

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO-MATEMÁTICAS



MAESTRÍA EN CIENCIA DE DATOS

*COMPARACIÓN DE MODELOS DE MACHINE Y DEEP
LEARNING PARA LA CLASIFICACIÓN DE AUTOMÓVILES
A PARTIR DE IMÁGENES*

POR

JUAN DE JESÚS AGUILAR SOLANO CON MATRÍCULA
1576327

A

07 DE JULIO DE 2021

CONTENIDO

Comparación de modelos de Machine y Deep Learning para la Clasificación de Automóviles a Partir de Imágenes	4
Introducción	4
Contexto del problema	5
Proceso actual	5
Mejora propuesta.....	6
Alcance del proyecto.....	6
Marco Teórico	6
Set de datos.....	7
Análisis descriptivo del set de datos	8
Metodología	9
Partición del set de datos.....	9
Preprocesado de datos para Modelos de Machine Learning	9
Preprocesado de datos para las Redes Neuronales	9
Arquitecturas de las redes neuronales	10
Experimentación	11
Parámetros utilizados en algoritmos	11
Hardware utilizado para el entrenamiento del modelo.....	12
Resultados	12
Procesamiento de las imágenes.....	12
Análisis de componentes principales	12
Regresión logística con PCA	13
Gaussian Naive Bayes.....	13
Random Forest (Criterio Gini)	13
Random Forest (Criterio Entropy).....	13
Red Neural VGG16.....	13
Red Neural MobileNetV2	14
EfficientNetB1	14
Resumen de resultados.....	15
Conclusiones	15
Trabajo a futuro	16
Bibliografía	16

TABLA DE FIGURAS

Ilustración 1. Formato de captura de clasificación de vehículos.	5
Ilustración 2. Formato de captura para medición del flujo vehicular.	5
Ilustración 3. Carpetas contenidas en el set de datos.	7
Ilustración 4. Algunos autos del set de datos con su respectiva etiqueta.	7
Ilustración 5. Cantidad de autos por categoría.	8
Ilustración 6. Cantidad de autos por año del modelo.	8
Ilustración 7. Partición de base de datos.	9
Ilustración 8. Esquema de entrenamiento de modelos de Machine Learning.	9
Ilustración 9. Esquema de entrenamiento de la Red Neuronal.	9
Ilustración 10. Arquitectura de la Red Neuronal VGG16	10
Ilustración 11. Arquitectura de la Red Neuronal MobileNetV2.	10
Ilustración 12. Arquitectura de la Red Neuronal EfficientNet	11
Ilustración 13. Gráfico de codo PCA.	13
Ilustración 14. Curvas de aprendizaje para la red VGG16.	14
Ilustración 15. Curvas de validación para la Red Neuronal MobileNetV2.	14
Ilustración 16. Curvas de aprendizaje para la Red Neuronal EfficientNetB1.	14
Ilustración 17. Curvas de validación para los diferentes modelos neuronales.	15

COMPARACIÓN DE MODELOS DE MACHINE Y DEEP LEARNING PARA LA CLASIFICACIÓN DE AUTOMÓVILES A PARTIR DE IMÁGENES

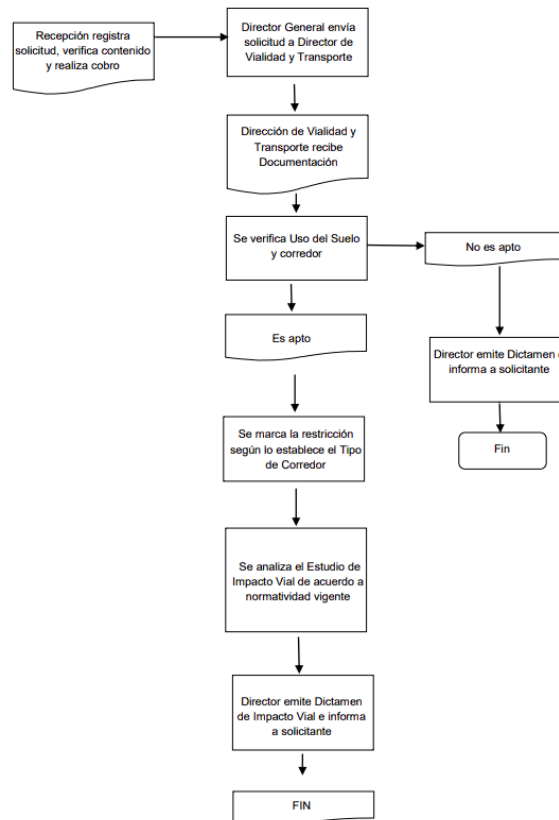
INTRODUCCIÓN

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.

TABLA 5.2.3

TRACTOCAMION ARTICULADO (T-S)			
NOMENCLATURA	NÚMERO DE EJES	NÚMERO DE LLANTAS	CONFIGURACIÓN DEL VEHICULO
T2-S1	3	10	
T2-S2	4	14	
T2-S3	5	18	
T3-S1	4	14	
T3-S2	5	18	
T3-S3	6	22	

Nota: Las configuraciones de tractocamión articulado deben circular por los caminos y puentes de jurisdicción federal con las luces encendidas permanentemente.

Ilustración 1. Formato de captura de clasificación de vehículos.

Periodo: _____									
MOV 1									
LAPSO		Autos (A)	Moto (M)	Bus urbano (BU)	Bus empresarial (BE)	Bus foráneo (BE)	Camión (C2 y C3)	Trailer remolque (T3-S3)	Trailer doble remolque
19:00-19:15		34							
19:15-19:30		32	2						
19:30-19:45									
19:45-20:00									

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 modelo para implementarlo en la tarea descrita en la sección anterior.

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 vehículos en tiempo real utilizando una Red Neuronal Convolutiva, el conjunto de datos para entrenar la red, consta de un total de 2800 imágenes, de las cuales 2240 fueron usadas 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 convolutiva 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].

SET DE DATOS

El set de datos a utilizar durante este proyecto es “The Stanford Cars Dataset”, una gran colección de imágenes de autos. Consiste en 16,185 imágenes etiquetadas en 196 clases basado en la marca del auto, modelo y año. Sin embargo, dada la naturaleza del proyecto, se hará una modificación a las etiquetas, ahora el set de datos contendrá solamente 10 clases, las cuales son las siguientes:

[Cab, Convertible, Coupe, Hatchback, Lujo, Minivan, Sedan, SUV, Van y Wagon]

Este set de datos se encuentra en carpetas y subcarpetas, de donde se extrajo las correspondientes etiquetas y modelo de auto. La configuración es la siguiente:

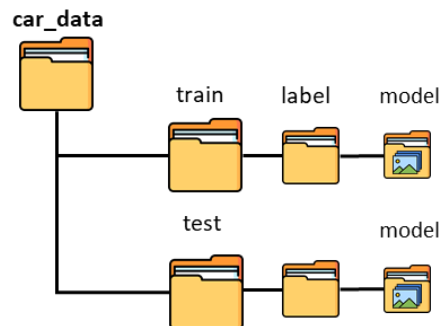


Ilustración 3. Carpetas contenidas en el set de datos.

Algunos de estos ejemplos se muestran a continuación

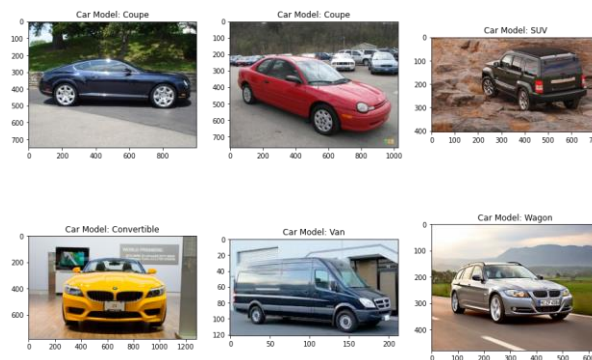


Ilustración 4. Algunos autos del set de datos con su respectiva etiqueta.

Análisis descriptivo del set de datos

A continuación, se muestra la composición de las imágenes de la base de datos por clase o categoría. Como puede observarse en los gráficos de barra, son los autos sedan los que componen en gran mayoría la base de datos, seguido de la clase SUV

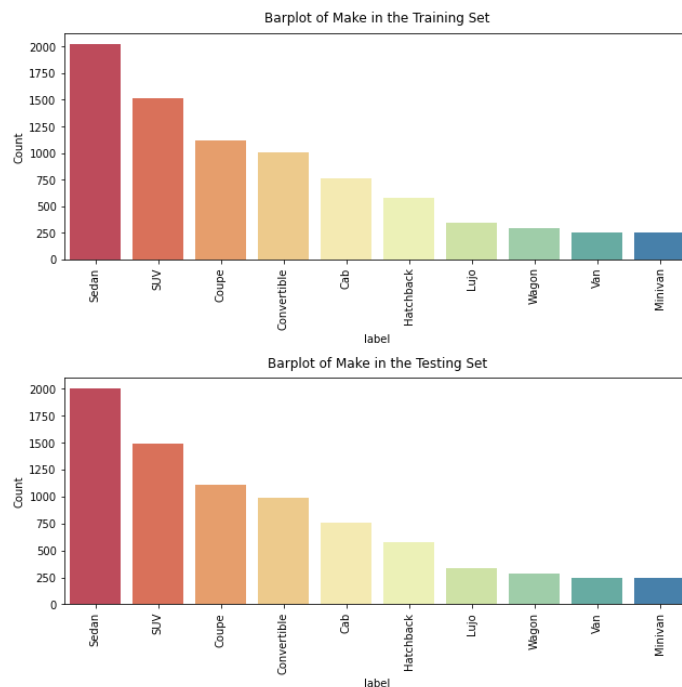


Ilustración 5. Cantidad de autos por categoría.

Por otra parte, los autos que componen este set de datos, en su mayoría son modelos del año 2012, siendo casi 5000 los autos totales contenidos.

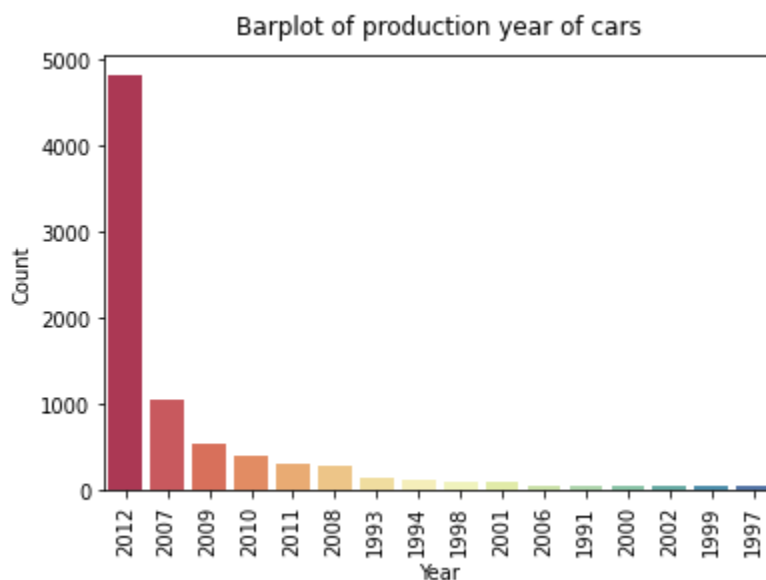


Ilustración 6. Cantidad de autos por año del modelo.

METODOLOGÍA

La metodología consiste en evaluar diferentes modelos de aprendizaje automático y Deep learning, la metodología general se presenta en las siguientes secciones.

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 el cual contiene 8144 imágenes (50%), el set de validación 4000 imágenes (25%) y el set de prueba, el cual contiene 4041 imágenes (25%).

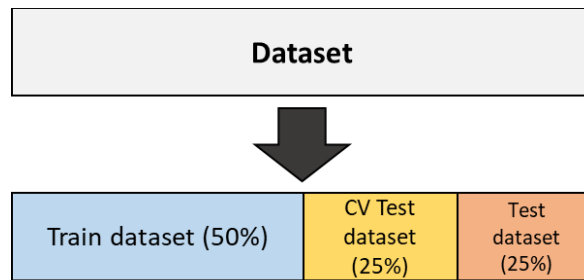


Ilustración 7. Partición de base de datos.

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.

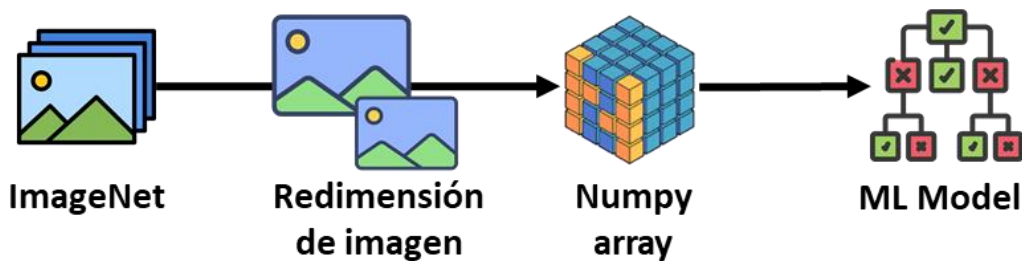


Ilustración 8. Esquema de entrenamiento de modelos de Machine Learning.

Preprocesado de datos para las Redes Neuronales

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

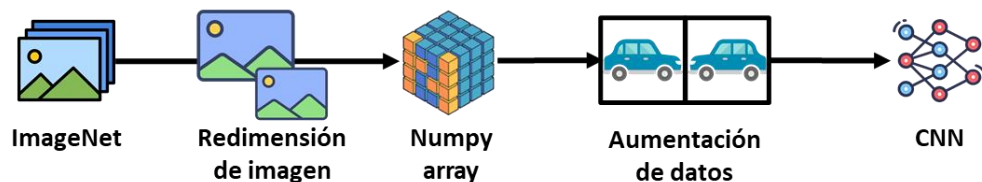


Ilustración 9. Esquema de entrenamiento de la Red Neuronal.

Éste es uno de los modelos más eficientes, esta red ha sido entrenada en el set de datos ImageNet. En particular, esta red escala uniformemente las dimensiones de ancho, profundidad y resolución manteniendo un balance. Reduce las dimensiones de los problemas de clasificación de manera significativa.

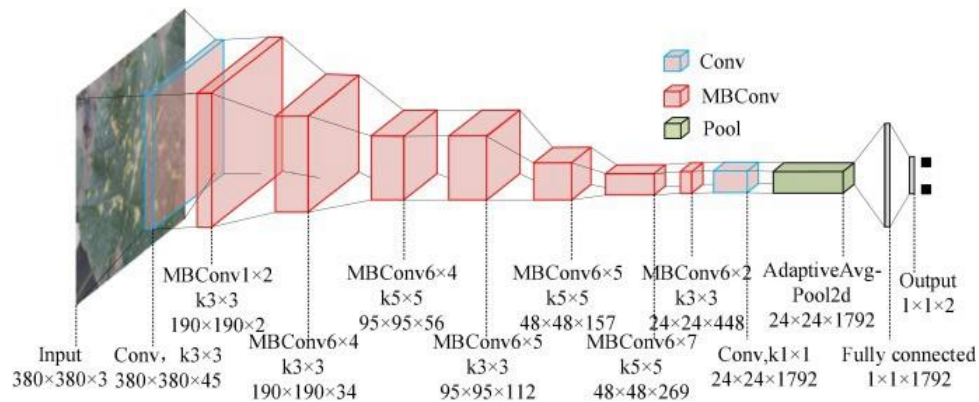


Ilustración 12. Arquitectura de la Red Neuronal EfficientNet

EXPERIMENTACIÓN

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

- 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 keras mediante la función `flow_from_directory`.
- Redimensión de imágenes: Para la redimensión de las imágenes se usará del módulo de scikitlearn image la función `resize`.
- *Aumentación de datos: este solo se llevará a cabo para las redes neuronales, se llevará a cabo mediante el módulo de ImageDataGenerator de la biblioteca keras.
- 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 $k=5$.
- Entrenamiento del modelo: El entrenamiento fue llevado a cabo mediante el set de datos de entrenamiento descrito en el punto anterior, se evaluará cada uno de los modelos descritos anteriormente, tando los modelos de machine learning como los modelos de Deep learning.
- Desempeño del modelo: todo el desempeño del modelo será medido mediante la métrica accuracy.

Parámetros utilizados en algoritmos

- 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 píxeles.

- Red Neural VGG16: Para las redes neuronales se utilizará un generador de imágenes, lo cual hará una aumentación de datos, los parámetros para dicha función fue una normalización de valores, es decir se dividió todo por 255, se utilizó un zoom_range de 0.2, un rango de rotación de 5, y una rotación horizontal, además se redimensionó la imagen a una dimensión de 128x128 píxeles. La red Neural consiste en un modelo preentrenado a la cual se le añadió una capa densa de 128 neuronas, con un activador relu y el kernel he_uniform como inicializador, además se añadió una capa de salida con 10 neuronas que representan las clases utilizadas en este proyecto, con softmax como función de activación. Para el compilado del modelo se utilizó la función de pérdida categorical_crossentropy, el optimizador de Adam y la métrica accuracy. Para la parte del entrenamiento se utilizaron 20 épocas, y un generador para los datos de validación del modelo.
- Red Neural MobileNet: El preprocesamiento utilizado para la red neural es el mismo que para la VGG16, al modelo se le agregó una capa Pooling2D, una capa de normalización de Batch, una capa densa con 1024 neuronas con relu como función de activación, otra capa de normalización de batch y por último una capa densa con 10 neuronas con softmax como función de activación (capa de salida). El compilado y entrenamiento del modelo se hizo exactamente como la red anterior.
- Red Neural EfficientNet B1: Para esta red neural se tomo cuenta el mismo preprocesamiento y los mismos parámetros de compilado y entrenamiento. A la red pre entrenada se le agregaron 2 capas, una capa Pooling2D y una capa de salida Densa con 10 neuronas y softmax como función de activación.

Hardware utilizado para el 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 épocas se llegó a un accuracy de validación de 0.98, mientras que el accuracy del set de prueba fue de 0.97.

RESULTADOS

Procesamiento de las imágenes

El procesamiento de las imágenes para los algoritmos de machine learning 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 (8144, 49152) para el set de entrenamiento y de (8041, 49152) para el set de prueba.

Análisis de componentes principales

Este análisis se llevó a cabo con relativa facilidad, sin embargo, el tiempo requerido para llevar este análisis fue de alrededor de 12 minutos seleccionando todas las componentes del sistema. El gráfico de codo se muestra a continuación:

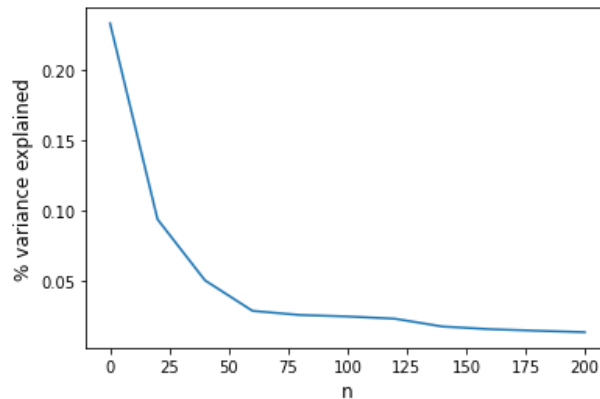


Ilustración 13. Gráfico de codo PCA.

El gráfico muestra una gran caída en la explicación de la varianza a partir de las 60 componentes aproximadamente. Para asegurar que la pérdida de información no sea significativa, se decidió realizar un PCA con 150 componentes, el cual explica el 81% de la varianza total y tomando un total de 1 minuto y 5 segundos en calcular las componentes, este análisis fue utilizado para realizar una regresión logística.

Regresión logística con PCA

El proceso de entrenamiento para la regresión logística fue bastante corto, tomando alrededor de 1 minuto en entrenarse, y obtener la métrica del modelo "Accuracy", del cual se obtuvo un 29% en el set de entrenamiento y un 23% en el set de validación.

Gaussian Naive Bayes

Este es uno de los algoritmos más simples de implementar, pues no requiere de muchos parámetros para ser entrenado, sin embargo, para este set de datos, el modelo presentó un accuracy del 13% para el set de entrenamiento y un 12% para el set de prueba, mostrando un bajo desempeño.

Random Forest (Criterio Gini)

Este algoritmo se llevó a cabo con rapidez, puesto que el entrenamiento y evaluación del modelo llevó a cerca de 12 segundos. El desempeño del modelo no fue el deseado, puesto que éste presenta problemas de sobreajuste y por ende presenta buen desempeño en el set de entrenamiento (77%) y un mal desempeño en el set de prueba (28%).

Random Forest (Criterio Entropy)

Este modelo a diferencia del que contiene el criterio de Gini, tarda más en ajustarse y además también presenta problemas de sobreajuste, obteniendo un accuracy de 88% y 27% para el set de entrenamiento y set de prueba respectivamente.

Red Neural VGG16

Esta red neural fue la primera en ser entrenada, el tiempo dedicado al entrenamiento fue de 150 minutos, lo cual dificulta su entrenamiento en equipos convencionales, sin embargo, el accuracy aumentó considerablemente comparado con los algoritmos de machine learning, teniendo un 68% y 52% en el set de entrenamiento y validación.

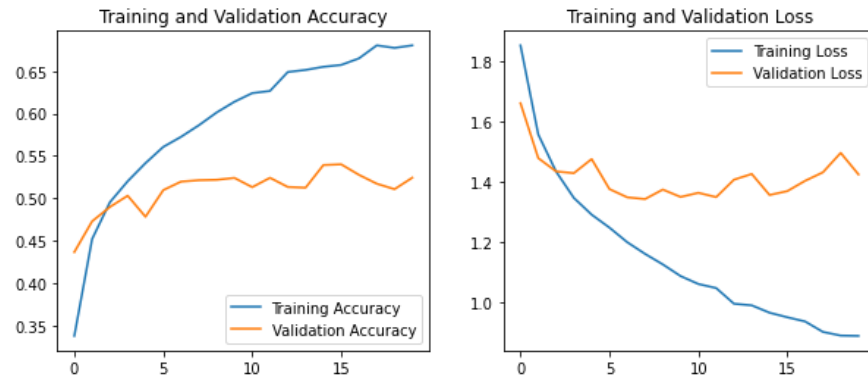


Ilustración 14. Curvas de aprendizaje para la red VGG16.

Red Neural MobileNetV2

Este modelo a diferencia de la red VGG16, tardó menos en entrenarse (45 minutos), lo cual representa un 30% con respecto a la red VGG16. El resultado de la métrica fue de 62% y 56% para el set de entrenamiento y set de validación, las curvas de aprendizaje se muestran a continuación.

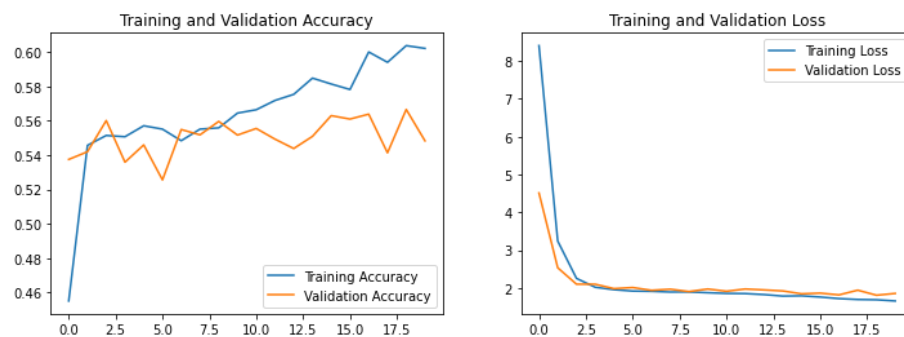


Ilustración 15. Curvas de validación para la Red Neuronal MobileNetV2.

EfficientNetB1

Este modelo fue el más tardado de todos, con un tiempo de entrenamiento de 233 minutos. Los resultados obtenidos fueron mejores que los modelos anteriores, obteniendo un accuracy de 71% y 63% para el set de entrenamiento y validación respectivamente.

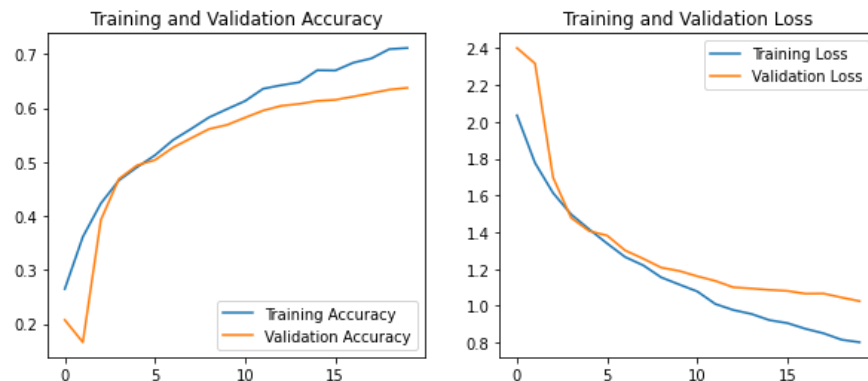


Ilustración 16. Curvas de aprendizaje para la Red Neuronal EfficientNetB1.

Resumen de resultados

A continuación, se presenta una tabla con las características obtenidas de cada modelo.

Modelo	Accuracy (train)	Accuracy (test)	Tiempo (min)
Regresión lineal + PCA	29%	23%	1
Gaussian Naive Bayes	13%	12%	1.5
Random Forest (Gini)	77%	28%	0.2
Random Forest (Entropy)	88%	27%	1.7
Red Neural VGG16	68%	52%	150
Red Neural MobileNet V2	62%	56%	45
Red Neural EfficientNet B1	71%	63%	233

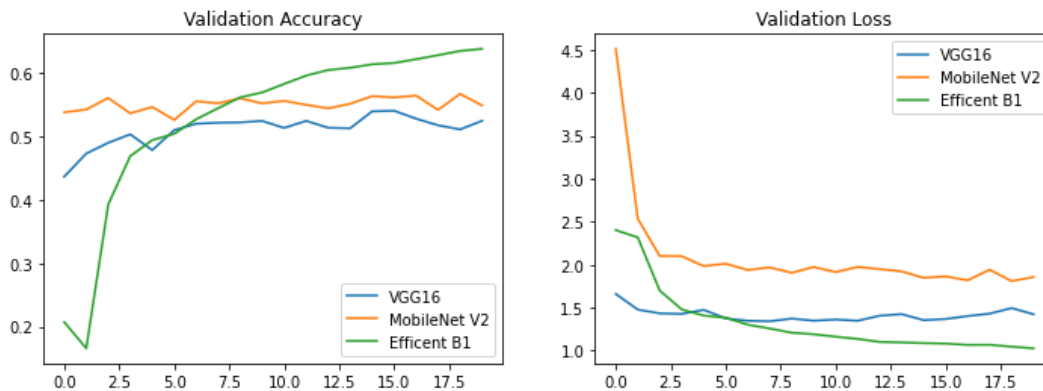


Ilustración 17. Curvas de validación para los diferentes modelos neuronales.

CONCLUSIONES

Durante este proyecto se logró implementar diferentes modelos para la clasificación de imágenes de autos, siendo el punto más relevante del proyecto, la gran brecha en el desempeño de los modelos de machine learning y Deep learning, donde los modelos de machine learning son bastante rápidos de entrenar pero con un desempeño muy pobre para la aplicación abordada en este proyecto y siendo las redes neuronales las que presentan mucho mejor desempeño, además, es la red Efficient B1 la que mejor desempeño mostró de todos los modelos con un accuracy de 63% en el set de validación, sin embargo, esta métrica puede mejorarse significativamente, puesto que las curvas de validación sugieren incrementar las épocas y las características de los datos (esto puede verse en el estancamiento de la curva de validación) a través de el aumento de la resolución de las imágenes, esto último no se abordó durante el proyecto debido al gran coste computacional que representa, pues los tiempos aumentan de manera considerablemente (casi 10 veces los tiempos actuales).

Puntos importantes del proyecto

- Se evaluaron diferentes modelos de machine y deep learning para la clasificación de auto.
- Existe una diferencia significativa entre los algoritmos de machine learning y los de Deep learning, siendo estos últimos los que mejor rendimiento presentan.

- El entrenamiento del modelo es considerablemente más tardado para las redes neuronales implementadas, incluso si éstas ya vienen pre entrenadas.
- La red EfficientNetB1 es la arquitectura que presenta los mejores resultados con respecto a los algoritmos anteriores.
- Los resultados de las métricas no son los resultados más favorables, las curvas de aprendizaje sugieren aumentar las características de los datos, esto se puede lograr a través del aumento en la resolución de las imágenes.

TRABAJO A FUTURO

Este proyecto es el inicio de una serie de pasos para lograr el objetivo inicial de contabilizar autos por clase, el siguiente paso consiste en focalizarse en la Red Neural Efficient B1 para optimizar sus parámetros y mejorar la clasificación de éste, además se trabajará con una mayor resolución y se dispondrá de un equipo de cómputo de mayor nivel para llevar a cabo las tareas mencionadas.

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