

Universidad de Lima
Facultad de Ingeniería y Arquitectura
Carrera de Ingeniería de Sistemas



ESTUDIO SOBRE LA CANTIDAD MÍNIMA DE MUESTRAS DE ENTRENAMIENTO PARA LA CLASIFICACIÓN DE MODELOS VEHICULARES

Tesis para optar el grado académico de título en Ingeniería de Sistemas

Carlos Enrique Castañeda Haro

Código 20161905

Asesor

Victor Hugo Ayma Quirita

Lima – Perú

Octubre de 2022

Estudio sobre la cantidad mínima de muestras para la clasificación de modelos vehiculares

Castañeda Haro, Carlos Enrique

20161905@aloe.ulima.edu.pe

Universidad de Lima

Resumen: La clasificación de objetos es uno de los campos de estudios más importantes de los últimos años y está asociado a la similitud de características entre los objetos y al continuo crecimiento de los conjuntos de datos de entrenamiento. En base a ello, aumentar el número de muestras de entrenamiento mejora el rendimiento de los clasificadores. Sin embargo, no hay estudios que determinen un estimado de cuántas muestras de entrenamiento son necesarias para generar clasificadores robustos. En esta investigación se intenta responder esta pregunta, enfocando el problema en la clasificación por marca y modelo vehicular. Para ello, se creó un conjunto de datos compuesto por 32 modelos vehiculares diferentes y se utilizó la red VGG16 para la tarea de extracción de características. Asimismo, se utilizaron los algoritmos de clasificación Máquinas de Vector Soporte (SVM), Bosques Aleatorios (RF), Árboles de Decisión (DT) y Naive Bayes (NB). Se realizaron conjunto de entrenamientos en los que se variaron el número de muestras de entrenamiento y el número de categorías a clasificar por cada algoritmo. En estos experimentos, el algoritmo SVM fue el de mayor precisión con un 96.82% para el caso de 32 modelos vehiculares diferentes. Finalmente, se determinó que a medida que se aumenta el número de modelos vehiculares a clasificar, es necesario aumentar las muestras de entrenamiento para estabilizar la precisión, y que el número mínimo de muestras para este comportamiento es de 400 muestras para el escenario de 2 categorías y de 700 muestras para el resto de los escenarios con más categorías.

Palabras Clave: Clasificación, Visión Computacional, Reconocimiento de Marca y Modelo Vehicular

Abstract: Object classification is one of the most important fields of study in recent times and it is associated with the similarity between objects and the continuous growth of training data sets. Based on this, increasing the number of training samples improves the performance of the classifiers. However, there are no studies that determine an estimate of how many training samples are necessary to develop solid classifiers. This research attempts to answer this question, focusing the problem on vehicle make and model recognition (VMMR). To do this, a data set composed of 32 different vehicle models was created and the VGG16 network was used for the feature extraction task. Likewise, the Support Vector Machine (SVM), Random Forest (RF), Decision Trees (DT) and Naive Bayes (NB) classification algorithms were used. A set of experiments were carried out in which the number of training samples and the number of categories to be classified by each algorithm were varied. In these experiments, the SVM algorithm was the most accurate with 96.82% for the case of 32 different vehicle models. Finally, it was determined that as the number of vehicle models to be classified is increased, it is necessary to increase the training samples, to stabilize the precision, and that the minimum number of training samples for this stabilization is 400 samples for the scenario of 2 categories and 700 samples for the rest of the scenarios with more categories.

Keywords: Classification, Computer Vision, Vehicles, Vehicle Make Model Recognition (VMMR)

1. INTRODUCCIÓN

La clasificación de objetos ha sido uno de los más importantes campos de estudio dentro del reconocimiento de patrones, la inteligencia artificial, la visión por computador y el análisis de imágenes (Kaur, N. y Kaur, Y., 2014). Usualmente, el proceso de clasificación consiste en asociar una etiqueta específica (conocida como clase) a los objetos que están siendo analizados. Sin embargo, el éxito de esta tarea se ve limitado principalmente por dos factores: la similitud de apariencia entre diferentes objetos y la disponibilidad y el uso de un número adecuado de muestras de entrenamiento para la creación de modelos de clasificación robustos (Dong et al, 2018).

Diversos estudios demuestran que el incremento de muestras es una técnica eficiente para sobreponerse a los inconvenientes generados por la similitud de apariencias entre objetos. Por ejemplo, Lehr (2019) exhibió un incremento del 40% en el rendimiento de la red convolucional de arquitectura DenseNet al incrementar muestras y combinarlas con una imagen de referencia para categorizar tornillos de la misma norma pero con diferentes tamaños. Asimismo, Hu et al. (2019) propuso un método de aumento de datos que previamente identifica la región de interés para obtener la mayor cantidad de información tras cada transformación. Dicho estudio desarrolló un comparativo sobre 4 conjuntos de datos públicos de imágenes correspondientes a aves, aviones, vehículos y perros que exhiben una fuerte similitud de apariencia intra clase. Para todos sus experimentos en los 4 conjuntos de datos, obtuvo una mejora de alrededor del 10% en comparación a una clasificación sin aumento de datos empleando una VGG 19 como

clasificador de referencia. Sin embargo, a pesar de los beneficios que ofrece el incremento de muestras para la generación de clasificadores más robustos, en la literatura existen muy pocos trabajos orientados a determinar la cantidad mínima de muestras para generar clasificadores con buenos rendimientos.

Así, el objetivo de este trabajo se enfoca en determinar la cantidad mínima de muestras para generar clasificadores robustos en escenarios donde los objetos muestren gran similitud en apariencia. Nuestro estudio se contextualiza en la clasificación de modelos vehiculares debido a sus múltiples aplicaciones en temas de transporte inteligente (Gholamhosseinian y Seitz, 2021), detección de vehículos robados (Kadambari y Nimmalapudi, 2020), gestión de tráfico vehicular (Ozkurt y Camci, 2009), entre otros. La metodología empleada comprende el incremento sistemático de muestras de imágenes de vehículos en función de los diferentes tipos de modelos vehiculares.

El resto de este trabajo se organiza de la siguiente manera, la sección 2 presenta una revisión de la literatura asociada a diferentes algoritmos de clasificación y su rendimiento al operar con distintos conjuntos de datos. La sección 3 introduce los fundamentos teóricos relacionados al análisis de imágenes y las diferentes etapas que este comprende. La sección 4 explica la metodología seguida para la realización de nuestros experimentos, así como las configuraciones utilizadas en los clasificadores y la creación del conjunto de datos a utilizar. La sección 5 informa los resultados obtenidos tras la experimentación y los compara contra los observados en la literatura existente. La sección 6, presenta las conclusiones encontradas tras los resultados obtenidos y como la variabilidad en el conjunto de datos, así como el número de categorías a identificar afectan al rendimiento de los clasificadores. Finalmente, el documento concluye en la sección 7 con propuestas a futuro que enriquezcan la investigación.

2. ESTADO DEL ARTE

En la investigación realizada por Huang et al. (2008), se propone un análisis en base a un discriminante lineal de dos dimensiones (Two- Dimensional Linear Discriminant Analysis 2DLDA) como extractor de características y al algoritmo de vecino más cercano (KNN) para clasificar los vehículos según marca y modelo. Se contó con un conjunto de datos de 1414 imágenes normalizadas en un formato de 100 x 50 píxeles, distribuidas 20 modelos vehiculares diferentes. La prueba de esta combinación de algoritmos nos mostró un 94.7% de efectividad para su mejor prueba en la precisión de reconocimiento a partir de la región de interés del vehículo, la cual se ubicó en la parte frontal de este. Además, se identificó que este método no era sensible a variaciones de color ni de iluminación.

Asimismo, Psyllos et al. (2009) proponen al descriptor SIFT para realizar el reconocimiento y clasificación del modelo vehicular. Psyllos separó el problema en 2 de modo que primero identifica la marca del vehículo a partir de la imagen del logo y posteriormente el modelo vehicular a partir de la máscara frontal del vehículo. Para esto se contó con la base de datos de Medialab LPR Database que corresponde a 10 clases de marcas y modelos vehiculares diferentes con 40 imágenes frontales del vehículo ajustadas a un formato de 1024 x 768 píxeles por cada clase. De manera independiente se obtuvo un 94 % de precisión para el reconocimiento de la marca vehicular y un 92% de precisión del modelo vehicular, de modo que en conjunto se obtuvo una tasa de reconocimiento del 87%. Se llegó a concluir que, si bien se obtuvo una tasa de precisión satisfactoria, se debería extender a otras vistas del vehículo para mejorar la robustez del modelo.

A su vez, Bu et al. (2017) desarrollan un modelo de reconocimiento de marca y modelo vehicular en base a una red neuronal CNN (Convolutional Neural Network). Se planteó la inclusión de una capa extra intermedia de convolución en la estructura Caffenet, para obtener una mejor determinación sobre las características intermedias de los vehículos. El conjunto de datos que se construyó tenía una distribución de 100 imágenes por cada una de las 19 clases de modelos vehiculares que consideraron. Asimismo, se consideró la inclusión de imágenes frontales y vistas laterales para vehículos en su mayoría del año 2014. Se realizó el entrenamiento sobre la estructura Caffenet original con el conjunto de datos creado y se logró una precisión del 73%. Para optimizar el entrenamiento de la red, se aplicaron técnicas de preprocesamiento sobre las imágenes del conjunto de datos para eliminar ciertos “ruidos” o “perturbaciones” y afinar la imagen, a través de filtros de Gauss y transformaciones de Fourier. Finalmente, se realizó el entrenamiento con el conjunto de datos pre procesado sobre la estructura Caffenet original y la Caffenet mejorada, con lo que se obtuvo precisiones del 93.65% y 94.75% respectivamente. En razón a ello, se concluyó que en efecto el preprocesamiento de las imágenes ayuda a mejorar la efectividad en el entrenamiento de las redes neuronales y que la inclusión de una capa intermedia provee ligera mejora en la clasificación de características del vehículo.

Por otro lado, Pearce y Pears (2011) realizaron una comparación entre 10 modelos diferentes dados de la combinación del uso de los extractores de características Canny Edges, Square Mapped Gradients, Square Mapped Gradients con preprocesamiento, Harris Corners y Local Normalised Harris Strengths con los algoritmos de clasificación K-Nearest Neighbour (Vecino más cercano) y Naive Bayes. Para la experimentación se utilizó un conjunto de datos que contenía un total de 262 imágenes repartidas en 22 clases de modelos vehiculares, siendo la clase 22 un conglomerado de diferentes vehículos no tan comunes ya que no se poseía muchas imágenes de pruebas de estos como clases aisladas, bajo el esquema de leave one out para realizar la prueba sobre el modelo. Como mejor

resultado para el clasificador KNN se obtuvo un 94.9%, mientras que para el clasificador Naive Bayes se obtuvo un 96% de éxito de tasa de reconocimiento. Finalmente, se concluyó que el clasificador de Naive Bayes era ligeramente superior al algoritmo KNN, y que, para el caso de clases no tan comunes o aisladas entre sí, puede ser recomendable agregarlas en una superclase que defina a estas categorías.

Por su parte, Shijie et al. (2017) utilizaron la base de la arquitectura de red convolucional VGG16 como extractor de características y el algoritmo SVM como clasificador para determinar la detección de enfermedades o pestes en la planta del tomate basados en imágenes de sus hojas. Para ello, se contó con un conjunto de datos con 400 imágenes de entrenamiento por cada una de las 10 pestes a clasificar y se logró una precisión promedio del 89%, con lo que se demostró que la red VGG16 es una alternativa como extractor de características.

Finalmente, Manzoor et al. (2019) utilizaron los descriptores de GiST (Generalized Search Tree) y HOG (Histogram of Oriented Gradients) para realizar la extracción de características de las imágenes y los algoritmos de machine learning SVM (Support Vector Machine) y los RF (Random Forest) para realizar una clasificación basada en la marca y modelo vehicular, de modo que se obtuvo 4 modelos diferentes de la combinación de los métodos de extracción con los de clasificación. Para el desarrollo de los modelos utilizaron el conjunto de datos público NTOU-MMR (National Taiwan Ocean University – Make Model Recognition). Este conjunto de datos contiene únicamente imágenes frontales con variaciones de +/- 20 grados en diferentes condiciones de clima e iluminación y está compuesto por 35 clases de marcas y modelos vehiculares en 2725 imágenes de entrenamiento en total. Por un lado, se observó que tras realizar la extracción de características en base a HOG se obtuvo como mejor experimento para la clasificación con RF un 94.43% de precisión, mientras que para el uso con SVM se obtuvo un 97.89%. Por otro lado, cuando se utilizó GiST se informó una precisión de 94.53% por RF y del 97.2% por SVM, lo que demostró que la clasificación por SVM brindaba una mejor eficiencia independientemente del método de extracción de características utilizado.

En síntesis, según lo expuesto en esta sección se puede evidenciar que existen avances en el área del reconocimiento vehicular, y que estos se centran en los procesos de extracción de características y clasificación. Sin embargo, son muy pocos los estudios realizados con énfasis en la determinación de la cantidad mínima de muestras requeridas para el entrenamiento de modelos robustos, a pesar de que este factor es sumamente relevante cuando se analizan escenarios con poca variabilidad en la apariencia intra clase. A continuación, en la siguiente sección se presentan los antecedentes y fundamentos teóricos asociados.

3. ANTECEDENTES

El procesamiento de imágenes consiste principalmente en la extracción de información a partir de objetos o regiones de interés de una imagen a fin de identificar propiedades distintivas para su posterior clasificación (Mansilla, 2015). Este proceso tiene como principales objetivos la corrección y eliminación de ruidos y distorsiones para la mejora de la imagen con el fin de facilitar las tareas de clasificación y análisis de imágenes (Shammi et al., 2014). Asimismo, la visión computacional define 5 etapas que en la mayoría de los casos se utilizan para tratar con los problemas relacionados al procesamiento de imágenes. Estas etapas son secuenciales y son la adquisición de imágenes, el pre procesamiento de imágenes, la segmentación de imágenes, la extracción de características y la clasificación, tal como muestra la Figura 3.1.

Figura 3.1

Diagrama de las etapas del procesamiento de imágenes por visión computacional



Nota. Adaptada de “Basics of Image Analysis”. Por F. Mendoza y R. Lu, 2015, *Springer*, p. 23 (https://doi.org/10.1007/978-1-4939-2836-1_2)

A continuación, se ahondará en cada una de las etapas mencionadas previamente, así como sus definiciones y algunos de los distintos métodos que estas comprenden.

3.1 Adquisición de imágenes

Esta etapa es el primer paso en el procesamiento de imágenes y consiste, tal como lo señala su nombre, en adquirir imágenes en formato digital para su posterior pre procesamiento. Estas imágenes son obtenidas principalmente a través de cámaras digitales, videocámaras, etc. y buscan representar una escena actual en formato digital. Asimismo, la adquisición de imágenes, en múltiples ocasiones se apoya en recuperar imágenes de una fuente o base de datos externa para su posterior procesamiento (Ong et al., 2020).

3.2 Pre procesamiento de imágenes

Se entiende por pre procesamiento a los procesos realizados para mejorar una imagen. Este procesamiento está comprendido por diversas técnicas para lograr que una imagen se visualice mejor en algunos sentidos o características específicas (Gonzales y Woods, 1996). En este sentido, se debe entender de que el pre procesamiento se muestra como una mejora netamente subjetiva de la persona que la lleva a cabo, pues se buscan resaltar con este método las características que sean consideradas como más relevantes de la imagen según se requiera (Gonzales y Woods, 1996). Estas transformaciones homogéneas permiten que las coordenadas de los vectores de las imágenes sean afectadas, de modo que mantengan la longitud de dichos vectores. Asimismo, son ampliamente utilizadas en temas de visión computacional cuando se requiere que un sistema difícil de visualizar sea más accesible para el usuario (Cashbaugh y Kitts, 2017). Entre algunos ejemplos de estas transformaciones pertenecientes al pre procesamiento de imágenes se encuentran la traslación, la rotación, el escalamiento, el “flipping” o reflexión, etc.

Estas transformaciones se pueden explicar a través de la matriz de transformación, que es representada como una matriz de 3×3 . Esta matriz permite que los vectores $v(x, y)$ que contienen los píxeles originales de las imágenes, sean alterados y generen un nuevo vector de coordenadas $v(x', y')$ que posean las transformaciones deseadas y mantengan las propiedades de la imagen original. De esta manera, cada transformación posee su propia matriz de transformación. A continuación, se explicarán las principales transformaciones y sus matrices asociadas.

Por un lado, la traslación es una operación que consiste en mover un sistema de coordenadas a través de un desplazamiento específico a través de los ejes x y y . Su matriz de transformación está denotada a través de la ecuación 1.

$$M_T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ e & f & 1 \end{bmatrix} \quad (1)$$

Donde M_T representa a la matriz de transformación asociada a la traslación, y los elementos e y f indican la traslación horizontal y vertical a aplicar respectivamente.

Por otro lado, el escalamiento es una operación que consiste en cambiar las dimensiones del sistema coordenado de la imagen. Su matriz de transformación está denotada a través de la ecuación 2.

$$M_E = \begin{bmatrix} a & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Donde M_E representa a la matriz de transformación asociada al escalamiento, y los elementos a y d indican los coeficientes de escalamiento horizontal y vertical a aplicar respectivamente.

Asimismo, la reflexión es una operación que consiste en invertir horizontal y/o verticalmente las coordenadas del sistema asociado a la imagen. Su matriz de transformación está denotada a través de la ecuación 3.

$$M_{RF} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Donde M_{RF} representa a la matriz de transformación asociada a la reflexión. En esta matriz, los coeficientes -1 indican el tipo de reflexión a realizar de modo que el -1 asociado a los índices $(1,1)$ de la matriz indica un comportamiento horizontal y el -1 asociado a los índices $(2,2)$ de la matriz indica el comportamiento vertical de la transformación.

Finalmente, la rotación es una operación que consiste, tal como lo indica su nombre, en rotar el sistema coordenado de la imagen de manera horaria o antihoraria respecto a un ángulo dado. Su matriz de transformación está denotada a través de la ecuación 4.

$$M_{RT} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Donde M_{RT} representa a la matriz de transformación, y el elemento θ indica el ángulo de rotación asociado a la transformación.

3.3 Segmentación

Según Mendoza y Lu (2015), la segmentación de imágenes es una de las etapas más importantes en el proceso de análisis de imágenes, ya que los datos extraídos posteriormente, dependen en gran medida de la precisión en esta operación. El principal objetivo de esta operación es dividir una imagen en regiones que tienen una fuerte correlación en razón a una región de interés. Asimismo, la segmentación puede obtenerse a través de técnicas como la umbralización, la segmentación basada en bordes y la segmentación basada en regiones.

3.4 Extracción de Características

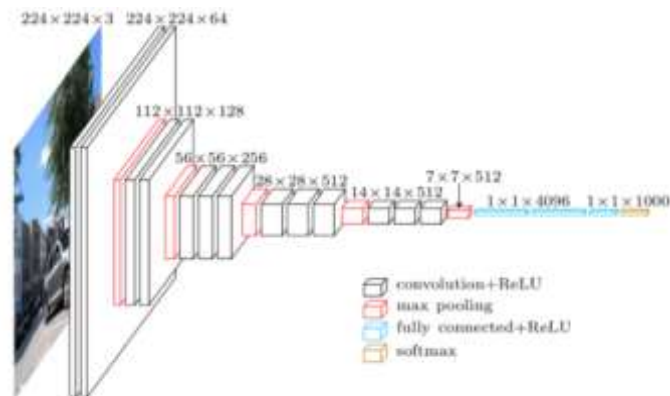
La etapa de extracción de características, tal como señala su nombre, consiste en la extracción de información discriminativa, de modo que se elimine la información redundante. Según Mansilla (2015), su propósito es reducir o limitar la dimensión del problema de reconocimiento, de modo que cuando el problema esté enfocado a las características de las formas de los objetos, se busque una representación externa asociada al contorno del objeto. Por otro lado, es recomendable buscar una representación interna, si se requiere información de las características de las regiones de interés, como el desvío de intensidad, descriptores de textura, etc.

Debido a su impacto en el proceso de clasificación, la extracción de características se ha tornado un campo de investigación dentro del procesamiento de imágenes y la visión por computador. Los métodos derivados de estas investigaciones se han aplicado en múltiples tareas como el reconocimiento de caracteres, verificación de documentos, lectura de direcciones postales, sistemas que requieren data manuscrita, etc. Para ello, se han desarrollado una amplia variedad de métodos de extracción de características como el “template matching”, la transformación de imágenes unitarias, los descriptores de fourier, etc. (Kumar G y Kumar P, 2015)

Por su parte, las redes neuronales en los últimos años se han mostrado también como una posible técnica para la extracción de características debido a su gran capacidad de representación y discriminación. Dentro de ellas, la Visual Geometry Group o VGG16 es una arquitectura de una red convolucional de 16 capas, tal como muestra la Figura 3.4.1, diseñada por Simonyan y Zisserman (2014) con la que lograron ganar la competencia de clasificación de imágenes ILSVRC (ImageNet Large Scale Visual Recognition Challenge) y que puede ser utilizada para el proceso de extracción de características.

Figura 3.4.1

Arquitectura por capas de la red VGG16



Nota. De “A review of deep learning in the study of materials degradation”. Por W. Nash, T. Drummond, N. Birbilis, 2018, *npj Materials Degradation* 2(37), p.3. (<https://doi.org/10.1038/s41529-018-0058-x>)

La Figura 3.4.1 nos muestra que, para el entrenamiento, las imágenes ingresadas deben estar ajustadas a un tamaño de 224 x 224 píxeles en formato RGB. Posteriormente, las imágenes son transformadas a un mapa de atributos a través de un conjunto de capas de convolución que aplican filtros de 3x3 píxeles, luego la VGG16 ajusta el paso de convolución a 1 píxel con el propósito de preservar la resolución después de las capas de convolución. Luego se lleva a cabo un proceso de “spatial pooling” por 5 capas de max pooling sobre una ventana de 2x2 píxeles. Tras las capas de max pooling, prosiguen 3 capas Fully connected, cuyas 2 primeras capas tienen 4096 dimensiones cada una y la tercera contiene una salida de 1000 dimensiones, que representa a cada una de las clases vistas en la base de datos de la competencia ILSVR. Bajo este esquema, las salidas previas a las capas Fully connected son tratadas como los descriptores o extractores de características, por lo que pueden ser usados para entrenar otros clasificadores (Shijie et al, 2017). Asimismo, el modelo proporcionado por la red VGG16 se ha convertido en uno muy reconocido por parte de la comunidad científica debido a que sus pesos pre entrenados se encuentran disponibles en línea, lo que facilita el

uso de este en tareas nuevas (Nash et al, 2018).

3.5 Clasificación

La clasificación de objetos en categorías o clases es en múltiples casos el objetivo final del proceso de análisis de imágenes, a partir de las propiedades o información relevantes que se buscaron extraer en la etapa previa para su medición. Este conjunto de propiedades conforma un patrón de características, del cual sus medidas y parámetros deben ser evaluados por un clasificador para determinar la categoría a la que pertenecen. (Mansilla, 2015)

En este sentido, se buscan definir patrones en base a arreglos de descriptores o características. A partir de ello, un conjunto de patrones que poseen propiedades en común entre ellos define las categorías en las que pueden ser clasificadas. Por este motivo, las características o descriptores extraídos y seleccionados deben poseer un alto grado de separabilidad entre las clases, ya que estas afectan al desempeño en el proceso de reconocimiento de los objetos (Gonzales y Woods, 1996).

Asimismo, a través del aprendizaje automático se han desarrollado diversos algoritmos para realizar esta tarea de clasificación, como lo pueden ser las Máquinas de Vector Soporte, los Árboles de decisión, Las Máquinas de Boltzmann, las Redes Neuronales, etc. Estos buscan extraer las características de las imágenes y en base a ellas, reconocer la categoría correspondiente (Bu et al., 2017).

A continuación, se ahonda en algunos de los principales algoritmos utilizados para realizar la tarea de clasificación.

3.5.1 Máquinas de Vector Soporte (SVM)

Bishop (2006) define a Las Máquinas de Vector Soporte o Support Vector Machines (SVM) por su definición en inglés, como un algoritmo de clasificación capaz de mejorar la capacidad de generalización de los modelos de Aprendizaje Automático a través de la reducción en la generalización del error en los problemas de clasificación que se pueden definir según la ecuación 5.

$$y(x) = w^T \phi(x) + b \quad (5)$$

Donde $\phi(x)$ denota una transformación de características espaciales en la que se toma a las variables w y b como parámetros arbitrarios. Asimismo, se establece que el conjunto de datos de entrenamiento de tamaño N puede expresarse como un conjunto de vectores x_1, \dots, x_n con sus respectivas categorías t_1, \dots, t_n .

Una vez definido el problema de clasificación, las SVM's buscan reducir el error en la clasificación a través del concepto de maximizar el margen, el cual se define como la distancia perpendicular entre los hiperplanos que definen a los límites de decisión y las muestras de entrenamiento, y se obtiene tras resolver la ecuación 6.

$$\arg \max_{w,b} \left\{ \frac{1}{\|w\|} \min_n [t_n (w^T \phi(x_n) + b)] \right\} \quad (6)$$

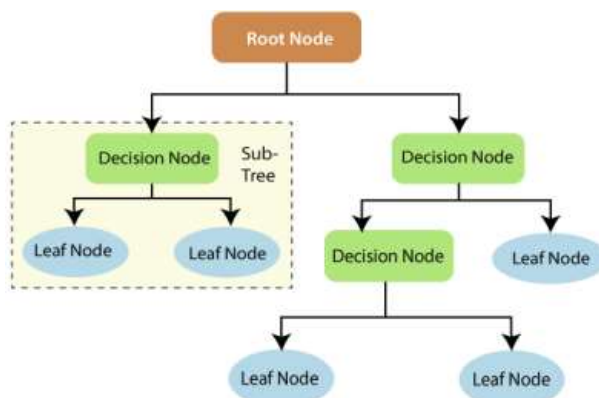
En esta el margen está dada por la distancia perpendicular al punto más cercano x_n desde el hiperplano generado por $y(x) = 0$ en el que se busca optimizar los parámetros w y b en razón de maximizar la distancia.

3.5.2 Árboles de Decisión (DT)

Taha y Mohsin (2021) definen a Los Árboles de decisión o Decision Trees (DT) por su símil en inglés, como un algoritmo de clasificación utilizado comúnmente en diversos campos como el aprendizaje automatizado, el procesamiento de imágenes y la identificación de patrones. Asimismo, los DT son un modelo sucesivo que une una serie de pruebas eficientemente, en las que una característica numérica es comparada contra un valor umbral en cada prueba. Esto se debe a que las reglas conceptuales son más sencillas de construir en comparación a pesos numéricos en las conexiones entre nodos de las redes neuronales. De esta manera, los Árboles de decisión son utilizados principalmente para propósitos de agrupamiento, así como en modelos de clasificación. Los "nodos" y las "ramas" están compuestas por cada árbol, en los que cada nodo representa distintas características en una categoría a ser clasificadas y donde cada subconjunto define un valor que puede ser tomado por el nodo, tal como lo muestra la Figura 3.5.2.1.

Figura 3.5.2.1

Modelo de Árbol de Decisión



Nota. De “Classification Based on Decision Tree Algorithm for Machine Learning”. Por B. Taha, A. Mohsin, 2021, *Journal of Applied Science and Technology Trends*, 2(1), p. 21. (<https://doi.org/10.38094/jastt20165>)

Asimismo, los Árboles de decisión operan bajo los criterios de “Entropía” y “Ganancia de Información”. La entropía es empleada para medir la impureza de un conjunto de datos, así como la aleatoriedad de este. El valor de la entropía siempre se encuentra entre 0 y 1, donde se busca que el valor de la entropía se acerque más a 0, lo cual se puede observar en la ecuación 7.

$$Entropy(S) = \sum_{i=1}^c P_i \log \log 2^{P_i} \tag{7}$$

Donde S representa el conjunto de datos con respecto al estado c , P_i es el ratio de número de muestras en el conjunto de datos S para el i -ésimo atributo de valor.

Por otro lado, la ganancia de información es un criterio utilizado para la segmentación y esta consiste en informar a través de la inferencia el conocimiento respecto al valor de una variable aleatoria. Se contrapone al funcionamiento de entropía ya que se busca que su valor sea mayor. La ganancia de información está representada por la ecuación 8.

$$Gain(S, A) = \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{8}$$

Donde el rango de atributo A se encuentra en $V(A)$ y la variable S_v es un subconjunto del ya definido del conjunto S igual al atributo de valor v .

3.5.3 Bosques Aleatorios (RF)

Los Bosques Aleatorios o *Random Forests* (RF) por su definición en inglés es un algoritmo utilizado en problemas de clasificación y regresión que nacen de la combinación de predictores de “árbol” de modo que cada uno de estos depende de los valores de un vector aleatorio independiente y de una misma distribución para cada “árbol” del “bosque”. En este sentido, los Bosques Aleatorios son entendidos como un conjunto o colección de “árboles” que poseen vectores aleatorios para cada “árbol” tal como lo muestra la ecuación 9.

$$\{h(x, \theta_k), k = 1, \dots\} \tag{9}$$

Donde la expresión $h(x, \theta_k)$ denota un clasificador de árbol en el que θ_k es un vector aleatorio independiente generado para cada “árbol” k y donde cada “árbol” perteneciente al “bosque” genera una salida unitaria para la clase más representativa del conjunto de entrada x .

De esta manera, dado un conjunto de clasificadores $h_1(x), h_2(x), \dots, h_k(x)$ y el conjunto de datos de entrenamiento establecido de la distribución de los vectores aleatorios X, Y se puede definir la función de margen denotada en la ecuación 10.

$$mg(X, Y) = av_k I(h_k(X) = Y) - av_k I(h_k(X) = j) \tag{10}$$

Donde $I(\cdot)$ es el indicador de la función de margen que permite medir el exceso sobre el que el promedio de salidas de los vectores aleatorios X, Y para la clase adecuada excede en el promedio de salidas para cualquier otra clase. Por tanto, mientras más grande sea el margen, mayor será la confianza en la clasificación.

Asimismo, para los Bosques Aleatorios, se puede denotar que a medida que los árboles crecen, las

secuencias θ_k converge en el resultado mostrado por la ecuación 11.

$$P_{X,Y}(P_{\theta}(h(X, \theta) = Y) - P_{\theta}(h(X, \theta) = j) < 0) \quad (11)$$

Este resultado, explica el porqué, a pesar de añadir más árboles de clasificación a los Bosques Aleatorios, estos no caen en el problema de overfitting o sobreajuste (Breiman, 2001).

3.5.4 Naive Bayes

Naive Bayes es un clasificador probabilístico de aprendizaje automático que opera relativamente bien en múltiples aplicaciones reales, a pesar de que asume que las características de un “objeto” son condicionalmente independientes dada la clase a la que pertenece. El proceso de aprendizaje de este clasificador se da en base a una estructura conocida, a partir de la que se calculan las posibles clases, así como las probabilidades condicionales tras el uso del conjunto de datos de entrenamiento, con lo que se obtienen los valores de estas probabilidades que serán utilizadas para clasificar nuevas observaciones.

Las redes bayesianas de esta forma se representan como las distribuciones de un conjunto de variables (características) $X = \{X_1, X_2, \dots, X_n\}$ que son usadas para el proceso de entrenamiento, en el que se identifican el aprendizaje de la estructura y de los parámetros de clasificación. Este aprendizaje se construye a partir de un grafo acíclico direccionado dado el conjunto X . En este grafo, cada nodo corresponde a la variable y cada arco denota una relación de causa entre 2 variables, donde un nodo asume el rol de padre y otro el de hijo. Asimismo, se usa la nomenclatura X_i para representar cada variable (característica) y su correspondiente nodo, y la variable $P_a(X_i)$ para representar un conjunto de padres relacionado al nodo X_i . De esta forma, la ecuación 12 representa la unión de probabilidades de distribución para la variable X mencionada, así como para sus asociaciones.

$$P(X) = \prod_{i=1}^n P(X_i | P_a(X_i)) \quad (12)$$

Sin embargo, para temas de clasificación se debe asumir que cada clase representa a un padre dentro de los nodos y que cada variable (característica) puede tener tan solo un padre como clase, lo cual permite una inducción óptima para el clasificador dado un conjunto de supuesto previos, lo cual se puede evidenciar en la ecuación 13.

$$P(C|X) = \frac{P(C) \prod_{i=1}^n P(X_i|C)}{P(X)} \quad (13)$$

Donde C representa una clase o categoría dada debido a que existe un conjunto de características X , en los que cada característica X_i son condicionalmente independientes entre sí (Taheri y Mammadov, 2013).

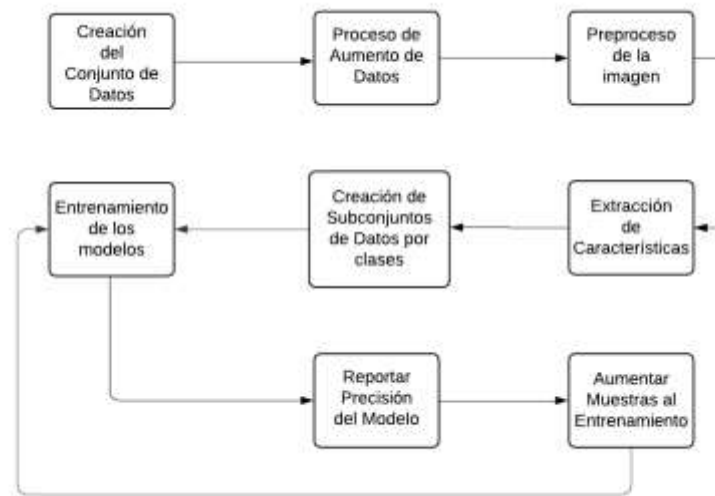
4. METODOLOGÍA

La clasificación de objetos es una de las tareas más recurrentes en los campos de la visión por computador, análisis de imágenes entre otros. La generación de clasificadores robustos depende de la cantidad de imágenes disponibles para sus entrenamientos, limitación que cobra mayor relevancia cuando los objetos exhiben una gran similitud en sus apariencias. En estos casos, la generación de una mayor cantidad de datos con muestras sintéticas permite sobreponerse en cierta medida a las limitaciones en la poca disponibilidad de muestras de entrenamiento. Sin embargo, esta práctica genera la incertidumbre de cuántas muestras se utilizarían para entrenar los modelos. Motivados por esta pregunta, nace la curiosidad por investigar cual es la cantidad mínima de muestras de entrenamiento para generar clasificadores robustos de objetos con apariencias similares. Para responder a esta pregunta, proponemos una metodología basada en el incremento del número de muestras en el entrenamiento de los clasificadores a razón del incremento de las categorías de objetos a clasificar.

La Figura 4.1 ofrece una breve representación de nuestra propuesta en la que el aumento de datos permite simular la disponibilidad de una mayor cantidad de muestras de entrenamiento. A continuación, se realizará una breve descripción de cada etapa mencionada en el diagrama de bloques.

Figura 4.1

Diagrama de Bloques del Proceso Incremental del Entrenamiento de Modelos



4.1 Creación del Conjunto de Datos

En base a lo expuesto por Manzoor et al. (2019) se han considerado como variables diferenciadoras al momento de definir las clases del conjunto de datos relativos a la marca y modelo del vehículo, así como el año de fabricación de dicho vehículo.

Tal como muestra la Tabla 4.1.1, se han considerado un total de 32 combinaciones diferentes de marcas y modelos vehiculares para la construcción del conjunto de datos. Asimismo, se cuenta con un total de 100 imágenes diferentes por cada combinación de marca – modelo vehicular, que serán utilizadas como muestras de entrenamiento. Asimismo, se tiene un total de 20 imágenes adicionales que se utilizarán como datos de prueba.

Tabla 4.1.1

Resumen de Marcas y Modelos Vehiculares registrados en el Conjunto de Datos

Marca	Modelo	Año Fabr.	Marca	Modelo	Año Fabr.
Toyota	Yaris	2013 - 2014	Kia	Sportage	2015 - 2016
Toyota	Tundra	2012 - 2013	Kia	Rio	2015 - 2016
Toyota	Rav	2015 - 2016	Kia	Picanto	2012 - 2013
Toyota	Prius	2013 - 2014	Kia	Cerato	2013 - 2014
Toyota	Corolla	2015 - 2016	Honda	Odyssey	2012 - 2013
Toyota	Camry	2012 - 2013	Honda	Civic	2012 - 2013
Toyota	Avalon	2012 - 2013	Honda	City	2013 - 2014
Toyota	4 Runner	2013 - 2014	Honda	Accord	2013 - 2014
Nissan	Versa	2015 - 2016	Chevrolet	Spark	2015 - 2016
Nissan	Tiida	2013 - 2014	Chevrolet	Corvette	2015 - 2016
Nissan	Sentra	2015 - 2016	Chevrolet	Captiva	2015 - 2016
Nissan	Qashqai	2015 - 2016	Chevrolet	Camaro	2015 - 2016
Lexus	NX	2014 - 2015	BMW	X3	2013 - 2014
Lexus	GX	2013 - 2014	BMW	M4	2015 - 2016
Lexus	GS	2012 - 2013	BMW	I3	2013 - 2014
Lexus	ES	2015 - 2016	BMW	320i	2013 - 2014

Las imágenes recolectadas son de las vistas frontales y laterales de los vehículos, tal como lo muestra la Figura 4.1.1 y principalmente provienen de repositorios abiertos como el que ofrece Kaggle en las cuales se disponen de múltiples conjuntos de datos. También, se usaron técnicas de “webscrapping” en otros sitios webs relacionados a la venta de vehículos, así como imágenes de fotos obtenidas por elaboración propia para completar el conjunto de datos.

Figura 4.1.1

Distintos Modelos vehiculares pertenecientes al conjunto de datos creado



4.2 Proceso de Aumento de Datos

En la búsqueda de enriquecer el conjunto de datos ya existente y, considerando que al entrenar los modelos para 32 clases probablemente se necesite una mayor cantidad de muestras de entrenamiento, se utilizan técnicas aumento de datos sobre las imágenes originales recolectadas, ya que el aumento de datos es una práctica comúnmente utilizada al operar con algoritmos de aprendizaje de máquina, puesto que permite ampliar el tamaño y la calidad de los conjuntos de datos a través de distintas transformaciones que permiten obtener mayor información de una imagen (Shorten y Khosroftaar, 2019). En razón a ello, primero se aplicará la rotación de modo que nuestros vectores de imágenes se multiplican por la matriz de rotación M_{RT} , con el fin de generar sobre ella rotaciones aleatorias en el rango de ± 7 grados, ya que se encuentran dentro del rango aceptado que indican Manzoor et al. (2019). Posteriormente, el resultado se multiplica por la matriz de traslación M_T de modo que la imagen se desplace aleatoriamente en el rango de ± 20 píxeles tanto vertical como horizontalmente. Finalmente, el resultado se multiplica por la matriz de reflexión M_{RF} restringida solo a la reflexión horizontal, de forma que solo de manera aleatoria la imagen pueda o no conseguir el efecto espejo deseado. Estas transformaciones se aplicaron a través de la plataforma Matlab, en la que se utilizaron las funciones disponibles para manejo de imágenes.

Tal como muestra la Figura 4.2.1, las transformaciones se aplican de manera secuencial sobre la imagen original. Por lo que, cada nueva imagen generada es el resultado de la combinación de las transformaciones mencionadas anteriormente. Asimismo, este proceso se ha realizado un total de 9 veces por cada imagen original disponible, con lo que finalmente el conjunto de datos aumenta de 3200 imágenes de entrenamiento a un total de 32000 imágenes. Además, a modo de aumentar la cantidad de imágenes de prueba, estas también han pasado por este proceso de aumento de datos con lo que se obtiene un total de 6400 imágenes disponibles.

Figura 4.2.1

a) Imagen Original b) Imagen con transformación de rotación y traslación c) Imagen con transformación de rotación, traslación y espejo horizontal



(a)



(b)



(c)

4.3 Preprocesamiento de la imagen

Para poder realizar la extracción de características de los vehículos, es necesario que las imágenes del conjunto de datos se encuentren en un formato de 224 x 224 píxeles, ya que es el parámetro de input aceptado por la red VGG16

que se utilizará. Por ello, se realiza un pre procesamiento de las imágenes, que consiste en un redimensionamiento de su formato no uniforme de $m \times n$ píxeles, en el que m y n representan el alto y ancho de la imagen respectivamente, al formato adecuado para la red VGG16.

Figura 4.3.1

a) Imagen Original del vehículo b) Imagen del vehículo redimensionada sin uso del template c) Imagen redimensionada del vehículo tras el uso del template.



Este preprocesamiento consiste, en primero en identificar el mayor valor de dimensión de alto y largo de la imagen original. Una vez se conoce este valor, se procede a generar una matriz de dimensiones $n \times n$, en la que n representa el mayor valor encontrado entre la dimensión de alto y largo. Esta matriz de dimensión $n \times n$ inicialmente contiene valores 0, que en una imagen representan un color negro. Asimismo, se extraen los valores que contiene la matriz asociada a la imagen original y se procede a ingresarlas en la matriz de dimensiones $n \times n$, de manera centrada, de modo que se genera una imagen con dimensiones de alto y anchos iguales, la cual ya puede ser multiplicada con nuestra matriz de escalamiento M_E con el fin de reescalar la imagen al formato deseado de 224×224 píxeles, puesto que, si realizamos el redimensionamiento previo a este paso, se puede afectar la forma original y las características de la imagen, tal como se muestra en la Figura 4.3.1 b). De esta manera, la Figura 4.3.1 c) nos permite observar cómo el vehículo mantiene sus características originales en comparación a la Figura 4.3.1 a), por lo que el redimensionamiento ya no deformaría a la imagen del vehículo. Este preprocesamiento se realiza sobre todas las imágenes del conjunto de datos y se realizó a través de la plataforma de Matlab a través de sus funciones disponibles para manejo de imágenes.

4.4 Extracción de Características

Como ya se mencionaba, para la extracción de características se utilizará la base de la estructura de la red convolucional VGG16, de la que se procede a eliminar las últimas 4 capas, ya que éstas se encuentran relacionadas con el proceso de clasificación.

Una vez que se poseen sólo las capas asociadas a la extracción de características, se procede a ingresar las imágenes redimensionadas en formato 224×224 píxeles para obtener las matrices correspondientes a las características de estas, las cuales se guardan en un formato npy. Este proceso de extracción de características se realizó en la plataforma de Jupyter Notebook, en la que se importaron distintas funciones asociadas a la librería de tensorflow para el uso de la red VGG16.

4.5 Creación de Subconjuntos de datos por Clases

Una vez que tenemos las características de cada uno de los vehículos del conjunto de datos, procedemos a generar subconjuntos en base al conjunto de datos principal, ya que la idea es hacer un comparativo de los modelos no solo aumentando la cantidad de muestras de entrenamiento, sino también aumentar la cantidad de clases de modelos vehiculares a clasificar. De esta manera, generamos 5 subconjuntos, que contienen las muestras de 2, 4, 8, 16 y 32 clases vehiculares. Cada nuevo subconjunto, mantiene las categorías del conjunto anterior, de modo que estas clases sean parte del nuevo análisis. Cada subconjunto cuenta inicialmente con un total de 25 muestras para el entrenamiento de los modelos, los cuales se aumentarán al término del entrenamiento.

4.6 Entrenamiento de los Modelos

Para el entrenamiento de modelos de clasificación, se utilizaron los algoritmos de Máquinas de Vector Soporte (SVM), Árboles de Decisión (DT), Bosques Aleatorios (RF) y Naive Bayes (NB), ya que según la literatura revisada, estos han demostrado altos índices de precisión en temas de reconocimiento vehicular. Asimismo, hemos trabajado con los parámetros por defecto para estos algoritmos, a modo de evaluar cómo operan en igualdad de condiciones sobre los subconjuntos creados en el proceso previo. Una vez entrenados los modelos, se procede a realizar las pruebas utilizando las imágenes separadas para este paso y a reportar la precisión alcanzada por los modelos. Finalmente, se

procede a aumentar las muestras de entrenamiento en los subconjuntos que se tienen. Tal como se menciona, se inicia con 25 muestras de entrenamiento y se aumentan a 50, 100, 150, 200, 400, 700 y 1000 muestras al término de cada entrenamiento por cada subconjunto de datos. De esta manera, al final se han entrenado un total de 160 modelos que nacen de la combinación de los 4 algoritmos de clasificación mencionados, los 5 subconjuntos originados de su separación por clases y los 8 experimentos con número de muestras que se aumentan entre experimentos. Este proceso de entrenamiento y prueba se realizó en la plataforma de Jupyter Notebook, en la que se importaron distintas funciones asociadas a la librería de tensorflow para el uso de los algoritmos de clasificación. Además, es necesario resaltar que todos los experimentos se llevaron a cabo en un procesador Intel Core i7 – 6700 de 3.4 GHz con una memoria RAM de 8GB.

5. RESULTADOS Y DISCUSIÓN

Tal como se mencionó previamente, la principal métrica a evaluar en cada modelo es la precisión con la que clasifica a los modelos vehiculares correctamente. Para ello, la Tabla 5.1 recoge las precisiones encontradas para los 160 modelos entrenados, tras la realización de las pruebas.

Por un lado, tal como se puede observar en la Tabla 5.1, los mejores resultados en todas las clases se presentan para el algoritmo de Máquina de Vector Soporte (SVM) logrando para el modelo que clasifica 32 modelos vehiculares diferentes con 1000 imágenes de entrenamiento por clase una precisión de 96.82%, seguido por los Bosques Aleatorios (RF), los Árboles de Decisión (DT) y el algoritmo de Naive Bayes (NB). Por otro lado, se puede observar que, en efecto, conforme se desean clasificar una mayor cantidad de modelos vehiculares, el número de muestras requiere ser incrementado, ya que de lo contrario la precisión de los modelos tiende a degradarse. Sin embargo, se puede evidenciar que este esfuerzo en incrementar el número de muestras es significativo hasta cierto límite, ya que, si bien sigue mejorando la precisión del modelo, la mejora ya no es tan grande como en el inicio. Por ejemplo, para el algoritmo de SVM al intentar clasificar 32 modelos vehiculares el aumentar de 100 a 400 imágenes de entrenamiento por cada modelo, nos representa una mejora de 63.52% a 91.11% en la precisión, pero cuando aumentamos de 400 a 1000 imágenes de muestra, la mejora es mínima, variando de 91.11% a 96.82%, a pesar de aumentar el doble de imágenes en comparación al primer caso señalado.

Tabla 5.1

Resumen Comparativo por Técnicas, Número de clases y Número de muestras de entrenamiento

# Clases	# imágenes de entrenamiento por clase	SVM	DT	NB	RF
2 clases	25 imágenes	0.7525	0.5575	0.6775	0.6975
	50 imágenes	0.825	0.575	0.6875	0.7075
	100 imágenes	0.895	0.6125	0.73	0.765
	150 imágenes	0.925	0.6425	0.7675	0.84
	200 imágenes	0.9375	0.72	0.77	0.86
	400 imágenes	0.9825	0.7725	0.795	0.935
	700 imágenes	0.9975	0.795	0.8225	0.965
	1000 imágenes	1	0.84	0.8325	0.9875
4 clases	25 imágenes	0.62125	0.4625	0.515	0.58625
	50 imágenes	0.68625	0.47625	0.59875	0.62375
	100 imágenes	0.83375	0.515	0.60875	0.70875
	150 imágenes	0.86375	0.53	0.60625	0.73125
	200 imágenes	0.91	0.59875	0.63375	0.79735
	400 imágenes	0.955	0.6175	0.675	0.86125
	700 imágenes	0.97625	0.66125	0.6925	0.90625
	1000 imágenes	0.98375	0.70125	0.71125	0.9375

(Continúa)

(Continuación)

8 clases	25 imágenes	0.4325	0.194375	0.304375	0.306875
	50 imágenes	0.536875	0.2325	0.363125	0.380625
	100 imágenes	0.70375	0.320625	0.445625	0.49625
	150 imágenes	0.795	0.33125	0.454375	0.579375
	200 imágenes	0.839375	0.356875	0.474375	0.640625
	400 imágenes	0.941875	0.445	0.48675	0.761875
	700 imágenes	0.974375	0.5175	0.49875	0.88625
	1000 imágenes	0.983125	0.571875	0.534375	0.92125
16 clases	25 imágenes	0.391875	0.1440625	0.244375	0.2921875
	50 imágenes	0.521875	0.1803125	0.31125	0.40125
	100 imágenes	0.680625	0.2090625	0.3578125	0.4971875
	150 imágenes	0.7584375	0.263125	0.3878125	0.5659375
	200 imágenes	0.8075	0.2865625	0.3878125	0.618125
	400 imágenes	0.90875	0.3471875	0.413125	0.7496875
	700 imágenes	0.96	0.4146875	0.4309375	0.8565625
	1000 imágenes	0.9765625	0.454375	0.4578125	0.8978125
32 clases	25 imágenes	0.3590625	0.10109375	0.2103125	0.23578125
	50 imágenes	0.4896875	0.129375	0.25953125	0.33
	100 imágenes	0.63515625	0.1709375	0.29015625	0.45953125
	150 imágenes	0.7346875	0.18953125	0.3040625	0.53296875
	200 imágenes	0.78890625	0.2253125	0.30453125	0.59265625
	400 imágenes	0.91109375	0.2865625	0.32375	0.73953125
	700 imágenes	0.9415725	0.36484375	0.35671875	0.84640625
	1000 imágenes	0.96815275	0.4134375	0.3825	0.89078125

Otro punto importante para resaltar es la precisión de los modelos entrenados bajo el algoritmo de Bosques Aleatorios. Si bien su precisión, es menor a la de los SVM, estos cuentan aún con un nivel aceptable de precisión, como por ejemplo en el caso de clasificación de 32 modelos vehiculares diferentes entrenados con 1000 muestras de entrenamiento por clase, en el que obtuvieron un 89.07% de precisión. Por otra parte, los algoritmos de Decision Tree y Naive Bayes en la experimentación no obtuvieron una alta precisión. Como se puede evidenciar, solo contaría con una precisión relativamente aceptable para los casos de clasificación de 2 y 4 modelos vehiculares y que aun así es menor en comparación a lo encontrado con los SVM y los Bosques Aleatorios. A pesar de su baja tasa de precisión en el reconocimiento, estos algoritmos de Decision Tree y Naive Bayes, también sirven como ejemplo del problema planteado de que el aumento de número de muestras no siempre brinda una mejora significativa en la precisión de los modelos, sino que es necesario desarrollar el uso de nuevos y mejores modelos.

Asimismo, es interesante comparar nuestros resultados frente a otras investigaciones asociadas al reconocimiento y clasificación vehicular. Para ello, la Tabla 5.2 nos muestra un comparativo resumen de nuestros experimentos frente a los experimentos de los autores mencionados en la sección de Estado del Arte. Como se puede observar, nuestro experimento más complejo se da en la clasificación de 32 modelos vehiculares, en la que nuestra precisión del 96.82% para el algoritmo de clasificación SVM es superior en comparación al resto de la literatura en la mayoría de los casos. Este punto es significativo, ya que, a pesar de clasificar una mayor cantidad de modelos, nuestra precisión se mantiene por encima. Sin embargo, en comparación a los modelos basados en SVM planteados

por Manzoor et al. (2019), nuestro modelo se comporta con una ligera menor efectividad en la precisión. Esto se puede explicar, debido a que Manzoor varía el parámetro C en los SVM con los que realiza la clasificación para validar el comportamiento de los SVM ante este cambio. Con lo que si nos comparamos frente a su experimento con el parámetro C con valor 2. Podemos notar que posee una precisión de menos del 95.5%, lo que sugiere que nuestros experimentos en condiciones similares se asemejan. Este análisis propone algunas alternativas de estudio a futuro, como lo son la variación en los parámetros de nuestros clasificadores.

Tabla 5.2

Resumen Comparativo de nuestro trabajo con otros encontrados en la literatura, en base a la tasa de precisión obtenida.

Autores	# Modelos a Clasificar	# Imágenes de entrenamiento	Extractor de Características	Algoritmo de Clasificación	% de Precisión del Mejor Experimento
Huang et al. (2008)	20	1414	2DLDA	KNN	94.70%
Psyllos et al. (2009)	10	400	SIFT	SIFT	87%
Pearce y Pears (2011)	22	262	Local Normalised Harris Strengths (LNHS)	KNN	94.90%
				Naive Bayes	96%
Bu, Lan y Xu (2017)	19	1900	CNN (Caffenet)	CNN (Caffenet)	94.75%
Manzoor et al (2019)	35	2725	HOG	RF (350 DT)	94.43%
			HOG	SVM (C=10)	97.89%
			GiST	RF (300 DT)	94.53%
			GiST	SVM (C=10)	97.20%
Nuestros Experimentos	32	32000	VGG16	SVM (C=1)	96.82%
				RF (DT=100)	89.07%

6. CONCLUSIONES

Este trabajo se enfoca en determinar la cantidad mínima de muestras de entrenamiento para obtener clasificadores robustos en escenarios en los que las apariencias entre objetos son similares, específicamente se abordó el estudio en el marco de la clasificación de modelos vehiculares. Para ello, se han implementado 4 algoritmos de clasificación (Máquinas de Vector Soporte, Árboles de decisión, Bosques aleatorios y Naive Bayes) que trabajan sobre los vectores de características otorgados por una Red VGG16 a la que se le han retirado la capa de clasificación. Asimismo, las imágenes de muestra han sufrido distintas transformaciones para enriquecer en cantidad y variabilidad el conjunto de datos. Finalmente, los experimentos se llevaron a cabo bajo una metodología secuencial en la que, tras obtener el rendimiento de un clasificador, se procedió a aumentar el número de muestras de entrenamiento utilizados.

Los resultados muestran que el algoritmo de Máquinas de Vector Soporte sobresale y obtiene el mejor rendimiento en el problema de clasificación de modelos vehiculares. Sobre los escenarios propuestos para este clasificador, se concluye que existe un número crítico de muestras de entrenamiento a partir del que ya no se generan ganancias en el rendimiento, por lo que este tiende a estabilizarse y la mejora ya no es sustancial. En el caso presentado de clasificación de modelos vehiculares, este número se encuentra alrededor de las 400 muestras de entrenamiento cuando se trata de 2 categorías a clasificar, mientras que para 4 a más categorías (en nuestro más complejo 32 categorías) el rendimiento de los clasificadores se estabiliza y tiende a ser aceptable a partir de las 700 muestras de entrenamiento. Por otro lado, para el caso de los demás clasificadores, existe la tendencia a seguir mejorando conforme se incrementa el número de muestras.

7. TRABAJOS FUTUROS

Esta investigación se centró en analizar el comportamiento de los algoritmos de clasificación en condiciones por defecto, sin realizar alguna variación en sus parámetros internos, por lo que a futuro se podría realizar un estudio analizando cómo la variación y el manejo de los parámetros pueden influir en los modelos entrenados y en sus porcentajes de precisión, ya que, según lo visto por Mazoor et al. (2019) la precisión de los modelos tiende a mejorar. Por otro lado, sería interesante utilizar la red VGG16 no sólo como un extractor de características, sino aprovechar su estructura y, a través de un proceso de Transfer Learning, reentrenar sus pesos para observar su comportamiento en este problema de clasificación. Finalmente, este estudio puede ser la base para futuros modelos que contribuyan en ámbitos de transporte inteligente, seguridad vial, detección de vehículos robados, entre otros temas relacionados.

AGRADECIMIENTOS

Agradezco a la universidad por todo lo que he aprendido a nivel profesional y personal en ella durante estos años. A toda la plana docente con la que tuve la dicha de compartir en clases o asesorías. A los profesores Victor Andrés Ayma y Victor Hugo Ayma, quienes me apoyaron durante todo el proceso de realización de este trabajo y con los que he aprendido a ser un mejor profesional. A mis abuelos y a mi padre que sé que siempre guían e iluminan mi camino. Y en especial a mi hermana y a mi madre, quien siempre estuvo ahí para darme ánimos a completar mis objetivos y me enseña día a día el significado de nunca rendirse.

REFERENCIAS

- Bishop, C. (2006). Pattern Recognition and Machine Learning.
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45, pp. 5-32.
- Bu, Q., Lan, S., & Xu, P. (2017). A CNN based car model recognition improvement. (1), 86–89. doi:10.1145/3094243.3094258
- Dong, Q., Wu, A., Dong, N., Feng, W. & Wu, S. (2018). A Convolution Neural Network for Parts Recognition Using Data Augmentation 2018 13th World Congress on Intelligent Control and Automation (WCICA), 2018, pp. 773-777, doi: 10.1109/WCICA.2018.8630451.
- Gonzalez R., Woods R., (2002) Digital Image Processing 2nd Edition
- Gholamhosseinian, A. & Seitz, J. (2021). Vehicle Classification in Intelligent Transport Systems: An Overview, Methods and Software Perspective, *IEEE Open Journal of Intelligent Transportation Systems*, vol. 2, pp. 173-194, 2021, doi: 10.1109/OJITS.2021.3096756.
- Hu, T., Qi, H., Huang, Q. & Lu, Y. (2019). See Better Before Looking Closer: Weakly Supervised Data Augmentation Network for Fine-Grained Visual Classification. doi:10.48550/ARXIV.1901.09891
- Huang, H., Zhao, Q., Jia, Y., & Tang, S. (2008). "A 2DLDA Based Algorithm for Real Time Vehicle Type Recognition," 2008 11th International IEEE Conference on Intelligent Transportation Systems, 2008, pp. 298-303, DOI: 10.1109/ITSC.2008.4732559.
- Kadambari, K.V., & Nimmalapudi, V.V. (2020). Deep Learning Based Traffic Surveillance System For Missing and Suspicious Car Detection. ArXiv, abs/2007.08783.
- Kaur, Er & Kaur, Yadwinder. (2014). Object classification Techniques using Machine Learning Model. *International Journal of Computer Trends and Technology*. DOI:18. 170-174. 10.14445/22312803/IJCTT-V18P140.
- Lehr, J., Schlüter, M. & Krüger, J. (2019). Classification of Similar Objects of Different Sizes Using a Reference Object by Means of Convolutional Neural Networks. 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2019, pp. 1519-1522, doi: 10.1109/ETFA.2019.8869120.
- Mansilla, N. (2015). Procesamiento de imágenes aplicada a la tipificación vacuna: análisis de indicadores geométricos basados en curvatura.
- Manzoor, M., Morgan, Y., & Bais, A. (2019). Real-Time Vehicle Make and Model Recognition System. *Machine Learning and Knowledge Extraction*. 1. 611-629. DOI:10.3390/make1020036.
- Mendoza, F. & Lu, R. (2015). Basics of Image Analysis. doi:10.1007/978-1-4939-2836-1_2.
- Nash, W., Drummond, T. & Birbilis, N. (2018). A review of deep learning in the study of materials degradation. *npj Mater Degrad* 2, 37 https://doi.org/10.1038/s41529-018-0058-x
- Ong, P. & Chong, T. & Woon Kiow, L. (2020). Development of Class Attendance System Using Face Recognition for Faculty of Mechanical and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia. doi:10.4018/978-1-7998-0182-5.ch001.
- Ozkurt C, Camci F. Automatic Traffic Density Estimation and Vehicle Classification for Traffic Surveillance Systems Using Neural Networks. *Mathematical and Computational Applications*. 2009; 14(3):187-196. https://doi.org/10.3390/mca14030187
- Pearce, G. & Pears, N. (2011). Automatic Make and Model Recognition from Frontal Images of Cars. *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance*, pp. 373-378.
- Psyllos A., Anagnostopoulos C. N. & Kayafas, E. (2009) SIFT-Based Measurements for vehicle model recognition 2103-2109

- Ruuska, S & Hämäläinen, W., Kajava, S. & Mughal, M., Pekka, M., & Mononen, J. (2018). Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle. *Behavioural Processes*. DOI:148. 10.1016/j.beproc.2018.01.004.
- Shawal, S, Shoyab, M. & Begum, S. (2014). Fundamentals of Digital Image Processing and Basic Concept of Classification. *International Journal of Chemical and Process Engineering Research*. 1. 98-108. 10.18488/journal.65/2014.1.6/65.6.98.108.
- Shijie, J., Peiyi, J., Siping, H., & Haibo, L. (2017). Automatic detection of tomato diseases and pests based on leaf images. *2017 Chinese Automation Congress (CAC)*, 2017, pp. 2537-2510, doi: 10.1109/CAC.2017.8243388.
- Shorten, C. & Khoshgoftaar, T.M. (2019). A survey on Image Data Augmentation for Deep Learning. *J Big Data* 6, 60 <https://doi.org/10.1186/s40537-019-0197-0>
- Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. 1–14. <http://arxiv.org/abs/1409.1556>
- Taha, B. & Mohsin, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, 02, pp. 20 - 28.
- Taheri, S., & Mammadov, M. (2013). "Learning the naive Bayes classifier with optimization models". *International Journal of Applied Mathematics and Computer Science*, vol.23, no.4, 2013, pp.787-795. <https://doi.org/10.2478/amcs-2013-0059>
- Zhu, X., Vondrick, C., Ramanan, D & Fowlkes, C. (2012). Do We Need More Training Data or Better Models for Object Detection?. *BMVC 2012 - Electronic Proceedings of the British Machine Vision Conference 2012*, pp1-10. <https://doi.org/10.5244/C.26.80>.

ESTUDIO SOBRE LA CANTIDAD MÍNIMA DE MUESTRAS DE ENTRENAMIENTO PARA LA CLASIFICACIÓN DE MODELOS VEHICULARES

INFORME DE ORIGINALIDAD

14%	12%	6%	9%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	Submitted to Universidad de Lima Trabajo del estudiante	2%
2	hdl.handle.net Fuente de Internet	1%
3	Tiantong Wang, Daniel Trugman, Youzuo Lin. "SeismoGen: Seismic Waveform Synthesis Using GAN With Application to Seismic Data Augmentation", Journal of Geophysical Research: Solid Earth, 2021 Publicación	<1%
4	es.scribd.com Fuente de Internet	<1%
5	Submitted to Deakin University Trabajo del estudiante	<1%
6	Seppo Sierla, Mohammad Azangoo, Kari Rainio, Nikolaos Papakonstantinou, Alexander Fay, Petri Honkamaa, Valeriy Vyatkin. "Roadmap to Semi-Automatic Generation of	<1%