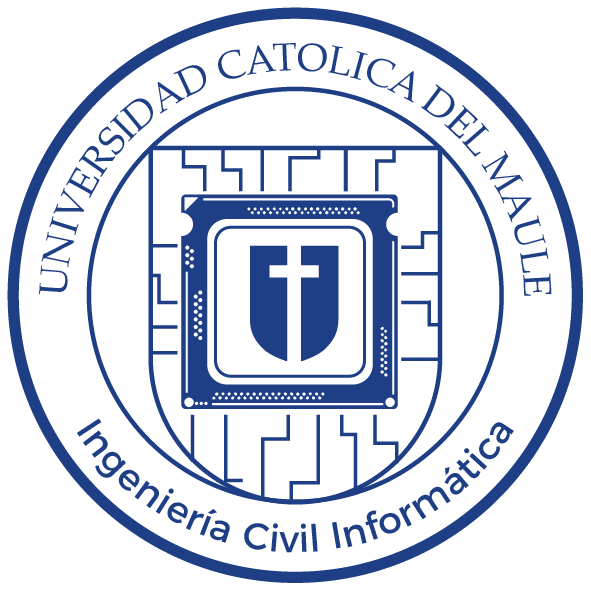
* + - 1. 

## Proyecto Final de IA

* + - 1. Comparación de Modelos de Redes Neuronales para el Control de Acceso Vehicular: Un Estudio Comparativo entre CNN, ResNet50 y VGG16
      2. Integrantes: Benjamin Charnock

Luciano Gonzalez

Daniel Pereira

Julian Arriola

* + - 1. Nombre de la Carrera: Ingeniería Civil informática

Nombre del Ramo: Inteligencia Artificial

* + - 1. Nombre del/ la docente: Sergio Hernández
      2. Fecha de entrega: 13 de diciembre de 2023

**Índice de Contenidos**

* + - 1. [**Introducción**](#_gjdgxs) 3
      2. [**Desarrollo**](#_30j0zll) 4
      3. [1. Problemática](#_3znysh7) 4
      4. [2. Requerimientos de software y hardware](#_2et92p0)5
      5. **Análisis Exploratorio** 6
      6. 1. Conjunto de datos, Filtrado y selección de clases 6
      7. 2. Estadísticas básicas de los datos 7
      8. **Preprocesamiento de datos**9

**Selección de modelos** 10

1. CNN 10
2. ResNet 50 12
3. VGG16 14

**Evaluación** 16

1. ResNet 50 16
2. VGG16 19
3. CNN 22
   * + 1. [**Conclusión**](#_1fob9te) 25

# Introducción

El procesamiento de imágenes se ha convertido en una importante área de aplicación de la inteligencia artificial. La capacidad de las máquinas para interpretar y extraer información valiosa de imágenes ha revolucionado campos como la medicina, la seguridad, etc.

En este contexto, las redes neuronales surgen como protagonistas destacadas en el campo del procesamiento de imágenes. Inspiradas en la estructura y funcionamiento del cerebro humano, estas redes aprenden patrones complejos y representaciones visuales, permitiendo la identificación de objetos, el reconocimiento facial, y la interpretación de contenido visual con una gran precisión.

En este informe, nos centraremos en la identificación y resolución de una problemática centrada en el procesamiento de imágenes mediante la aplicación de redes neuronales, siendo la problemática en este caso el control de acceso de autos y camiones, control que en determinadas circunstancias presenta vulnerabilidades explotables que puede generar brechas en las medidas de seguridad. CNN, ResNet50 y VGG16 serán los modelos que se utilizarán, siendo el objetivo principal comparar sus arquitecturas a nivel de precisión, rendimiento, etc.

Se busca en definitiva ofrecer una solución a la problemática planteada, además de ofrecer una visión clara de la efectividad y adaptabilidad de cada modelo.

# Desarrollo

* **Problemática:**

En el ámbito del control de acceso de autos y camiones, nos enfrentamos a desafíos significativos que afectan tanto la seguridad como la eficacia de los sistemas existentes. Los métodos tradicionales de autenticación, como tarjetas magnéticas y códigos de barras, presentan vulnerabilidades explotables que pueden ser aprovechadas para eludir las medidas de seguridad establecidas. Este problema se agrava debido a la incapacidad de los sistemas actuales para adaptarse a situaciones dinámicas y adversas, como cambios en las condiciones de iluminación, obstáculos visuales y la diversidad de tipos de vehículos presentes. Esta falta de adaptabilidad compromete la precisión de los controles de acceso, generando riesgos significativos de accesos no autorizados.

Esta problemática impacta directamente a los administradores de instalaciones de control de acceso vehicular, así como a los usuarios finales que confían en la seguridad de dichos sistemas. Los accesos no autorizados podrían resultar en situaciones de riesgo, pérdida de propiedad o incluso amenazas a la seguridad pública, dependiendo del contexto de la instalación.

El avance tecnológico en el campo de las redes neuronales ofrece una oportunidad única para abordar estas vulnerabilidades. Las capacidades de aprendizaje y adaptación de las redes neuronales pueden proporcionar soluciones más robustas y seguras, capaces de enfrentar escenarios desafiantes y mejorar la precisión del control de acceso vehicular.

Sin embargo, es crucial abordar los desafíos asociados con la implementación de estas tecnologías, como la integración con sistemas existentes, costos asociados y garantizar la eficacia en entornos dinámicos. Superar estos obstáculos permitirá avanzar hacia soluciones más seguras y adaptables, mejorando significativamente la calidad y confiabilidad de los sistemas de control de acceso vehicular.

* **Requerimientos de Software:**

Software de Procesamiento de Imágenes: Se requiere un software especializado, como TensorFlow, para analizar características distintivas de las razas caninas. TensorFlow Image facilitará la carga, preprocesamiento y manipulación eficiente de imágenes, crucial para implementar redes neuronales convolucionales (CNN) en la clasificación de razas.

Entorno de Desarrollo: Utilizamos Jupyter Notebook y Visual Studio Code como entornos de desarrollo integrados (IDE), brindando versatilidad y permitiendo una comparación eficiente del rendimiento durante el desarrollo.

Bases de Datos de Imágenes: Acceso a bases de datos con imágenes representativas de las razas dachshund y beagle es esencial para entrenar y validar el modelo de clasificación.

* **Requerimientos de Hardware:**

**GPU:** Se utiliza una GPU compatible con TensorFlow para acelerar eficientemente el entrenamiento de la red neuronal más específica. En este caso, se emplean tarjetas gráficas GTX 1650 y 1660 para obtener un rendimiento óptimo.

**CPU Potente:** Se utiliza una CPU de alto rendimiento para la manipulación eficiente de datos y operaciones. En este proyecto, se emplean procesadores Intel Core i5.

**Memoria RAM:** Se utilizan al menos 16 GB de RAM para manejar grandes conjuntos de datos y procesos intensivos en memoria.

**Conexión a Internet:** Se utiliza para acceder a recursos en línea y es esencial para la ejecución básica del proyecto, como la obtención de imágenes y actualizaciones de software.

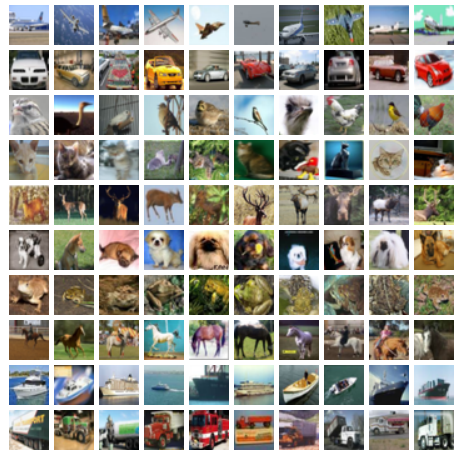
**Análisis Exploratorio de Datos**

* **Conjunto de Datos**

En este análisis, utilizamos el conjunto de datos CIFAR-10, una base ampliamente reconocida en el ámbito de la visión por computadora. CIFAR-10 consta de imágenes que abarcan 10 clases distintas, como aviones, automóviles, aves, gatos, ciervos, perros, entre otros. Originalmente diseñado para tareas de clasificación multiclase, cada clase representa un tipo específico de objeto o escenario, proporcionando diversidad y complejidad para evaluar y entrenar modelos de aprendizaje automático.

El conjunto de datos se divide en conjuntos de entrenamiento y pruebas, donde cada imagen tiene una resolución de 32x32 píxeles y está etiquetada con una de las 10 categorías mencionadas anteriormente. En este análisis, nos enfocamos exclusivamente en dos clases específicas: "Automobile" y "Truck". Esta selección se fundamenta en la relevancia práctica para aplicaciones como la detección de vehículos, lo que permite una exploración más específica y dirigida en el contexto de nuestro análisis.

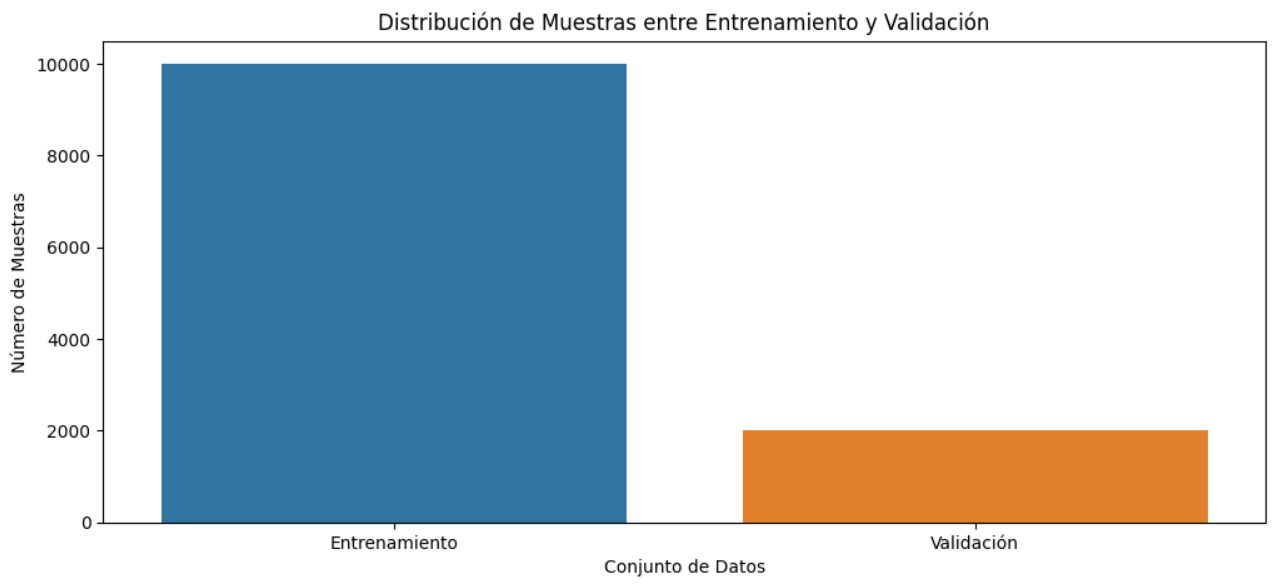
* **Filtrado y Selección de Clases**

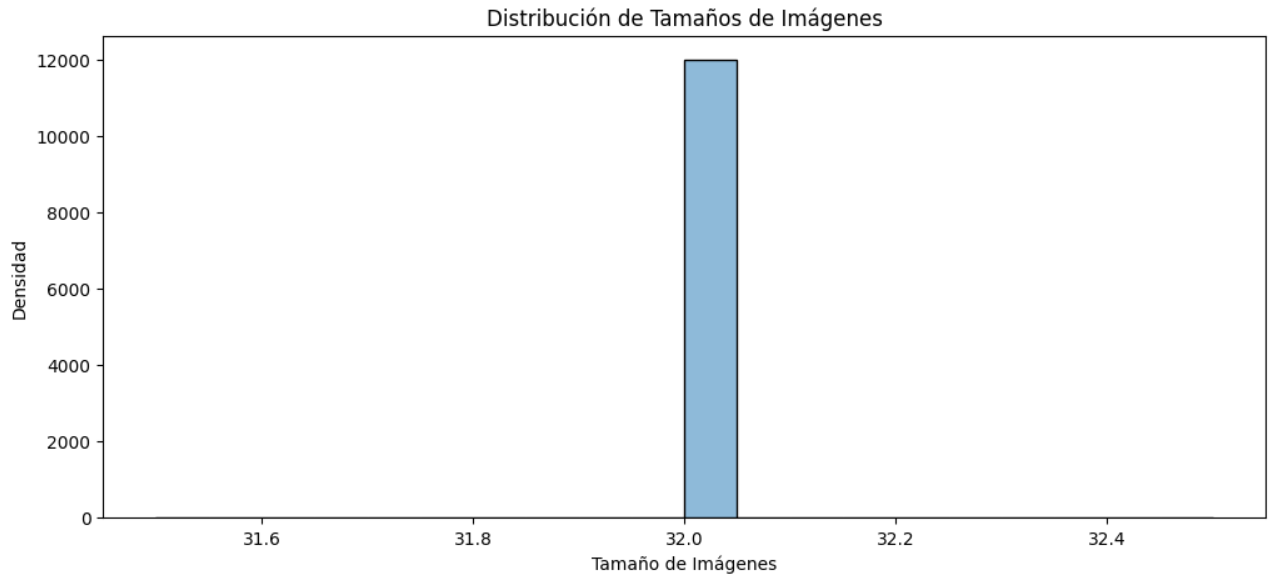
La decisión de centrarse en las clases "Automobile" y "Truck" se basa en la importancia de la tarea. Al restringir el análisis a estas clases, se busca simplificar la tarea de clasificación y concentrar los esfuerzos en un problema específico relacionado con la identificación de vehículos en imágenes.

Conjunto de datos que contiene CIFAR-10

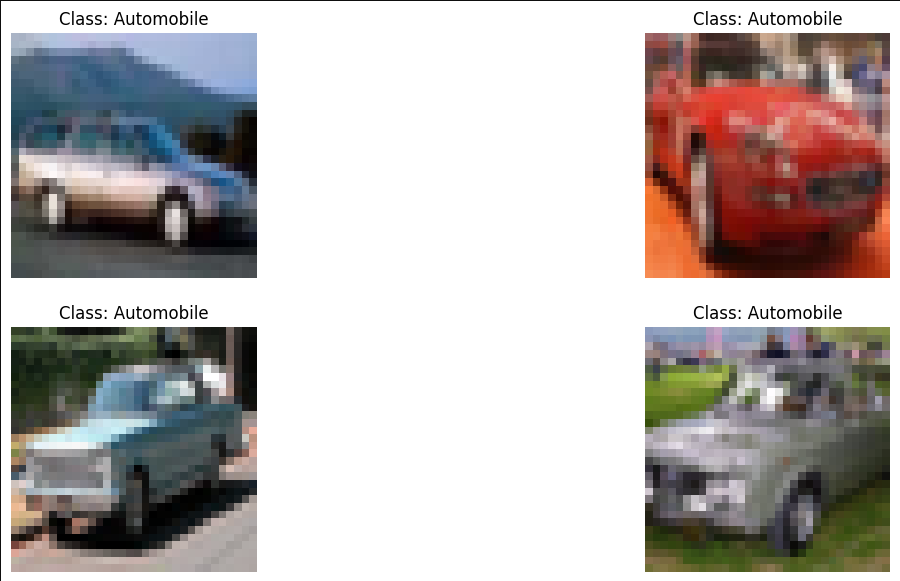
* **Estadísticas Básicas De Los Datos:**

| **Número total de imágenes antes del filtrado** | 60.000 |
| --- | --- |
| **Número total de imágenes después del filtrado** | 12.000 |
| **Número de imágenes de entrenamiento** | 10.000 |
| **Número de imágenes de prueba** | 2.000 |
| **Total de autos (Entrenamiento)** | 5.000 |
| **Total de camiones (Entrenamiento)** | 5.000 |
| **Total de autos (Prueba)** | 1.000 |
| **Total de camiones (Prueba)** | 1.000 |



******

Ejemplos de Imágenes a utilizar





**Pre-Procesamiento de Datos**

Estos pasos de preprocesamiento aseguran que las imágenes estén en un formato adecuado y que las etiquetas se representen de manera óptima para la tarea específica de clasificación binaria. El preprocesamiento es una fase esencial para preparar los datos antes de introducirlos en los modelos de aprendizaje profundo.

**Normalización**: En esta etapa, se ha realizado una normalización de píxeles en las imágenes. Este proceso no solo ajusta los valores de intensidad de los píxeles para que estén en una escala uniforme entre 0 y 1, sino que también incluye la manipulación de media y varianza para garantizar una distribución consistente de los datos. La normalización es esencial para que los modelos de aprendizaje profundo puedan procesar las imágenes de manera eficiente y consistente, evitando discrepancias en las magnitudes de los valores de píxeles.

**Ajuste de Etiquetas:** Dado que el objetivo es realizar una clasificación binaria entre las clases "Automobile" y "Truck", se llevó a cabo un ajuste en las etiquetas originales. Las clases se han transformado de manera que "Automobile" sea representada como 0 y "Truck" como 1. Esta adaptación simplifica la tarea de clasificación al reducirla a un problema binario.

**Aumento de Datos:** Para mejorar la capacidad del modelo para generalizar a nuevas instancias, se ha implementado el aumento de datos. Este paso implica la generación de nuevas muestras de entrenamiento a partir de las muestras existentes mediante transformaciones como rotaciones, inversiones, recortes, etc. El aumento de datos contribuye a diversificar el conjunto de entrenamiento y mejora la robustez del modelo.

**Conversión de Etiquetas a Categóricas:** Las etiquetas ajustadas se han convertido en representaciones categóricas. Esta conversión es necesaria para que los modelos puedan comprender y manejar las clases de manera efectiva durante el entrenamiento. Se utilizó una técnica de codificación categórica que asigna un vector binario único a cada clase, permitiendo así que los modelos realicen una clasificación precisa.

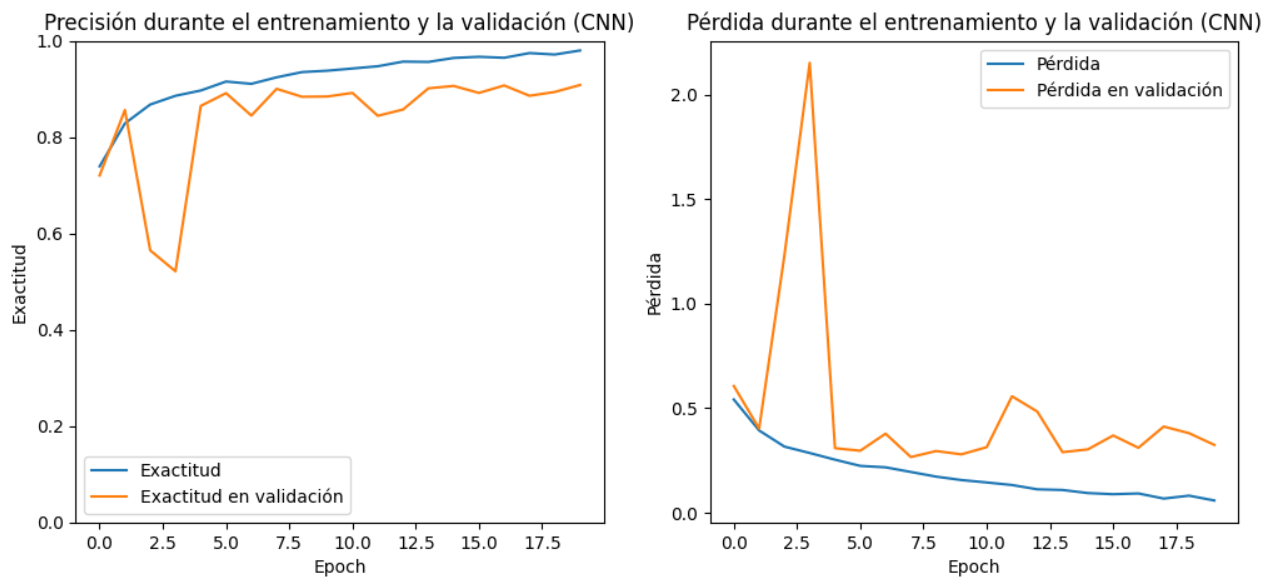
**Selección de Modelos**

* **Convolutional Neural Network (CNN) y Capas de Convolución:**

La Convolutional Neural Network (CNN) es una arquitectura fundamental en visión por computadora y desempeña un papel crucial en la clasificación de imágenes, como en el caso de clasificación de razas caninas. Entender el funcionamiento de las capas de convolución es esencial para apreciar cómo el modelo extrae y procesa características distintivas de las imágenes.

* **Operación de Convolución:** La convolución se realiza mediante el uso de un filtro (kernel) que se aplica a la imagen para extraer características específicas. Por ejemplo, un filtro diseñado para detectar bordes revelará una representación enfocada en esos bordes. Esta operación se visualiza como la aplicación de un patrón a una imagen para identificar características particulares.
* **Capas de Convolución:** Operando iterativamente sobre la imagen, las capas de convolución utilizan un filtro que se desplaza, realizando operaciones en cada posición y generando un mapa de características destacadas. Durante la iteración, el filtro multiplica punto a punto los coeficientes por la porción de la imagen, generando un píxel en la imagen de salida. Este proceso se repite hasta que el filtro ha barrido toda la imagen, y la imagen resultante se pasa a la siguiente capa, como la capa de reducción.

| **Arquitectura del Modelo CNN**  **Model: "sequentiaI\_5"** |  |  |
| --- | --- | --- |
| **Layer(type)** | **Output Shape** | **Param #** |
| **Conv2D** | **(None, 30, 30, 32)** | **896** |
| **MaxPooling2D** | **(None, 15, 15, 32)** | **0** |
| **Conv2D** | **(None, 13, 13, 64)** | **18496** |
| **MaxPooling2D** | **(None, 6, 6, 64)** | **0** |
| **Conv2D** | **(None, 4, 4, 128)** | **73856** |
| **Flatten** | **(None, 2048)** | **0** |
| **Dense** | **(None, 512)** | **1049088** |
| **BatchNormalization** | **(None, 512)** | **2048** |
| **Dropout** | **(None, 512)** | **0** |
| **Dense** | **(None, 2)** | **1026** |
| **Total params:1145410**  **Trainable params: 1144386**  **Non-trainable params: 1024** |  |  |



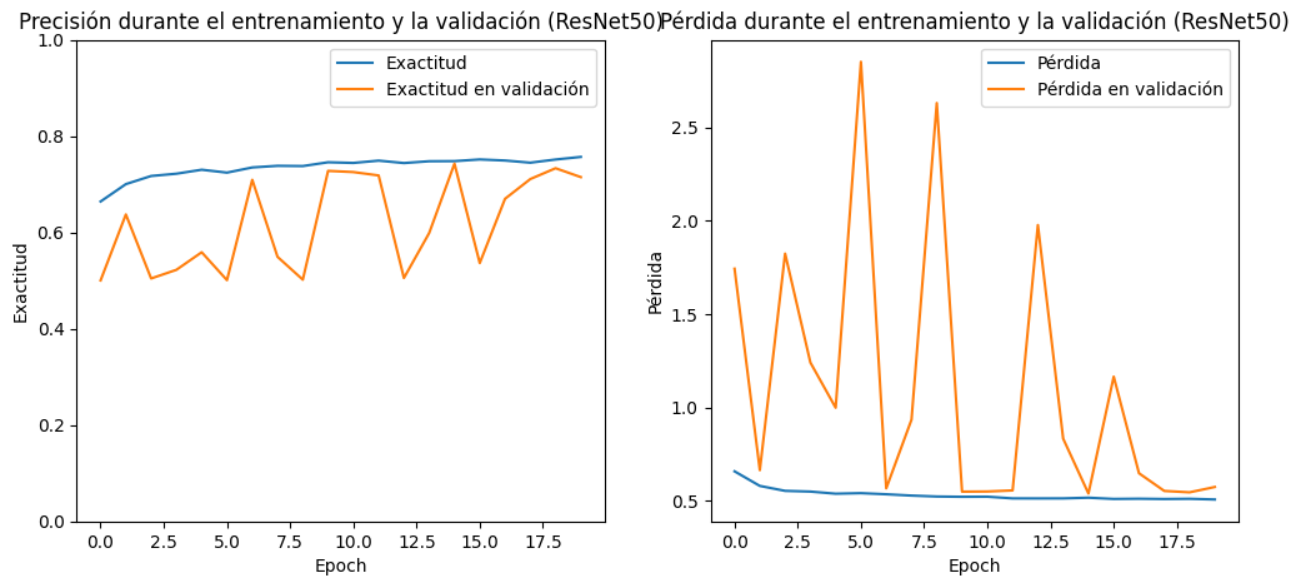
**ResNet50:**

ResNet50 es una arquitectura de red neuronal profunda desarrollada por Microsoft Research. La "Res" indica "residual," y el "50" refiere al número de capas. Destaca por su capacidad para entrenar redes neuronales profundas, gracias a la introducción de bloques residuales. Estos bloques tienen conexiones de atajo que permiten que la información original fluya sin cambios, facilitando el entrenamiento de redes más profundas.

**Características de ResNet50:**

* Bloques residuales para mitigar la degradación del rendimiento al aumentar la profundidad.
* El número "50" se refiere a la cantidad total de capas, incluyendo convolucionales, de agrupación y completamente conectadas.
* Existen versiones más profundas como ResNet101 y ResNet152.

| **Arquitectura del Modelo ResNet50**  **Model: "sequentiaI\_3"** |  |  |
| --- | --- | --- |
| **Layer(type)** | **Output Shape** | **Param #** |
|  |  |  |
| **resnet50 (Functional)** | **(None, 1, 1, 2048)** | **23587712** |
| **GlobalAveragePooling2D** | **(None, 2048)** | **0** |
| **Dense** | **(None, 512)** | **1049088** |
| **BatchNorma1ization** | **(None, 512)** | **2048** |
| **Dropout)** | **(None, 512)** | **0** |
| **Dense** | **(None, 2)** | **1026** |
| **Total params:24639874**  **Trainable params: 1051138**  **Non-trainable params: 23588736** |  |  |

****

**VGG16:**

VGG16, desarrollada por el Visual Graphics Group de la Universidad de Oxford, es conocida por su simplicidad y consistencia en el diseño de capas. Utiliza convoluciones de 3x3 y agrupación máxima en todas las capas convolucionales.

**Características de VGG16:**

* Convoluciones 3x3 con activación ReLU en todas las capas convolucionales.
* Agrupación máxima (max pooling) para reducir la dimensionalidad.
* Capas densas completamente conectadas en la parte convolucional.
* Comúnmente utilizado pre entrenado en ImageNet.

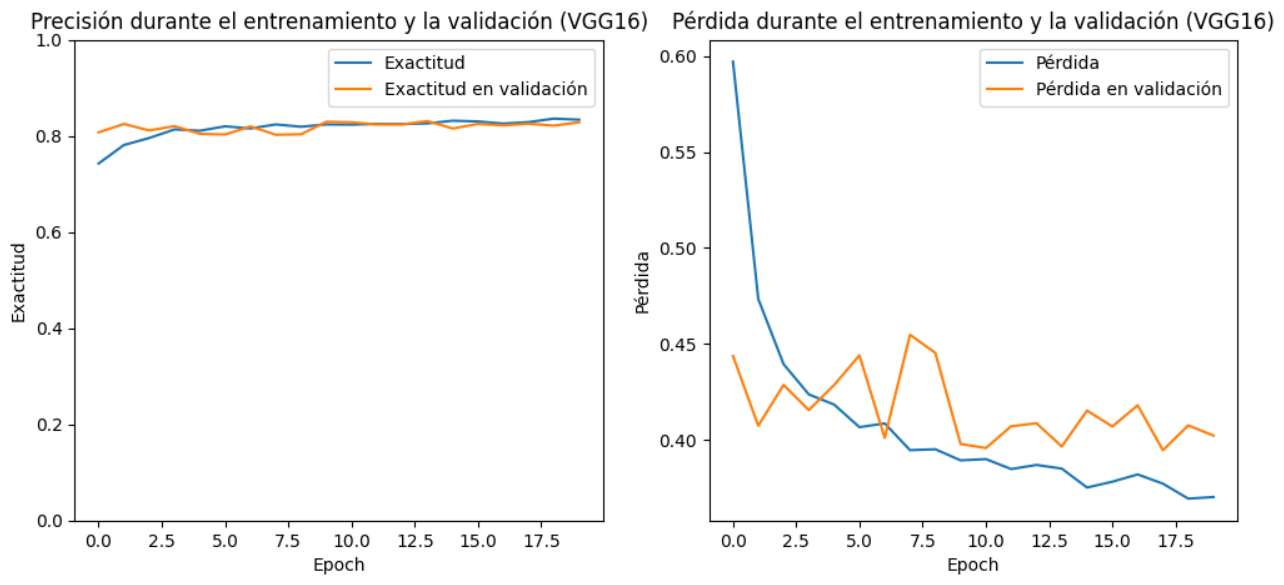
**Uso en el Código (Común para Ambos):** En el código proporcionado, se emplea la arquitectura correspondiente pre entrenada en ImageNet. Se ajustan con capas adicionales para adaptarlas al problema específico de clasificación binaria de vehículos.

**Ventajas y Consideraciones (Común para Ambos):** Aunque ResNet50 y VGG16 difieren en complejidad, ambos son efectivos en tareas de clasificación. ResNet50 aborda el desafío de profundidad, mientras que VGG16 destaca por su estructura simple.

**Aplicación en el Problema:** La elección de ResNet50 o VGG16 sugiere la búsqueda de arquitecturas robustas para clasificación binaria de vehículos. Ambas son opciones probadas y adaptadas al análisis de imágenes proporcionadas.

**Arquitectura del Modelo VGG16**

| **Arquitectura del Modelo VGG16**  **Model: "sequentiaI\_4"** |  |  |
| --- | --- | --- |
| **Layer(type)** | **Output Shape** | **Param #** |
| **Functional** | **(None, 1, 1, 512)** | **14714688** |
| **Flatten** | **(None, 512)** | **0** |
| **Dense** | **(None, 512)** | **262656** |
| **BatchNorma1ization** | **(None, 512)** | **2048** |
| **Dropout)** | **(None, 512)** | **0** |
| **Dense** | **(None, 2)** | **1026** |
| **Total params: 14980418**  **Trainable params: 264706**  **Non-trainable params: 14715712** |  |  |

****

**Evaluación**

**Modelo ResNet50**

El modelo ResNet50 fue entrenado durante 20 épocas, registrando una mejora significativa en la precisión y la pérdida durante este periodo.Durante el entrenamiento, la precisión en el conjunto de entrenamiento alcanzó aproximadamente un 75%, mientras que en el conjunto de validación se mantuvo alrededor del 50%. Este comportamiento sugiere un posible sobreajuste del modelo a los datos de entrenamiento.

Aquí se presenta un resumen del entrenamiento

**Epoch 1/20**

**250/250 [==============================] - 35s 125ms/step - loss: 0.6581 - accuracy: 0.6644 - val\_loss: 1.7428 - val\_accuracy: 0.5005**

**Epoch 20/20**

**250/250 [==============================] - 27s 108ms/step - loss: 0.5074 - accuracy: 0.7570 - val\_loss: 0.5743 - val\_accuracy: 0.7150**

**63/63 [==============================] - 5s 76ms/step - loss: 0.5426 - accuracy: 0.7375**

**Accuracy en el conjunto de prueba (ResNet50): 0.737500011920929**

| **Classification Report (ResNet50):** |
| --- |

|  | **precision** | **recall** | **f1-score** | **support** |
| --- | --- | --- | --- | --- |
| **Automobile** | **0.83** | **0.60** | **0.70** | **1000** |
| **Truck** | **0.69** | **0.87** | **0.77** | **1000** |
| **accuracy** |  |  | **0.74** | **2000** |
| **macro avg** | **0.76** | **0.74** | **0.73** | **2000** |
| **weighted avg** | **0.76** | **0.74** | **0.73** | **2000** |

El modelo ResNet50 demuestra un desempeño notable en la tarea de clasificación de vehículos, logrando una precisión general del 74%. Al analizar las métricas para cada clase:

* **Automobile (Coche):**
  + **Precisión:** 83%
  + **Recall:** 60%
  + **F1-Score:** 70%
* El modelo muestra una precisión sólida al identificar automóviles, aunque con cierta dificultad en la recuperación de instancias específicas de esta clase.
* **Truck (Camión):**
  + **Precisión:** 69%
  + **Recall:** 87%
  + **F1-Score:** 77%
* El modelo destaca en la identificación de camiones, exhibiendo una buena capacidad para reconocer esta clase con una alta tasa de recall y una precisión aceptable.

La evaluación global, considerando el promedio ponderado de las métricas (weighted avg), refleja un equilibrio satisfactorio entre ambas clases, indicando que el modelo ResNet50 tiene la capacidad de generalizar efectivamente en situaciones diversas. La precisión del 74% en el conjunto de prueba respalda la eficacia del modelo en la clasificación de vehículos.



Ejemplos de predicción del modelo ResNet50

**Modelo VGG16**

El modelo VGG16 fue entrenado durante 20 épocas y evaluado en el conjunto de pruebas. Durante el entrenamiento, la precisión en el conjunto de entrenamiento y validación mostró una tendencia positiva, alcanzando aproximadamente un 82% en ambos conjuntos al finalizar el entrenamiento, en la evaluación con el conjunto de prueba, el modelo VGG16 demostró una precisión del 81.75%.

Aquí se presenta un resumen del entrenamiento:

**Epoch 1/20**

**250/250 [==============================] - 34s 134ms/step - loss: 0.5970 - accuracy: 0.7427 - val\_loss: 0.4438 - val\_accuracy: 0.8075**

**Epoch 20/20**

**250/250 [==============================] - 57s 227ms/step - loss: 0.3704 - accuracy: 0.8339 - val\_loss: 0.4024 - val\_accuracy: 0.8285**

**63/63 [==============================] - 12s 183ms/step - loss: 0.4088 - accuracy: 0.8175**

**Accuracy en el conjunto de prueba (VGG16): 0.8174999952316284**

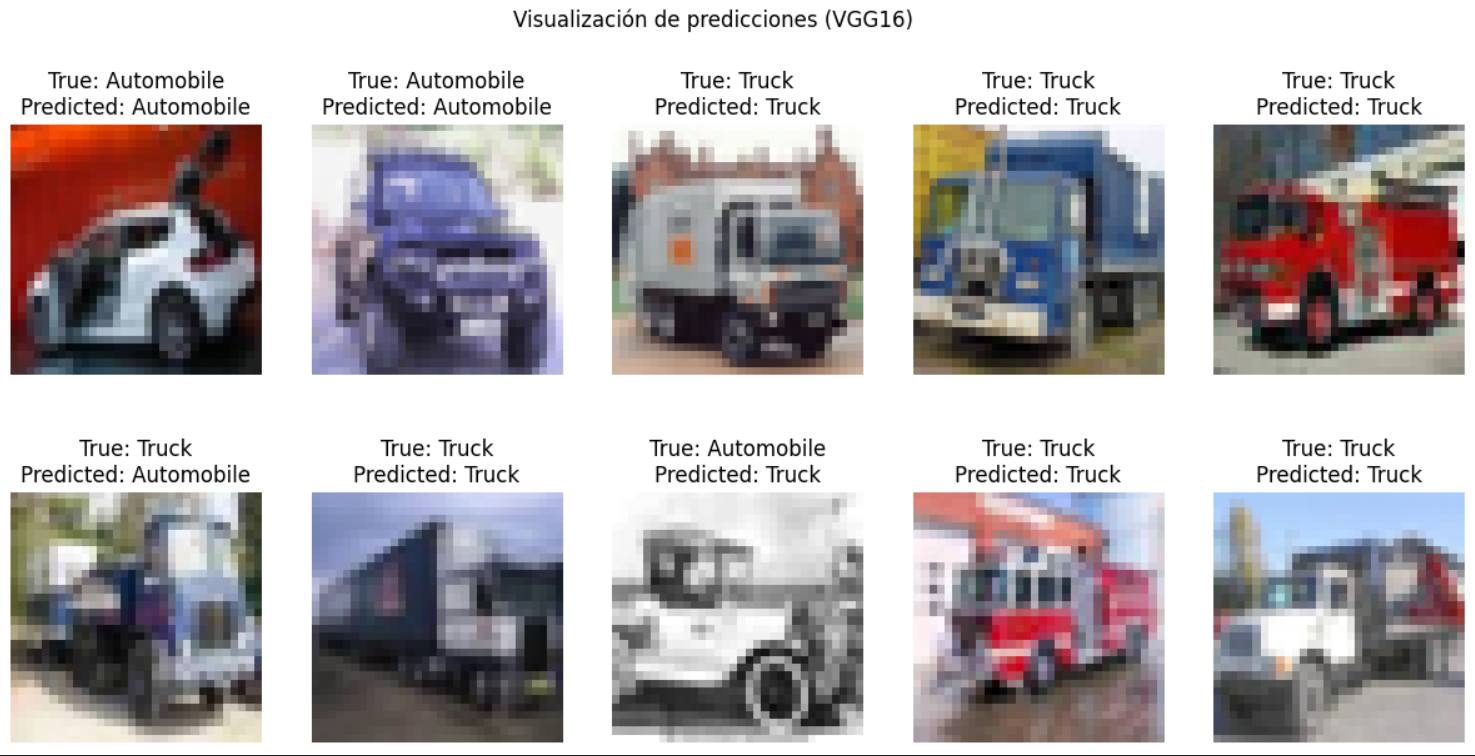
| **Classification Report (VGG16):** |
| --- |

|  | **precision** | **recall** | **f1-score** | **support** |
| --- | --- | --- | --- | --- |
| **Automobile** | **0.84** | **0.78** | **0.81** | **1000** |
| **Truck** | **0.80** | **0.85** | **0.82** | **1000** |
| **accuracy** |  |  | **0.82** | **2000** |
| **macro avg** | **0.82** | **0.82** | **0.82** | **2000** |
| **weighted avg** | **0.82** | **0.82** | **0.82** | **2000** |

El modelo VGG16 exhibe un rendimiento sólido en la clasificación de vehículos, alcanzando una precisión general del 82%. Desglosando las métricas por clase:

* **Automobile (Coche):**
  + **Precisión**: 84%
  + **Recall**: 78%
  + **F1-Score**: 81%
* El modelo demuestra una alta precisión y recall para la clase de automóviles, indicando una capacidad robusta para reconocer esta categoría de vehículos.
* **Truck (Camión):**
  + **Precisión**: 80%
  + **Recall:** 85%
  + **F1-Score**: 82%
* Similar al caso de automóviles, el modelo presenta una combinación sólida de precisión y recall para la clase de camiones.

La evaluación global, representada por las métricas de promedio ponderado (weighted avg), revela un equilibrio efectivo entre ambas clases. La precisión general del 82% en el conjunto de prueba sugiere que el modelo VGG16 es altamente competente en la tarea de clasificación de vehículos.



Ejemplos de predicción del modelo VGG16

**Modelo CNN**

El modelo CNN personalizado se entrenó durante 20 épocas y se evaluó en el conjunto de prueba. Durante el entrenamiento, la precisión en el conjunto de entrenamiento y validación mejoró constantemente, alcanzando alrededor del 90% en ambos conjuntos al finalizar el entrenamiento.

Aquí se presenta un resumen del entrenamiento:

**Epoch 1/20**

**250/250 [==============================] - 9s 30ms/step - loss: 0.5417 - accuracy: 0.7395 - val\_loss: 0.6063 - val\_accuracy: 0.7205**

**Epoch 20/20**

**250/250 [==============================] - 11s 45ms/step - loss: 0.0588 - accuracy: 0.9801 - val\_loss: 0.3247 - val\_accuracy: 0.9085**

**63/63 [==============================] - 1s 10ms/step - loss: 0.3702 - accuracy: 0.8960**

**Accuracy en el conjunto de prueba (CNN): 0.8960000276565552**

| **Classification Report (CNN):** |
| --- |

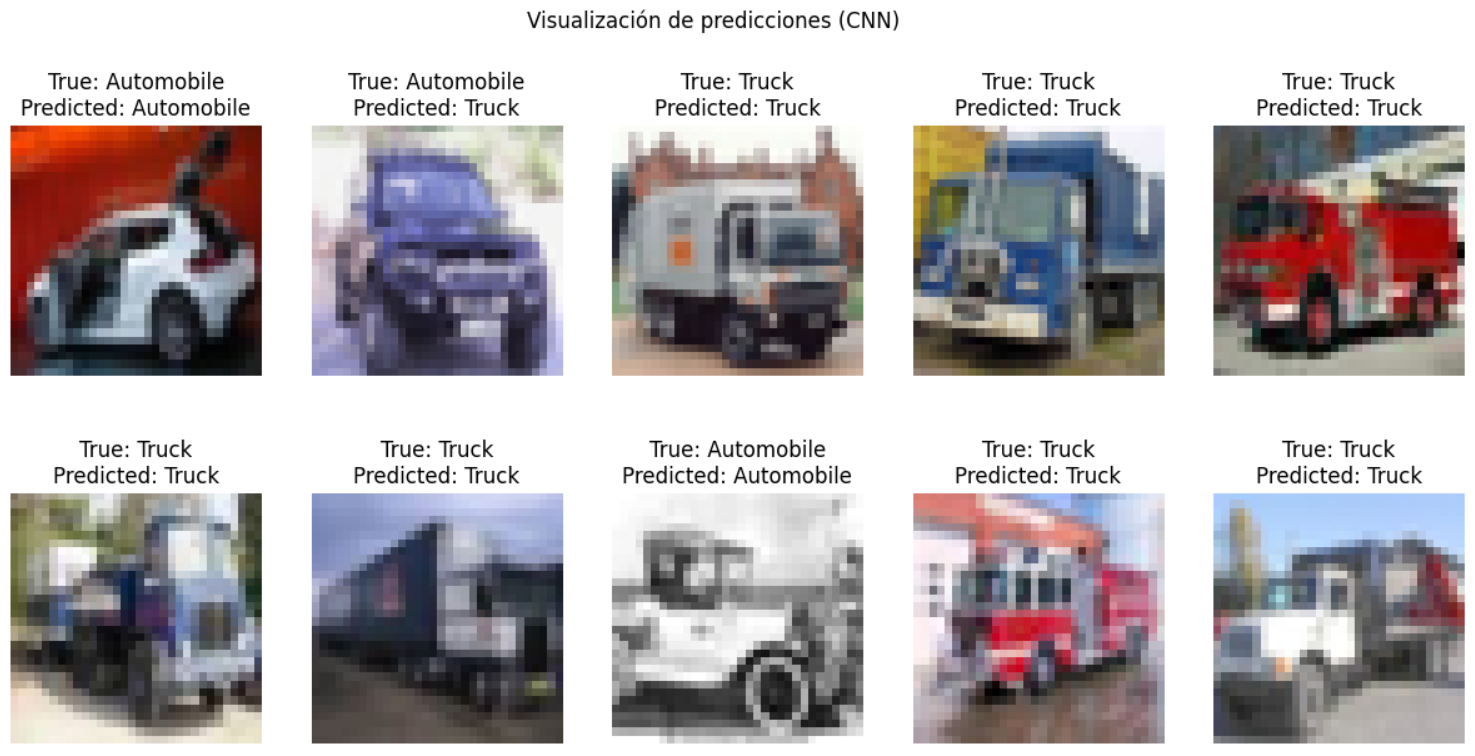
|  | **precision** | **recall** | **f1-score** | **support** |
| --- | --- | --- | --- | --- |
| **Automobile** | **0.92** | **0.87** | **0.89** | **1000** |
| **Truck** | **0.88** | **0.92** | **0.90** | **1000** |
| **accuracy** |  |  | **0.90** | **2000** |
| **macro avg** | **0.90** | **0.90** | **0.90** | **2000** |
| **weighted avg** | **0.90** | **0.90** | **0.90** | **2000** |

El modelo de Red Neuronal Convolucional (CNN) ha demostrado un rendimiento excepcional en la clasificación de vehículos, logrando una asombrosa precisión global del 90%. Al profundizar en las métricas específicas para cada clase, se presentan los siguientes resultados:

* **Automobile (Coche):**
  + **Precisión**: 92%
  + **Recall:** 87%
  + **F1-Score:** 89%
* El modelo ha demostrado una precisión excepcional en la identificación de automóviles, con una combinación sólida de recall y F1-Score.
* **Truck (Camión):**
  + **Precisión:** 88%
  + **Recall:** 92%
  + **F1-Score:** 90%
* La capacidad del modelo para clasificar camiones también es notable, destacando un alto nivel de precisión y recall.

La evaluación global, representada por las métricas de promedio ponderado (weighted avg), subraya el rendimiento equilibrado del modelo en ambas clases. Con una precisión global del 90% en el conjunto de prueba, el modelo CNN se destaca como una herramienta efectiva y confiable para la tarea de clasificación de vehículos, evidenciando una capacidad robusta para generalizar a través de diferentes

En este apartado se representa la visualización de las predicciones de cada modelo seleccionado, donde la línea *True* especifica la clase de imagen (**Automobile o Truck**), mientras que *Predicted* es la clase que el modelo identifica tras haber hecho su respectivo análisis, donde esta puede acertar o equivocarse en mayor o menor frecuencia dependiendo del modelo.



Ejemplos de predicción del modelo CNN

# 

# Conclusiones

El presente trabajo abordó la problemática del control de acceso de autos y camiones mediante la aplicación de técnicas de aprendizaje automático, específicamente en el procesamiento de imágenes. Se exploraron y compararon tres modelos diferentes: ResNet50, VGG16 y una Red Neuronal Convolucional (CNN) personalizada.

En el análisis exploratorio de datos, se utilizó el conjunto de datos CIFAR-10, enfocándonos en las clases "Automobile" y "Truck". Se llevaron a cabo pasos de preprocesamiento, normalización y ajuste de etiquetas para preparar los datos para el entrenamiento de los modelos.

Los modelos ResNet50 y VGG16, pre entrenados en ImageNet, fueron adaptados para la tarea de clasificación binaria de vehículos. Ambos modelos demostraron un rendimiento sólido, con una precisión global del 74% para ResNet50 y 82% para VGG16 en el conjunto de prueba. Además, se implementó una Red Neuronal Convolucional (CNN) personalizada, la cual superó a los modelos pre entrenados con una precisión asombrosa del 90%. Este resultado destaca la capacidad de las redes personalizadas para adaptarse de manera más específica a la tarea en cuestión.

En cuanto a los requerimientos de software y hardware, se destacó la importancia de herramientas especializadas como TensorFlow para el procesamiento de imágenes y entornos de desarrollo como Jupyter Notebook y Visual Studio Code. La utilización de GPUs y CPUs potentes fue esencial para la eficiencia en el entrenamiento de los modelos.

El desarrollo del proyecto tiene un potencial considerable para su aplicación en una variedad de entornos, desde instalaciones industriales hasta complejos residenciales, donde la mejora de la seguridad y la optimización de los procesos son prioridad. La capacidad del sistema para adaptarse a condiciones cambiantes y resistir vulnerabilidades lo posiciona como una perfecta herramienta para el control de acceso.