

Modelo de Clasificación de Imágenes de Vehículos Usando Redes Neuronales

Miguel Angel Cabrera Victoria

Resumen - Este trabajo presenta un modelo de clasificación de imágenes de vehículos utilizando redes neuronales utilizando aprendizaje supervisado. El objetivo principal es desarrollar un sistema eficiente que clasifique tres tipos de vehículos a partir de imágenes mediante una arquitectura de una red neuronal conectada. La propuesta fue implementada siguiendo el proceso de aprendizaje, para entrenar y evaluar el modelo, se utilizaron imágenes preprocesadas de un conjunto de datos de vehículos capturados estando dichos vehículos en una posición de perfil. La arquitectura de la red incluye una capa de entrada, dos capas intermedias y una capa de salida. Los resultados experimentales muestran que el modelo propuesto alcanza una precisión de 95% en la clasificación.

I- Introducción

La clasificación de imágenes es una aplicación de Machine Learning frecuente en diferentes campos, sin embargo, estos tipos de sistemas requieren una identificación precisa y rápida de los tipos de vehículos para la mejora en la toma de decisiones y eficiencia operativa. En el contexto de carreteras y casetas de pago, la capacidad de clasificar correctamente los vehículos en tiempo real es crucial para automatización de procesos, el tráfico que pudiera generarse y la implementación de cobro basados en la clase de vehículo.

A pesar de los avances en las técnicas de clasificación, muchos métodos actuales enfrentan diversos problemas, como la necesidad de grandes volúmenes de datos etiquetados y la dificultad para clasificar vehículos en distintos escenarios. Factores como la variación en la iluminación, los ángulos de cámara y las condiciones climáticas afectan significativamente la precisión del reconocimiento, lo que limita la eficiencia de los modelos en situaciones reales.

Este trabajo propone la implementación de un modelo de clasificación de imágenes basado en redes neuronales para el reconocimiento de diferentes tipos de vehículos en casetas de pago de carreteras. Para alcanzar este objetivo, se entrena un modelo de red neuronal utilizando un conjunto de datos que incluye diversas categorías de vehículos que este etiquetado, algunas metodologías empleadas incluye técnicas de preprocesamiento, ajuste de hiperparámetros y validación cruzada para garantizar el funcionamiento correcto del modelo y poder abordar a una conclusión sobre el modelo.

II. Metodología

A. Recolección y Preparación de los datos

Para la recolección de las imágenes se optó por escoger 120 imágenes de vehículos privados, como automóviles, motocicletas y camionetas estilo pick up que estuvieran de perfil con un fondo color blanco tanto para el conjunto de entrenamiento como para el de prueba, ya que como se mencionó anteriormente unas de las problemáticas que tienen los modelos de clasificación son los factores como la variación en distintos escenarios, como la iluminación y los ángulos de cámara, por lo que de esta manera se puede reducir el volumen de datos de como se puede ver en las siguientes Figuras.

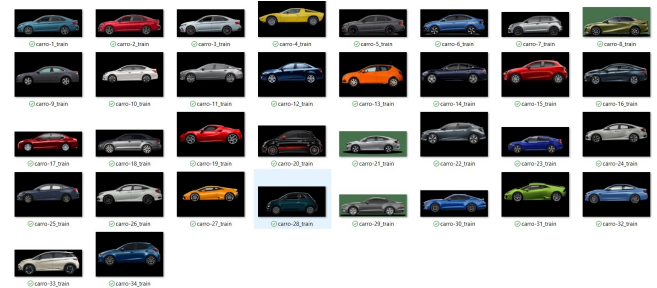


Fig. 1. Imágenes de entrenamiento con la etiqueta de carro

Para garantizar una correcta clasificación, cada imagen fue almacenada dentro de una carpeta correspondiente al tipo de vehículo etiquetado, organizando el conjunto de datos en función de la categoría del vehículo, al igual se procuro que las frecuencias de las etiquetas del conjunto de entrenamiento sea los más similares posibles tal como se ve en la figura 3.

| Etiquetas |
|-------------|
| Carro |
| Pickup |
| Motocicleta |

Fig. 2 Etiquetas asignadas para los datos

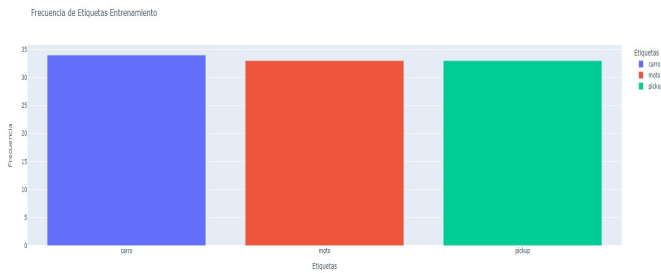


Fig. 3 Frecuencia de Etiquetas en el conjunto de entrenamiento

Finalmente, todas las imágenes fueron redimensionadas a una resolución uniforme de 128 x 128 píxeles para estandarizar el tamaño y facilitar el procesamiento por la red neuronal, además las imágenes se convirtieron a escalas de grises para reducir la complejidad del modelo y los múltiples colores que pudieran tener para solo centrarse en las características importantes de los vehículos. Esta conversión permite que el modelo se enfoque en la forma y los contornos, lo que es considerado fundamental para tener una clasificación precisa como se muestra en la Figura 3, y finalmente al tener los vectores de cada imagen se normalizan sabiendo que al tener un formato RGB se divide 255.

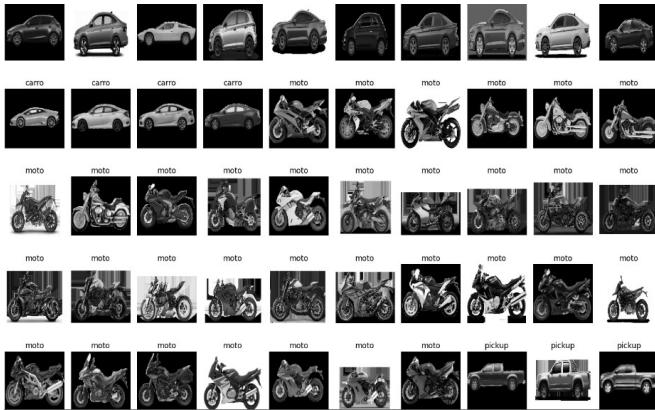


Fig. 3. Imágenes preprocesadas

B. Creación y evaluación del modelo

Para la creación del modelo, se implementó una red neuronal utilizando la librería Keras, que ofrece una interfaz de alto nivel sobre Tensor Flow, facilitando el diseño y la implementación de arquitecturas de manera más explícita. La red neuronal diseñada es compuesta por capas densas conectadas secuencialmente, una capa de entrada de 16,384 entradas, una primera capa intermedia de 300 neuronas, una segunda capa de 128 neuronas y una capa de salida de 3 salidas debido a que solo son 3 categorías.

Dentro de las capas intermedias y la de salida se implementaron dos diferentes funciones de activación, para las capas intermedias se aplicó la función **Rectified Linear Unit**

o comúnmente conocida como ReLU debido a que tienen una mejor capacidad de manejar problemas que no son lineales y ayuda a tener una mejor convergencia en el modelo. La capa de salida utiliza una función de activación softmax para generar una probabilidad por cada categoría, permitiendo realizar una clasificación con más de dos categorías, finalmente al modelo se le aplicó un algoritmo de optimización Adam, el cual es una extensión de la gradiente descendente estocástica para poder actualizar los pesos de una red manera iterativa basado en conjunto de entrenamiento, otro de los hiperparámetros que se modificó es el número de épocas pasando a ser solamente 9 para poder hacer que el modelo pudiera tener buenos resultados en las evaluaciones.

Además de la construcción de modelo, para asegurar que el modelo de clasificación de imágenes interprete correctamente las etiquetas de los vehículos, se realizó una transformación de etiquetas categóricas a valores numéricos.

| Etiqueta Original | Etiqueta Codificada |
|-------------------|---------------------|
| carro | 0 |
| moto | 1 |
| pickup | 2 |

Fig. 4. Imágenes de entrenamiento con la etiqueta de carro

Esta conversión es esencial ya que los modelos de Machine Learning no pueden procesar directamente las etiquetas textuales, posteriormente se realizó una codificación adicional conocida como one-hot encoding, la cual consiste en convertir las etiquetas numéricas en un formato binario, donde cada clase es representada por un vector exclusivo, permitiendo calcular al modelo las probabilidades individuales para cada clase, mejorando la precisión en la asignación final de la categoría correspondiente.

II. RESULTADOS

Para evaluar el modelo de clasificación, se dividió en dos etapas primero durante la fase de entrenamiento y la fase de validación, se monitoreó el rendimiento del modelo utilizando diferentes métricas para asegurar que no haya overfitting y evaluar la capacidad del modelo de generalizar.

A. Fase de Entrenamiento

Durante el entrenamiento de la red neuronal, se registró el historial de pérdida en función de las épocas establecidas, este historial permitió visualizar como es que se comporta la función de costo. La pérdida del modelo disminuyó de manera constante durante las primeras épocas, indicando que el modelo estaba aprendiendo de los datos de entrenamiento, sin embargo, a partir de cierto punto, la función de costo

incrementa un pequeño porcentaje teniendo una pequeña oscilación a pesar aunque están presentes, no fueron de gran importancia para afectar la capacidad de aprendizaje del modelo, teniendo un comportamiento estable y una capacidad de aprendizaje optima.

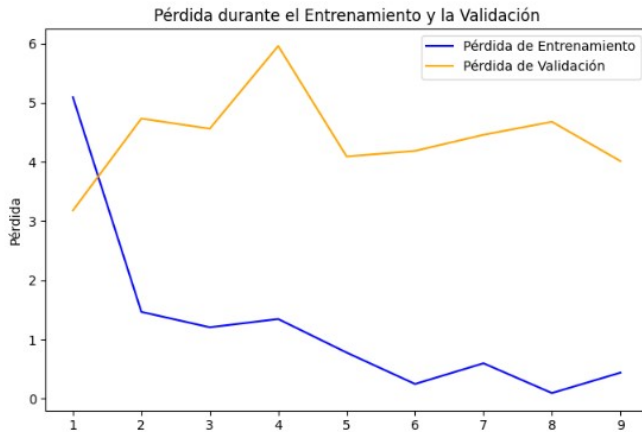


Fig. 5. Pérdida durante el Entrenamiento y la Validación

B. Evaluación con Métricas de Clasificación

Para evaluar el rendimiento del modelo, se utilizaron métricas clásicas de clasificación, como la exactitud, la precisión, sensibilidad (recall), especificidad y F1, con el fin de evaluar el rendimiento del modelo. Estas métricas proporcionan una visión detallada del desempeño del modelo en términos de su rendimiento y capacidad para clasificar correctamente las imágenes.

La exactitud representa el porcentaje total de predicciones correctas realizadas por el modelo sobre el conjunto de datos. Con un valor de 95%, el modelo demuestra una alta capacidad para clasificar correctamente las imágenes.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

La precisión indica la proporción de verdaderos positivos sobre el total de predicciones positivas. Un valor de 95.6% refleja que el modelo tiene una alta probabilidad de que los vehículos etiquetados que pertenecen a una clase especifican sea correcta, minimizando los falsos positivos.

$$Precision = \frac{TP}{TP + FP}$$

El recall mide la capacidad del modelo para identificar correctamente todas las instancias de una clase particular. Un valor de 96% indica que el modelo es efectivo en la detección de todas las imágenes relevantes de cada clase de vehículo, asegurando un buen rendimiento en la clasificación.

$$Recall = \frac{TP}{TP + FN}$$

Es la medida armónica entre la precisión y el recall, proporcionando un balance entre ambas métricas. Un F1-Score de 95% indica que el modelo mantiene un buen equilibrio entre la precisión y la capacidad de detección, lo cual es crucial para asegurar un rendimiento.

$$F1-score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

C. Evaluación adicional con Matriz de Confusión, Validación Cruzada y Curvas ROC/AUC

Para complementar la evaluación, se implementaron técnicas adicionales, como la validación cruzada con K-Fold y el análisis de curvas de ROC/AUC, junto con una matriz de confusión para una comprensión más profunda.

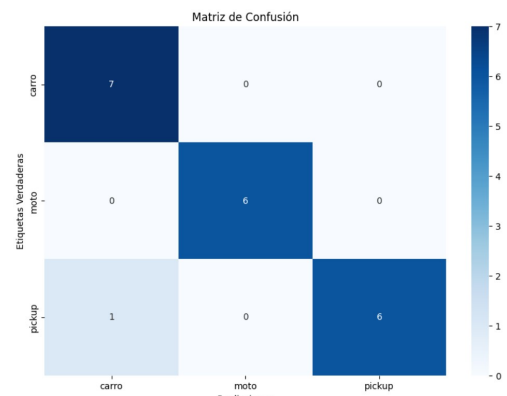


Fig. 6. Matriz de Confusión

Como se muestra observa en la Figura 6, la diagonal principal representa las clasificaciones correctas para clase otorgada, el modelo clasificó correctamente 7 instancias en la clase de carros, 6 instancias en la clase de motos y 6 instancias en la clase de pickup, sin embargo, como se observa en la clase de pickup clasifica un ejemplo como carro, una de las posibles razones es debido a las características similares que llegan a tener dichos vehículos.

La validación cruzada K-Folds se implementó como una técnica para evitar problemas de overfitting, buscando obtener una estimación más precisa sobre la capacidad de generalización del modelo. Este método divide los datos en K subconjuntos para este trabajo se asignaron 5 subconjuntos ya que al dividirlos en 5 subconjunto la mayoría de los modelos de Machine Learning se comportan de mejor manera y de igual manera para equilibrar el costo computacional para poder tener un resultado de una forma más rápida.

En cada iteración, uno de los pliegues se utiliza como conjunto de prueba, mientras que los pliegues restantes se utilizan para entrenar el modelo, repitiendo el proceso 5 veces, al tener los resultados de cada iteración se promedian para obtener una

métrica de rendimiento general del modelo teniendo como resultado un 86% , durante este proceso, se registra tanto la pérdida como la precisión en cada pliegue, la razón por lo que se decidió capturar la pérdida es poder visualizar el rendimiento de aprendizaje del modelo para poder identificar las interacciones donde tuvo mayor dificultad para ajustarse a los datos de entrenamiento.

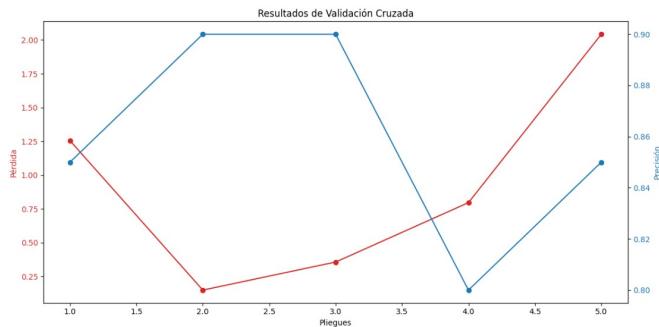


Fig. 7. Resultados de Validación Cruzada

Adicionalmente, se evaluó el modelo utilizando la curva ROC (Receiver Operation Characteristics) y el área bajo a la curva (AUC). Estas métricas es esencial para medir la capacidad de discriminación del modelo, evaluando la habilidad de diferenciar la clase de Verdaderos Positivos y Falsos Positivos respectivamente a un umbral, al momento de evaluar el modelo, se obtuvo un AUC promedio de 0.99, lo cual indica que tiene un buen rendimiento en la clasificación, indicando es consistente en la tarea de distinguir los tres diferentes vehículos, demostrando una alta sensibilidad y especificidad en el proceso.

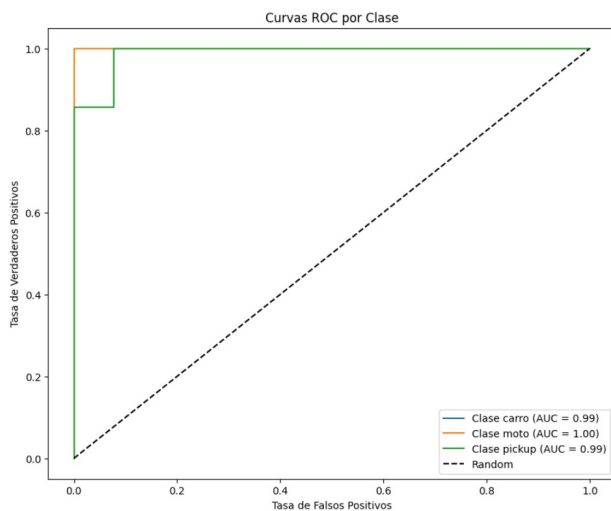


Fig. 7. Curva ROC por Clase

IV. Conclusión

El desarrollo de un modelo clasificación de vehículos utilizando redes neuronales se ha comprobado ser una propuesta eficiente para identificar automáticamente los tres diferentes tipos de vehículos en las casetas de pago, aunque como se sabe el conjunto de datos solo presenta tres tipos de vehículos de los múltiples tipos que hay. Con un enfoque en la reducción de la complejidad del modelo y la implementación de técnicas de preprocesamiento de imágenes, se ha logrado hacer que el sistema tenga una alta tasa de precisión.

REFERENCIAS

- [1] Kingma, D. Lei Ba, J. (2015). *Adam: A Method for Stochastic Optimization* Available: <https://arxiv.org/pdf/1412.6980>
- [2] OpenCV Documentation. (-). Getting Started with Images. Available: https://docs.opencv.org/4.x/db/tutorials_display_image.html
- [3] Scikit Learn (-). Roc_Curve. Available: [roc_curve — scikit-learn 1.5.2 documentation](https://scikit-learn.org/1.5.2/roccurve.html)
- [4] Chollet, F. (2012). The sequential mode. Available: [The Sequential model \(keras.io\)](https://keras.io/sequential/)
- [5] Duran, J. (2019). A Quick Guide on Preprocessing Facial Images for Neuronal Network using OpenCV using Python. Available: [A Quick Guide on Preprocessing Facial Images for Neural Networks using OpenCV in Python | by Jaime Durán | yottabytes | Medium](https://medium.com/@yottabytes/a-quick-guide-on-preprocessing-facial-images-for-neural-networks-using-opencv-in-python-by-jaime-duran-yottabytes-medium-1234567890)
- [6] Rafae, A. (2020) Main Challenges in Image Classification. Available: [Main Challenges in Image Classification | by Abdur Rafae | Towards Data Science](https://towardsdatascience.com/main-challenges-in-image-classification-by-abdur-rafay-towards-data-science-1234567890)