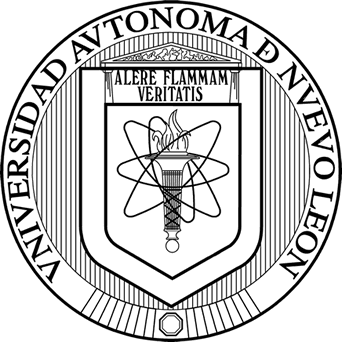
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO-MATEMÁTICAS



MAESTRÍA EN CIENCIA DE DATOS

*MEDICIÓN DEL FLUJO VEHICULAR MEDIANTE EL USO DE UNA RED NEURAL CONVOLUCIONAL*

POR

JUAN DE JESÚS AGUILAR SOLANO CON MATRÍCULA 1576327

A

04 DE AGOSTO DE 2021

Contenido

[Introducción 3](#_Toc108885474)

[Contexto del problema 4](#_Toc108885475)

[Proceso actual 4](#_Toc108885476)

[Mejora propuesta 5](#_Toc108885477)

[Marco Teórico 5](#_Toc108885478)

[Metodología 5](#_Toc108885479)

[Descripción General 5](#_Toc108885480)

[Partición del set de datos 6](#_Toc108885481)

[Descripción de la Red Neural 6](#_Toc108885482)

[Arquitectura de la red neural 6](#_Toc108885483)

[Experimentación 7](#_Toc108885484)

[Resultados 7](#_Toc108885485)

[Procesamiento de las imágenes 7](#_Toc108885486)

[Entrenamiento del modelo 8](#_Toc108885487)

[Curvas de Aprendizaje 8](#_Toc108885488)

[Conclusiones 9](#_Toc108885489)

[Bibliografía 9](#_Toc108885490)

**TABLA DE FIGURAS**

[Ilustración 1. Formato de captura de clasificación de vehículos. 4](#_Toc109496864)

[Ilustración 2. Formato de captura para medición del flujo vehicular. 4](#_Toc109496865)

[Ilustración 4. Partición de base de datos. 7](#_Toc109496866)

[Ilustración 5. Esquema de entrenamiento de la Red Neuronal. 7](#_Toc109496867)

[Ilustración 6. Arquitectura de la Red Neuronal propuesta. 7](#_Toc109496868)

[Ilustración 7. Curvas de aprendizaje del modelo. 9](#_Toc109496869)

# MEDICIÓN DEL FLUJO VEHICULAR MEDIANTE EL USO DE UNA RED NEURONAL CONVOLUCIONAL

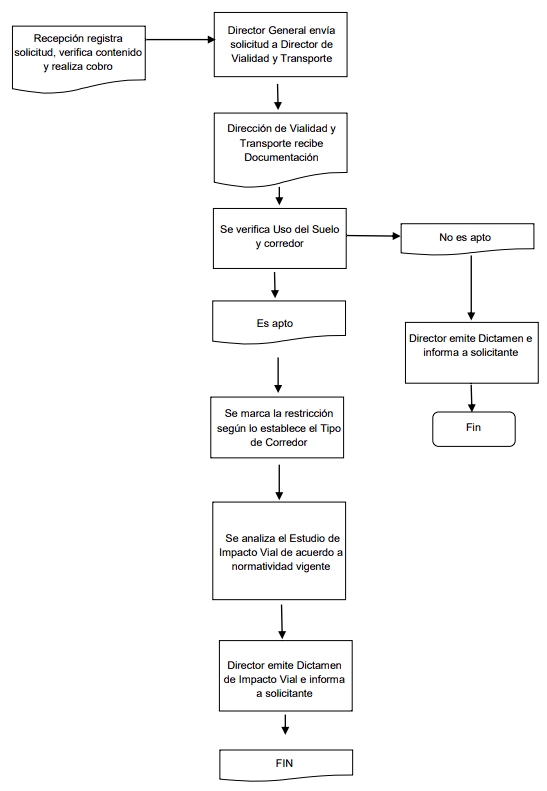
## Introducción

La mancha urbana en el área Metropolitana de Monterrey se ha intensificado en los últimos años, esto debido a la inmigración y al crecimiento económico que se ha dado en esta ciudad. Como consecuencia al crecimiento económico y poblacional, el congestionamiento en el tráfico vehicular se ha vuelto una problemática de relevancia para los todos los habitantes, pues esto repercute en pérdidas de tiempo y un gasto económico mayor derivado del congestionamiento vehicular.

En el ámbito comercial, conocer el comportamiento del tráfico vehicular en la zona es fundamental cuando se piensa en una expansión, debido a que el tránsito vehicular es un factor económico muy importante para el negocio, y por lo tanto, para la toma de decisiones.

Cuando una empresa quiere expandir su infraestructura comercial, ésta es responsable de presentar un ***“Manifiesto de Impacto Vial”*** el cual tiene como propósito principal, evaluar el impacto generado por el inmueble en la zona, y así poder proponer soluciones para mitigar los problemas derivados de dicho inmueble comercial.

El procedimiento puede variar en menor medida según el estado en el que se realiza la obra, para fines de demostración, se muestra el diagrama de flujo del proceso a seguir para realizar el manifiesto de impacto vial, proporcionado por el estado de Hermosillo a través del documento “lineamientos de impacto vial” [1].



Existen consultorías dedicadas exclusivamente al desarrollo de Manifiestos de Impacto Vial, estos llevan a cabo análisis del flujo vehicular para brindar información importante que pueda repercutir en la toma de decisiones, en este artículo solo se presentará la parte de interés para el proyecto propuesto.

## Contexto del problema

Parte del Manifiesto de Impacto Vial consiste en evaluar el flujo vehicular en una zona circundante que varía acorde a la zona y magnitud del proyecto. Para llevar a cabo la medición del dicho flujo vehicular, se sigue la siguiente metodología.

### Proceso actual

El flujo de trabajo actual para obtener la información necesaria requerida para el análisis, es la siguiente:

1. La empresa de interés brinda la ubicación de interés para la expansión de su negocio.
2. De acuerdo a la zona, se eligen niveles de profundidad, esto quiere decir, se establece un radio perimetral que se usará como área de investigación.
3. Se buscan puntos clave con mayor densidad de flujo vehicular, los cuales servirán como nodos para nuestro análisis.
4. Se instalan cámaras y éstas tomarán video las 24 horas del día durante un periodo establecido.
5. Se analizan los videos y se cuenta el número de vehículos, tipo de vehículo, entre otros campos de *manera manual*.
6. La información obtenida es resumida y se envía para un análisis posterior.

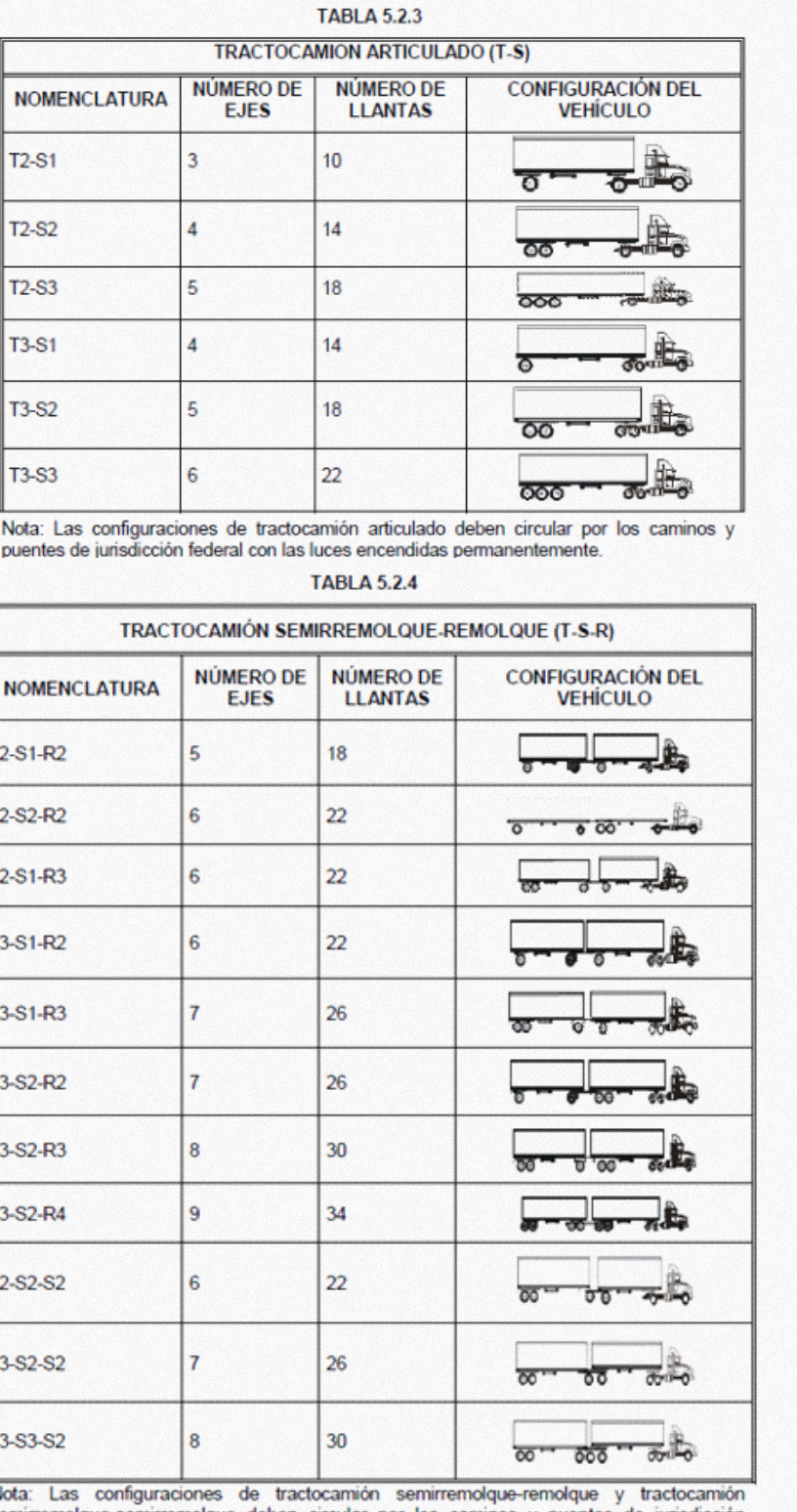


Ilustración 1. Formato de captura de clasificación de vehículos.

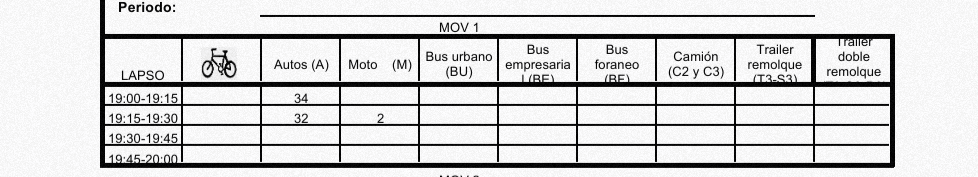


Ilustración 2. Formato de captura para medición del flujo vehicular.

### Mejora propuesta

La propuesta consiste en la implementación de un clasificador de imágenes, que pueda clasificar el tipo de auto a partir de imágenes extraídas de un video, que monitorea el flujo vehicular en determinados puntos de la ciudad.

## Alcance del proyecto

Durante este proyecto se pondrán a prueba diferentes modelos para clasificar autos a partir de imágenes, y a partir de aquí se evaluará su desempeño, tiempo de cómputo y, por lo tanto, la eficiencia del modelo para poder definir cual es el mejor para la tarea seleccionada.

## Marco Teórico

Se han realizado investigaciones de clasificación de automóviles, muchos de ellos con resultados bastante interesantes, a continuación, se muestran algunos de ellos.

Moataz-Bellah logro construir una CNN con un 98% de “Accuracy” como medida de desempeño, esto fue logrado mediante una arquitectura propuesta por el autor, como preprocesamiento solo se redimensionaron las imágenes a un tamaño de 32x32 px, esto hizo que el algoritmo fuera relativamente rápido de entrenar, esa red solo es capas de detectar vehículos, es decir, si la foto contiene vehículo o no [2].

Hicham et. al. lograron crear un clasificador de autos con cámaras de tráfico, el modelo fue una CNN la cual fue entrenada con la biblioteca keras-tensorflow para distinguir 4 clases, siendo éstos Autobús escolar, Ambulancia, Auto de Policía y Camión de transporte. El set de datos utilizados por el equipo fue creado a partir de los videos tomados por las cámaras, ésta cuenta con un total de 2400 muestras, las cuales fueron se distribuyen de manera equitativa en las diferentes clases. El desempeño del modelo fue medido mediante la métrica “Precision” logrando un valor de 0.89 [2].

Nusrat Jahan et. al. desarrollaron un sistema de clasificación de vehícules en tiempo real utilizando una Red Neuronal Convolucional, el conjunto de datos para entrenar la red, consta de un total de 2800 imágenes, de las cuales 2240 fueron usaras para entrenamiento y 560 para prueba. El preprocesado para estas imágenes fue la aumentación de datos, lo cual consiste en una serie de transformaciones realizadas a la imagen (rotación, reescalado, corte, flip vertical, flip horizontal y zoom. La red convolucional propuesta fue optimizada utilizando el algoritmo de Adam, la métrica utilizada para evaluar el modelo fueron 3, Presicion, Recall y F1-Score, los resultados se resumen en la siguiente tabla [4].

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Presicion** | **Recall** | **F1-Score** |
| Class 0 | 0.98 | 0.94 | 0.96 |
| Class 1 | 0.95 | 0.98 | 0.95 |
| Class 2 | 0.96 | 0.98 | 0.98 |
| Class 3 | 0.99 | 0.96 | 0.97 |
| Avg Accuracy | 0.97 | 0.96 | 0.96 |

Albion Krasniqui implementó diferentes modelos para clasificar imágenes de autos usando la base de datos “Stanford Cars Dataset”, el modelo consistía en clasificar el modelo del vehículo (196 clases para este conjunto de datos). Los modelos utilizados durante este trabajo fueron 7, empezando con métodos de Machine Learning con algoritmos como regresión logística, SVM, random forest y modelos de Deep Learning con redes neuronales como VGG16, MobileNet y EfficientNet B1. El resultado de los modelos fue bajo, alcanzando un accuracy de 0.045 con regresión logística hasta un 0.136 con random forest, mientras que para Deep learning los resultados se incrementaron drásticamente, obteniendo un accuracy de 0.68 a 0.71 para la red VGG16 y EfficientNet B1 respectivamente [5].

## Metodología

La metodología consiste en evaluar diferentes modelos de aprendizaje automático y Deep learning, los modelos a evaluar son los siguientes:

* Regresión lineal con PCA: Se pondrá a prueba la regresión logística en conjunto con un análisis de componentes principales (PCA) de la cual se seleccionarán 150 componentes, como preprocesado se redimensionaron las imágenes a una resolución de 128x128 pixeles.
* Gaussian Naive Bayes: Para la evaluación de este algoritmo solo se realizó la redimensión de imágenes a una resolución de 128x128 pixeles, no se ajustó ningún otro parámetro.
* Random Forest: Este algoritmo fue llevado a cabo tomando en cuenta como parámetros 60 árboles o estimadores y una profundidad máxima de 10 ramas, el algoritmo se llevó a cabo mediante multiprocesos. Como preprocesado se redimensionaron las imágenes a una resolución de 128x128 pixeles.
* Red Neural VGG16:
* Red Neural MobileNet:
* Red Neural EfficientNet B1:

### Descripción General

La metodología propuesta consiste en los siguientes pasos:

1. Conversión a imágenes: durante este proceso el video capturado se convertirá a imágenes, extrayendo los fotogramas de video, dado que la cámara graba a 24 fotogramas por segundo, se extraerán 4 fotogramas cada segundo estos serán al inicio 0.0 segundos, 0.25, 0.50, 0.75 y 1.0 segundo.
2. Redimensión de la imagen: con la finalidad de tener una entrada de tamaño fijo, todas las imágenes serán redimensionadas a un tamaño de 200x200 pixeles.
3. Entrenamiento de la red: como primer acercamiento, la red Neural será entrenada con el conjunto "Stanford Cars Dataset”, esta Red Neural tendrá una arquitectura VGG-16 y con ella se clasificarán las imágenes provenientes del video.
4. Desempeño de la red: se utilizará la métrica Accuracy para medir el desempeño de la red en el set de validación y prueba.

### Partición del set de datos

El set de datos se dividirá en 3 partes, primeramente, el set de datos original se partirá en dos partes, el set de entrenamiento (80%) y el set prueba (20%), mientras que el set de entrenamiento será dividido en 2 partes, set de entrenamiento (64%) y el set de validación (16%).

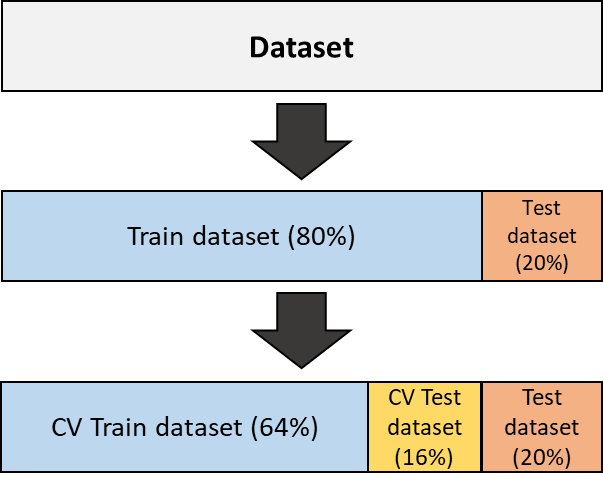


Ilustración 4. Partición de base de datos.

### Preprocesado de datos para Modelos de Machine Learning

Aquí se describe con mayor detalle el flujo de procesos a realizar para el entrenamiento de los algoritmos de regresión lineal, Naive Bayes y Random Forest.

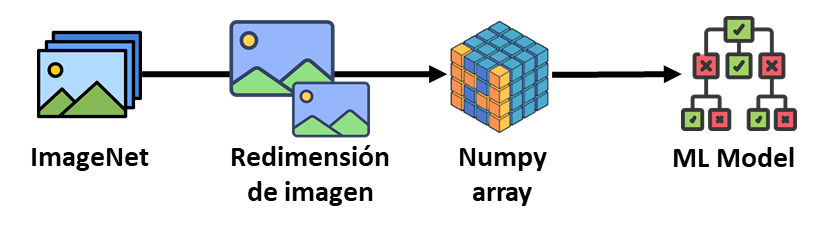


Ilustración 5. Esquema de entrenamiento de modelos de Machine Learning.

### Preprocesado de datos para las Redes Neurales

Aquí se describe con mayor detalle el flujo de procesos a realizar para el entrenamiento de la red neuronal.

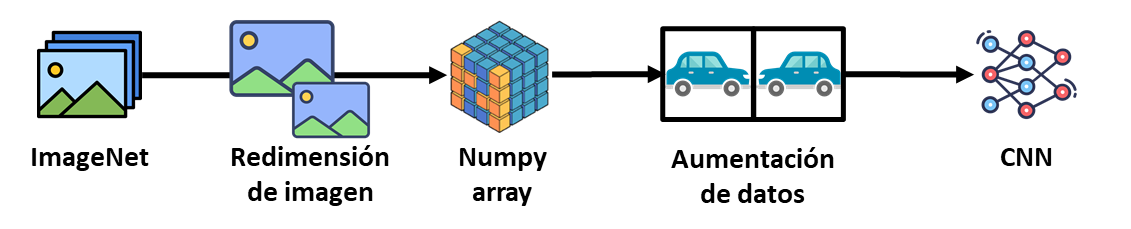


Ilustración 5. Esquema de entrenamiento de la Red Neuronal.

### Arquitectura de la red neural

La red neuronal utilizada en este proyecto se compone de las siguientes capas.



Ilustración 6. Arquitectura de la Red Neuronal propuesta.

## Experimentación

La metodología propuesta en la sección anterior se llevará a cabo de la siguiente manera:

* Extracción de fotogramas: Para la extracción de fotogramas en imágenes se utilizará el módulo de python opencv moviepy.
* Lectura de imágenes: se realizarán mediante la ruta de acceso, para ello se escanearán las carpetas que contienen las imágenes a través de las funciones del sistema mediante la biblioteca os.
* Redimensión de imágenes: Para la redimensión de las imágenes se usará del módulo de scikitlearn image la función resize.
* Red Neuronal: La red neuronal será creada y entrenada mediante la biblioteca keras con tensorflow.
* Datos para el modelo: la evaluación del modelo se llevará a cabo mediante el uso de un cross-validation con k=5, para ello primero se usará la función train\_test\_split de scikitlearn para dividir nuestro conjunto de entrenamiento y prueba.
* Entrenamiento del modelo: El entrenamiento fue llevado a cabo mediante el set de datos de entrenamiento descrito en el punto anterior, todo este proceso se llevará a cabo mediante el módulo de keras y tensorflow.
* Desempeño del modelo: todo el desempeño del modelo será medido mediante la métrica accuracy.

## Resultados

### Procesamiento de las imágenes

El procesamiento de las imágenes se realizó con relativa facilidad, el tiempo que tomó en transformar y cargar el set de imágenes fue de alrededor de un minuto, el conjunto de datos fue guardado en un arreglo numpy cuya dimensión es de (17760, 32, 32, 3).

### Construcción del modelo

La construcción del modelo se llevó a cabo con la configuración mencionada en la sección anterior, el número total de parámetros entrenables es de 55,938, la configuración completa se muestra a continuación:

**Model: "sequential\_1"**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

====================================================================

conv2d\_2 (Conv2D) (None, 28, 28, 32) 2432

max\_pooling2d\_2 (MaxPooling 2D) (None, 14, 14, 32) 0

conv2d\_3 (Conv2D) (None, 10, 10, 32) 25632

max\_pooling2d\_3 (MaxPooling 2D) (None, 5, 5, 32) 0

flatten\_1 (Flatten) (None, 800) 0

dense\_3 (Dense) (None, 32) 25632

dropout\_2 (Dropout) (None, 32) 0

dense\_4 (Dense) (None, 64) 2112

dropout\_3 (Dropout) (None, 64) 0

dense\_5 (Dense) (None, 2) 130

====================================================================

Total params: 55,938

Trainable params: 55,938

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Entrenamiento del modelo

El entrenamiento de la red se llevó a cabo en un tiempo de aproximadamente 6 minutos, con un CPU AMD Ryzen 5 3600 6-Core y 16 GB RAM. En 50 epocas se llegó a un accuracy de validación de 0.98, mientras que el accuracy del set de prueba fue de 0.97.

### Curvas de Aprendizaje

Las curvas de aprendizaje accuracy y los, se muestran a continuación.

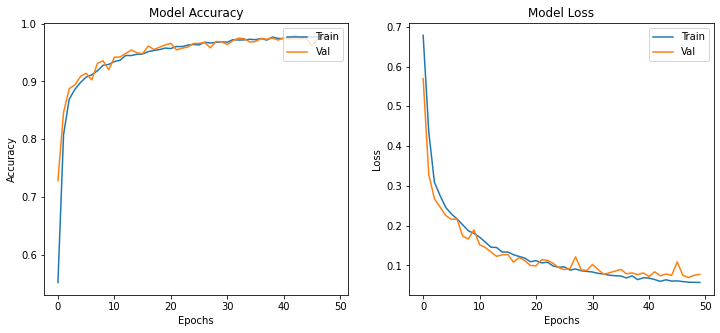


Ilustración 7. Curvas de aprendizaje del modelo.

De estas curvas podemos observar que el modelo se sigue ajustando y no hay caídas en la métrica del conjunto de validación, por lo que podemos concluir que no existe sobre ajuste en esta Red Neural, lo cual hace que se desempeñe bastante bien en el set de prueba.

## Conclusiones

* El entrenamiento del modelo fue entrenado y obteniendo resultados muy buenos, alcanzando un Accuracy de 97%.
* El modelo no presentó sobre entrenamiento sobre la base de datos de entrenamiento.
* El tiempo de entrenamiento para este modelo fue de alrededor de 8 minutos, esto se debe a la resolución de imagen tomada, lo cual reduce considerablemente el cómputo.

## Bibliografía

[1] IMPLAN Hermosillo. (s. f.). Ayuntamiento de Hermosillo. Recuperado 20 de julio de 2022, de https://www.implanhermosillo.gob.mx/

[2] Moataz-Bellah Ahmed. (2022, 6 julio). Vehicle Detection 98 % Accuracy. Kaggle. https://www.kaggle.com/code/moatazbellahahmed/vehicle-detection-98-accuracy

[3] Hicham, B., Ahmed, A., & Mohammed, M. (2018). Vehicle Type Classification Using Convolutional Neural Network. 2018 IEEE 5th International Congress on Information Science and Technology (CiSt). doi:10.1109/cist.2018.8596500

[4] Jahan, N., Islam, S., & Foysal, M. F. A. (2020). Real-Time Vehicle Classification Using CNN. 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT). doi:10.1109/icccnt49239.2020.9225623

[5] Krasniqi, A. (2022, 15 abril). Vehicle Classification using Machine Learning. Medium. Recuperado 23 de julio de 2022, de https://medium.com/@albionkrasniqi22\_80133/vehicle-classification-742403117f43