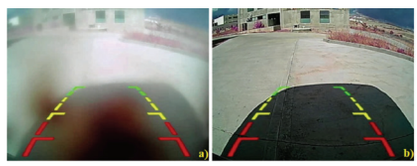
Redes neuronales convolucionales: un enfoque para la detección de obstrucción visual en cámaras de reversa automotrices



Convolutional neural networks: an approach for visual obstruction detection in automotive reversing camer

Luis C. Reveles-Gómez, Huizilopoztli Luna-García, José M. Celaya-Padilla and Rosa A. García-Hernández
Universidad Autónoma de Zacatecas. Unidad Académica de Ingeniería Eléctrica. Jardín Juárez 147. Centro - 98000 Zacatecas (México).

DOI: https://doi.org/10.6036/10865 | Recibido: 28/feb/2023 • Inicio Evaluación: 02/mar/2023 • Aceptado: 08/jun/2023

To cite this article: CONVOLUTIONAL NEURAL NETWORKS: AN APPROACH FOR VISUAL OBSTRUCTION DETECTION IN AUTOMOTIVE REVERSING CAMER. *DYNA.* March – April 2024, vol. 99, n.2, pp. 181-187. DOI: https://doi.org/10.6036/10865

ABSTRACT

- In recent years, the study of Artificial Intelligence in the automotive industry has led to the design of intelligent systems applied to road safety, highlighting the importance of improving road safety worldwide, and thus reducing the number of accidents annually. One of the main functions of these systems is, for example, pedestrian detection, which is performed by cameras and radar-type sensors, among others. However, environmental factors cause visibility problems and obstructions that make pedestrian detection difficult and lead to collisions. With the purpose of contributing to the solution of the exposed problem, two case studies using Convolutional Neural Networks are applied in this research. The first using a pre-trained model (Inception V3) and the second, a proposed model (RvIsNet) to detect dirt on the lens of a vehicle's reverse camera. These types of factors directly affect visibility, which leads to an increased risk of collision when reversing the vehicle. Applying a general data mining methodology, we obtained a result of 0.9549 and 0.9416 accuracy, respectively, for the models used.
- **Keywords:** Convolutional Neural Networks, Classification, Obstruction, Detection, Reversing camera, Inception V3.

RESUMEN

En los últimos años, el estudio de la Inteligencia Artificial en la industria automotriz ha dado lugar al diseño de sistemas inteligentes aplicados a la seguridad vial, destacando la importancia de mejorar la seguridad vial en todo el mundo, y reducir así el número de accidentes anuales. Una de las principales funciones de estos sistemas es, por ejemplo, la detección de peatones, que se realiza mediante cámaras y sensores tipo radar, entre otros. Sin embargo, factores ambientales provocan problemas de visibilidad y obstrucciones que dificultan la detección de peatones y provocan colisiones. Con el fin de contribuir a la solución del problema expuesto, en esta investigación se aplican dos casos de estudio utilizando Redes Neuronales Convolucionales. El primero utilizando un modelo pre-entrenado (Inception V3) y el segundo, un modelo propuesto (RvIsNet) para detectar suciedad en la lente de la cámara de marcha atrás de un vehículo. Este tipo de factores afectan directamente a la visibilidad, lo que conlleva un mayor riesgo de colisión al dar marcha atrás el vehículo. Aplicando una metodología general de minería de datos, obtuvimos un resultado de 0,9549 y 0,9416 de exactitud, respectivamente, para los modelos utilizados.

Palabras clave: Redes neuronales convolucionales, clasificación, obstrucción, detección, cámara de reversa, Inception V3.

1. INTRODUCCIÓN

Los traumatismos debidos a los accidentes de tránsito constituyen la principal causa de defunción entre los jóvenes con edades comprendidas entre los 5 y 29 años, a nivel mundial [1]. El 46% de las personas que fallecen en el mundo a consecuencia de accidentes de tránsito son peatones, ciclistas, conductores o motociclistas [2]. Reducir las cifras y las consecuencias de los accidentes de tráfico, es una tarea relevante para los gobiernos y los investigadores, de acuerdo con Pérez Requena [3]. Es importante contribuir en la disminución de muertes derivadas de accidentes automovilísticos desde el enfoque tecnológico. El avance en las Tecnologías de la Información y Comunicación (TICs), ha originado el uso e implementación de nuevas herramientas tecnológicas. Por ejemplo, la Inteligencia Artificial (IA) en la industria automotriz [4]. Se han desarrollado propuestas tecnológicas, con el fin de contribuir en la disminución del número de accidentes viales por colisiones con peatones u objetos del entorno.

La detección de peatones en los últimos años se ha convertido en un tema importante de investigación dentro de la industria. La aplicación de técnicas de Visión Artificial [5] y el Deep Learning (DL) [6] mismas que se derivan de la IA, se ven presentes para dar solución a los retos que se presentan para detectar peatones. Entre los retos más relevantes se encuentran los problemas de visibilidad, oclusión, pose, baja resolución en imágenes, distorsión y problemas al detectar formas en pequeña escala, efectos del clima, poca cantidad de imágenes lo que dificulta la evaluación en tiempo real [7]. Sin embargo, uno de los principales desafíos es la visibilidad de las cámaras de los vehículos y especialmente en la marcha en reversa. La cual se ve afectada por diversos factores como lluvia, niebla, tierra, entre otros [8]. Lo anterior aumenta el riesgo de sufrir alguna colisión cuando se da marcha atrás al vehículo.

Dentro de las técnicas más utilizadas de DL por diversos autores en la literatura para dar solución a la detección de peatones, se encuentran las Redes Neuronales Convolucionales (CNN, por sus siglas

en inglés) [9]. Las CNN son una clase de red neuronal artificial profunda con diferentes aplicaciones, incluidas tareas complejas, como la clasificación de imágenes y el reconocimiento de objetos, que conducen a una mejora significativa en la clasificación de imágenes en varios temas, como la industria automotriz [10].

En la literatura se han realizado algunas investigaciones relacionadas con la detección de obstrucción y mejora de la visibilidad en cámaras. Un ejemplo de ello es el trabajo presentado por Uricar M. et al. [8] proponen métodos para clasificar las partes sucias, así como métodos para estimar la escena detrás de las partes sucias utilizando Redes Adversarias Generativas (GANs, por sus siglas en ingles). En Liu Y. et al. [11], resuelven el problema de la visibilidad afectada por la lluvia, proponiendo un algoritmo de detección de peatones con un módulo de eliminación de lluvia que mejora la precisión de detección en varios escenarios lluviosos, utilizando una Red Neuronal Recurrente (RNN, por sus siglas en ingles) y una red generativa y discriminatoria (GAN). Relacionado con la investigación anterior Porav et al. [12] presentan un método para segmentar imágenes afectadas por gotas de lluvia, utilizando dos bases de datos con imágenes con el mismo escenario, pero un conjunto afectando la imagen con gotas de lluvia. Utilizando GANs detectan y eliminan el ruido en las imágenes afectadas por las gotas. Así mismo en otra investigación de Uricar M. et al. [13] realizan un análisis de la detección de suciedad dividiéndola en tres partes, transparente, opaca y muy sucia, utilizando una propuesta de CNN basada en GANs y analizando en cámaras, frontales, traseras y colocadas en los costados del vehículo, además crean una base de datos con los tipos de imágenes con los tipos de suciedad para fomentar una mayor investigación. Algunas otras investigaciones relacionadas con el problema de poca visibilidad por ejemplo Heo D. et al. [14] se enfocan en la detección de peatones en tiempo real utilizando imágenes térmicas tomadas por la noche mediante el uso algoritmos de detección de peatones basado en CNNs como YOLO, que difiere de los métodos convencionales basados en clasificadores.

Por lo tanto, tratar con el problema de la visibilidad en las cámaras de los automóviles es el primer paso antes de la detección peatones u objetos, ya que es complicado o casi imposible detectar si las cámaras están obstruidas por algún agente, ya sea tierra, lodo, gotas de agua, entre otros. Gracias al avance de DL y el rápido desarrollo de las CNN, han surgido diferentes arquitecturas eficaces y robustas, utilizadas para detección de peatones y aplicaciones directamente en la industria automotriz, por mencionar algunas arquitecturas; VGG16 [15], Inception V3 [16], ResNet50 [17], GoogLeNet [18], DenseNet [19], YOLO [20], entre otras.

Pocos estudios abordan la detección de obstrucciones en las cámaras. Sin embargo, esta investigación propone una solución rentable para detectar obstrucciones y baja visibilidad en cámaras de marcha atrás de automóviles. La principal aportación es la detección de obstrucciones en cámaras de marcha atrás para evitar accidentes durante la marcha atrás. Utilizando dos casos de estudio; Inception V3 y una arquitectura RvlsNet propuesta, se describen y aplican CNNs a una base de datos de imágenes con y sin obstrucciones adquiridas en el desarrollo de este trabajo. El objetivo es determinar una arquitectura óptima para lograr una precisión de clasificación superior al 90%.

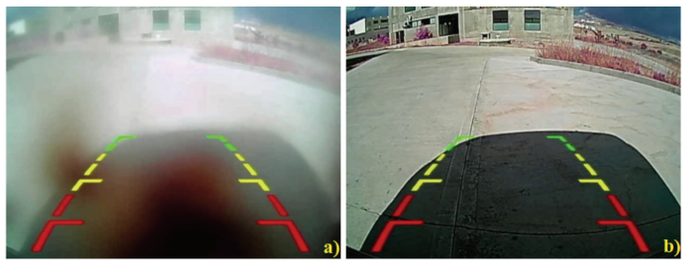


Fig. 2. a) Imagen tomada con la lente obstruida con barro, mostrando manchas en gran parte de la imagen, b) Imagen tomada con la lente limpia, obteniendo una imagen con excelente visibilidad.

2. MATERIALES Y MÉTODOS

En esta sección, se describe la metodología utilizada para llevar a cabo la investigación, que se muestra en la Fig. 2. Así mismo, se describen los detalles de las CNNs implementadas para la detección de obstrucciones en imágenes provenientes de la cámara de reversa. Este estudio se centra en resolver el problema de visibilidad en las lentes de la cámara de reversa comparando un modelo CNN preentrenado; Inception V3, con nuestro modelo propuesto; RvlsNet.

2.1. ADQUISICIÓN

En esta investigación se desarrolló un sistema de adquisición de imágenes de la cámara de marcha atrás de vehículos, ya que no existe en la literatura un conjunto de datos con estas características. Para implementar este sistema, se utilizó una cámara de marcha atrás ZHAOCI modelo 8L, disponible en el mercado y compatible con cualquier automóvil, así como un conversor (Audio/Video) a HDMI (High-Definition Multimedia Interface) y una capturadora de vídeo, con una estructura como la mostrada en la Fig. 1. (Ver sección: material complementario). Estas herramientas se instalaron en dos vehículos diferentes, un Mazda HR-V, y un Chevrolet Silhouette, y se calibraron para proporcionar una visión similar a la de un vehículo que tiene una cámara de marcha atrás de fábrica. Las imágenes se recogieron mediante un programa Python que se ejecutaba de forma sincronizada mientras el vehículo circulaba marcha atrás, simulando situaciones de conducción cotidianas.

La Fig. 2 muestra un ejemplo de los dos tipos de imágenes adquiridas: imágenes con baja visibilidad e imágenes sin ningún agente obstructor de la visibilidad.

El número total de imágenes de la base de datos adquirida es de 4,636, divididas en dos clases: 2,196 no obstruidas y 2,440 obstruidas, con un tamaño de píxel de imagen de 1.280 x 720 píxeles.

2.2. METODOLOGÍA PROPUESTA

Para el desarrollo de esta investigación se ha utilizado la metodología general de Minería de Datos propuesta por Fayyad U. [21] . La Fig. 3 muestra la metodología seguida, la cual se divide en tres etapas a considerar: preprocesamiento, implementación de las CNNs, y, validación de los modelos.

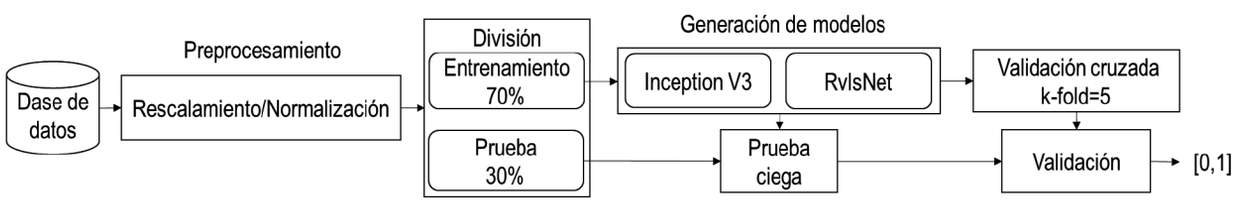


Fig. 3. Metodología propuesta.

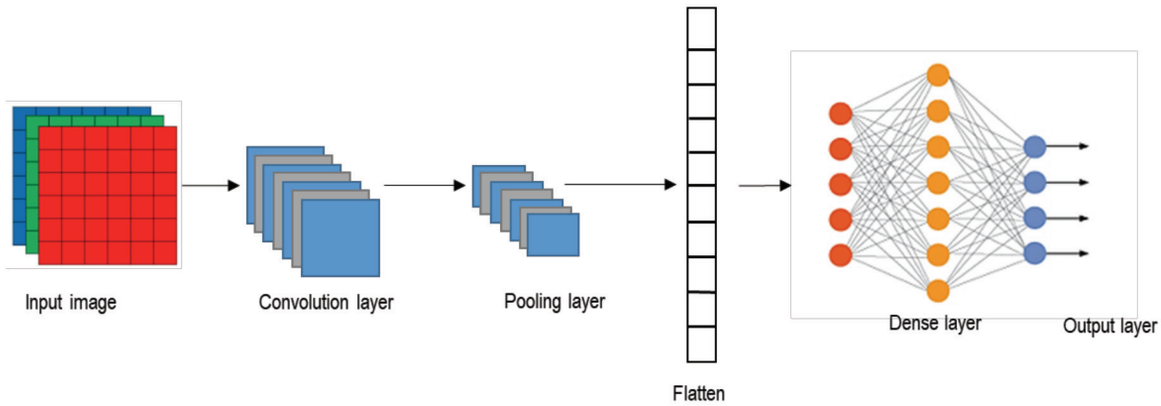


Fig. 4. Arquitectura típica de una CNN (Elaboración propia).

A continuación, se describe cada fase de la metodología.

2.2.1.Preprocesamiento de datos

En esta fase, se preparan las imágenes antes de utilizar alguna CNN. Las imágenes originales tienen un tamaño de 1280x720 pixeles. Ya que cada arquitectura de CNN tiene especificaciones diferentes respecto al tamaño de la imagen de entrada, se realiza un re-escalamiento para dejar acorde las imágenes. Ajustando el tamaño de cada imagen a una resolución de 299x299 pixeles y 150x150 pixeles, para la arquitectura de Inception V3 y RvlsNet respectivamente. Para el cambio del tamaño en las imágenes se utilizó la interpolación [22], puesto que existen diversos tipos de interpolación, para esta investigación se aplicó la interpolación bicúbica para la reducción del tamaño de la imagen. La interpolación de bicúbica [23] dio menor perdida de información comparándola respecto a la imagen original. Para llevar a cabo la interpolación bicúbica es necesario seguir con la ecuación (1)[24], que se muestra a continuación:

$$G_{bic}(x,y) = \sum_{n=-1}^{2}\sum_{m=-1}^{2} f_{(i+m,j+n)} p_{m+1}(s) p_{n+1}(t) \qquad (1)$$

En la ecuación representa las coordenadas del píxel en la imagen, $f_{(i+m,j+n)}$ representa el valor del píxel en la imagen original y p_{m+1} son polinomios cúbicos; $p_0(u) = \frac{-u^3+2u^2-u}{2}$, $p_1(u) = \frac{3u^3-5u^2+2}{2}$, $p_2(u) = \frac{-3u^3+4u^2+u}{2}$, $p_3(u) = \frac{u^3-u^2}{2}$ Estas funciones se utilizan para calcular los coeficientes de interpolación y lograr una interpolación suave y precisa entre píxeles vecinos. Donde "u" generalmente toma valores entre 0 y 1, y se utiliza para determinar la posición relativa del punto de interpolación dentro de la región de cuatro píxeles vecinos, y $s = x - x_i$ y $t = y - y_i$.

Además de rescalar las imágenes, también se aplicó la ecuación (2), que corresponde a la normalización, es decir, utilizar el valor máximo que puede tener un píxel y dividirlo en cada uno de los píxeles de la imagen. Por tratarse de imágenes de 8 bits, el valor máximo que puede tener un pixel es 255, resultando en rangos entre 0-255.Para realizar la normalización, consideramos float64, un tipo de dato que contiene una precisión de 17 dígitos decimales, manteniendo así un rango mínimo y máximo de 0 y 1 respectivamente sin afectar la composición de la imagen, lo que demuestra que el proceso de normalización no modifica el almacenamiento de la imagen en sí.

$$Img_{norm} = \frac{Img - V_{px\_min}}{V_{px\_max} - V_{px\_min}} \qquad (2)$$

Donde *Img* se refiere a la imagen original, V_{px_min} el valor mínimo de píxeles de la imagen, V_{px_max} el valor máximo de píxeles de la imagen y Img_{norm} corresponde a la imagen normalizada.

El proceso de re-escalamiento y normalizar previo a usar CNNs se le conoce como preprocesamiento, puesto que, se tiene un menor costo computacional o de procesamiento trabajar con datos pequeños como 0 y 1.

2.2.2. Redes neuronales convolucionales

Las CNN están diseñadas para procesar datos que vienen en forma de matrices múltiples, por ejemplo, una imagen. Existen cuatro ideas clave detrás de CNNs que aprovechan las propiedades de las señales naturales: conexiones locales, pesos compartidos, agrupación y el uso de múltiples capas [6]. Las CNNs están compuestas de diferentes filtros/núcleos. Estos constituyen un conjunto de parámetros entrenables que pueden hacer convolucionar espacialmente a una imagen dada. Con la finalidad de detectar características como bordes y formas. Este alto número de filtros esencialmente aprende a capturar características espaciales de la imagen en función de los pesos aprendidos a través de la propagación hacia atrás [25].

La arquitectura de una CNN típica (véase la Fig. 4) se estructura en una serie de etapas. La primera consiste en la entrada de la imagen, después las capas de convolución y reducción/agrupación, seguidas de una etapa de aplanamiento y, por último, una capa densa (red neuronal artificial).

El funcionamiento general de una CNN es lo siguiente; La imagen de entrada se divide en campos receptivos que alimentan una capa convolucional que extrae características de la imagen de entrada (por ejemplo, detecta líneas verticales, vértices, etc.). El siguiente paso es la capa de reducción y agrupamiento (Pooling layer) que reduce la dimensionalidad de las características extraídas manteniendo la información más importante. Luego se alimenta una red neuronal multicapa con las características en forma de un vector columna (Flatten). Por último, la salida final de la red es un grupo de nodos (Dense layer) que clasifican el resultado. Entre los parámetros que pueden manipularse en una CNN se encuentran el tamaño de la imagen de entrada, el tamaño del filtro y del núcleo de convolución, el número de épocas, el número de neuronas de la capa densa y el número de salidas de la red. Las arquitecturas robustas CNN mencionadas anteriormente tienen estos parámetros establecidos, probados en una miríada de aplicaciones de reconocimiento de imágenes. A continuación, se describen las dos arquitecturas que se utilizaron en este trabajo de investigación.

2.2.2.1. Inception V3

Inception V3 [16], es una arquitectura convolucional profunda ampliamente utilizada para tareas de clasificación. Tiene múltiples bloques de construcción simétricos y asimétricos, donde cada bloque tiene varias ramas de circunvoluciones, agrupamiento medio,

agrupamiento máximo, concatenado, abandonos y capas totalmente conectadas [10]. Esta red tiene 42 capas en total y cuenta con 29,3 millones de parámetros. La Fig. 5 (véase la sección: material complementario) muestra la arquitectura de la red Inception V3 [16]-[26].

La Tabla 3 (véase la sección: material complementario) muestra las características de la arquitectura de Inception V3, los tamaños de entrada de cada capa y el número de parámetros utilizados.

2.2.2.2. Propuesta RvlsNet

RvlsNet se diseñó basándose en la arquitectura típica de una CNN, tomando también como referencia los modelos propuestos en la investigación de Fahim A. et al. [27]. La Fig. 6 (véase la sección: material complementario) muestra la arquitectura RvlsNet diseñada con ocho capas: cuatro capas de convolución que realizan la extracción de características de la imagen y cuatro capas de reducción. Además, se añadieron tres capas totalmente conectadas de 128 neuronas cada capa con funciones de activación ReLu [28] y una neurona de salida con una función sigmoidea a partir de la cual se determinará un 0 o un 1.

Dentro de las especificaciones de esta arquitectura, cuenta con un total de 1,076,929 parámetros. Es importante mencionar que esta arquitectura no es robusta en comparación con las arquitecturas ya establecidas en la literatura antes mencionada, al no ser una arquitectura robusta, tiene un menor peso de procesamiento. La Tabla 4 (Ver sección: material complementario) muestra las capas y el tamaño del kernel, así como la entrada necesaria para que la arquitectura funcione de manera óptima.

2.2.3. Validación de las CNN

Por último, para evaluar el rendimiento de la CNN propuesta y de Inception V3, se utilizaron algunas métricas estadísticas. Como la matriz de confusión y las curvas ROC [29]. La matriz de confusión es una herramienta que permite visualizar el rendimiento de un algoritmo de inteligencia artificial [30]. Cada columna de la matriz representa el número de predicciones en cada clase. Mientras que cada fila representa las instancias en la clase real (ver Tabla 5 en la sección: material complementario). Uno de los beneficios de las matrices de confusión es proporcionar información general sobre el modelo en su capacidad de predicción.

La matriz de confusión proporciona los datos que se muestran en la Tabla 5. A partir de los cuales se pueden calcular métricas de evaluación adicionales, estas métricas se muestran en la Tabla 6 (Ver sección: material complementario) [10]-[31].

Otra métrica utilizada a menudo para la evaluación del rendimiento de los algoritmos de DL es la curva ROC. La curva ROC es una representación gráfica bidimensional que muestra la relación entre la tasa de verdaderos positivos y la tasa de falsos positivos en diferentes umbrales de clasificación.

Para evaluar algoritmos, puede ser conveniente sintetizar el rendimiento de la curva ROC en un único valor escalar que refleje el rendimiento previsto. Una estrategia común consiste en calcular el área bajo la curva ROC, también conocida como AUC. El AUC es una métrica ampliamente utilizada para evaluar la capacidad discriminatoria de un modelo. Un valor de AUC-ROC cercano a 1 indica un modelo con alta capacidad de clasificación, mientras que un valor cercano a 0,5 sugiere un rendimiento similar al azar [29]-[32]-[33].

Además, se consideró implementar la técnica de validación cruzada (CV) k-fold [34] con el fin de aumentar la robustez en el análisis de los datos correspondientes y verificar si existe o no un sobreajuste en los datos. La CV k-fold implica dividir el conjunto de datos de entrenamiento ("Train") en k subconjuntos, de los cuales k-1 partes se utilizan para el entrenamiento y la porción restante se reserva para la validación. Este proceso se repite k veces, cubriendo toda la base de datos para realizar un análisis completo. Una vez obtenido el modelo, el conjunto de datos de prueba ("Test") se emplea para la prueba ciega (BT, por sus siglas en inglés) y evaluar el rendimiento del modelo. Una vez validado los modelos con esta técnica, aplicar la prueba a ciegas en los modelos, definirá las métricas de los modelos que describen su comportamiento al mostrarles datos que nunca se le mostraron, arrojando una predicción de salida de la clase correspondiente, en este caso una salida binaria ("0" y "1") que se interpreta como no obstrucción y obstrucción.

Basándose en lo anterior y en la metodología propuesta, se obtuvieron diferentes resultados. Estos se describen a continuación.

3. RESULTADOS

En este estudio, se evaluaron dos modelos de CNN, Inception V3 y una RvlsNet, para realizar la detección de obstrucciones en la cámara de marcha atrás de un vehículo con el fin de evitar accidentes al dar marcha atrás. Los modelos utilizados en este estudio fueron entrenados utilizando el conjunto de datos adquiridos a través del sistema de adquisición implementado específicamente para esta investigación. Se consideraron los hiperparámetros de entrenamiento correspondientes a cada modelo y se detallan en la Tabla 7. Es importante destacar que los hiperparámetros del modelo RvlsNet fueron seleccionados a través de un proceso experimental, y los valores presentados en la Tabla 7 representan los resultados más óptimos obtenidos durante el proceso.

El objetivo de esta investigación es clasificar imágenes con el lente de la cámara obstruido y no obstruido. Su finalidad es contribuir a la reducción las colisiones que se dan cuando el vehículo está en reversa. A continuación, se muestran los resultados de los dos casos de estudios aplicados en esta investigación.

3.1. CROSS VALIDATION K-FOLD

3.1.1. Caso de estudio 1: Inception V3

Para el primer caso, se utilizó el modelo pre-entrenado Inception V3. Utilizando los hiperparámetros de entrenamiento mostrados en la Tabla 7 para Inception V3, utilizando así 500 épocas para

Hiperparámetro	Modelo	
	RvlsNet	Inception V3
Entrada	150x150	299x299
Funciones de activación	ReLu	ReLu
Épocas	30	500
Optimizador	Adam	Descenso gradiente
Capas convolucionales	4	48
Tamaño del lote	32	32
kernel	3x3	5x5,3x3,1,1
Tamaño de la agrupación	2x2	2x2
Filtros	32,64,128,128	32,64,128
Pérdida	Entropía cruzada binaria	Entropía cruzada
Función de salida	Sigmoide	Softmax
Número de clases	2	2
Capas densas	3 x 128 neuronas	1 x 1024 neuronas

Tabla 7. Hiperparámetros de entrenamiento del modelo.

			Train			
k	UAC	Exactitud	Precisión	Sensibilidad	Especificidad	F1-Score
1	0.9977	0.9975	1.0000	0.9953	1.0000	0.9977
2	0.9984	0.9982	1.0000	0.9967	1.0000	0.9983
3	0.9977	0.9975	1.0000	0.9954	1.0000	0.9977
4	0.9984	0.9982	1.0000	0.9967	1.0000	0.9983
5	0.9987	0.9986	1.0000	0.9974	1.0000	0.9987
Mean	**0.9982**	**0.9980**	**1.0000**	**0.9963**	**1.0000**	**0.9981**

			Validación			
k	UAC	Exactitud	Precisión	Sensibilidad	Especificidad	F1-Score
1	0.9985	0.9983	0.9963	1.0000	0.9970	0.9981
2	0.9974	0.9971	1.0000	0.9947	1.0000	0.9974
3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4	0.9974	0.9972	1.0000	0.9947	1.0000	0.9973
5	0.9960	0.9957	1.0000	0.9921	1.0000	0.9960
Mean	**0.9979**	**0.9977**	**0.9993**	**0.9963**	**0.9994**	**0.9978**

Tabla 8. CV Inception V3.

el entrenamiento y un tamaño de imagen de entrada de 299x299. En primer lugar, se realizó el CV utilizando el 70% del conjunto de datos para realizar esta prueba, los resultados se muestran en la Tabla 8.

Los resultados de la Tabla 8, muestran el comportamiento del modelo en k-iteraciones ante la base de datos, entrenando y validando respectivamente. Tanto en Train como en la validación tenemos las métricas por arriba de 0.99, lo que nos indica la capacidad del modelo para clasificar los datos de entrada y realizar la predicción correspondiente con el 70% de los datos.

3.1.2. Caso de estudio 2: RvlsNet

Así mismo, en el segundo caso, se evaluó el modelo propuesto RvlsNet, mediante la técnica de CV, en las mismas condiciones de Inception V3, sin embargo, los parámetros de entrenamiento en RvlsNet mostrados en la Tabla 7, son distantes al de Inception V3, esto, dado que es una arquitectura menos robusta, con menos parámetros de entrenamiento, utilizando así pues 30 épocas para entrenar, dado que con esta cantidad de épocas el modelo alcanzaba su punto más optimo, además de un tamaño de imagen de 150x150. Se obtuvieron los resultados que se muestran a continuación en la Tabla 9.

			Train			
k	UAC	Exactitud	Precisión	Sensibilidad	Especificidad	F1-Score
1	0.9977	0.9979	0.9961	1.0000	0.9955	0.998
2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	0.9981	0.9982	0.9967	1.0000	0.9962	0.9983
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Mean	**0.9992**	**0.9992**	**0.9986**	**1.0000**	**0.9983**	**0.9993**

			Validación			
k	UAC	Exactitud	Precisión	Sensibilidad	Especificidad	F1-Score
1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	0.9872	0.9873	0.9869	0.9895	0.9848	0.9882
3	0.9974	0.9972	1.0000	0.9947	1.0000	0.9973
4	0.9821	0.9831	0.9717	0.9974	0.9668	0.9844
5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Mean	**0.9933**	**0.9935**	**0.9917**	**0.9963**	**0.9903**	**0.9940**

Tabla 9. CV RvlsNet.

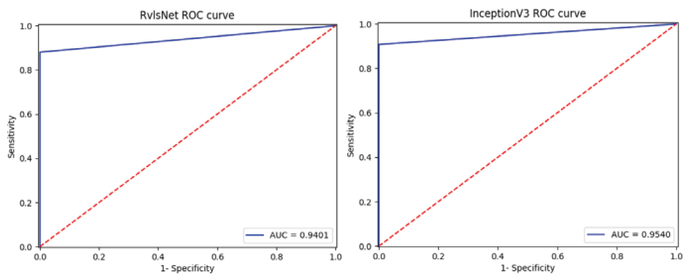


Fig. 7. Curvas ROC Curvas ROC.

Los resultados mostrados en la Tabla 9, nos da la información necesaria del comportamiento de nuestro modelo con el 70% de los datos, las métricas mostradas nos indica que nuestro modelo aprende bien de los datos y se ajusta bien en el aprendizaje, mostrando que entre el Train y el Validation en el modelo muestra un rendimiento sobresaliente. Los resultados indican una capacidad excepcional para clasificar correctamente las instancias y un equilibrio sólido entre la precisión y el Recall.

3.2. PRUEBA A CIEGAS

Para verificar el comportamiento de los modelos, ambos se sometieron a un BT, esta prueba se realizó utilizando el 30% del conjunto de datos que no se utilizó en el entrenamiento. El objetivo del BT es evaluar la capacidad del modelo para generalizar y realizar predicciones precisas en situaciones reales. Los resultados del BT pueden proporcionar una evaluación más fiable del rendimiento real del modelo, ya que no está influido por el ajuste del modelo a los datos utilizados durante el entrenamiento.

La Tabla 10 (Ver sección: material complementario) muestra la matriz de confusión resultante del BT de los modelos. La salida de los modelos como tal es binaria, teniendo "0" como etiqueta de una imagen sin obstrucción y "1" como imágenes con obstrucción, ambos modelos muestran una alta capacidad para clasificar las imágenes sin ninguna obstrucción (Imágenes etiquetadas con "0"), obteniendo el 50% del total de las imágenes del BT. Ambos modelos mostraron predicciones incorrectas al predecir imágenes con obstrucción, con errores del 4,6% y 5,5% en Inception V3 y RvlsNet respectivamente.

Asimismo, se calculó la curva ROC para determinar el AUC en base a la predicción de las imágenes de BT, obteniéndose la Fig. 7, que muestra la curva ROC de los dos modelos, donde se puede observar que RvlsNet se sitúa sólo 0.129 por debajo de Inception.

Además de las curvas ROC y utilizando las matrices de confusión presentadas en la Tabla 10, se calcularon las otras métricas de evaluación que se pueden observar en la Tabla 11, estas métricas nos dan un resumen conciso del desempeño de los modelos en esta prueba, tomando como las más significativas en esta investigación el resultado de la Exactitud y el AUC, ya que estas dos métricas indican las predicciones correctas en general y el buen desempeño en la clasificación de instancias positivas y negativas.

Modelo	AUC	Exactitud	Precisión	Sensibilidad	Especificidad	F1-Score
Inception V3	*0.9540*	*0.9549*	1.0000	0.9102	1.0000	0.9549
RvlsNet	*0.9401*	*0.9416*	1.0000	0.8900	1.0000	0.9412

Tabla 11. Resumen de las métricas de evaluación del modelo.

4. DISCUSIÓN

En este estudio, la adquisición de datos se realizó mediante la captura de imágenes de la cámara de marcha atrás. El conjunto de datos obtenido incluye imágenes reales con distintos niveles de visibilidad, desde excelente hasta muy baja debido a obstrucciones de la cámara. En comparación con otros conjuntos de datos disponibles públicamente, como los de INRIA y Citypersons [35], el conjunto de datos utilizado en esta investigación se compone exclusivamente de imágenes captadas por cámaras de reversa. Los resultados obtenidos demuestran la eficacia del modelo propuesto para la detección de obstrucciones en cámaras de marcha atrás. Cabe destacar que, en algunos estudios anteriores, los autores han utilizado conjuntos de datos sintéticos, creados por medio de computadora.

Estudios recientes han empleado CNN para la detección de peatones. En la Tabla 12 (véase la sección de material complementario) se detallan los resultados de otros investigadores que han utilizado enfoques CNN y conjuntos de datos modificados para la detección de agentes obstructivos en imágenes.

Los resultados de la Tabla 12 muestran aspectos destacables. Las técnicas detectan eficazmente los obstáculos, pero se centran más en los problemas de visibilidad de la lluvia. Las CNN destacan en la clasificación de imágenes, lo que abre vías para aplicaciones de seguridad en automoción.

Uricar M. et al. [8], en su investigación utilizan GANs para regenerar una imagen tapada con barro, sin embargo, aunque cumplen con el objetivo y realizan la acción, no muestran el nivel de precisión de su algoritmo, ni cómo evalúan su desempeño. Liu [11], proponen una solución utilizando una combinación de Redes Neuronales Recurrentes (RNN) y Redes Generativas Adversariales (GANs) para eliminar la lluvia en las imágenes y posteriormente lograr la detección de peatones. Aunque su investigación aborda el problema de visibilidad dado por la lluvia, el AP obtenido es bajo, lo que limita su precisión en un sistema cuando se implementa.

En la investigación de Porav et al. [12], realizan un interesante método para resolver la obstrucción generada por las gotas de lluvia, utilizando la segmentación de imágenes consiguen resolver el problema, sin embargo, aunque el método es de bajo coste computacional, los resultados obtenidos son muy bajos, lo que implica un problema a la hora de aplicar su sistema. Uricar M et al. realizaron una fascinante investigación centrada en la detección y eliminación de suciedad de varias cámaras, incluida la de marcha atrás, en coches autónomos. Emplearon GANs y obtuvieron resultados prometedores con un Recall superior al 98%. Sin embargo, la precisión fue relativamente baja, lo que se tradujo en una mayor tasa de falsos positivos. Lograr un equilibrio entre estas métricas es crucial para garantizar la seguridad.

Los modelos se evaluaron utilizando múltiples métricas: exactitud, precisión, sensibilidad, especificidad, F1-Score y AUC. Inception V3 demostró resultados superiores, sin embargo, los resultados de RvlsNet están muy de cercas a Inception V3. Esta investigación introduce una arquitectura robusta y una arquitectura sencilla para abordar la obstrucción sin regenerar las imágenes sintéticas. El enfoque propuesto supera las precisiones de los estudios relacionados. Los modelos robustos como Inception V3 son intensivos en cálculo y exigen un alto costo computacional lo que dificulta integrarlos en sistemas en tiempo real de bajo costo. La arquitectura no robusta propuesta alcanza una precisión del 94,16%, adecuada para la prevención de accidentes. Inception V3 es limitada en sistemas embebidos de baja potencia. RvlsNet es más flexible en entornos reales. Las arquitecturas pre-entrenadas reducen su rendimiento en sistemas embebidos dada a la arquitectura robusta y pesada que presentan.

Por otro lado, una de las limitaciones de este trabajo está relacionada con el número de escenas de las imágenes, ya que éstas se tomaron únicamente durante el día. En este estudio, se propone una arquitectura CNN para detectar la obstrucción en la cámara de marcha atrás de los vehículos y así, evitar accidentes. La principal ventaja de la propuesta de investigación es lograr una reducción en el tiempo de detección, ya que la arquitectura tiene menos capas, menos parámetros, y esto resulta poder aplicarlo en un sistema de bajo costo computacional, a diferencia de Inception V3.

5. CONCLUSIONES

El objetivo de este trabajo de investigación es realizar la detección de obstrucciones en objetivos de cámaras de marcha atrás y atacar uno de los problemas de visibilidad más comunes en las cámaras de vehículos. Utilizando dos modelos de CNN; Inception V3 y el modelo RvlsNet propuesto, se obtuvo un rendimiento de 0,9549 y 0,9416 de exactitud respectivamente. Mostrando la capacidad de clasificar las imágenes y así poder determinar cuando la lente de la cámara está obstruida, superando ligeramente a trabajos relacionados.

El modelo propuesto para esta investigación resulta ser un modelo no robusto, con un bajo coste computacional para ser aplicado. Esto se debe a su arquitectura. Por lo tanto, se concluye que un modelo de arquitectura más simple puede alcanzar resultados similares a modelos más robustos con muchas capas de convolución y costo computacional, abriendo la posibilidad de ejecutarlo en microcontroladores con bajo poder de procesamiento. Se logró el objetivo de clasificar y detectar la obstrucción del objetivo de la cámara de marcha atrás. Como trabajo futuro, se pretende detectar peatones con la cámara de marcha atrás, desarrollando un sistema de varios módulos, incluyendo la detección de obstrucción para conseguir una detección sofisticada, además de implementar un sistema que cuando haya obstrucción, solucione el problema limpiando la cámara o avisando al conductor de la posible obstrucción, manteniendo el mismo rendimiento y así evitar o reducir el número de accidentes de tráfico.

REFERENCIAS

[1] Organización Mundial de la Salud, "Informe Mundial Sobre Prevención De Los Traumatismos Causados Por El Tránsito," World Heal. Organ., pp. 52–53, 2004, DOI: https://doi.org/https://doi.org/10.1016/S1131-3587(07)75236-6.

[2] J. Pablo and A. Quezada, "Accidentes automotrices como problema de salud pública.," Cuaderno de investigación,Instituto Belisario Domínguez, 2015.

[3] J. Á. Pérez Requena, "Las Nuevas Tecnologías Aplicadas a la Seguridad Vial," Tesis Doctoral,Programa de Doctorado en Ciencias Sociales, UCM, Muricia, 2018.

[4] J. Aguirre et al., "Aplicación de la inteligencia artificial en la industria automotriz," Iber. J. Inf. Syst. Technol., pp. 149–159, 2021.

[5] L. E. Sucar and G. Gómez, "Vision Computacional," Inst. Nac. Astrofísica, Óptica y Electrónica, p. 185, 2011, [Online]. Available: http://ccc.inaoep.mx/~esucar/Libros/vision-sucar-gomez.pdf

[6] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015, DOI: https://doi.org/https://doi.org/10.1038/nature14539.

[7] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," pp. 304–311, 2010, DOI: https://doi.org/https://doi.org/10.1109/cvpr.2009.5206631.

[8] M. Uřičář, H. Rashed, A. Ranga, A. Dahal, and S. Yogamani, "Visibilitynet: Camera visibility detection and image restoration for autonomous driving," IS T Int. Symp. Electron. Imaging Sci. Technol., vol. 2020, no. 16, pp. 1–8, 2020, DOI: https://doi.org/https://doi.org/10.2352/ISSN.2470-1173.2020.16.AVM-079.

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278–2323, 1998, DOI: https://doi.org/https://doi.org/10.1109/5.726791.
[10] V. Maeda-Gutiérrez et al., "Comparison of convolutional neural network architectures for classification of tomato plant diseases," Appl. Sci., vol. 10, no. 4, p. 1245, 2020, DOI: https://doi.org/https://doi.org/10.3390/app10041245.
[11] Y. Liu, J. Ma, Y. Wang, and C. Zong, "A novel algorithm for detecting pedestrians on rainy image," Sensors (Switzerland), vol. 21, no. 1, pp. 1–15, 2021, DOI: https://doi.org/10.3390/s21010112.
[12] H. Porav, T. Bruls, and P. Newman, "I Can See Clearly Now : Image Restoration via De-Raining," 2019, DOI: https://doi.org/https://doi.org/10.48550/arXiv.1901.00893.
[13] M. Uricar, P. Krizek, G. Sistu, and S. Yogamani, "SoilingNet: Soiling Detection on Automotive Surround-View Cameras," May 2019, Accessed: May 22, 2023. [Online]. Available: http://arxiv.org/abs/1905.01492
[14] D. Heo, E. Lee, and B. C. Ko, "Pedestrian detection at night using deep neural networks and saliencymaps," IS T Int. Symp. Electron. Imaging Sci. Technol., no. January, 2018, DOI: https://doi.org/10.2352/J.ImagingSci.Technol.2017.61.6.060403.
[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, Sep. 2015. DOI: https://doi.org/10.48550/arxiv.1409.1556.
[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, vol. 2016-Decem, pp. 2818–2826. DOI: https://doi.org/10.1109/CVPR.2016.308.
[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Dec. 2016, vol. 2016-Decem, pp. 770–778. DOI: https://doi.org/10.1109/CVPR.2016.90.
[18] C. Szegedy et al., "Going deeper with convolutions," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, vol. 07-12-June, pp. 1–9. DOI: https://doi.org/10.1109/CVPR.2015.7298594.
[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Aug. 2017, vol. 2017-Janua, pp. 2261–2269. DOI: https://doi.org/10.1109/CVPR.2017.243.
[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, vol. 2016-Decem, pp. 779–788. DOI: https://doi.org/10.1109/CVPR.2016.91.
[21] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From Data Mining to Knowledge Discovery in Databases," AI Mag., vol. 17, no. 3 SE-Articles, p. 37, Mar. 1996, DOI: https://doi.org/10.1609/aimag.v17i3.1230.
[22] P. Thévenaz, T. Blu, and M. Unser, "Image interpolation and resampling," Handb. Med. imaging, Process. Anal., vol. 1, no. 1, pp. 393–420, 2000.
[23] J. W. Hwang and H. S. Lee, "Adaptive image interpolation based on local gradient features," IEEE Signal Process. Lett., vol. 11, no. 3, pp. 359–362, 2004.
[24] R. Keys, "Cubic convolution interpolation for digital image processing," IEEE Trans. Acoust., vol. 29, no. 6, pp. 1153–1160, 1981.
[25] M. García Villanueva and L. Romero Munoz, "Diseño de una arquitectura de red neuronal convolucional para la clasificación de objetos," Cienc. Nicolaita #81, pp. 46–61, 2020, [Online]. Available: https://www.cic.cn.umich.mx/cn/article/download/517/410/2388#:~:text=Las CNNs están compuestas de,características como bordes y formas.
[26] S. Zorgui, S. Chaabene, B. Bouaziz, H. Batatia, and L. Chaari, A Convolutional Neural Network for Lentigo Diagnosis, vol. 12157 LNCS. Springer International Publishing, 2020. DOI: https://doi.org/10.1007/978-3-030-51517-1_8.
[27] F. Ahmed, B. A. Topu, and S. M. M. Islam, "HOG and Gabor Filter Based Pedestrian Detection using Convolutional Neural Networks," 2nd Int. Conf. Electr. Comput. Commun. Eng. ECCE 2019, pp. 1–6, 2019, DOI: https://doi.org/10.1109/ECACE.2019.8679133.
[28] J. Schmidt-Hieber, "Nonparametric regression using deep neural networks with ReLU activation function," Ann. Stat., vol. 48, no. 4, pp. 1875–1897, 2020.
[29] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," IEEE Trans. Knowl. Data Eng., vol. 17, no. 3, pp. 299–310, 2005, DOI: https://doi.org/10.1109/TKDE.2005.50.
[30] O. Caelen, "A Bayesian interpretation of the confusion matrix," Ann. Math. Artif. Intell., vol. 81, no. 3, pp. 429–450, 2017, DOI: https://doi.org/10.1007/s10472-017-9564-8.
[31] R. Borja-Robalino, A. Monleón-Getino, and J. Rodellar, "Estandarización de métricas de rendimiento para clasificadores Machine y Deep Learning," Rev. Ibérica Sist. e Tecnol. Informação, no. E30, pp. 184–196, 2020.
[32] A. M. Carrington et al., "Deep ROC analysis and AUC as balanced average accuracy, for improved classifier selection, audit and explanation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 1, pp. 329–341, 2022.
[33] A. M. Carrington et al., "Deep ROC analysis and AUC as balanced average accuracy to improve model selection, understanding and interpretation," arXiv Prepr. arXiv2103.11357, 2021.
[34] O. L. Laura, "Evaluation of Classification Algorithms using Evaluación de Algoritmos de Clasificación utilizando Validación Cruzada," Laccei Int. Multi-Conference Eng. Educ. Technol., no. July 2019, pp. 1–6, 2019, [Online]. Available: http://dx.doi.org/10.18687/LACCEI2019.1.1.471
[35] S. Zhang, R. Benenson, and B. Schiele, "CityPersons: A Diverse Dataset for Pedestrian Detection," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4457–4465. DOI: https://doi.org/10.1109/CVPR.2017.474.
[36] F. L. Saca, A. F. Ramírez, and C. A. Cruz, "Prototipo Funcional Para Clasificación De Imágenes Con Salida De Audio En Un Sistema Embebido Con Red Neuronal Convolucional.," Pist. Educ., vol. 40, no. 130, 2018.

MATERIAL COMPLEMENTARIO

https://www.revistadyna.com/documentos/pdfs/_adic/10865_1.pdf