



Inteligencia artificial y su aplicación en la seguridad vial

José Anibal Arias Aguilar

Universidad Tecnológica de la Mixteca, División de Estudios de Posgrado, Heroica Ciudad de Huajuapán de León, Oaxaca, México.

ORCID: 0000-0002-8838-877X

Eduardo Aragón López

Universidad Tecnológica de la Mixteca, División de Estudios de Posgrado, Heroica Ciudad de Huajuapán de León, Oaxaca, México.

ORCID: 0009-0009-2787-9282

Recepción: 11 de noviembre de 2024.

Aceptación: 28 de marzo de 2025.

Septiembre 2025 • número de revista 13 • DOI: 110.22201/dgtic.26832968e.2025.13.53

Inteligencia artificial y su aplicación en la seguridad vial

Resumen

Debido a los altos índices de mortandad derivados de la conducción distraída, la cual es ocasionada por el desvío de la atención de la conducción segura, de la carretera y del entorno del conductor, es que surgen los Sistemas de Asistencia a la Conducción (ADAS, por sus siglas en inglés) como una forma de prevenir estos siniestros y asistir al conductor durante su trayecto. En el presente trabajo, se detectan y clasifican la conducción segura y nueve actividades distractoras en los conductores de automóviles mediante el uso de redes neuronales convolucionales, aplicadas primero a la base de datos "State Farm Distracted Driver Detection" y también a una base de datos propia. Además, con el uso de la biblioteca "TensorFlow Lite", el modelo se implementa en una aplicación móvil para su uso en tiempo real. El modelo resultante logra una exactitud del 91% en entrenamiento y del 95% en su validación. Con la implementación móvil, la exactitud del modelo se reduce en un 5%, sin embargo, se logran detectar y diferenciar correctamente las distracciones y se demostró la posibilidad de adaptar un ADAS en un dispositivo móvil.

Palabras Clave: ADAS, inteligencia artificial, redes neuronales, accidentes viales, dispositivos móviles.

Artificial intelligence and its application in road safety

Abstract

Due to the high mortality rates resulting from distracted driving, which is caused by the diversion of attention from safe driving, the road and the driver's environment, Advanced Driver Assistance Systems (ADAS) emerge to prevent these accidents and assist the driver during their journey. In this work, safe driving and nine distracting activities in car drivers are detected and classified by using convolutional neural networks, applied first to the "State Farm Distracted Driver Detection" database and also to a self-created database. In addition, with the use of the "TensorFlow Lite" library, the model is implemented in a mobile application for real-time use. The resulting model achieves an accuracy of 91% in training and 95% in validation. In the mobile implementation, the accuracy of the model is reduced 5%, however, distractions are correctly detected and differentiated and the possibility of adapting an ADAS in a mobile device was demonstrated.

Keywords: ADAS, artificial intelligence, neural networks, traffic accidents, mobile devices.

Introducción

De acuerdo con la Organización Mundial de la Salud [1], en su informe sobre la situación de la seguridad vial 2023, se registraron alrededor de 1.19 millones de muertes por año desde el 2010 hasta la actualidad, siendo una de las principales causas de muerte de niños y jóvenes de entre los 5 y 29 años de edad. Debido al incremento del parque vehicular, se han desarrollado medidas para prevenir los altos índices de mortandad causados por los accidentes viales, entre ellos, encontramos a los Sistemas de Asistencia a la Conducción capaces de auxiliar a los conductores en la tarea compleja que conlleva la conducción segura. Sin embargo, los avances que han tenido estos sistemas por parte de la industria se han centrado únicamente en los modelos más recientes y caros, es decir, entre más funciones de seguridad contenga mayor será el costo del vehículo. Por otro lado, las investigaciones desarrolladas por parte de la comunidad científica se han basado principalmente en sistemas de visión artificial, los cuales buscan asistir al conductor mediante su monitorización con cámaras.

Liu et al.[2] utilizan redes neuronales convolucionales aplicadas a la visión por computadora; dentro de su marco propuesto, incluyen, en primer lugar, una red convolucional maestra (red preentrenada) que le transmitirá su conocimiento a una red estudiante; en conjunto, son capaces de detectar 9 distracciones con una exactitud del 95.64%. Por otro lado, Mbouna et al. [3] presentaron un análisis basado en el estado de apertura de los ojos y la postura de la cabeza para detectar signos de somnolencia, utilizando Máquinas de Soporte Vectorial (SVM) para hacer la clasificación entre distracción o conducción. Similarmente, en Fan y Shangbing [4], se trabaja con una red de fusión de características locales multiescala basada en imágenes para detectar 9 distracciones, obteniendo un 90.25% de exactitud. Por otra parte, en Xing et al. [5], se realizó un modelo que clasifica 3 actividades distractoras y otras 4 actividades de conducción normal; mediante la red preentrenada Alexnet, alcanzaron un 79% de exactitud en la clasificación.

Desarrollo

Materiales y métodos

En el presente estudio, se detalla la metodología de investigación llevada a cabo para la implementación de un Sistema de Asistencia a la Conducción (ADAS, por sus siglas en inglés) en un dispositivo móvil con sistema operativo Android, con el objetivo de monitorear al conductor y prevenirlo de actividades distractoras y potencialmente peligrosas. De esta manera, el sistema busca prevenir el riesgo de sufrir accidentes viales activando una alerta sonora cuando éste mismo, por medio del monitoreo en tiempo real, logre detectar una de las distracciones propuestas.

Para la captura de las fotografías con las que se construyó la base de datos, se trabajó con conductores de automóviles en un ambiente controlado a la luz de día, mientras que, para realizar los entrenamientos de las redes neuronales convolucionales, se utilizó un laboratorio computacional.

Para poder implementar el modelo de red neuronal convolucional en un dispositivo móvil, se requirió de un dispositivo con sistema operativo Android (versión indistinta) con el modo de depuración y desarrollador activados, necesarios para poner en funcionamiento el sistema por medio de una aplicación desarrollada en Android Studio.

Estrategias de desarrollo

De acuerdo con Aguirre Barrera y Aguirre Barrera [6], la metodología en cascada es uno de los modelos que incluye ciertas ventajas en la realización de proyectos de software y de aplicaciones móviles, debido a que se organiza por medio de fases y cada una de ellas cuenta con entregables específicos; además, Velásquez et al [7] menciona que, con este tipo de metodología, resulta más sencillo realizar una planificación general del proyecto dado que, desde la primera fase, se acuerdan tanto los requerimientos como los recursos necesarios del mismo y, en las siguientes fases, se procede únicamente a llevar a cabo las actividades de desarrollo y prueba correspondientes con los recursos previamente planteados.

En este contexto, el ADAS se desarrolló siguiendo la metodología en cascada, destacando que esta metodología propone un orden y seguimiento riguroso de sus fases. Por tal motivo, fue necesario llevar a cabo una modificación en la Fase 4 (ver Fig. 1), en la cual, se lleva a cabo un proceso iterable correspondiente a los entrenamientos de los modelos de redes neuronales convolucionales y la búsqueda de los hiperparámetros del modelo óptimo.

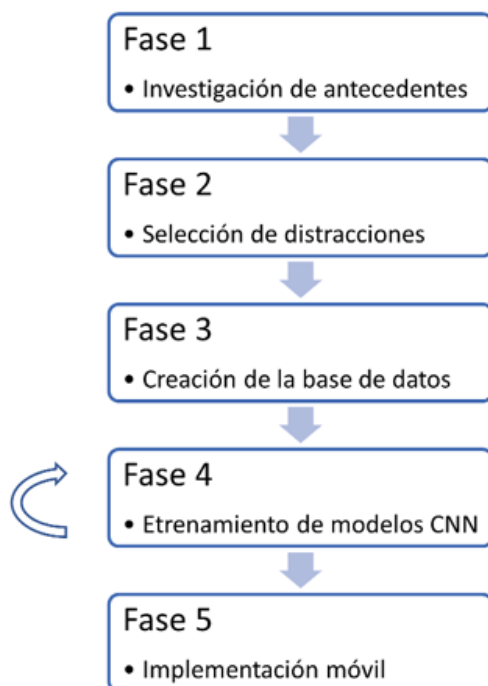


Fig. 1. Metodología utilizada para el desarrollo del proyecto.

Tomando en consideración la investigación realizada durante la Fase 1, se llegó a la conclusión de utilizar técnicas de aprendizaje profundo, más concretamente, utilizar redes neuronales convolucionales debido a su uso particularmente eficaz en las tareas de extracción de características y clasificación de imágenes. De acuerdo con Artola Moreno y Pérez Carrasco [8], este tipo de redes neuronales artificiales optimizan las tareas de análisis de las imágenes por medio de la aplicación de filtros convolucionales encargados de extraer las características más representativas de cada imagen para su interpretación y clasificación.

En las Fases 2 y 3, se analizaron bases de datos con imágenes de conductores de automóvil realizando actividades distractoras. Como demuestran los estudios presentados en los trabajos de la introducción, las actividades distractoras más peligrosas fueron aquellas que involucran el uso de dispositivos móviles y las que incluyen apartar una o ambas manos del volante; motivo por lo cual, y a modo de comparación de resultados, se optó por trabajar con las distracciones presentadas en la Tabla I.

Tabla I
Selección de distracciones

Código	Nombre
Clase 0	Conducción segura
Clase 1	Enviando SMS con mano derecha
Clase 2	Hablando por teléfono con mano derecha
Clase 3	Enviando SMS con mano izquierda
Clase 4	Hablando por teléfono con mano izquierda
Clase 5	Manipulando la radio/GPS
Clase 6	Ingiriendo bebidas
Clase 7	Buscando algo en la parte de atrás
Clase 8	Maquillándose/Arreglándose el cabello
Clase 9	Hablando con el pasajero

Para las primeras pruebas, se trabajó con la base de datos que se encuentra disponible en la página de Kaggle [9], la cual contiene 102,150 imágenes dentro de 10 clases diferentes; las imágenes cuentan con un tamaño de 640x480 y, durante el entrenamiento de los modelos de red, se redimensionaron a un tamaño de 48x48 para reducir el costo computacional de los cálculos matemáticos; en cada uno de los entrenamientos, se modificaron los hiperparámetros de los modelos hasta obtener los óptimos (ver Tabla II).

Tabla II

Mejores hiperparámetros con la base de datos de prueba.

Hiperparámetros	
Tasa de aprendizaje	0.0001
Tamaño del lote	200
Número de épocas	25

Durante la Fase 4, se capturaron fotografías con imágenes de conductores con la finalidad de desarrollar una base de datos propia y, de esta manera, adaptar el modelo para su uso en un contexto local, en conductores mexicanos (ver Fig. 2). Ésta cuenta con 10,000 imágenes que fueron etiquetadas y redimensionadas a un tamaño de 384x384 en formato RGB.



Fig. 2. Base de datos generada de las distracciones a detectar en conductores mexicanos.

Para los entrenamientos de los modelos de red, se utilizó la “transferencia de aprendizaje” con la finalidad de solventar la disminución de la cantidad de datos. Durante las pruebas con otros modelos, el modelo de red preentrenado que obtuvo mejores resultados fue la arquitectura de red “EfficientNet b1”. Éste es un modelo de red neuronal convolucional ligero en tamaño de almacenamiento (30.1 Mb) y en cantidad de parámetros entrenables (7,794,184); con éste, se alcanzó una mayor exactitud dentro de los entrenamientos, motivos por los cuales, se optó por utilizarlo debido a que uno de los requerimientos principales es que el modelo resultante sea lo más ligero posible en tamaño de almacenamiento debido a que estará almacenado dentro de una aplicación Android.

Este modelo realiza un método de escalamiento que ajusta uniformemente todas las dimensiones de profundidad, ancho y resolución de las imágenes de entrada, fue entrenado para clasificar 1,000 clases distintas a una resolución de 240x240 y está disponible para su uso libre por investigadores en el repositorio de redes preentrenadas de la librería de TensorFlow. Su arquitectura se muestra en la Figura 3.

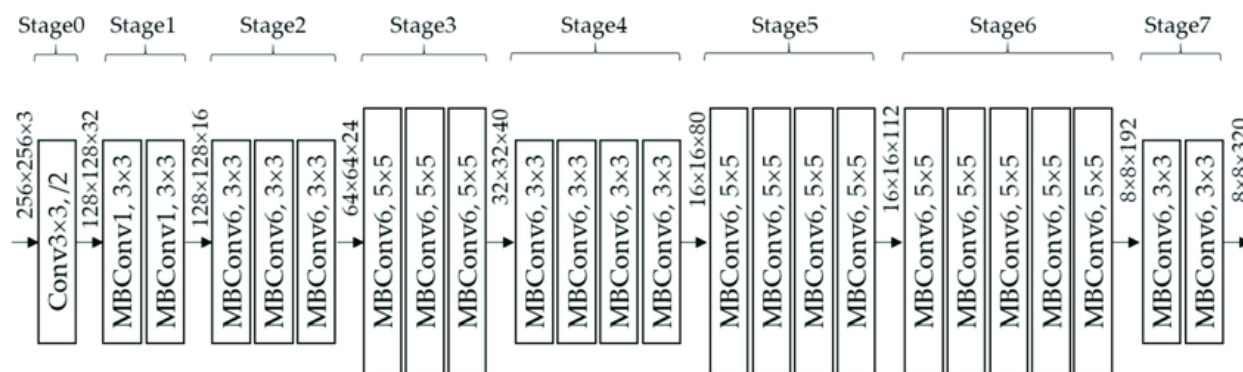


Fig. 3. Arquitectura del modelo "EfficientNet b1".

En la Fase 5, a pesar de contar con un modelo relativamente ligero, aún es necesario realizar un proceso de conversión con la herramienta “TensorFlow Lite”, la cual permite implementar los modelos de aprendizaje de forma eficiente en dispositivos con recursos de procesamiento y memoria limitados. Así pues, en comparación con el formato del modelo original, éste cuenta con un tamaño de almacenamiento aún más reducido y ayuda a lograr una inferencia más rápida, lo que logra reducir el consumo de energía para su uso en este

tipo de dispositivos. Esta misma aplicación cumple la función de detectar y realizar la predicción de clasificar, en tiempo real, las acciones del conductor; en busca de una de las 10 clases con las que fue entrenado el modelo, mediante un cuadro delimitador, enmarca al conductor al mismo tiempo que indica la etiqueta con la clase detectada junto con su grado de confianza (ver Fig. 4).

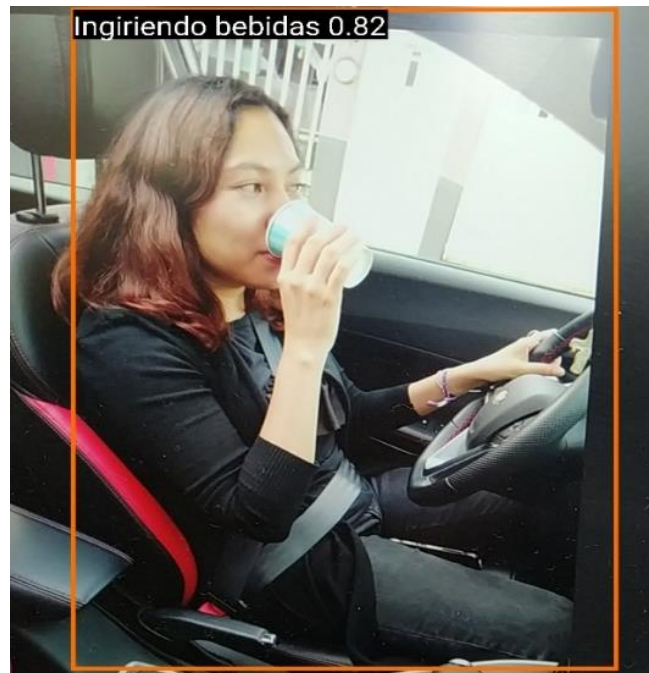


Fig. 4. ADAS en funcionamiento.

La aplicación incluye una interfaz de configuración en la que se recomienda que sólo una persona capacitada pueda manipularlo, debido a que, en ella, se puede incrementar o disminuir el umbral de detección del sistema. Lo anterior nos puede ayudar tanto a clasificar con menos errores el tipo de distracción y la clase a la que pertenece la detección, como a poder detectar, de manera menos precisa pero más rápida, una distracción. Los valores predeterminados y los recomendados de la interfaz de configuración se muestran en la Figura 5.

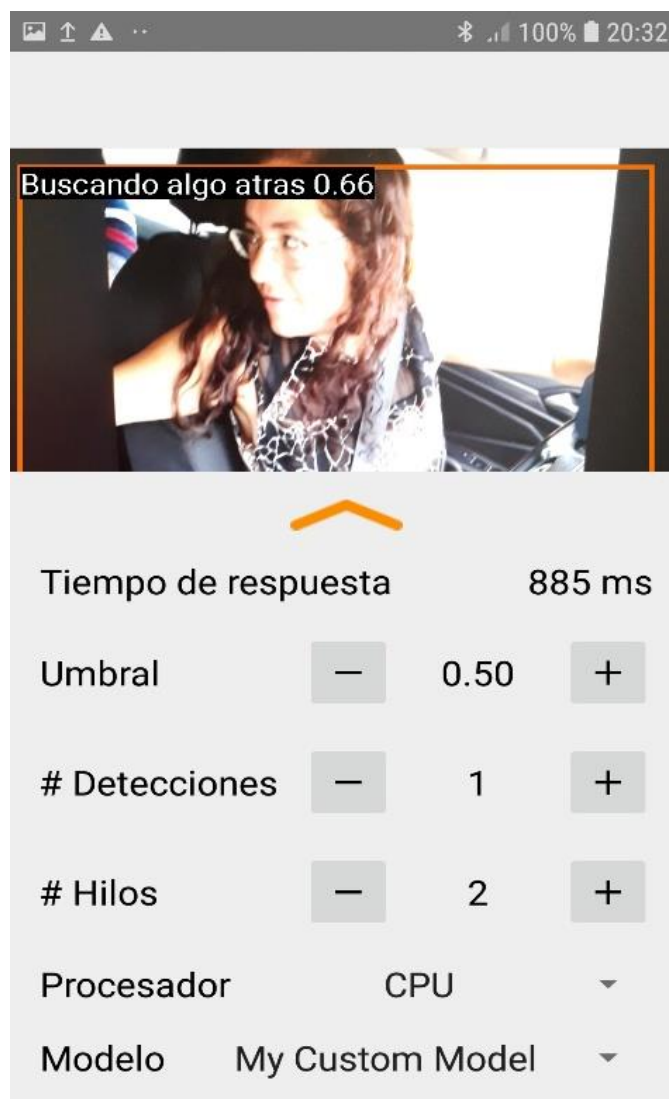


Fig. 5. Interfaz de configuración de la aplicación.

El ADAS también incorpora un sistema de advertencia, el cual se activa una vez que detecta una de las actividades distractoras y mediante un audio le indica al conductor la recomendación “Por favor, no te distraigas”.

Finalmente, para evaluar los resultados obtenidos por los modelos, se llevó a cabo una comparación entre el porcentaje de exactitud obtenido en cada modelo, el desempeño en sus gráficas de exactitud y pérdida, y el análisis de la matriz de confusión para problemas de clasificación multiclase.

Resultados

El primer paso para evaluar el desempeño del ADAS se dio mediante el cálculo de la exactitud y pérdida de los modelos de aprendizaje; una vez comparados, se seleccionó el modelo con el mejor resultado en exactitud. La Tabla III contiene los resultados obtenidos del modelo entrenado con el conjunto de datos [9], mientras que la Tabla IV contiene los resultados que se obtuvieron con el conjunto de datos que fue creada utilizando la transferencia de aprendizaje con el modelo preentrenado “Efficient Net b1”.

Tabla III

Resultados obtenidos con el conjunto de datos de prueba.

Entrenamiento	Exactitud	0.93
	Pérdida	0.21
Validación	Exactitud	0.94
	Pérdida	0.095

Tabla IV

Resultados obtenidos con el conjunto de datos creada y con transferencia de aprendizaje.

Entrenamiento	Exactitud	0.91
	Pérdida	0.41
Validación	Exactitud	0.95
	Pérdida	0.25

Como se puede apreciar en la Tabla IV, el modelo óptimo que se obtuvo fue al utilizar el modelo preentrenado “EfficientNet b1” con la base de datos creada; dicho modelo alcanza una exactitud de 95%.

Las gráficas de la Figura 6 y la Figura 7 corresponden a las curvas de exactitud y pérdida generadas durante el aprendizaje del modelo óptimo. Es importante utilizarlas como método de evaluación debido a que son un indicador de un correcto o mal aprendizaje por parte del modelo: si las curvas de entrenamiento y validación son similares entre sí, como en la Figura 6 y la Figura 7, se demuestra un correcto aprendizaje y se evita de problemas como el “sobreajuste”.

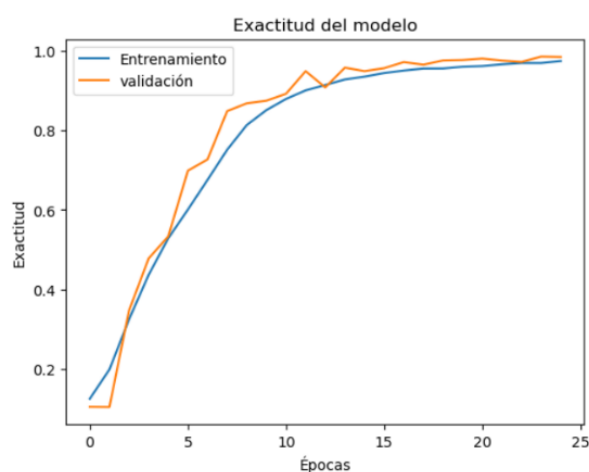


Fig. 6. Gráfica de exactitud del modelo.

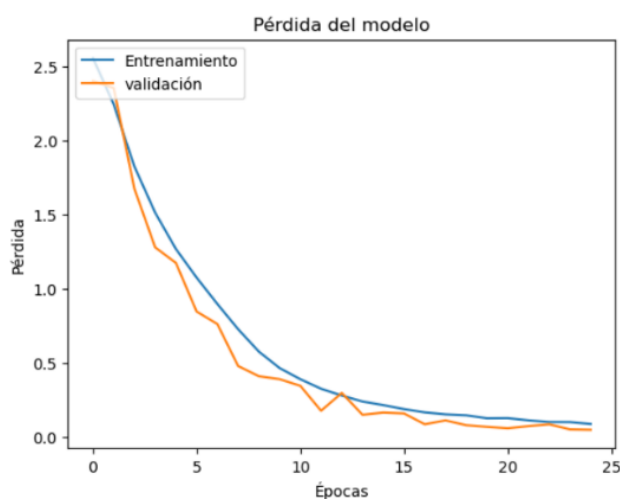


Fig. 7. Gráfica de pérdida del modelo.

El modelo resultante es capaz de clasificar 10 clases diferentes, enlistadas en la Tabla 1. La clase 0 “Conducción segura” corresponde a la clase por defecto a la cual deben de ir todos los conductores de automóvil y, por consiguiente, es la clase en la cual el sistema no emitirá ninguna alerta; por otro lado, las demás clases corresponden a distracciones potencialmente peligrosas.

Finalmente, como parte del análisis del desempeño del modelo, se utilizó una matriz de confusión (ver Fig. 8) en la que se pudo obtener que la clase que mejor es clasificada por el sistema corresponde a la clase 1 “Enviando SMS con mano derecha”, mientras que la clase más difícil de diferenciar entre las demás corresponde a la clase 8 “Maquillándose/Arreglándose el cabello”.

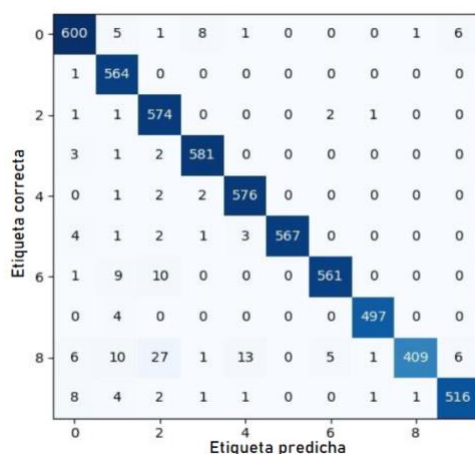


Fig. 8. Matriz de confusión obtenida del modelo óptimo en donde se aprecia un óptimo desempeño de clasificación.

Es importante destacar que, basado en la evaluación de los resultados, se obtuvo un modelo óptimo equiparable con los presentados en el estado del arte, los cuales reportan un aproximado del 90% en la exactitud de sus modelos. Aunado a esto, se cumplió el propósito de innovación, que consiste en implementar un modelo de aprendizaje profundo en un dispositivo móvil. A modo de resumen, la Tabla V muestra una comparación entre los modelos del estado del arte y la propuesta desarrollada en este artículo. En este sentido, los modelos del estado del arte son capaces de clasificar dos o más distracciones, sin embargo, carecen de una implementación real (en campo) del modelo.

De esta manera, los resultados obtenidos con la implementación revelan que el modelo presentado es equiparable con las soluciones ya propuestas. Finalmente, cabe aclarar que el sistema se ve influenciado por diversos factores que afectan su desempeño, tales como los cambios de iluminación y el movimiento constante de la cámara dentro del vehículo.

Tabla V

Comparación del ADAS con el estado del arte

Proyecto	Distracciones analizadas	Tecnología empleada	Evaluación del modelo
[2]	Detección de 9 distracciones	Redes neuronales convolucionales	Exactitud del 95.64%, no implementado
[3]	Detección de somnolencia	Máquinas de Soporte Vectorial	Alta precisión
[4]	Detección de 9 distracciones	Redes neuronales convolucionales	Exactitud del 90.25%, no implementado
[5]	Detección de 3 distracciones	Redes neuronales convolucionales	Exactitud del 79%, no implementado
ADAS	Detección de 9 distracciones	Redes neuronales convolucionales	Exactitud del 95%, implementado en un dispositivo móvil

Conclusiones y trabajo a futuro

El alto índice de accidentes viales derivados de una distracción por parte del conductor se ha convertido en un tema de gran relevancia para la industria y para la comunidad científica, quienes se han dedicado a investigar posibles soluciones que puedan aminorar esta problemática. La mayoría de los sistemas propuestos se centran en analizar sólo alguna distracción en específico, mientras que otras conllevan técnicas más complejas y difíciles de

reproducir, mismas que implican un aumento en los costos económicos como computacionales. Este artículo abordó la implementación de un ADAS en un dispositivo móvil mediante la metodología en cascada, cuyo objetivo es poder contribuir a la detección y alerta de distracciones en conductores, buscando así ayudar a disminuir el riesgo de sufrir un accidente de tránsito. Los resultados obtenidos en la implementación del modelo demuestran haber obtenido resultados equiparables con los trabajos mencionados en el estado del arte, los cuales se centraron en analizar una o dos distracciones, además de que es destacable que en ninguno de los trabajos propuestos se llevó a cabo una implementación (en un dispositivo móvil) como la propuesta en este trabajo.

Como parte del trabajo a futuro, se tiene contemplado trabajar mediante un análisis temporal de las distracciones con el uso de redes neuronales recurrentes, además de incorporar distracciones más complejas tales como la somnolencia y el estado de ebriedad.

Referencias

- [1] Informe sobre la situación mundial de la seguridad vial 2023 [Global status report on road safety 2023]. Ginebra: Organización Mundial de la Salud; 2024. Licencia: CC BY-NC-SA 3.0 IGO. [En línea]. Disponible: <https://www.who.int/es/publications/i/item/9789240086517>
- [2] D. Liu, T. Yamasaki, Y. Wang, K. Mase y J. Kato, "Toward Extremely Lightweight Distracted Driver Recognition With Distillation-Based Neural Architecture Search and Knowledge Transfer," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 764-777, Jan. 2023, doi: 10.1109/TITS.2022.3217342
- [3] R. Oyini Mbouna, S. G. Kong y M. G. Chun, "Visual Analysis of Eye State and Head Pose for Driver Alertness Monitoring," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1462-1469, Sept. 2013, doi: 10.1109/TITS.2013.2262098.
- [4] L. S. Fan y G. Shangbing, "Optimally-Weighted Multi-Scale Local Feature Fusion Network for Driver Distraction Recognition," in *IEEE Access*, vol. 10, pp. 128554-128561, 2022, doi: 10.1109/ACCESS.2022.3224585.

- [5] Y. Xing et al., "End-to-End Driving Activities and Secondary Tasks Recognition Using Deep Convolutional Neural Network and Transfer Learning," 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 2018, pp. 1626-1631, doi: 10.1109/IVS.2018.8500548.
- [6] J. Aguirre Barrera y S. Aguirre Barrera, "Metodologías para el desarrollo de proyectos", Unicatólica, 2021. [En línea]. Disponible: <https://repository.unicatolica.edu.co/handle/20.500.12237/2037>
- [7] S. M. Velásquez, J. D. Vahos Montoya, M. E. Gómez Adasme, E. J. Restrepo Zapata, A. A. Pino, y S. Londoño Marín, "Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software", in Rev. Cintex, vol. 24, n.º 2, pp. 13-23, dic. 2019.
- [8] A. Artola Moreno y J. A. Pérez Carrasco. "Clasificación de imágenes usando redes neuronales convolucionales en Python". Trabajo de fin de grado en Ingeniería de las Tecnologías de Telecomunicación, Universidad de Sevilla, Sevilla, 2019. [En línea]. Disponible: <https://biblus.us.es/bibing/proyectos/abreproy/92402/fichero/TFG-2402-ARTOLA.pdf>
- [9] Kaggle. "State Farm Distracted Driver Detection". 2024 [En línea]. Disponible: <https://www.kaggle.com/competitions/state-farm-distracted-driver-detection/data>