

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Modelo de imágenes y reconocimiento facial para la detección temprana de sueño para conductores de forma preventiva por medio de la implementación de *Machine Learning*

Trabajo de graduación en modalidad de Tesis presentado por Guido Sebastián Padilla Aldana para optar al grado académico de Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

Guatemala,

2023

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería




Modelo de imágenes y reconocimiento facial para la detección temprana de sueño para conductores de forma preventiva por medio de la implementación de *Machine Learning*

Trabajo de graduación en modalidad de Tesis presentado por Guido Sebastián Padilla Aldana para optar al grado académico de Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

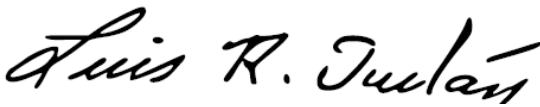
Guatemala,


2023


Vo.Bo.:

(f) 
Luis Furlán Roberto Collver

Tribunal Examinador:

(f) 
Luis Furlán Roberto Collver

(f) 
Douglas Leonel Barrios Gonzalez

(f) 
Luis Alberto Suriano Saravia

Fecha de aprobación: Guatemala, 7 de diciembre de 2023.

| | |
|--|-------------|
| Lista de figuras | VI |
| Lista de tablas | VII |
| Resumen | VIII |
| Abstract | IX |
| 1. Introducción | 1 |
| 2. Antecedentes | 2 |
| 3. Justificación | 4 |
| 4. Objetivos | 5 |
| 4.1. Objetivo general | 5 |
| 4.2. Objetivos específicos | 5 |
| 5. Alcance | 6 |
| 6. Marco teórico | 7 |
| 6.1. Fatiga y somnolencia | 7 |
| 6.1.1. ¿Qué es la fatiga? | 7 |
| 6.1.2. ¿Qué es la somnolencia? | 7 |
| 6.1.3. Implicaciones | 8 |
| 6.2. Rasgos faciales de somnolencia y fatiga | 8 |
| 6.3. Inteligencia artificial y <i>machine learning</i> | 9 |
| 6.3.1. Definición y conceptos clave de la inteligencia artificial y machine learning | 9 |
| 6.3.2. Modelos de <i>machine learning</i> | 9 |
| 6.3.3. Visión artificial | 11 |
| 6.3.4. Redes neuronales y convolucionales | 11 |
| 6.4. Modelaje y herramientas | 14 |
| 6.4.1. Preprocesamiento y preparación de imágenes | 14 |
| 6.4.2. Herramientas | 16 |

| | |
|---|-----------|
| 6.4.3. Extras | 17 |
| 7. Metodología | 18 |
| 7.1. Modelo de procesamiento de imágenes faciales para el reconocimiento del estado de los ojos humanos | 18 |
| 7.1.1. Recolección de datos | 18 |
| 7.1.2. Análisis de datos | 19 |
| 7.1.3. Aumento de datos | 23 |
| 7.1.4. División del conjunto de datos | 24 |
| 7.1.5. Desarrollo del modelo | 24 |
| 7.2. Uso final del modelo con fórmula de PERCLOS | 24 |
| 7.3. Servidor y aplicación para el consumo de la aplicación | 29 |
| 7.3.1. Servidor | 29 |
| 7.3.2. Aplicación | 30 |
| 8. Resultados | 31 |
| 8.1. Modelo de reconocimiento facial basada en el estado de los ojos humanos | 31 |
| 8.1.1. Desarrollo del modelo | 31 |
| 8.1.2. Resultados del modelo final | 33 |
| 8.1.3. Resultados en pruebas controladas | 34 |
| 8.1.4. Resultados en pruebas vehiculares | 40 |
| 8.1.5. Comprobación del modelo y fórmula | 45 |
| 9. Conclusiones | 52 |
| 10. Recomendaciones | 54 |
| 11. Bibliografía | 55 |
| 12. Anexos | 57 |

Lista de figuras

| | | |
|-----|--|----|
| 1. | PERCLOS fórmula | 8 |
| 2. | Proceso de la transformación de HOG, cada grupo de celdas es transformado a un histograma local, que como resultado final da un vector capaz de interpretar por otro algoritmo(Alonso, 2016) | 10 |
| 3. | Representación gráfica de hiperplano | 10 |
| 4. | Arquitectura de una red neuronal simple | 12 |
| 5. | Representación gráfica de la función ReLu (Artola Moreno, 2019) | 12 |
| 6. | Representación gráfica de <i>MaxPooling</i> (Artola Moreno, 2019) | 13 |
| 7. | Función sigmoideal (Alvarado, 2023) | 13 |
| 8. | Función softmax (Siddharth Sharma, 2020) | 14 |
| 9. | Ejemplos de sobreajuste y subajuste en base a los modelos basados en Regresión, Clasificación y Aprendizaje Profundo (Minhas, 2021) | 15 |
| 10. | Ejemplos de una imagen aumentada (Antreas Antoniou, 2018) | 15 |
| 11. | Ejemplo de imágenes pertenecientes al grupo de datos MRL | 18 |
| 12. | Primera parte de reporte | 19 |
| 13. | Segunda parte de reporte | 20 |
| 14. | Tercera parte de reporte | 20 |
| 15. | Primera parte del reporte del segundo grupo de datos completo | 21 |
| 16. | Segunda parte del reporte del segundo grupo de datos completo | 22 |
| 17. | Tercera parte del reporte del segundo grupo de datos completo | 22 |
| 18. | Proceso de análisis de vídeo | 25 |
| 19. | Función para obtención de <i>frames</i> de un vídeo | 26 |
| 20. | Función para la obtención del valor de PERCLOS basado en una lista de predicciones "I" | 26 |
| 21. | Función para clasificar basado en un <i>frame</i> | 27 |
| 22. | Función para predecir las predicciones dentro de un vídeo | 27 |
| 23. | Uso final de las funciones en conjunto | 28 |
| 24. | Set de vídeos de conductores bostezando (Abtahi et al., 2020) | 28 |
| 25. | Ejemplos de las pruebas de la base de datos de DROZY | 29 |
| 26. | Resultados de pruebas realizadas por AutoGluon | 31 |
| 27. | Resumen de modelo inicial | 32 |
| 28. | Muestra de foto nocturna de prueba | 33 |

| | | |
|-----|--|----|
| 29. | Gráfica de pestañeos (0-Cerrado, 1-Abierto, 2-No detección de Cara) sobre tiempo en segundos de los vídeos de hombres (Modelo de Autgluon) | 35 |
| 30. | Gráfica de pestañeos (0-Cerrado, 1-Abierto, 2-No detección de Cara) sobre tiempo en segundos de los vídeos de mujeres | 36 |
| 31. | Gráfica de valor promedio de PERCLOS sobre frames válidos (frames catalogados con 1 o 0) de videos de mujeres | 37 |
| 32. | Gráfica de pestañeos (0-Cerrado, 1-Abierto, 2-No detección de Cara) sobre tiempo en segundos de los vídeos de hombres | 38 |
| 33. | Gráfica de valor promedio de PERCLOS sobre frames válidos (frames catalogados con 1 o 0) de vídeos de hombres | 39 |
| 34. | Giro de la cabeza hacia un lado del sujeto de prueba en ambiente controlado | 40 |
| 35. | Escenario de prueba de noche | 40 |
| 36. | Estado de los ojos a lo largo del tiempo del primer sujeto en 40 segundos (0-Cerrados, 1-Abiertos, 2-Cara no reconocida) | 41 |
| 37. | valor de PERCLOS sobre frames válidos (5 frames por cálculo) | 41 |
| 38. | Visualización de pestañeo prueba de noche | 42 |
| 39. | Predicciones de ojos del sujeto de prueba donde el movimiento es constante y desestabilizado | 42 |
| 40. | Perclos del sujeto de prueba donde el movimiento es constante y desestabilizado | 43 |
| 41. | Visualización de pestañeo de ojos del sujeto de prueba donde el movimiento es constante y desestabilizado | 43 |
| 42. | Predicciones de ojos del sujeto de prueba donde este se siente fatigado | 44 |
| 43. | Perclos del sujeto de prueba donde este se siente fatigado | 44 |
| 44. | Escenario de sujeto con indicios de fatiga | 45 |
| 45. | Frecuencia de PERCLOS con las 3 etapas de la prueba de DROZY siendo 3 el estado mayor de privación de sueño | 45 |
| 46. | Señales de polimnografía en base al sujeto 4 de la base de datos DROZY | 46 |
| 47. | Tiempos de reaccion en base al sujeto 4 de la base de datos DROZY | 46 |
| 48. | Mapa de correlación entre las variables de los sujetos de prueba de DROZY, agrupadas por nivel de la prueba y segundos dentro de cada fase | 47 |
| 49. | Primera parte de relaciones gráficas de DROZY | 48 |
| 50. | Segunda parte de relaciones gráficas de DROZY | 49 |
| 51. | Tercera parte de relaciones gráficas de DROZY | 50 |
| 52. | Cuarta parte de relaciones gráficas de DROZY | 51 |

Lista de tablas

| | | |
|----|---|----|
| 1. | Parámetros de la aumentación de datos | 24 |
| 2. | Matriz de confusión de red neuronal inicial | 32 |
| 3. | Resultados del modelo final con respecto a datos de prueba no pertenecientes a datos de entrenamiento, siendo el identificador C (conjunto completo) y P (conjunto de 4000 datos) . | 33 |
| 4. | Matriz de confusión de SVM C (Conjunto completo) con respecto a datos de entrenamiento . | 34 |
| 5. | Resultados del modelo final con respecto a datos de entrenamiento, SVM C (Conjunto Completo) | 34 |
| 6. | Matriz de confusión de SVM C (Conjunto completo) con respecto a datos fuera del conjunto de entrenamiento | 34 |
| 7. | Métricas SVM C (Conjunto completo) con respecto a datos fuera del conjunto de entrenamiento | 34 |

El proyecto realizado propone un modelo para detectar somnolencia, a través de imágenes completas faciales, enfocado en el estado de los ojos. Este modelo a través de algoritmos de *machine learning* reconoce los parpadeos que se realizan, de manera que se analizan para determinar si durante cinco fotogramas, la cerradura de los ojos es prolongada o no, indicando indicios de fatiga, por medio de la fórmula investigada de PERCLOS. Se tomó en consideración la posibilidad de utilizar luz infrarroja para mantener la detección en la noche. Como resultados finales se obtuvo un modelo de SVM en conjunto con el algoritmo de HOG, con una precisión en entrenamiento, validación y prueba de 0.98, 0.94 y 0.86 respectivamente.

El desarrollo del modelo también señala que el modelo puede fallar por causas externas, lo que indica que este debe ser utilizado como una señal de indicios de somnolencia en conjunto con otras señales. Se propone la mejora del mismo a través de futuros trabajos.

Palabras clave: inteligencia artificial, somnolencia, fatiga, algoritmos

Abstract

The project proposes a model to detect drowsiness, through complete facial images, focused on the state of the eyes. This model, through extitmachine learning algorithms, recognizes the blinks that are made, so that they are analyzed to determine if for five frames, the lock of the eyes is prolonged or not, indicating signs of fatigue, by means of the investigated formula of PERCLOS. The possibility of using infrared light to maintain detection at night was considered. As final results, an SVM model was obtained in conjunction with the HOG algorithm, with an accuracy in training, validation and testing of 0.98, 0.94 and 0.86 respectively.

The development of the model also points out that the model can fail due to external causes, indicating that it should be used as a signal of signs of sleepiness in conjunction with other signals. It is proposed to improve it through future work.

keywords: artificial intelligence, drowsiness, fatigue, algorithms

CAPÍTULO 1

Introducción

Los viajes largos de vehículos pesados son comunes para el transporte con distintos fines, desde materiales a vehículos. Los recorridos de los viajes tienden a ser largos y pesados para el conductor designado. Los accidentes son un tema común en la vía vehicular y son causados por diversos factores: imprudencias, mala infraestructura en los recorridos, señalizaciones, fatiga, etc.

La fatiga en viajes largos es un tema delicado ya que tiende a provocar los peores accidentes. La fatiga causa que el conductor pierda completamente el control del vehículo, y en muchos de los casos resulta en la pérdida de vidas. Esto puede ser prevenido con anticipación si se logra supervisar el estado del conductor, de modo que se pueda prevenir con anticipación o en algunos casos programar los viajes a posterior.

La inteligencia artificial (IA) es un tema que sigue creciendo en popularidad, y en uso, de forma elevada. La IA ofrece la capacidad de realizar tareas y trabajos, que son cansados y difíciles de resolver. Es por ello que, por medio de la IA se ha utilizado para ayudar a facilitar el trabajo en muchos escenarios, desde tareas de baja a alta importancia. En relación a los accidentes vehiculares, se han realizado pruebas exitosas con cámaras, pero bajo simulaciones controladas (Rondon L, 2013).

Se propuso la utilización de modelos de Inteligencia Artificial para detectar si un sujeto tiene indicios de fatiga o no, fundamentado por una investigación donde se comparan varios métodos, incluyendo la de utilizar IA junto con PERCLOS (Jay D. Fuletra, 2013). Específicamente la investigación se enfoca en el reconocimiento facial enfocado en los ojos y sus pestaños. Se realizaron y comprobaron estudios de la capacidad de la inteligencia artificial en relación a esta problemática, y los resultados apuntan a que, en efecto, esta técnica ayuda a detectar indicios de somnolencia (Jay D. Fuletra, 2013).

En la presente investigación se propuso de forma inicial el uso de una red neuronal convolucional, puesto a prueba bajo varias circunstancias de entrenamiento y prueba, luego de realizar descartes y modificaciones se propuso otros dos modelos los cuales son SVM (*Support Vector Machines*) y redes neuronales convolucionales prefabricadas por librerías como Autogluon, se pudo observar que luego de los resultados obtenidos, se obtienen fallas y mejoras, que ayudaron a descartar y postular el mejor modelo, que como resultado se decidió implementar la solución en base a un modelo SVM junto con el algoritmo de HOG, que mediante el cálculo de PERCLOS logró determinar indicios de somnolencia o fatiga de un sujeto.

Los accidentes causados por conductores dentro de las carreteras son un problema significativo en todo el mundo. Según un estudio realizado por la universidad de San Carlos de Guatemala en 2018 [1], se indicaba que los accidentes de tránsito eran la segunda causa de muerte, de forma diaria ocurrían 17 accidentes, el 47 por ciento de estos resultaban en fallecimientos. (Heredia et al., s.f.)

Otro estudio realizado por la universidad Nacional de San Antonio Abad del Cusco (Rondon L, 2013), señalaba que en base a una encuesta realizada a 238 personas, quienes eran conductores de buses interprovinciales, que la fatiga era la causa más común en accidentes en carreteras, los resultados finales fueron los siguientes:

- 55 % fatiga o cansancio
- 24 % imprudencia del chofer
- 5 % exceso de velocidad
- 3 % falla mecánica
- 1 % mal estado de las pistas
- 1 % negligencia de las empresas de transporte

En relación al problema, la confianza se ve afectada, un estudio realizado por el Congreso Latinoamericano y IV Congreso Peruano de Ergonomía Lima (Altamirano I, 2016), señalaban que esta se veía afectada debido a la falta de reconocimiento de cansancio por los conductores y la presión creada a partir de sistemas que puedan controlar la fatiga. Por lo cual propone que la creación de un sistema o modelo que identifique la fatiga, no asegura su fiabilidad, basado en nuevas técnicas por parte de los conductores para ignorar estos. A pesar de la desconfianza, se han realizado sistemas que difieren en enfoques y logros.

Comúnmente el factor más utilizado y el cual fue tomado a consideración, que es mencionado por un estudio realizado en Bharati Vidyapeeth's College of Engineering, una universidad en india, en el

año 2013 (Jay D. Fuletra, 2013), es la identificación del estado de los ojos los cuales conforman parte del reconocimiento facial. Se menciona dentro del mismo estudio que el uso de varias métricas hace que el sistema tienda a ser más efectivo, por ejemplo: factores relacionados a la cara, signos vitales, telemetría del carro, etc. También se toma en cuenta que los sistemas se clasifican en dos tipos: intrusivos y no intrusivos. Lo que se busca con el desarrollo de este modelo es la comodidad del usuario por medio de obtener resultados sin perturbar su concentración.

La inteligencia artificial ha aportado un avance significativo dentro del área de la implementación de modelos que brinden una seguridad al conductor ante la fatiga. Esto ayuda a prevenir accidentes. La implementación de este modelo puede brindar una gran ayuda a las compañías, además de salvar vidas, sin mencionar que los costos serán mínimos debido a la base planteada.

¿Por Qué La Propuesta De Un Modelo Es Viable Para Resolver El Problema?

1. Bajo un estudio realizado por la universidad de San Carlos de Guatemala, los accidentes en el año 2018 (Heredia et al., s.f.), eran la segunda causa más común de mortalidad. El modelo presente es creado para el inicio de la solución de un sistema a futuro que pueda prevenir dichos accidentes.
2. Gracias a los estudios realizados en Bharati Vidyapeeth's College of Engineering, una universidad en india, en el año 2013 (Jay D. Fuletra, 2013), se señala que los ojos, que pertenecen a los rasgos faciales, son una solución planteada y efectiva para determinar indicios de fatiga, además de poder ser unificada con otras señales para aumentar la efectividad.

4.1. Objetivo general

1. Desarrollar un modelo de reconocimiento facial basado en *machine learning* que detecte los pestañeos realizados por los ojos, que con ayuda de la fórmula PERCLOS, puedan dar señal de indicios de fatiga a conductores de manera preventiva.

4.2. Objetivos específicos

1. Entrenar el modelo de reconocimiento facial con suficientes datos de modo que este sea efectivo, a través de imágenes y videos recolectados.
2. Evidenciar que el modelo de detección facial enfocado en el estado de los ojos puede ser usado como una señal de indicios de somnolencia o fatiga.
3. Tomar en consideración casos donde la cámara no pueda determinar los factores, proponiendo opciones como luz infrarroja en caso de un viaje nocturno.
4. Hacer uso de técnicas de ciencia de datos para lograr realizar el modelo de reconocimiento facial.

Se desarrolló un modelo que da indicaciones de indicios de fatiga o somnolencia en conductores de forma preventiva. Este puede fallar si se toma como única métrica. Sin embargo, en conjunto con otras señales éste puede llegar a ser útil. Por lo cual este modelo puede ser utilizado y refinado en futuras investigaciones. Se puede implementar un sistema más completo donde se tomen en consideración señales como los signos vitales, dirección vehicular, postura de sujeto, etc. Los sistemas en los que se implementen este conjunto de señales, pueden llegar a dar resultados más efectivos. También dependiendo del sistema se puede disponer o no de más recursos afectando a las medidas realizadas de manera positiva y directamente proporcional a los recursos disponibles.

6.1. Fatiga y somnolencia

6.1.1. ¿Qué es la fatiga?

En un estudio realizado en la Universidad Privada Rafael Belloso Chacín (Bracho-Paz, 2020) señala que, la fatiga laboral es un proceso que conduce a la insatisfacción en el trabajo, la disminución del rendimiento y la aparición de síntomas físicos y psicológicos. Se relaciona con el síndrome de *burnout*, un estado de agotamiento físico y mental que se caracteriza por la disminución del interés en el trabajo, la sensación de despersonalización y la baja autoestima. El *burnout* es común en trabajos repetitivos y extensos, como el manejo de vehículos pesados a grandes distancias.

Esta misma investigación referenciada (Bracho-Paz, 2020) menciona también que, la fatiga se clasifica en tres tipos: física, mental y combinada. La fatiga física se caracteriza por la disminución de las capacidades físicas, visuales, auditivas y musculares. La fatiga mental se produce por actividades repetitivas y de uso constante del cerebro. La fatiga combinada es la combinación de la fatiga física y mental en el trabajo.

6.1.2. ¿Qué es la somnolencia?

La somnolencia es un estado de decaimiento y adormecimiento que puede ser causado por la privación del sueño, enfermedades, medicamentos o cambios de zona horaria, esto mencionado por el acta médica peruana, revista científica (Edmundo Rosales Mayor, 2010). La somnolencia puede tener efectos negativos, como la pérdida de concentración y la relajación muscular. Esta puede clasificarse en dos tipos: objetiva y subjetiva. La somnolencia objetiva se basa en la tendencia del individuo a quedarse dormido. La somnolencia subjetiva se basa en la percepción subjetiva de la necesidad de dormir.

6.1.3. Implicaciones

La fatiga y la somnolencia, en sus diferentes tipos, pueden tener implicaciones graves en el ámbito laboral, especialmente en el caso de los conductores de vehículos pesados u ordinarios en viajes largos. Los accidentes relacionados con la fatiga son un serio problema de seguridad vial, con consecuencias devastadoras para los conductores, los otros usuarios de la carretera y la sociedad en general.

6.2. Rasgos faciales de somnolencia y fatiga

La fatiga y somnolencia se manifiestan en los ojos humanos a través de cambios en la orientación de la mirada y en el patrón de parpadeo. El parpadeo es un reflejo fisiológico que cumple la función de renovar y distribuir la película lagrimal de los ojos. Se clasifica en voluntario y no voluntario (Wiechers, 2009). El parpadeo no voluntario es causado por estímulos exteriores, como el dolor o un objeto cercano al ojo. Sin embargo, también puede ser causado por la somnolencia, que indica un estado cercano al sueño. El parpadeo voluntario es realizado por el sujeto a voluntad.

Los pestañeos son un indicador de somnolencia en las personas adultas. Según estudios realizados en el año 2018 por la universidad de Kristen Maranatha en Indonesia (Pasaribu, 2018), el rango normal de pestañeos por minuto es de 10 a 20. Este rango puede disminuir cuando una persona está somnolienta o fatigada. Por lo tanto, un número reducido de pestañeos puede ser un signo de somnolencia.

La métrica PERCLOS (Percentage of the Time Eyelids are Closed) es un indicador de somnolencia basado en la frecuencia de pestañeo, como se puede observar en la Figura 1 (George, 2012). Para calcularlo, se capturan fotogramas a través de un tiempo prolongado. De modo que el valor es el porcentaje de la cantidad de fotogramas donde el ojo está cerrado, dividido el número total de fotogramas. La fórmula de PERCLOS es la siguiente:

$$PERCLOS (\%) = \frac{\text{Suma total de fotogramas donde el ojo esta cerrado}}{\text{Suma de fotogramas total}} \times 100$$

Figura 1. PERCLOS fórmula

Un estudio realizado en el año 2018 en la universidad de tecnológica de india kharagpur (Junaedi & Akbar, 2018), encontró que PERCLOS es una señal correcta que da indicios de fatiga sobre un sujeto. En este estudio, se probó el valor de PERCLOS con varios porcentajes, y se encontró que un valor de 60 % o más es un buen indicador de somnolencia .

6.3. Inteligencia artificial y *machine learning*

6.3.1. Definición y conceptos clave de la inteligencia artificial y machine learning

Inteligencia artificial (IA)

La inteligencia artificial (IA) es la capacidad de las computadoras de procesar y analizar grandes volúmenes de datos, que mediante algoritmos puedan utilizar y aprender de estos datos, de manera que realicen tareas que requieran de inteligencia humana (ROUHIAINEN, 2018) . La IA puede cometer menos errores que los humanos, ya que no se ve afectada por la fatiga o la emoción. Además, la IA tiene una capacidad de análisis mayor que los humanos, siempre y cuando disponga de la memoria y el poder computacional suficiente.

Machine learning (ML)

El concepto de machine learning es un enfoque dentro de la inteligencia artificial, que se basa en la capacidad de las computadoras de poder aprender a partir de un entrenamiento, sin la necesidad de estar programadas específicamente para esa tarea (ROUHIAINEN, 2018). Existen 3 tipos de aprendizaje los cuales son los siguientes:

1. Aprendizaje supervisado: Es donde previamente los datos son etiquetados por un ser humano, es decir, existe una intervención previa humana antes del aprendizaje, de modo que se necesita una retroalimentación (ROUHIAINEN, 2018).
2. Aprendizaje no supervisado: Aquí no se requiere de una intervención humana para clasificar los datos por lo cual se realiza de manera autónoma (ROUHIAINEN, 2018).
3. Aprendizaje de refuerzo: Por último, este aprendizaje realiza su desarrollo en base a un refuerzo positivo o negativo, lo que ayuda a tomar decisión a la máquina (ROUHIAINEN, 2018).

6.3.2. Modelos de *machine learning*

Técnicas de extracción de características HOG (*Histogram of Oriented Gradients*)

El algoritmo HOG es un método de extracción de características para imágenes que se basa en la orientación de los gradientes en regiones locales de la imagen (Alonso, 2016). El algoritmo divide la imagen en celdas rectangulares, y para cada celda calcula un histograma de la orientación de los gradientes de los píxeles que la componen.

El histograma se calcula dividiendo el espacio de orientaciones en un número determinado de bins, y contando el número de píxeles en cada bin. Un bin es un compartimento en el que se clasifican los datos. En el caso del algoritmo HOG, los bins se utilizan para clasificar las orientaciones de los gradientes.

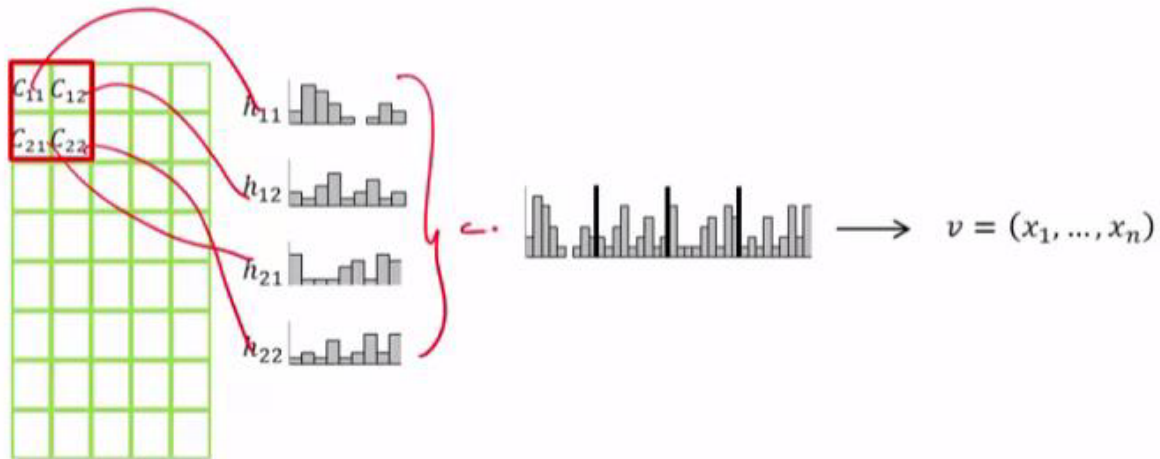


Figura 2. Proceso de la transformación de HOG, cada grupo de celdas es transformado a un histograma local, que como resultado final da un vector capaz de interpretar por otro algoritmo(Alonso, 2016)

SVM (*Support Vector Machines*)

Las máquinas de soporte de vectores (SVM) son un algoritmo de aprendizaje automático supervisado, que se basa en la clasificación de dos clases representadas por puntos, que por medio de un hiperplano son separadas (Alonso, 2016). El hiperplano es una superficie en un espacio n-dimensional que maximiza el margen entre los dos conjuntos de puntos. El margen es la distancia entre el hiperplano y los puntos más cercanos de cada clase, el hiperplano puede ser representado por varias funciones, siendo lineal la más común y en el caso del proyecto radial.

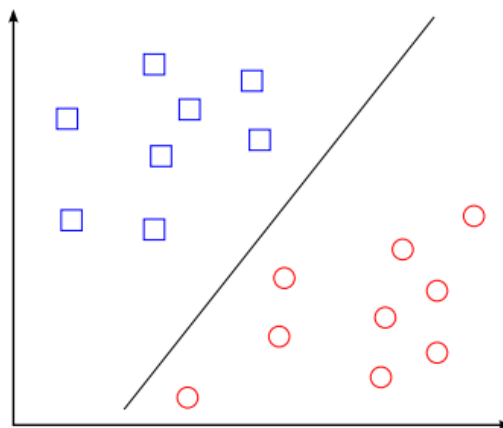


Figura 3. Representación gráfica de hiperplano

6.3.3. Visión artificial

El campo enfocado al análisis de imágenes por computadora, es conocido como visión artificial. La visión artificial es la unión de conceptos, algoritmos y técnicas que ayudan a procesar una imagen en información con valor para una computadora (Edmundo Rosales Mayor, 2010). Esta información puede utilizarse para realizar tareas como la clasificación de imágenes o, en el caso de este proyecto, el reconocimiento de los ojos y su relación con la somnolencia.

6.3.4. Redes neuronales y convolucionales

En relación a la inteligencia artificial y la visión artificial, las redes neuronales artificiales (ANN por sus siglas en inglés) son una tecnología que trata de replicar el funcionamiento del cerebro humano (Olabe, 2013). El cerebro humano está compuesto por neuronas que están conectadas entre sí para formar redes. Las redes neuronales artificiales simulan esta estructura, con cada neurona artificial llamadas PE (*Process Element*) representando una neurona biológica.

Las PE están conectadas entre sí por capas. Cada capa recibe información de la capa anterior y la procesa para generar una salida. La salida de una capa se puede utilizar como entrada para la siguiente capa (Olabe, 2013). De esta manera, la información se propaga a través de la red hasta que se llega a la salida final. La salida final de una red neuronal artificial puede ser utilizada para realizar una variedad de tareas, incluyendo el reconocimiento de objetos, la clasificación de imágenes y la predicción de resultados.

En base a los conceptos descritos, surgen las redes neuronales convolucionales (CNN por sus siglas en inglés). Estas redes neuronales son más especializadas en identificar características dentro de las entradas o input mencionados, de manera que en primera instancia se identifican propiedades o formas básicas que, al pasar por las capas ocultas, se desarrollan de tal manera que al final pueden reconocer formas o propiedades más complejas, como podría ser un rostro, ojo, boca, etc (Artola Moreno, 2019).

Las redes neuronales convolucionales se relacionan con las imágenes debido a sus convoluciones, que son el proceso en el que se toman grupos de píxeles cercanos de una imagen y se operan matemáticamente en contra de un núcleo, matriz que contiene píxeles de filtro (Artola Moreno, 2019). Esto produce una nueva matriz de salida, conocida como matriz de activación. De esta manera, se pueden identificar propiedades y, a su vez, producir una salida capaz de interpretarse según el propósito de la red.

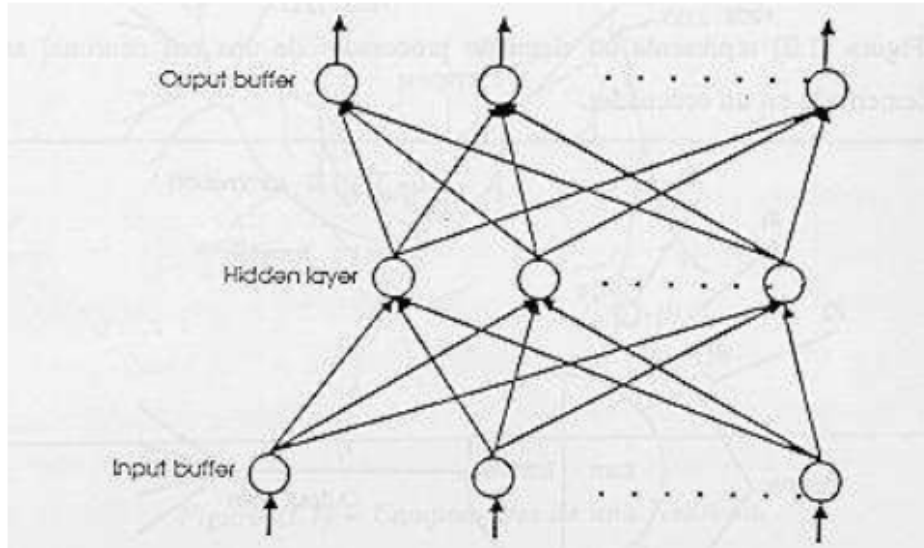


Figura 4. *Arquitectura de una red neuronal simple*

ReLU (*Rectified Linear Unit*)

La unidad lineal rectificadora por sus siglas en inglés (ReLU), es comúnmente la función más utilizada en las redes neuronales, esta retorna un valor 0 si recibe una entrada negativa, de caso contrario retorna el mismo valor de entrada (Artola Moreno, 2019).

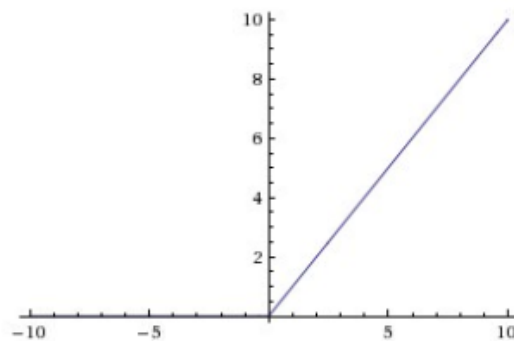


Figura 5. *Representación gráfica de la función ReLU (Artola Moreno, 2019)*

MaxPooling

Función que extrae las propiedades más valiosas dentro de una matriz de salida, de manera que se divide en grupos la matriz, donde se toma el valor máximo representado por números dentro de cada grupo, para obtener como resultado una matriz con los valores máximos. Ayudando a obtener las características más importantes de una imagen (Artola Moreno, 2019).

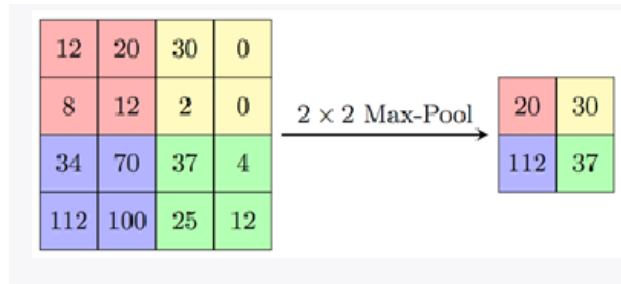


Figura 6. Representación gráfica de MaxPooling (Artola Moreno, 2019)

Dropout o dilución

La función de *dropout* o dilución como bien conocida en español, es una técnica de regularización que permite controlar el sobreajuste. Esta técnica funciona desactivando un número de neuronas en la red neuronal de manera aleatoria (Artola Moreno, 2019). Esto obliga a las neuronas restantes a aprender a trabajar de manera independiente, lo que ayuda a evitar que la red se ajuste demasiado a los datos de entrenamiento. El valor de la probabilidad determina qué porcentaje de neuronas se desactivarán. Los valores cercanos a 0 desactivan menos neuronas, mientras que los valores cercanos a 1 desactivan más neuronas. .

Sigmoide

La función de sigmoide hace referencia a la siguiente ecuación $f'(x) = 1 - \text{sigmoide}(x)$, en donde la salida será entre el rango de 0 a 1 (Siddharth Sharma, 2020), siendo este el resultado final a partir de una entrada, permite realizar una predicción gracias a su salida, ya que se sabe que una probabilidad está en este rango.

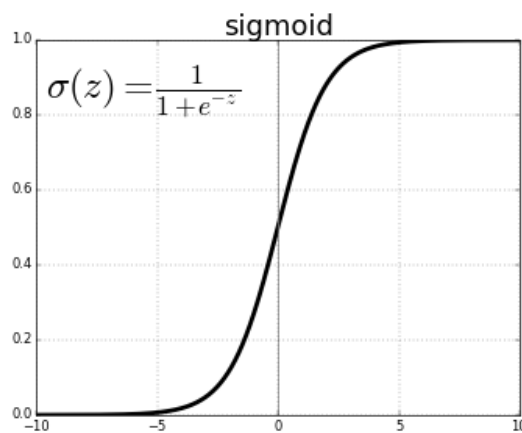


Figura 7. Función sigmoideal (Alvarado, 2023)

Softmax (función exponencial normalizada)

En cuanto a la función de *softmax*, es una combinación de funciones sigmoideas (Siddharth Sharma, 2020), es decir que se trata de clasificar la probabilidad de múltiples clases y es representada por la siguiente ecuación similar a la sigmoidea.

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{for } j = 1, \dots, K.$$

Figura 8. Función softmax (Siddharth Sharma, 2020)

6.4. Modelaje y herramientas

6.4.1. Preprocesamiento y preparación de imágenes

En base a esto, es importante entender conceptos relacionados al procesamiento de imágenes y entrenamiento de modelos de *machine learning*. Todos los procesos empiezan por la recolección de datos, se necesita de una cantidad inicial de datos significativa para poder entrenar un modelo. Además, se requiere diversidad en los datos, ya que al no tenerla, resulta en modelos con un error llamado sobreajuste.

El sobreajuste consiste en entrenar el modelo específicamente para los datos de entrenamiento de modo que no esté preparado para nuevos datos. Por su parte, el subajuste ocurre cuando el modelo no puede encontrar un patrón o tendencia relacionado a los datos, lo que resulta en un desempeño bajo con otros datos. (HAIDER KHALAF JABBAR, 2015)

La preparación de imágenes también es importante en el entrenamiento de modelos. Los modelos se preparan con un grupo de datos (imágenes) de entrenamiento que contiene características particulares (ROUHIAINEN, 2018). Al momento de predecir en base a estos datos, es importante definir las mismas características al valor a predecir, en este caso a la imagen que será procesada para predecir.

Por ejemplo, si los datos de entrenamiento son en blanco y negro, se tendrá que pasar por un preprocesamiento la imagen a predecir para poseer las mismas características que los datos de entrenamiento. En ese caso, se realizaría una transformación de colores a blanco y negro. Además, se pueden realizar preparaciones básicas como la normalización de imágenes, que consiste en llevar a un estándar de rango de valores para trabajar con una misma capacidad de comprensión.

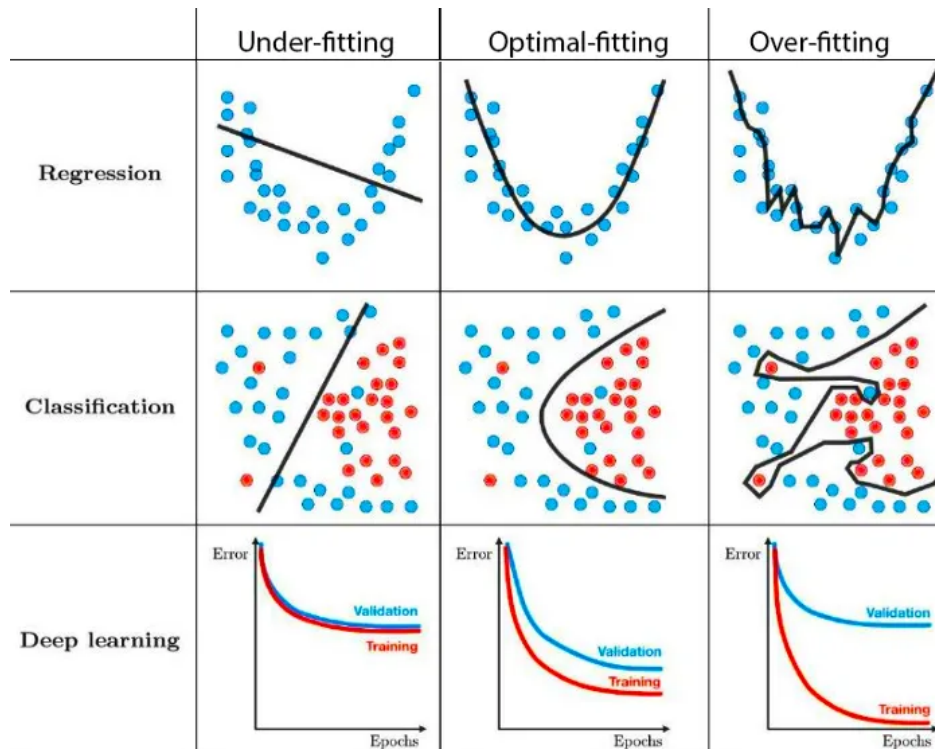


Figura 9. Ejemplos de sobreajuste y subajuste en base a los modelos basados en Regresión, Clasificación y Aprendizaje Profundo (Minhas, 2021)

Cuando hay insuficiencia de imágenes o un desbalance en el tipo de las mismas, en algunos casos, se puede utilizar la técnica de aumento de datos. Esta técnica consiste en tomar el grupo inicial de datos y aplicar técnicas de transformación, como rotaciones, aumento de tamaño, traslados y cualquier transformación de matriz posible (Antreas Antoniou, 2018). Esto ayuda a obtener una cantidad mayor de datos inicial, que permite que los modelos se adapten a estos datos nuevos y eviten el sobreajuste, así como la capacitación a casos nuevos.



Figura 10. Ejemplos de una imagen aumentada (Antreas Antoniou, 2018)

6.4.2. Herramientas

Tensorflow (versión: 2.9.1)

Esta es una librería desarrollada por Google, de código abierto, para desarrollar e implementar soluciones de inteligencia artificial a través de soluciones simples y eficientes, principalmente para el desarrollo de redes neuronales (Google Research Brain Team, 2023).

Keras (versión: 2.9.0)

Es otra librería de código abierto en python desarrollada por François Chollet, enfocada en redes neuronales que se ejecuta por encima de tensorflow permitiendo así conservar la simpleza e intuitividad en su uso (Keras Contributors, 2023).

Sklearn (versión: 1.2.2)

Es una librería de código abierto, desarrollada en python, por David Cournapeau que permite el acceso a varios algoritmos disponibles de inteligencia artificial, como las SVM y el algoritmo de HOG (Scikit-learn Developers, 2023).

Dlib (versión: 19.24.2)

Es una librería de código abierto, multiplataforma, desarrollada en c++ para la solución de problemas mediante machine learning. En el caso del proyecto esta librería provee la capacidad de poder tomar áreas de interés dentro de una imagen, para ser específicos la cara y sus ojos que es lo que estaremos analizando, y todo esto por medio de funciones simples y robustas (Davis E. King, 2022).

Autogluon (versión: 0.8.2)

Es una librería de python que permite la exploración de diversos modelos con un conjunto de datos logrando encontrar el que se mejor adapte al problema dado Brinda un análisis detallado para tomar decisiones en base a los resultados.(Community, s.f.).

Pandas-profiling (versión: 3.6.6)

Librería de python que permite el análisis de un grupo de datos, por medio de visualización de gráficas, patrones, métricas, etc. (Inc, 2023)

6.4.3. Extras

Flask (versión: 3.0.0)

Es un *framework* de python que permite desarrollar soluciones web de manera minimalista, en la mayoría de los casos para realizar microservicios, en el caso del proyecto se utilizó esta tecnología para desarrollar un servidor minimalista de bajo consumo de recursos, debido a que se utilizó en un celular, con el fin de ser el puente entre el modelo y la interfaz gráfica, permitiendo enfocarnos en los modelos y sus entornos y abstrayendo estos a un uso por medio de consumo en servicio directo a nuestra interfaz gráfica realizada (Team, 2023).

Kivy (versión: 2.2.1)

Framework de python de código abierto que permite crear aplicaciones multiplataforma desde aplicaciones móviles hasta aplicaciones de escritorio basadas en windows, macOS, linux. El uso del proyecto fue de aplicación móvil para lanzar una aplicación que corriera en segundo plano y contuviera al servidor de modo que escuchará peticiones (Contributors, 2011).

7.1. Modelo de procesamiento de imágenes faciales para el reconocimiento del estado de los ojos humanos

7.1.1. Recolección de datos

Como primer paso se investigó para obtener un conjunto de datos inicial que fuera completo y de licencia gratis para su uso. Se encontró uno en Kaggle (Lab, 2021b) que contiene imágenes tomadas con luz infrarroja en diversas condiciones de luz, diferentes sujetos, diferentes dispositivos realizando la toma. Este está conformado por 4000 imágenes, divididas de forma balanceada en 2000 casos de ojos abiertos y 2000 de ojos cerrados.

Existen varias versiones de este conjunto de datos, variando de 2000 a 84,898 datos, siendo la última cifra el tamaño original de este grupo de datos. Las imágenes fueron obtenidas y tomadas por un grupo de maestros y estudiantes pertenecientes a la universidad de "Technical University of Ostrava"(Lab, 2021a), y pertenecen a un total de 37 diferentes personas donde tienen las mismas condiciones que las 4000 imágenes.

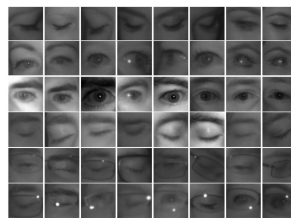


Figura 11. Ejemplo de imágenes pertenecientes al grupo de datos MRL

7.1.2. Análisis de datos

Se realizó un reporte por medio de la librería pandas-profiling Figura 12, Figura 13 y Figura 14, en donde se observó que las imágenes estaban correctamente balanceadas debido a que el mismo reporte indica los desbalance importantes pertenecientes al grupo de datos de 4000 imágenes, sin embargo, pese al análisis y ver un conjunto de datos variado, se determinó que las imágenes estaban tomadas en casos controlados donde la cámara pareciera estar de frente o con un mínimo traslado.

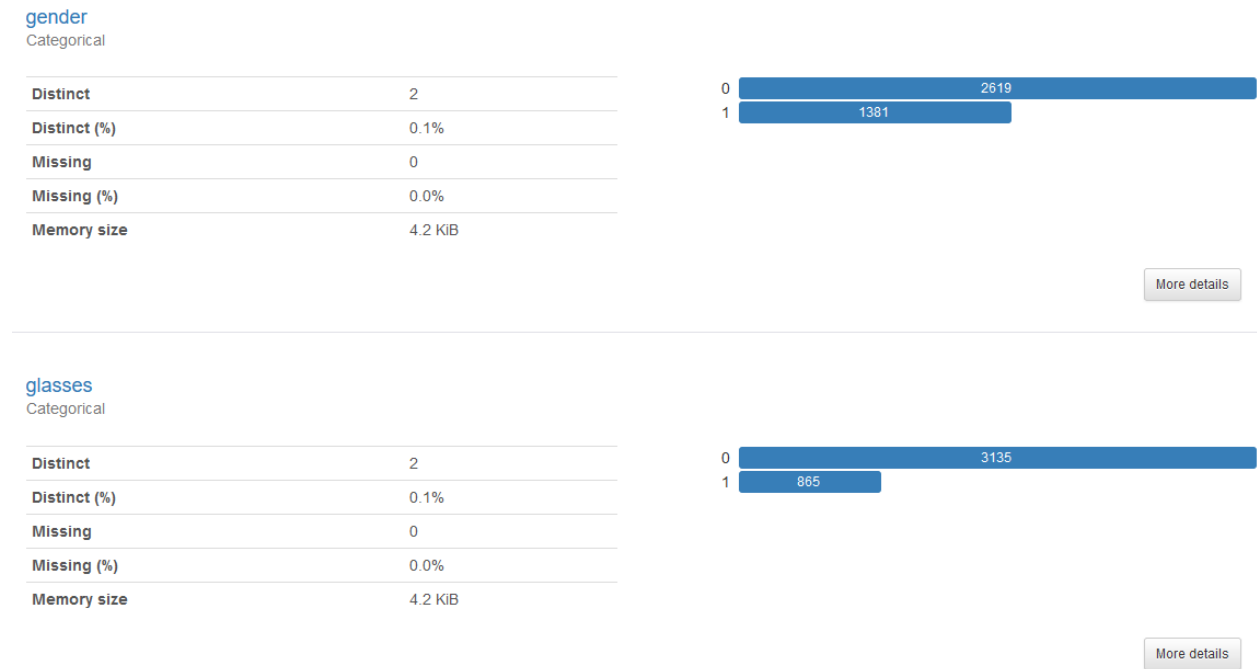


Figura 12. Primera parte de reporte

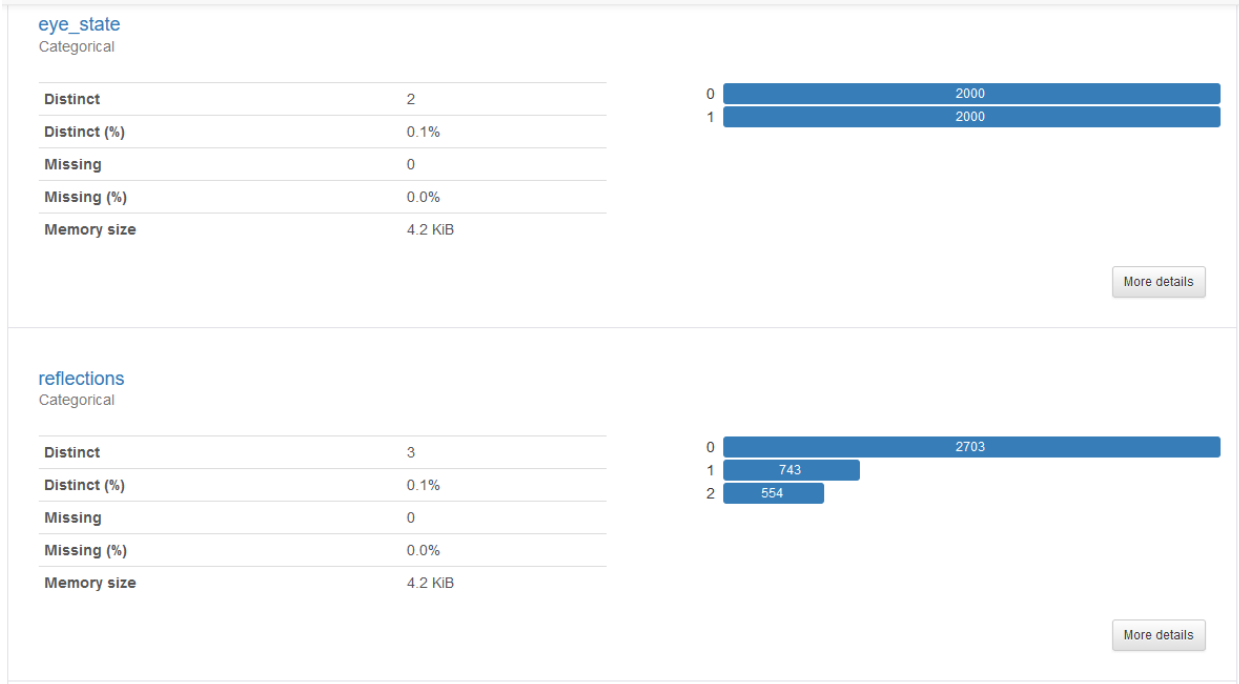


Figura 13. Segunda parte de reporte

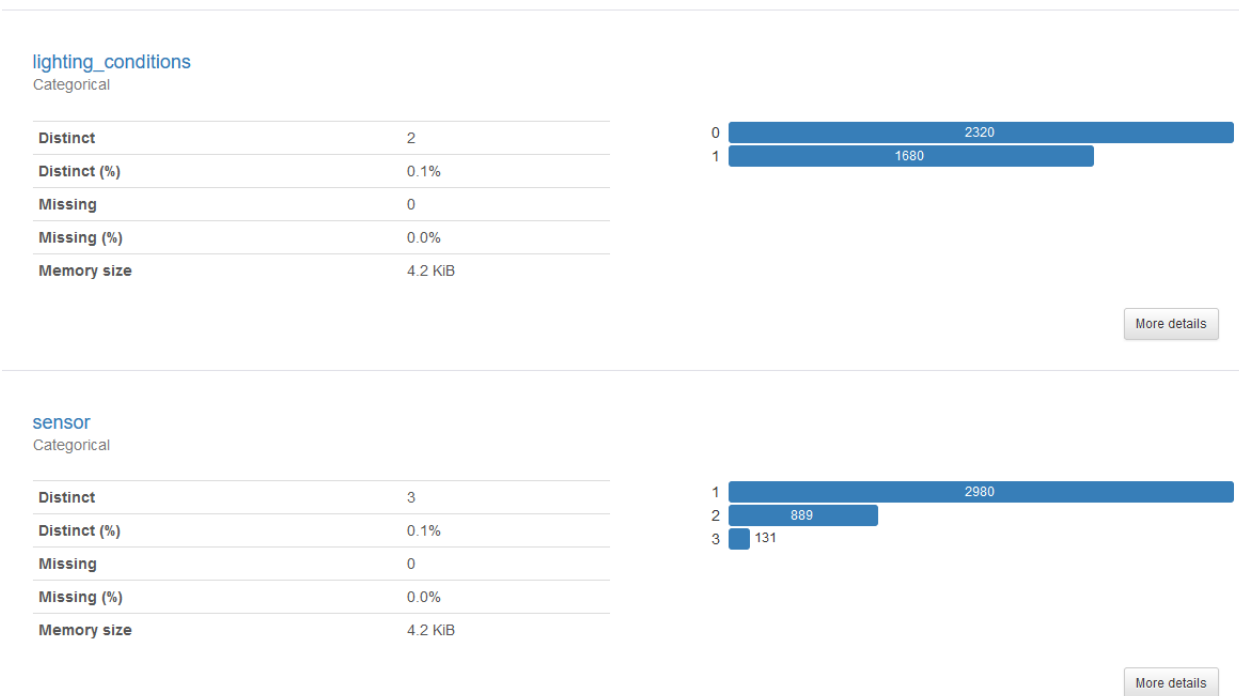


Figura 14. Tercera parte de reporte

En cuanto al análisis de los 84,898 datos Figura 15, Figura 16 y Figura 17, este tuvo un comportamiento distinto, empezando por un desbalance notorio y significativo del sensor con el que se tomaron las fotografías.

Debido a esto se decidió centrar la distribución de los datos con el sensor con mayor frecuencia de imágenes. Al tener solamente uno podemos tener la certeza de que todas las imágenes fueron hechas con el mismo sensor.

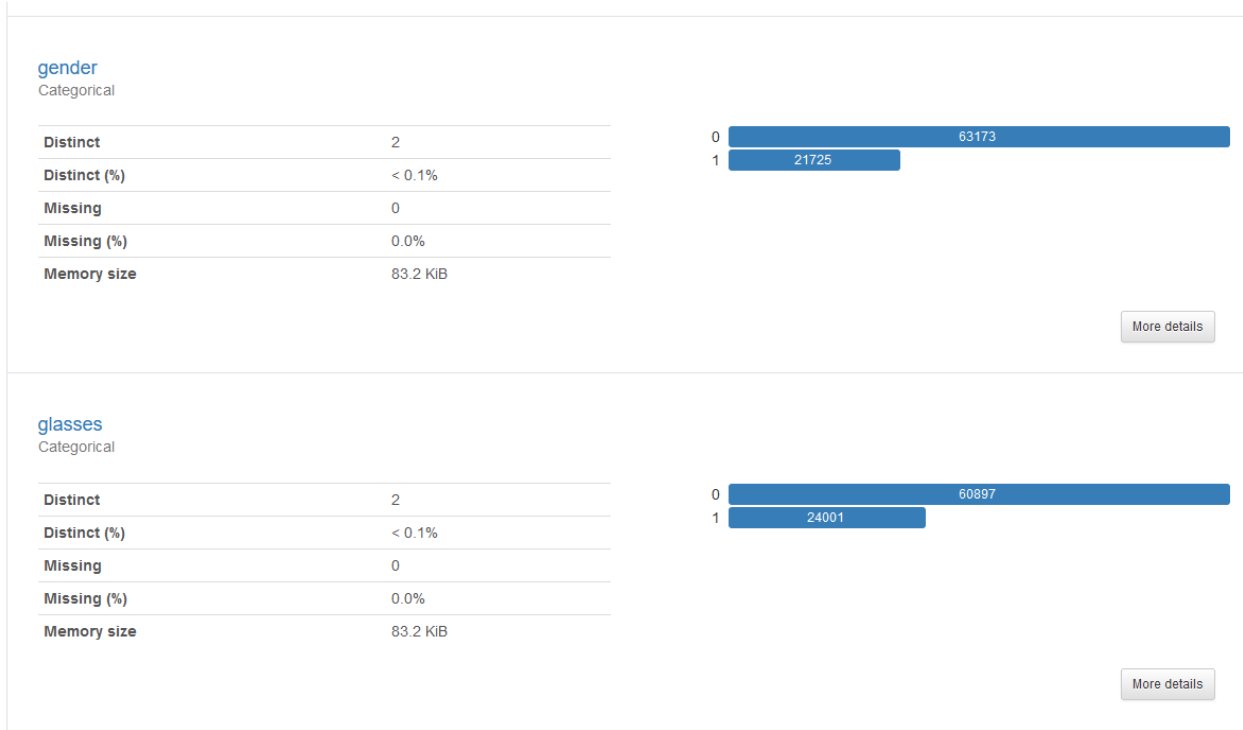


Figura 15. Primera parte del reporte del segundo grupo de datos completo

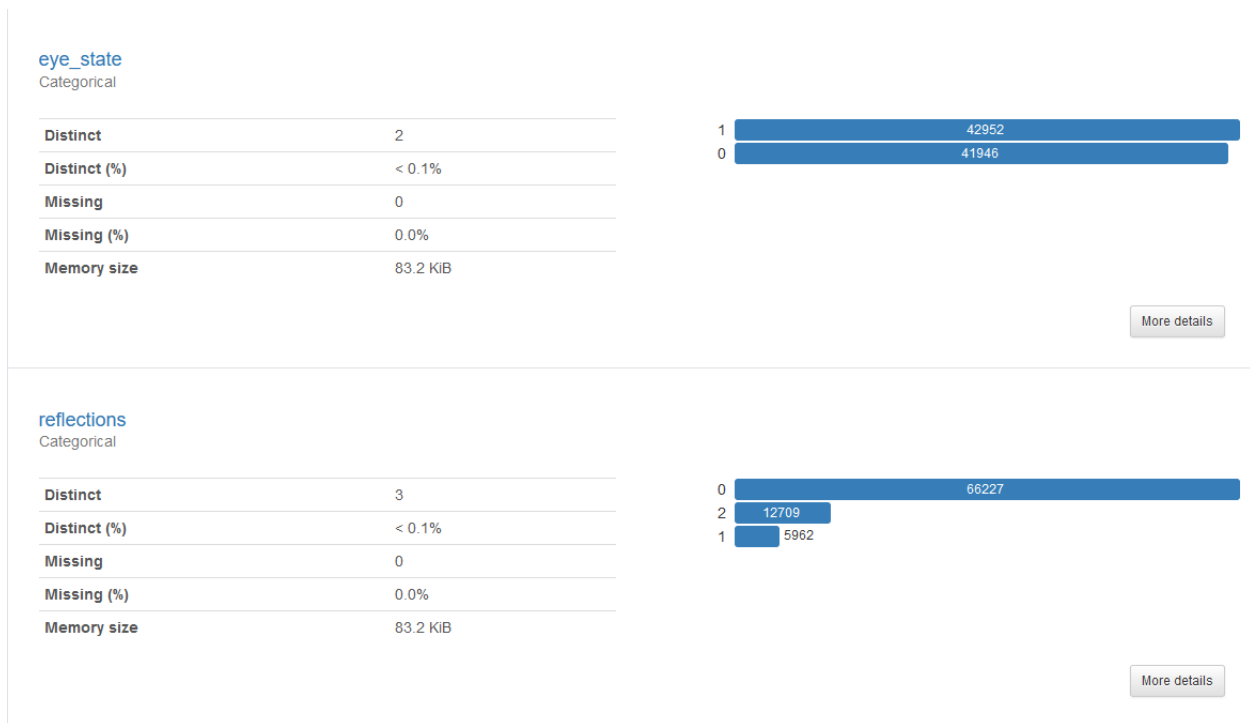


Figura 16. Segunda parte del reporte del segundo grupo de datos completo

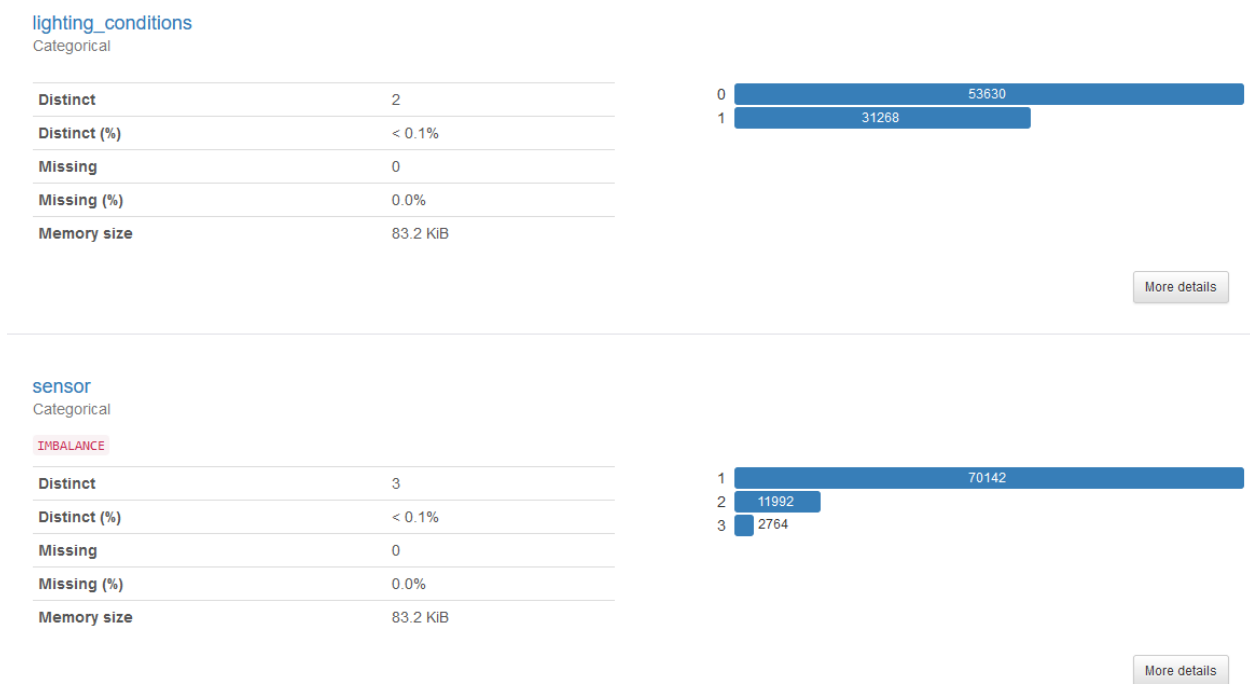


Figura 17. Tercera parte del reporte del segundo grupo de datos completo

7.1.3. Aumento de datos

Una vez se exploraron y analizaron los datos para reparar posibles desbalance en los datos o pérdida de datos se empezó con el aumento de datos, al comprender que las imágenes estaban bajo un ambiente controlado surge la necesidad de poder obtener casos diversos relacionados a estos datos, como se explicó anteriormente esta técnica permite transformar los datos en casos variados de los datos ayudando a obtener una mayor diversidad y controlar el caso del sobreajuste por el modelo, causando que este se adapte a escenarios variados y no casos controlados como en los que las imágenes fueron hechas.

Bajo la tarea final del modelo de predecir el estado de los ojos dentro de un carro, debido a que los ojos tienden a guardar una relación con la fatiga y puede ser determinado por medio de la frecuencia de pestaños, es importante captar si estos están abiertos o cerrados, una vez entendido esto se plantearon varios casos, en el caso de día la variación de la luz será más intensa, y en la noche el contrario con la luz, la cámara puede estar rotada debido a la disposición de tamaño donde se coloque, los ojos debido a la posición de la cabeza pueden estar rotados, etc.

En base a esto se definieron los parámetros dentro del aumento de datos, realizado por medio de las librerías de Tensorflow y sus funciones disponibles para configurar el aumento, con las siguientes transformaciones:

- La normalización de los datos, esto más que una transformación es para trabajar con los mismos rangos en imágenes no importando su formato inicial de los píxeles.
- Rotación en los ejes x, y, z de modo que se cubran casos donde la cámara o la persona esté en un ángulo distinto del caso de frente o con 0 grados de rotación.
- Traslado de eje x y y debido a que los ojos pueden ser detectados con traslados y la imagen de los ojos puede no quedar centrada.
- Volteo horizontal, esto es rotar en 180 grados la imagen de modo que, por ejemplo, un ojo izquierdo puede transformarse en un ojo derecho, permitiendo así mayor diversidad de datos y capacidad mayor de adaptación al modelo.
- Modificación de brillo que es importante ya que en casos de día el brillo será mayor y en casos nocturnos éste será menor.
- Ampliación o disminución de la imagen, para que que el ojo se mire completo con cejas o solamente el ojo.

| parámetro | valor |
|-------------------------------|------------|
| rescale | 1/255 |
| rotation range | 20 |
| width shift range | 0.02 |
| height shift range | 0.01 |
| horizontal flip | True |
| brightness range | [0.7, 1.2] |
| shear range | 0.02 |
| zoom range | 0.07 |
| featurewise std normalization | True |
| samplewise std normalization | True |
| fill mode | nearest |

Tabla 1. *Parámetros de la aumentación de datos*

7.1.4. División del conjunto de datos

Se separó los datos en dos conjuntos, uno para entrenar el modelo y otro para prueba. Se decidió hacer la división con el 80 por ciento de los datos para entrenamiento y el 20 por ciento para la prueba.

7.1.5. Desarrollo del modelo

Selección De Modelos

Se analizaron los distintos modelos para la prueba final en conjunto con PERCLOS, de modo que se seleccionaron y se eliminaron para dar como resultado final al mejor o mejores modelos para el uso posterior de esto en las pruebas finales. En este paso se incluyó la prueba de hiperparámetros en los modelos y el descarte de los mismos en base a pruebas preliminares, donde se obtuvieron resultados para hacer una comparación entre estos.

7.2. Uso final del modelo con fórmula de PERCLOS

Una vez seleccionado los modelos, se utilizaron vídeos recolectados para poner a prueba la fórmula de PERCLOS. Se visualizaron los valores de PERCLOS a través del tiempo, así como las predicciones realizadas por el modelo antes de ser analizadas por la fórmula. De esta manera, se pudo identificar si la fórmula es efectiva para determinar la fatiga de los sujetos. Además, se pudieron identificar los casos en que la fórmula podría fallar, proponiendo soluciones o explicaciones a estos fenómenos.

La manera de extraer la región de interés de los ojos, se realizó por medio de la librería de dlib para extraer la cara y luego los ojos. Se extraía el ojo derecho que era el que estaría en ángulo contra la cámara. Luego se predecía con el modelo, el estado de cerradura, obteniendo todos los estados de cerradura del vídeo, que como punto final eran analizados por la fórmula PERCLOS.

La metodología propuesta consiste en obtener un frame por segundo de un vídeo. Luego, se calcula

el PERCLOS de 5 frames consecutivos, obteniendo el promedio de ese valor. Este proceso se repite hasta el final del vídeo. Se obtiene una gráfica que representa el valor de PERCLOS a lo largo del tiempo. Los picos altos de la gráfica indican un indicio de fatiga, mientras que los valores bajos indican que los ojos permanecen abiertos o que la frecuencia de pestañeo es baja.

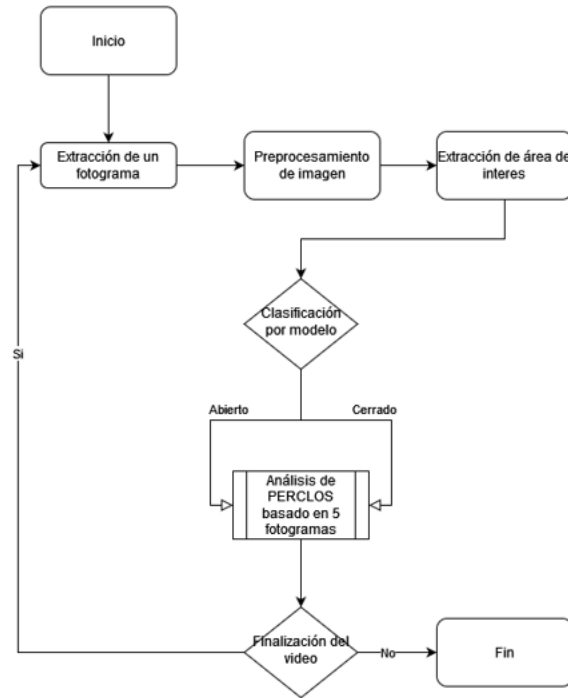


Figura 18. *Proceso de análisis de vídeo*

A continuación se observan las funciones utilizadas para lograr el análisis de un vídeo, de este modo puede ser replicado en futuras investigaciones:

```

1 face_detector = dlib.get_frontal_face_detector()
2 eye_detector = dlib.shape_predictor('./shape_predictor_68_face_landmarks.dat')
3 model = MultiModalPredictor.load("models_image_only/")
4 """ model = joblib.load("svm_model_complete_data.pkl") """
5 def get_frames_from_video(video_path, fps=None):
6     """Get the frames from a video in AVI format at a given FPS.
7
8     Args:
9     video_path: The path to the video file.
10    fps: The desired FPS.
11
12    Returns:
13    A list of frames.
14    """
15
16    # Create a video capture object
17    cap = cv2.VideoCapture(video_path)
18
19    # Get the total number of frames in the video
20    num_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
21
22    # Calculate the frame interval
23    #frame_interval = int(fps)
24    frame_interval = int(cap.get(cv2.CAP_PROP_FPS))
25
26    # Create a list to store the frames
27    frames = []
28
29    if frame_interval == 0:
30        frame_interval = 30
31        if fps:
32            frame_interval = fps
33
34
35    # Iterate over the frames in the video
36    for i in range(0, num_frames):
37        # Read the next frame
38        ret, frame = cap.read()
39
40        # If the frame is not empty, add it to the list
41        if i%frame_interval == 0:
42            frames.append(frame)
43
44    # Release the video capture object
45    cap.release()
46
47    return frames

```

Figura 19. Función para obtención de frames de un vídeo

```

1 def perclosRate(l):
2     perclos = []
3     perclos_values = []
4     if l.count(2) > 1:
5         l.remove(2)
6     for i in l:
7         perclos.append(i)
8         if len(perclos) == 5:
9             per_perclos = perclos.count(0)/5*100
10            perclos_values.extend([per_perclos]*5)
11            perclos = []
12    return perclos_values

```

Figura 20. Función para la obtención del valor de PERCLOS basado en una lista de predicciones "l"

```

1 def predict_frame(model, frame, lx, ly, lw, lh):
2     """Predict the class of a frame image using a model.
3
4     Args:
5     model: A model.
6     frame: A frame image.
7
8     Returns:
9     The predicted class.
10    """
11
12    img = np.array(frame)
13
14    faces = face_detector(img)
15
16    # Initialize eyes_roi
17    eyes_roi = None
18    #print("FACES", len(faces))
19
20    for face in faces:
21        # Get face landmarks
22        landmarks = eye_detector(img, face)
23
24        # Extract and resize left eye
25        # Adjust these values as needed
26        left_eye_x = max(0, landmarks.part(36).x -
27                        lx) # Extend by 35 pixels
28        left_eye_y = max(0, landmarks.part(36).y -
29                        ly) # Extend by 35 pixels
30        left_eye_width = landmarks.part(
31            39).x - landmarks.part(36).x + lw # Extend by 70 pixels
32        left_eye_height = landmarks.part(
33            39).y - landmarks.part(36).y + lh # Extend by 70 pixels
34
35        left_eye = img[left_eye_y:left_eye_y + left_eye_height,
36                      left_eye_x:left_eye_x + left_eye_width]
37        left_eye = cv2.resize(left_eye, (224, 224))
38        # Combine both eyes
39        # Duplicate the grayscale channel
40        eyes_roi = left_eye
41        cv2.imwrite('gray_image.png', left_eye)
42        break # Break after detecting eyes in the first face
43
44    if eyes_roi is not None:
45
46        prediction = model.predict({'image': ["gray_image.png"]})
47        if prediction == 0:
48            return 0
49        else:
50            return 1
51    return None

```

Figura 21. *Función para clasificar basado en un frame*

```

1 def get_predictions_video(video_path, fps, lx, ly, lw, lh):
2     # Get the frames from the video
3     frames = get_frames_from_video(video_path, fps)
4
5     # Iterate over the frames in the video and make predictions
6     predictions = []
7     for frame in frames:
8         predictions.append(predict_frame(model, frame, lx, ly, lw, lh))
9
10    # Do something with the prediction
11    return predictions

```

Figura 22. *Función para predecir las predicciones dentro de un vídeo*

```
1 predictions = get_predictions_video("ruta.avi", 35, 35, 38, 60, 60)
2 preclos = perclosRate(predictions_cleaned)
```

Figura 23. *Uso final de las funciones en conjunto*

Prueba de modelo (controladas)

Se probó en pruebas controladas, es decir simulaciones, donde los sujetos no se mantenían en conducción, solamente simulaban el caso de estar dentro de un carro y con una iluminación correcta. Grabadas por un estudio realizado para detectar bostezos donde se contiene vídeos recolectados con variación en los sujetos de prueba, género, lentes, iluminación, etnia, etc (Abtahi et al., 2020).

En este conjunto de vídeos, a pesar de que su función principal es la detección de bostezos puede ser utilizado para probar somnolencia a través de los ojos. Los vídeos son grabados de manera frontal con una cámara de resolución 640x640, su contenido total es de 5.1 gb. Las pruebas se realizaron con un PERCLOS de 60 por ciento con 5 frames y un frame por segundo de vídeo.

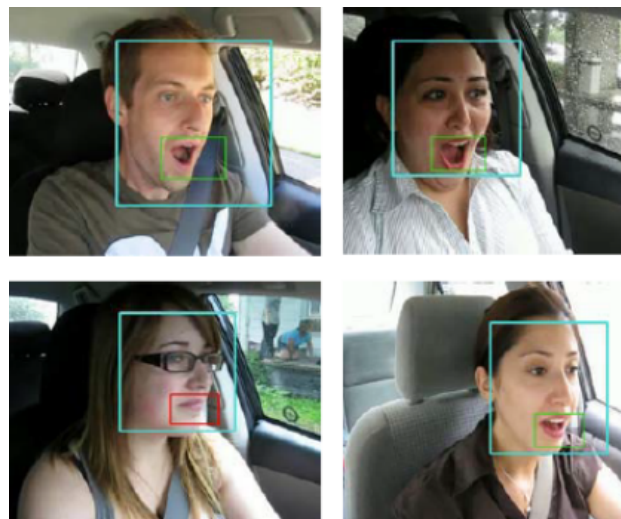


Figura 24. *Set de vídeos de conductores bostezando (Abtahi et al., 2020)*

Prueba de modelo (vehiculares)

Finalmente, el modelo se probó en casos vehiculares para comprobar que la fórmula de PERCLOS y el modelo daban indicios de fatiga en casos de uso reales. La metodología era la misma que la de pruebas controladas, analizar un vídeo con la métrica de PERCLOS y sus predicciones de pestañeo. Se probaron casos donde los sujetos estuvieron activos o fatigados.

Comprobación de modelo y fórmula

Se decidió comprobar la efectividad en base a vídeos pertenecientes a la base de datos DROZY, donde en esta se contiene archivos que graban una polisomnografía (Prueba para diagnosticar trastornos de sueño por medio de señales eléctricas), una prueba de reacción, una escala de sueño subjetiva, y un vídeo el cual es relacionado a los archivos mencionados, estos se relacionan con un sujeto, el cual fue puesto a una prueba de privación de sueño, para observar y examinar como el cuerpo humano se vería afectado en relación a esto. (Massoz et al., 2016)



Figura 25. Ejemplos de las pruebas de la base de datos de DROZY

Son 14 sujetos con sus correspondientes archivos, debido a una causa desconocida mencionada en la investigación de la base de datos, se menciona que hubo una pérdida de archivos al recuperarlos, por lo cual en las gráficas que se realizaran en conjunto con el análisis habrá una falta de estos. La metodología planteada propone analizar los vídeos con el modelo desarrollado, luego comparar los resultados obtenidos contra los archivos mencionados, y encontrar si existe una relación entre estos, de modo que pueda probarse que la fórmula de PERCLOS en conjunto con el modelo es efectivo.

7.3. Servidor y aplicación para el consumo de la aplicación

Este proyecto fue pensado primero como un mega proyecto y no un trabajo individual de trabajo profesional. Por tal motivo se desarrolló un servidor para comunicar la solución propuesta por el modelo de imágenes con los otros módulos de los compañeros de trabajo.

7.3.1. Servidor

Para utilizar el modelo con el celular, se creó un servidor sencillo con el *framework* Flask. Esto permitió utilizar todo el potencial de Python, ya que contiene librerías de uso sencillo para transformar y procesar imágenes. La imagen que será evaluada se obtendrá desde la aplicación y será clasificada por Python en el servidor. Esto nos da mayor libertad, ya que podemos utilizar las mismas funciones que se utilizaron en un cuaderno de Jupyter Notebook. Además, esta solución permite aislar el código que interactúa con el usuario y el que procesa el análisis.

7.3.2. Aplicación

Se creó una aplicación de celular con el *framework* kivy, la cual se levantaba y en segundo plano mantenía el servidor corriendo de manera independiente. Permitiendo la comunicación entre el servidor y la aplicación principal que se creó en el módulo de otro integrante del grupo. Se realizó de esta manera por términos de tiempo y como solución sencilla y eficiente.

8.1. Modelo de reconocimiento facial basada en el estado de los ojos humanos

8.1.1. Desarrollo del modelo

Gracias a la investigación realizada en el año 2022 por el instituto nacional de tecnología en Nagpur, India (Jay D. Fuletra, 2013), se pudo observar que las redes convolucionales son una opción bastante eficiente en problemas relacionados a la clasificación de imágenes para detectar somnolencia. Sin embargo, se utilizó la librería AutoGluon para ver si se podían obtener mejores resultados con otros modelos.

| | model | score_test | score_val | pred_time_test | pred_time_val | fit_time | pred_time_test_marginal | pred_time_val_marginal | fit_time_marginal | stack_level | can_infer | fit_order |
|---|---------------------|------------|-----------|----------------|---------------|-------------|-------------------------|------------------------|-------------------|-------------|-----------|-----------|
| 0 | MultiModalPredictor | 0.989572 | 0.9896 | 60.436221 | 15.096951 | 3698.916270 | 60.436221 | 15.096951 | 3698.916270 | 1 | True | 8 |
| 1 | WeightedEnsemble_L2 | 0.989572 | 0.9896 | 60.443228 | 15.104470 | 3699.953839 | 0.007007 | 0.007519 | 1.037569 | 2 | True | 9 |
| 2 | LightGBMXt | 0.587759 | 0.6188 | 0.003049 | 0.040397 | 0.973758 | 0.003049 | 0.040397 | 0.973758 | 1 | True | 2 |
| 3 | LightGBMLarge | 0.587759 | 0.6188 | 0.007946 | 0.006920 | 0.513180 | 0.007946 | 0.006920 | 0.513180 | 1 | True | 7 |
| 4 | LightGBM | 0.587759 | 0.6188 | 0.027504 | 0.028876 | 3.629066 | 0.027504 | 0.028876 | 3.629066 | 1 | True | 1 |
| 5 | XGBoost | 0.587759 | 0.6188 | 0.078331 | 0.029070 | 1.288698 | 0.078331 | 0.029070 | 1.288698 | 1 | True | 4 |
| 6 | NeuralNetTorch | 0.587759 | 0.6188 | 0.134664 | 0.014223 | 54.041266 | 0.134664 | 0.014223 | 54.041266 | 1 | True | 5 |
| 7 | CatBoost | 0.587759 | 0.6188 | 0.142742 | 0.020189 | 2.549569 | 0.142742 | 0.020189 | 2.549569 | 1 | True | 3 |
| 8 | VowpalWabbit | 0.587759 | 0.6188 | 0.435592 | 0.060209 | 5.582622 | 0.435592 | 0.060209 | 5.582622 | 1 | True | 6 |

Figura 26. Resultados de pruebas realizadas por AutoGluon

Puede observarse en los resultados brindados por la librería que los mejores modelos pertenecen a 2 redes neuronales que fueron seleccionadas y configuradas por la librería AutoGluon. Con esto se valida la idea inicial de que una red neuronal es un enfoque correcto que puede llevar a obtener resultados eficientes.

La red neuronal seleccionada obtuvo un 82.09 por ciento de exactitud al momento de entrenarla. A continuación, se muestra resumen del modelo:

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)              (None, 222, 222, 32)       896
max_pooling2d (MaxPooling2D) (None, 111, 111, 32)       0
conv2d_1 (Conv2D)            (None, 109, 109, 64)       18496
max_pooling2d_1 (MaxPooling2D) (None, 54, 54, 64)         0
conv2d_2 (Conv2D)            (None, 52, 52, 128)        73856
max_pooling2d_2 (MaxPooling2D) (None, 26, 26, 128)        0
flatten (Flatten)            (None, 86528)              0
dense (Dense)                 (None, 512)                44302848
dropout (Dropout)            (None, 512)                0
dense_1 (Dense)              (None, 1)                  513
-----
Total params: 44396609 (169.36 MB)
Trainable params: 44396609 (169.36 MB)
Non-trainable params: 0 (0.00 Byte)

```

Figura 27. Resumen de modelo inicial

Selección de modelos

Se utilizó el modelo de red neuronal para verificar su capacidad de clasificar nuevas imágenes, los resultados obtenidos fueron deficientes, en comparación a los que se obtuvieron al entrenarse con los datos iniciales, teniendo como resultado una exactitud de 0.62 y una precisión de 0.63, siendo reflejado en la matriz de confusión Tabla 6.

| | Abierto | Cerrado |
|---------|---------|---------|
| Abierto | 8 | 8 |
| Cerrado | 4 | 12 |

Tabla 2. Matriz de confusión de red neuronal inicial

Se propusieron como alternativas los modelos generados por la propia librería de Autogluon y un modelo basado en la extracción de características de HOG en conjunto con las máquinas de soporte vectorial (SVM) con un hiperplano basado en una función de base radial, al igual que la red neuronal se pusieron a prueba y se visualizaron sus resultados, los cuales mejoraron en relación al primer modelo.

En la Tabla 3 pueden observarse los resultados de los modelos con relación a datos fuera del conjunto inicial, obtenidos de vídeos recolectados por un estudio (Abtahi et al., 2020) y propias. Se puede observar que el modelo CNN carece de unos resultados aceptables. Luego las SVM como segunda opción, en especial la que fue utilizada con los datos completos, posee unos resultados prometedores y eficientes, sin mencionar que este algoritmo es el más sencillo.

| | Exactitud | Precisión | Exhaustividad | Especificidad |
|-----------------------------------|-----------|-----------|---------------|---------------|
| SVM P | 0.69 | 0.81 | 0.69 | 0.65 |
| SVM C | 0.81 | 0.86 | 0.81 | 0.81 |
| CNN P | 0.62 | 0.63 | 0.62 | 0.62 |
| Autgluon Tabular Imagen C | 0.5 | 0.25 | 0.5 | 0.33 |
| Autgluon MultiModal Imagen C | 0.91 | 0.92 | 0.91 | 0.91 |
| Autgluon Tabular Imagen + Texto C | 1 | 1 | 1 | 1 |

Tabla 3. Resultados del modelo final con respecto a datos de prueba no pertenecientes a datos de entrenamiento, siendo el identificador C (conjunto completo) y P (conjunto de 4000 datos)

Como última opción, AutoGluon brinda unos resultados superiores con respecto a las opciones MultiModal Imagen y Tabular Imagen Texto, se puede observar que incluso la puntuación llega a ser perfecta en el caso del último modelo. Por otro lado el modelo Tabular Imagen posee resultados deficientes, regresando a los mejores modelos, en especial el modelo con una puntuación perfecta. Se consideró su uso pero sucedió lo siguiente, este al ser puesto a prueba en condiciones nocturnas Figura 28 decaía y fallaba. Por lo cual se regresó a sus alternativas MultiModal Imagen y SVM C, las cuales continuaban en funcionamiento aún bajo condiciones nocturnas.

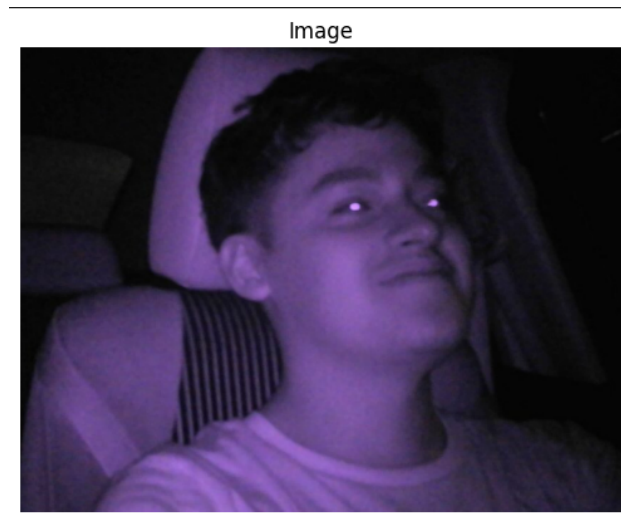


Figura 28. Muestra de foto nocturna de prueba

8.1.2. Resultados del modelo final

El modelo final obtuvo una precisión y exactitud eficiente, siendo 0.98 y 0.98 en entrenamiento, 0.93 y 0.94 en prueba, y finalmente 0.86 y 0.91 con datos fuera del conjunto inicial. Se puede observar que con datos fuera del conjunto inicial de datos, este tuvo una reducción de efectividad, sin embargo se mantuvo eficiente.

| | | |
|---------|---------|---------|
| | Abierto | Cerrado |
| Abierto | 1113 | 78 |
| Cerrado | 73 | 1136 |

Tabla 4. Matriz de confusión de SVM C (Conjunto completo) con respecto a datos de entrenamiento

| | | |
|---------------|-----------|-----------|
| Conjunto | Precisión | Exactitud |
| Entrenamiento | 0.98 | 0.98 |
| Prueba | 0.93 | 0.94 |

Tabla 5. Resultados del modelo final con respecto a datos de entrenamiento, SVM C (Conjunto Completo)

| | | |
|---------|---------|---------|
| | Abierto | Cerrado |
| Abierto | 8 | 8 |
| Cerrado | 4 | 12 |

Tabla 6. Matriz de confusión de SVM C (Conjunto completo) con respecto a datos fuera del conjunto de entrenamiento

| | | | | |
|--------|-----------|-----------|---------------|-------------|
| Modelo | Exactitud | Precisión | Exhaustividad | Especificad |
| SVM C | 0.81 | 0.86 | 0.81 | 0.81 |

Tabla 7. Métricas SVM C (Conjunto completo) con respecto a datos fuera del conjunto de entrenamiento

8.1.3. Resultados en pruebas controladas

Al poner a prueba el primer modelo de AutoGluon MultiModal Imagen C, sucedió lo observado en la Figura 29, donde podemos observar que todas las predicciones fueron ojos cerrados, indicando que con vídeos su eficiencia se vio altamente perjudicada, por lo cual se procedió a tomar el modelo de SVM C en el desarrollo de los resultados.

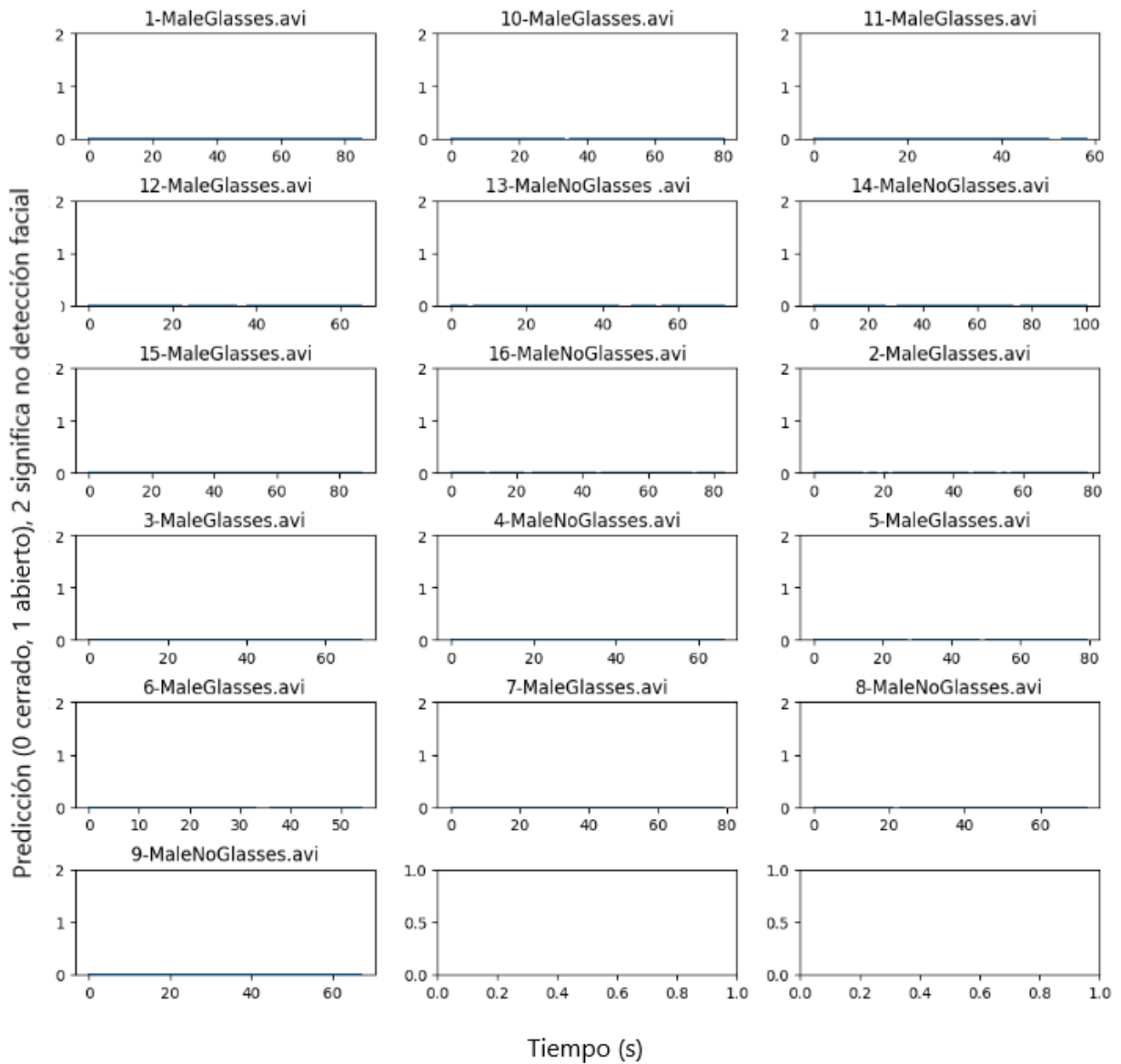


Figura 29. Gráfica de pestaños (0-Cerrado, 1-Abierto, 2-No detección de Cara) sobre tiempo en segundos de los videos de hombres (Modelo de Autgluon)

De los vídeos obtenidos, se generaron Figura 32 Figura 30, Figura 33 y Figura 31. Las gráficas Figura 32 y Figura 30 muestran los estados de los ojos a lo largo del tiempo. Al observarlas detenidamente, se pudo determinar que los lentes tienden a confundir al modelo en algunas tomas. Además, los giros de cabeza pueden causar que el modelo clasifique incorrectamente, como se muestra en la Figura 34. Esto se debe a que los sujetos conversaban con otro sujeto durante la prueba. Sin embargo, los resultados también ayudan a identificar los casos en los que el modelo falla. Es importante observar que el modelo también genera resultados satisfactorios en algunos casos, como se muestra en Figura 33 y Figura 31.

Puede observarse que PERCLOS es una medida que da resultados eficientes para denotar indicios de fatiga. A pesar de esto podemos notar otros errores, uno de estos es que en algunos vídeos los sujetos tienden a parpadear de forma exagerada en un intervalo de tiempo corto. Y sucede que la gráfica de PERCLOS se ve

afectada indicando un porcentaje alto, a pesar de que el sujeto no presenta signos de somnolencia.

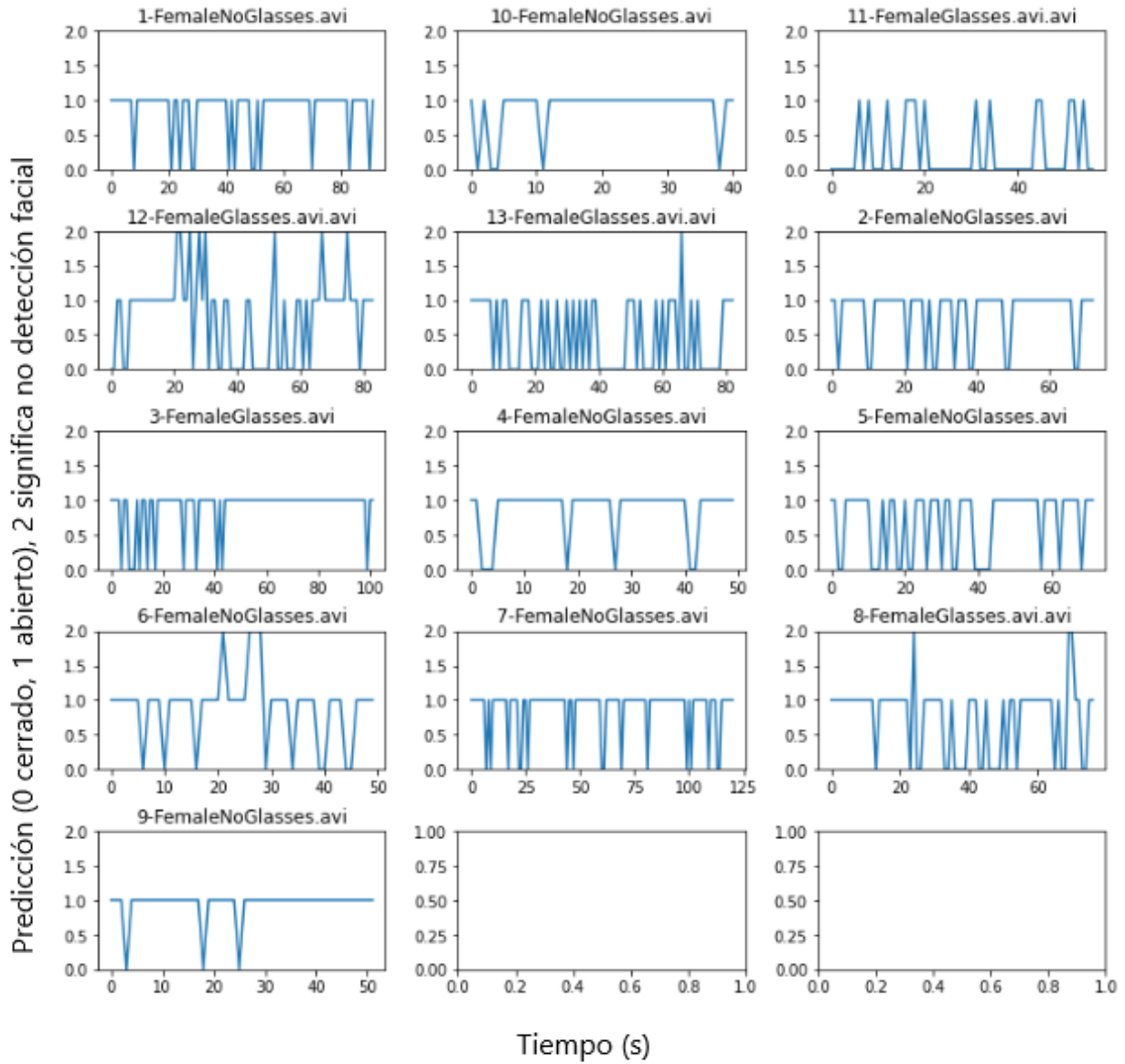


Figura 30. Gráfica de pestaños (0-Cerrado, 1-Abierto, 2-No detección de Cara) sobre tiempo en segundos de los vídeos de mujeres

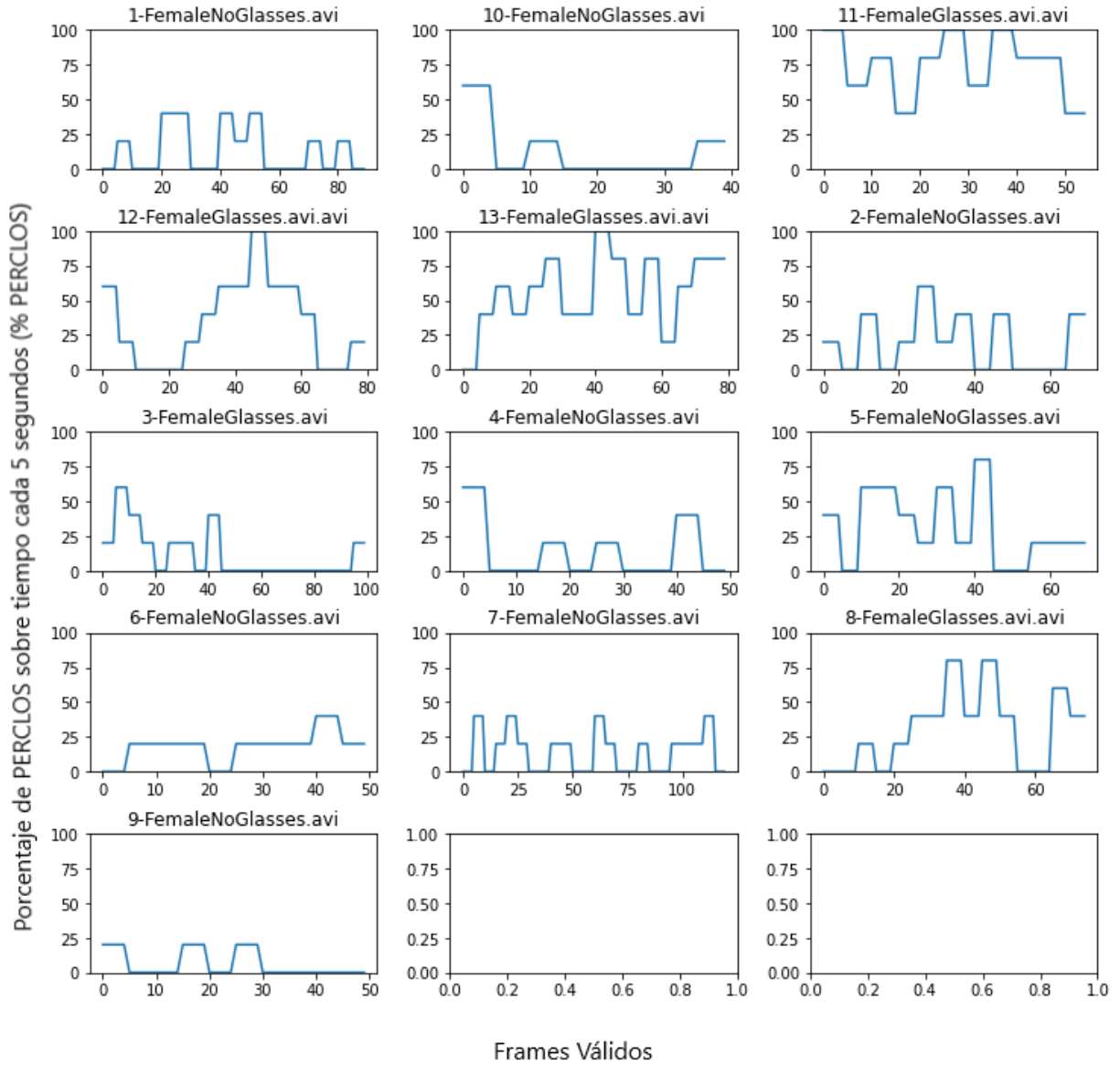


Figura 31. Gráfica de valor promedio de PERCLOS sobre frames válidos (frames catalogados con 1 o 0) de videos de mujeres

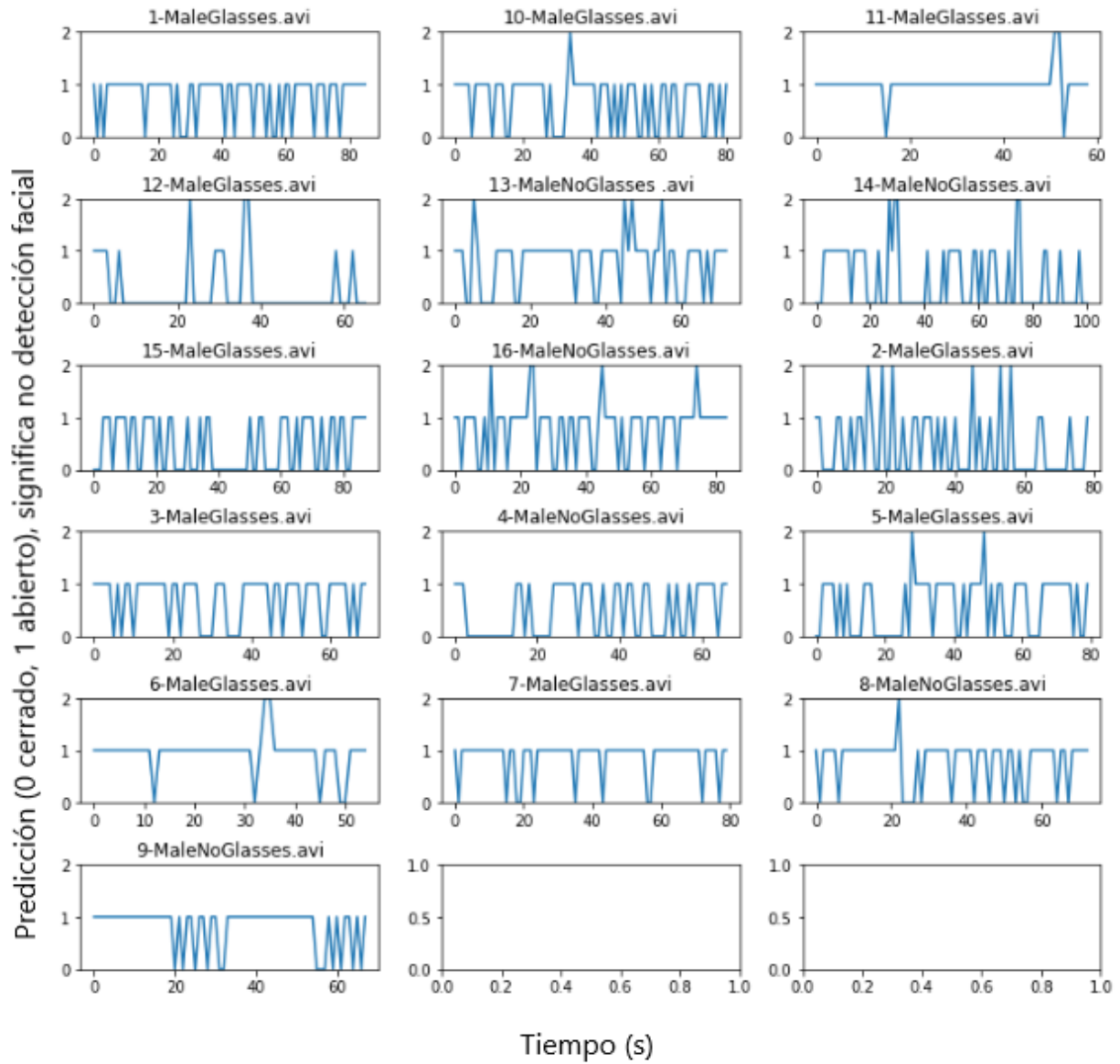


Figura 32. Gráfica de pestaños (0-Cerrado, 1-Abierto, 2-No detección de Cara) sobre tiempo en segundos de los vídeos de hombres

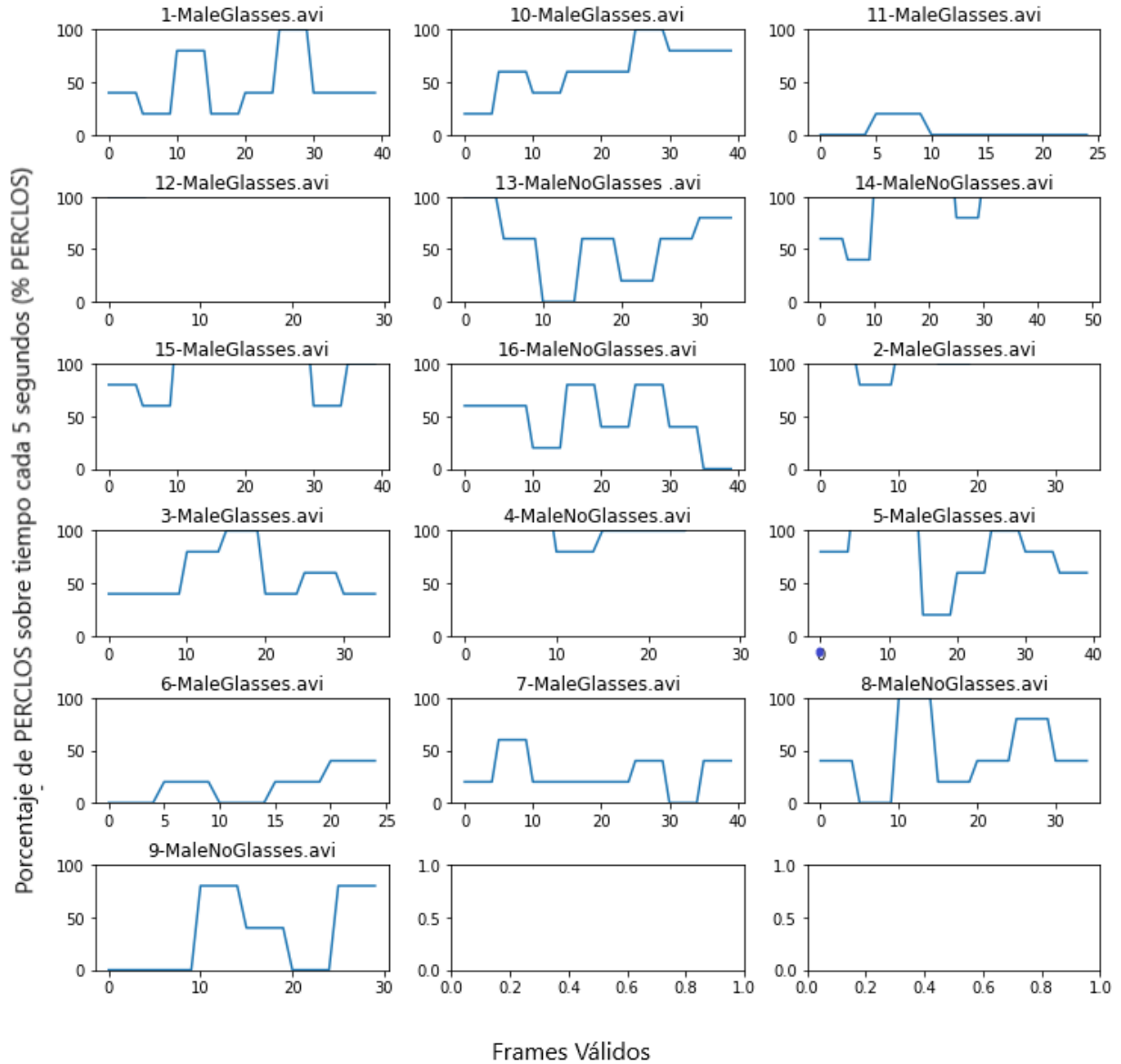


Figura 33. Gráfica de valor promedio de PERCLOS sobre frames válidos (frames catalogados con 1 o 0) de vídeos de hombres



Figura 34. Giro de la cabeza hacia un lado del sujeto de prueba en ambiente controlado

8.1.4. Resultados en pruebas vehiculares

Finalmente, se puso a prueba en conjunto con la fórmula de PERCLOS, en este caso con el modelo SVM C con un PERCLOS igual al de pruebas controladas del 60 por ciento. Se realizaron pruebas con la cámara dentro de un carro apuntando a la cara del piloto, algo que se puede observar es que los pilotos tienden a observar la cámara por reflejo, algo normal y causado por la sensación de ser observados Figura 35.



Figura 35. Escenario de prueba de noche

Luego de analizar el vídeo y observar el estado de los ojos con respecto al tiempo en la Figura 36. Se puede observar que la cara no puede ser reconocida en algunos casos debido al ángulo en el que está la cara, como ejemplo de girar esta misma, negando el análisis de la imagen por completo. Sin embargo se pudo capturar una cantidad necesaria de frames necesarios para analizar. En el caso de esta prueba el sujeto sostenía que su estado era activo sin ninguna fatiga o cansancio.

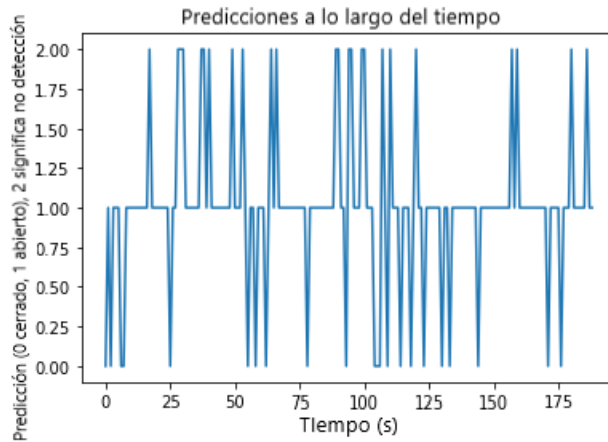


Figura 36. Estado de los ojos a lo largo del tiempo del primer sujeto en 40 segundos (0-Cerrados, 1-Abiertos, 2-Cara no reconocida)

En base al estado dicho por el sujeto, se obtuvo la siguiente Figura 37, donde como primer punto podemos observar que el valor de PERCLOS permanece bajo con un 60 por ciento como indicador de somnolencia. Haciendo sentido a lo que el sujeto sentía, y si analizamos los picos menores de 40 por ciento podemos observar gracias a Figura 37 y Figura 36, que se tratan de pestañeos comunes Figura 38, los cuales concuerdan en su mayoría con el vídeo de prueba del sujeto.

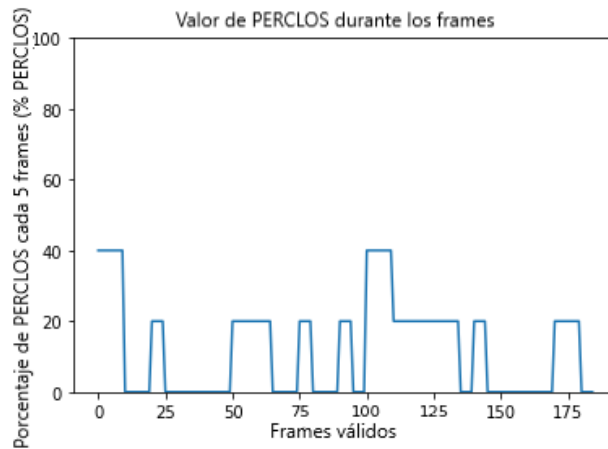


Figura 37. valor de PERCLOS sobre frames válidos (5 frames por cálculo)



Figura 38. Visualización de pestañeo prueba de noche

Luego tenemos otra prueba donde sucede que la cámara está bajo un movimiento constante Figura 41. Lo que hace que los frames terminen siendo borrosos y difíciles de analizar, esto se ve reflejado en las gráfica Figura 39 del estado de los ojos, donde el valor de 2 que hace referencia a una falta de reconocimiento ocurre con mayor incidencia que los demás valores. Lo que nos hace observar que el hecho de que la cámara está fija es un factor importante a resaltar al momento de clasificar, tomando este factor en cuenta el modelo puede ser o no efectivo. Además cabe mencionar la resolución de la cámara la cual es menor, causando la deformación de las imágenes durante el vídeo afectando negativamente a las clasificaciones.

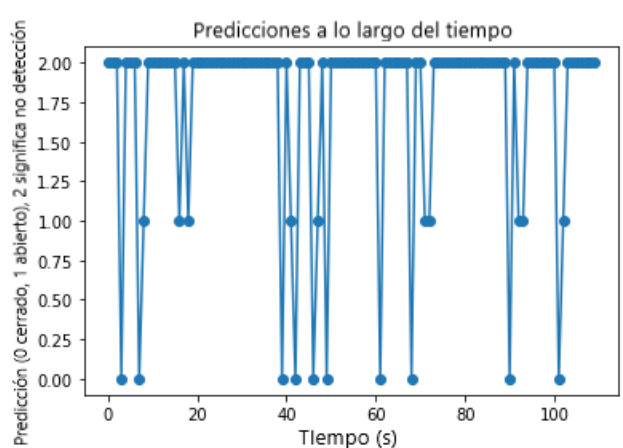


Figura 39. Predicciones de ojos del sujeto de prueba donde el movimiento es constante y desestabilizado

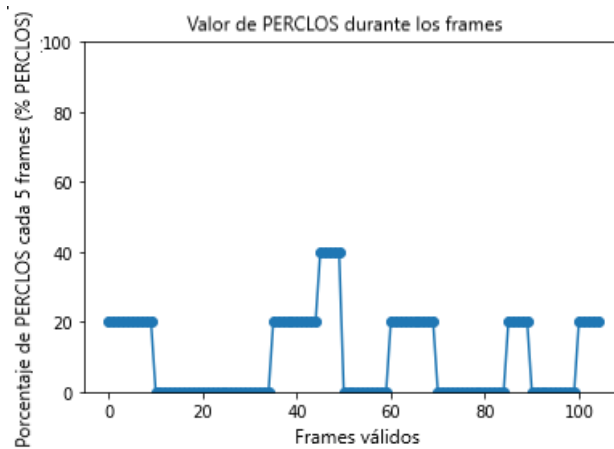


Figura 40. Perclos del sujeto de prueba donde el movimiento es constante y desestabilizado



Figura 41. Visualización de pestañeo de ojos del sujeto de prueba donde el movimiento es constante y desestabilizado

Se realizó una última prueba Figura 44 con un sujeto el cual según sus indicaciones y considerando su reciente actividad se encontraba con fatiga y sueño. El sujeto indicó que no había dormido bien el día de la prueba, mencionó que como todos los días fue a trabajar seguido de ir al gimnasio. Podemos observar primero que en la gráfica que analiza sus cerraduras de ojos Figura 42, estos son más frecuentes a lo largo del tiempo. Lo que indica que los ojos se mantienen cerrados en un mayor tiempo que un sujeto activo.

Luego, al analizar la gráfica del valor de PERCLOS Figura 43, podemos observar de una manera más notoria la diferencia entre un sujeto activo y uno fatigado. El valor de PERCLOS alcanza en un punto el 80 por ciento, y luego podemos observar que se mantiene entre 20 y 40 por ciento. A diferencia del sujeto que se mantiene activo este no se mantiene con los ojos abiertos de una manera más constante, sin mencionar que el sujeto con fatiga alcanza de forma frecuente el valor de 60 por ciento.

Como otro factor importante a resaltar es que la distancia entre la cámara y el sujeto es importante para que se pueda reconocer la cara y sus ojos. Ya que en este caso se obtuvieron más valores donde no se pudo identificar la cara, que además de ser causados por giros repentinos de la cabeza también se vieron influenciados por este factor. Pese a esto la descripción del sujeto concordó con los valores de PERCLOS los cuales se mantuvieron elevados a comparación de un sujeto activo de forma notoria y se obtuvieron valores elevados del mismo en algunos fragmentos del vídeo.

Podemos observar que si bien el modelo y la medida de PERCLOS pueden dar indicios acerca

de fatiga y somnolencia en pilotos. Este puede tener sus fallas relacionadas a factores humanos, como los giros de cabeza naturales de las personas, el movimiento de las mismas personas para acomodarse, lentes de los sujetos. Así como factores no humanos los cuales pueden ser movimientos de cámara y la propia incertidumbre del modelo el cual puede fallar. Esto nos da la señal que el modelo de detección facial enfocado a los ojos, puede llegar a funcionar de una forma más efectiva en conjunto al agregarse otros parámetros de detección, como bien pueden los signos vitales, postura, dirección del vehículo, etc.

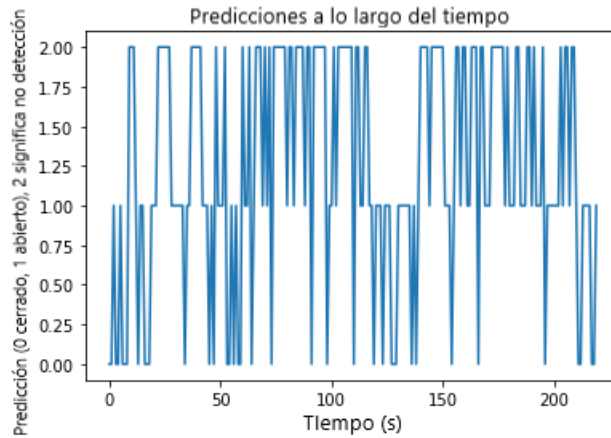


Figura 42. Predicciones de ojos del sujeto de prueba donde este se siente fatigado

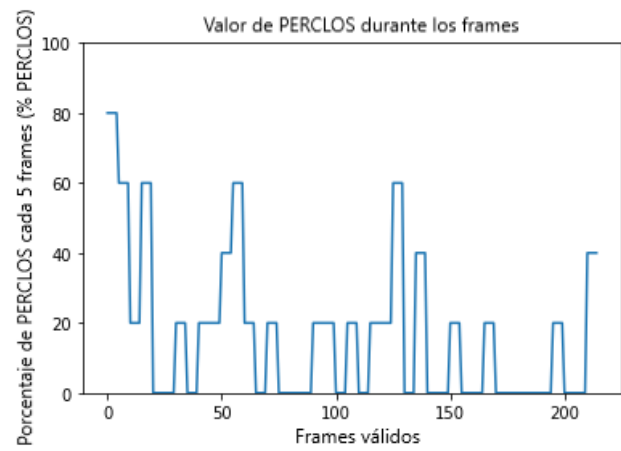


Figura 43. Perclos del sujeto de prueba donde este se siente fatigado



Figura 44. Escenario de sujeto con indicios de fatiga

8.1.5. Comprobación del modelo y fórmula

Se pudo observar dentro de las pruebas subjetivas de KSS, las cuales según la etapa de la prueba realizada en los sujetos de DROZY, a medida que esta avanzaba esta tuvo aumento, significando que los sujetos se sintieron fatigados a medida que se les privaba mas del sueño. Luego se pudo observar esta misma relación en los resultados del modelo desarrollado, donde a medida que la etapa de la prueba transcurría, valores como 100 o 80 de PERCLOS se hacían notar de manera mas significativa. Figura 45

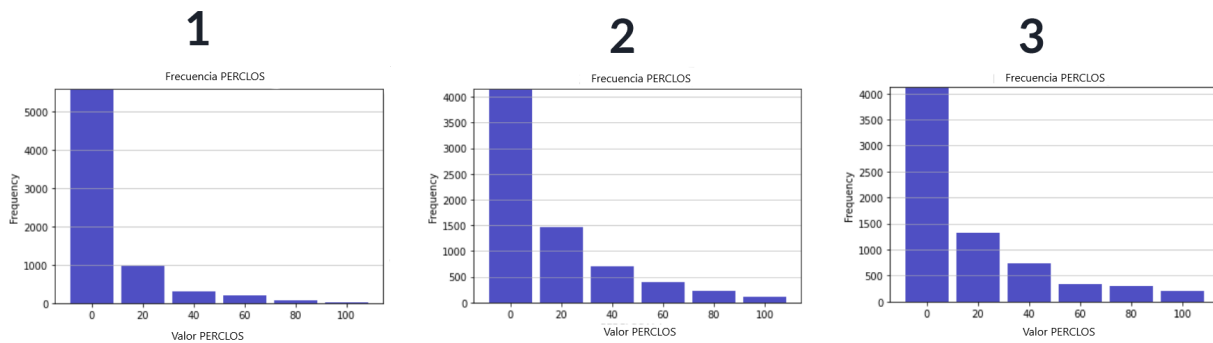


Figura 45. Frecuencia de PERCLOS con las 3 etapas de la prueba de DROZY siendo 3 el estado mayor de privación de sueño

Finalmente, al analizar los archivos de la PSG (polisomnografía) y los tiempos de reacción de los sujetos en relación a la etapa de la prueba, pudo observarse lo mismo Figura 47 y Figura 46, de manera que la somnolencia se veía presente en un aumento de los tiempos de reacción y una disminución en algunas señales eléctricas registradas por la PSG, que daban indicios a un estado de fatiga del sujeto.

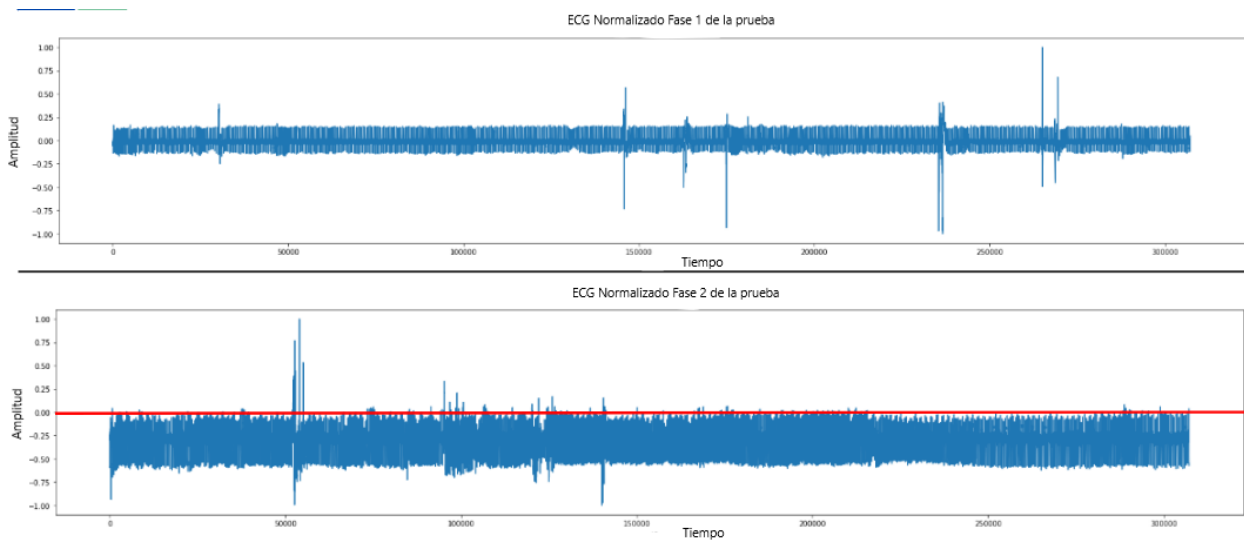


Figura 46. Señales de polimnografía en base al sujeto 4 de la base de datos DROZY

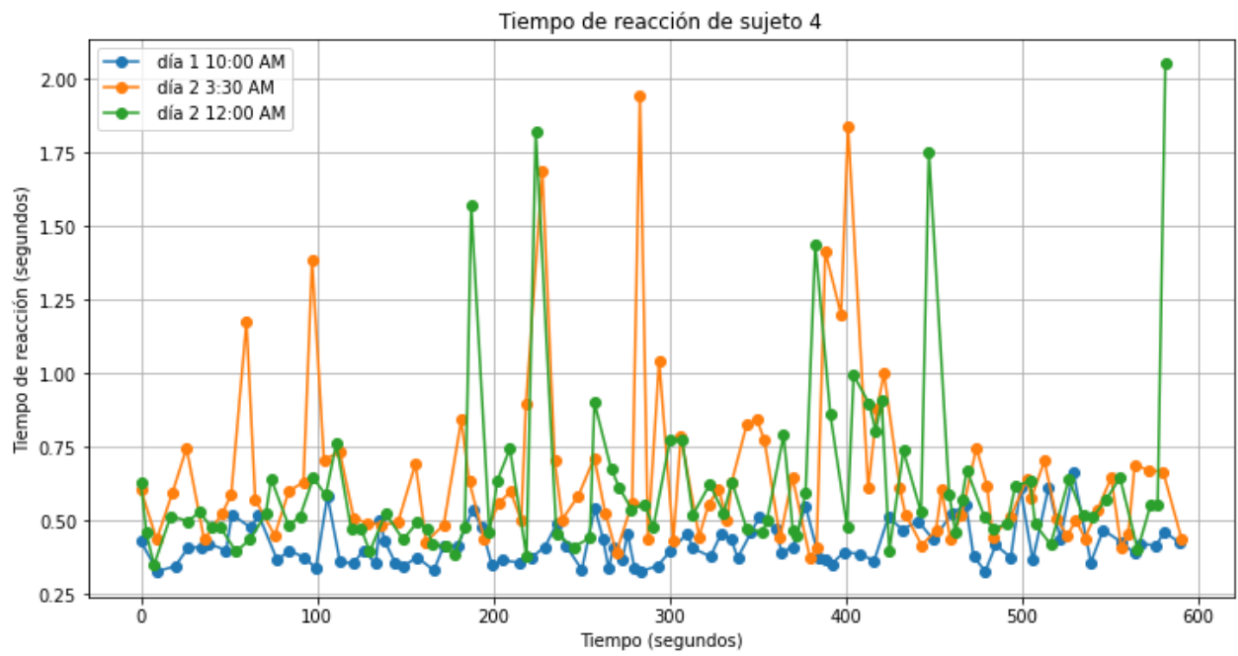


Figura 47. Tiempos de reacción en base al sujeto 4 de la base de datos DROZY

Luego de observar las relaciones se decidió por encontrar la magnitud numérica de estas relaciones, es decir con una función, si eran directamente o inversamente proporcionales a la privación del sueño. Al agrupar los datos por el nivel de prueba y sus segundos recorridos en cada fase de la prueba, pudimos observar que la fórmula de PERCLOS guardaba relación con las distintas señales Figura 48, recolectadas por la base de datos DROZY. Verificando así su eficiencia y uso correcto de esta, de manera objetiva y subjetiva relacionada a la fatiga y somnolencia de un sujeto.

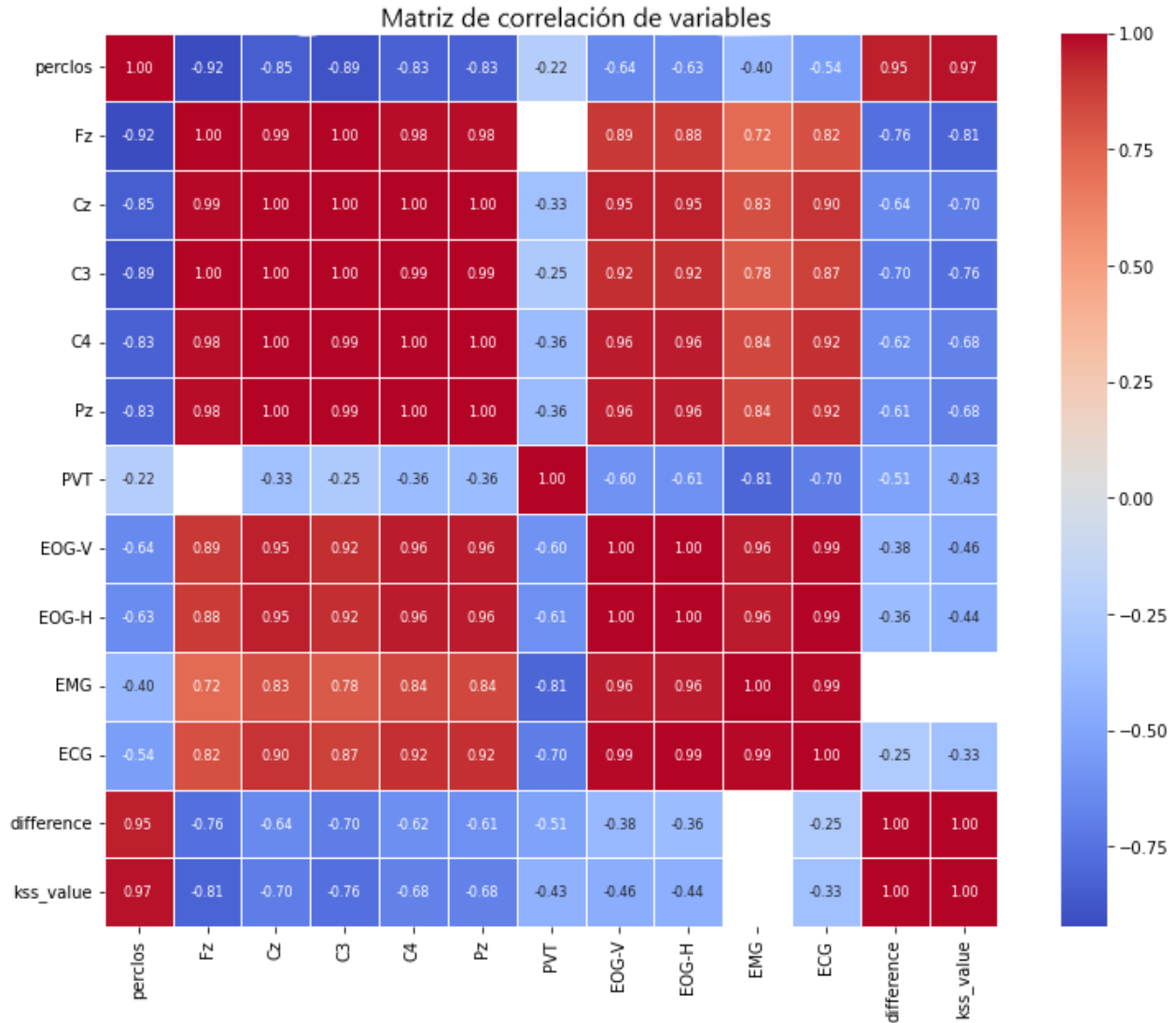


Figura 48. Mapa de correlación entre las variables de los sujetos de prueba de DROZY, agrupadas por nivel de la prueba y segundos dentro de cada fase

Las relaciones importantes a resaltar con los datos de DROZY son las siguientes:

- A medida que el valor de PERCLOS incrementa las señales de Fz, Cz, C3, C4, y Pz disminuyen, guardando una relación inversamente proporcional Figura 49 y Figura 50, la cual concuerda con los efectos de la somnolencia y el cuerpo humano. Las señales mencionadas, son eléctricas relacionadas al cerebro, es decir que por lo observado, el cerebro disminuye su actividad, lo cual hace referencia a un

estado de fatiga que a su vez se nota con PERCLOS.

- Al igual que las señales del cerebro, las oculares las cuales son EOG-V y EOG-H, siendo vertical y horizontal respectivamente, disminuyen de forma considerable, esto a su vez concuerda con la fatiga en el ser humano, donde la vista se ve afectada. Figura 50 Figura 51
- De la misma forma las señales cardíacas y musculares, las cuales son ECG y EMG respectivamente, disminuyen de la misma manera, al avanzar en la privación del sueño. Esto nuevamente guarda relación directa con los estados de la fatiga, donde el cuerpo entra en un estado de relajación cercano al sueño, siendo verificado de igual manera por la fórmula de PERCLOS. Figura 51
- Por último, de manera mas objetiva, los tiempos de reacción el cual es la variable difference, incrementan dando lugar a una relación directamente proporcional con PERCLOS. Esto guardando relación con una fatiga significativa, donde se dificulta la toma de desiciones y ejecución de tareas. De igual manera la fatiga o somnolencia subjetiva concuerda con los valores de KSS, que guardan la misma relación que los tiempos de reacción. Figura 52

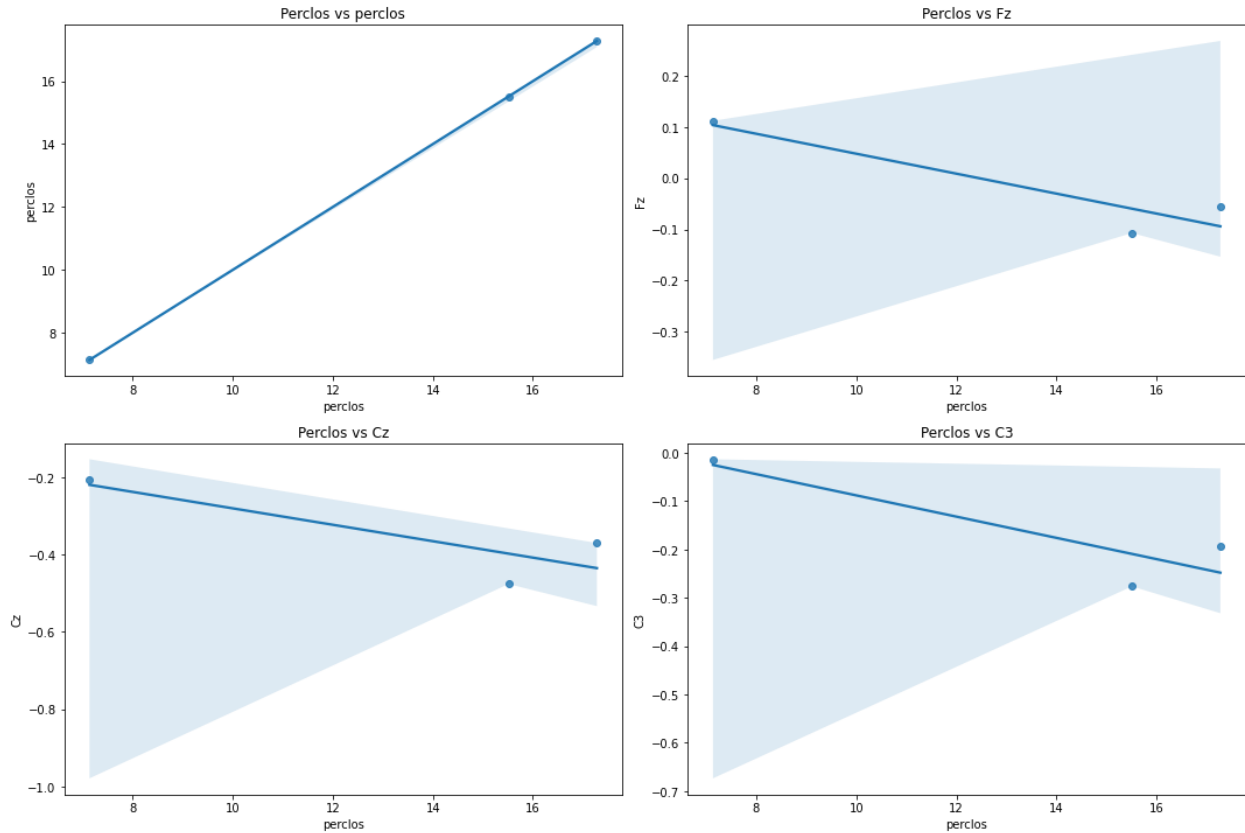


Figura 49. Primera parte de relaciones gráficas de DROZY

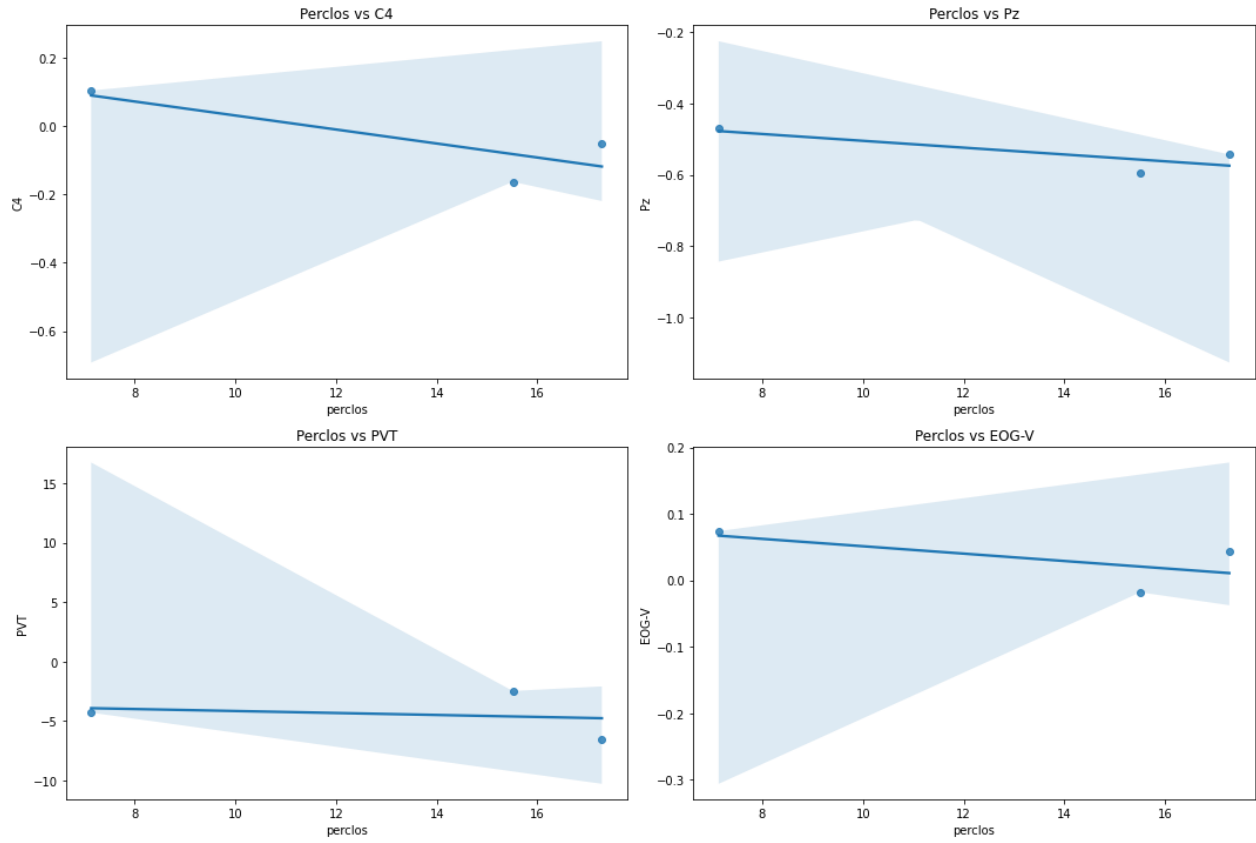


Figura 50. Segunda parte de relaciones gráficas de DROZY

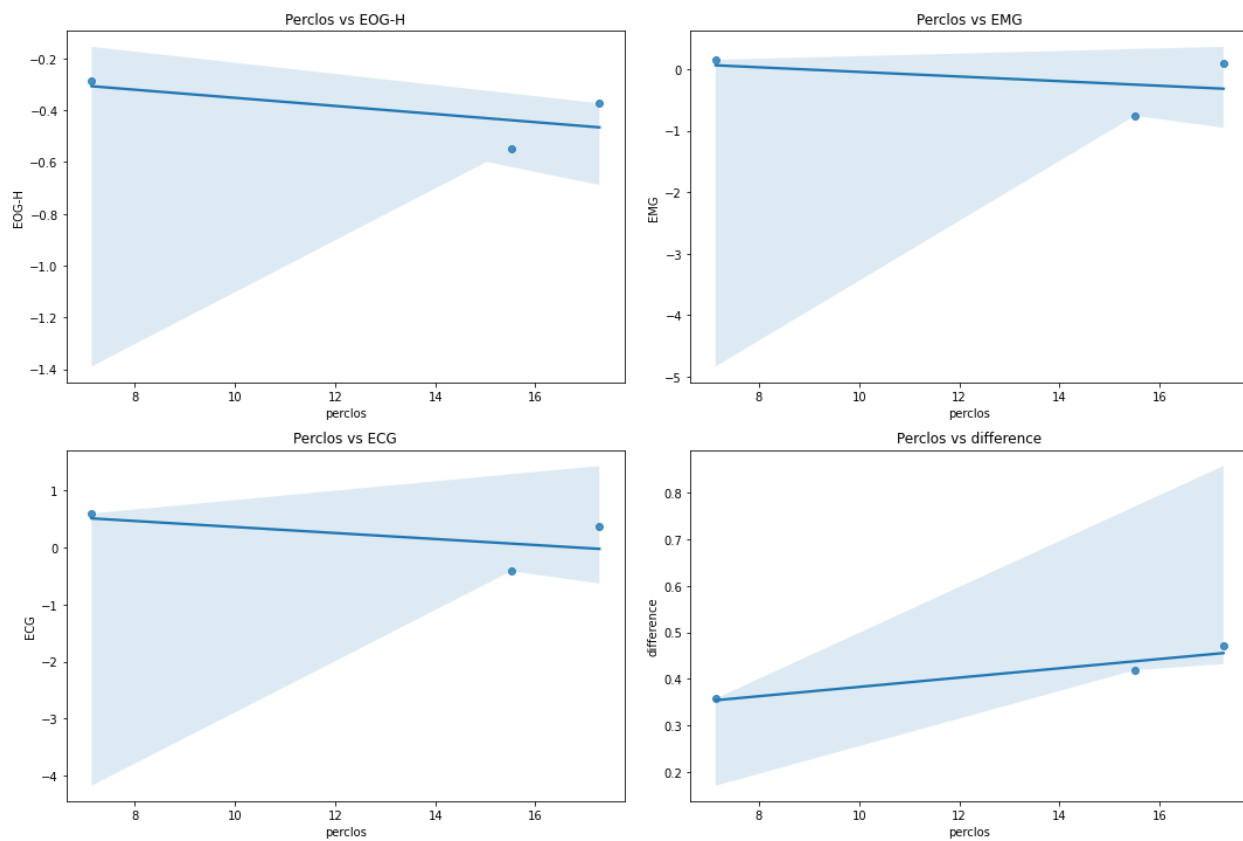


Figura 51. Tercera parte de relaciones gráficas de DROZY

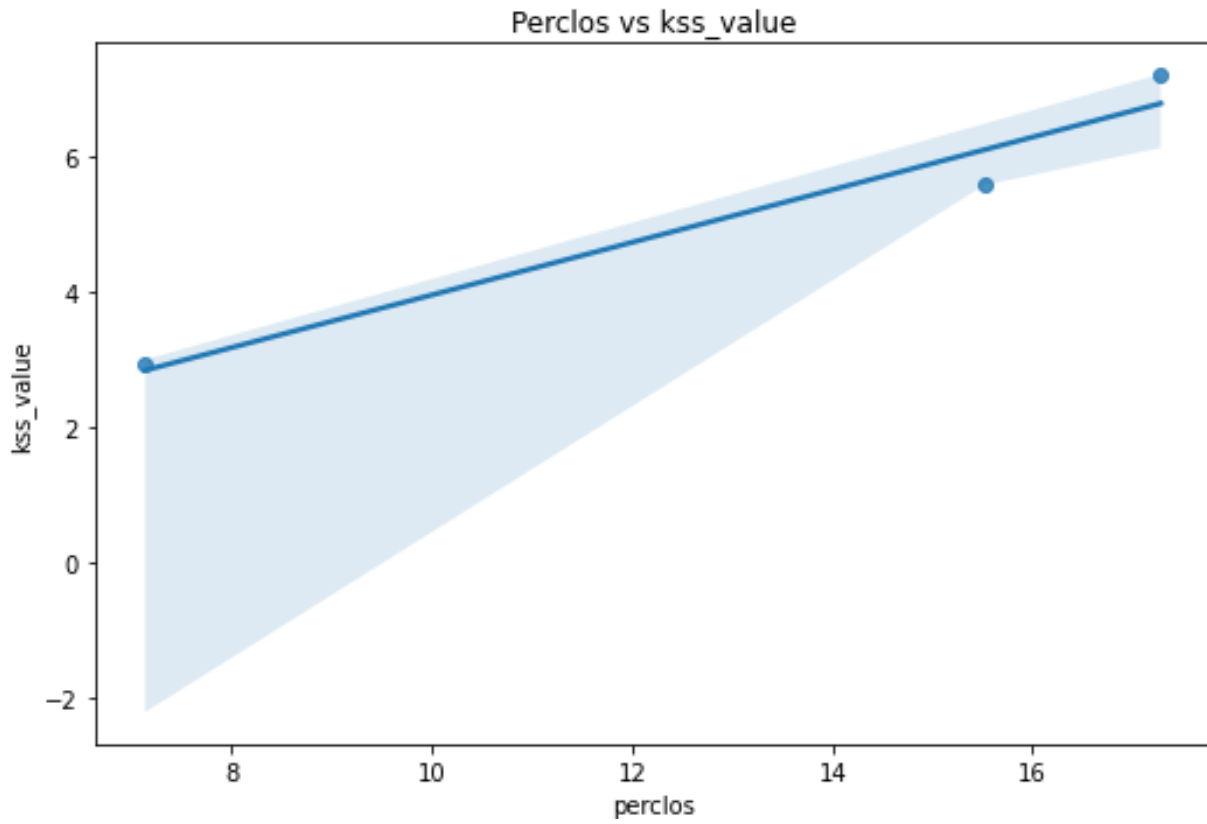


Figura 52. Cuarta parte de relaciones gráficas de DROZY

Conclusiones

Se desarrolló un modelo de detección de cerradura de los ojos que funciona de forma eficiente tanto de día como de noche que, en combinación con la fórmula de PERCLOS, permite detectar si el piloto presenta indicios de fatiga o no. El modelo se basa en máquinas de vectores de soporte en conjunto con el algoritmo de HOG.

Se validó el modelo en relación a imágenes diferentes al grupo de datos de entrenamiento. Obteniendo una exactitud de 81 por ciento, sin embargo, el rendimiento del modelo pudo mejorar y ser más efectivo si este hubiera poseído más datos tanto de prueba como de entrenamiento, y utilizado técnicas adicionales para el mejoramiento de eficiencia del mismo.

Se logró por medio de conocimiento adquirido de ciencia de datos, aplicar las técnicas correctas para obtener los resultados deseados. Se obtuvieron datos para entrenar al modelo, recolectado de distintas fuentes de uso gratuito.

Se observó que, pese a que los modelos obtienen métricas con rendimientos eficientes, estos puestos a prueba en situaciones reales y no ideales, tienden a decaer de manera que son descartados por completo. Sin embargo, esto corrobora que ninguno modelo es perfecto y dependerá de los datos con los que se entrene y los escenarios en los que se pongan a prueba.

Se determinó que la posición de la cámara es algo de suma importancia y esta debería de estar fija sin movimiento alguno. Ya que de esto depende fuertemente la efectividad del modelo, también se denota que existen factores no controlables y controlables, que afectan el modelo de forma positiva y negativa. Por lo cual es importante tomar en consideración al momento de desarrollar un producto final.

Se observó que, si bien las redes neuronales normales y convolucionales son el auge de hoy en día. No necesariamente significan la solución a todos los problemas, por términos de memoria, rapidez, comprensión y simpleza. En el caso del presente proyecto al final el modelo de SVM fue el que mejor se desempeñó siendo este menos complejo que sus alternativas.

Se analizó que la recolección de datos tanto de prueba y de entrenamiento, en base a resolución de las imágenes juegan un papel importante en el desarrollo y resultados de modelos de clasificación a través de

imágenes. Esto debido a que al momento de recolectarlos al ser de la misma resolución eliminan pasos de preprocesamiento, y en el momento de prueba ayudan a que preprocesar y probar estos sea más sencillo.

Se concluyó que el modelo puede dar resultados efectivos, por medio de la fórmula de PERCLOS, los cuales son validados por factores relacionados a la fatiga o somnolencia, de manera objetiva y subjetiva, como se pudo observar en rasgos físicos y psicológicos.

CAPÍTULO 10

Recomendaciones

Al trabajar con modelos de inteligencia artificial, debido a las librerías que se utilizan como tensorflow o sklearn, es eficiente tener el uso de estos modelos directamente compatibles con los lenguajes en los que se desarrollan. Se recomienda el uso de *frameworks* basados en python y no en aquellos que no, debido a que la importación de estos suele ser una tarea complicada.

Se incita a seguir recolectando datos en pruebas más cercanas al uso final del producto donde el carro esté en movimiento. Ya que este tipo de imágenes y vídeos son los que realmente cubren el caso de uso del modelo, tanto como para su entrenamiento y su validación, permitiendo tener una mayor eficiencia.

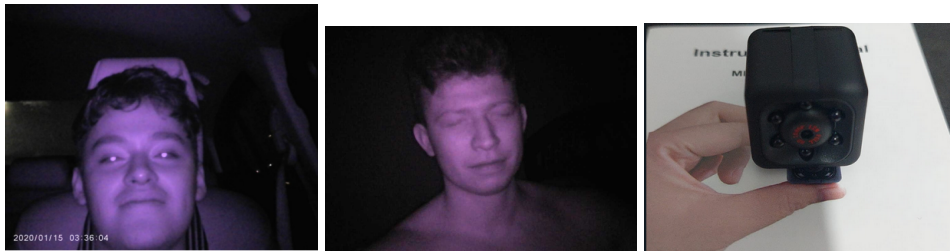
Se propone el uso de otras medidas para la detección de somnolencia, esto debido a que la solución planteada si bien nos permite saber un indicio, puede llegar a fallar. Esto quiere decir que no funciona, sino que propone el uso de más medidas para corroborar las medidas realizadas por el modelo de reconocimiento facial, basado en los ojos y la fórmula de PERCLOS. En algunos casos reemplazar la señal del modelo debido a los factores que niegan el análisis de imágenes los cuales se mencionaron.

- Abtahi, S., Omidyeganeh, M., Shirmohammadi, S., & Hariri, B. (2020). *YawDD: Yawning Detection Dataset*. <https://doi.org/10.21227/e1qm-hb90>
- Alonso, J. A. R. (2016). Generacion de algoritmos para clasificacion de obstaculos [[Accessed 23-10-2023]].
- Altamirano I, M. M., Castellucci I. (2016). Fatiga y somnolencia en conductores, ¿un problema conductual u organizacional? El caso de una empresa de buses en Chile. *Investigación*.
- Alvarado, I. (2023). Redes Neuronales [[Accessed 05-11-2023]].
- Antreas Antoniou, H. E., Amos Storkey. (2018). Data Augmentation Generative Adversarial Networks. *Research*.
- Artola Moreno, Á. (2019). Clasificación de imágenes usando redes neuronales convolucionales en Python. *Research*.
- Bracho-Paz, D. (2020). Síndrome de burnout y fatiga laboral. *Investigación*.
- Community, A. (s.f.). AutoGluon — auto.gluon.ai [[Accessed 22-10-2023]].
- Contributors, K. (2011). Kivy: Cross-platform Python Framework for NUI — kivy.org [[Accessed 22-10-2023]].
- Davis E. King. (2022). *Dlib C++ Library* [Accessed: October 20, 2023]. <http://dlib.net/>
- Edmundo Rosales Mayor, J. R. D. C. M. (2010). Somnolencia: Qué es, qué la causa y cómo se mide. *Investigación*.
- George, A. (2012). DESIGN AND IMPLEMENTATION OF REAL-TIME ALGORITHMS FOR EYE TRACKING AND PERCLOS MEASUREMENT FOR ON BOARD ESTIMATION OF ALERTNESS OF DRIVERS. *Investigación*.
- Google Research Brain Team. (2023). *TensorFlow* [Accessed: October 20, 2023]. <https://www.tensorflow.org/?hl=es-419>
- HAIDER KHALAF JABBAR, D. R. Z. K. (2015). METHODS TO AVOID OVER-FITTING AND UNDER-FITTING IN SUPERVISED MACHINE LEARNING (COMPARATIVE STUDY). *Research*.
- Heredia, A. C. D., González, J. A. L., Barrios, P. E. G., & Ramírez, P. A. R. (s.f.). MUERTE INMEDIATA, PRECOZ Y TARDÍA SECUNDARIA A TRAUMATISMOS POR ACCIDENTES DE TRÁNSITO OCURRIDOS EN EL DEPARTAMENTO DE GUATEMALA EN EL AÑO 2018.
- Inc, Y. L. (2023). pandas-profiling — pypi.org [[Accessed 06-11-2023]].
- Jay D. Fuletra, D. B. (2013). A Survey on Driver's Drowsiness Detection Techniques. *Recent and Innovation Trends in Computing and Communication*.

- Junaedi, S., & Akbar, H. (2018). Driver Drowsiness Detection Based on Face Feature and PERCLOS. *Journal of Physics: Conference Series*, 1090(1), 012037. <https://doi.org/10.1088/1742-6596/1090/1/012037>
- Keras Contributors. (2023). *Keras: The Python Deep Learning API* [Accessed: October 20, 2023]. <https://keras.io/>
- Lab, M. R. (2021a). | MRL — mrl.cs.vsb.cz [[Accessed 23-10-2023]].
- Lab, M. R. (2021b). Drowsiness Detection — kaggle.com [[Accessed 23-10-2023]].
- Massoz, Q., Langohr, T., François, C., & Verly, J. G. (2016). The ULg multimodality drowsiness database (called DROZY) and examples of use. *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1-7. <https://doi.org/10.1109/WACV.2016.7477715>
- Minhas, M. S. (2021). Techniques for handling underfitting and overfitting in Machine Learning [[Accessed 05-11-2023]].
- Olabe, X. B. (2013). *Redes neuronales artificiales y sus aplicaciones*. Escuela Superior de Ingeniería de Bilbao, EHU.
- Pasaribu, N. T. B. (2018). Drowsiness Detection According to the Number of Blinking Eyes Specified From Eye Aspect Ratio Value Modification. *Research*.
- Rondon L, P. F. (2013). Reconocimiento de somnolencia en conductores bajo condiciones simuladas. *Investigación*.
- ROUHIAINEN, L. (2018). *INTELIGENCIA ARTIFICIAL 101 COSAS QUE DEBES SABER HOY SOBRE NUESTRO FUTURO*. alienta.
- Scikit-learn Developers. (2023). *scikit-learn - Machine Learning in Python* [Accessed: October 20, 2023]. <https://scikit-learn.org/stable/>
- Siddharth Sharma, A. A., Simone Sharma. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *Research*.
- Team, F. D. (2023). *Flask* [Accessed: October 20, 2023]. <https://flask.palletsprojects.com/en/3.0.x/>
- Wiechers, D. E. G. (2009). El ojo: estructura y función. *Párpados y conjuntiva • Aparato lagrimal*.

CAPÍTULO 12

Anexos



Anexo 1. Tomas nocturnas y dispositivo empleado para la investigación.