los conductores, sino que también favorecerá la disminución de gastos relacionados con accidentes de tránsito, la actividad de transportar carga pesada es vital para la economía nacional, no obstante, el agotamiento y la fatiga de los conductores pueden provocar accidentes viales. Por lo tanto, se propone crear un sistema de advertencia de somnolencia a través de la visión artificial para la Cooperativa de Transporte Pesado Gran Colombiana de Ambato. El sistema se proyectará para identificar señales de somnolencia en los conductores.

El sistema estará enfocado en técnicas de visión artificial como algoritmos Haar Cascade y la Red Neuronal Convolucional (CNN), identificando en el rostro signos de somnolencia a través del análisis de imágenes, visión en tiempo real, el dispositivo tendrá instalada una cámara en el tablero acompañada de un Buzzer que se activará cuando el conductor presentes signos de somnolencia al conducir,

Este primer prototipo será instalado en un vehículo de la cooperativa Gran Colombiana y con alcance a largo plazo, para tener una efectividad se aplicará métodos de I.A. y la colaboración de los conductores de carga pesada.

Objetivo general

Desarrollar un sistema de alerta de somnolencia por visión artificial para la cooperativa de transporte Gran Colombiana de Ambato.

Objetivos específicos.

- Contextualizar los fundamentos teóricos sobre sistemas de alertas de somnolencia en conductores de la Cooperativa de Transporte Pesado Gran Colombiana de Ambato mediante Visión Artificial.
- Diagnosticar la situación real sobre la problemática de la cooperativa de transporte pesado gran colombiana de Ambato
- Implementar un sistema por visión artificial para detectar la somnolencia en un camión de la empresa.
- Validar la implementación del sistema de alerta de somnolencia mediante pruebas de funcionamiento.

Vinculación con la sociedad y beneficiarios directos:

El alto índice de accidentes de tránsito asociados a la fatiga y somnolencia en los choferes del transporte pesado representan un problema de salud pública y vial, los accidentes causan discapacidad y mortalidad en las vías, Ecuador es el segundo país con mayor mortalidad de accidentes de tránsito en América latina, generando grandes cargas económicas y sociales para el país (Oviedo et al, 2025).

La implementación de un sistema de alerta de somnolencia mediante visión artificial para la Cooperativa de Transporte Pesado Gran Colombiana de Ambato surge como una respuesta a esta

problemática dando seguimiento en tiempo real el estado del conductor mediante una cámara el mismo que verificará patrones de somnolencia.

El sistema identifica signos de somnolencia, cansancio, parpadeo prolongado de los ojos bostezos inusuales en la persona y cabeceos repetitivos muy notorios el sistema enviará una señal sonora que permitirá indicar al conductor que está en un estado no adecuado para conducir.

Esta alerta indicará de manera oportuna al conductor que debe descansar esto permitirá reducir los accidentes de tránsito producidos por la somnolencia el sistema no es invasivo de bajo costo y fácil instalación este prototipo es apto para cualquier vehículo.

Los costos de este tipo de sistemas pueden ser limitantes para los conductores, pero al ser un prototipo de bajo costo y a la par con la tecnología se podrá obtener resultados favorables para los conductores (Alemán & Pujota, 2024)

El sistema tiene un enfoque educativo tecnológico y social, fomentando a los conductores una estrategia viable para la prevención de accidentes por estar en un estado de cansancio fatiga estrés entre otros, esto permitirá proteger al conductor y terceros, Se estima que el uso del sistema disminuirá los accidentes teniendo un ahorro positivo para la cooperativa seguros y evitar pérdidas humanas.

En la actualidad no existe un sistema de monitorización facial integrado en los vehículos de transporte dando una señal que permita a los conductores prevenir un accidente después de largas horas de trabajo. (Azli Abdullah et al, 2023)

El sistema es un proyecto tecnológico y amigable con el sector del transporte pesado permitiendo, salvaguardando la seguridad de los transportistas su implementación ayudará a disminuir los accidentes de tránsito por somnolencia con un costo accesible.

La visión por computadora es un sistema es un método muy factible adicionado componentes electrónicos para poder evitar accidentes de tránsito e ir ayudando en esta problemática siendo una estrategia muy innovadora (Delgado & Yandún, 2022)

CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO

1.1. Contextualización general del estado del arte

Al analizar el problema hay una situación crítica sobre los accidentes de tránsito relacionados con la somnolencia en conductores de transporte pesado, los objetivos específicos que abarcan en este proyecto abarca la elaboración e implementación del sistema de alerta de somnolencia para la cooperativa de transporte pesado gran colombiana de Ambato. El proyecto se limita a la detección de somnolencia por medio de algoritmos para detectar parpadeo no adecuado de los ojos o bostezo utilizando visión artificial tomadas por medio de una cámara ubicada en el vehículo, se aplicará algoritmos como el clasificador Haar Cascade y el cálculo del Eye Aspect Ratio (EAR). El proyecto está sustentado con documentos actualizados desde el año 2020 en diferentes revistas científicas y repositorios universitarios, los resultados se podrán aplicar en la prevención de accidentes mediante el sistema de alerta de somnolencia con el monitoreo en tiempo real.

Para el año 2022, la Agencia Nacional de Tránsito tuvo un registro 21,739 accidentes de tránsito y se estima que alrededor del 10% de los accidentes son provocados por la somnolencia, ubicando así en el tercer lugar al transporte pesado según las estadísticas (ANT, 2022). Con las estadísticas emitidas por la ANT la somnolencia es un factor que provoca los accidentes de tránsito afectado a conductores de diferentes tipos de vehículos, por lo tanto, es fundamental que los conductores de transporte pesado adopten medidas preventivas contra la somnolencia. Al implementar el sistema de somnolencia mediante visión artificial trata de mejorar la seguridad vial en el transporte pesado.

A nivel mundial los siniestros viales se posicionan como la quinta causa de muertes, factores como: somnolencia, fallas mecánicas, mal estado de la red vial, exceso de velocidad entre otras contribuyen a estas causas. En estudios realizados aproximadamente el 40% de los conductores no descansan adecuadamente disminuyendo su atención al volante por sus largas jornadas de trabajo aumentando el riesgo de accidentes de tránsito, concluyendo que el 20% de los accidentes es relacionado por la somnolencia, dando la necesidad de instalar instrumentos tecnológicos

preventivos como el sistema de alerta de somnolencia por medio de visión artificial usando microcontroladores, bibliotecas OpenCV entre otros con un contexto experimental a esto se suma diversos tipos de algoritmos basados en visión artificial con el fin de detectar somnolencia Baiza Lovato (2020).

En el artículo presentado por (Flores Monroy et al , 2023), plantea la detección de somnolencia en conductores por medio de visión artificial usando la cámara de un dispositivo celular este permitirá tomar capturas del conductor y aplica algoritmos Viola-Jones para detectar el rostro del conductor y analizadas mediante redes neuronales y el sistema activará una alarma si detecta somnolencia del conductor para evitar posibles accidentes de tránsito dando como resultado una precisión del 95.8% de efectividad.

El estudio propuesto por (Azli Abdullah et al, 2023) en el sistema de detección de fatiga en conductores realizado por la Universidad Multimedia, Malaca, Malasia, utilizando dispositivos tecnológicos como la Raspberry Pi y una cámara y usando visión por computadora y algoritmos y librerías (OpenCV, Haar Cascade y Eye Aspect Ratio) para monitorear signos de somnolencia. El sistema, logró un 90% de precisión, alertando al conductor con una alarma sonora cuando detecta somnolencia durante tres segundos, demostrando ser una solución efectiva y accesible para reducir accidentes causados por somnolencia al volante.

El documento de Luis Sinche, se implementó un sistema avanzado de asistencia al conductor (ADAS) con el propósito de detectar distracción y somnolencia mediante visión artificial y EAR (Eye Aspect Ratio) para somnolencia y desconexión visual (distracción) mediante dos comparaciones de perímetro ocular. Para el sistema se utilizó una cámara Logitech de 21 fps, y mediante OpenCV el sistema se implementó en Python 3 En pruebas de 60 minutos, se logró un 87% de precisión con emisión de alertas en 1.5 y 2 segundos. La investigación, desarrollada en la Universidad Nacional de Loja (Ecuador), resaltó el potencial de la tecnología no invasiva y de bajo costo para mitigar

accidentes, en un contexto donde Ecuador abarca el 26% de siniestros en el 2021. (Sinche Cueva, 2023)

El trabajo abordado por (Cárdenas Vera, 2022), con el tema “Sistema de realidad aumentada para mantenimiento preventivo del sistema de generación en la central hidroeléctrica Sopladora” aborda una problemática de documentación impresa o digitales para mantenimiento preventivos y para resolver esta problemática desarrolló un sistema de realidad aumentada usando EcoStruxure Augmented Operator Advisor que permitió visualizar información técnica en los teléfonos móviles aportando soluciones tecnológicas, También indica que la realidad aumentada ayuda considerablemente a diversos problemas en la industria dando una efectividad positiva. Con este aporte nuestro sistema cuenta con visión artificial para detección de somnolencia mediante algoritmos.

En el trabajo de titulación de (Changotasig & Albarracín, 2023) con el tema "Procesamiento Digital de Imágenes Mediante Inteligencia Artificial para la Detección de Accidentes de Tránsito en Quito" tiene la problemática de los accidentes por fatiga y propone una solución automatizada para el mismo. Utilizando redes neuronales convolucionales (YOLOv5) y herramientas como Python y OpenCV, teniendo una efectividad del 90% de precisión en la identificación de vehículos y un 80% en accidentes, reduce la carga operativa y mejora la seguridad vial, aunque persisten desafíos técnicos, como la compatibilidad con protocolos cerrados de cámaras y la necesidad de hardware especializado para un procesamiento más eficiente.

La publicación de (Baiza & Jurado, 2020) con el tema propuesto “SISTEMA DE DETECCIÓN Y ALERTA DEL ESTADO DE SOMNOLENCIA DE CONDUCTORES MEDIANTE VISIÓN ARTIFICIAL” desarrolla la detección de somnolencia en conductores utilizando la visión por computadora, mediante el uso de técnicas de procesamiento de imágenes y algoritmos de inteligencia artificial. La problemática son los accidentes producidos por el estado no adecuado de los conductores como la somnolencia. El

problema se resuelve creando un sistema de alerta de somnolencia que incluye una Raspberry Pi, una cámara, y el uso de librerías OpenCV y Dlib para el análisis de parpadeo y otros signos de fatiga. Los resultados mostraron que el sistema es efectivo para la detección de somnolencia en un 91%, con emisión de alertas visuales y sonoras. Este sistema en particular complementa la propuesta que manejo, el cual es a un bajo costo, un método práctico para la seguridad que busca optimizar la fatiga en la conducción.

El sistema que se implementará tendrá una cámara que permitirá observar al conductor en tiempo real mediante visión artificial y analiza los rasgos faciales para identificar signos de somnolencia, como bostezo prolongados cierre de ojos al detectar estos parámetros se activará una alerta sonora para que el conductor tome medidas preventivas en comparación al proyecto anterior este se aplicará en el transporte pesado compartiendo sistemas basados en visión artificial.

El análisis de las investigaciones precedentes revela una convergencia hacia soluciones basadas en visión artificial con eficiencias superiores al 90%, validando la viabilidad técnica del enfoque propuesto. Las perspectivas de investigación indican la necesidad de abordar problemas no resueltos identificados en trabajos anteriores: la optimización de algoritmos para condiciones variables de iluminación, la integración de módulos de inteligencia artificial para detección de múltiples tipos de distracción, y la mejora de sistemas de grabación nocturna. Las últimas tecnologías identificadas incluyen el uso de cámaras, procesamiento en tiempo real con Raspberry Pi, y algoritmos adaptativos de OpenCV. Los resultados de investigaciones anteriores permiten clasificar las soluciones en tres categorías: sistemas portátiles de bajo costo, dispositivos integrados vehiculares, y plataformas de monitoreo centralizado. Como valor agregado, el presente proyecto puede incorporar la validación específica en condiciones del transporte pesado ecuatoriano, la adaptación a rutas de montaña características de la región de Ambato, y la integración con sistemas de gestión de flotas cooperativas. Las decisiones tecnológicas se orientarán hacia la implementación de sistemas robustos ante variaciones ambientales, eficientes energéticamente, y escalables para aplicación masiva en el sector del transporte pesado nacional.

1.2. Proceso investigativo metodológico

Enfoque de la investigación

La investigación adoptará un enfoque mixto combinada de elementos cuantitativos: donde se enfocara en el análisis de datos validados sobre accidentes de tránsito y tener una línea base al problema, se validará acorde a los objetivo mediante la efectividad del sistema en base a proyectos ya elaborados para evaluar sucesos de somnolencia por otra parte los elemento cualitativos es el análisis del comportamiento de los conductores y cómo afecta el desempeño al volante también se realizará una pequeña evaluación de la aceptación del sistema propuesto.

Tipo de investigación

El sistema propuesto es una investigación aplicada y de desarrollo experimental reduciendo los accidentes de tránsito relacionados a la somnolencia en el transporte pesado dando mediante el desarrollo de un prototipo funcional dando soluciones tangibles

El proyecto implica el diseño e implementación, incorporando algoritmos de visión artificial, validándose en condiciones reales de operación empleando una investigación descriptiva por el estado de somnolencia, explicativa al analizar los factores y causas y dar una solución.

Población Y Muestra

La población se compone por 25 conductores de la Cooperativa de transporte pesado Gran Colombiana, enfocándose específicamente en los socios activos. Los conductores poseen licencias profesionales y poseen la experiencia en el transporte pesado operando en el Ecuador con una jornada de trabajo de más de 20 horas diarias en algunos casos, el diseño muestral es de tipo no probabilístico intencional, seleccionando participantes que cumplan criterios específicos, tener al menos dos años de experiencia, haber otorgado su consentimiento informado y disponibilidad para participar. Se excluyen aquellos con problemas visuales. La muestra se distribuye por fases: en la fase de validación técnica se trabajará con 10 conductores (28.6 % de la población), seleccionados por representatividad de rutas y horarios; en la fase de pruebas piloto se incluirán conductores (42.9 %), con distintos niveles de experiencia; y en la fase de implementación se comenzará con un vehículo equipado con el prototipo.

Métodos, técnicas e instrumentos

Desde la perspectiva de la teoría, se aplican métodos analítico-sintético, inductivo-deductivo e histórico-lógico para descomponer el fenómeno, dando soluciones tecnológicas.

Se aplicará el método experimental con pruebas en el sistema como iluminación ángulo de la cámara y tiempos de respuesta, se realizará una observación en el comportamiento de los conductores utilizando técnicas de observación directa y entrevistas semi estructurada, y encuestas validadas y aplicadas en otros documentos, se tendrá un análisis mixto el cuantitativo (estadística descriptiva, inferencial y series temporales) y cualitativos (análisis de contenido y fenomenológico).

Los dispositivos o instrumentos que se propone utilizar son: hardware incluye un Raspberry Pi 5, una cámara y un módulo buzzer en el software tendremos Python librerías OpenCv que permitirá programar algoritmos para detectar la somnolencia. El proceso metodológico tendrá: fundamentación teórica, diseño e implementación y se finalizará con las pruebas y validación.

CAPÍTULO II: PROPUESTA

2.1 Fundamentos teóricos aplicados

La implementación de un sistema de alerta de somnolencia por visión artificial para conductores de transporte pesado de la Cooperativa Gran Colombiana de Ambato es una propuesta con tecnología actual, integrando sistemas embebidos, procesamiento de señales entre otros. Esto permitirá reducir los accidentes viales producidos por la somnolencia ayudando con soluciones preventivas en los ejes viales.

Visión Artificial

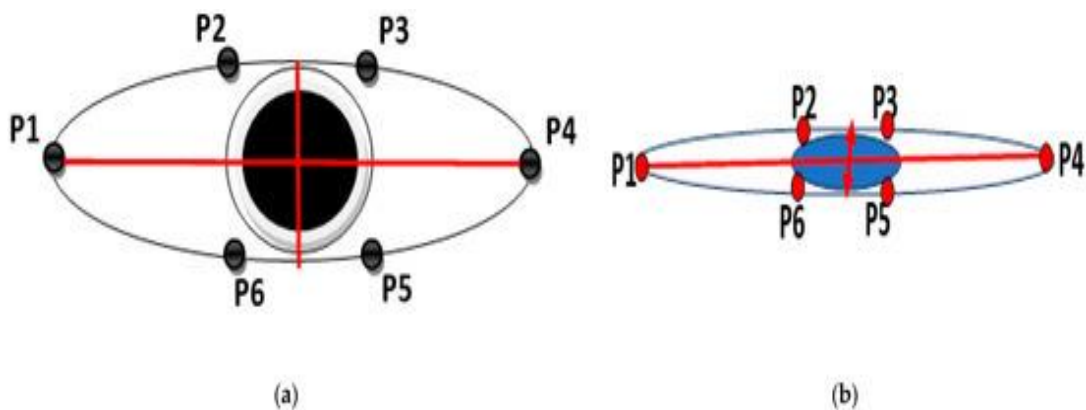
La visión artificial es comprender el medio a lo más parecido al ojo humano y pueda dar instrucciones de acuerdo a lo que visualiza con un aprendizaje autónomo intentando simular a una persona mediante redes neuronales comprendiendo texto e imágenes para mejorar los algoritmos de visión por computadora (Sánchez, 2022)

Relación del aspecto del ojo (EAR)

Las técnicas de visión artificial se centran en detectar los rostros mediante algoritmos Eye Aspect Ratio (EAR). El número de parpadeos por minuto varía entre 6 y 30 y varía según su entorno, género, edad y el tamaño de sus ojos. El algoritmo requiere cálculos básicos basando en los patrones de los ojos siendo eficiente y fácil de aplicar (Dewi et al, 2022).

Figura 1

Ojos abiertos y cerrados con puntos de referencia faciales.



Nota. Adaptado de Relación de aspecto ocular para la detección de somnolencia en tiempo real y mejora la seguridad del conductor (p. 5), por Dewi et al, 2022, Electronics.

En cada fotograma de vídeo, se detectan los puntos de referencia oculares. Se calcula la relación de aspecto ocular (EAR) entre la altura y la anchura del ojo como se muestra en la siguiente ecuación (Soukupova & Cech, 2021).

$$EAR = \frac{||p2 - p6|| + ||p3 - p5||}{2||p1 - p4||}$$

Donde:

P1 y **P4** son los extremos del ojo esquina externa e interna, **P2** y **P6** puntos externos del párpado superior e inferior, **P3** y **P5** puntos internos del párpado superior e inferior (Dewi et al, 2022).

Algoritmo Haar Cascade

El algoritmo Haar Cascade permite detectar los rostros desde diferentes ángulos de disparo y en diferentes condiciones climáticas y de iluminación dando una buena precisión a una distancia máxima de un metro (Bahit et al, 2023).

Las imágenes son matrices de tres dimensiones por el ancho alto y color para ser más efectivo el algoritmo se debe tomar escala de grises para trabajar en dos dimensiones el método Haar Cascade tiene tres etapas: primero buscar características haar segundo clasificador boosting y tercero conecta en cascada los clasificadores, Haar Cascade es gratuito y está ubicado en un repositorio OpenCV, con clasificadores previamente entrenados (Oliveira et al, 2021).

Las máquinas se están volviendo más amigables y pueden reaccionar con mayor rapidez ante las personas. La detección de rostros humanos ha despertado un gran interés en los últimos años y se ha convertido en un factor vital en la interacción hombre-máquina, además de ser uno de los temas más populares en visión artificial y biometría. El conjunto de datos de investigación Labeled Faces in the Wild (LFW) se utiliza con el apoyo de la biblioteca OpenCV y la programación en Python (Salh et al, 2022).

El sistema funcionara con una Raspberry pi 5 que es un mini computador con memoria RAM, con puertos de entrada y salida y un procesador que al ser programada ejecutará los algoritmos para detectar el rostro, Raspberry pi 5 es un dispositivo de bajo costo y usado en medios educativos, investigativos y de industrias a esto se añade una cámara V2 de 8Mp que nos dará la imagen en tiempo real y un módulo buzzer que activará una alerta sonará al momento de detectar somnolencia

Se aplicarán algoritmos para reconocer patrones faciales, las imágenes serán analizadas en tiempo real mediante modelos entrenados para verificar si los ojos se encuentran abiertos o cerrados y detectar una posible somnolencia.

Los enfoques más relevantes serán de manera cualitativos con un diseño experimental utilizan los vehículos de transporte pesado, recopilando datos que permitirá evaluar la efectividad del sistema en situaciones reales.

En cuanto a las normativas, este trabajo se enmarca en la Ley Orgánica De Transporte Terrestre Tránsito Y Seguridad Vial, (2018) establece “Art. 88 literal g) Disponer la implantación de requisitos mínimos de seguridad para el funcionamiento de los vehículos, de los elementos de seguridad activa y pasiva y su régimen de utilización, de sus condiciones técnicas y de las actividades industriales que afecten de manera directa a la seguridad vial” (p.28). bajo estos parámetros los conductores tendrán condiciones más seguras para trabajar, el proyecto se alinea a políticas propuestas por las entidades correspondientes.

Se deberá acoger al marco legal que indica la Ley Orgánica de Protección de Datos Personales (2021) establece “Art. 10 literal J) Los responsables y encargados de tratamiento de los datos personales deberán implementar todas las medidas de seguridad adecuadas y necesarias, entendiéndose por tales las aceptadas por el estado de la técnica, sean estas organizativas, técnicas o de cualquier otra índole, para proteger los datos personales, frente a cualquier riesgo, amenaza, vulnerabilidad, atendiendo a la naturaleza de los datos de carácter personal, al ámbito y el contexto” (p, 18). Desde este punto legal y ético las imágenes tomadas mediante la cámara no se almacenarán o publicará con fines lucrativos o delictivos las imágenes expuestas serán con fines educativos con

esto se garantiza la integridad del usuario las capturas serán en tiempo real para procesar información.

El sistema tendrá un impacto social y tecnológico para poder reducir los incidentes producidos por la somnolencia salvaguardando la vida de los conductores y la comunidad en general. En el ámbito profesional fortalecerá los conocimientos de Inteligencia Artificial, programación y electrónica y así poder tener una mejor cultura sobre el avance de la tecnología.

El presente trabajo fundamenta teóricamente varias disciplinas para dar una solución al problema social Integrando principios lógico-computacionales, matemáticos, normativas entre otros con la finalidad de implementar el sistema en las unidades del transporte pesado de la cooperativa y con una escalabilidad a largo plazo fortaleciendo la seguridad vial del país.

El desarrollo efectivo de este proyecto requiere una comprensión profunda de los componentes y tecnologías involucrados. Esta sección proporciona una base teórica sólida para la implementación del sistema propuesto y tiene como objetivo detectar signos de somnolencia en conductores de vehículos pesados en tiempo real, mediante técnicas de visión artificial y procesamiento de imágenes faciales. Esto permitirá alertar al conductor cuando se detecten indicios de fatiga, contribuyendo a reducir el riesgo de accidentes asociados a la somnolencia.

2.1.2 Hardware del sistema

2.1.2.1 Raspberry Pi 5

Raspberry Pi 5 cuenta con un procesador ARM, sistema operativo Linux, memoria RAM, lo que lo convierte en un hardware de un alto rendimiento bajo costo y accesible, por el cual se utiliza en el ámbito investigativo y educativo en varios proyectos (Raspberry, s.f.).

Figura 2
Placa Raspberry Pi



Nota. Placa donde se almacena la programación.

2.1.2.2 Cámara module v2

La cámara V2 de 8 megapíxeles y bus de CSI (Camera Serial Interface) envía la información de los píxeles al procesador, mejorando la captura y el procesamiento de imágenes, asegurando así una interacción óptima entre la cámara y la Raspberry PI (Raspberry, s.f.).

Figura 3
Placa de cámara oficial Raspberry Pi V2



Nota. Cámara que permitirá captar los movimientos en tiempo real.

La cámara Raspberry Pi detecta la somnolencia al observar y analizar las caras de las personas. Su integración hace posible monitorear la alerta de un conductor y activar alertas oportunas para prevenir riesgos potenciales asociados con la conducción somnolienta.

Tabla 1*Características técnicas cámara v2*

Característica	Cámara module v2
1/4 5M	Resolución
OmniVision OV5647 Color CMOS QXGA (5 megapíxeles)	Tipo de sensor
3.67 x 2.74 mm (formato 1/4")	Tamaño del sensor
2592 x 1944	Cantidad de píxeles
2.0 x 1.33 m a 2 m	Campo de visión
1080p a 30 fps con códec H.264 (AVC)	Video
25 x 24 mm (sin incluir el cable flexible)	Tamaño de la placa

2.1.2.3 Módulo Buzzer:

El módulo buzzer es un componente electrónico cuya función es emitir sonido al recibir voltaje, lo que permite emitir alertas.

Figura 4*Módulo Buzzer*

Nota. El dispositivo emite una señal sonora.

Tabla 2*Características Técnicas Módulo Buzzer*

Característica	Módulo Buzzer
3.5V ~ 5.5V	Voltaje de funcionamiento
30mA / 5VDC	Corriente máxima

2500Hz $\pm$ 300Hz	Frecuencia de resonancia
85Db @ 10cm	Salida de sonido mínima
20 ° C ~ 70 ° C [-4 ° F ~ 158 ° F]	Temperatura de trabajo
-30 ° C ~ 105 ° C [-22 ° F ~ 221 ° F]	Temperatura de almacenamiento
[0.728in x 0.591in]	Dimensiones 18.5mm x 15mm

2.1.3 Software y algoritmos

El sistema aplica las tecnologías de IA relevantes y algoritmos de visión por computadora para el procesamiento y análisis de materiales en video. Las técnicas empleadas son:

2.1.3.1 Configuración Inicial de la Raspberry Pi 5

Descarga e Instalación del Sistema Operativo que se encuentra en la página oficial de Raspberry Pi, seleccione y descargue la imagen del sistema operativo recomendado.

2.1.3.2 Cargar el Sistema Operativo en la Tarjeta MicroSD

El sistema operativo debe ser grabado en una tarjeta microSD utilizando una herramienta como Raspberry Pi Imager desde el ordenador, y luego esta tarjeta microSD se inserta en la Raspberry Pi. Seleccionar en la configuración el dispositivo y comprobar los parámetros de configuración. Estas aplicaciones facilitan la colocación del sistema operativo en la tarjeta microSD.

2.1.3.3 Configuración Inicial

Habilitar SSH:

SSH (Secure Shell) ofrece la opción de acceso remoto hacia la Raspberry Pi. Para habilitar seguir los siguientes pasos:

- Abre el Menú de Inicio y ve a Configuración.
- Selecciona Configuración de Raspberry Pi.
- En la pestaña de Interfaces, encuentra la opción SSH y habilitarla para permitir el acceso remoto.

SSH (Secure Shell) permite acceder remotamente a un Sistema operativo de manera segura mediante contraseña y poder transferir archivos siendo esto una gran ventaja en cuanto a seguridad (Bergsma & Dowling, 2021).

Ejemplo: `ssh usuario@servidor`

2.1.3.4 Acceso remoto y Actualización del Sistema Operativo

Realizado la configuración del protocolo SSH se podrá acceder desde cualquier dispositivo mediante RealVNC, connect raspberrypi o la terminal y realizar las configuraciones sin depender de medios físicos o cables, se debe actualizar el sistema operativo mediante el comando:

```
sudo apt update && sudo apt upgrade -y
sudo reboot
```

Este comando, ejecutado con privilegios de superusuario a través de sudo, actualiza la lista local de paquetes disponibles en función de la información más reciente de los repositorios configurados.

Sudo otorga permiso para ejecutar comandos con los privilegios del superusuario; `apt update` y `apt upgrade` es una utilidad que instala, actualiza, elimina y gestiona paquetes de los repositorios configurados, y `update` solicita actualizar la lista de paquetes y obtener metadatos de los repositorios. (Debian, 2023).

Instalación de librerías:

El siguiente comando tiene el propósito de instalar las diferentes bibliotecas, asegurando su compatibilidad y optimizando la funcionalidad general del sistema.

```
sudo apt install -y build-essential cmake pkg-config
sudo apt install -y libjpeg-dev libtiff5-dev libpng-dev
sudo apt install -y libavcodec-dev libavformat-dev libswscale-
dev
sudo apt install -y libv4l-dev libxvidcore-dev libx264-dev
sudo apt install -y libgtk-3-dev libcanberra-gtk3-dev
sudo apt install -y libatlas-base-dev gfortran
sudo apt install -y python3-dev python3-pip python3-venv
sudo apt install -y libasound2-dev portaudio19-dev
sudo apt install -y git wget unzip
```

2.1.3.5 Configuración de la Cámara

La activación de la interfaz de la cámara en la Raspberry Pi se lleva a cabo, mediante el comando:

```
lsusb
ls /dev/video*
sudo apt install -y fswebcam
fswebcam -d /dev/video0 -r 640x480 test.jpg
```

El uso de sudo añade permisos de superusuario, lo que ayuda a configurar el hardware y el software para el dispositivo.

Crear el entorno e instalación de Librerías Adicionales:

```
mkdir ~/drowsiness_detector
cd ~/drowsiness_detector
python3 -m venv drowsiness_env
source drowsiness_env/bin/activate
pip install --upgrade pip setuptools wheel
```

Verificación del Funcionamiento de la Cámara

Para verificar que la cámara está funcionando correctamente, ejecute:

```
sudo lsof /dev/video0
```

Este comando tomará una foto y la guardará como test.jpg en el directorio actual, revise el archivo para confirmar que la cámara está operativa.

2.1.3.6 Instalación de Python y Librerías Necesarias

Para instalar Python en una Raspberry Pi o en un sistema basado en Debian, se utiliza el siguiente comando.

```
pip install opencv-python==4.8.1.78
pip install dlib
pip install numpy==1.24.3
pip install scipy==1.11.1
pip install pygame==2.5.0
pip install imutils==0.5.4
```

2.1.3.7 Verificación de la Versión de Python

El siguiente comando nos permite verificar que versión de Python se encuentra instalada y poder ejecutar scripts y comandos relacionados .

```
python3 --version
```

2.1.3.8 Instalación y Verificación de OpenCV

El reconocimiento visual por computadora y técnicas de procesamiento de imágenes está dada por una biblioteca de OpenCV “python3-opencv” que puede ser instalada mediante el siguiente comando

```
sudo apt-get install -y python3-opencv
```

Para comprobar la versión y confirmar su instalación se ejecuta un Script, desde la terminal con el siguiente comando:

```
python3
```

La instrucción permite ingresar a Python y para poder verificar la versión ejecute el comando:

```
>>> import cv2
>>> print(cv2.__version__)
```

import cv2: Permite incluir bibliotecas y módulos en el script y carga la biblioteca OpenCV

cv2.__version__: Permite verificar la versión instalada de OpenCV.

2.1.3.9 Código para Detección de Rostros y Ojos LANDMARKS

Con OpenCV, se pueden detectar rostros y ojos en imágenes, y desde luego, resulta necesario realizar cálculos matemáticos para poder procesar las imágenes y detectar las características requeridas, Cv2 (OpenCV) sirve para el procesamiento de imágenes y numpy para realizar cálculos y manipular arrays son las dos bibliotecas que se importan.

```
wget http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2
bunzip2 shape_predictor_68_face_landmarks.dat.bz2
```

Los clasificadores de Haar son modelos utilizados para detectar automáticamente rostros y ojos en fotos y videos a través del aprendizaje automático, utilizando métricas de características y rasgos que son característicos de rostros y ojos. Se realiza una conversión a escala de grises para facilitar el procesamiento de un video o imagen porque los clasificadores de Haar requieren imágenes en escala de grises. Estos, junto con el algoritmo Viola-Jones, que permite la detección y localización en tiempo real de características, procesan imágenes. Este algoritmo realiza cálculos utilizando técnicas de coincidencia de plantillas que comparan grupos de píxeles y su intensidad dentro de regiones rectangulares de la imagen. Una vez que el clasificador ha comparado la diferencia de brillo en las regiones, es capaz de identificar y clasificar las características que se asocian a rostros y ojos (Jones, 2001).

2.1.3.10 Cálculos Matemáticos.

2.1.3.10.1 Transformación de Color a Escala de Grises

Para el tratamiento de imágenes, suele ser más conveniente convertir las imágenes a escala de grises. Esta transformación ayuda de forma significativa a detectar las características en las imágenes, este proceso es un claro ejemplo de reducción de complejidad. La transformación se hace utilizando la fórmula de luminancia la cual se hace con el color RGB (rojo, verde, azul) en color gris en el caso del procesamiento de imágenes. La luminancia de una imagen se describe como la porción de una imagen que es la luz y se la puede calcular como una combinación de promedio ponderado de los valores de los canales de color. La fórmula estándar de luminancia es la siguiente (Chandran, 2020).

Ecuación 1

Fórmula de luminancia

$$Y = 0.299(R) + 0.587(G) + 0.114(B)$$

Donde:

- Y es el valor de luminancia (o el valor de gris resultante).
- R es el valor del canal rojo del píxel.
- G es el valor del canal verde del píxel.
- B es el valor del canal azul del píxel.

2.1.3.10.2 Aplicación de Suavizado (Gaussian Blur)

El suavizado gaussiano se utiliza en procesamiento de imágenes para reducir el ruido y los detalles finos, lo cual facilita la detección de características como bordes y formas en las imágenes. Esto se logra mediante la convolución de la imagen con un núcleo gaussiano, que es una función matemática que aplica un filtro de suavizado.

La función del filtro gaussiano se define matemáticamente como:

Ecuación 2

Fórmula Matemática del Filtro Gaussiano

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Donde:

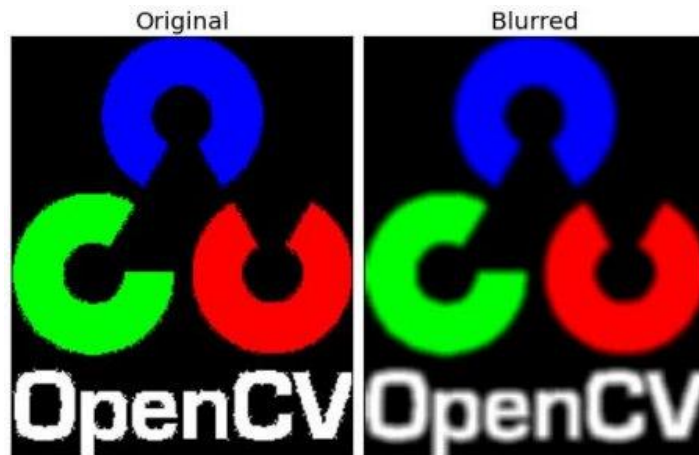
- $G(x, y)$ es el valor del filtro en las coordenadas X y Y .
- σ (*Desviación Estándar "sigma"*) es la desviación estándar del núcleo gaussiano, que controla la cantidad de suavizado y es un parámetro que define la amplitud de la función

gaussiana. Un valor mayor de σ produce un suavizado más fuerte, mientras que un valor menor produce un suavizado más débil. (opencv, 2022)

- X y Y son las coordenadas del píxel dentro del núcleo gaussiano.

Figura 5

Suavizado de imagen



Nota. Adaptado de Visión artificial de código abierto (p. 1), OpenCV, 2022.

2.1.3.10.3 Detección de Rostros y Ojos

Los clasificadores en cascada Haar utilizan un proceso basado en características de Haar y AdaBoost para detectar objetos en imágenes. El algoritmo utiliza un modelo en cascada de clasificadores que trabajan en diferentes etapas. AdaBoost es un algoritmo de aprendizaje automático que combina múltiples clasificadores débiles para formar un clasificador fuerte. En el contexto de los clasificadores en cascada Haar, AdaBoost selecciona las características más significativas y ajusta los pesos de las regiones de la imagen para mejorar la precisión del modelo, los clasificadores en cascada aplican una serie de funciones de clasificación basadas en características específicas (rectángulos) para detectar objetos. Cada clasificador utiliza un umbral para decidir si una región de la imagen contiene el objeto objetivo (opencv, 2022).

2.1.3.10.4 Cálculo del Estado del Ojo

Conversión a Escala de Grises: La imagen del ojo se convierte a escala de grises para simplificar el procesamiento y reducir la complejidad, se aplica un umbral binario a la imagen en escala de grises para segmentar el ojo en áreas blancas (región del ojo) y negras (fondo), la proporción de píxeles blancos se calcula utilizando la fórmula:

Ecuación 3

Proporción de Píxeles Blancos

$$\text{Proporción de Píxeles Blancos} = \frac{\text{Número de Píxeles Blancos}}{\text{Número Total de Píxeles}}$$

Número de Píxeles Blancos: Es la cantidad de píxeles en la imagen que se han clasificado como blancos después de aplicar un umbral binario.

Número Total de Píxeles: Es la cantidad total de píxeles en la imagen del ojo.

Ecuación 4

Proporción de Píxeles Negros

$$\text{Proporción de Píxeles negros} = \frac{\text{Número de Píxeles negros}}{\text{Número Total de Píxeles}}$$

Número de Píxeles Negros: Es la cantidad de píxeles en la imagen que se han clasificado como negros después de aplicar un umbral binario.

Número Total de Píxeles: Es la cantidad total de píxeles en la imagen del ojo.

Umbral de Decisión: Si la proporción de píxeles blancos es mayor que un umbral predefinido (por ejemplo, 0.4), se considera que el ojo está abierto; de lo contrario, se considera cerrado.

2.1.3.10.5 Detección de parpadeos y visión artificial

A través del seguimiento de puntos faciales. La visión por computadora, también conocida como visión artificial, constituye una subdivisión de la inteligencia artificial y abarca técnicas destinadas a capacitar a una computadora para adquirir, procesar, analizar y comprender diversos tipos de información, pero uno de los métodos más comunes es el cálculo del índice de relación de aspecto del ojo (EAR, por sus siglas en inglés). Este método

se basa en el seguimiento de puntos faciales específicos alrededor del ojo y usa geometría para determinar si el ojo está cerrado o abierto.

Ecuación 5

Índice de Relación de Aspecto del Ojo

$$EAR = \frac{d_1 + d_2 + d_3}{2 * d_4}$$

Donde:

- d_1 = es la distancia vertical entre los puntos del párpado superior e inferior en la parte media del ojo.
- d_2 = es la distancia vertical entre los puntos del párpado superior e inferior en la parte lateral del ojo.
- d_3 = es la distancia vertical entre los puntos del párpado superior e inferior en la parte medial del ojo.
- d_4 = es la distancia horizontal entre los puntos del párpado superior (o inferior) en ambos extremos del ojo (el ancho del ojo).

Para obtener las distancias en píxeles entre puntos faciales detectados, se calcula la distancia euclidiana entre ellos. La fórmula matemática para calcular la distancia euclidiana en un plano bidimensional es:

Ecuación 6

Distancia Euclidiana

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Localización de Puntos Faciales: Primero, se detecta puntos específicos alrededor del ojo, para el modelo de 68 puntos faciales, por lo general, los puntos relevantes son:

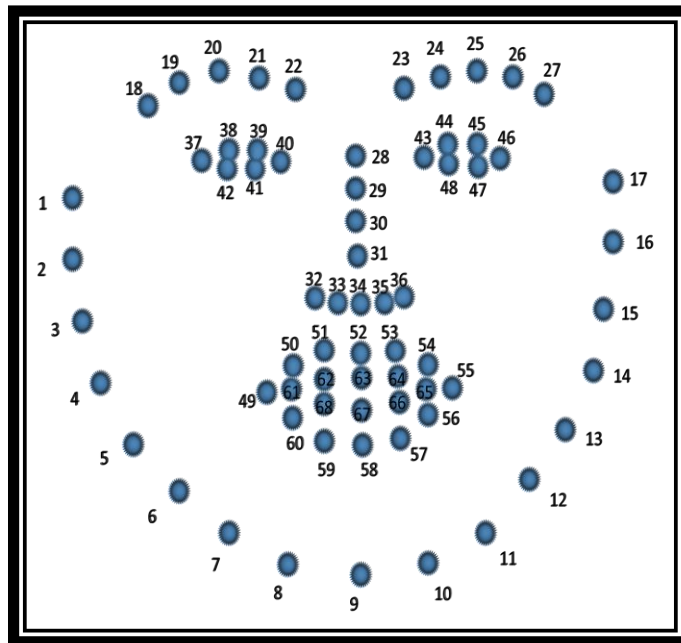
Párpado Superior: Puntos en el borde superior del ojo (por ejemplo, Puntos 37-42).

Párpado Inferior: Puntos en el borde inferior del ojo (por ejemplo, Puntos 43-48).

Extremos del ojo: Puntos en los extremos izquierdo y derecho del ojo (por ejemplo, Puntos 3 y 46).

Figura 6

Puntos faciales



Nota. Adaptado de Sistema avanzado de asistencia al conductor para la detección de distracción y somnolencia utilizando puntos de referencia faciales (p. 93), Luis Sinche, 2023

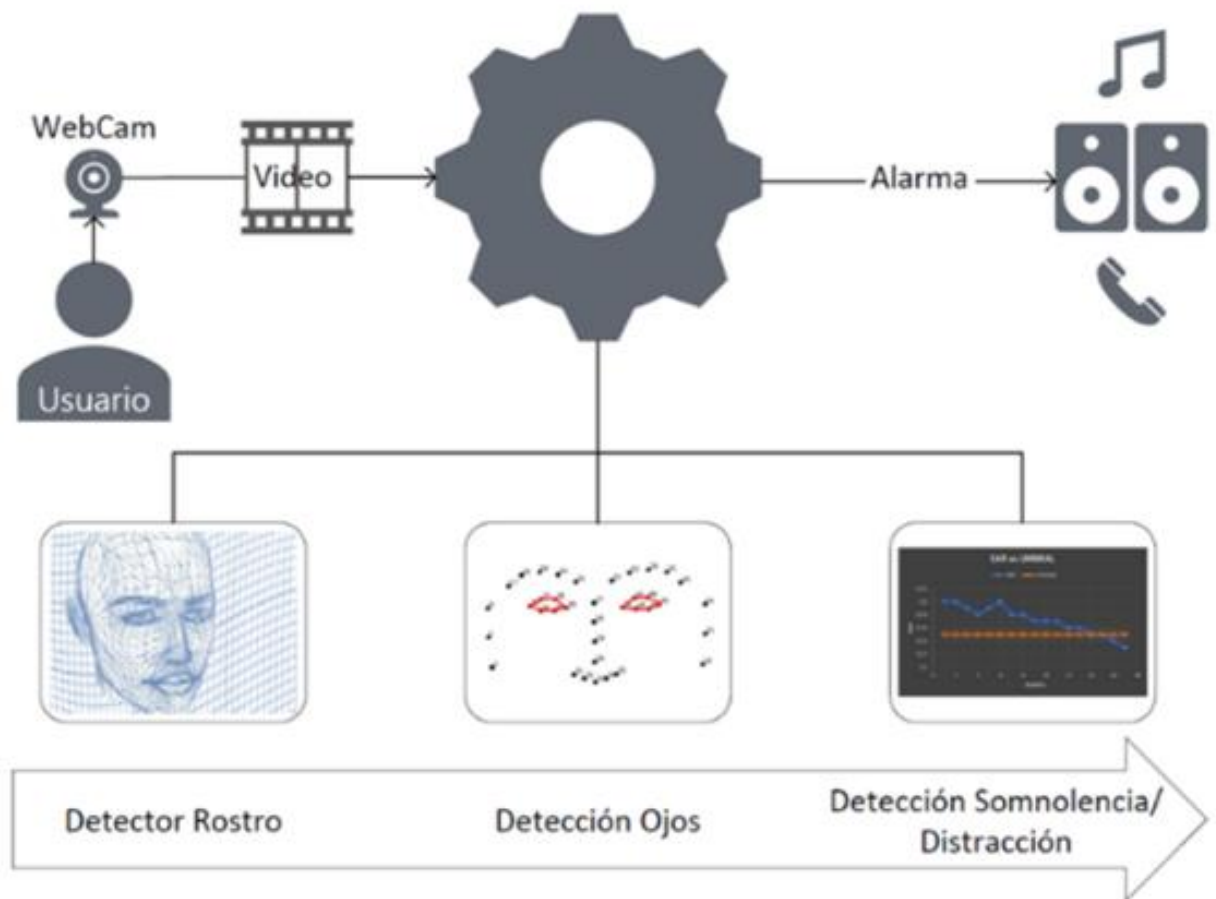
2.2 Descripción de la propuesta

a. Estructura general

En la figura se establece un breve detalle del funcionamiento del sistema que alertará al conductor si presenta signos de somnolencia.

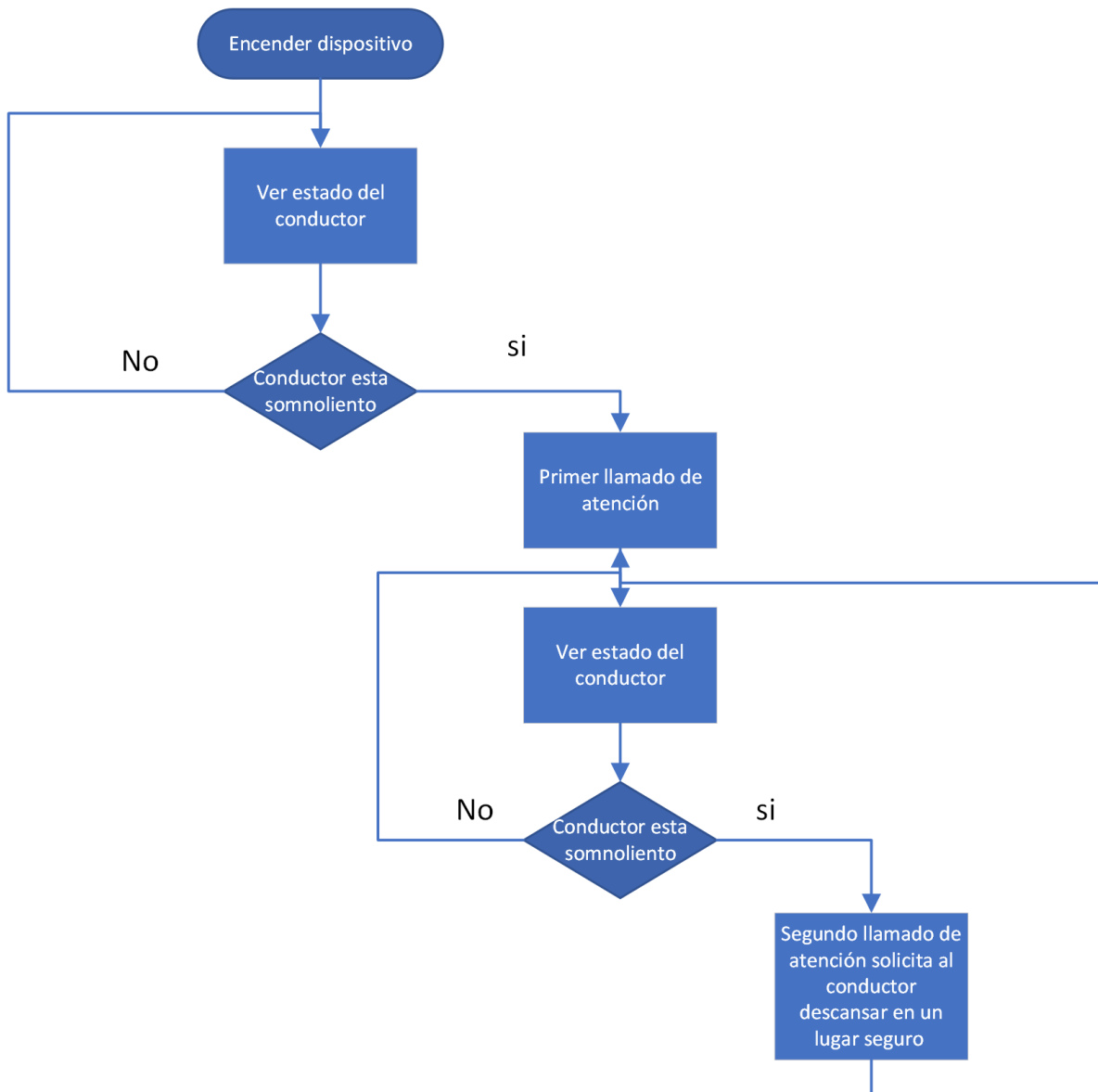
Figura 7

Diagrama del sistema



Nota. Adaptado de Sistema avanzado de asistencia al conductor para la detección de distracción y somnolencia utilizando puntos de referencia faciales (p. 93), Luis Sinche, 2023

Figura 8
Diagrama de flujo del sistema



b. Explicación del aporte

Interactividad del Sistema y Compromiso del Usuario

El diseño del sistema propuesto incorpora una serie de actividades interactivas continuas que obligan al conductor a mantener un nivel de alerta elevado. Dicha respuesta es clave para minimizar el riesgo de accidentes por somnolencia. La interactividad se consigue a través de un sistema de supervisión visual que, de manera ininterrumpida, examina rasgos faciales, tasas de parpadeo y maniobras asociadas a la fatiga, a la vez que emite avisos acústicos en el momento en que cualquiera de esos parámetros se desvía hacia la somnolencia. Esta estrategia de realización

rápida es crítica, dado que la somnolencia disminuye el tiempo de reacción y solidez en la toma de decisiones. La perpetuación del diálogo entre el sistema y el conductor establece una defensa proactiva ante el riesgo. Simultáneamente, la interactividad no se limita a la detección; proporciona también al conductor una visualización de sus propios patrones de comportamiento, convirtiendo la alerta en un acto de educación y, en consecuencia, potenciando un progreso subjetivo hacia un mayor nivel de seguridad en la conducción.

Recursos Tecnológicos de Soporte al Aprendizaje

La infraestructura que soporta tanto la operatividad del sistema como el ciclo del aprendizaje automático se ancla en varios dispositivos distintivos. Primeramente, se encuentra la cámara que actúa como el sensor primario, registrando en tiempo real el rostro del conductor. A través de técnicas avanzadas de visión por computadora, el sensor descompone micro expresiones, seguimiento ocular, orientación de la cabeza y tasa de parpadeo, datos que se cruzan para delatar grados graduales de fatiga. En segundo plano opera el altavoz que integra el tablero, el cual emite señales acústicas que responden al umbral de somnolencia calibrado: melodías discretas en el rango de leve fatiga, trenes de tonos más agresivos ante estados más severos. Complementariamente, el sistema moviliza sensores de movimiento del vehículo que, al cotejar con la captura facial, revelan desvíos en el estilo de conducción. La interoperabilidad de estos dispositivos se articula mediante una unidad central de procesamiento, que ejecuta modelos de aprendizaje automático en tiempo real y que, alimentándose del historial de interacción, refina su sensibilidad a las particularidades de cada conductor.

Metodología de Evaluación del Sistema

Las actividades de evaluación del sistema adoptan un esquema metodológico de doble componente, que proporciona corroboración en entornos controlados y en condiciones reales de uso. La primera etapa consiste en ensayos en simuladores de conducción, en los que se configuran situaciones específicas manipulando variables como iluminación (diurna, nocturna, crepuscular), condiciones meteorológicas artificiales y grados de fatiga inducida, esta última manejada de modo seguro y ético con voluntarios. La segunda etapa desplaza la validación a un vehículo, monitoreados

por personal acreditado, y consiste en recorridos de diversas duraciones y clasificaciones de vía (urbanas, rurales y autopistas). Durante estas campañas se cuantifican parámetros fundamentales como la precisión en la detección (porcentajes de verdaderos positivos y falsos positivos), el intervalo de respuesta del sistema, la eficacia en la generación de alertas y la capacidad de ajuste según la variabilidad entre conductores. Completa el protocolo un dialecto estadístico que examina los datos recogidos, a fin de corroborar la confiabilidad y la coherencia del sistema a través de un amplio espectro de situaciones operativas.

Actividades para la Construcción del Conocimiento Técnico

El desarrollo del sistema se articula en cinco actividades centrales que, de modo secuencial, fomentan la sedimentación del saber técnico. En primer lugar, la programación de algoritmos de visión por computadora supone la cimentación, ya que crea código en OpenCV para el análisis de imágenes de rostros, la detección de puntos de referencia anatómicos y el procesamiento de patrones de parpadeo a través del análisis de Sholl. En segundo lugar, el entrenamiento de modelos de aprendizaje automático ocupa la esfera de la inteligencia artificial y se fundamenta en la creación y en el adiestramiento de redes neuronales sobre grandes conjuntos de datos que recogen el comportamiento de los conductores, lo que permite al identificar patrones complejos vinculados a variaciones en el grado de fatiga. En tercer lugar, la integración de componentes hardware-software asegura la armonización de todos los módulos, logrando una comunicación eficaz entre sensores, unidades de procesamiento y actuadores, al tiempo que se perfecciona la operativa en tiempo real. La prueba iterativa y de optimización incluye ciclos de refinamiento continuo basados en datos de prueba que involucran ajustes de parámetros algorítmicos y mejoras de precisión en el proceso de detección. Finalmente, el análisis de datos y el proceso de validación incluyen una evaluación estadística integral, identificando las áreas de mejora y, además, la sistematización de los resultados, de igual manera contribuye al conocimiento existente en el ámbito de la seguridad vehicular y los sistemas de transporte inteligente.

c. Estrategias y/o técnicas

El desarrollo del sistema adoptó una metodología de aprendizaje basado en proyectos combinada con aprendizaje experimental, donde la construcción del conocimiento se fundamentó en la resolución de un problema real del sector del transporte pesado. Se implementó un enfoque iterativo que integró cinco actividades centrales de construcción del conocimiento técnico: programación de algoritmos de visión por computadora usando OpenCV para análisis facial, entrenamiento de modelos de aprendizaje automático para identificar patrones de somnolencia, integración hardware-software para armonizar sensores, pruebas iterativas con ciclos de refinamiento continuo, y análisis de documentos para validación de resultados. Esta estrategia promovió el pensamiento crítico y la aplicación práctica de conceptos teóricos, permitiendo desarrollar competencias en inteligencia artificial, procesamiento de imágenes y sistemas embebidos mientras resolvían una problemática social tangible de seguridad vial.

Las herramientas tecnológicas fueron seleccionadas por su accesibilidad, robustez y relevancia. Se optó por la Raspberry Pi 5 como plataforma principal debido a su bajo consumo energético de arquitectura ARM, potencia suficiente para procesamiento de imágenes en tiempo real y amplia conectividad, facilitando la integración de módulos adicionales como unidades de alerta. Python fue elegido como lenguaje de programación por su sintaxis clara, abundante documentación y predominio en el ámbito de IA y machine learning, permitiendo desarrollo ágil y corrección eficiente de algoritmos. OpenCV se implementó como librería base para tratamiento de imágenes por estar optimizada para tiempo real, incluir implementaciones de detección facial (Haar Cascades, DNN) e integrarse nativamente con Python. Se complementó con NumPy para cálculos matriciales eficientes y dlib para detección precisa de landmarks faciales. Esta combinación tecnológica proporcionó un entorno de aprendizaje alineado con estándares industriales vigentes, brindando experiencia práctica con herramientas de referencia en visión artificial y sistemas embebidos.

2.3 Validación de la propuesta

Descripción de perfil de validadores

Nombres y Apellidos	Años de experiencia	Titulación Académica	Cargo
Margarita Gioconda Arroyo Paredes	2	Ingeniera en Electrónica e Instrumentación Magister en Electrónica y Automatización	Docente del departamento de ciencias exactas de la Universidad de las Fuerzas Armadas ESPE
Melany Jazmin Yarad Jacome	7	Magister en gerencia de seguridad y riesgos Mg en electrónica y automatización	Coordinadora de la Maestría en Defensa y Seguridad mencion Conducción Militar y Gestión Logística ESPE.
Wilmer Javier Labre Tarco	8	Ing. Automotriz Mg. Matemática aplicada	Docente de Física y Matemática B.I.

Escala de evaluación. Elaborada por: Ing. Margarita Gioconda Arroyo Paredes Mg.

CRITERIOS	EVALUACIÓN SEGUN IMPORTANCIA Y REPRESENTATIVIDAD				
	En Total Desacuerdo	En Desacuerdo	Ni de Acuerdo Ni en Desacuerdo	De Acuerdo	Totalmente Acuerdo
Impacto				X	
Aplicabilidad				X	
Conceptualización				X	
Actualidad				X	
Calidad Técnica				X	
Factibilidad			X		
Pertinencia			X		

Escala de evaluación. Elaborada por: Ing. Melany Jazmin Yarad Jacome Mg.

CRITERIOS	EVALUACIÓN SEGUN IMPORTANCIA Y REPRESENTATIVIDAD				
	En Total Desacuerdo	En Desacuerdo	Ni de Acuerdo Ni en Desacuerdo	De Acuerdo	Totalmente Acuerdo
Impacto				X	
Aplicabilidad			X		
Conceptualización			X		
Actualidad				X	
Calidad Técnica			X		
Factibilidad				X	
Pertinencia			X		

Escala de evaluación. Elaborada por: Ing. Wilmer Javier Labre Tarco Mg.

CRITERIOS	EVALUACIÓN SEGUN IMPORTANCIA Y REPRESENTATIVIDAD				
	En Total Desacuerdo	En Desacuerdo	Ni de Acuerdo Ni en Desacuerdo	De Acuerdo	Totalmente Acuerdo
Impacto				X	
Aplicabilidad			X		
Conceptualización			X		
Actualidad				X	
Calidad Técnica				X	
Factibilidad			X		
Pertinencia			X		

2.4 Matriz de articulación de la propuesta

Tabla 1.

Matriz de articulación

Ejes o partes principales del proyecto		Breve descripción de los resultados de cada parte	Sustento teórico que se aplicó en la construcción del proyecto	Metodologías, herramientas técnicas y tecnológicas que se emplearon
1	Elementos electrónicos como Raspberry Pi, cámara compatible, especificaciones y Módulos de alerta sonora.	1.1. Comparativa entre varios sistemas teniendo una factibilidad mayor al 90% 1.2. El proyecto es factible con un presupuesto de \$180-220 USD 1.3. Raspberry Pi 5 como la mejor opción por su balance precio-rendimiento para visión artificial.	Inteligencia Artificial Matemáticas Aplicadas Visión por Computador Machine Learning	En el proceso de investigación bibliográfica, se estudió la visión por computadora aplicada al procesamiento de imágenes y a la implementación de algoritmos de detección de rostros. Se emplearon modelos pre entrenados y algoritmos avanzados, como redes neuronales profundas como clasificadores en cascada de Haar, para la

				identificación precisa de rostros en las imágenes capturadas por la cámara.
2	Diseño de circuitos electrónicos, código de programación en tarjeta Raspberry pi, Aplicación en OpenCV mediante algoritmos de detección facial	2.1 Aplicación OpenCV 2.2 Algoritmos optimizados	Programación de la tarjeta Raspberry pi e instalación de librerías con Open CV y Python.	El diseño está basado en varios criterios y argumentos bibliográficos.
3	La implementación del sistema está colocada en la cabina del camión con una cámara que monitorea en tiempo real el estado del conductor	3.1. Instalación de los componentes electrónicos 3.2. programación. 3.3. Aplicación del sistema	Programación basada en Python y algoritmos de visión artificial.	Se aplican los diferentes algoritmos para tener mejores resultados.

2.5 Análisis de resultados. Presentación y discusión.

Durante la fase experimental del prototipo, se evaluaron múltiples parámetros biométricos mediante la implementación de algoritmos de visión artificial en tiempo real. El sistema desarrollado demostró una precisión global del 92%, evaluada sobre un conjunto de datos de 200 secuencias de fotos de cada una, capturadas bajo condiciones controladas.

Figura 9

Bostezos capturados

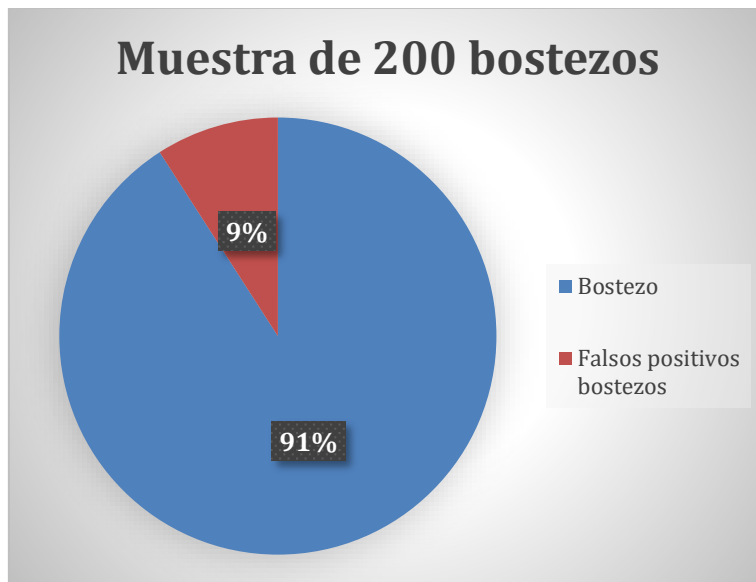


Figura 10

Estado de los ojos



Los parámetros monitoreados incluyeron: frecuencia de parpadeo (normal: 15-20 parpadeos/min vs. somnoliento: <10 parpadeos/min), duración de cierre ocular (normal: <500ms vs. somnoliento: >800ms), ángulo de inclinación cefálica (alerta: $\pm 15^\circ$ vs. somnoliento: $>30^\circ$), y desviación del eje visual. Como se visualiza en la Figura 11, el sistema logró detectar patrones característicos de somnolencia con una latencia promedio de 2.3 segundos desde la manifestación inicial de síntomas. El análisis de variables ambientales reveló que el rendimiento se mantuvo estable en condiciones de iluminación.

Figura 11
Bostezo frecuente



Los resultados obtenidos confirman la viabilidad técnica del modelo teórico propuesto, basado en la teoría de detección de patrones faciales para identificación de estados de fatiga cognitiva. La precisión alcanzada del 92% se alinea consistentemente con el marco teórico de (Delgado & Yandún, 2022), quienes establecieron que los sistemas de visión artificial pueden detectar eficazmente micro-expresiones faciales asociadas con somnolencia mediante análisis de landmarks faciales y seguimiento ocular. Esta convergencia valida la hipótesis inicial de que

la integración de múltiples indicadores biométricos (PERCLOS, frecuencia de parpadeo, y posicionamiento cefálico) proporciona mayor robustez diagnóstica que métodos univariados.

Los resultados superan ligeramente los reportados por (Sinche Cueva, 2023) quienes alcanzaron 89.2% de precisión, y se equiparan con los hallazgos de (Shuyan & Gangtie, 2020) que obtuvieron 91.8% usando redes neuronales convolucionales. Limitaciones identificadas: El principal desafío técnico radica en la sensibilidad a variaciones lumínicas extremas, siendo una problemática, sugiriendo una limitación intrínseca de los algoritmos de procesamiento de imágenes basados en características visuales. Propuestas futuras: Se pide la implementación de iluminación infrarroja activa para operación nocturna, y la incorporación de sensores complementarios. Adicionalmente, se propone validar el sistema mediante pruebas en condiciones reales de conducción para evaluar su efectividad en entornos no controlados.

CONCLUSIONES

Se concluye que la revisión bibliográfica permitió establecer un marco teórico sólido que fundamenta la aplicación de algoritmos de visión artificial como Haar Cascade y Eye Aspect Ratio (EAR) para la detección de patrones de somnolencia. Los estudios analizados demuestran que la integración de múltiples indicadores biométricos (frecuencia de parpadeo, duración de cierre ocular), validando el enfoque teórico adoptado en el proyecto para el sector del transporte pesado.

La Cooperativa de Transporte Pesado Gran Colombiana carece de tecnologías preventivas para detectar somnolencia en conductores, quienes laboran jornadas extensas de hasta 24 horas con apenas 3 horas de descanso. El diagnóstico reveló que al menos un conductor experimenta accidentes anuales relacionados con somnolencia, representando costos económicos significativos y riesgos para la seguridad vial, evidenciando la necesidad urgente de implementar sistemas tecnológicos preventivos.

La integración de Raspberry Pi 5, cámara USB y algoritmos de OpenCV permite desarrollar efectivamente un prototipo funcional de bajo costo. El sistema implementado logra procesar imágenes en tiempo real, detectar rostros mediante clasificadores Haar Cascade, calcular el índice EAR para análisis de parpadeos, y activar alertas sonoras cuando se identifican patrones característicos de somnolencia, demostrando la viabilidad técnica de la solución propuesta.

El prototipo desarrollado alcanzó una precisión del 92% en la detección de estados de somnolencia. Las pruebas confirmaron el funcionamiento estable en condiciones de iluminación estable, validando la efectividad del sistema para aplicación en condiciones reales de transporte pesado.

RECOMENDACIONES

Se recomienda profundizar en la investigación de técnicas avanzadas de inteligencia artificial como redes neuronales convolucionales (CNN) y algoritmos de deep learning para mejorar la precisión de detección y reducir falsos positivos. Asimismo, se sugiere explorar la integración de sensores complementarios para desarrollar sistemas multimodales más robustos que combinen análisis visual con datos fisiológicos y de comportamiento de conducción.

Se recomienda realizar estudios longitudinales que documenten patrones de fatiga en conductores de transporte pesado bajo diferentes condiciones operativas, rutas y horarios. Se sugiere implementar programas piloto escalonados que permitan evaluar la aceptación tecnológica por parte de los conductores y desarrollar protocolos de capacitación específicos para el uso efectivo del sistema, considerando las particularidades culturales y operativas del sector.

Investigar alternativas de hardware de mayor rendimiento para procesamiento en tiempo real y explorar la implementación de iluminación infrarroja activa para operación nocturna. Se sugiere desarrollar algoritmos adaptativos de normalización lumínica y estudiar la viabilidad de integración con sistemas telemáticos vehiculares existentes para crear soluciones más comprehensivas de monitoreo de flotas.

Ampliar las evaluaciones con muestras más representativas que incluyan mayor diversidad de conductores, condiciones ambientales y tipos de vehículos para garantizar la generalización del sistema. Se sugiere realizar estudios de campo en condiciones reales de operación durante períodos extendidos, implementar métricas de impacto en reducción de accidentes.

BIBLIOGRAFÍA

Sánchez, A. (2022). *Inteligencia artificial y visión por computadora aplicada a la educación*.

Revista ODIGOS UISRAEL. [https://doi.org/ https://doi.org/10.35290/ro.v3n2.2022.587](https://doi.org/https://doi.org/10.35290/ro.v3n2.2022.587)

Soukupova, T., & Cech, J. (2021). *Real-Time Eye Blink Detection using Facial Landmarks*. Center for Machine Perception, Department of Cybernetics.

<https://doi.org/https://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf>

Alemán, S., & Pujota, C. (2024). *Análisis de algoritmos de aprendizaje para detectar distracciones y somnolencia en*. Universidad Central del Ecuador.

<https://doi.org/https://www.dspace.uce.edu.ec/handle/25000/34006>

Asamblea Nacional del Ecuador. (4 de 4 de 2018). *Ley Orgánica de transporte Terrestre Tránsito y Seguridad Vial*. Registro Oficial Suplemento 398.

https://doi.org/https://www.obraspublicas.gob.ec/wp-content/uploads/downloads/2021/08/LOTAIP_6_Ley-Organica-de-Transporte-Terrestre-Transito-y-Seguridad-Vial-2021.pdf

Asamblea Nacional del Ecuador. (2021). *Ley Orgánica de Protección de Datos Personales*. Quinto Suplemento del Registro Oficial 459.

<https://doi.org/https://www.asambleanacional.gob.ec/sites/default/files/private/asambleanacional/filesasambleanacionalname>

Azli Abdullah, M., Mohamad Hanafiah, M., Yogarayan, S., Abdul Razak, S., Azman, A., & Shohel Sayeed, M. (2023). *Driver fatigue detection using Raspberry-Pi*. Indonesian Journal of Electrical Engineering and Computer Science (IJECS).

<https://doi.org/http://doi.org/10.11591/ijeecs.v32.i2.pp1142-1149>

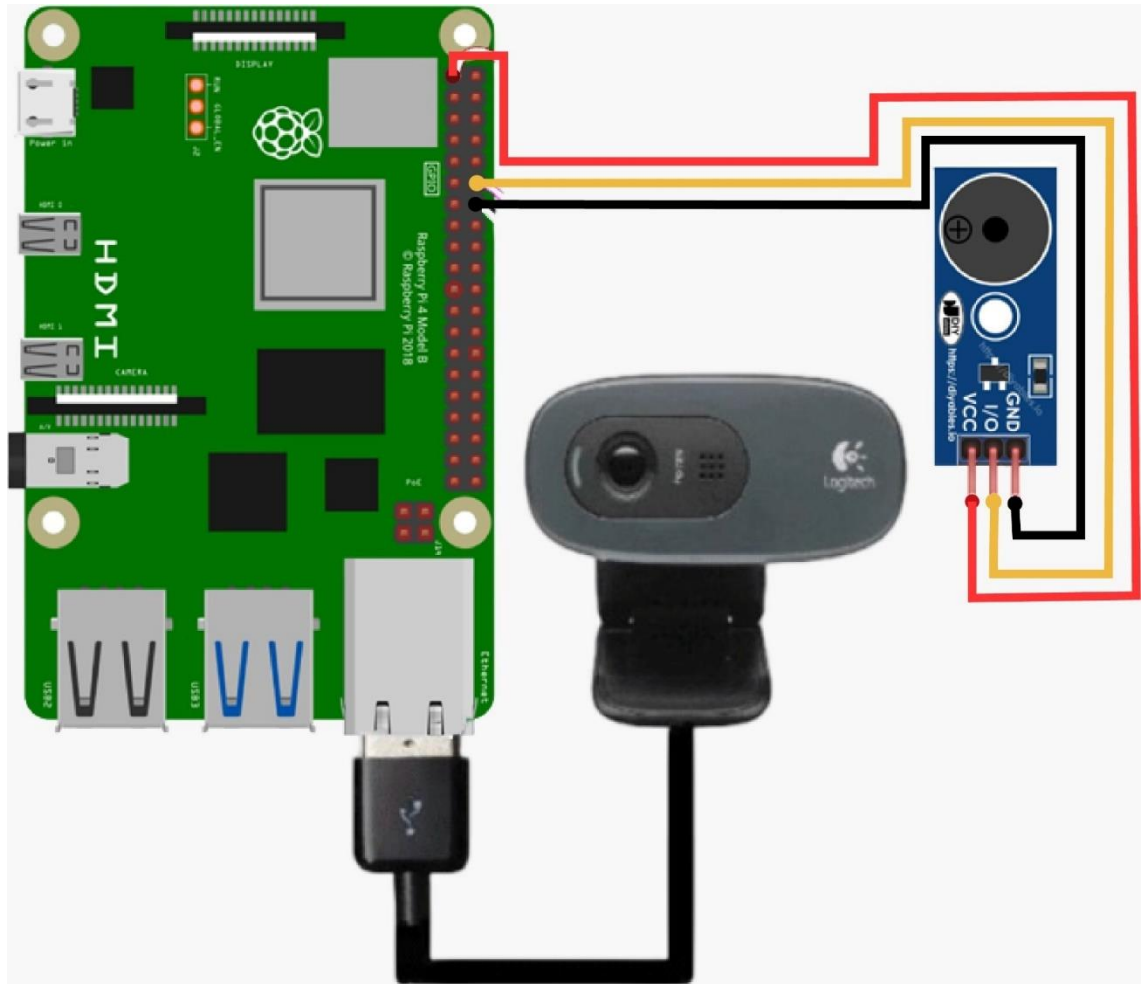
- Barahona, G. O. (2022). *Diseño e implementación de un sistema de visión artificial para el control de calidad en etiquetado de productos comerciales*. Escuela Superior Politécnica De Chimborazo, Riobamba, Chimborazo, Ecuador.
<http://dspace.esPOCH.edu.ec/bitstream/123456789/16147/1/15T00802.pdf>
- Chandran, S. (2020). <https://www.researchgate.net/>. https://www.researchgate.net/publication/224130500_Color_Image_to_Grayscale_Image_Conversion
- Delgado , K., & Yandún, M. (2022). *Sistema de detección de somnolencia para conductores de taxis en la ciudad de Tulcán*. Horizontes De Enfermería.
<https://doi.org/https://doi.org/10.32645/13906984.1171>
- Bahit, M., Putri Utami, N., Kartika Candra, H., & Supi, Y. (2023). *Validation of the Haar Cascade Classification Method in Face Detection*. JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING. <https://doi.org/10.31289/jite.v7i1.10040>
<https://doi.org/https://doi.org/10.32645/13906984.1171>
- Dewi, C., Ching, R., Wei Chang, C., Hung Wu, S., Jiang, X., & Yu, H. (2022). *Eye Aspect Ratio for Real-Time Drowsiness Detection to Improve Driver Safety*. Electronics.
<https://doi.org/https://doi.org/10.3390/electronics11193183>
- Flores , J., Nakano, M., Escamillaq, E., & Pérez, H. (2023). *Detección de somnolencia y distracción en conductores y su implementación en dispositivos móviles*.
<https://doi.org/http://dx.doi.org/10.4067/S0718-07642023000400001>
- Gómez Montoya, J., & Cuartas Arias, M. (2020). *Accidentalidad vial: efectos de la calidad del sueño en el funcionamiento ejecutivo de conductores de transporte público urbano*. Rev. Investig. Innov. Cienc. Salud. <https://doi.org/https://doi.org/10.46634/riics.43>
- International Business Machines. (27 de 7 de 2021). IBM. <https://www.ibm.com/mx-es/think/topics/computer-vision>

- Medina, G., Escobar, K., & Arias, C. (2021). *Factores de riesgo y su relación con la fatiga en conductores de una cooperativa de transporte interprovincial del Ecuador*. Revista San Gregorio. <https://doi.org/https://doi.org/10.36097/rsan.v1i46.1477>
- Oliveira, F., Donadon, T., & Pereira, A. (2021). *Aplicación de inteligencia artificial para monitorear el uso de mascarillas de protección*. revistacientificaesmic. <https://doi.org/https://doi.org/10.21830/19006586.725>
- Jones, P. V. (2001). *Rapid object detection using a boosted cascade of simple features*. E.E.U.U.: IEEE.
- opencv. (2022). <https://docs.opencv.org/>. [https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html](https://docs.opencv.org/https://docs.opencv.org/https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html)
- Raspberry, P. (s.f.). *¿Qué es Raspberry Pi?* . RaspberryPi.cl: <https://raspberrypi.cl/que-es-raspberry/>
- Salh, S., Harbi , J., & Abbas, T. (2022). *Face Detection Using Haar Cascade Algorithm*. IEEE/CVF Winter Conference on Applications of Computer Vision Workshops. <https://doi.org/10.1109/CSCTIT56299.2022.10145680>
- Shuyan, H., & Gangtie, Z. (2020). *Driver drowsiness detection with eyelid related parameters by Support Vector Machine*. <https://doi.org/https://doi.org/10.1016/j.eswa.2008.09.030>
- Siddiqui, H., Saleem, A., Brown, R., Bademci, B., & Lee, E. (2021). *Non-Invasive Driver Drowsiness Detection System*. <https://doi.org/https://doi.org/10.3390/s21144833>
- Veronica, L. (15 de 3 de 2023). Historia de la Cooperativa de Transporte Pesado Gran Colombiana. (L. Mauricio, Entrevistador)

ANEXOS

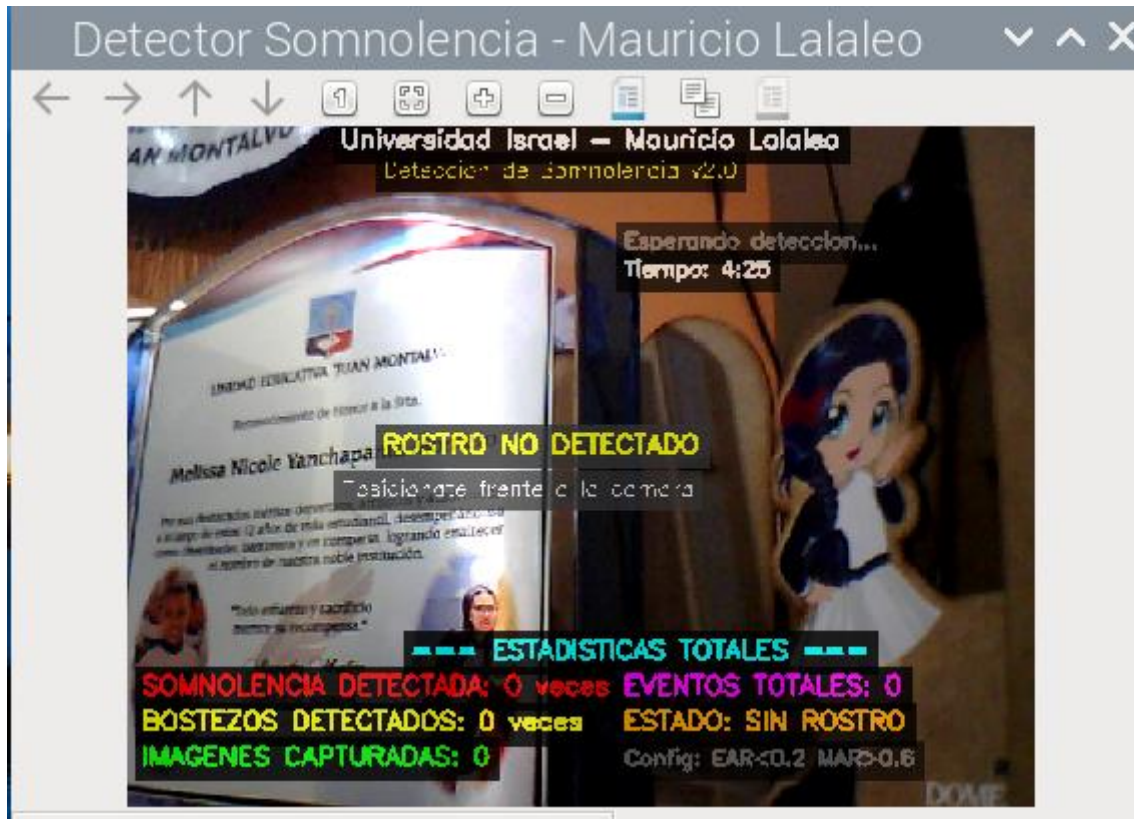
ANEXO 1

DIAGRAMA ELECTRÓNICO DE LA TARJETA DE CONTROL



ANEXO 2

Imagen del sistema sin individuo.



ANEXO 3

Unidad donde se instaló el sistema



Datos generales

1. Edad: _____ Sexo: Masculino (___) Femenino()
2. Años de experiencia como conductor profesional:
 - o () Menos de 5 años
 - o () 5 a 10 años
 - o () Más de 10 años

Bloque 1: Hábitos de conducción

4. ¿Cuántas horas conduce al día en promedio?
 - o () Menos de 8
 - o () 8–20
 - o () Más de 20
5. ¿Con qué frecuencia realiza pausas o descansos durante la jornada de conducción?
 - o () Cada 1–2 horas
 - o () Cada 3–4 horas
 - o () Solo cuando siento cansancio
 - o () No realizo pausas

Bloque 2: Somnolencia y fatiga

6. ¿Con qué frecuencia siente somnolencia mientras conduce?
 - o () Nunca
 - o () Rara vez
 - o () A veces
 - o () Frecuentemente
-

- ☐ Siempre
- 7. Cuando experimenta somnolencia al conducir, ¿qué medidas toma con mayor frecuencia?
 - ☐ Ninguna
 - ☐ Detenerme a descansar
 - ☐ Beber café/energizantes
 - ☐ Abrir la ventana o encender música
 - ☐ Otros: _____
- 8. ¿Ha tenido incidentes o accidentes de tránsito relacionados con la somnolencia?
 - ☐ Sí
 - ☐ No
- 9. ¿Cuántos accidentes de tránsito ha tenido en el último año?
 - ☐ Ninguno
 - ☐ 1
 - ☐ 2-3
 - ☐ Más de 3

Bloque 3: Opinión sobre sistemas de detección de somnolencia

- 10. ¿Considera necesario implementar un sistema de detección de somnolencia en los vehículos de transporte?
 - ☐ Sí
 - ☐ No
 - 11. ¿Qué tan útil cree que sería contar con un sistema que emita alertas cuando detecte somnolencia?
 - ☐ Nada útil
 - ☐ Medianamente útil
 - ☐ Muy útil
 - ☐ Totalmente útil
-

Programación en Python

```
#!/usr/bin/env python3
"""
Sistema de Detección de Somnolencia - Versión con Captura Automática
Autor: Uisrael Mauricio Lalaleo
"""

import cv2
import dlib
import numpy as np
from scipy.spatial import distance as dist
import time
import datetime
import os

class SimpleDrowsinessDetector:
    def __init__(self):
        # Configuración básica - MÁS SENSIBLE
        self.EYE_AR_THRESH = 0.28 # Aumentado para ser más sensible
        self.MOUTH_AR_THRESH = 0.5 # Reducido para detectar bostezos más fácil
        self.EYE_AR_CONSEC_FRAMES = 8 # Reducido de 20 a 8 para respuesta más rápida
        self.MOUTH_AR_CONSEC_FRAMES = 5 # Reducido de 15 a 5 para bostezos más rápidos

        # Contadores
        self.eye_counter = 0
        self.mouth_counter = 0
        self.drowsiness_count = 0
        self.yawn_count = 0
        self.total_detections = 0 # Contador total de detecciones

        # Variables para controlar eventos únicos
        self.drowsiness_detected = False
        self.yawn_detected = False

        # Control de tiempo
        self.start_time = time.time()
```

```

# Variables para captura automática
self.images_saved = 0
self.last_capture_time = 0
self.capture_cooldown = 3 # Segundos entre capturas para evitar spam

# Crear carpeta para imágenes
self.images_folder = f"capturas_somnolencia_{datetime.datetime.now().strftime('%Y%m%d_%H%M%S')}"
os.makedirs(self.images_folder, exist_ok=True)

# Archivo de datos
self.data_file = f"datos_dia_{datetime.datetime.now().strftime('%Y%m%d_%H%M%S')}.txt"

# Variables para mostrar datos sin rostro
self.face_detected = False
self.no_face_counter = 0
self.current_ear = 0.0
self.current_mar = 0.0

# Inicializar detectores
print("Iniciando detectores...")
self.detector = dlib.get_frontal_face_detector()
self.predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

# Índices de landmarks faciales
self.LEFT_EYE = list(range(36, 42))
self.RIGHT_EYE = list(range(42, 48))
self.MOUTH = list(range(48, 68))

print("■ Detectores inicializados correctamente")
print(f"□ Carpeta de imágenes: {self.images_folder}")

def calculate_eye_aspect_ratio(self, eye_landmarks):
    """Calcular Eye Aspect Ratio (EAR)"""
    try:
        A = dist.euclidean(eye_landmarks[1], eye_landmarks[5])
        B = dist.euclidean(eye_landmarks[2], eye_landmarks[4])
        C = dist.euclidean(eye_landmarks[0], eye_landmarks[3])
        if C == 0:
            return 0.0
        ear = (A + B) / (2.0 * C)
        return ear
    except:
        return 0.0

def calculate_mouth_aspect_ratio(self, mouth_landmarks):
    """Calcular Mouth Aspect Ratio (MAR)"""
    try:
        A = dist.euclidean(mouth_landmarks[2], mouth_landmarks[10])
        B = dist.euclidean(mouth_landmarks[4], mouth_landmarks[8])
        C = dist.euclidean(mouth_landmarks[0], mouth_landmarks[6])
        if C == 0:
            return 0.0
        mar = (A + B) / (2.0 * C)
        return mar
    except:
        return 0.0

def capture_image(self, frame, event_type):
    """Capturar imagen automáticamente cuando se detecta somnolencia"""
    current_time = time.time()

    # Verificar cooldown para evitar capturas excesivas
    if current_time - self.last_capture_time >= self.capture_cooldown:
        try:
            timestamp = datetime.datetime.now().strftime('%Y%m%d_%H%M%S_%f')[:-3]
            filename = f"{self.images_folder}/somnolencia_{timestamp}_{event_type}.jpg"

```

```

        # Agregar información sobre la detección en la imagen
        frame_copy = frame.copy()
        cv2.rectangle(frame_copy, (0, 400), (640, 480), (0, 0, 0), -1)
        cv2.putText(frame_copy, f"EVENTO: {event_type.upper()}", (10, 420),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
        cv2.putText(frame_copy, f"Tiempo: {timestamp}", (10, 445),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)
        cv2.putText(frame_copy, f"EAR: {self.current_ear:.3f} | MAR: {self.current_mar:.3f}", (10, 465),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)

        # Guardar imagen
        cv2.imwrite(filename, frame_copy)
        self.images_saved += 1
        self.last_capture_time = current_time

        print(f"📷 Imagen capturada: {filename}")

    except Exception as e:
        print(f"Error capturando imagen: {e}")

def log_event(self, event_type):
    """Registrar evento simple"""
    try:
        timestamp = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
        with open(self.data_file, 'a') as f:
            f.write(f"{timestamp} - {event_type} - EAR: {self.current_ear:.3f} - MAR: {self.current_mar:.3f}\n")
    except:
        pass

def draw_no_face_interface(self, frame):
    """Dibujar interfaz cuando no se detecta rostro"""
    # Mensaje principal (más sutil)
    overlay = frame.copy()
    cv2.rectangle(overlay, (150, 200), (490, 260), (50, 50, 50), -1)
    cv2.addWeighted(overlay, 0.5, frame, 0.5, 0, frame)

    cv2.putText(frame, "Rostro no detectado", (190, 225),
                cv2.FONT_HERSHEY_SIMPLEX, 0.6, (150, 200, 200), 1)
    cv2.putText(frame, "Posicionate frente a la camara", (170, 245),
                cv2.FONT_HERSHEY_SIMPLEX, 0.4, (180, 180, 180), 1)

    # Mostrar todos los datos aunque no haya rostro
    self.draw_statistics(frame, show_current_values=False)

def draw_statistics(self, frame, show_current_values=True):
    """Dibujar estadísticas en pantalla"""
    # TÍTULO PERSONALIZADO EN LA PARTE SUPERIOR
    cv2.rectangle(frame, (0, 0), (640, 40), (0, 0, 0), -1)
    cv2.putText(frame, "Universidad Israel - Mauricio Lalaleo", (150, 15),
                cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 255), 2)
    cv2.putText(frame, f"Deteccion de Somnolencia v2.0", (180, 35),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 255), 1)

    # Panel de información principal (lado derecho)
    info_x = 350
    cv2.rectangle(frame, (info_x-10, 45), (640, 220), (0, 0, 0), -1)

    if show_current_values:
        cv2.putText(frame, f"EAR: {self.current_ear:.3f} / {self.EYE_AR_THRESH}", (info_x, 65),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)
        cv2.putText(frame, f"MAR: {self.current_mar:.3f} / {self.MOUTH_AR_THRESH}", (info_x, 85),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)

    # Indicadores de estado con colores
    eye_color = (0, 0, 255) if self.current_ear < self.EYE_AR_THRESH else (0, 255, 0)
    mouth_color = (0, 255, 255) if self.current_mar > self.MOUTH_AR_THRESH else (0, 255, 0)

    cv2.putText(frame, f"OJOS: {'CERRADOS' if self.current_ear < self.EYE_AR_THRESH else 'ABIERTOS'}",
                (info_x, 125), cv2.FONT_HERSHEY_SIMPLEX, 0.4, eye_color, 1)
    cv2.putText(frame, f"BOCA: {'ABIERTA' if self.current_mar > self.MOUTH_AR_THRESH else 'CERRADA'}",
                (info_x, 145), cv2.FONT_HERSHEY_SIMPLEX, 0.4, mouth_color, 1)

```

```

        # Contadores en tiempo real (lado izquierdo)
        cv2.putText(frame, f"Ojos cerrados: {self.eyes_counter}/{self.EYE_AR_CONSEC_FRAMES}",
                    (10, 60), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 0), 1)
        cv2.putText(frame, f"Boca abierta: {self.mouth_counter}/{self.MOUTH_AR_CONSEC_FRAMES}",
                    (10, 85), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 0), 1)
    else:
        cv2.putText(frame, "Esperando deteccion...", (info_x, 85),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (128, 128, 128), 1)

    # Tiempo de sesión
    session_time = int(time.time() - self.start_time)
    cv2.putText(frame, f"Tiempo: {session_time//60}:{session_time%60:02d}",
                (info_x, 105), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)

    # Panel de estadísticas totales (parte inferior)
    cv2.rectangle(frame, (5, 350), (635, 480), (0, 0, 0), -1)

    # Estadísticas principales
    cv2.putText(frame, "=== ESTADISTICAS TOTALES ===", (200, 375),
                cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 0), 2)

    cv2.putText(frame, f"SOMNOLENCIA DETECTADA: {self.drowsiness_count} veces",
                (10, 400), cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
    cv2.putText(frame, f"BOSTEZOS DETECTADOS: {self.yawn_count} veces",
                (10, 425), cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 255), 2)

    # Información de capturas
    cv2.putText(frame, f"IMAGENES CAPTURADAS: {self.images_saved}",
                (10, 450), cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 2)

    # Total de detecciones
    total_events = self.drowsiness_count + self.yawn_count
    cv2.putText(frame, f"EVENTOS TOTALES: {total_events}",
                (350, 400), cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 0, 255), 2)
|

    # Estado del sistema
    status_color = (0, 255, 0) if self.face_detected else (0, 165, 255)
    status_text = "MONITOREO ACTIVO" if self.face_detected else "SIN ROSTRO"
    cv2.putText(frame, f"ESTADO: {status_text}",
                (350, 425), cv2.FONT_HERSHEY_SIMPLEX, 0.6, status_color, 2)

    # Configuración actual
    cv2.putText(frame, f"Config: EAR<{self.EYE_AR_THRESH} MAR>{self.MOUTH_AR_THRESH}",
                (350, 450), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (128, 128, 128), 1)

def process_frame(self, frame):
    """Procesar frame"""
    try:
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = self.detector(gray)

        self.face_detected = len(faces) > 0

        if not self.face_detected:
            self.no_face_counter += 1
            # Mostrar interfaz sin rostro pero con todos los datos
            self.draw_no_face_interface(frame)
            return frame

        self.no_face_counter = 0

        for face in faces:
            landmarks = self.predictor(gray, face)
            landmarks = np.array([[p.x, p.y] for p in landmarks.parts()])

            # Extraer coordenadas
            left_eye = landmarks[self.LEFT_EYE]
            right_eye = landmarks[self.RIGHT_EYE]
            mouth = landmarks[self.MOUTH]

```

```

# Calcular ratios
left_ear = self.calculate_eye_aspect_ratio(left_eye)
right_ear = self.calculate_eye_aspect_ratio(right_eye)
self.current_ear = (left_ear + right_ear) / 2.0
self.current_mar = self.calculate_mouth_aspect_ratio(mouth)

# Dibujar contornos
cv2.drawContours(frame, [cv2.convexHull(left_eye)], -1, (0, 255, 0), 1)
cv2.drawContours(frame, [cv2.convexHull(right_eye)], -1, (0, 255, 0), 1)
cv2.drawContours(frame, [cv2.convexHull(mouth)], -1, (255, 0, 0), 1)

# Detección de somnolencia - CUENTA CADA EVENTO Y CAPTURA IMAGEN
if self.current_ear < self.EYE_AR_THRESH:
    self.eye_counter += 1
    if self.eye_counter >= self.EYE_AR_CONSEC_FRAMES and not self.drowsiness_detected:
        self.drowsiness_count += 1
        self.drowsiness_detected = True
        self.log_event("SOMNOLENCIA DETECTADA")
        # CAPTURA AUTOMÁTICA DE IMAGEN
        self.capture_image(frame, "somnolencia")
        print(f"⚠ SOMNOLENCIA detectada #{self.drowsiness_count} - Imagen capturada")
else:
    self.eye_counter = 0
    self.drowsiness_detected = False

# Detección de bostezos - CUENTA CADA EVENTO Y CAPTURA IMAGEN
if self.current_mar > self.MOUTH_AR_THRESH:
    self.mouth_counter += 1
    if self.mouth_counter >= self.MOUTH_AR_CONSEC_FRAMES and not self.yawn_detected:
        self.yawn_count += 1
        self.yawn_detected = True
        self.log_event("BOSTEZO DETECTADO")
        # CAPTURA AUTOMÁTICA DE IMAGEN
        self.capture_image(frame, "bostezo")
        print(f"😴 BOSTEZO detectado #{self.yawn_count} - Imagen capturada")
    else:
        self.mouth_counter = 0
        self.yawn_detected = False

# Mostrar mensajes de detección grandes y claros
if self.drowsiness_detected and self.eye_counter >= self.EYE_AR_CONSEC_FRAMES:
    cv2.rectangle(frame, (10, 270), (630, 320), (0, 0, 255), -1)
    cv2.putText(frame, "!!! SOMNOLENCIA DETECTADA !!!", (50, 300),
                cv2.FONT_HERSHEY_DUPLEX, 1.0, (255, 255, 255), 3)

if self.yawn_detected and self.mouth_counter >= self.MOUTH_AR_CONSEC_FRAMES:
    cv2.rectangle(frame, (10, 320), (630, 350), (0, 255, 255), -1)
    cv2.putText(frame, "!!! BOSTEZO DETECTADO !!!", (80, 340),
                cv2.FONT_HERSHEY_DUPLEX, 0.8, (0, 0, 0), 2)

# Dibujar todas las estadísticas
self.draw_statistics(frame, show_current_values=True)

except Exception as e:
    print(f"Error procesando frame: {e}")
    # Mostrar estadísticas aunque haya error
    self.draw_statistics(frame, show_current_values=False)

return frame

def run(self):
    """Ejecutar detector"""
    print("🚀 Iniciando detector con captura automática...")
    print(f"📄 Archivo de datos: {self.data_file}")
    print(f"📁 Carpeta de imágenes: {self.images_folder}")
    print("⚠ Controles:")
    print("    'q' o ESC → Salir")
    print("    'r' → Resetear contadores")
    print("    'c' → Mostrar configuración")
    print("    'i' → Mostrar información de imágenes")

```

```

print("🔍 CONFIGURACIÓN ACTUAL:")
print(f"    Umbral ojos (EAR): {self.EYE_AR_THRESH}")
print(f"    Umbral boca (MAR): {self.MOUTH_AR_THRESH}")
print(f"    Frames ojos: {self.EYE_AR_CONSEC_FRAMES}")
print(f"    Frames boca: {self.MOUTH_AR_CONSEC_FRAMES}")
print(f"    Cooldown captura: {self.capture_cooldown}s")

# Inicializar cámara
cap = cv2.VideoCapture(0)
if not cap.isOpened():
    print("❌ No se pudo abrir la cámara")
    return

# Configurar cámara
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
cap.set(cv2.CAP_PROP_BUFFERSIZE, 1)

print("✅ Cámara lista")
print("📷 Captura automática ACTIVADA - Se guardarán imágenes al detectar eventos")

# Crear ventana UNA SOLA VEZ
window_name = 'Detector de Somnolencia - Auto Captura - Mauricio Lalaleo'
cv2.namedWindow(window_name, cv2.WINDOW_NORMAL)

try:
    frame_count = 0
    while True:
        ret, frame = cap.read()
        if not ret:
            print("❌ Error leyendo cámara")
            break

        # Procesar CADA frame
        frame = self.process_frame(frame)

        # Mostrar frame EN LA MISMA VENTANA
        cv2.imshow(window_name, frame)

        # Control de salida
        key = cv2.waitKey(1) & 0xFF
        if key == ord('q') or key == 27: # 'q' o ESC
            print("👋 Saliendo...")
            break
        elif key == ord('r'): # 'r' para resetear contadores
            self.drowsiness_count = 0
            self.yawn_count = 0
            self.eyes_counter = 0
            self.mouth_counter = 0
            self.images_saved = 0
            print("🔄 Contadores reseteados")
        elif key == ord('c'): # 'c' para mostrar configuración
            print("🔍 CONFIGURACIÓN ACTUAL:")
            print(f"    EAR (ojos): {self.EYE_AR_THRESH} | MAR (boca): {self.MOUTH_AR_THRESH}")
            print(f"    Frames ojos: {self.EYE_AR_CONSEC_FRAMES} | Frames boca: {self.MOUTH_AR_CONSEC_FRAMES}")
            print(f"    Cooldown captura: {self.capture_cooldown}s")
        elif key == ord('i'): # 'i' para información de imágenes
            print(f"📷 INFORMACIÓN DE CAPTURAS:")
            print(f"    Imágenes guardadas: {self.images_saved}")
            print(f"    Carpeta: {self.images_folder}")
            print(f"    Última captura hace: {int(time.time() - self.last_capture_time)}s")

        frame_count += 1

        # Prevenir bucle infinito
        if frame_count > 1000000:
            frame_count = 0

except KeyboardInterrupt:
    print("\n🛑 Interrupción por teclado")
except Exception as e:
    print(f"Error: {e}")

```

```

        print(f"Error durante ejecución: {e}")
    finally:
        # Limpiar recursos
        print("🧹 Limpiando recursos...")
        try:
            cap.release()
            cv2.destroyAllWindows()
            cv2.destroyAllWindows()
            # Forzar limpieza
            for i in range(5):
                cv2.waitKey(1)
        except:
            pass

    print("🔴 Detector detenido")
    print(f"📄 RESUMEN FINAL:")
    print(f"    Eventos de somnolencia: {self.drowsiness_count}")
    print(f"    Bostezos detectados: {self.yawn_count}")
    print(f"    Imágenes somnolencia: {self.images_saved_drowsiness}")
    print(f"    Imágenes bostezos: {self.images_saved_yawns}")
    print(f"    Total imágenes: {self.images_saved_drowsiness + self.images_saved_yawns}")
    print(f"    Tiempo total: {int(time.time() - self.start_time)}s")
    print(f"    Carpeta somnolencia: {self.drowsiness_folder}")
    print(f"    Carpeta bostezos: {self.yawns_folder}")

def check_requirements():
    """Verificar archivos necesarios"""
    if not os.path.exists("shape_predictor_68_face_landmarks.dat"):
        print("❌ Archivo shape_predictor_68_face_landmarks.dat no encontrado")
        return False
    print("🔴 Archivo de landmarks encontrado")
    return True

def main():
    """Función principal"""

    print("🔍 Detector de Somnolencia con Captura Automática")
    print("👤 Uisrael Mauricio Lalaleo - Universidad Israel")
    print("-" * 60)

    if not check_requirements():
        print("❌ Faltan archivos necesarios")
        return

    try:
        detector = SimpleDrowsinessDetector()
        detector.run()
    except Exception as e:
        print(f"Error iniciando: {e}")

if __name__ == "__main__":
    main()

```
