

main ▾

[Go to file](#)[Add file ▾](#)[Code ▾](#)**About**

Proyecto final para el Bootcamp en Visión Computacional para los Objetivos de Desarrollo Sostenible Hackcities

[Readme](#)[MIT license](#)[0 stars](#)[1 watching](#)[0 forks](#)**Releases**

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Contributors 2

- 
- BrunoBustos96**
- Bruno Bustos
-
- 
- isra23aj**

Languages

- Jupyter Notebook 99.6%
- Python 0.4%

 BrunoBustos96	Merge branch 'main' of https://github.com/BrunoBustos96/DeteccionSomnolenciaCNN ...	2 minutes ago	66
 1_Detector_De_Somnolencia_P...	Created using Colaboratory	yesterday	
 2_Desarrollo_del_modelo_Dete...	Created using Colaboratory	23 hours ago	
 3_Aplicacion_del_Modelo_Final...	Created using Colaboratory	5 hours ago	
 3_aplicacion_del_modelo_final_...	Subiendo el ejecutable .py	31 minutes ago	
 DeteccionSomnolenciaCNN_Pa...	Links y Captura pág Github	1 hour ago	
 LICENSE	Update LICENSE	2 hours ago	
 LinksAlTrabajo.txt	Links y Captura pág Github	1 hour ago	
 README.md	Update README.md	20 minutes ago	
 ledsANDmotor.ino	Subiendo código de Arduino	3 minutes ago	

README.md



Detección de Somnolencia usando Redes Neuronales Convolucionales

Proyecto final para el [Bootcamp en Visión Computacional para los Objetivos de Desarrollo Sostenible Hackcities](#)



DeteccionSomnolenciaCNN

El Backend del proyecto final de HackCities fue desarrollado en Pycharm y Colaboratory utilizando librerías de Procesamiento de Imágenes y Aprendizaje Profundo.

Motivación

Los accidentes de tránsito son un grave problema en nuestro país, y aunque los efectos del alcohol y el cambio de carril sean las principales causas de dichos accidentes, la tercera más relevante es la somnolencia de los conductores, problema que puede ser atacado con una solución que implemente visión computacional en dispositivos de bajo costo.

Dentro de los Objetivos de Desarrollo Sostenible, este proyecto se concentra principalmente en 2: ODS 3 "Salud y bienestar", buscando mejorar la seguridad vial de peatones y conductores; y el ODS 11 "Industria, innovación e infraestructura", buscando proponer soluciones innovadoras en nuestro entorno que puedan tener un impacto económico positivo.

Link al video



Capturas de pantalla e imágenes importantes

Definición del problema

Objetivos de desarrollo sostenible objetivo:

The image shows two cards from the United Nations Sustainable Development Goals. The top card is green and focuses on Goal 3: Health and Well-being. It features a white box containing the number '3' and the text 'SALUD Y BIENESTAR' above a heart rate monitor icon with a heart. To the right of the box, the text reads: 'Garantizar una vida sana y promover el bienestar para todos en todas las edades'. The bottom card is orange and focuses on Goal 9: Industry, Innovation, and Infrastructure. It features a white box containing the number '9' and the text 'INDUSTRIA, INNOVACIÓN E INFRAESTRUCTURA' above an icon of three stacked cubes. To the right of the box, the text reads: 'Construir infraestructuras resilientes, promover la industrialización inclusiva y sostenible y fomentar la innovación'.

Objetivos

Objetivos

OBJETIVO GENERAL

Desarrollar un sistema de alerta de sueño al volante que informe al conductor su estado de vigilia en tiempo real con un modelo de CNNs.



OBJETIVOS ESPECÍFICOS

- Recolectar un dataset óptimo de rostros con ojos abiertos y cerrados
- Generar un modelo con métricas mayores a 0.90
- Aplicar el modelo en tiempo real con una cámara web
- Aplicar un sistema de alerta tanto sensorial/visual (manilla) y auditivo.

Modelo de entrenamiento CNN

ENFOQUES Y RESULTADOS

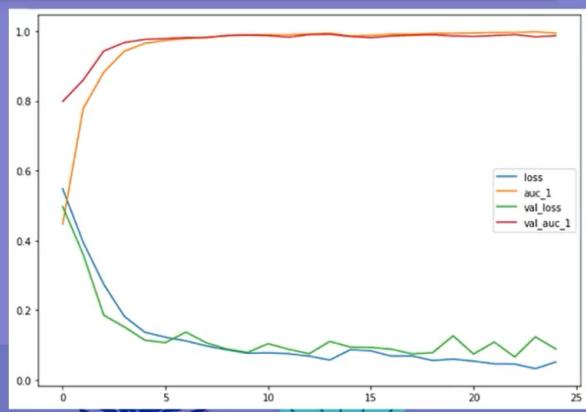
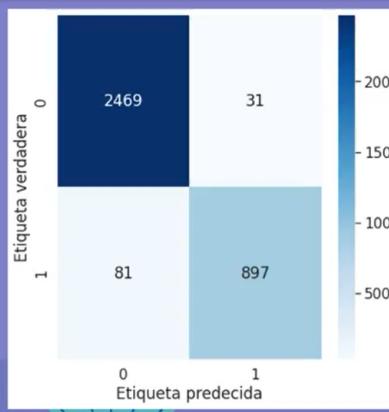
MODELO DE ENTRENAMIENTO CON CNNs

Model: "sequential_1"		
Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 78, 78, 32)	896
max_pooling2d_2 (MaxPooling 2D)	(None, 39, 39, 32)	0
conv2d_3 (Conv2D)	(None, 37, 37, 32)	9248
max_pooling2d_3 (MaxPooling 2D)	(None, 18, 18, 32)	0
flatten_1 (Flatten)	(None, 10368)	0
dense_4 (Dense)	(None, 256)	2654464
dropout_3 (Dropout)	(None, 256)	0
dense_5 (Dense)	(None, 512)	131584
dropout_4 (Dropout)	(None, 512)	0
dense_6 (Dense)	(None, 1024)	525312
dropout_5 (Dropout)	(None, 1024)	0
dense_7 (Dense)	(None, 1)	1025
<hr/>		
Total params: 3,322,529		
Trainable params: 3,322,529		
Non-trainable params: 0		

Métricas

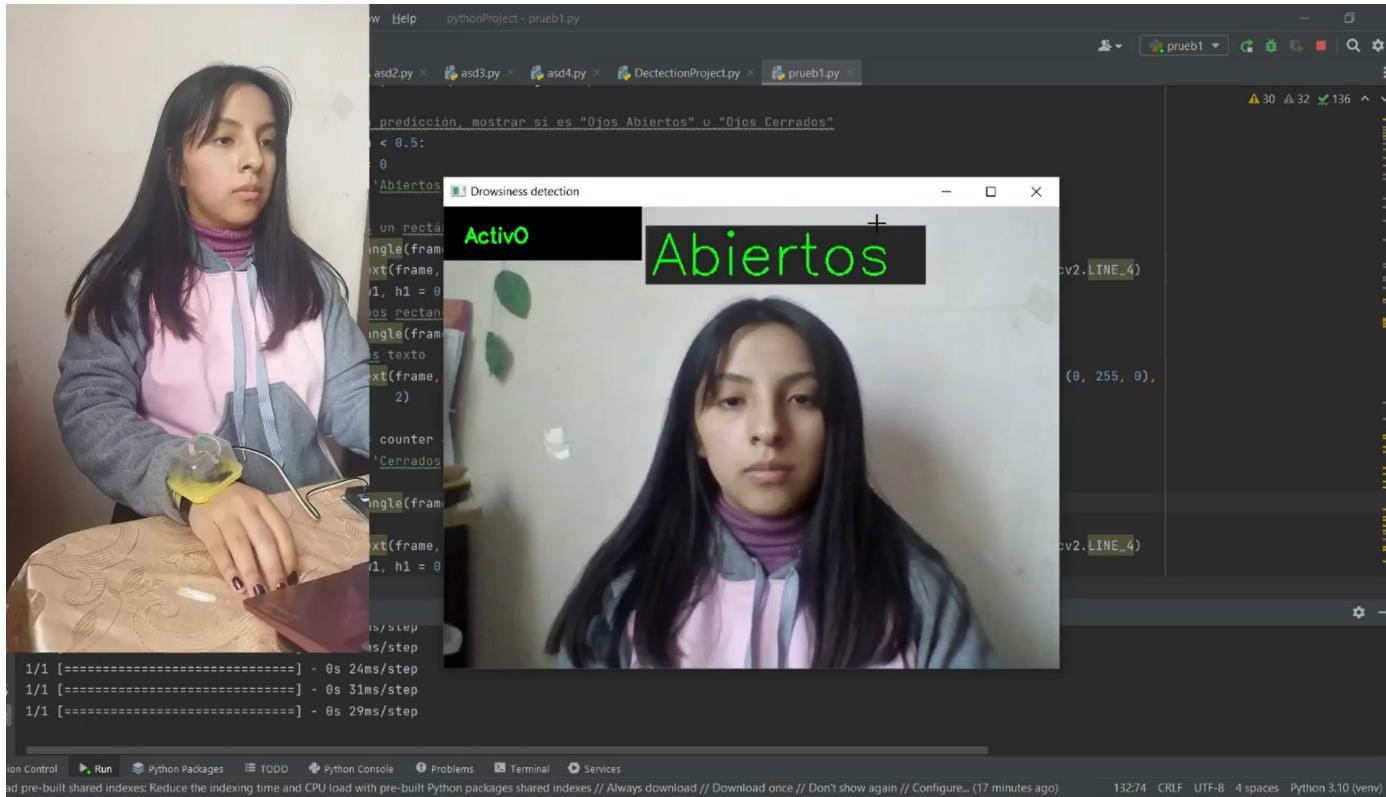
ENFOQUES Y RESULTADOS MÉTRICAS, CREACIÓN DE UN PROTOTIPO FÍSICO

ACCURACY	PRECISION	RECALL	SPECIFICITY	F1_SCORE
96.77%	96.65%	91.71%	98.76%	94.12%

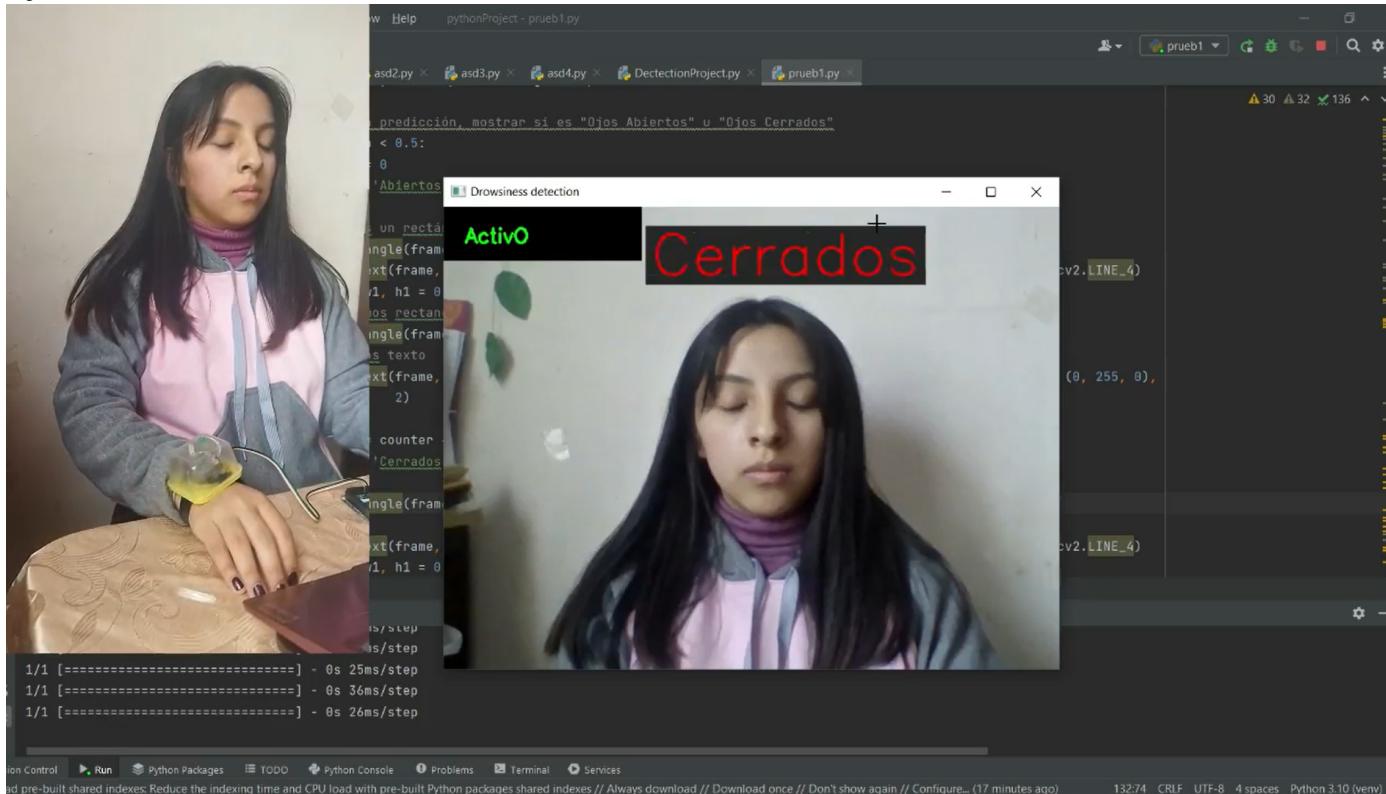


Resultados experimentales

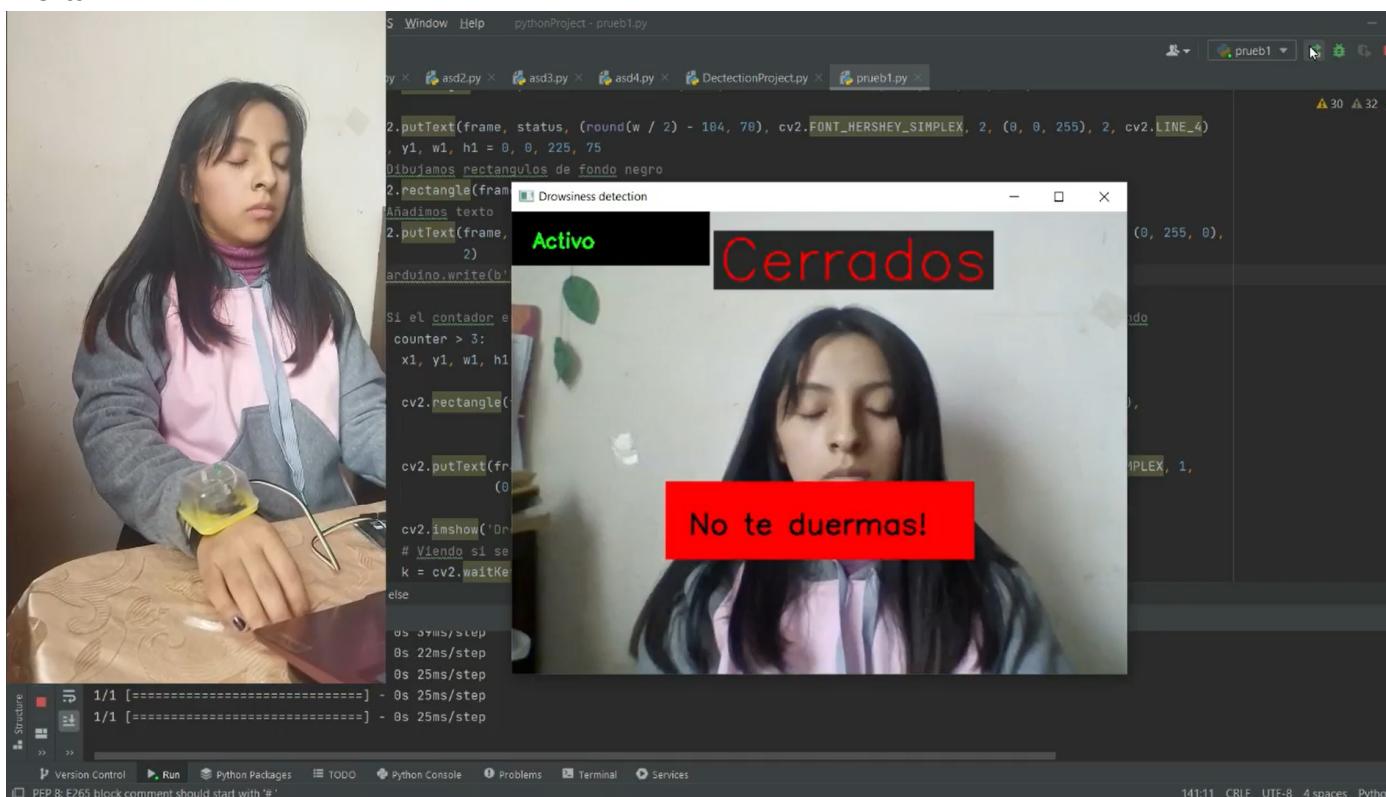
- Ojos abiertos



- Ojos cerrados



- Alerta



Tecnologías utilizadas

- Arduino mega
- 2 leds (rojo y verde)
- Motor de 5Vdc
- Cables

Frameworks utilizados

- Numpy

- Pandas
- OpenCV
- TensorFlow
- Keras
- face_recognition

Funcionalidades más importantes

- Extracción del área de los ojos para una clasificación más precisa
- Modelo de Redes Neuronales Convolucionales de varias capas con una precisión alta (>90%)
- Clasificación en tiempo real de somnolencia con etiquetas
- Adición de un prototipo que alerta de manera sensorial, auditiva y visual en el modelo en vivo en caso de somnolencia

Instalación

- Es importante tener instaladas, fuera de las librerías típicas de Ciencia de Datos (Numpy, Matplotlib, Pandas), librerías como OpenCV, TensorFlow, Keras, face_recognition, playsound (A través del comando pip)
- Se recomienda tener la cámara web activa en un entorno Desktop
- Se recomienda tener la GPU activa (En Colaboratory)
- En algunos sistemas operativos puede ser necesario instalar homebrew, cmake, dlib antes de instalar face_recognition, OpenCV y Playsound
- **Para la aplicación en vivo (Aplicacion del modelo final con la cámara web) se recomienda un IDE de escritorio (Pycharm, Spyder) corriendo un entorno virtual con todas las librerías necesarias**

Créditos

- Israel Algodón (25 %)
- Bruno Bustos (25 %)
- Tracy Loza (50 %)
- Funciones de detección adaptadas de: <https://towardsdatascience.com/drowsiness-detection-using-convolutional-neural-networks-face-recognition-and-tensorflow-56cdcf8315ad>
- GitHub del autor del artículo: <https://github.com/dustin-stew>
- Estructura de las Redes Neuronales Convolucionales inspiradas de:
https://github.com/EdwinTSalcedo/Bootcamp-Computer-Vision-for-the-SDGs/blob/master/notebooks/Laboratorio%2011_%20Redes%20Neuronales%20Convolucionales.ipynb
- Base para el uso de imágenes en Colab a través de OpenCV:
https://github.com/EdwinTSalcedo/Bootcamp-Computer-Vision-for-the-SDGs/blob/087b9e0474e5497a03c6c8073e79721d24da3ce1/notebooks/Laboratorio%201_%20Introductivo%C3%B3n%20a%20Colab,%20OpenCV%20y%20Numpy.ipynb
- Github del autor de los últimos 2 Notebooks: <https://github.com/EdwinTSalcedo>
- Métricas del modelo decididas a partir de: <https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec>
- Objetivos de desarrollo sostenible decididos a partir de la información proveída por Natalia Guerreros Lima en: <https://github.com/EdwinTSalcedo/Bootcamp-Computer-Vision-for-the-SDGs/blob/f2dd3b0eb4edb18d308a472bb5904c74b8eefe09/slides/ODS%20primera%20parte.pdf>

Secciones adicionales (Los resultados experimentales son más profundizados en el video y en nuestras imágenes importantes)

Trabajos Relacionados

La Detección de somnolencia en los conductores de vehículos automotores, fue un tema a tratar hace varios años, desde publicaciones científicas hasta proyectos y tesis para la obtención de grado académicos. Por este motivo es que fueron existiendo distintos métodos para la implementación de este tipo de proyecto para los usuarios que manejan vehículos.

Entre los avances más significativos para este tipo de detección, son los siguientes:

- **Sistemas por Visión de Computadora:** es una implementación en el campo de la inteligencia artificial basada en computadoras, el cual se subdivide en Visión Artificial y Redes Neuronales. Es un sistema no invasivo que aventaja por la sencillez de su implementación.

Actualmente los fabricantes de automóviles ya lanzan al mercado automóviles con esta tecnología para la detección de somnolencia como se muestra en la siguiente tabla:

Fabricante	Comportamiento que se estudia	Respuesta
Ford	Movimiento del vehículo en carretera	Alerta visual y auditiva
Lexus	Ojo del conductor	Alerta visual y auditiva, frenado del vehículo
Mercedes	Movimiento del vehículo en carretera	Alerta visual y auditiva
VW	Ojo del conductor	Alerta visual y auditiva
Volvo	Movimiento del vehículo en carretera	Alerta visual y auditiva

Este tipo de métodos trabajan primordialmente a partir del análisis de expresiones faciales y el movimiento de la cabeza, con modelamiento en 2D y 3D, posterior a esto se lo manda a un modelo de aprendizaje automático.

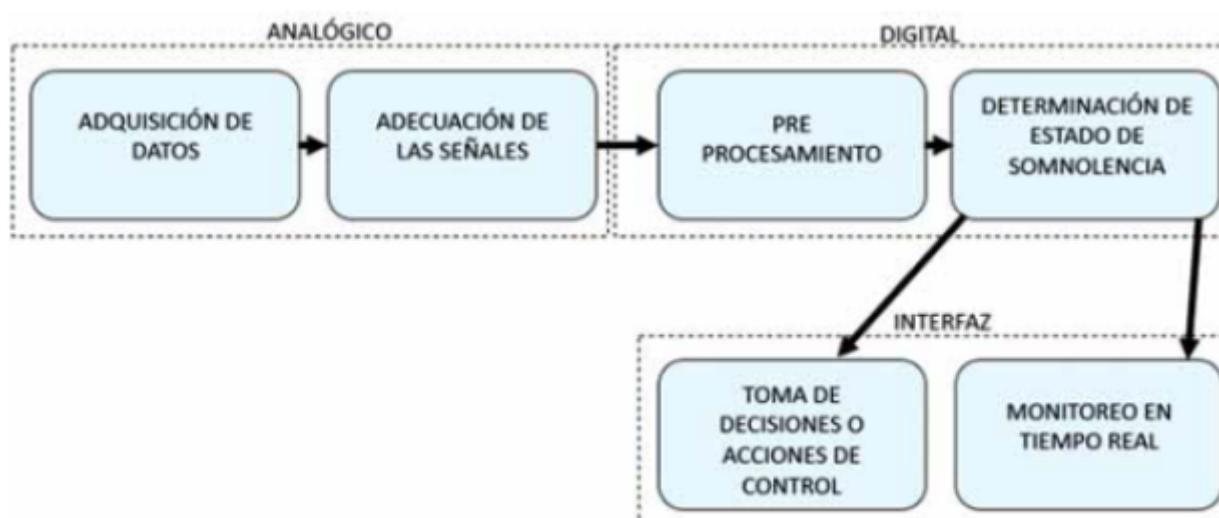


Figura 2. Diagrama de bloques básicos de un sistema activo de detección de somnolencia (Yeo, y otros, 2009).

Los sistemas electrónicos de los vehículos son una herramienta imprescindible para prevenir accidentes de tránsito, porque con ellos se puede hacer un monitoreo continuo y en tiempo real del vehículo, el conductor y sus acciones. Mas específico, el sistema de detección de sueño que se adapta al vehículo es una herramienta esencial para aplicaciones domésticas, de carga, industriales y otras de larga duración trabajando frente al volante, causando fatiga, puede conducir a la activación de diferentes fases de un sueño descontrolado, con tan lamentables consecuencias como la muerte

Dependiendo de la técnica utilizada, el sistema de detección de somnolencia del conductor monitorea continuamente el comportamiento físico del conductor, los patrones de movimiento del vehículo o las condiciones ambientales. Los métodos para detectar la somnolencia generalmente se dividen en tres categorías principales:

- Técnicas basadas en parámetros de comportamiento
- Técnicas basadas en parámetros vehiculares
- Técnicas basadas en parámetros fisiológicos

Técnicas Basadas en parámetros de Comportamiento

Los parámetros de comportamiento son un medio no invasivo para detectar la somnolencia. Estas técnicas miden la fatiga del conductor en función de parámetros de comportamiento del conductor como: Tasa de cierre de los ojos, parpadeo, posición de la cabeza, expresiones faciales, bostezos. La tasa de cierre de los ojos es uno de los indicadores más utilizados para detectar la somnolencia en función de las observaciones del estado de los ojos. PERCLOS es la tasa de cierre de los ojos durante un período de tiempo, y los resultados indican que los ojos están abiertos o cerrados. El sistema de detección basado en bostezos analiza los cambios en la geometría de la boca del conductor dormido, incluidos: mayor apertura de la boca, posición de los labios, etc. La tecnología basada en el comportamiento utiliza cámaras y tecnología de visión por computadora para extraer características del comportamiento.

Los problemas asociados a las medidas de comportamiento son los factores ambientales, como la iluminación, el brillo y las condiciones de la carretera, que influyen en la credibilidad y la precisión de la medición

Técnicas Basadas en parámetros de Vehiculares

Las técnicas basadas en parámetros del vehículo buscan detectar la fatiga del conductor en función de las características del vehículo, como cambios frecuentes de carril, fluctuaciones de velocidad del vehículo, ángulo de dirección y fuerza de agarre del volante. Estas mediciones requieren sensores en partes del vehículo, como el volante, el acelerador y los pedales de freno. Las señales generadas por estos sensores se utilizan para analizar la fatiga del conductor. El objetivo principal de estas tecnologías es observar los patrones de conducción y detectar la fatiga y el deterioro del rendimiento de conducción debido a la fatiga.

Sin embargo, la medición de la fatiga del conductor, según el movimiento del vehículo, es limitada porque los valores de medición pueden verse fácilmente afectados por factores externos como las características geométricas de las carreteras y las condiciones meteorológicas.

Técnicas Basadas en parámetros de Fisiológicos

Técnicas basadas en parámetros fisiológicos detectan la somnolencia en función del estado físico del conductor, como, por ejemplo: Frecuencia cardíaca, frecuencia del pulso, frecuencia respiratoria, temperatura corporal, etc. Estos parámetros biológicos son más fiables y precisos para detectar la fatiga porque están relacionados con lo que le sucede físicamente al conductor. El cansancio y la somnolencia modifican parámetros fisiológicos como la presión arterial, la frecuencia cardíaca y la disminución de la temperatura corporal. Un sistema de detección de somnolencia basado en la fisiología detecta estos cambios y alerta al conductor cuando está a punto de dormirse. La ventaja de este enfoque es alentar al conductor a descansar antes de que aparezcan los síntomas físicos de fatiga.

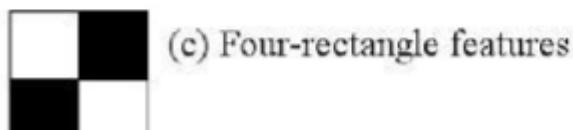
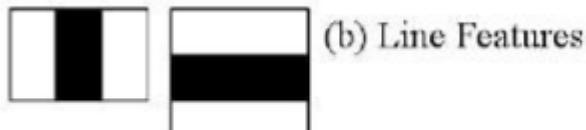
Estas medidas son invasivas, por lo que requieren la colocación de electrodos directamente en el cuerpo de los conductores. Este método a veces resulta irritante para el conductor, por lo que resulta difícil de aplicar.

Entre la recolección de datos, se puede dar con la implementación y agregación de datos previamente entrenados, para luego ser exportados a una cámara web que vaya procesando esta información, de esta manera arrojar resultados para conocer el estado o clasificación de los mismos. Por otro lado, pueden ser recolectados mediante la misma cámara recolectando imágenes cada cierto periodo de tiempo, primeramente, con la detección del rostro, identificación del iris del ojo y así poder determinar el estado de somnolencia, con estos resultados poder lanzar o ejecutar señales de alerta como ser, sonidos, luces o cualquier otro tipo de técnica que permita anunciar el nivel de somnolencia.

El desarrollo de distintos módulos que trabajan sobre visión artificial es dependiente de redes neuronales artificiales, con la implementación de patrones de Training para datos que muestran sobre ojos abiertos, cerrados, entreabiertos y cualquier otra posición o estado en el que se encuentre. Para este propósito se utilizan distintos entornos de desarrollo, distintos lenguajes de programación como C, C++, Python, Java. Distintas librerías como ser OpenCV, Tensorflow, que permitirán la implementación más eficiente de los mismos modelos que se requieran integrar a los distintos proyectos. Por parte de hardware se microcontroladores, comunicación mediante USB

Método Haarcascade de Viola y Jones

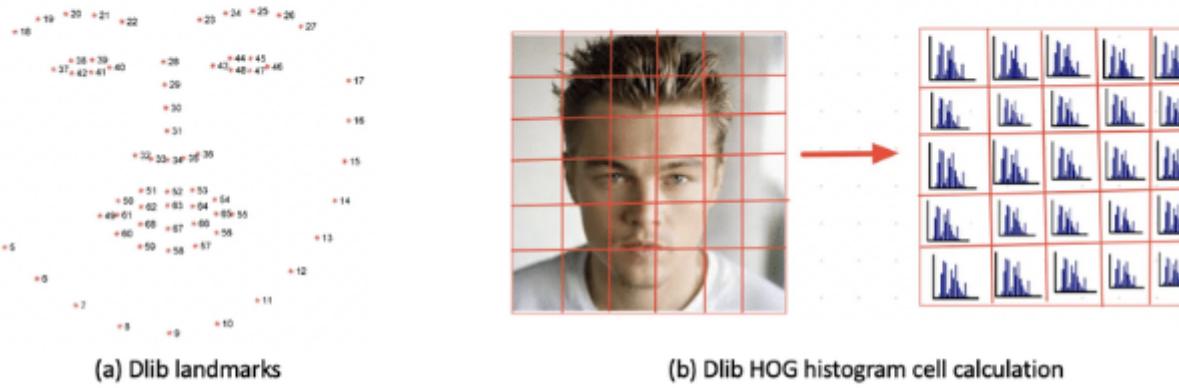
Paúl Viola y Michael J. Jones han desarrollado un algoritmo que consta de dos partes principales: un clasificador en cascada y un entrenador de clasificadores, que tienen costos computacionales muy bajos y garantizan una discriminación rápida. Las funciones de Haar que mediante clasificadores en cascada es una propuesta en su documento "Rapid Object Detection using a Boosted Cascade of Simple Features" en 2001, este es un método de aprendizaje automático que partir de un clasificador de imágenes positivas (imágenes con rostros) y negativas (imágenes sin rostros) para luego ser utilizado en la detección de objetos y/o características.



Al buscar un rostro y analizar la imagen, debe atravesar la misma por completo. Esto se hace obteniendo un pequeño marco de la imagen original. La mayoría de los fotogramas analizados corresponden a fragmentos de imagen en los que falta el rostro. Entonces, una forma rápida y fácil es analizar una pequeña ventana de la imagen, descartarla si no es una cara y no volverla a reprocesar. Esto optimiza el tiempo que lleva explorar posibles áreas del rostro a lo largo de la imagen.

Detector de rostro HOG en DLIB

Este es un modelo ampliamente utilizado para detectar 5 filtros: frontal, izquierda, derecha, rostro girado a la derecha y hacia la izquierda (Gupta, 2018). Para este clasificador se desarrolló un algoritmo de aprendizaje autónomo denominado Support Vector Machine (SVM), con esto se pretende realizar la predicción de la somnolencia para realizar la clasificación. Este descriptor luego se pasa al SVM entrenado, que lo clasifica como "cara" o "no cara", con parámetros relacionados con los párpados extraídos de los datos EOG recogidos en un simulador de conducción proporcionado por el proyecto SENSATION. El conjunto de datos se divide primero en tres niveles incrementales de somnolencia, y luego se realiza una prueba emparejada para identificar cómo se asocian los parámetros con el estado de somnolencia de los conductores. Con todas las características, se construye un modelo de detección de somnolencia SVM. Los resultados de la validación muestran que la precisión de la detección de la somnolencia es bastante alta, especialmente cuando los sujetos tienen mucho sueño.



Garcés M., Salgado J., Cruz J., Cañon W. (2015), Sistemas de Detección de Somnolencia en Conductores: inicio, desarrollo y futuro. <https://dialnet.unirioja.es/descarga/articulo/5432283.pdf> Gupta, V. (22 de octubre de 2018). Learn OpenCV. <https://www.learnopencv.com/face-detection-opencv-dlib-and-deep-learning-c-python/> Ramzan M, Ullah H, Mahmood S., Amina I., Ilyas M. (1 mayo 2019), A Survey on State-of-the-Art Drowsiness Detection Techniques.

<https://ieeexplore.ieee.org/abstract/document/8704263> Sanchez E., Limaquispe M., (junio 2018), Sistema de Detección de somnolencia mediante Inteligencia Artificial en Conductores de Vehículos para Alertar la Ocurrencia de Accidentes de Tránsito.

<https://repositorio.unh.edu.pe/bitstream/handle/UNH/2327/TESIS-2018-INGENIERIA%20ELECTRONICA-LIMAQUISPE%20MIGUEL.pdf?sequence=1&isAllowed=y>

Métodos

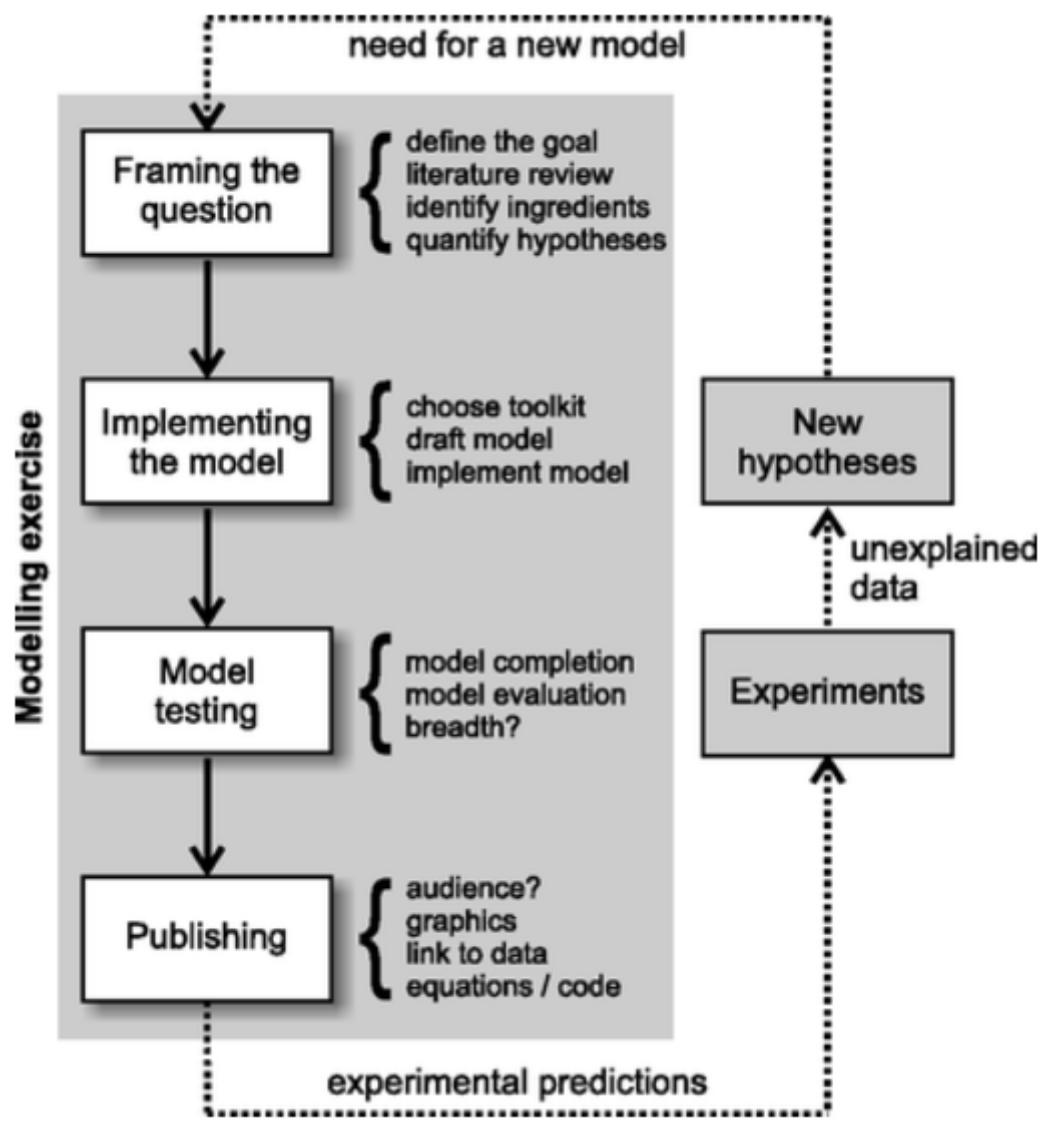
Los pasos a seguir durante este proyecto fueron inspirados en: [A How-to-Model Guide for Neuroscience](#), combinando con funcionalidades de un sistema completo aprendidas durante el presente [Bootcamp](#), ya que:

1. Se definió un objetivo
2. Se revisó la literatura
3. Se determinaron los ingredientes básicos de un posible modelo (Explorando las alternativas aprendidas durante el bootcamp)
4. Se definió una posible hipótesis (Proponiendo encontrar un modelo que supere el 90% de accuracy)

Dentro de la implementación del modelo:

5. Se seleccionaron las herramientas a ser implementadas, donde se consideraron enfoques alternativos
6. Se planificó el modelo, la arquitectura y la aplicación en vivo utilizados durante el proyecto
7. Se implementó el modelo en Colaboratory de Google y Jupyter Notebooks
8. Se completó el modelo modificando posibles hiperparámetros
9. Se testeó y evaluó el modelo en vivo alcanzando muy buenas métricas y combinandolo con un sistema de retroalimentación multisensorial

Para finalmente:



Logos



UNIVERSIDAD
CATÓLICA
BOLIVIANA

Licencia

The MIT License

Copyright (c) 2022 SplitVision

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.