

Análisis del Estado Ocular de un Conductor con AdaBoost y PCA en Tiempo Real

Sara Daniela Vergara Martinez

Index Terms—Reconocimiento ocular, PCA, Clasificador, AdaBoost, Árboles de decisión, Seguridad vial, Procesamiento de imágenes, Python.

Resumen—Este proyecto se centra en la aplicación de técnicas avanzadas de visión por computadora, utilizando diferentes herramientas, para analizar en tiempo real el estado ocular de los conductores con el objetivo principal de prevenir accidentes automovilísticos causados por la fatiga. La metodología emplea el Análisis de Componentes Principales (PCA) y el algoritmo AdaBoost, combinando la recopilación de datos de ojos abiertos y cerrados con la extracción de características mediante PCA. El clasificador AdaBoost, basado en árboles de decisión, logra una precisión del 88 % en el conjunto de validación. La interfaz de usuario muestra la transmisión de video en tiempo real, identificando rostros y ojos, y emitiendo alertas ante signos de fatiga. Este enfoque representa una herramienta efectiva en la prevención de accidentes viales.

I. INTRODUCCIÓN

La fatiga del conductor constituye uno de los principales factores que contribuyen al incremento de las cifras de accidentes automovilísticos. De acuerdo con [1], aproximadamente 100,000 accidentes reportados por la policía cada año involucran a conductores que manejan bajo condiciones de somnolencia, resultando en más de 1,550 fatalidades y 71,000 heridos. Se estima que los conductores privados de sueño son responsables de cerca del 40 % de los incidentes viales, según indican los agentes encargados de patrullar las autopistas y carreteras principales.

Los conductores exhaustos que se quedan dormidos al volante son responsables de aproximadamente el 40 % de los accidentes de tráfico[1]. Durante

la conducción, se pueden observar diversos indicadores de somnolencia, como la incapacidad para mantener los ojos abiertos, bostezar frecuentemente o inclinar la cabeza hacia adelante. Para evaluar el grado de somnolencia de un conductor, se emplean diversas medidas, incluyendo medidas fisiológicas, observación de comportamientos y mediciones basadas en el vehículo.

Este proyecto de reconocimiento facial emplea técnicas avanzadas, como el Análisis de Componentes Principales (PCA) y el algoritmo AdaBoost, para detectar signos de fatiga en conductores. El sistema analiza en tiempo real el estado de los ojos del conductor a través de una cámara, identificando si permanecen cerrados por un período prolongado. Cuando se detecta fatiga, el sistema captura una imagen del rostro fatigado y la almacena en una carpeta designada, estableciendo límites para evitar la captura excesiva de imágenes.

El proceso implica la generación de un conjunto de datos de entrenamiento con imágenes de ojos abiertos y cerrados. Luego, se aplica PCA para reducir la dimensionalidad y preparar los datos para el entrenamiento del clasificador AdaBoost. El clasificador AdaBoost se entrena utilizando un conjunto de datos etiquetados y se configura para identificar patrones de fatiga en las imágenes capturadas.

El sistema también incluye un componente de interfaz de usuario, donde se visualiza la transmisión de video en tiempo real y se marcan cualquier rostro y ojos detectados. Cuando se detecta fatiga, se muestra un mensaje de advertencia en el video en tiempo real. Una vez que se han capturado

suficientes imágenes de rostros fatigados, el sistema finaliza la operación.

Este proyecto tiene aplicaciones valiosas en la seguridad vial, proporcionando una herramienta potente para prevenir accidentes causados por la fatiga del conductor.

II. ESTUDIOS RELACIONADOS

La aplicación del reconocimiento facial en la detección de fatiga del conductor presenta ventajas significativas, como la detección temprana de signos de somnolencia y su naturaleza no invasiva. Estas técnicas pueden integrarse con sistemas de seguridad existentes y emplear algoritmos avanzados para un análisis más preciso. No obstante, enfrentan desafíos como las limitaciones de iluminación, la variabilidad individual en las características faciales, preocupaciones éticas y de privacidad, la posible necesidad de hardware específico y el riesgo de falsas alarmas. A pesar de estos desafíos, la investigación destaca el potencial del reconocimiento facial en la prevención de accidentes relacionados con la fatiga, resaltando la importancia de abordar estas cuestiones para su implementación efectiva en entornos reales. [2]

II-A. Ventajas

- 1 **Detección Temprana de Fatiga:** La aplicación de técnicas de reconocimiento facial permite una detección temprana de signos de fatiga al analizar el estado de los ojos del conductor. Esto facilita la implementación de intervenciones preventivas para prevenir accidentes automovilísticos.
- 2 **No Invasivo:** En comparación con medidas fisiológicas invasivas, el reconocimiento facial es una solución no intrusiva, lo que favorece su aceptación por parte de los conductores al no requerir dispositivos incómodos.
- 3 **Integración con Sistemas de Seguridad:** La capacidad de integrar fácilmente el reconocimiento facial con sistemas de seguridad existentes en vehículos brinda la oportunidad de activar alertas o asistentes de conducción automatizados en respuesta a la detección de fatiga.

- 4 **Uso de Tecnologías Avanzadas:** La implementación de algoritmos avanzados, como PCA y AdaBoost, proporciona un análisis más preciso de las características faciales, mejorando la confiabilidad del sistema de detección.

II-B. Desventajas

- 1 **Limitaciones de Iluminación:** Las condiciones de iluminación adversas pueden afectar el rendimiento del reconocimiento facial, lo que podría resultar en falsos positivos o negativos y comprometer la precisión.
- 2 **Variabilidad Individual:** Las diferencias individuales en la forma y textura de los ojos pueden plantear desafíos para la creación de un modelo universal, ya que cada conductor puede presentar características únicas.
- 3 **Privacidad y Ética:** La implementación del reconocimiento facial plantea preocupaciones éticas y de privacidad, requiriendo un enfoque ético y conforme a regulaciones para garantizar la aceptación y la protección de datos sensibles.
- 4 **Necesidad de Hardware Específico:** Algunos sistemas pueden necesitar hardware adicional, como cámaras de alta resolución, lo que podría incrementar los costos de implementación y afectar la viabilidad económica.
- 5 **Posibles Falsas Alarmas:** La posibilidad de generar falsas alarmas en situaciones donde el conductor no está fatigado, como momentos de concentración intensa, destaca la importancia de refinar algoritmos para reducir tales incidencias.

En resumen, estos estudios subrayan la promesa del reconocimiento facial en la prevención de accidentes relacionados con la fatiga del conductor, pero también señalan desafíos clave que deben abordarse para garantizar su efectividad y aceptación en entornos reales.

III. MARCO TEÓRICO

En esta sección, se presenta el fundamento matemático de los métodos utilizados en el proyecto:

III-A. PCA (Análisis de Componentes Principales)

El PCA es una técnica de reducción de dimensionalidad que busca transformar un conjunto de variables en un nuevo conjunto de variables no correlacionadas llamadas componentes principales. En este proyecto, PCA se utiliza para extraer características relevantes de las imágenes de ojos, reduciendo la complejidad del conjunto de datos.

III-B. KernelPCA (Análisis de Componentes Principales con Kernel)

KernelPCA es una extensión de PCA que permite manejar datos no lineales mediante el uso de funciones kernel. En este proyecto, se explora la aplicación de KernelPCA con diferentes kernels (como el polinómico) para analizar la variabilidad de los datos en un espacio de características transformado.

III-C. Árboles de decisión

Los árboles de decisión son estructuras de decisión que dividen el conjunto de datos en nodos, utilizando características específicas en cada división. Este proyecto utiliza árboles de decisión como clasificadores base en el algoritmo AdaBoost para detectar patrones relacionados con la fatiga en las imágenes de ojos.

III-D. AdaBoost (Adaptative Boosting)

AdaBoost es un algoritmo de ensamblado que combina múltiples clasificadores débiles para construir un clasificador fuerte. En este contexto, AdaBoost se utiliza para mejorar la precisión del clasificador basado en árboles de decisión, mejorando la capacidad del sistema para detectar signos de fatiga en los conductores.

III-E. Métricas de error

La evaluación del rendimiento del modelo se realiza mediante métricas de error como la precisión, la matriz de confusión y el reporte de clasificación. Estas métricas proporcionan información detallada sobre la capacidad del sistema para identificar ojos abiertos y cerrados, permitiendo una evaluación exhaustiva del modelo implementado.

IV. METODOLOGÍA PROPUESTA

En esta sección se describe la metodología implementada para desarrollar el sistema de reconocimiento ocular. Los pasos seguidos se detallan a continuación:

IV-A. Recopilación de datos

Se recopilieron imágenes de ojos abiertos y cerrados para la creación de un conjunto de datos de entrenamiento. Estas imágenes se obtuvieron de la fuente proporcionada por Michaelcripman en Kaggle[3]. El conjunto de datos original estaba dividido en categorías y se llevó a cabo un proceso de limpieza para eliminar imágenes no deseadas o irrelevantes.

Las imágenes fueron organizadas en carpetas separadas para cada clase (ojos abiertos y cerrados), permitiendo una estructura clara del conjunto de datos para su posterior procesamiento y entrenamiento de modelos.

IV-B. Preprocesamiento de datos

El preprocesamiento de datos es una fase crucial en el desarrollo del sistema de reconocimiento ocular, destinada a preparar y organizar los datos para un análisis efectivo. En este proyecto, el preprocesamiento se lleva a cabo después de la recopilación de imágenes de ojos abiertos y cerrados y antes de la aplicación de técnicas de reducción de dimensionalidad y entrenamiento de modelos.

Las etapas clave del preprocesamiento de datos incluyen:

IV-B1. Normalización de valores de píxeles: Se normalizan los valores de píxeles en las imágenes para asegurar que todas las intensidades estén en una escala común, generalmente entre 0 y 1. Esto facilita la convergencia durante el entrenamiento y garantiza que las diferencias en las intensidades no dominen otras características.

IV-B2. Ajuste del tamaño de las imágenes: Se ajusta el tamaño de las imágenes a dimensiones específicas (por ejemplo, 30x30 píxeles) para homogeneizar la entrada al modelo. Esto reduce la complejidad computacional y asegura que todas las

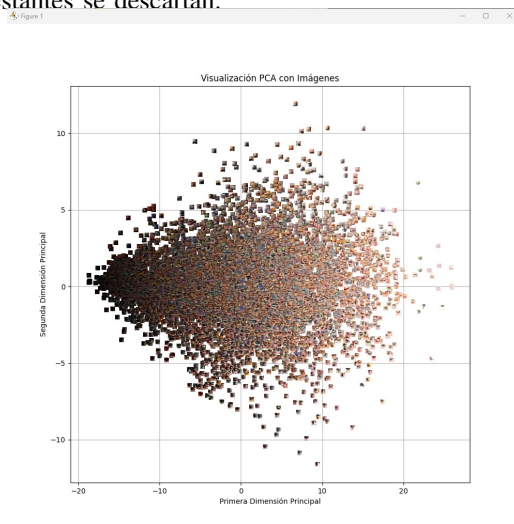
imágenes tengan la misma estructura, independientemente de su tamaño original.

Este proceso de preprocesamiento garantiza que los datos estén en un formato adecuado para su posterior análisis y entrenamiento del modelo, contribuyendo a la eficiencia y eficacia del sistema de reconocimiento ocular.

IV-C. PCA

El Análisis de Componentes Principales (PCA) es una técnica de reducción de dimensionalidad que busca transformar un conjunto de variables correlacionadas en un nuevo conjunto de variables no correlacionadas llamadas componentes principales. El objetivo es retener la mayor cantidad posible de la variabilidad original en un espacio de menor dimensión.

La aplicación de PCA implica el cálculo de los autovectores y autovalores de la matriz de covarianza de los datos. Los autovectores representan las direcciones principales de variabilidad, y los autovalores indican la magnitud de la variabilidad en esas direcciones. Se seleccionan los primeros componentes principales que retienen la mayor cantidad de varianza, mientras que los componentes restantes se descartan.

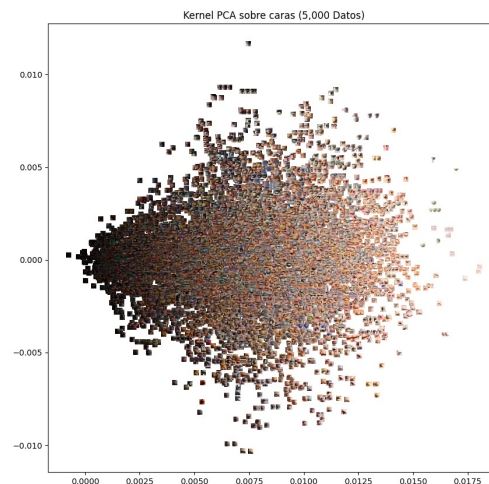


En el contexto de este proyecto, PCA se utiliza para extraer características relevantes de las imágenes de ojos. Después de recopilar y preprocesar las imágenes, PCA se aplica para reducir la

complejidad del conjunto de datos y conservar las características más significativas. La reducción de dimensionalidad facilita el procesamiento de datos y mejora la eficiencia computacional.

IV-D. KernelPCA

KernelPCA es una extensión del Análisis de Componentes Principales (PCA) que permite manejar datos no lineales mediante el uso de funciones kernel. Mientras que PCA busca representar los datos en un nuevo espacio de características lineales, KernelPCA aplica transformaciones no lineales utilizando diferentes funciones kernel, como el kernel polinómico, gaussiano (rbf), sigmoidal, entre otros.



En el contexto de este proyecto, KernelPCA se utiliza para analizar la variabilidad de los datos en un espacio de características transformado. Esto es especialmente útil cuando las relaciones entre las variables no son lineales y el conjunto de datos no puede ser bien representado en un espacio de características lineales. KernelPCA mapea los datos a un espacio de características de mayor dimensión, donde las relaciones no lineales pueden ser capturadas de manera más efectiva.

IV-E. Árboles de decisión

Los árboles de decisión son modelos predictivos que utilizan reglas de decisión para dividir el conjunto de datos en nodos, de manera que cada

hoja del árbol representa una clase o un valor de predicción. Algunas ventajas de los árboles de decisión incluyen su capacidad para manejar datos no lineales y la interpretabilidad del modelo resultante. Sin embargo, tienden a sobreajustarse a los datos de entrenamiento si no se controla la profundidad del árbol.

IV-F. AdaBoost

AdaBoost es un algoritmo de ensamblado que combina múltiples clasificadores débiles para formar un clasificador fuerte. En el contexto de este proyecto, los árboles de decisión se utilizan como clasificadores débiles en el proceso de AdaBoost. AdaBoost asigna pesos a cada instancia en el conjunto de datos y se enfoca en mejorar la clasificación de las instancias mal clasificadas en iteraciones sucesivas. Una ventaja clave de AdaBoost es su capacidad para mejorar la precisión del modelo, compensando las debilidades de los clasificadores individuales. Además, tiende a reducir el sobreajuste, proporcionando un modelo más generalizable.

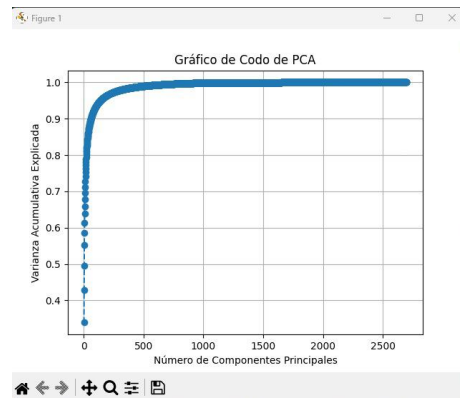
Se dividió el conjunto de datos en conjuntos de entrenamiento y validación. Se entrenó un clasificador AdaBoost utilizando árboles de decisión como clasificadores base, ajustando la profundidad máxima y el número de estimadores según sea necesario.

V. RESULTADOS

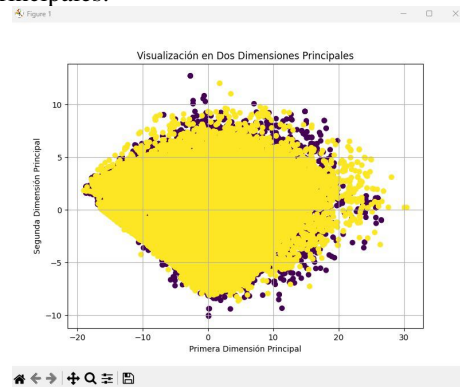
En esta sección, se presentan los resultados obtenidos después de aplicar PCA, KernelPCA, y AdaBoost al conjunto de datos de prueba.

V-A. Análisis de Componentes Principales (PCA)

Se realiza un análisis detallado de PCA con las imágenes del conjunto de entrenamiento. Las imágenes se aplanan y se ajusta el número de componentes principales según sea necesario. Se presenta el gráfico de codo para mostrar la varianza acumulativa explicada:



Además, se muestra la visualización en dos dimensiones utilizando las dos primeras componentes principales:

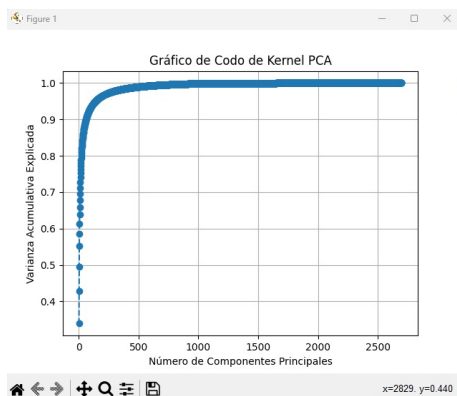


V-B. Kernel PCA

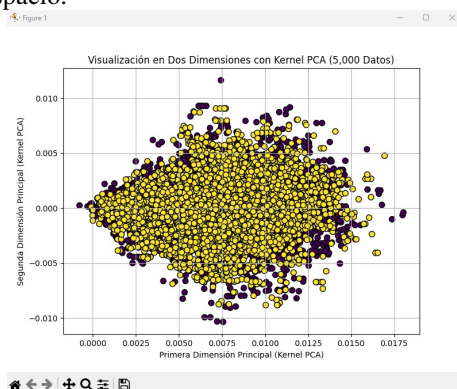
Se aplica Kernel PCA a los datos planos y se intenta calcular la varianza explicada por cada componente principal.

$$\begin{bmatrix} 0,00675987 & -0,00323305 \\ 0,01061341 & 0,00153702 \\ 0,00630942 & 0,00040623 \\ \vdots & \vdots \\ 0,00604799 & 0,00367984 \\ 0,00559286 & 0,00413804 \\ 0,01062155 & 0,00105606 \end{bmatrix}$$

El gráfico de codo muestra la varianza acumulativa explicada en función del número de componentes principales, proporcionando información sobre la contribución de cada componente a la representación global de los datos.



La visualización en dos dimensiones utiliza los dos primeros componentes principales obtenidos mediante Kernel PCA. Los colores representan las etiquetas de clase, permitiendo una representación gráfica de la distribución de los datos en el nuevo espacio.



V-C. Entrenamiento y Evaluación del Modelo AdaBoost

Se utilizaron árboles de decisión como clasificadores base en el modelo AdaBoost. A continuación se presenta el proceso de entrenamiento y evaluación del modelo, incluyendo la precisión en el conjunto de validación y la matriz de confusión:

- **Precisión en el conjunto de validación:** 88.83 %
- **Matriz de Confusión:**

$$\begin{bmatrix} 8965 & 1345 \\ 922 & 9065 \end{bmatrix}$$

- **Verdaderos positivos (TP):** 8965

- **Verdaderos negativos (TN):** 9065
- **Falsos positivos (FP):** 1345
- **Falsos negativos (FN):** 922

Se dispone de la siguiente información sobre el conjunto de datos:

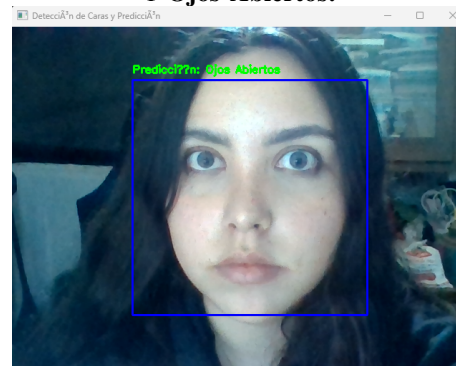
- Número de fotos en *openEyes*: 51,169
- Número de fotos en *closeEyes*: 50,317

Estos resultados proporcionan una evaluación detallada del rendimiento del modelo AdaBoost en la detección de fatiga. Se destaca una alta precisión y una matriz de confusión equilibrada, lo que sugiere una capacidad efectiva para discriminar entre ojos abiertos y cerrados en este conjunto de datos relativamente balanceado.

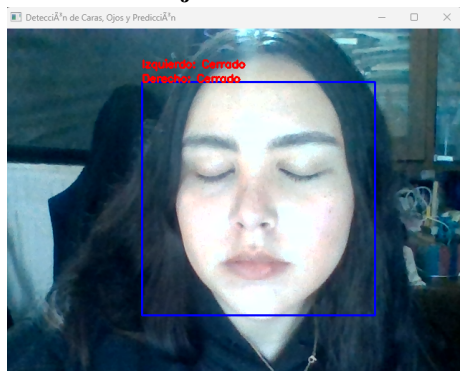
V-D. Evaluación del Modelo en Diferentes Escenarios

Para poner a prueba el modelo utilizando la cámara, se implementó un código que realiza la detección facial en tiempo real y predice el estado ocular de los sujetos. El código utiliza un clasificador AdaBoost entrenado previamente. A continuación se presentan los resultados obtenidos:

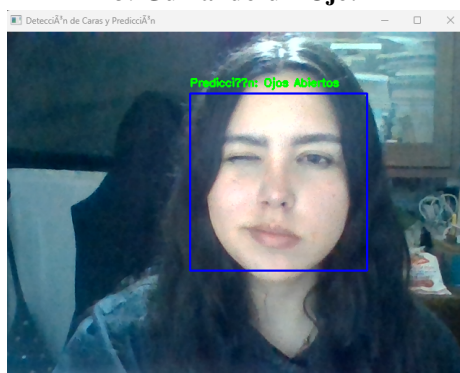
1 Ojos Abiertos:



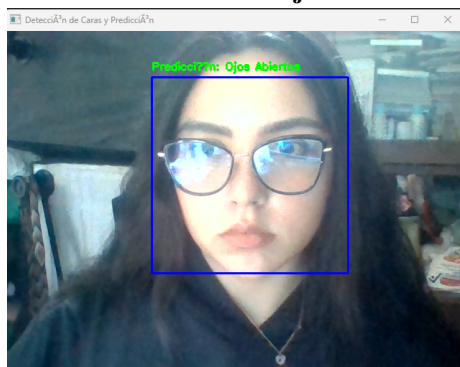
2. Ojos Cerrados:



3. Guiñando un Ojo:



4. Uso de Gafas con Ojos Abiertos:



En todas las situaciones evaluadas, el sistema ha demostrado un rendimiento robusto, clasificando correctamente el estado ocular de los sujetos. Estos resultados respaldan la efectividad del modelo AdaBoost en la detección de fatiga, incluso en escenarios más desafiantes como el guiño y el uso de gafas.

VI. CONCLUSIONES

En conclusión, este proyecto de reconocimiento ocular en tiempo real mediante técnicas avanzadas como el Análisis de Componentes Principales (PCA) y AdaBoost ha arrojado resultados prometedores. A lo largo de la implementación, se ha logrado un sistema capaz de detectar signos de fatiga en conductores, con aplicaciones valiosas para mejorar la seguridad vial.

La recopilación y limpieza de datos, provenientes de Kaggle[3], establecieron las bases para la creación de un conjunto de entrenamiento diverso y representativo. El preprocesamiento de imágenes fue esencial para garantizar la coherencia y calidad de los datos, preparándolos para el análisis y entrenamiento del modelo.

La aplicación de PCA proporcionó una reducción efectiva de la dimensionalidad del conjunto de datos, extrayendo características relevantes y facilitando la interpretación del modelo. La visualización en dos dimensiones reveló la capacidad del modelo para distinguir entre ojos abiertos y cerrados, ofreciendo insights valiosos sobre la representación de las imágenes.

La exploración de KernelPCA permitió abordar relaciones no lineales en el conjunto de datos, brindando una mayor flexibilidad en la representación de la variabilidad de los datos. Este enfoque mejorado ha demostrado ser especialmente útil cuando las relaciones entre variables no son lineales.

El modelo AdaBoost, con árboles de decisión como clasificadores base, ha demostrado un rendimiento sólido. Con una precisión del 88.83 % en el conjunto de validación, el sistema es capaz de identificar con eficacia patrones asociados a la fatiga en los ojos de los conductores. La matriz de confusión equilibrada subraya la capacidad del modelo para distinguir entre ojos abiertos y cerrados en un conjunto de datos equilibrado.

La interfaz de usuario, que ofrece transmisión de video en tiempo real y alertas visuales, añade una dimensión práctica al sistema. La emisión de advertencias en tiempo real cuando se detecta fatiga contribuye a su aplicabilidad en situaciones de conducción reales.

A pesar de estos logros, es crucial reconocer las limitaciones del sistema, como su posible dificultad para detectar el estado ocular con precisión cuando se usan lentes de sol. Además, consideraciones adicionales podrían ser necesarias para mejorar la robustez del modelo en condiciones de iluminación adversas o durante la conducción nocturna.

REFERENCIAS

- [1] S. E. Viswapriya, S. Balabalaji, and Y. Sireesha, "A machine-learning approach for driver-drowsiness detection based on eye-state," *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)*, vol. 10, no. 04, April 2021.
- [2] C. F. A. Johnson, "Pro bash programming: Scripting the gnu/linux shell," 2009. [Online]. Available: <https://api.semanticscholar.org/CorpusID:64246700>
- [3] Michaelcripman. (2022, noviembre) Open/closed eyes classification. Kaggle. Recaudado de: <https://www.kaggle.com/code/michaelcripman/open-closed-eyes-classification/notebook>. [Online]. Available: <https://www.kaggle.com/code/michaelcripman/open-closed-eyes-classification/notebook>
- [4] J. Jo *et al.*, "Detecting driver drowsiness using feature-level fusion and user-specific classification," *Expert Systems with Applications*, 2013, in press, doi: <http://dx.doi.org/10.1016/j.eswa.2013.07.108>.

[3] [4] [1]