



**IMPLEMENTACION DE UN SISTEMA PILOTO PARA LA DETECCION DE
CANSANCIO EN CONDUCTORES MEDIANTE MACHINE LEARNING**

Cristian David Guarnizo Rengifo

Código:20441821406

Danilo Cortes Silvestre

Código:20441826461

Universidad Antonio Nariño

Programa Ingeniería Electrónica

Facultad de Ingeniería Mecánica, Electrónica y Biomédica

Ibagué, Colombia

2023

**Implementación de un sistema piloto para la detección de cansancio en
conductores mediante métodos de Machine learning**

Cristian David Guarnizo Rengifo

Danilo Cortes Silvestre

Proyecto de grado presentado como requisito parcial para optar al título de:

Ingeniero Electrónico

Director (a):

Ingeniero Electrónico SERGIO ALEJANDRO ORJUELA

Codirector (a):

Ingeniero Electrónico RICARDO PINO DIAZ

Línea de Investigación:

Electrónica Digital

Universidad Antonio Nariño

Programa Ingeniería Electrónica

Facultad de Ingeniería Mecánica, Electrónica y Biomédica

Ibagué, Colombia

2023

NOTA DE ACEPTACIÓN

El trabajo de grado titulado

_____.

Cumple con los requisitos para optar

Al título de _____.

Firma del Tutor

Firma Jurado

Firma Jurado

Ciudad, Día Mes Año.

Dedicatoria

*"A los obstáculos que encontré en mi camino:
ustedes me recordaron que cada desafío es
una oportunidad de crecimiento."*

Agradecimientos

Quiero expresar mi más sincero agradecimiento a las personas e institución que hicieron posible la culminación de este proyecto de vida

En primer lugar, agradezco de todo corazón a mis padres, quienes han sido una fuente inagotable de apoyo, amor y sacrificio a lo largo de esta travesía académicas. Su esfuerzo y confianza en mi son invaluable, y este logro es también suyo

A la universidad Antonio Nariño por brindarme la oportunidad de adquirir conocimientos, crecer como persona y hacer realidad este sueño.

A mis queridos compañeros de carrera, quienes no solo compartieron aulas y proyectos, sino también risas, desafíos y momentos inolvidables.

Este proyecto no habría sido posible sin el respaldo de todas estas personas y entidades, y les estoy profundamente agradecido. Cada uno de ustedes ha dejado una huella imborrable en este viaje y ha contribuido significativamente a mi éxito. Gracias.

Danilo Cortes Silvestre

Agradecimientos

Quisiera expresar mi profundo agradecimiento a todas las personas que contribuyeron de una u otra manera a la realización de esta tesis. Sus apoyos, orientación y aliento fueron fundamentales en este largo y arduo proceso de investigación.

También quiero expresar mi gratitud a mi familia, mis padres, hermano y demás familiares, por su inquebrantable apoyo emocional y financiero. Sin su amor y confianza, esta tesis no habría sido posible.

Por último, pero no menos importante, agradezco a todas las personas anónimas cuyas obras, investigaciones y contribuciones previas en el campo de estudio fueron una fuente fundamental de referencia, inspiración y motivación para culminar esta investigación.

Este logro es el resultado del esfuerzo y el apoyo de muchas personas, y por eso quiero agradecer a todos los que hicieron posible la culminación de esta tesis.

¡Gracias!

Cristian David Guarnizo Rengifo

Contenido

Pág.

Contenido	
Resumen.....	11
Abstract.....	12
Introducción	13
Objetivos	17
Justificación	18
Definiciones.....	19
1. CAPITULO 1.....	21
1.1 Marco teórico.....	21
1.1.1 Machine learning	28
1.1.2 Deep learning.....	31
1.1.3 Redes Neuronales	32
1.2 Tecnologías usadas.....	36
1.2.1 Jetson TX2.....	36
1.2.2 Hardware	36
1.2.3 Software	36
1.3 Imágenes	37
1.3.1 Procesamiento Digital de Imágenes.....	38
1.3.2 Etiquetas	41
1.3.3 Etapas clave en la clasificación de imágenes	42
2. CAPITULO 2.....	43
2.1 Metodología	43

2.1.1	Reconocimiento de muestras de sueño y cansancio en conductores.....	44
2.1.2	Investigación y elección de herramientas de software y hardware	47
2.1.3	Instalación del sistema y herramientas en la Jetson TX2.....	50
2.2	Base de datos Drowsy Driver Detection Dataset	56
2.2.1	Procesamiento y análisis de datos.....	57
2.2.2	Clasificación y gestión de imágenes en carpetas	59
2.3	Obtención de imágenes.....	64
3.	CAPITULO 3.....	69
3.1	Preprocesamiento de Datos.....	69
3.1.1	División en Conjunto de Entrenamiento y Prueba	71
3.1.2	Entrenamiento del Modelo	72
3.1.3	Preparación del conjunto de datos de entrenamiento para la detección de somnolencia en conductores.....	73
3.2	Proceso de entrenamiento	76
4.	CAPITULO 4.....	85
4.1	Conclusiones y recomendaciones.....	85
4.1.1	Conclusiones.....	85
4.1.2	Recomendaciones	87
	Referencias Bibliográficas.....	89
	Bibliografía	89

Lista de Figuras

Figura 1 Componentes de la Fatiga.....	21
Figura 2 Técnicas de medición somnolencia conductor.....	25
Figura 3 Señal de control sistema de advertencia.....	26
Figura 4 Gestos del conductor.....	27
Figura 5 Aprendizaje supervisado.....	30
Figura 6 Subíndice inteligencia artificial.....	32
Figura 7 estructura jerárquica de un sistema basado en RNA.....	33
Figura 8 Modelo genérico de una neurona artificial.....	35
Figura 9 Adquisición de imagen Digital.....	37
Figura 10 pixeles número digital.....	40
Figura 11 Etiquetas detección de cansancio.....	41
Figura 12 Etapas metodología.....	43
Figura 13 Jetson TX2.....	45
Figura 14 Comportamiento físico presencia de fatiga.....	46
Figura 15 Instalación imagen Jetson TX2.....	52
Figura 16 System configuration-Licencias.....	52
Figura 17 Usuario Jetson tx2.....	53
Figura 18 Instalación de sistema operativo.....	53
Figura 19 Interfaz sistema operativo Jetson.....	54
Figura 20 Instalación componentes SDK.....	54
Figura 21 Transferencia de archivos Jetson TX2.....	55
Figura 22 Proceso final instalación Jetson TX2.....	55
Figura 23 dataset Ojo humano.....	58
Figura 24 Flujo óptico.....	60
Figura 25 Ejemplo de Interfaz captura de datos.....	65
Figura 26 funcionamiento experimental del algoritmo.....	67
Figura 27 Datos MRL Eye Dataset.....	71
Figura 28 Código entrenamiento del modelo.....	75
Figura 29 Código ejecución tiempo real.....	77
Figura 30 Prueba funcionamiento cámara Jetson TX2.....	78
Figura 31 Detección de fatiga.....	79

Lista de tablas

Pág.

Tabla 1:Etapas clasificación de imágenes	42
Tabla 2 Clasificación del proceso de entrenamiento	59

Resumen

El cansancio en conductores se ha convertido en un problema de gran relevancia en términos de seguridad vial. Por tal motivo una opción para contrarrestar los accidentes automovilísticos ocasionados por micro sueños se emplean métodos de detección facial y corporal usando herramientas basadas en machine Learning que permiten la vigilancia constante de las expresiones faciales y los movimientos corporales con el propósito de generar una alerta. Sin embargo, cuando nos enfocamos en el entorno de conductores promedio la disponibilidad de recursos y aplicaciones de nuevas tecnologías se ve limitada. La problemática aborda la necesidad de crear un sistema capaz de detectar los gestos y movimientos faciales y/o corporales en condiciones de sueño. En este sentido, se propone generar un modelo de aprendizaje autónomo utilizando machine Learning mediante un sistema paralelo integrado a la placa de desarrollo Jetson tx2 que permita emitir una alerta cuando el conductor presente síntomas de sueño.

El alcance de nuestro proyecto se basa en un monitoreo constante de señales de sueño que se presenta en los conductores, aplicando herramientas de machine Learning para procesar y generar un análisis eficiente de la información. Como resultado se activará una alarma visual o auditiva en caso de ser necesario. Además, consideramos la posibilidad de implementar el sistema piloto a vehículos como medida preventiva para reducir accidentes de tránsito

Palabras claves: machine learning-jetson tx2-sistema piloto

Abstract

Fatigue in drivers has become a significant issue in terms of road safety. To address the car accidents caused by microsleep episodes, facial and body detection methods are employed using machine learning-based tools, allowing constant monitoring of facial expressions and body movements to generate alerts. However, when focusing on the average driver's environment, the availability of resources and new technology applications is limited. The problem at hand addresses the need to create a system capable of detecting facial and/or body gestures and movements during sleep conditions. In this regard, we propose to generate an autonomous learning model using machine learning through a parallel system integrated with the Jetson TX2 development board to issue an alert when the driver shows signs of drowsiness.

The scope of our project is based on continuous monitoring of sleep signals in drivers, applying machine learning tools to process and generate efficient information analysis. As a result, a visual or auditory alarm will be activated if necessary. Furthermore, we consider the possibility of implementing the pilot system in vehicles as a preventive measure to reduce traffic accidents.

Keywords: machine learning-jetson tx2-pilot system

Introducción

Los conductores al presentar signos de cansancio obtienen deficiencias al responder a estímulos y eventos cruciales, disminuyendo una variedad de habilidades cognitivas y de desempeño psicomotor viéndose afectado la atención multitarea, la memoria, la coordinación y la percepción visuoespacial, esto conlleva a cometer errores y aumentar la posibilidad de sufrir un accidente de tráfico. La detección temprana del sueño en conductores podrá prevenir accidentes de tráfico y de esta forma salvar vidas. En este contexto, la tecnología ofrece una solución prometedora usando métodos de machine learning (Boyle, 2008)

En los últimos años ha aumentado el interés por investigar y desarrollar modelos predictivos para evitar el riesgo de colisión debido a la fatiga, aplicando técnicas de procesamiento digital de imágenes y aprendizaje profundo. Existen distintos estudios que han investigado la detección de fatiga y cansancio en conductores usando tecnologías

basadas en sensores como electroencefalogramas y una variedad de medidas de rendimiento en simuladores de conducción en tiempo real, Sin embargo, estas tecnologías son muy efectivas, pero tienen limitaciones como el costo y la complejidad al usar estos elementos, (Henry, 2006)

Actualmente se han desarrollado importantes estudios utilizando métodos de machine learning puesto que al entrenar un algoritmo se logra la capacidad de aprender e identificar patrones en datos para tener una respuesta optima, entre los últimos estudios realizados se encuentra el trabajo propuesto por (Selvaraj, 2019) titulado “Driver Drowsiness Detection Using Machine Learning Techniques” en donde los autores realizan una investigación

robusta sobre la detección de fatiga en conductores mediante técnicas de machine learning en el que incluyen a su investigación métodos como: redes neuronales, árboles de decisión, support vector Machine (SVM) y K-nearest neighbors en donde indagan y analizan estos métodos que se utilizan para entrenar y enseñar modelos para aplicarlos en cámaras, sensores fisiológicos y datos en el vehículo (Selvaraj, 2019)

El cansancio en conductores se destaca como un factor primordial en el aumento de los incidentes viales. A pesar de la recomendación de la fundación CEA de disfrutar de 8 horas diarias de sueño, las estadísticas revelan una realidad preocupante, aproximadamente el 19% de la población no logra cumplir este ideal, mientras que el 91% por distintos motivos no puede escoger su horario de descanso, aunque la actividad del sueño va disminuyendo después de los 50 años, los accidentes por cansancio y fatiga se presentan en jóvenes entre la edad de 18 y 20 años. A pesar de que se han desarrollado dispositivos capaces de alertar al conductor cuando presente síntomas de sueño, estos en su mayoría suelen ser costosos y estar fuera del alcance en gran parte de los vehículos (Fundacion_CEA)

De acuerdo con datos entregados por el instituto nacional de medicina legal y ciencias forenses indican que el 28% de las muertes más violentas en Colombia en lo corrido del año 2021 fueron causados por accidentes de tránsito, en la cual su mayoría fueron ocasionados por micro sueños, según expertos aseguran que si un conductor llega a tener un micro sueño mientras se desplaza en su automóvil a 80 Km por hora puede recorrer 45 metros sin algún tipo de reacción. También la agencia Nacional de Seguridad Vial entrega estadísticas en donde indica que entre las 02:00 y 05:00 de la madrugada a su vez 14:00 y

16:00 de la tarde son los momentos en donde el sueño se presenta con mayor frecuencia en los conductores (Villamizar, 2022)

Dado este problema se crea la necesidad de diseñar e implementar un sistema piloto para la detección de cansancio en conductores utilizando técnicas de machine learning, como lo es aprendizaje profundo basado en un sistema de redes neuronales, y un lenguaje de programación llamado Python; simultáneamente conectado a una placa de Desarrollo Jetson TX2. Al abordar este problema, se utilizaron técnicas de reconocimiento de patrones y procesamiento Digital de imágenes para detectar características que posee el conductor, se realizaron análisis en el momento exacto en donde el conductor inicia con síntomas de sueño guiándonos del rostro y algunos movimientos corporales tomando como referencia el parpadeo frecuente cierre parcial de ojos ,inclinación de cabeza, bostezos, y toques constantes de las manos al rostro

La realización de este proyecto puede desempeñar un papel importante en los esfuerzos actuales para elevar los estándares de seguridad vial y reducir la incidencia de accidentes de tráfico causados por la fatiga presentada en los conductores.

En el primer capítulo se centra en la presentación de los conceptos claves necesarios para comprender el mundo de machine learning, lo cual abarca las redes neuronales, librerías, entre otros términos necesarios. También se aborda en este capítulo la definición de imagen, y todo el proceso de su procesamiento

En el segundo capítulo, se adentra en la metodología usada, comenzando desde el procesamiento de datos hasta la configuración inicial del sistema piloto

En el tercer capítulo, se presentan los resultados del correcto funcionamiento de nuestro sistema piloto, concluyendo así la parte teórico practica

Finalmente, el cuarto capítulo, resume los resultados de nuestro proyecto y proporciona recomendaciones para ser usadas en investigaciones futuras

Objetivos

Objetivo general

Implementar un sistema con montaje en laboratorio, que funcione para la detección de somnolencia en conductores e identificar estos datos para la activación de un sistema de alarma que indique a los conductores cuándo tomar medidas preventivas mediante métodos de procesamiento de imágenes y machine learning.

Objetivo específico

1. Analizar técnicas avanzadas de procesamiento digital de imágenes, como detección de bordes segmentación y reconocimientos de objetos
2. Plantear y emplear algoritmos de aprendizaje automático como redes neuronales para el análisis de las imágenes y la detección de patrones que puedan indicar cansancio en el conductor
3. Diseñar un prototipo funcional utilizando la Jetson TX2 del sistema piloto de detección de cansancio.
4. Proporcionar un sistema de alerta que notifique al conductor en tiempo real si se detecta un nivel alto de fatiga o cansancio utilizando señales visuales o auditivas que no distraigan la atención del conductor y que le permita tomar medidas preventivas

Justificación

La creación del sistema piloto sobre la detección de cansancio en conductores mediante machine learning se justifica por la necesidad de poder prevenir y evitar accidentes automovilísticos logrando salvar vidas. Se conoce que los conductores con síntomas de sueños pierden habilidades psicomotoras disminuyendo el tiempo de reacción y llevándolos a una posible tragedia

Según el director de agencia nacional de seguridad vial "Un micro sueño se caracteriza por durar tan solo unos segundos en los que el cerebro pareciera "DESCONECTARSE". Es tan corto que muchos conductores ni siquiera se percatan, sin embargo, estos espacios de desconexión al volante elevan al máximo la probabilidad de un siniestro vial." (Lota, 2020). Además, el proyecto es adecuado para solucionar el problema identificado, ya que hace uso de tecnologías avanzadas como machine learning y entre otras para detectar y prevenir la fatiga en los conductores. La aplicación del machine learning permite un análisis preciso y eficiente de los patrones de sueño y fatiga en los conductores lo que mejora la capacidad del sistema para detectar de manera oportuna los síntomas de fatiga y alertar al conductor

El proyecto también es importante ya que el cansancio presentado en los conductores es un problema común en muchos países y ha sido objetivo de investigación en distintas partes del mundo. se ha identificado un gran interés por parte de entidades como organizaciones gubernamentales y empresas privadas en la solución de este problema.

Definiciones

- **Machine Learning:** Machine learning o aprendizaje automático en español, es un rama de la inteligencia artificial que se centra en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender y mejorar su rendimiento a través de la experiencia y el análisis de datos obteniendo un mayor rendimiento en tareas específicas. En lugar de ser programadas utilizan algoritmos para identificar patrones
- **Jetson Tx2:** es una plataforma de computación pequeña y potente diseñada para aplicaciones de inteligencia artificial y aprendizaje profundo puede procesar datos y realizar tareas de inteligencias artificial directamente en el dispositivo sin depender de una conexión a internet
- **Redes Neuronales Artificiales:** Consisten en capas de neuronas artificiales interconectadas que se utilizan en tareas de aprendizaje automático, como reconocimiento de patrones, procesamiento de lenguaje natural y visión por computadora.
- **Conjunto de Datos (Dataset):** Es un conjunto de ejemplos o instancias utilizados para entrenar, validar o probar algoritmos de aprendizaje automático. Cada ejemplo en un conjunto de datos contiene información sobre entradas y salidas, lo que permite al algoritmo aprender de los datos de entrenamiento.
- **Pesos Sinápticos:** En el contexto de redes neuronales, los pesos sinápticos son parámetros ajustables que determinan la importancia de las conexiones entre

neuronas. Estos pesos se ajustan durante el entrenamiento de la red para que esta pueda aprender y realizar tareas específicas.

- **Función de Coste:** Es una métrica que se utiliza para evaluar el estado óptimo del aprendizaje automático durante el entrenamiento, el objetivo es minimizar esta función para que el modelo haga predicciones precisas.
- **Aprendizaje Supervisado:** Es un enfoque de aprendizaje automático en el que un algoritmo se entrena utilizando un conjunto de datos etiquetado que contiene ejemplos de entrada y las salidas deseadas correspondientes. El algoritmo aprende a hacer predicciones basadas en ejemplos etiquetados.
- **Modelado de Sinapsis:** Se refiere al proceso de ajuste de los pesos sinápticos que conectan las neuronas en una red neuronal artificial. Durante el entrenamiento, los algoritmos de aprendizaje ajustan estos pesos para que la red pueda realizar tareas específicas.
- **Memoria Distribuida:** En el contexto de redes neuronales, se refiere a la capacidad de almacenar información y conocimiento de manera distribuida en múltiples neuronas, en lugar de en una ubicación central. Esto permite que las redes neuronales capturen patrones complejos.
- **Adaptabilidad al Entorno:** La capacidad de una red neuronal para ajustarse y adaptarse a cambios en el entorno o en los datos de entrada.

1. CAPITULO 1

1.1 Marco teórico

Cansancio en Conductores

El cansancio en conductores o más conocido con el termino de fatiga, es catalogado como uno de los principales factores de riesgo el cual tiene una acción directa o indirecta en accidentes. Cuando hablamos de la fatiga, se refiere al agotamiento excesivo que provoca un deterioro en la capacidad de llevar a cabo actividades cotidianas el cual puede ser causado por una carga de trabajo excesiva o un prolongado estado de cansancio, lo que conlleva a mostrar disminución de la función muscular y mental. Para detectarla, se puede distinguir entre dos factores clave: El cansancio físico, también conocida como fatiga muscular, o cansancio mental. El efecto de fatiga se divide en tres aspectos esenciales:

Figura 1 Componentes de la Fatiga



Fuente: Imagen autoría propia

Estos efectos presentados a la hora de conducir surgen debido a excesos en tareas cotidianas poniendo como ejemplo principal el trabajo, junto con una falta de tiempo adecuado para descansar

Es fundamental resaltar que el cansancio en conductores no se limita únicamente al trabajo, sino que también puede estar conectado a la realización de tareas cotidianas

La conducción, especialmente en trayectos largos, puede considerarse como una actividad de alto agotamiento físico-mental. Esto se debe a factores como la repetición de tareas, la tensión en ciertos grupos musculares debido a una postura mantenida y constante en la silla del vehículo y la posición de los pies en los pedales, a su vez la necesidad de mantenerse alerta en la carretera genera estrés y fatiga. Cuando estos problemas se combinan con períodos de descanso insuficientes, el cansancio puede representar no solo como un riesgo en la seguridad del conductor, sino que también en la seguridad de otros conductores

Se examinan las circunstancias del entorno del conductor en donde aumenta el estado de cansancio. Fuente extraída del artículo (Unidad de Formación y Comportamiento de conductores, 2012)

- La primera circunstancia que afecta al momento de conducir el vehículo es cuando se presentan vibraciones constantes y una mala iluminación esto conlleva a aumentar las posibilidades de sueño

- La segunda circunstancia que afecta la movilidad por carreteras y aumenta la posibilidad de quedarse dormido es la conducción por vías ya recorridas, la conducción en horas de la noche y las condiciones climáticas como exceso de calor también pueden contribuir.

- La edad, y estados emocionales también son factores influyentes donde aumenta el riesgo de cansancio en conductores

- El cansancio visual es uno de los temas más importantes a la hora de conducir, ya que la capacidad de los ojos para mantener una imagen nítida en la retina se mejora notoriamente con un descanso adecuado. Los síntomas de fatiga se encuentran principalmente en entornos con escasa iluminación, como la conducción nocturna y recorridos continuos

Para prevenir la fatiga, es fundamental reconocer sus síntomas como lo son:

Variación en el conocimiento: La fatiga afecta la variación sensorial del entorno, especialmente la percepción visual. La fatiga visual suele manifestarse primero, debido a la reducción del movimiento ocular provocada por el cansancio.

Cansancio físico: La fatiga puede dar lugar a sensaciones de cansancio y enfermedad como dolor de cabeza, dolor muscular(especialmente en el cuello, espalda y los brazos), tensión muscular, mareos, rigidez y disminución de la movilidad .

Disminución de la vigilancia: La concentración en la conducción disminuye a medida que aumenta la duración del viaje, lo que hace que mantener la atención sea una tarea desafiante en viajes largos.

Disminución instintiva : En situaciones peligrosas es importante que el conductor actúe de manera rápida y precisa. Sin embargo, la fatiga acorta el tiempo de reacción ante eventos inesperados.

Descoordinación en movimientos : La precisión, coordinación y velocidad de los movimientos necesarios para la conducción se ven significativamente afectados por la fatiga.

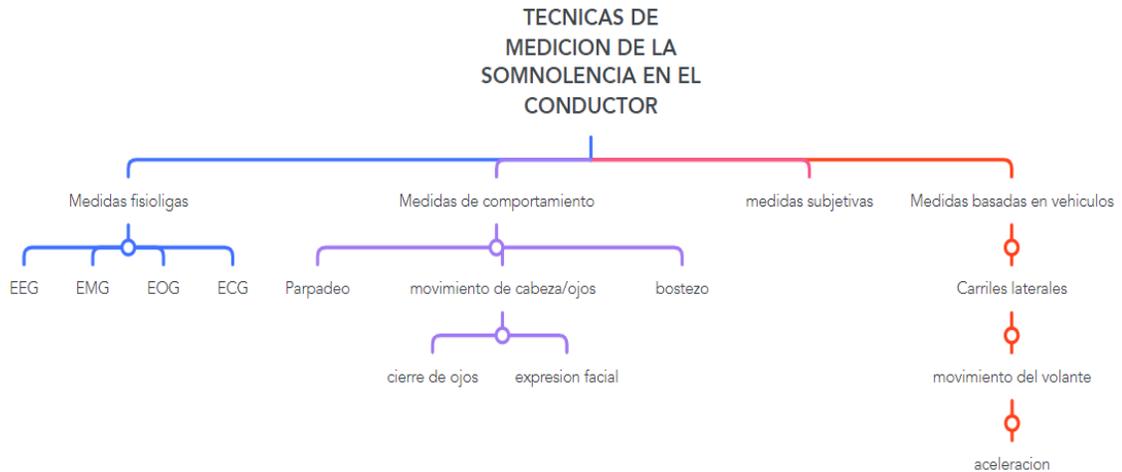
Estos factores pueden dar lugar a comportamientos inapropiados, como la aceleración involuntaria, falta de respeto de la distancia de seguridad, maniobras bruscas en las curvas y cambios en la trayectoria, aumentando el riesgo de un accidente automovilístico.

Para combatir la fatiga, es fundamental reconocer la necesidad de descansar. La fatiga solo desaparece con el descanso adecuado. Ignorar la fatiga y continuar conduciendo pone en peligro la seguridad en la carretera. El uso de energizantes no es una solución inteligente, ya que oculta la fatiga sin eliminarla. Es esencial comprender que la fatiga y el sueño solo se superan con el descanso adecuado. (Unidad de Formación y Comportamiento de conductores, 2012)

En este contexto el machine learning se presenta como una herramienta prometedora para abordar el desafío del cansancio en conductores. La fatiga como se hizo énfasis anteriormente es un problema significativo que afecta la seguridad tanto del conductor como el resto de conductores. La detección temprana de signos de cansancio puede marcar significativamente la diferencia al disminuir los altos índices de accidentalidad

Técnicas y Medidas de somnolencia en el conductor

Figura 2 Técnicas de medición somnolencia conductor

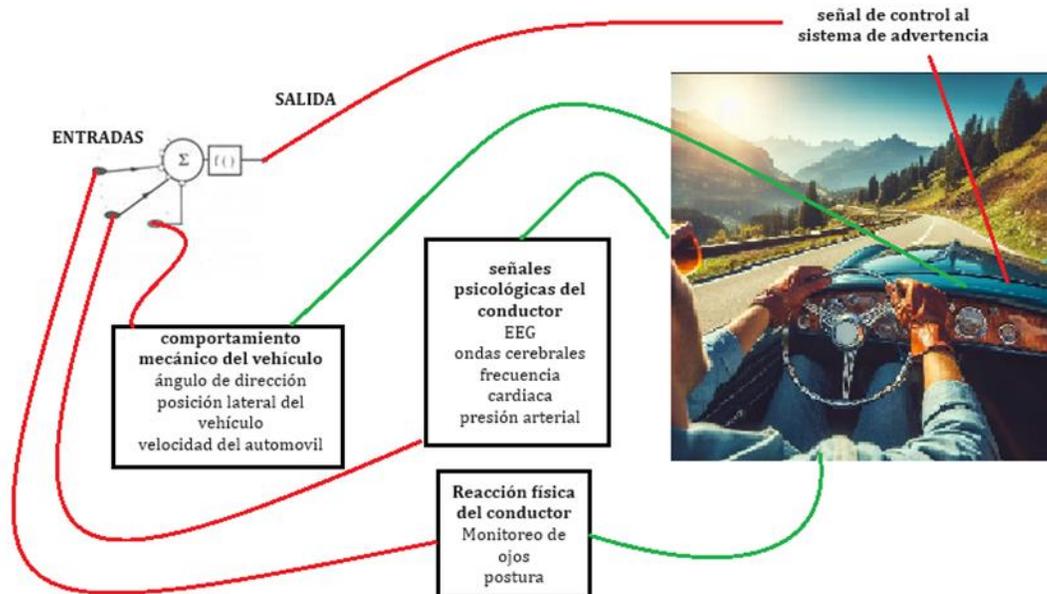


Fuente: Imagen autoría propia mapa conceptual técnicas de medición de somnolencia

La detección de la somnolencia en conductores se divide en tres partes fundamentales:

- **Relacionada con el vehículo**
- **comportamiento del conductor**
- **aspectos fisiológicos**

Figura 3 Señal de control sistema de advertencia



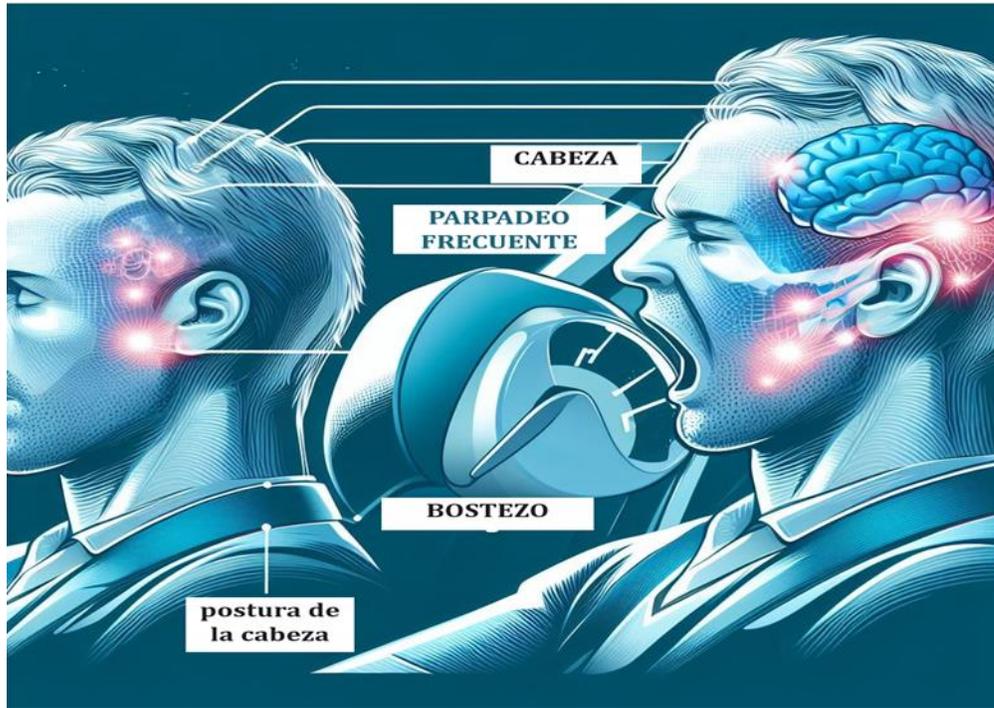
Fuente: Imagen de autoría propia

se ilustran las tres distintas aproximaciones para la detección de la somnolencia. Estos tres parámetros conforman la base de la detección de la somnolencia. Un análisis detallado de estas mediciones proporcionará una visión de los sistemas actuales, los desafíos presentados y las mejoras necesarias para desarrollar un sistema robusto. Se hace seguimiento constante a unas múltiples métricas, como las desviaciones o cambio de carril, inclinaciones del volante, entre otros. Cualquier alteración en estos parámetros que supere un umbral predefinido señala de manera significativa un aumento en la probabilidad de que el conductor se encuentre somnoliento.

Comportamiento físico del conductor: El comportamiento físico-mental del conductor el cual incluye postura de la cabeza, parpadeo de ojos, bostezos entre otros. Se monitorea a

través de una cámara y se alerta al conductor si se detecta algunos de estos síntomas de somnolencia

Figura 4 Gestos del conductor



Otras técnicas de Detección de somnolencia: Según la investigación **Driver Drowsiness Detection System and Techniques: A Review** existen varias categorías de tecnologías que pueden analizar y detectar a tiempo el cansancio en conductores. En la primera se hace mención al uso de cámaras para rastrear el comportamiento de una persona, incluyendo el monitoreo de pupilas, seguimiento de la boca para detectar bostezos, posición en la cabeza y otros factores. La segunda tecnología hace énfasis en el reconocimiento de voz. A menudo la voz de una persona puede tener pistas importantes sobre cuan cansada está el conductor. Así mismo se mencionan técnicas subyacentes de detección de cansancio que se utilizan con mayor frecuencia

- ECG Y EEG (Electrocardiograma y electroencefalograma)
- LBP (patrón binario local)
- Movimiento del volante
- Detección óptica

(Vandna Saini et al, 2014)

1.1.1 Machine learning

El aprendizaje automático, comúnmente conocido como Machine Learning, es un término que tiene gran importancia en la sociedad y que no es simplemente una moda pasajera. Este campo representa una puerta abierta hacia nuevas posibilidades tecnológicas, cuando hablamos de Machine learning, nos referimos a la ciencia que permite a las computadoras realizar predicciones basadas en datos. En forma más simple el machine learning se basa en la idea de que los procesadores pueden aprender de manera automática de la experiencia en vez de programar reglas o instrucciones explícitas, se utilizan algoritmos y modelos que permiten a las maquinas adaptarse a los datos y llevar a cabo tareas específicas sin necesidad de intervención humana, dichos algoritmos generalizan comportamientos a partir de ejemplos dados lo que permite tomar como entrada diversas variables y generar una única salida como respuesta. Un aspecto importante es que el aprendizaje automático debe ser capaz de reconocer situaciones problemáticas y tomar decisiones basadas en las estrategias aprendidas

Uno de los principales objetivos del aprendizaje automático es construir agentes que se adapten dinámicamente a nuevas circunstancias sin necesidad de un entrenamiento previo puesto que estos sistemas deben ser versátiles y capaces de funcionar con una extensa

variedad de datos de entrada incluyendo datos incompletos, inciertos, con ruido o inconsistencias, entre otros

El machine learning consta de Dos fases iniciales para un correcto reconocimiento, en la primera fase el sistema selecciona las características más relevantes de un objeto y en la segunda fase compara estas características obtenidas por unas conocidas mediante el proceso de emparejamiento de patrones esto conlleva a que cuando las diferencias sean significativas el sistema sea capaz de adaptarse. (Diferencia entre el aprendizaje automático y las redes neuronales. (s/f). La diferencia entre objetos y términos similares., 2018)

Topología de métodos

El algoritmo de aprendizaje automático desarrolla un modelo basado en un conjunto de datos de entrada, que representa lo que se conoce como un conjunto de entrenamiento.

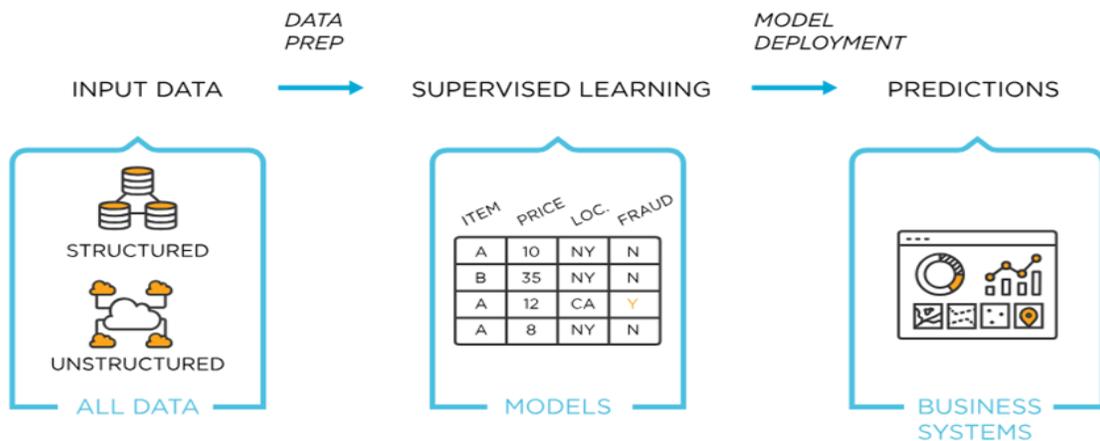
Durante esta etapa de aprendizaje, el algoritmo compara las salidas generadas por los modelos en construcción con las salidas ideales que estos modelos producen, a medida que se realiza esta comparación el algoritmo ajusta y mejora la precisión del modelo esta fase de comparación y ajuste crean la base del proceso de aprendizaje automático y puede llevarse a cabo de manera supervisada o no supervisada, dependiendo del enfoque del problema

Aprendizaje supervisado

El aprendizaje supervisado es una subcategoría de Machine learning y la inteligencia artificial la cual se basa en el uso de conjuntos de datos etiquetados para entrenar algoritmos, con el objetivo de clasificar datos o realizar predicciones precisas. En este enfoque existe un componente externo que compara los datos generados por el modelo con los datos esperados proporcionando un tipo de retroalimentación para ajustar el modelo de manera óptima.

Para lograr esto se requiere suministrar al modelo un conjunto de datos de entrenamiento que contenga tanto los datos de entrada como las salidas esperadas correspondientes, permitiendo que el algoritmo aprenda y se adapte a través de la iteración continua hasta alcanzar un rendimiento adecuado, como se observa en la (figura aprendizaje supervisado)

Figura 5 Aprendizaje supervisado



Fuente: imagen extraída del repositorio de la universidad católica (GONZALES, 2022)

[https://repository.ucatolica.edu.co/server/api/core/bitstreams/853a3b3d-bf5f-463f-948b-](https://repository.ucatolica.edu.co/server/api/core/bitstreams/853a3b3d-bf5f-463f-948b-702c372ccfa6/content)

[702c372ccfa6/content](https://repository.ucatolica.edu.co/server/api/core/bitstreams/853a3b3d-bf5f-463f-948b-702c372ccfa6/content)

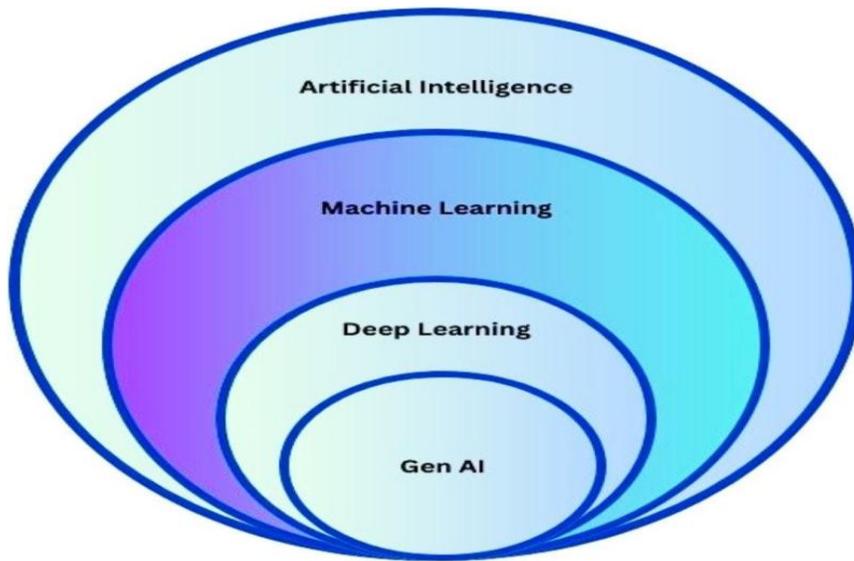
1.1.2 Deep learning

En la era actual, nos encontramos inmersos en un mundo donde los datos fluyen en grandes cantidades diversificadas, abarcando desde datos estructurados hasta los más complejos y no estructurados. Este contenido de información, aunque valiosa a menudo resulta desbordante dificultando su análisis y comprensión. Es en este contexto el Deep learning, una rama avanzada de la inteligencia artificial, emerge como una poderosa herramienta para descifrar y aprovechar todo el conocimiento contenido en estos datos, logrando obtener avances significativos en diversas áreas del conocimiento

El Deep learning es una rama dentro del campo del aprendizaje automático que se asemeja al funcionamiento del sistema neurológico humano. Se basa en una red neuronal jerárquica con múltiples capas de unidades de procesamiento, similar a como opera el cerebro humano. Cada capa está en la capacidad de identificar características específicas en los datos, como ejemplo los bordes de los objetos o la inclinación. Esta tecnología está inspirada en la forma en que las diferentes áreas del cerebro humano se especializan en tareas concretas

Al ser una subcategoría del Machine learning, estas redes intentan emular el comportamiento del cerebro humano, aunque a un nivel más limitado. A medida que se añaden más capas ocultas la red neuronal puede aprender a partir de una gran cantidad de datos y mejorar su precisión (IBM, 2023)

Figura 6 Subíndice inteligencia artificial



Fuente: (gupta, 2023)

1.1.3 Redes Neuronales

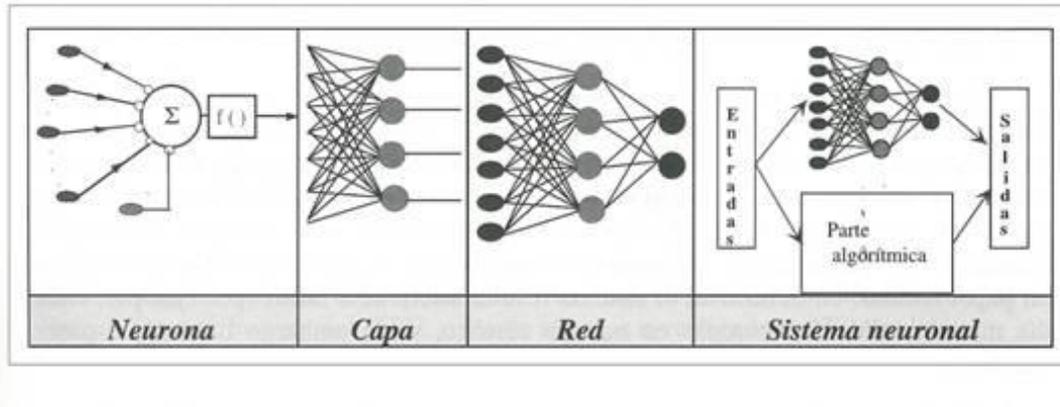
Las redes neuronales artificiales (RNA) han sido un campo de estudio inspirado en inteligencia artificial, ya que se basa principalmente en el funcionamiento del cerebro para solucionar problemas complejos. Para comprender como funcionan estas redes neuronales, es necesario analizar tres conceptos clave que una neurona artificial debe emular

- Procesamiento paralelo
- Memoria distribuida
- Adaptabilidad del entorno

Teniendo en cuenta estos conceptos que forman parte fundamental del aprendizaje autónomo podemos concluir que, para la realización de una red neuronal artificial, el mecanismo principal son las neuronas artificiales, dichas neuronas se organizan en múltiples capas, donde se combinan para formar una red neuronal. Por último, un conjunto de redes neuronales a su vez conectadas con las interfaces de entrada y salida y los

módulos adicionales convencionales necesarios, conformaran el sistema global de proceso (Moreno, 2019)

Figura 7 estructura jerárquica de un sistema basado en RNA



Fuente: (Patiño, 2011)

Procesamiento Paralelo: El procesamiento paralelo es un aspecto fundamental en las redes neuronales artificiales. En el cerebro humano miles de millones de neuronas trabajan simultáneamente para procesar información. En una Red neuronal artificial este concepto se traduce en la capacidad de procesar múltiples entradas de manera simultánea. Cada neurona artificial realiza operaciones en paralelo a sus entradas y transmite una señal de salida. Esta capacidad de procesamiento masivo es lo que permite a las redes neuronales abordar tareas complejas y manejar grandes volúmenes de datos de manera eficiente.

Memoria distribuida: La memoria distribuida se refiere a la capacidad de las RNA para almacenar información de manera distribuida en su estructura. A diferencia de los sistemas de memoria convencionales, donde se almacena información de manera localizada, las redes neuronales artificiales distribuyen la información en la red a través de conexiones.

sinápticas. Cada conexión sináptica tiene un peso que determina su fuerza y contribución a la red en su conjunto. Esta distribución de la memoria permite a las Redes Neuronales aprender y recordar patrones complejos en los datos de entrada

Adaptabilidad al entorno: La adaptabilidad al entorno es un aspecto esencial de las redes neuronales artificiales en el mundo real los entornos y las tareas pueden cambiar con el tiempo. Las redes neuronales deben ser capaces de ajustar sus conexiones sinápticas y pesos para adaptarse a nuevas circunstancias y aprender de la experiencia este proceso de adaptación se conoce como aprendizaje siendo un proceso importante de las redes neuronales

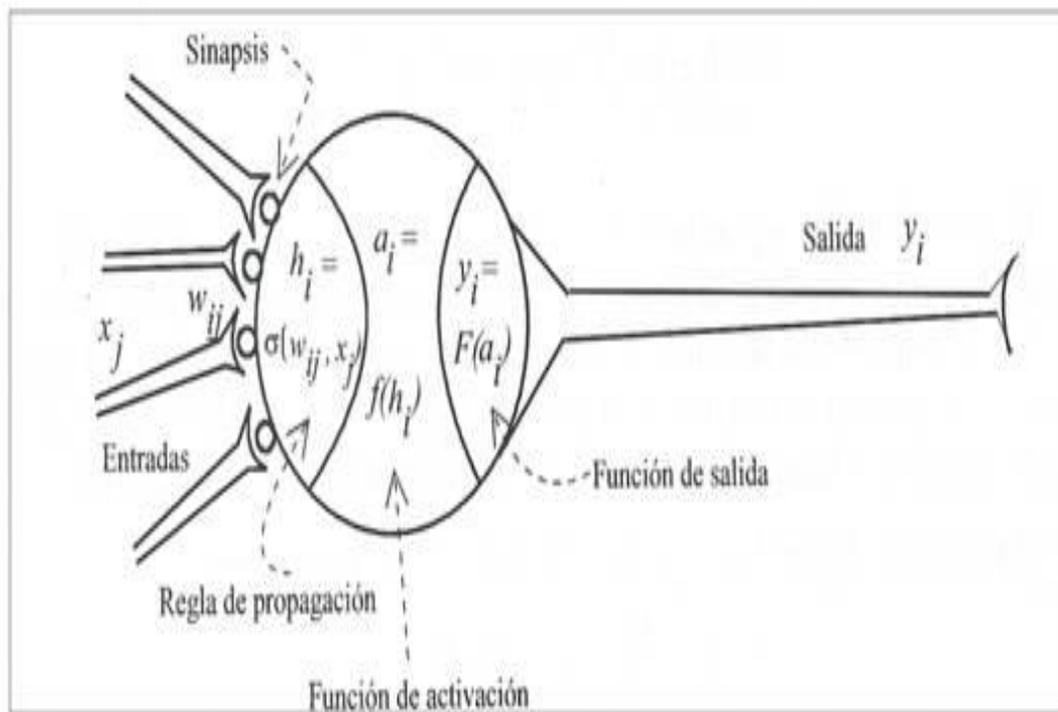
El proceso de aprendizaje es una RNA que implica la búsqueda de un grupo óptimo de pesos sinápticos que admita a la red hacer una tarea específica eficiente. El aprendizaje es un proceso interactivo en el que la red ajusta gradualmente estos pesos sinápticos en función de la diferencia entre las salidas reales y las salidas deseadas. El objetivo es minimizar esta diferencia y mejorar el rendimiento de la red

Existen dos niveles principales de entrenamiento de las redes neuronales

- **Modelado de las sinapsis:** En este nivel, el aprendizaje se centra en ajustar los pesos sinápticos que conectan las neuronas en la red. Los algoritmos de aprendizaje utilizan métodos numéricos iterativos para minimizar una función de costo. Esta función de coste mide la diferencia entre las salidas reales y las salidas deseadas. Los algoritmos buscan modificar los pesos sinápticos de manera que esta diferencia sea lo más pequeña posible este proceso de modelado de sinapsis es esencial para que la red pueda aprender a realizar tareas específicas y reconocer patrones en los datos

Creación o Destrucción de neuronas: En ocasiones el aprendizaje también puede interferir en la creación o destrucción de neuronas en la red. Esto permite que las redes neuronales modifiquen su arquitectura y se adapten mejor a ciertas tareas específicas, Al agregar o eliminar neuronas. La red puede ajustar su capacidad y complejidad para abordar diferentes problemas. Este nivel de aprendizaje es particularmente importante cuando se enfrenta a ciertos cambios significativos en su entorno o directamente a las tareas que debe realizar la red neuronal (Moreno, 2019)

Figura 8 Modelo genérico de una neurona artificial



Fuente: (CONCEPTOS BASICOS SOBRE REDES NEURONALES , s.f.)

1.2 Tecnologías usadas

1.2.1 Jetson TX2

La NVIDIA Jetson Tx2 es una potente placa de desarrollo destinada a aplicaciones de inteligencia artificial y visión por computadora equipada con una GPU NVIDIA pascal que proporciona un rendimiento de inteligencia artificial de alto nivel, está diseñada específicamente para ejecutar aplicaciones de inteligencia artificial y es capaz de ejecutar modelos de aprendizaje profundo complejos de forma eficiente. Además, cuenta con una amplia gama de puertos y conectividad para adaptarse a diversas aplicaciones y proyectos

1.2.2 Hardware

La implementación de este proyecto se ha llevado a cabo utilizando tecnología proporcionada por la universidad Antonio Nariño. Lo que permite que los modelos de machine Learning utilizados funcionen de manera eficiente y puedan ser entrenados de manera efectiva.

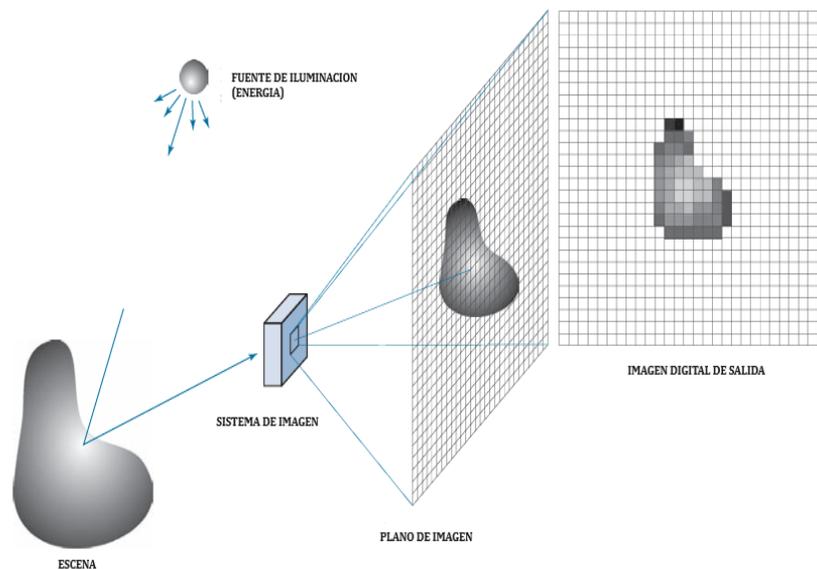
1.2.3 Software

En el desarrollo de este proyecto, se empleó el software preinstalado en la Jetson, el cual se basa en el sistema operativo UBUNTU 18.04. Para la implementación de las funcionalidades requeridas se usó el lenguaje de programación Python junto con distintas bibliotecas en las cuales destacan OPEN CV y Tensor Flow. Estas herramientas permitieron el procesamiento de imágenes y la creación de modelos de aprendizaje automático para alcanzar los objetivos del proyecto de manera eficaz. A su vez como entorno de desarrollo y visualización se empleó Spyder una interfaz de programación que facilitó la programación y depuración del código.

1.3 Imágenes

Imagen Digital: Una imagen se define como una representación digital o función bidimensional $f(x,y)$ donde x e y son representadas como coordenadas espaciales(plano), y la amplitud de f en cualquier par de coordenadas (x,y) esto se denomina como intensidad o nivel de gris de la imagen en ese punto. Una imagen digital se conforma cuando (x,y) y los valores de intensidad de f son todos cantidades finitas y discretas, esto se refiere al procesamiento de imágenes digitales mediante una computadora digital. La imagen digital está compuesta por un numero finitos de elementos en donde cada uno tiene una ubicación propia y un valor específico, dichos elementos se conocen por el termino de “PIXEL” (gonzalez, 2018)

Figura 9 Adquisición de imagen Digital



Fuente: (Flores D. W., s.f.)

1.3.1 Procesamiento Digital de Imágenes

El procesamiento digital de imágenes (PDI) se refiere al procesamiento computarizado de imágenes. Existen diferencias importantes entre el procesamiento de imágenes y la visión en el sentido biológico. Ciertas imágenes pueden derivarse de la captura de cualquier parte del espectro electromagnético. Esto significa que los sistemas de PDI pueden procesar imágenes generadas a partir de rayos X, rayos gamma, resonancia magnética, microondas y otras fuentes. (cuevas, 2010)

Esencialmente, la materia prima del PDI y la visión es la imagen, que se considera una representación del mundo físico y contiene información valiosa capturada a través del muestreo. Mientras que la visión humana está restringida al espectro visible, las máquinas pueden percibir y procesar imágenes en un rango mucho más amplio, lo que les permite capturar y analizar información de diversas fuentes.

Las líneas entre el campo del Procesamiento Digital de Imágenes y otros campos como el Análisis de Imágenes y la Visión por Computadora pueden estar conectadas entre sí, la Visión por Computadora intenta utilizar computadoras para simular las capacidades de percepción visual humana. Esto incluye el aprendizaje, la inferencia y la toma de decisiones basadas en entradas visuales, y se encuentra claramente dentro del ámbito de la Inteligencia Artificial. Podemos considerar tres tipos de procesos que comienzan con el PDI y desembocan en la Visión por Computadora:

Procesos de Bajo Nivel: Estos procesos implican operaciones como la reducción de ruido, mejora de contraste y filtros de enfoque en imágenes. Se caracterizan porque tanto las entradas como las salidas son imágenes.

Procesos de Nivel Medio: Aquí se realizan operaciones como la segmentación y clasificación de objetos individuales en imágenes. Aunque las entradas suelen ser imágenes, las salidas consisten en atributos extraídos de esas imágenes, como contornos, bordes o la identidad de objetos individuales.

Procesos de Alto Nivel: Estos procesos implican la obtención de significado a partir de un conjunto de objetos reconocidos, es decir, análisis de imágenes. Finalmente, se realizan las funciones cognitivas asociadas con la visión, como el reconocimiento de rastros faciales (VILET, 2005)

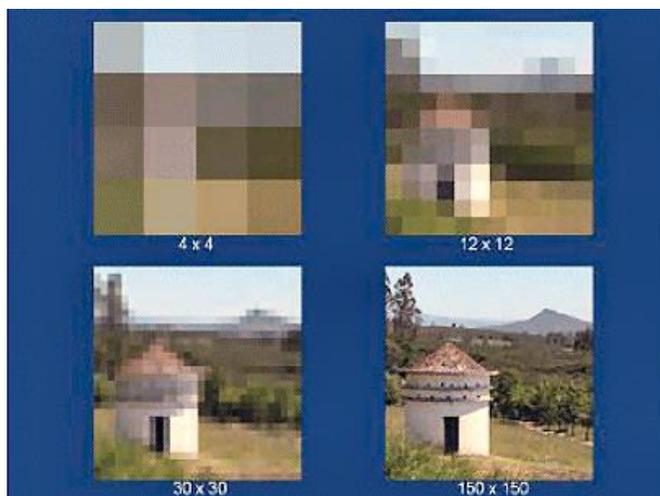
Clasificación de imágenes

La clasificación de imágenes es un proceso esencial en el campo del procesamiento de imágenes y la visión por computadora. Su principal objetivo es asignar cada píxel de una imagen a una categoría o tema específico. Este proceso de clasificación digital conlleva la creación de un mapa temático detallado que proporciona información valiosa. El cual permite determinar el número de píxeles y, por lo tanto, calcular la superficie correspondiente a cada categoría. A través de esta clasificación, una imagen multibanda se transforma en otra imagen de igual tamaño y características que la original, con la diferencia significativa de que el valor del ND (Número Digital) de cada píxel ya no está relacionado con la radiación detectada por el sensor, sino que actúa como una etiqueta que identifica la categoría asignada a ese píxel.

La clasificación de imágenes se favorece en gran medida de procesos de corrección que mejoran la exactitud y calidad de los resultados. No obstante, es importante destacar que es posible realizar una clasificación basada exclusivamente en los valores de ND de la imagen original. Esto se debe a que las categorías temáticas suelen definirse en función de las condiciones específicas de la escena que se está clasificando. Bajo este enfoque, no es necesario conocer a fondo las condiciones de adquisición, ya que el objetivo principal es identificar y discriminar las clases dentro de la imagen. No se busca que esta identificación sea extrapolable a otras situaciones, sino que sea relevante para la tarea actual.

Al abordar la clasificación de imágenes en términos de clases digitales, es esencial diferenciar entre las clases de formación y las clases espectrales. Las clases de formación son las categorías de interés que el analista o investigador intenta identificar en la imagen. Por otro lado, las clases espectrales se refieren a grupos de píxeles con características uniformes en términos de brillo en las diferentes bandas espectrales. (Mihaich, 2014)

Figura 10 píxeles número digital



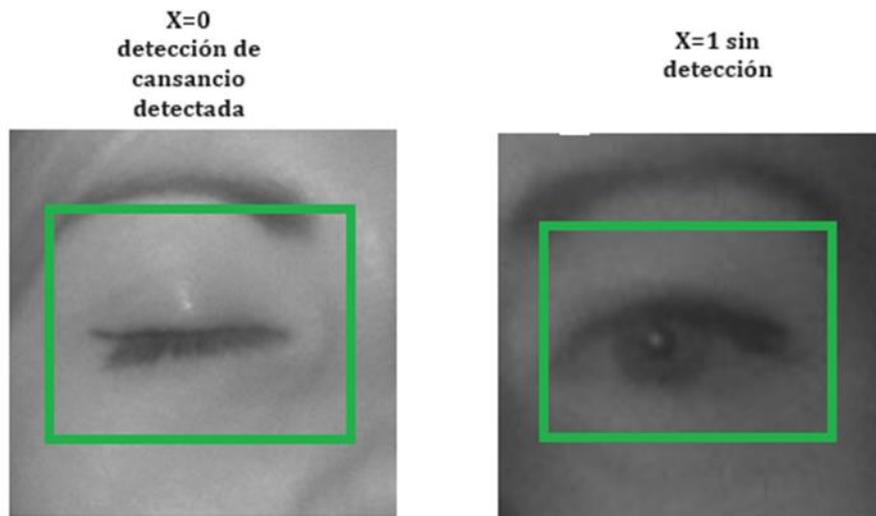
Fuente: (Berlanga)

1.3.2 Etiquetas

Una etiqueta en el procesamiento de imágenes y el aprendizaje automático es una representación o categorización que se asigna a un elemento o conjunto de Datos para identificar su pertenencia a una categoría específica, se utilizan mayormente para organizar y clasificar información de manera que facilite el análisis de imágenes

En la figura observamos un ejemplo de categorización de imágenes en donde muestra el estado del ojo junto con su etiqueta que nos indica a que categoría pertenece

Figura 11 Etiquetas detección de cansancio



Fuente: Autoría propia simulación sistema detección de cansancio

1.3.3 Etapas clave en la clasificación de imágenes

Tabla 1: Etapas clasificación de imágenes

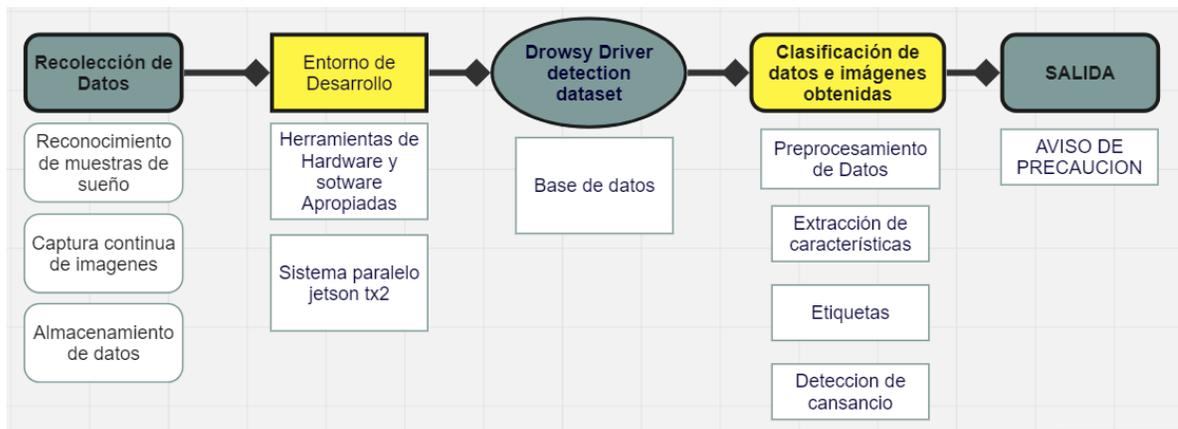
ETAPA	DESCRIPCION
Recolección de imágenes	Este proceso inicia con la adquisición de imágenes que se pueden adquirir de diversas fuentes como cámaras sensores etc.
Preprocesamiento	Es común realizar operaciones de preprocesamiento para mejorar la calidad de las imágenes
Extracción de características	En esta etapa se extraen características significativas de las imágenes, pueden ser tipos como color textura forma y estructura
Construcción del conjunto de datos	Se recompilan y etiquetan un conjunto de imágenes para entrenar y probar el modelo de clasificación
Entrenamiento del modelo	Se utiliza un algoritmo de aprendizaje automático o red neuronal para entrenar un modelo de clasificación
Validación y ajuste	Después del entrenamiento el modelo se valida utilizando el conjunto de prueba para evaluar su rendimiento. Se pueden realizar ajustes en el modelo
Clasificación de imágenes	Al tener el modelo entrenado se utiliza para clasificar nuevas imágenes. El modelo asigna una categoría a cada imagen
Evaluación de rendimiento	Se evalúa el rendimiento del modelo en función de métricas como la precisión, la sensibilidad y la especificidad. Esto sirve para determinar que tan bien se está clasificando
Optimización	Se puede realizar mejoras en el modelo, como la optimización de parámetros o la selección de características más relevantes para lograr un rendimiento optimo

2. CAPITULO 2

2.1 Metodología

Este trabajo se centra en la implementación de técnicas de machine learning, Deep learning y redes neuronales para detectar el cansancio en conductores. Para lograr un sistema preciso y robusto , se utilizó un sistema paralelo integrado en una Jetson TX2. Estos datos incluirán imágenes que se procesarán y mejorarán con el fin de identificar los síntomas de fatiga y proporcionar una alerta cuando se detecte que el conductor presenta síntomas de sueño.

Figura 12 Etapas metodología



Fuente: Mapa conceptual etapas de metodología realizado en <https://www.lucidchart.com/pages/es/ejemplos/diagrama-de-flujo-online>

En primer lugar, es importante la clasificación de la información que se encuentra contenida dentro de las imágenes extraídas de los conductores, lo que inicialmente se estableció como la etiqueta proporcionada a las notaciones de aprendizaje supervisado. Esto es necesario para convertir las variables categóricas en variables numéricas, así evitando inconvenientes entre los distintos aspectos del modelo predictivo. A

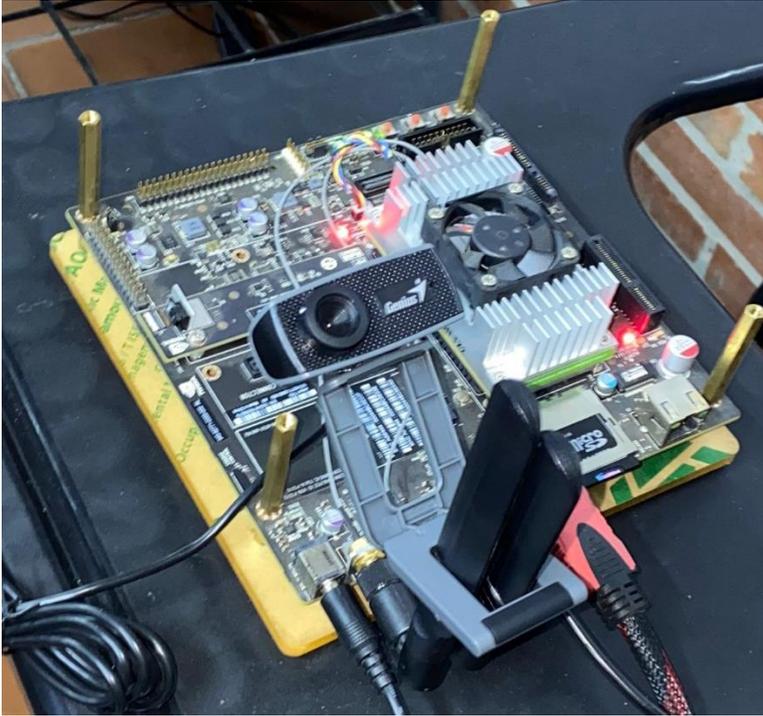
continuación, se mostrará y explicará el proceso realizado para cada etapa establecida del proyecto.

2.1.1 Reconocimiento de muestras de sueño y cansancio en conductores

El primer paso para lograr nuestro objetivo principal es recopilar muestras de datos visuales en conductores tanto en estados de sueño como de alerta durante la conducción. Estas muestras de datos las usaremos más adelante para entrenar y validar un modelo de detección de muestras de sueño y cansancio en conductores.

Captura continua de imágenes: Durante el proceso de activación del sistema piloto simulando el proceso de conducción, se utilizó una cámara conectada directamente a la Jetson TX2 para obtener imágenes destinadas al reconocimiento de cansancio. Esta cámara funciono de manera continua, capturando imágenes constantes para distinguir los indicadores visuales relevantes. Es importante destacar que esta etapa está conectada con el entrenamiento previamente programado basado en drowsy driver detection.

Figura 13 Jetson TX2



Indicadores de somnolencia: Durante el proceso de captura de imágenes, se prestará atención a indicadores visuales clave en la somnolencia, como:

- La caída de los párpados: Se monitoreará si el conductor muestra tendencias a mantener los ojos cerrados durante periodos prolongados o si experimenta parpadeos lentos y pesados
- Otros indicadores visuales: También se presta atención a cualquier otro comportamiento visual anormal que sugiera somnolencia, como movimientos inusuales de la cabeza. Estos signos adicionales nos ofrecen información adicional a nuestro algoritmo sobre el estado de alerta del conductor

Figura 14 Comportamiento físico presencia de fatiga



Fuente: (Real-time Driver Drowsiness Detection for Embedded System)

Sincronización de datos: Al momento de capturar imágenes de manera continua, se registra la ubicación precisa de cada imagen. Este registro de ubicación es fundamental para análisis y evaluaciones futuras. Permitiendo una correlación entre los indicadores visuales y patrones relevantes para el estudio del cansancio en conductores

Almacenamiento de datos: Las imágenes capturadas se almacenarán en un sistema de almacenamiento seguro en una estructura de datos. Esto asegura que las imágenes estén disponibles en el sistema piloto para su procesamiento y análisis. Las imágenes se estructurarán de manera que sea fácil para acceder a su información y uso en el entrenamiento y validación de modelos de machine learning,

- **Etiquetado de datos:** Cada imagen capturada se somete a un proceso de etiquetado automático usando algoritmos de procesamientos de imágenes, se analizaron indicadores de somnolencia anteriormente mencionados como el

pestañeo frecuente y movimiento de cabeza, Estos algoritmos se le asignan etiquetas que indican si el conductor se encuentra en un estado de somnolencia o alerta en el momento de la captura de la imagen. Este etiquetado es de suma importancia para el aprendizaje de las futuras redes neuronales y modelos de machine learning usados en el sistema piloto de detección de cansancio

- **Revisión de datos:** La implementación de esta revisión en el sistema piloto es crucial ya que asegura que los datos usados para detectar el cansancio del conductor sean confiables y precisos. Estos datos forman parte importante en el sistema de detección de somnolencia lo que a su vez contribuyen a la seguridad y efectividad del sistema en situaciones de conducción en tiempo real
- **Preparación de datos:** Los datos se organizarán en un formato adecuado para su posterior uso en el entrenamiento y validación de modelos de machine Learning.

Consideraciones adicionales:

- **Privacidad:** Se deben tomar medidas para proteger la privacidad de los conductores, como la desidentificación de datos o la obtención de consentimiento informado.
- **Calidad de imagen:** Las cámaras deben tener una resolución y calidad suficiente para capturar los indicadores de somnolencia de manera efectiva.

2.1.2 Investigación y elección de herramientas de software y hardware

En esta fase del proyecto de detección de cansancio en conductores, es importante investigar, identificar y seleccionar las herramientas de hardware y software adecuadas.

Esto conlleva a la elección de bibliotecas de software específicas y determinación del hardware necesario para el procesamiento en tiempo real

Durante esta etapa, se busca y evalúa las herramientas disponibles que mejor se adapten a las necesidades del sistema piloto. Esto incluye software especializado en machine learning y procesamiento de imágenes y un hardware de alto rendimiento que sea capaz de procesar datos en tiempo real

Investigación de bibliotecas de software: Se llevo a cabo una investigación a fondo para identificar las mejores bibliotecas de software que se adapten a las necesidades del sistema. OpenCV se seleccionará para el procesamiento de imágenes debido a su amplia gama de funciones y compatibilidad con la Jetson TX2. TensorFlow se elegirá para el entrenamiento de modelos de machine learning debido a su versatilidad y soporte comunitario. Otra de las herramientas de software que se utilizaran en el sistema es la base de datos Drowsy data detection dataset donde se almacenará el banco de imágenes para el entrenamiento de nuestro modelo.

Investigación de hardware: Se investigarán las opciones de hardware disponibles para el procesamiento en tiempo real de las imágenes capturadas y la ejecución del modelo. La Jetson TX2 se elegirá por su capacidad de procesamiento de alto rendimiento y su capacidad para ejecutar aplicaciones de inteligencia artificial en dispositivos embebidos.

- **Compatibilidad y optimización:** Se verificará la compatibilidad de las bibliotecas de software seleccionadas (OpenCV y TensorFlow) con la Jetson TX2. Se implementarán optimizaciones para garantizar que el hardware aproveche al máximo las capacidades de procesamiento de la Jetson TX2.

- **Configuración del entorno de desarrollo:** Se configurará el entorno de desarrollo en la Jetson TX2 para que sea compatible con las bibliotecas de software seleccionadas. Esto incluirá la instalación de las bibliotecas y la configuración de cualquier dependencia necesaria.
- **Pruebas de rendimiento:** Se realizaron pruebas de rendimiento en la Jetson TX2 para garantizar que el hardware y el software funcionen de manera eficiente y cumplan con los requisitos de procesamiento en tiempo real.

Identificación de las ventajas de la Jetson TX2: En esta etapa, se identificó las ventajas del uso de la Jetson TX2 como plataforma de hardware para el proyecto de detección de somnolencia en conductores.

Capacidad de procesamiento en tiempo real: La Jetson TX2 es una plataforma de alto rendimiento que es capaz de detectar y procesar imágenes en tiempo real. Esto es esencial para la detección oportuna de fatiga en conductores, ya que permite analizar continuamente las imágenes almacenadas y tomar decisiones rápidas en función de los signos de somnolencia detectados.

Eficiencia energética: La Jetson TX2 es conocida por su eficiencia energética. Esto es muy importante para aplicaciones en vehículos, ya que permite que el sistema funcione de manera continua sin agotar tan rápido la energía de la batería del vehículo. La eficiencia energética también contribuye a una operación más sostenible y segura.

Soporte para aceleración de hardware: La Jetson TX2 cuenta con unidades de procesamiento de gráficos (GPU) y unidades de procesamiento de visión (NPU) que podemos aprovechar para acelerar cargas de trabajo de machine learning. Esto con el fin de mejorar significativamente el rendimiento de las tareas de detección de somnolencia en

conductores y nos permite una respuesta más rápida a los indicadores de fatiga y cansancio en tiempo real.

Análisis detallado:

Se llevo a cabo un análisis detallado de las ventajas identificadas, centrándose en cómo cada una de ellas contribuirá al éxito del proyecto. Por ejemplo, se analizó cómo la capacidad de procesamiento en tiempo real permitirá una detección rápida y precisa de la somnolencia, mientras que la eficiencia energética garantizará una operación continua y sostenible en vehículos.

Integración de ventajas en la implementación: Se planifico la forma en que estas ventajas se integrarán en la implementación del sistema piloto. Se desarrollo algoritmos y modelos de machine learning que aprovechan la capacidad de procesamiento en tiempo real y la aceleración de hardware de la Jetson TX2.

Pruebas y validación: Se llevo a cabo pruebas en el sistema piloto para verificar que las ventajas identificadas se traduzcan en un rendimiento óptimo en condiciones reales. Esto incluye pruebas de detección de somnolencia en situaciones de conducción simuladas.

2.1.3 Instalación del sistema y herramientas en la Jetson TX2

En este proceso se da los pasos necesarios para la instalación del sistema operativo en la Jetson TX2 el cual fue un pilar importante en la creación del sistema piloto:

Como primer paso se Descarga la última versión de SDK Manager de Nvidia en una PC con Ubuntu 18.04. Mientras se descarga, ponga el TX2 en modo de recuperación siguiendo estos pasos:

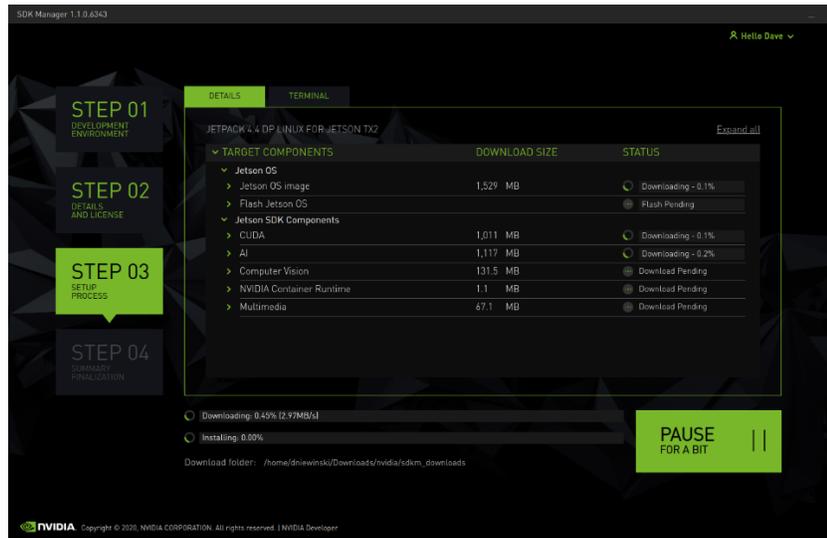
- Conecte el TX2 a su PC usando el cable micro USB provisto.
- Asegúrese de que el TX2 esté apagado
- necesitará usar un concentrador USB pequeño, ya que el Jetson TX2 solo tiene un puerto USB).
- Mantenga presionado el botón REC
- Presiona el botón de poder.
- Instale SDK Manager ejecutando los siguientes comandos:

```
cd <folder where you downloaded SDK manager>
```

```
sudo dpkg -i sdkmanager_<version>_amd64.deb
```

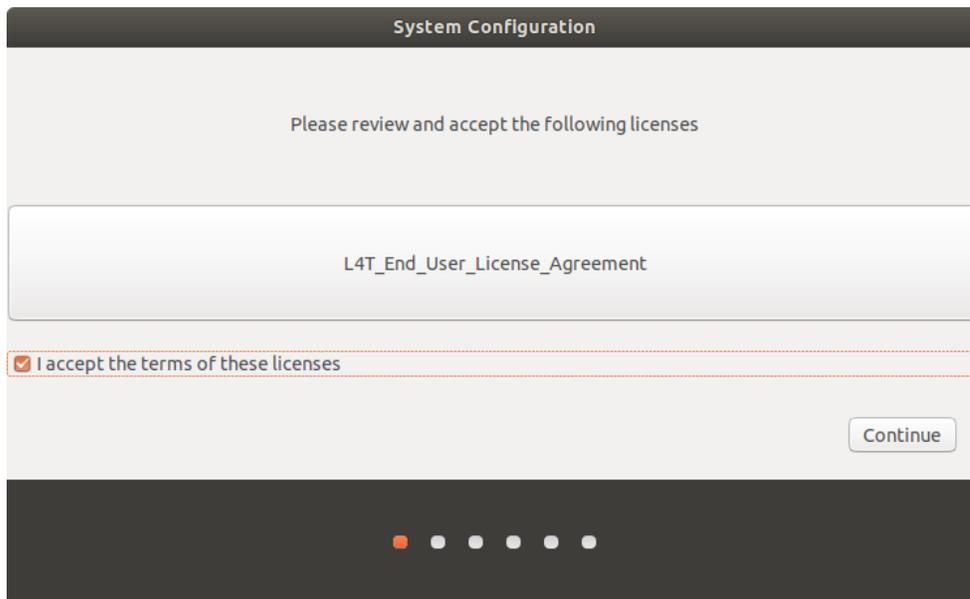
Instalación imagen Jetson TX2

Figura 15 Instalación imagen Jetson TX2



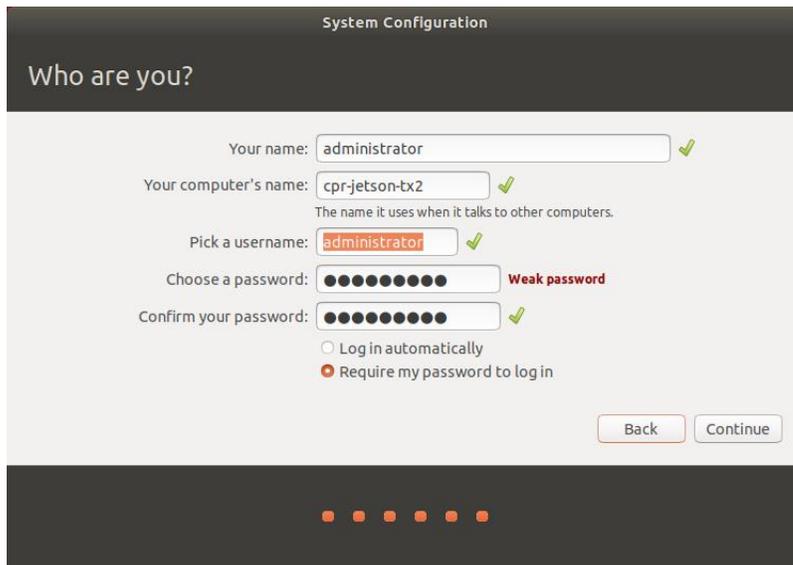
Durante la instalación, asegúrese de conectar un teclado y un monitor al Jetson. En el primer arranque, seguirá los pasos habituales de configuración de Ubuntu. Se Aceptan las licencias

Figura 16 System configuration-Licencias



Elija un nombre de host, nombre de usuario y contraseña para la máquina.

Figura 17 Usuario Jetson tx2

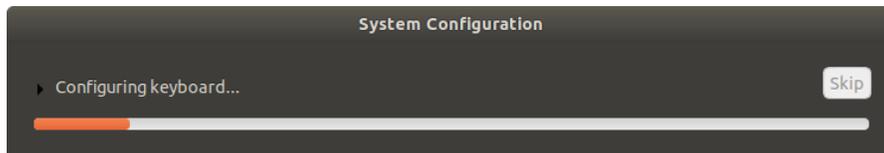


The image shows a 'System Configuration' window with the title 'Who are you?'. It contains several input fields and options:

- 'Your name:' with the value 'administrator' and a green checkmark.
- 'Your computer's name:' with the value 'cpr-jetson-tx2' and a green checkmark. Below it is the text 'The name it uses when it talks to other computers.'
- 'Pick a username:' with the value 'administrator' and a green checkmark.
- 'Choose a password:' with a masked password field and a red warning 'Weak password'.
- 'Confirm your password:' with a masked password field and a green checkmark.
- Two radio buttons: 'Log in automatically' (unselected) and 'Require my password to log in' (selected).
- 'Back' and 'Continue' buttons at the bottom right.
- A progress indicator at the bottom consisting of five orange dots.

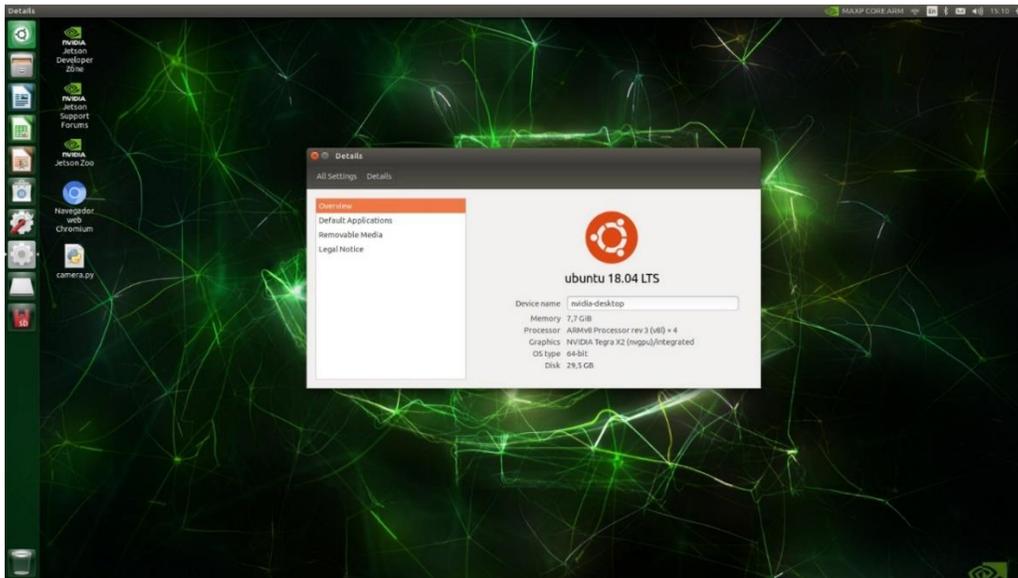
Se Completará la instalación e instalarán los paquetes estándar restantes.

Figura 18 Instalación de sistema operativo



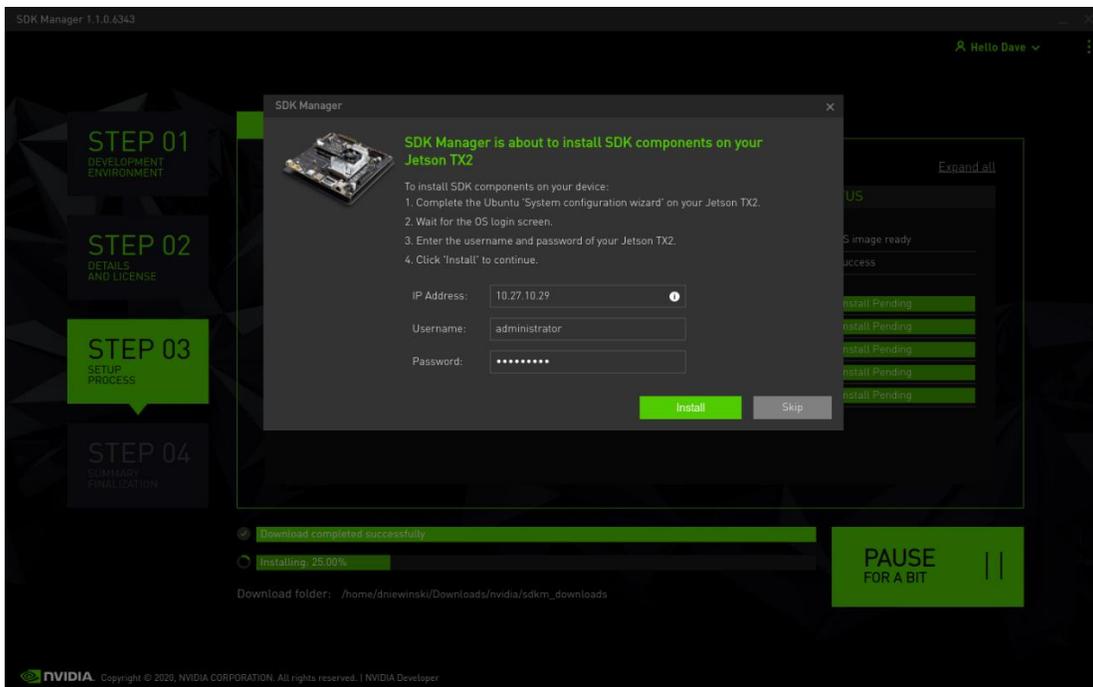
Una vez que el sistema operativo esté configurado, se dará acceso al escritorio.

Figura 19 Interfaz sistema operativo Jetson



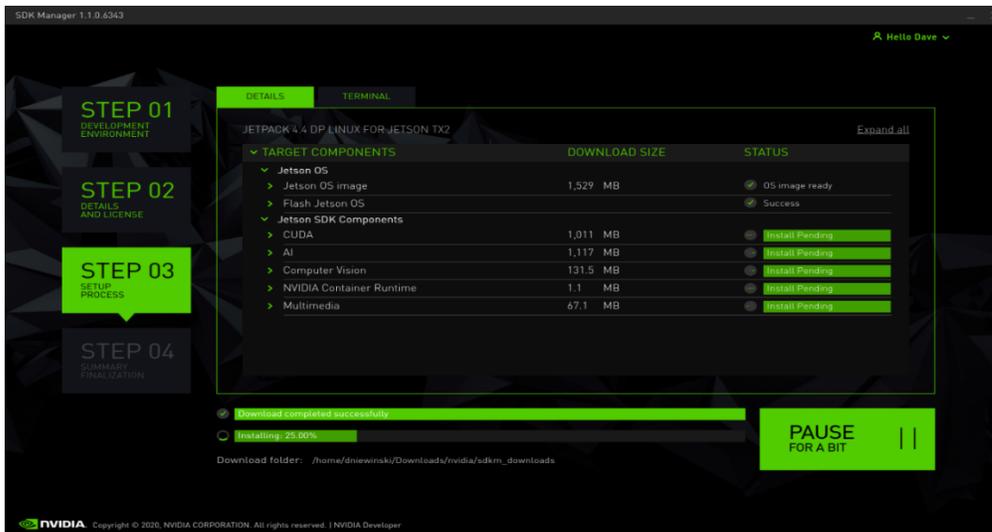
De vuelta en su máquina host, estará esperando para instalar los componentes SDK adicionales en su Jetson. Ingrese el nombre de usuario, la contraseña y la dirección IP que encontró arriba.

Figura 20 Instalación componentes SDK



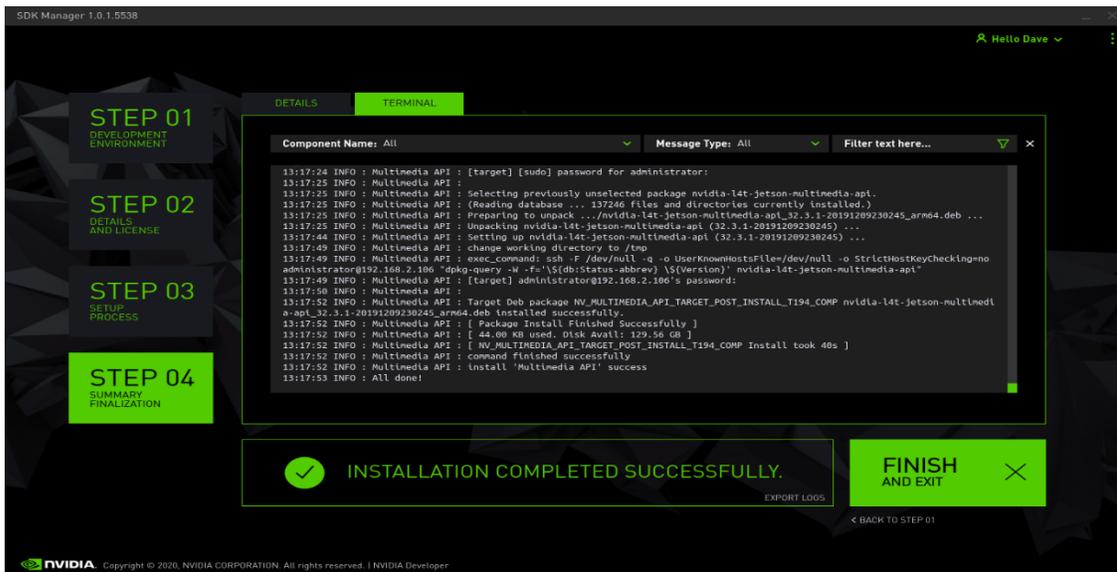
La instalación se conectará al Jetson remoto a través de la red. Continuará la instalación transfiriendo los archivos e instalándolos.

Figura 21 Transferencia de archivos Jetson TX2



Una vez finalizado el proceso, puede hacer clic en TERMINAR para cerrar la ventana.

Figura 22 Proceso final instalación Jetson TX2



En conclusión, se siguieron todos los pasos cuidadosamente proporcionados en la página oficial de Nvidia para la instalación de la Jetson TX2. Algunos pasos, tanto importantes

como menos importantes se omitieron en esta descripción detallada con el propósito de mantener el texto enfocado en los aspectos esenciales de la instalación. La implementación de estos pasos adicionales puede ser necesaria según los requisitos específicos del sistema piloto

2.2 Base de datos Drowsy Driver Detection Dataset

El “Drowsy Driver Detection Dataset” contiene una colección de datos variada en donde incluye imágenes faciales de conductores en distintos estados de somnolencia y alerta. También registra datos como la frecuencia de parpadeo y otros datos como bostezo y movimiento de cabeza. Estos datos fueron recompilados en diversas condiciones y entornos de conducciones. Para este proyecto se accederá a la base de datos antes mencionada y se descargarán los conjuntos de datos necesarios. Dichos datos se utilizarán en el proceso de entrenamiento de algoritmos de machine learning y en la configuración del sistema piloto.

Con el fin de validar el proceso de entrenamiento los datos se dividirán en dos conjuntos denominados “entrenamiento y prueba”. El conjunto de entrenamiento se utilizará para el entrenamiento de modelos de machine learning, mientras que el conjunto de prueba se utilizará para medir y evaluar la precisión y rendimiento del sistema.

Es de suma importancia realizar un preprocesamiento de los datos con el propósito de que el sistema piloto se efectuó correctamente

El conjunto de datos de detección de somnolencia se genera utilizando el conjunto de datos MRL y Closed Eyes in Wild (CEW), así como nuestro propio conjunto de datos único. Este conjunto de datos a gran escala que comprende imágenes de ojos humanos

abiertos y cerrados, puede usarse principalmente para la detección de ojos y abarcar aún más para la detección de somnolencia.

Las imágenes del conjunto de datos se tomaron en diversas condiciones, incluidas diversas condiciones de iluminación, distancia, resolución, ángulo de la cara y ángulo de los ojos. Estos parámetros ayudan a obtener buenos resultados con posibilidades mínimas de obtener una precisión baja.

Hay varias versiones de este conjunto de datos, en donde la versión 1 tiene 10.000 imágenes divididas en 5000 imágenes para ojos cerrados y 5000 imágenes para ojos abiertos. Las imágenes de 5k en la Versión 2 se dividen en 2500 imágenes para ojos cerrados y 2500 imágenes para ojos abiertos. En la Versión 3 se puede encontrar otra colección de 10.000 imágenes, divididas en 5.000 imágenes para cada uno de los ojos cerrados y abiertos . 2000 imágenes de ojos cerrados y 2000 imágenes de ojos abiertos conforman la cuarta versión del conjunto de datos, es decir, la versión 4 con imágenes de 4k en total.

A continuación, se referencia la página o sitio web donde se puede encontrar las imágenes para entrenar el modelo de Deep learning

<https://www.kaggle.com/datasets/prasadvpatil/mrl-dataset>

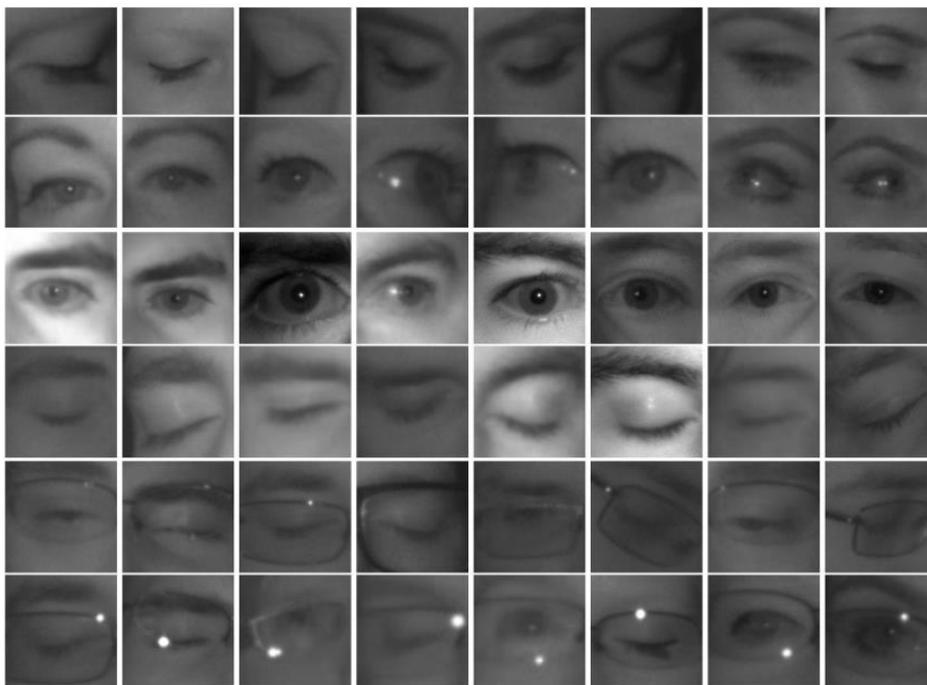
2.2.1 Procesamiento y análisis de datos

En esta etapa, las imágenes obtenidas de los conductores en diversos estados ya sea de somnolencia o de alerta, se procesarán y analizarán con el propósito de extraer caracteres importantes y evaluar el estado de alerta del conductor mediante algoritmos especializados en procesamiento de imágenes y análisis del comportamiento del conductor

Los datos de nuestro modelo son obtenidos del Drowsy Driver Detection Dataset, una base de datos con gran amplitud de datos que capturan a los conductores en situaciones de somnolencia.

El procesamiento y análisis de datos es crucial para convertir dichas imágenes en información que nos servirá para entrenar nuestro modelo de detección de cansancio.

Figura 23 dataset Ojo humano



Para la construcción del conjunto de datos utilizado en el proceso de entrenamiento y prueba se definieron 3 clases que nos serán de gran utilidad para la clasificación de imágenes

Tabla 2 Clasificación del proceso de entrenamiento

CLASE	DEFINICION
BAJO	<ul style="list-style-type: none">• Escasa posibilidad de dormirse
MEDIO	<ul style="list-style-type: none">• Mediana posibilidad de dormirse
ALTO	<ul style="list-style-type: none">• Elevada posibilidad de dormirse

Estas imágenes se encuentran ubicadas dentro de la base de datos MRL EYE DATASET.

Es importante destacar que las imágenes en esta base de datos presentan distintas longitudes, ya que no todas están representadas con las mismas dimensiones refiriéndonos a términos de píxeles. Cada imagen varía en tamaño, lo que añade una capa adicional de complejidad al proceso de análisis y clasificación

2.2.2 Clasificación y gestión de imágenes en carpetas

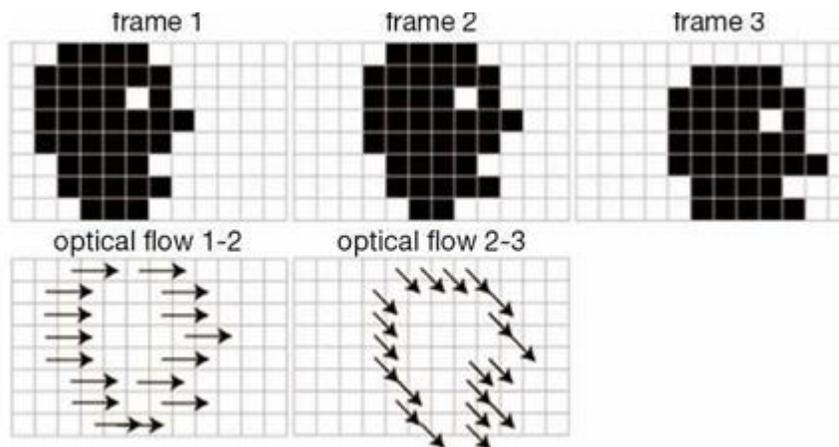
Para entrenar y evaluar nuestro sistema de detección de cansancio en conductores, es fundamental organizar las imágenes en conjuntos de datos adecuados. El ajuste de las imágenes es una etapa crucial en el proceso de preprocesamiento. Una vez que se ha seleccionado el conjunto de datos y se han organizado los subconjuntos de validación, prueba y entreno, se empieza a realizar modificaciones en las imágenes para iniciar el análisis y clasificación.

Cada imagen del conjunto de datos se nombra en función de propiedades y la etiqueta se deriva del nombre de la carpeta en la que se encuentra. Para las técnicas de clasificación de imágenes se inicia cargando todas las imágenes de entrenamiento para después aplicar los siguientes pasos

1. Conversión del espacio de color a RGB: las imágenes se convierten al espacio de color RGB ya que open CV lee las imágenes en formato BGR por defecto (TRABAJANDO CON CANALES DE COLOR EN IMÁGENES DIGITALES, CON Python Y ‘opencv’. , 2023)
2. Conversión a escala de grises: Las imágenes se transforman a escala de grises (1 canal) esto para simplificar el algoritmo y reducir los requisitos de la imagen
3. Cambio de tamaño de imagen
4. Vectorización de la imagen: Cada imagen se aplana y se convierte en una matriz numérica (un arreglo unidimensional o vector).

Es importante destacar que el último punto relacionado con la vectorización de la imagen no se aplica el preprocesamiento de la red neuronal, puesto que este tipo de red espera matrices o arreglos multidimensionales como entrada

Figura 24 Flujo óptico



Fuente : (Desarrollo de un Sistema de Detección de Movimiento basado en Flujo Óptico en Raspberry Pi)

Estructura de carpetas:

La estructura de carpetas, se utiliza para organizar el conjunto de datos destinados a entrenar, probar y validar en el modelo de Machine Learning, en este caso, específicamente para la detección de fatiga en conductores. Se menciona a continuación como están conformadas las estructuras dentro de las carpeta

Dataset: Esta es la carpeta principal la cual contiene todo el conjunto de datos. Aquí se almacenan todos los datos.

Entrenamiento: Esta carpeta se utiliza para el conjunto de datos de entrenamiento. Estos datos se utilizan para enseñar al modelo de Machine Learning a reconocer patrones de fatiga y alerta.

Ejemplo:

- **Alerta: Imágenes de conductores sin gafas en estado de alerta.**
- alerta_1.jpg: Ejemplo de conductor sin gafas en estado de alerta.
- alerta_2.jpg: Otro ejemplo de conductor sin gafas en estado de alerta.
- **Fatiga: Imágenes de conductores sin gafas con signos de sueño y cansancio.**
- fatiga_1.jpg: Ejemplo de conductor sin gafas con fatiga.
- fatiga_2.jpg: Otro ejemplo de conductor sin gafas con fatiga.
- **Alerta Gafas: Imágenes de conductores con gafas en estado de alerta.**
- alerta_gafas_1.jpg: Ejemplo de conductor con gafas en estado de alerta.
- alerta_gafas_2.jpg: Otro ejemplo de conductor con gafas en estado de alerta.
- **Fatiga_Gafas: Imágenes de conductores con gafas con signos de sueño y cansancio.**
- fatiga_gafas_1.jpg: Ejemplo de conductor con gafas con fatiga.
- fatiga_gafas_2.jpg: Otro ejemplo de conductor con gafas con fatiga.

Conjunto de datos de prueba para evaluar el rendimiento del modelo después del entrenamiento.

- Alerta: Imágenes adicionales de conductores sin gafas en estado de alerta para pruebas.

- alerta_101.jpg: Ejemplo de conductor sin gafas en estado de alerta para pruebas.

- alerta_102.jpg: Otro ejemplo de conductor sin gafas en estado de alerta para pruebas.

- Fatiga: Imágenes adicionales de conductores sin gafas con signos de sueño y cansancio para pruebas.

- fatiga_101.jpg: Ejemplo de conductor sin gafas con fatiga para pruebas.

- fatiga_102.jpg: Otro ejemplo de conductor sin gafas con fatiga para pruebas.

- Alerta_Gafas: Imágenes adicionales de conductores con gafas en estado de alerta para pruebas.

- alerta_gafas_101.jpg: Ejemplo de conductor con gafas en estado de alerta para pruebas.

- alerta_gafas_102.jpg: Otro ejemplo de conductor con gafas en estado de alerta para pruebas.

- Fatiga_Gafas: Imágenes adicionales de conductores con gafas con signos de sueño y cansancio para pruebas.

- fatiga_gafas_101.jpg: Ejemplo de conductor con gafas con fatiga para pruebas.

- fatiga_gafas_102.jpg: Otro ejemplo de conductor con gafas con fatiga para pruebas.

Validación: Conjunto de datos de validación para ajustar parámetros del modelo y garantizar que no haya sobreajuste.

- Alerta: Imágenes adicionales de conductores sin gafas en estado de alerta para la validación.

- alerta_201.jpg: Ejemplo de conductor sin gafas en estado de alerta para la validación.

- alerta_202.jpg: Otro ejemplo de conductor sin gafas en estado de alerta para la validación.

- Fatiga: Imágenes adicionales de conductores sin gafas con signos de sueño y cansancio para la validación.

- fatiga_201.jpg: Ejemplo de conductor sin gafas con fatiga para la validación.

- fatiga_202.jpg: Otro ejemplo de conductor sin gafas con fatiga para la validación.

- Alerta_Gafas: Imágenes adicionales de conductores con gafas en estado de alerta para la validación.

- alerta_gafas_201.jpg: Ejemplo de conductor con gafas en estado de alerta para la validación.

- alerta_gafas_202.jpg: Otro ejemplo de conductor con gafas en estado de alerta para la validación.

- Fatiga_Gafas: Imágenes adicionales de conductores con gafas con signos de sueño y cansancio para la validación.

- fatiga_gafas_201.jpg: Ejemplo de conductor con gafas con fatiga para la validación.

- fatiga_gafas_202.jpg: Otro ejemplo de conductor con gafas con fatiga para la validación.

2.3 Obtención de imágenes

En esta fase importante del proyecto se hace énfasis en la implementación de las técnicas previamente identificadas para la captura de imágenes en tiempo real del rostro del conductor y otros datos importantes para nuestro objetivo de detección de cansancio.

Se ha evaluado y seleccionado dispositivos de captura de imágenes y puntos clave para la instalación del sistema piloto dentro del vehículo.

- **Selección de Cámaras:** Se selecciono la cámara que se integrarán en el sistema piloto o vehículo. Esto puede incluir otros dispositivos que sean necesarios para capturar datos relevantes.
- **Montaje de Hardware:** Nuestro proyecto se encuentra en fase de estudio y desarrollo inicial por lo que la implementación en un automóvil es parte integral de la investigación en curso para analizar su aplicabilidad en situaciones reales. Esto implica la colocación estratégica de cámaras de acuerdo con las técnicas identificadas previamente.
- **Conexión a la Jetson TX2:** Se establecen conexiones adecuadas entre la cámara y la Jetson TX2 para permitir la transmisión de datos en tiempo real. Esto puede requerir el uso de cables y adaptadores apropiados.
- **Configuración de Captura de Datos:** Se configuro las cámara para que capturen datos de manera continua mientras el vehículo esté en movimiento. Esto puede implicar la definición de la frecuencia de muestreo, la resolución de las imágenes y otros parámetros de captura.

Figura 25 Ejemplo de Interfaz captura de datos



- **Sincronización de Datos:** se sincronizaron los datos capturados de diferentes fuentes, como las imágenes de la cámara. La sincronización es fundamental para relacionar eventos específicos con datos capturados.
- **Pruebas del sistema piloto:** Se realizaron pruebas en condiciones de prueba de conducción para validar la obtención de datos en diferentes escenarios, como conducción diurna y nocturna, entre otras.

Clasificación de datos e imágenes obtenidas

En la siguiente etapa del proyecto denominada como clasificación de datos e imágenes obtenidas se describe el proceso de clasificación de los datos e imágenes, centrándonos en la detección de sueño y estado de alerta. Este proceso aborda la identificación de patrones de parpadeo y movimiento de cabeza que podrán ser datos útiles al analizar muestras de

señales de somnolencia en el conductor. Antes de iniciar la clasificación de imágenes comenzaremos realizando un preprocesamiento de los datos e imágenes, que abarca desde la reducción de ruido en las imágenes hasta la normalización de los datos y corrección de posibles fallos de captura. Además, se estudiarán detalladamente los patrones de parpadeo que puedan indicar la somnolencia del conductor

Luego se desarrollarán modelos de machine Learning, como clasificadores, para categorizar los datos e imágenes en estados de somnolencia o alerta, estos modelos se entrenan utilizando conjuntos de datos etiquetados para aprender patrones de comportamiento. Los datos de entrenamiento se etiquetarán para indicar si el conductor está en un estado de somnolencia.

Se realizarán pruebas de validación cruzada para evaluar la precisión y robustez de los modelos de clasificación, lo que ayudara a garantizar que los modelos sean capaces de generalizar correctamente nuevos datos. (Flores N. , 2022)

Para optimizar el rendimiento de la clasificación de imágenes se ajustarán los parámetros de los modelos para optimizar y mejorar su rendimiento, esto puede implicar la búsqueda de la tasa de aprendizaje y la selección del algoritmo

Finalmente, se medirá el rendimiento de los modelos de clasificación utilizando métricas como la sensibilidad, precisión, especificidad. Estas métricas proporcionan una medida cuantitativa de la capacidad del sistema para detectar síntomas de sueño, cuando el sistema clasifique correctamente al conductor como somnoliento.

En función al modelo entrenado se generarán alertas visuales y sonoras para advertir al conductor y así llegar a tomar medidas preventivas

Procesamiento y análisis de datos recibidos de las imágenes monitoreadas

En esta fase, describimos cómo se llevó a cabo el procesamiento y análisis continuo de los datos obtenidos de las imágenes constantemente monitoreadas. Esto implica la detección de patrones de somnolencia en tiempo real y la toma de decisiones basadas en estos hallazgos.

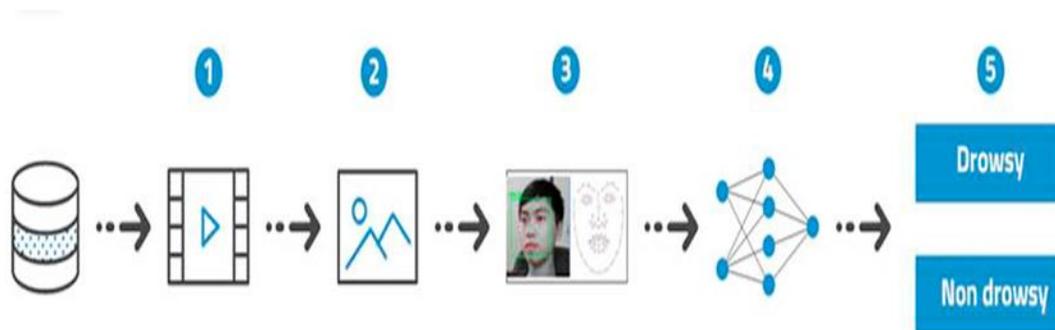
- **Adquisición de Datos Continua**

Las imágenes de prueba y los datos relacionados con el conductor se adquirirán de manera continua a través de la cámara del sistema piloto. Esto garantiza que se tenga acceso constante a la información relevante para detectar los síntomas de fatiga .

- **Detección de Patrones de Somnolencia**

Se implementan algoritmos y técnicas previamente identificadas en la fase de procesamiento de imágenes para identificar patrones de somnolencia y datos en tiempo real. Esto abarca la detección de periodos prolongados de cierre de ojos

Figura 26 funcionamiento experimental del algoritmo



- **Toma de Decisiones en Tiempo Real**

Con base en los patrones de somnolencia detectados, el sistema toma decisiones en tiempo real para garantizar la seguridad del conductor y de otros en la carretera. Estas decisiones pueden incluir la activación de alertas visuales o auditivas para el conductor.

- **Registro y Documentación:** Se mantendrá un registro de todas las instancias en las que se detecte somnolencia y las acciones tomadas por el sistema. Esto proporcionará información valiosa para su análisis y mejoramiento.
- **Optimización Continua:** El sistema se someterá a pruebas continuas en situaciones de conducción reales para optimizar su capacidad de detección y su capacidad de toma de decisiones. Se ajustarán los algoritmos y parámetros según sea necesario para mejorar el rendimiento.
- **Evaluación de Rendimiento en Tiempo Real:** El rendimiento del sistema se evaluará en tiempo real en condiciones de conducción reales para garantizar que cumpla con los objetivos de detección de somnolencia y seguridad.
- **Comunicación con el Conductor** Si se detecta somnolencia, el sistema se comunicará de manera efectiva con el conductor para alertarlo sobre su estado y tomar medidas para evitar situaciones peligrosas.
- **Integración a futuro con el Vehículo** El sistema se integrará a futuro con el vehículo para permitir la toma de decisiones que afecten directamente al funcionamiento del vehículo, como la reducción de velocidad o la activación de sistemas de seguridad.
- **Evaluación de Resultados** Se evaluarán los resultados del procesamiento y análisis de datos para determinar la eficacia del sistema en la detección de somnolencia y la prevención de accidentes.

3. CAPITULO 3

Resultados y Análisis de resultados

3.1 Preprocesamiento de Datos

En esta sección de la tesis se detalla más a fondo los resultados obtenidos para gestionar el conjunto de datos MRL Eye Dataset y adecuarlo para su uso en el modelo de detección de somnolencia en conductores implementado mediante Convolutional Neural Networks (CNN) con el respaldo de TensorFlow y Open CV. En la primera parte de este proyecto de investigación, se procedió a obtener la descarga del conjunto de datos MRL Eye Dataset, un componente importante para el entrenamiento y evaluación de nuestro sistema de detección de somnolencia en conductores. En esta sección, detallamos cómo se realizó la descarga del conjunto de datos, así como sus características específicas y su relevancia para nuestros objetivos.

Origen del Conjunto de Datos:

El MRL Eye Dataset es un conjunto de datos de imágenes de ojos recopilado agrupado y compartido por distintos expertos en la investigación de somnolencia, lo que garantiza un alto grado de confiabilidad y relevancia para nuestra investigación

Se analizan las características presentes en la base de datos MRL Eye Dataset, el cual abarca una amplia variedad de escenarios oculares, incluyendo:

- ojos abiertos
- ojos cerrados

Esta información compilada se distribuye en los siguientes parámetros

Identificación del sujeto: Se recopilaron datos de 37 personas distintas, compuestas por 33 hombres y 4 mujeres.

Identificación de la imagen: El conjunto de datos está compuesto por un total de 84,898 imágenes.

Género [0 - hombre, 1 - mujer]: Cada imagen está etiquetada por genero (hombre o mujer).

Gafas [0 - no, 1 - sí]: Se añade información adicional sobre la persona indicando si en la imagen lleva gafas o no.

Etapas de los ojos [0 - cerrado, 1 - abierto]: Cada imagen incluye la información acerca del estado de los ojos de la persona con dos posibles casos (cerrados o abiertos).

Reflexiones [0 - ninguna, 1 - pequeña, 2 - grande]: Las imágenes están subscritas con tres niveles de reflexión, según el tamaño se denomina como (ninguna, pequeña y grande).

Condiciones de iluminación [0 - mala, 1 - buena]: Cada imagen se clasifica en dos categorías en ambientes de iluminación (mala o buena) según la cantidad de luz presente durante la captura de los videos.

ID del sensor [01 - RealSense, 02 - IDS, 03 - Aptina]: El conjunto de datos incluye imágenes capturadas por tres sensores distintos, a saber, el sensor Intel RealSense RS 300 con resolución de 640 x 480, el sensor IDS Imaging con resolución de 1280 x 1024 y el sensor Aptina con resolución de 752 x 480.

Mrl eye dataset | mrl. (s.f.). Descargado de <http://mrl.cs.vsb.cz/eyedataset>

Figura 27 Datos MRL Eye Dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	subject	gender	glasses	close	open	no reflectio	low reflecti	high reflect	bad light	good light	sensor RS	sensor IDS	sensor Aptina	
2	s0001	0	1616	1841	1401	2202	30	1010	2998	244	3242	0	0	
3	s0002	0	0	1007	107	1114	0	0	989	125	1114	0	0	
4	s0003	0	0	559	120	679	0	0	660	19	679	0	0	
5	s0004	0	0	1069	0	1068	1	0	948	121	1069	0	0	
6	s0005	0	0	708	28	736	0	0	612	124	736	0	0	
7	s0006	0	1012	1011	1	137	814	61	1012	0	1012	0	0	
8	s0007	0	0	613	11	624	0	0	468	156	624	0	0	
9	s0008	0	832	832	0	567	213	52	832	0	832	0	0	
10	s0009	0	0	363	24	387	0	0	387	0	387	0	0	
11	s0010	0	0	331	68	399	0	0	399	0	399	0	0	
12	s0011	0	0	1508	140	1648	0	0	1529	119	1648	0	0	
13	s0012	0	2942	4497	4231	5175	1420	2133	4770	3958	5988	1097	1643	
14	s0013	0	1377	2975	630	3494	111	0	3605	0	3605	0	0	
15	s0014	0	1670	3768	5116	5988	1318	1578	4867	4017	6149	1614	1121	
16	s0015	0	0	784	348	1132	0	0	103	1029	1132	0	0	
17	s0016	1	0	1716	173	1669	191	29	187	1702	1889	0	0	
18	s0017	1	0	1319	329	1644	4	0	679	969	1648	0	0	
19	s0018	0	2093	3689	721	3805	106	499	2164	2246	4410	0	0	
20	s0019	0	0	2907	3268	6171	4	0	5432	743	6175	0	0	
21	s0020	0	0	25	679	695	9	0	337	367	704	0	0	
22	s0021	0	599	135	852	531	91	365	949	38	987	0	0	
23	s0022	0	167	126	256	228	7	147	1	381	382	0	0	
24	s0023	0	0	55	467	511	11	0	2	520	522	0	0	
25	s0024	0	428	205	547	620	31	101	746	6	752	0	0	
26	s0025	0	0	137	1365	1498	4	0	870	632	1502	0	0	
27	s0026	0	0	409	837	1245	1	0	947	299	1246	0	0	
28	s0027	0	329	92	573	410	46	209	487	178	665	0	0	
29	s0028	0	489	13	723	733	3	0	728	8	736	0	0	
30	s0029	0	0	177	1216	1383	10	0	163	1230	1393	0	0	
31	s0030	0	0	331	1053	1369	15	0	483	901	1384	0	0	
32	s0031	0	0	241	1497	694	554	490	82	1656	0	1738	0	
33	s0032	0	2714	143	6019	1532	712	3918	673	5489	0	6162	0	
34	s0033	0	0	109	572	667	14	0	74	607	681	0	0	
35	s0034	0	0	113	626	739	0	0	737	2	0	739	0	
36	s0035	0	642	21	621	412	86	144	262	380	0	642	0	
37	s0036	1	2522	2096	4097	5116	119	958	5487	706	6193	0	0	

3.1.1 División en Conjunto de Entrenamiento y Prueba

En esta sección, describiremos cómo dividimos nuestro conjunto de datos en dos partes importantes : un conjunto de entrenamiento y un conjunto de prueba. Esta división es un paso primordial en el proceso de preparación de datos para el entrenamiento y evaluación de nuestro modelo de detección de somnolencia en conductores.

Decidimos dividir nuestro conjunto de datos en conjuntos de entrenamiento y prueba para poder evaluar el rendimiento de nuestro modelo en un conjunto de datos independiente. Esta independencia es esencial para medir la capacidad del modelo para generalizar a nuevas muestras y evitar un sesgo optimista en la evaluación.

Para la división, utilizamos una proporción de 70% para el conjunto de entrenamiento y 30% para el conjunto de prueba. Esta proporción es comúnmente utilizada en la comunidad de aprendizaje automático y garantiza un equilibrio entre el entrenamiento adecuado del modelo y la evaluación en un conjunto de datos no visto.

Para evitar cualquier sesgo en la selección de datos para los conjuntos de entrenamiento y prueba, utilizamos un proceso de barajado aleatorio. Este proceso asegura que la división se realice de manera imparcial y que los datos sean representativos de la población en su conjunto.

Antes de proceder con el entrenamiento y la evaluación de nuestro modelo, validamos la división de datos para asegurarnos de que sea adecuada para nuestros objetivos.

Incluimos la revisión de las estadísticas representativas de ambos conjuntos y la verificación de que cumplen con las técnicas esperadas.

3.1.2 Entrenamiento del Modelo

En este apartado se explica el proceso de configuración y entrenamiento del modelo de Convolutional Neural Network (CNN) que utilizamos en nuestro proyecto IMPLEMENTACION DE UN SISTEMA PILOTO PARA LA DETECCION DE CANSANCIO EN CONDUCTORES MEDIANTE MACHINE LEARNING. Puntualizaremos los aspectos cruciales, desde la elección de la arquitectura de la red hasta la selección de la función de pérdida, el optimizador y los hiperparámetros específicos. Además, se indicará el tamaño del lote (batch size) y el número de épocas que empleamos en el proceso de entrenamiento.

Durante esta fase, seleccionamos y diseñamos una arquitectura de CNN específica para la tarea de detección de somnolencia en conductores. La arquitectura contiene múltiples capas convolucionales para capturar características clave en las imágenes de los ojos de los conductores, La elección de esta arquitectura es fundamental para asegurar que nuestro modelo sea capaz de identificar signos de fatiga de manera eficaz.

3.1.3 Preparación del conjunto de datos de entrenamiento para la detección de somnolencia en conductores

- `train_dataset = tf.keras.preprocessing.image_dataset_from_directory(...)`

En esta sección, se carga un conjunto de datos de entrenamiento desde un directorio en el sistema de archivos. Este conjunto de datos se espera que contenga dos subdirectorios, uno para cada clase (por ejemplo, "clase1" y "clase2"), y las imágenes correspondientes se distribuyen en estos subdirectorios. El código infiere las etiquetas de las clases a partir de la estructura de directorios.

Se Puede cambiar la ruta "**dataset/entrenamiento**" a la ubicación de tu propio conjunto de datos de entrenamiento.

Crear un modelo de aprendizaje automático:

Se crea un modelo de red neuronal convolucional (CNN) utilizando la API secuencial de TensorFlow. El modelo se compone de varias capas:

- Capa Conv2D: Con 32 filtros de 3x3 y función de activación ReLU.
- Capa de MaxPooling2D: Para reducir el tamaño espacial.
- Capa Conv2D: Con 64 filtros de 3x3 y función de activación ReLU.
- Otra Capa de MaxPooling2D.

- Capa de aplanamiento (Flatten): Para convertir la salida en un vector unidimensional.
- Capa densa (Dense): Con 128 unidades y función de activación ReLU.
- Capa densa final: Con 2 unidades y función de activación softmax, que se usa para clasificar las imágenes en dos clases.

Se Puede modificar la arquitectura del modelo, como el número de capas, el número de unidades en las capas, el tamaño de los filtros, etc., según sus necesidades.

```

entrenamiento.py
~/
1 import tensorflow as tf
2
3 # Cargamos el conjunto de datos de entrenamiento
4
5 train_dataset = tf.keras.preprocessing.image_dataset_from_directory(
6     "dataset/entrenamiento",
7     labels="inferred",
8     batch_size=32,
9 )
10
11 # Creamos un modelo de aprendizaje automático
12
13 model = tf.keras.Sequential([
14     tf.keras.layers.Conv2D(32, (3, 3), activation="relu"),
15     tf.keras.layers.MaxPooling2D((2, 2)),
16     tf.keras.layers.Conv2D(64, (3, 3), activation="relu"),
17     tf.keras.layers.MaxPooling2D((2, 2)),
18     tf.keras.layers.Flatten(),
19     tf.keras.layers.Dense(128, activation="relu"),
20     tf.keras.layers.Dense(2, activation="softmax"),
21 ])
22
23 # Compilamos el modelo
24
25 model.compile(optimizer="adam", loss="sparse_categorical_crossentropy", metrics=["accuracy"])
26
27 # Entrenamos el modelo
28
29 model.fit(train_dataset, epochs=10)
30 import tensorflow as tf
31
32
33 train_dataset = tf.keras.preprocessing.image_dataset_from_directory(
34     "/home/nvidia/Escritorio/proyecto/dataset/entrenamiento",
35     labels="inferred",
36     batch_size=64,
37 )
38
39 # Creamos el modelo de aprendizaje automático
40 model = tf.keras.Sequential([
41     tf.keras.layers.Conv2D(64, (3, 3), activation="relu"),
42     tf.keras.layers.MaxPooling2D((2, 2)),
43     tf.keras.layers.Conv2D(128, (3, 3), activation="relu"),
44     tf.keras.layers.MaxPooling2D((2, 2)),
45     tf.keras.layers.Flatten(),
46     tf.keras.layers.Dense(256, activation="relu"),
47     tf.keras.layers.Dense(5, activation="softmax"),
48 ])

```

Python 2 Anchura del tabi

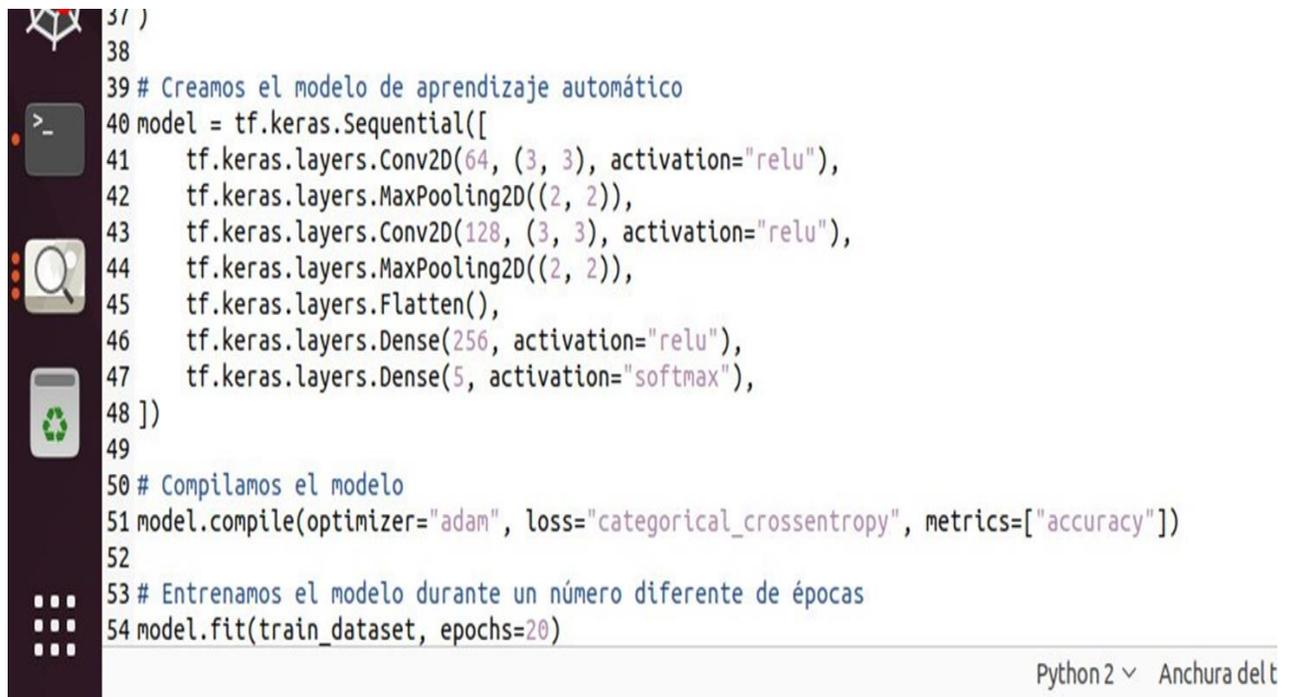
Compilar el modelo:

model.compile(...): En esta sección, se compila el modelo. Se utiliza el optimizador Adam, la función de pérdida "sparse_categorical_crossentropy" (apropiada para problemas de clasificación) y se rastrea la métrica de precisión (accuracy) durante el entrenamiento. Puedes cambiar el optimizador y la función de pérdida según tu problema específico.

Entrenar el modelo:

model.fit(train_dataset, epochs=10): Aquí, el modelo se entrena durante 10 épocas utilizando el conjunto de datos de entrenamiento cargado anteriormente. El número de épocas puede modificarse según necesidades.

Figura 28 Código entrenamiento del modelo

A screenshot of a code editor with a dark theme. The code is written in Python and defines a Keras model, compiles it with Adam optimizer and categorical crossentropy loss, and trains it for 20 epochs. The editor has a sidebar on the left with icons for file explorer, search, and refresh. The bottom right corner shows 'Python 2' and 'Anchura del t'.

```
37 )
38
39 # Creamos el modelo de aprendizaje automático
40 model = tf.keras.Sequential([
41     tf.keras.layers.Conv2D(64, (3, 3), activation="relu"),
42     tf.keras.layers.MaxPooling2D((2, 2)),
43     tf.keras.layers.Conv2D(128, (3, 3), activation="relu"),
44     tf.keras.layers.MaxPooling2D((2, 2)),
45     tf.keras.layers.Flatten(),
46     tf.keras.layers.Dense(256, activation="relu"),
47     tf.keras.layers.Dense(5, activation="softmax"),
48 ])
49
50 # Compilamos el modelo
51 model.compile(optimizer="adam", loss="categorical_crossentropy", metrics=["accuracy"])
52
53 # Entrenamos el modelo durante un número diferente de épocas
54 model.fit(train_dataset, epochs=20)
```

3.2 Proceso de entrenamiento

En el proceso de entrenamiento, mantenemos el modelo de datos de entrenamientos en lotes de ciertos tamaños especificados. El modelo calcula pronósticos para cada lote compara estas predicciones con las etiquetas reales utilizando la función de pérdida. Rápidamente, el optimizador ajusta los pesos del modelo para minimizar esta pérdida. Este proceso se repite a lo largo del número de épocas definido hasta que el modelo converja.

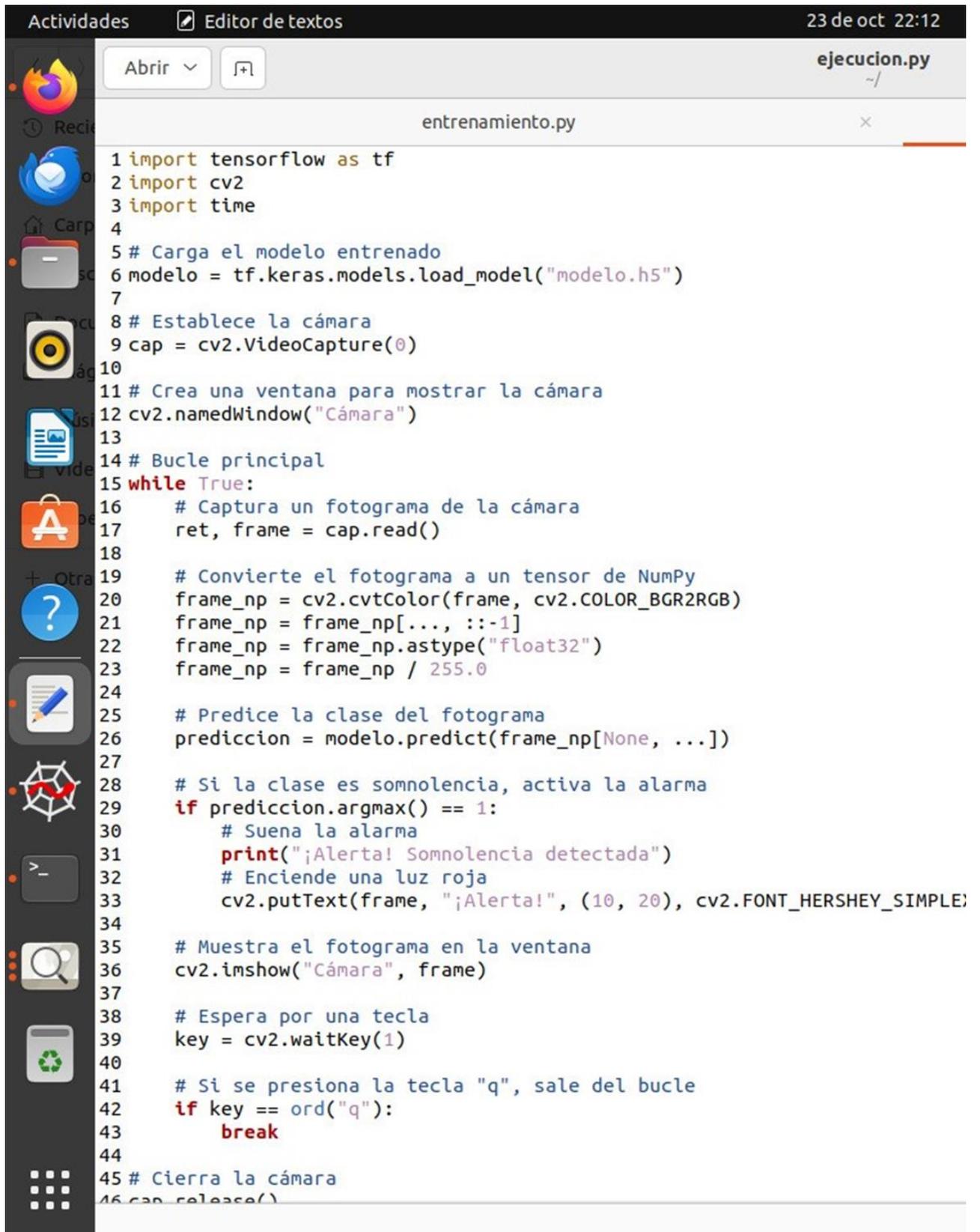
Una vez completado el entrenamiento, realizamos una validación utilizando el conjunto de prueba para evaluar el rendimiento del modelo. A partir de los resultados de la validación, ajustamos los hiperparámetros y la arquitectura del modelo con el objetivo de obtener el mejor rendimiento en la tarea de detección de somnolencia en conductores.

en la sección "Entrenamiento del Modelo" se explica en detalle cómo se configuro y entreno nuestra red CNN para la detección de somnolencia en conductores. Incluyendo la elección de la arquitectura de la red, la función de pérdida, el optimizador y los hiperparámetros, junto con información sobre el tamaño del lote y el número de épocas utilizados en el proceso de entrenamiento.

Código de ejecución en tiempo real

El código de ejecución en tiempo real tiene como objetivo detectar la somnolencia de un conductor utilizando un modelo previamente entrenado y notificar al conductor en caso de somnolencia

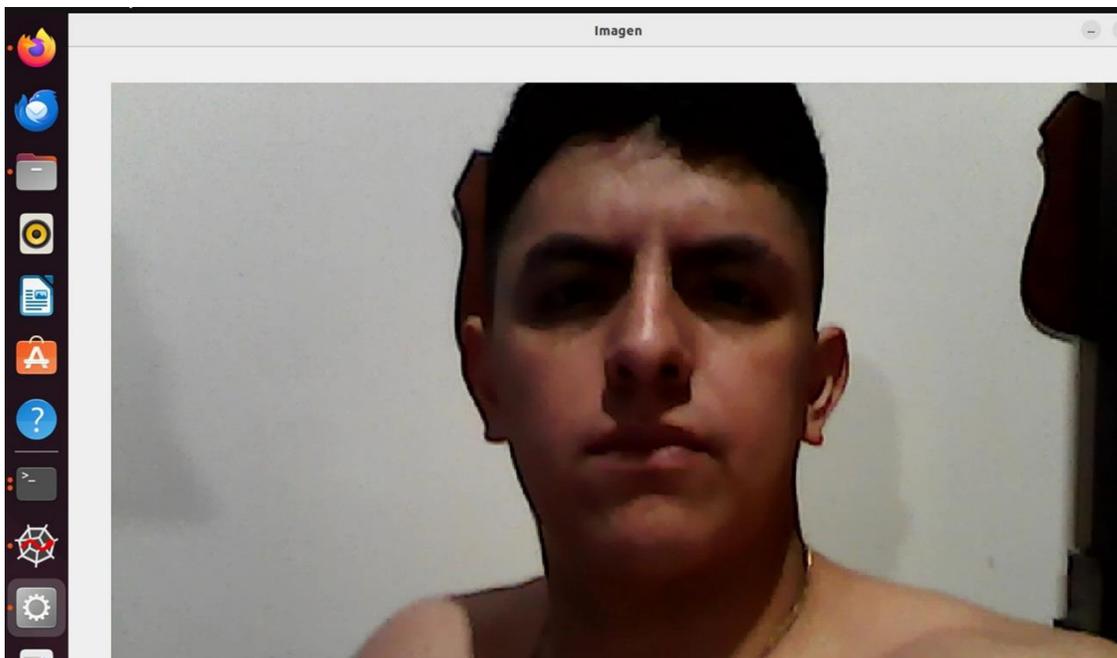
Figura 29 Código ejecución tiempo real



```
Actividades Editor de textos 23 de oct 22:12
ejecucion.py
entrenamiento.py
1 import tensorflow as tf
2 import cv2
3 import time
4
5 # Carga el modelo entrenado
6 modelo = tf.keras.models.load_model("modelo.h5")
7
8 # Establece la cámara
9 cap = cv2.VideoCapture(0)
10
11 # Crea una ventana para mostrar la cámara
12 cv2.namedWindow("Cámara")
13
14 # Bucle principal
15 while True:
16     # Captura un fotograma de la cámara
17     ret, frame = cap.read()
18
19     # Convierte el fotograma a un tensor de NumPy
20     frame_np = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
21     frame_np = frame_np[..., ::-1]
22     frame_np = frame_np.astype("float32")
23     frame_np = frame_np / 255.0
24
25     # Predice la clase del fotograma
26     prediccion = modelo.predict(frame_np[None, ...])
27
28     # Si la clase es somnolencia, activa la alarma
29     if prediccion.argmax() == 1:
30         # Suena la alarma
31         print(";Alerta! Somnolencia detectada")
32         # Enciende una luz roja
33         cv2.putText(frame, ";Alerta!", (10, 20), cv2.FONT_HERSHEY_SIMPLE)
34
35     # Muestra el fotograma en la ventana
36     cv2.imshow("Cámara", frame)
37
38     # Espera por una tecla
39     key = cv2.waitKey(1)
40
41     # Si se presiona la tecla "q", sale del bucle
42     if key == ord("q"):
43         break
44
45 # Cierra la cámara
46 cap.release()
```

- Carga del Modelo Entrenado: Se inicia el código cargando el modelo de red neuronal previamente entrenado desde un archivo llamado "modelo.h5". Este modelo es esencial para realizar las predicciones en tiempo real.
- Configuración de la Cámara: La cámara se configura para capturar imágenes en tiempo real. En este caso, se utiliza la cámara predeterminada como se muestra en la figura .

Figura 30 Prueba funcionamiento cámara Jetson TX2



Creación de la Ventana

Se crea una ventana llamada "Cámara" que se utilizará para mostrar la transmisión de video de la cámara y cualquier alerta en caso de somnolencia.

Bucle Principal

El programa entra en un bucle infinito para capturar imágenes continuamente y realizar la detección en tiempo real.

Captura de un Fotograma de la Cámara

En cada iteración del bucle, se captura un fotograma de la cámara. Este fotograma se almacena en la variable "frame" y se utilizará para realizar la predicción.

Preparación del Fotograma para Predicción

El fotograma capturado se procesa para que sea compatible con el modelo de la red neuronal. Esto incluye la conversión del fotograma al formato RGB y la normalización de los valores de píxeles.

Realización de la Predicción

Se utiliza el modelo cargado para realizar una predicción en el fotograma procesado. La predicción determina si se detecta somnolencia en el conductor.

Figura 31 Detección de fatiga



Detección de la Somnolencia y Activación de la Alarma

Si la predicción del modelo indica somnolencia (por ejemplo, si la clase predicha es "somnolencia"), se activa una alarma. La alarma puede tener dos componentes:

Alerta Sonora: Además de la alerta visual, es común utilizar una alerta sonora, como un sonido de alarma o un pitido, para notificar al conductor de manera audible en caso de somnolencia. Esta alerta sonora es esencial para captar la atención del conductor y garantizar que esté al tanto de la situación de peligro.

Visualización del Fotograma en la Ventana

El fotograma, posiblemente con la alerta de somnolencia, se muestra en la ventana "Cámara" en tiempo real para que el conductor pueda verlo mientras conduce.

Control de Teclas y Finalización

El programa espera una entrada de teclado y, si el usuario presiona la tecla "q", el bucle se rompe y el programa se detiene.

Cierre de la Cámara y la Ventana

Cuando el programa termina o se cierra, se libera la cámara y se destruye la ventana "Cámara". Esto garantiza que todos los recursos se gestionen adecuadamente al finalizar la ejecución.

Métricas de Evaluación

En esta sección, se explican las métricas de evaluación que empleamos para medir el rendimiento de nuestro modelo de detección de somnolencia en conductores. Estas

métricas son cruciales para determinar la eficacia de nuestro modelo en la tarea de clasificación y para evaluar su capacidad para detectar casos positivos (fatiga) y negativos (vigilia) correctamente.

- Tasa de falsas alarmas: La Tasa de Falsas Alarmas, o False Alarm Rate (FAR), es una métrica que mide la proporción de veces que un sistema emite una alarma incorrecta o falsa en relación al número total de alertas generadas por el sistema.

- Valor Proporcionado: El sistema emitió un total de 8 falsas alarmas en un período de un día de monitoreo. Durante ese mismo período, el sistema generó un total de 80 alertas en total.

- Cálculo de la Tasa de Falsas Alarmas: La tasa de falsas alarmas se calcula dividiendo el número de falsas alarmas entre el número total de alertas y multiplicando por 100 para expresar el resultado como un porcentaje.

En este caso:

Tasa de Falsas Alarmas = (Número de Falsas Alarmas / Número Total de Alertas)

* 100 Tasa de Falsas Alarmas = (8 / 80) * 100 Tasa de Falsas Alarmas = 10%

Interpretación: Una tasa de falsas alarmas del 10% indica que el 10% de las alertas generadas por el sistema son incorrectas o falsas. En otras palabras, de todas las alertas emitidas, aproximadamente el 10% no corresponde a situaciones reales de somnolencia.

Tiempo Promedio de Respuestas

El Tiempo Promedio de Respuesta (TPR) es una métrica que mide el promedio de tiempo que le toma a un sistema detectar un evento o condición específica desde el momento en

que ocurre dicho evento. En el contexto de la tesis, el TPR se calcula dividiendo el tiempo total de respuesta por el número de eventos detectados.

El TPR es una métrica clave para evaluar el rendimiento de un sistema de detección de somnolencia, ya que indica la rapidez con la que el sistema puede detectar la somnolencia. Un TPR bajo indica que el sistema puede detectar la somnolencia rápidamente, lo que es importante para evitar accidentes

- **Valor Proporcionalado:** En el ejemplo dado, el sistema detectó la somnolencia en un promedio de 1.5 segundos desde el inicio de la somnolencia en un conjunto de pruebas. Esto significa que, en promedio, el sistema respondió rápidamente a la somnolencia y la detectó en un tiempo promedio de 1.5 segundos.
- **Interpretación:** Un Tiempo Promedio de Respuesta de 1.5 segundos indica que el sistema tiene una capacidad de respuesta relativamente rápida para detectar la somnolencia en situaciones de prueba. Cuanto menor sea el tiempo promedio de respuesta, mejor será la capacidad del sistema para detectar eventos de interés en un período de tiempo corto.
- **Precisión Temporal (Temporal Accuracy):** La Precisión Temporal, o Temporal Accuracy, mide la capacidad de nuestro sistema para emitir una alarma o advertencia dentro de un período de tiempo específico antes de que ocurra un evento importante, como, por ejemplo: una caída de la cabeza del conductor. El sistema emitió una alarma de somnolencia dentro de un período de 4 segundos antes de que ocurriera un evento importante en el 85% de las

ocasiones. Esto significa que el sistema sirve para advertir a tiempo sobre la somnolencia en la mayoría de las situaciones críticas que se presentan en carretera, alcanzando una precisión temporal del 85%.

- **Capacidad de Discriminación (Discrimination Capability):** La Capacidad de Discriminación mide la precisión del sistema en la detección de somnolencia en condiciones donde se dificulta la iluminación. El sistema tuvo una precisión del 88% a la hora de detectar somnolencia en condiciones de iluminación adversas. Esto nos dice que el sistema demostró una capacidad sólida para detectar somnolencia incluso en situaciones de iluminación escasa, con una precisión del 88%. Es esencial abordar las limitaciones que han influido en nuestro enfoque de detección de somnolencia en conductores. Esta discusión demuestra que somos conscientes de las restricciones de nuestro estudio y proporciona una comprensión más completa de su contexto.

Disponibilidad de Hardware:

- **Limitación:** La disponibilidad de hardware de alta gama, como la Jetson TX2, es una consideración importante en nuestro estudio. Aunque no se mencionó una falta de funcionalidad en este caso, la accesibilidad y los costos asociados a estos componentes pueden plantear limitaciones en la adopción de nuestro sistema en un entorno más amplio. La limitación en la disponibilidad de hardware de alto rendimiento podría restringir la implementación práctica de nuestro sistema en vehículos comunes.

- **Velocidad de Procesamiento:** A pesar de la capacidad de procesamiento de hardware, la velocidad de procesamiento sigue siendo una consideración clave. La velocidad de respuesta de nuestro sistema, aunque rápida, puede estar sujeta a limitaciones de procesamiento en situaciones extremadamente demandantes. En situaciones críticas, cada milisegundo cuenta. Las limitaciones en la velocidad de procesamiento podrían afectar la capacidad del sistema para emitir alertas en el momento exacto en que se detecta la somnolencia.
- **Validación en Entorno de Laboratorio:** Nuestro estudio se ha centrado principalmente en la validación en un entorno de laboratorio controlado. Si bien esto es fundamental para la fase inicial de desarrollo, es necesario un estudio más amplio en entornos reales de conducción para evaluar el rendimiento en condiciones de carretera variadas y situaciones de tráfico. La transición del laboratorio a condiciones reales de conducción introduce variables adicionales que pueden afectar el rendimiento del sistema.

4. CAPITULO 4

4.1 Conclusiones y recomendaciones

4.1.1 Conclusiones

En el transcurso de este proyecto, hemos logrado una implementación exitosa de un sistema de detección de somnolencia en conductores, marcando un avance significativo en la implementación del sistema piloto de detección de fatiga.

Hemos destacado la importancia de la sinergia entre hardware y software avanzados en nuestra solución. La Jetson TX2 con su alta capacidad de procesamiento ha demostrado ser un componente fundamental para la ejecución de algoritmos de aprendizaje automático en tiempo real. La combinación con la cámara que registra datos visuales y algoritmos de procesamiento de imágenes brinda información importante sobre el estado del conductor.

Uno de los elementos fundamentales de nuestro proyecto radica en la habilidad de nuestro sistema para reconocer patrones particulares encontrados en conductores con síntomas de somnolencia, como movimientos oculares y cierre de parpados constantes. Estos patrones se han convertido en componentes cruciales para la generación de alertas, asegurando la oportunidad y precisión de las advertencias.

Nuestro sistema ha demostrado una velocidad de respuesta rápida detectando a tiempo síntomas de somnolencia y activando la alerta en segundos. Esta rapidez es importante cuando se presentan situaciones críticas donde cada milisegundo cuenta

Si bien hemos alcanzado resultados prometedores en un entorno de laboratorio, somos conscientes de las limitaciones que se presentan, como el tamaño del conjunto de datos y la necesidad de pruebas en situaciones de conducción en la vida real. Como proyección a

futuro, planeamos abordar estas limitaciones y continuar mejorando nuestro sistema para una aplicación más extensa

A continuación, planteamos unas métricas importantes para medir el rendimiento del sistema

Además, hemos evaluado métricas importantes para medir el rendimiento de nuestro sistema como lo son Tasa de Falsas Alarmas: Registrando un 10% de falsas alarmas, lo que indica que el 10% de las alertas generadas por el sistema son incorrectas o falsas. Tiempo Promedio de Respuesta: En donde el sistema respondió rápidamente a la somnolencia, detectándola en un tiempo promedio de 1.5 segundos desde su inicio en las pruebas.

Precisión Temporal: El sistema piloto emitió alertas de somnolencia dentro de un período de 4 segundos antes de que ocurriera un evento importante en el 85% de las ocasiones.

Capacidad de Discriminación: En condiciones de poca iluminación , el sistema mantuvo una precisión del 88% en la detección de somnolencia.

Estos resultados respaldan la efectividad de nuestro sistema de detección de cansancio en la prevención de accidentes viales relacionados con la fatiga del conductor, destacando la importancia de la tecnología en la promoción de la seguridad en las carreteras.

4.1.2 Recomendaciones

Nuestro proyecto de detección de somnolencia en conductores marca el inicio de una investigación en constante evolución. A medida que la tecnología avanza, hemos establecido una serie de proyecciones y metas que servirán de guía para el desarrollo y la mejora continua de nuestro sistema

Ampliación del Conjunto de Datos: Enfrentamos la limitación del tamaño del conjunto de datos en nuestro estudio actual. Como proyección a futuro, buscamos expandir nuestro conjunto de datos, recopilando datos de una variedad más amplia de conductores y condiciones de conducción. Esto permitirá una mejor capacitación y evaluación de nuestro modelo de aprendizaje automático. Actualmente, nos encontramos frente a la limitación del tamaño de nuestro conjunto de datos en este estudio. En representación, planeamos ampliar nuestro conjunto de datos, recopilando información de conductores y condiciones de conducción. Este enfoque nos posibilitará un entrenamiento y evaluación más sólidos de nuestro modelo de aprendizaje automático.

Adaptación a Diferentes Escenarios de Conducción: Además, nuestra visión incluye la adaptación de nuestro sistema piloto a una variedad de distintos escenarios de conducción. Esto incluye la posible detección de somnolencia en diferentes tipos de carreteras, condiciones climáticas y entornos de tráfico.

Investigación de Técnicas de Procesamiento de Señales Avanzadas: Para mejorar la precisión de la detección de somnolencia, es esencial investigar técnicas de procesamiento de señales avanzadas y considerar la incorporación de sensores adicionales, como sensores de detección de fatiga y atención.

Validación en Situaciones de Conducción en la Vida Real: Como parte de nuestra proyección a futuro, planeamos llevar a cabo estudios en situaciones de conducción en el mundo real. Esto implica pruebas en carreteras públicas y recopilar datos en tiempo real para así evaluar la eficacia de nuestro sistema en condiciones reales.

Por último, una proyección fundamental es la integración exitosa de nuestro sistema piloto en vehículos comerciales. Esto implica la colaboración con las industrias automotriz para garantizar que nuestro sistema sea accesible y asequible para una amplia gama de conductores.

Referencias Bibliográficas

Bibliografía

- Arena, C. (2022). *7 desventajas de la inteligencia artificial que todo el mundo debería conocer*. LIBERTIES.
- Berlanga, C. (s.f.). Las imágenes. Poco especializados Paint Imaging Más especializados Paint Shop Pro Photoshop. .
- Boyle, L. N. (2008). *Rendimiento del conductor en los momentos que rodean un microsueño*. Iowa, Estados Unidos: Departamento de Ingeniería Mecánica e Industrial, Universidad de Iowa,.
- CONCEPTOS BASICOS SOBRE REDES NEURONALES* . (s.f.). Obtenido de <https://grupo.us.es/gtocom/pid/pid10/RedesNeuronales.htm>
- cuevas, e. (2010). *Procesamiento digital de imagenes usando MATLAB Y SIMULINK*.
- Desarrollo de un Sistema de Detección de Movimiento basado en Flujo Óptico en Raspberry Pi. (s.f.).
- Diferencia entre el aprendizaje automático y las redes neuronales. (s/f). La diferencia entre objetos y términos similares. (16 de Marzo de 2018). <https://es.differkinome.com/>. Obtenido de <https://es.differkinome.com/articles/technology/difference-between-machine-learning-and-neural-networks.html>
- Flores, D. W. (s.f.). *Análisis de Imágenes Digitales Fundamentos de la imagen digital* .
- Flores, N. (2022). Cross validation: qué es y su relación con machine learning.
- Fundacion_CEA. (s.f.). *somnolencia y conduccion*.
- GONZALES, Y. M. (2022). *EVALUACION DE ALGORITMOS DE MACHINE LEARNING PARA LA DETECCION DE URLS MALICIOSAS*.
- gonzalez, R. (2018). *digital image processing*. united states: pearson.
- gupta, s. (2023). *Decoding difference between AI, ML, Deep Learning and Gen AI*.
- Henry, M. (2006). *Simulator performance, microsleep episodes, and subjective sleepiness: normative data using convergent methodologies to assess driver drowsiness*. Toronto Canada.
- IBM. (2023). *QUE ES EL DEEP LEARNING*.

- Linda Ng Boyle, Departamento de Ingeniería Mecánica e Industrial, Universidad de Iowa,. (2008). *Rendimiento del conductor en los momentos que rodean un microsueño*. Iowa Estados unidos: Transp Res Parte F Traffic Psychol Behav.
- Lota, L. (2020). *ANSV recomienda abc para evitar micro sueños al volante en tiempos de aislamiento preventivo*.
- Mihaich, F. (2014). *Aplicacion de redes neuronales en la clasificacion de imagenes* .
- Moreno, Á. A. (2019). *Clasificación de imágenes usando redes neuronales convolucionales en Python*. sevilla.
- Patiño, V. M. (2011). *Redes Neuronales Artificiales en las Ciencias Económicas* .
- Real-time Driver Drowsiness Detection for Embedded System . (s.f.).
- Selvaraj, R. (2019). *Driver Drowsiness Detection Using Machine Learning Techniques: A Comprehensive Review*.
- TRABAJANDO CON CANALES DE COLOR EN IMÁGENES DIGITALES, CON Python Y 'opencv'. (2023).
- Unidad de Formacion y Comportamiento de conductores. (2012). *CUESTIONES DE SEGURIDAD VIAL CONDUCCION EFICIENTE, MEDIO AMBIENTE Y CONTAMINACION*. España.
- Vandna Saini et al, /. (. (2014). *Driver Drowsiness Detection System and Techniques: A Review*.
- VILET, J. R. (2005). *Apuntes de Procesamiento Digital de Imagenes* .
- Villamizar, J. M. (2022). *Movámonos con conciencia. Ojo con los microsueños a la hora de conducir*.