

# Car Image Classification

DS 340 Final Project

Jassim Kanoo, Angelo Orciuoli

(Ordered Alphabetically)

## Introduction

Artificial intelligence stands as one of the most transformative technologies in history. Due to its widespread applicability, our team decided to harness it for a project centered around automobiles—a subject of mutual interest. Our objective was to develop an image classifier capable of distinguishing between 5 different car types, including Sedans, SUVs, Hatchbacks, Sports cars, and Pick-up Trucks. We selected these specific types of cars for both variety and similarity. To be more specific, Pickup Trucks and Sports Cars look quite different from each other while Hatchbacks and Sedans look similar and potentially be mistaken for each other by people. By having diversity and similarity in our data, we were set up to evaluate whether our models would be extremely accurate, if they correctly classified cars belonging to different categories but look similar, or somewhat accurate, if they only correctly classified cars that are clearly distinguishable. The motivation behind this project stemmed from the practical utility of such a classifier in various industries, including automotive manufacturing, as well as becoming an application in technologies related to self-driving vehicles and dashcam software. Additionally, with the ever-growing volume of digital images and videos related to automobiles on the internet and social media platforms, there is a pressing need for automated systems that can accurately categorize and analyze these visuals. Finally, another use case is that car classifiers could serve as a valuable tool for car enthusiasts, automotive journalists, and industry analysts who frequently encounter vast amounts of car-related imagery in their work. To achieve this, we employed the TensorFlow machine learning library, using sequential modeling to create 5 convolutional neural networks. Models 1 and 2 use a VGG16 architecture, a predeveloped deep architecture that has a strong performance in image classification tasks, which we expected to be our most effective approach that is particularly beneficial given our limited dataset. Models 3, 4, and 5 are also sequential models, but vary in architecture and depth, meant to capture the tradeoff between efficiency and accuracy.

## Methodology

To develop our car type image classifier, our process involved several key steps: data collection, data augmentation, model building, and result interpretation.

### 1. Data Collection:

We sourced JPEG images of cars from Kaggle, an online database. These images, with dimensions of 180x180, were organized into subdirectories by car type. Our dataset included 1,493 images of SUVs, 726 of Sports Cars, 729 of Hatchbacks, 1,669 of Pick-up Trucks, and 1,222 of Sedans. Each image was labeled according to the subdirectory it was stored in, with paths saved to ensure they were correctly identified as JPEG files for training and testing purposes. We meticulously prepared the data, allocating 75% of the images in each category for training and 12.5% for validation and 12.5% for testing.

### 2. Data Augmentation

When first creating and testing our models, we found bias in favor of SUVs or Pickup Trucks and suspected it could be due to the difference in the number of images across the 5 different car types (Pickup Trucks and SUVs had more than double the number of images Hatchback and Sports Cars had). To accommodate for this, we implemented data augmentation to make up for the difference in input images. After augmenting the data, where we rotated and inverted pictures, each car type had somewhere between 1300 and 1330 images for training, 200 for validation, and 200 for testing. Although this is not equivalent to gathering more input data from external sources, implementing this data augmentation did increase our models' accuracy.

### 3. Model Development:

- Two Transfer Learning Models (VGG16): We utilized the VGG16 architecture, excluding the top layer to customize it for our specific needs, classifying among 5 car categories. The two VGG16 models were trained over seven epochs, incorporating a ReLU activation function for hidden layers and a softmax activation in the output layer, ideal for handling multiclass categorization. A particular benefit of VGG16 is that it has 1000 pre-trained classes in its model, two of them being 'sports\_car' and 'pickup.' Our second VGG16 model extracts these two classes and decodes the class labels from the ImageNet dataset. Then, we use it to customize our model by reclassifying those into our Sports\_Car and Pickup\_Truck categories. This approach enables us to leverage the knowledge encoded within VGG16's pre-trained weights, adapting it specifically to our task of classifying car types, thus enhancing the model's performance and efficiency.

- Three More Sequential Models: Model 3 is a 15-layer model constructed including five sets of convolution and pooling layers, along with flattening, dropout, and dense layers. Each convolutional layer employed a ReLU activation function, transitioning to a softmax activation for the output layer. The pooling layers utilized a (2,2) pool size. Model 4 uses a simpler architecture with fewer convolutional layers and a single dropout layer for regularization, allowing us to gain insight on the tradeoff between accuracy and efficiency. Lastly, Model 5's architecture is a deeper version of Model 3's, by adding additional convolutional and pooling layers, resulting in a deeper and more complex network. This model also provided insight on accuracy and efficiency differences between models.

#### **4. Model Compilation:**

For all models, we used the Adam optimizer and the 'sparse\_categorical\_crossentropy' loss function. This setup ensured consistency in how the models were trained and evaluated against the validation data. We considered using 'categorical\_crossentropy', which requires assignment of a binary sequence of length 5 to each image (zeros representing car categories an image does not belong to), but ultimately decided to stick with 'sparse\_categorical\_crossentropy' because our models' accuracy was already quite high, with a low loss.

#### **5. Concluding Methodology**

By carefully managing these steps, we strived to build a robust model that not only classifies car types with high accuracy but also explores the capabilities of CNNs in distinguishing complex features in automotive imagery.

## Results

Based on 5 different CNNs, our second VGG16 model (model 2 in the .ipynb file) was our best model as it yielded the highest accuracy (97.3%) and the least loss (0.1). Although the accuracy seems inflated, VGG16 is a pre-trained model that processes millions of images over 1000 different categories, thereby enhancing the generalization capabilities which causes an additional 5 classes, summing to ~8500 images, essentially negligible to VGG16's model. Here are the accuracy and loss scores for each of our models (Accuracy, Loss): Model 1 (95.6%, 0.14), Model 2 (97.3%, 0.1), Model 3 (85.1%, 0.44), Model 4 (91.5%, 0.64), Model 5 (94.2%, 0.36). We derived these scores from evaluating our model over the testing set. Achieving such high accuracy and low loss indicates the efficacy of the model in capturing and representing intricate patterns within the dataset. This robust performance underscores the benefits of leveraging transfer learning from pre-trained models like VGG16, as well as sequential models, which learn feature representations from vast amounts of diverse data. In addition to computing accuracy and loss scores, we pulled external images to upload and process through our models to see what they would be classified as. After testing various images of different cars, among our 5 categories, we found our model was best at predicting Sports Cars, Pickup Trucks, and Hatchbacks. Similar to the issue we encountered before augmenting the data, we found that our model now has a bias towards Sports Cars and Pickup Trucks, which makes sense since they were the only two categories that we pulled from the VGG16 network. However, when comparing other models, we found that pulling from VGG16 pre-trained classes led to more success when testing on external images.

## Conclusion

After running our models, we concluded that convolutional neural networks can classify images of cars with a high degree of accuracy. Given our results we found using pre-trained networks extremely beneficial as it streamlined the process of creating a deep and robust model, as well as providing us with classes to pull from. When future machine learning or artificial intelligence opportunities arise, we now have an instinct to start with pre-trained models and backtrack or build off from them rather than develop our own model from scratch. In terms of the topic of our project, we have learned the integration of car classification artificial intelligence into modern day technology can allow society to advance further into areas that automation has not been successful in before. There are inevitable use cases for car classification in the future of driving.