Car-Truck Classification Using Various State-Of-The-Art CNN Models.

Babalwa Dumane  
*Computer Science Department*  
 *University of the Western Cape*Cape Town, South Africa  
3719746@myuwc.ac.za

Ntando Mniki  
*Computer Science Department*  
 *University of the Western Cape*Cape Town, South Africa  
3538564@myuwc.ac.za

# Introduction

The advancement of image processing, recognition and retrieval has progressed immensely, and this is influenced by the variety of datasets collected and the variety of convolutional neural networks.

Object recognition in images has blown up in research recently and this has made vehicle recognition possible to explore [1]. Vehicle Recognition is used in various avenues of interest, such as vehicle security where this approach can be used in vehicle theft prevention, the analysis of traffic, the electric toll (e-toll) collection process, and some smart cities.

Cars possess a couple of unique distinctive properties, which is why Convolutional Neural Networks (CNN) are used to perform these classifications [2]. Multiple studies and reports have shown that the use of Convolutional Neural Networks on large scale image datasets have produced high accuracy results in multiple classification tasks [3]. Vehicle detection is usually the first approach, followed by the classification approach, and both of these are performed by using Convolutional Neural Network models and architecture.

CNN models achieve high results in general Image classification, and can underperform with fine grained images and more especially if the images appear small or are cluttered. This was however improved by the use of a cross-convolutional-layer pooling method [4]. Cross Convolutional-layer pooling methods is a strategy that withdraws subarrays of feature maps of a convolutional layer as local features and uses the feature maps of the following convolutional layers as pooling channels. Then the drawn-out features are pooled with the same channels, to create progressively strong representations [2].

Experiments are conducted on a car/truck dataset, using three state of the art CNN models together with the proposed model which has 4 convolutional layers, each followed by a pooling layer and two fully connected layers. These experiments show the state-of-the-art models ResNet50 and VGG16 outperforming our proposed model as compared to the LeNet which had an accuracy of 50%

# Motivation

A two-class vehicle type classification is proposed. The objective is to build a model that can classify a two-class vehicle type using the car-truck dataset from Kaggle, and compare these results to state of the art classification models. We propose to create a good performing model that will take in raw data and produce desired results with high validation accuracies and a good learning rate, using Convolutional Neural Networks (CNN). Our data contains pictures of small cars and trucks, the desired output would be able to predict whether an image is a car or a truck. We will use different approaches from the old-fashioned CNN modelling to more advanced modelling such as LeNet, ResNet50 and VGG16, comparing the accuracy and errors of the testing data to perform these predictions.

## Dataset:

The dataset contains 789 images of cars and trucks. The cars are a total of 393 images and the trucks are a total of 396 images making 788 altogether. The dataset is then split by 80/20. Where 80% was for training, and 20% was for testing and validation.

Images from the internet were used to perform car type predictions using the various models created and trained on.

# Literature Review

This project was a lower scale version of car make and model recognition, only recognizing the type of the car, whether it is a truck or a car, and while there have not been much implementations focusing on only classifying cars and trucks, the technologies that can be used are quite similar to a CMMR system.

## Preliminaries

### Using Symmetrical Surf

There have been previous applications of car make and model recognition (CMMR), these include Jun-Wei Hsieh et al. [6], who used symmetrical SURF for their vehicle make and model recognition (VMMR). SURF, as described in this paper, is Speeded Up Robust Features. It is a feature detector for vision-based applications, but it is unable to detect symmetrical objects [6]. Their paper suggests the use of a symmetrical SURF descriptor to enhance the power of SURF to detect all symmetrical matching pairs through a mirroring transformation [6]. They then suggest the adoption of a vehicle make and model recognition application to prove the practicality of the symmetrical SURF method. The symmetrical descriptor is applied to determine the region of interest (ROI) of each vehicle on the road without using motion features. They found out that the advantages of this are; there is no need for background subtraction and it is efficient for real-time applications. Model and Make Recognition (MMR) poses challenges, these are multiplicity and ambiguity problems. The multiplicity problem happens when one vehicle model has different model shapes and the ambiguity problem stems from two different vehicles from different companies have similar shapes [6]. To address the problems, Jun-Wei Hsieh et al., suggests the use of a grid division scheme. The grid division scheme separates the vehicle into several boxes or grids and different weak classifiers trained on these grids are put together to build a strong classifier. This classifier can accurately recognize each vehicle, because of the rich representation power of the grid-based method combined with the high accuracy of vehicle detection [6].

Jun-Wei Hsieh et al., used a database from an SVM library in [9] to train the vehicle classifier. This database contains a collection of 2846 images. The testing database contains 4090 images. Twenty-nine vehicle makes and models were collected in [6] to evaluate performance. They recorded the speed of the system to be 21*fps* and the velocity of observed vehicles was permitted to be up to 65 km/h. They tested the application under a number of conditions. These including vehicle detection under cloudy days, under rainy days, when the front light of the vehicle was turned on, when irrelative objects appeared in the scenes and when it was during the night [6]. They found their method to work better under cloudy conditions as opposed to sunny conditions. They also found it to work well in night scenes as opposed to day scenes. Their method yielded a precision of 98.48, false alarm of 1.34, miss rate of 0.49 and operated at a speed of 43.83*fps.* This proved better, in comparison to other methods.[6]

### Coarse-to-fine Convolutional Neural Network Architecture for fine grained Car Make and Model Recognition

Fine grained vehicle recognition is identifying manufacture and detailed model of a vehicle [7]. It is a challenging problem in intelligent transport systems due to the subtle intra-category variation in appearance. This is addressed by locating parts that can easily be distinguished, where the most significant variation appears, based on the large training set [7].

In [7], Fang et al. propose a coarse-to-fine method to achieve this, in which the distinguished areas are detected automatically based on feature maps extracted by convolutional neural networks. They established a mapping from feature maps to the input image in order to locate the regions and these regions are repeatedly refined until there are no more qualified ones. Global and local features are then extracted from the whole images and the detected regions, respectively [7]. Based upon low-level variation and holistic cues within the global and local features, a one-versus-all support vector machine (SVM) is used for classification.

Different methods have been proposed in such fields, which take advantage of 3D model and low-level features, such as Scale-Invariant Feature Transform (SIFT), Histogram of Oriented Gradients (HOG) and Salford Predictive Modeler (SPM) to identify vehicle makes and models. Vision-based fine-grained vehicle model recognition, however, is a tough task due to subtle variation in appearance among lower level vehicle categories, with which the fine-grained model cannot be easily distinguished even by a human without domain knowledge [7]. This is why fine-grained vehicle recognition needs more powerful, distinguished features to classify objects between certain classes. This is quite challenging, which is why there are quite a few fine-grained vehicle model recognition applications [7].

A coarse-to-fine CNN for fine-grained vehicle model recognition is proposed in [7], where the most unique or distinguishable parts are automatically detected via feature maps generated by the CNN. This method allows the model to learn which parts of the image are significant for identifying subordinate-level model variation. CNN is able to learn hierarchical features from large datasets using the multilayer feed-forward structure [7]. The feature learning process used in the study in [7] is a coarse-to-fine process that extracts features from local and global regions. The local features can describe low-level variation, while the global features describe more general variations. A one-versus-all SVM classifier is used based on the learned features. The results were an accuracy of 98.29% over 281 vehicle makes and models, outperforming most state-of-the-art approaches.

### 3D curve alignment for Car Make and Model Recognition

In [10], a study about using 3D curve alignment for CMMR is done. They argue that most existing methods are limited to recognition based on a fixed point or view. They propose using 3D curve alignment to combat this issue which allows verification of a car’s make and model from an arbitrary view. Their model consists of 3D space curves obtained by back-projecting image curves onto silhouette-based visual hulls and then refining them using three-view curve matching. These are then matched to 2D image curves using a 3D view-based alignment technique. The aim is to be able to recognize the make and model of a car over a wide range of angles.

In [10], each car model is represented using a set of non-parametric 3D curves. They construct the curves from natural images as opposed to constructing them from Computer-Aided Design (CAD). They use a view-based representation consisting of separate partial 3D curve models instead of using a single global 3D model, where each model is associated with the viewpoint corresponding to each of the training images. To align the 3D curve model to a new test image, they find the closest training image and then align the corresponding partial 3D curve model to the edges in the test image by estimating a rigid 3D perspective image transformation [9]. The advantage of this is that the subtle view dependent features can be modelled and the visibility of the curves is handled naturally.

### Car Make and Model Recognition under night-time lighting conditions

In [13], a study of CMMR under night-time lighting conditions is conducted because a majority of the studies on CMMR conduct their studies during daytime and therefore present solutions to the problem during daytime when the cars are visible. There are a few methods that can still identify car makes and models at night, however their accuracies are low. In this study, a new CMMR method for night scenes is proposed. The model uses new feature selection methods and a one-class classifier to identify a vehicle of interest. The front lights of a vehicle tend to be in limited lighting conditions and this results in a brightness and glare. This causes blurring when the most important features are captured, resulting in recognition inaccuracy. To combat this, this paper proposes that the features to be used in the CMMR should be captured from the rear view because features are the least affected by the brightness and glare. Moreover, the shapes of the backlights of a vehicle as well as the license plate contain information which can be utilized during CMMR.

They proceed to use the majority vote of three classifiers to classify the prominent features, to increase the vehicle identification accuracy. The classifiers used are one-class support vector machines (OCSVM), decision tree (DT) and k-nearest neighbors (kNN). They propose a system consisting of two processes, these are training and classification. In the training process, target images and other car model images are input to the feature extraction process, which extracts features of interest. This process involves license plate detection, tail light detection and feature extraction. Feature subset selection is then applied to determine the ideal feature set for the particular target model.

The classifier is trained from parameters of obtained training data to identify the model and the final result is the trained target car model which is used in the classification process. They perform a real time classification, where a stream of images from CCTV footage containing different vehicle models is considered. The images undergo training, feature extraction, feature selection and then finally classification. Predefined features are first extracted, then optimized features of a particular target vehicle model are used to classify with the target vehicle model. The result is either the target or the other vehicle model [13].

They used a dataset containing 421 car models with a total of 766 images. The dataset consists of two types of models: target models, and the other models. There are 100 of the target models used to classify against other car models. Each target model has at least four sample images and there is one image per other car model [4]. They used MATLAB version R2013a for the implementation of the classifiers. Tenfold cross-validation was utilized for the confirmation of the experimental results in each model. They tested each target model against 420 other models using tenfold cross-validation; four feature sets were evaluated separately. They obtained an average accuracy of 93.8%. The accuracies of the four feature sets were reported at 93.7%, 94.0%, 93.6% and 93.8% respectively [13].

### Deep Convolutional Neural Networks with Spatially Weighted Pooling for Fine-Grained Car Recognition

Unlike generic object classification, fine-grained classification distinguishes subcategories within the same car category [2]. In [2], they use pooling as part of their process. Pooling involves localizing various discriminative parts of an object, or the car in this case, each corresponding to a human-specified object part. Local features corresponding to each part are pooled together to obtained a pooled feature vector used for classification [2]. Then part-based detectors are trained in a supervised method. Deep Convolutional Neural Networks have recently been shown to outperform methods like SIFT or the Fisher vector. Extracting features from the convolutional layers of a DCNN pretrained on ImageNet and using them to replace SIFT, allows the fisher vector with DCNN features achieve spectacular results on a number of classification tasks [2].

In the study, they found that even though DCNNs achieve good results in generic object classification, they still have a poorer performance compared to the aforementioned methods in fine-grained classification tasks. They found that this is because their architectures are not optimal for fine-grained objects, especially in small or cluttered objects. Cross-convolutional-layer pooling presents a breakthrough [4]. Cross-convolutional-layer pooling extracts subarrays of convolutional feature maps (CFMs) of a convolutional layer as local features and uses the CFMs of the successive convolutional layer as pooling channels [2]. The extracted features are then pooled with these pooling channels to generate more robust image representations. This method yields great results on several popular visual classification tasks [2]. A bilinear CNN framework [11] is proposed for fine-grained visual recognition, where two feature extractors based on DCNNs are used. One extracts the features and the other generates pool channels [2].

In [2], they propose a spatially weighted pooling (SWP) technique which enhances the robustness and effectiveness of the feature representation of the foremost DCNNs for the fine-grained car classification. The SWP has a predefined number of spatially weighted masks or pooling channels. The SWP pools are extracted features of DCNNs with the guidance of its learnt masks [2]. The SWP layer can be embedded in most DCNNs, including AlexNet, VGGNets and ResNets, with slight changes. This method addresses the setbacks or disadvantages of the previously mentioned methods.

Four fine-grained car datasets are used for experimenting the proposed method and results show that the SWP method improves the accuracies of fine-grained vehicle classification. ResNet101 and VGG16 achieve 90.9% and 85.4% accuracy, respectively, on the Cars196 dataset. The proposed method, on the other hand, improves on this accuracy bringing performance up to 93.1% and 90.7% respectively which are the best reported results on the Cars196 dataset and CompCars dataset [2].

### Lightweight Convolutional Neural Networks for vehicle recognition with thermal infrared images

Kang et al. [12] demonstrates the use of thermal infrared cameras in the field of driver assistance systems (DAS), particularly in detecting or recognizing vehicles in night scenes. A dataset of thermal infrared images involving four vehicle classes was used, these classes were bus, truck, van and car. The study does not particularly classify the make and model of vehicles, it rather identifies the type of vehicle and presents the idea of using thermal images to be able to identify these vehicles at night. This is especially useful for driver assistance systems at night to avoid traffic injuries and road accidents. There are driver assistance systems that function excellently during the day but they may not be suitable for night-time. The proposed system in the study aims to combat that problem. They are able to classify the vehicles because there are differences of heat distribution among vehicle types. A thermal infrared camera can sense the radiation of the objects and can measure the objects without disturbing the temperature field measured [12]. They utilized a dataset of 4108 pre-processed images and manually truncated it into 504 buses, 376 trucks, 224 vans and 886 cars.

They designed a CNN called Net1to use as the reference network, having four layers to classify four vehicle types. They had strides set to 1 and kernel sizes set to 3 by 3. They had four maximum pooling layers with strides set to two and an output layer with nodes set to number of categories. The activation function used was Rectified Linear Units (ReLU) on each layer except for the pooling and input layers. SoftMax is used on the output layer.

They also construct nine CNNs (Net2 – Net10) based on Fire module (from SqueezeNet) which are lightweight CNNs. Comparing the performance of the nine CNNs with Net1 as the reference convolutional network, Net9 was found to be the optimal one with a classification accuracy of 97% with 10.6% of the number of parameters on Net1 [12].

They conclude that night-time thermal infrared images are of high quality, which allows 24-hour traffic monitoring.

### Vehicle Make Recognition using Convolutional Neural Networks

In [8] they propose a method to recognize the make and model of moving vehicles based on CNN. They follow a process that is, to detect the moving vehicle using frame difference which results in a binary image. This image is used to detect the vehicle’s front view by a symmetry filter and the front view is then used to identify the vehicle using a CNN algorithm.

They propose a convolutional neural network with several convolutional layers, max-pooling layers which serve to increase the tolerance of translation in the image, a relu layer which gains the non-linear properties of the network and a SoftMax loss layer. The neural network requires a large memory size and a lot of training time because there are billions of parameters involved in the neural network. To combat this, a CNN is developed where the weights are shared between small portions of the images.

The input images were resized into 67 by 67 images and fed into a convolutional layer by a filter of 10 by 10 followed by a max pooling layer. This results in 20 feature maps. They repeated this process once more to produce 50 feature maps. Using a convolutional layer, 500 neurons were generated and fed into a ReLU layer to achieve nonlinearity. These neurons are then fully connected and forwarded to softmax loss to produce 107 classes [8].

The dataset used consists of 3210 car images with 107 models and 30 images per model. Each image was shifted by 10 pixels to create a neighboring image. The detection algorithm yielded a 100% accuracy which concludes that it is very effective. The proposed CNN method was compared to local binary pattern (LBP), local Gabor binary pattern (LGBP) and scale-invariant feature transform (SIFT). CNN achieved 88.4% accuracy, SIFT achieved 78.3%, LGBP achieved 68.8% and LBP achieved 46.0% [6]. It is obvious that the CNN method is the best.

## Similar existing systems:

### Automatic Toll Collection (ATC) SYSTEM

Authors Nadiya Shvaia, Abul Hasnata et al. of the report “Accurate classification for Automatic Vehicle Type Recognition based on ensemble classifiers” conducted research for vehicle recognition and its use in Automatic Toll Collection (ATC) [5]. According to these researchers this has already been implemented in other first world countries, and humans are still used to monitor and manually check the correctness and incorrectness of classification, this of course for the economy-based reasons. A simple misclassification could cost the company a lot, that’s why they still have humans to check their systems and the outputs they produce. These researchers' report is driven by this and they plan on refining and enhancing the current ATC system that uses Optical Senses (OS) for CNN car classification [5]. They came to a realization that the Optical Sensors makes various misclassifications of vehicle classes and decided that the solution to this problem is to recondition and upgrade the sensor itself, the other option was to look deeper and use other sources of information. The current ATC system takes a picture and the assistant, who is a person, must come and check the classification and fix it manually if it is classified incorrectly. This then unlocked the opportunity to use the images and to come up with an automated computational “vision-based vehicle type classifier using an efficient method” [5] such as the CNN. This can then be used together with the optical senses to intensify the overall performance. They were then motivated to produce a system that uses data from both the optical senses and the camera by putting together results from the optical senses and the image features and classifiers. The classification aims and problems that arise with AVR depend on the verification and vehicle maker and model recognition and ATC. “While verification measures the similarity of two vehicle images, VMMR performs fine grained classification with hundreds of possible classes” [5]. In order to develop a good and complex Automatic Toll Collection system, complex and different variations of data are collected and the images are then categorized according to certain appearance and physical features. These researchers could collect this data and conduct their report courtesy of VINCI Autoroutes company. In the data collection phase, they classified the images in 5 classes, from motorbikes, small private cars, Vans, heavy vehicles and large trucks. They were also captured in different conditions such as the ideal clear image, the image with lighting and occlusion images. This is important for an ATC system because the billing system is also based on the image classes, the conditions are beneficial for the training and testing of the model. Toll gates have multiple optical sensors, and cameras. There are around four optical sensors, two on each lane that surround the car, they are used to measure the weight, the height and the number of axles of the vehicle. The camera is used to take pictures and video that will later be checked by the operators if they are correct. The current ATC systems also use the Optical sensors to determine the car make and model [5]. For the ATC system, the proposed Neuron Network classifier attains notable improvement as a “stand-alone” classifier. There are some restrictions such as correctly distinguishing the correct categories, but these can be solved using the optical sensors to be classified correctly. This shows that a powerful method for the ATC system problem can be produced by properly and effectively putting together the optical sensors and the image classifiers. The proposed car-type recognition model used, executes the method at two layers. The first layer puts together the outputs of the two optic sensors based and convolutional NN-based types of classifiers. It gives a vector as input for the following layer. The second layer joins the classifiers choices acquired by training on various weighted sets (output from the feature layer). “The Gradient Boosting (GB) strategy is applied to perform this task” [5]. The proposed strategy is assessed on the gathered dataset and compared with the current system. Results show that it fundamentally beats the current system with an enormous (99.03% contrasted with 52.77%) average and consequently mitigates the need to utilize the huge amounts of humans. Additionally, examination with a set of CNN based models proves that it performs better than the other approaches. Using the two proposed models which is the CNN and the gradient Boost method to train the data and test against the validation data, there was an increase in accuracy which went up by a significance of 3.2% [5].

# Methodology

## Preprocessing

To preprocess our data, we augmented our images, using ImageDataGenerator, which is a Keras deep learning neural network library. This is a procedure used to artificially extend the size of the dataset by making adjusted renditions of images in the dataset. This technique can make varieties of the pictures that can improve the capacity of the models to sum up what they have learnt to new pictures.

The ImageDataGenerator techniques used are rescaling, horizontal flip, height shift range of 0.5, rotation range of 30 degrees and we filled the image using fill\_mode ‘nearest’.

Our proposed model was performing badly on the newly generated images, we then resulted in training on the original dataset and used the augmented dataset to perform predictions, to see if our trained models could recognize the different variations of similar images.

## Evaluation Process

We trained the model on 80% of the data, which was training data, then 20% of the validation data. The test data was manually created using images from the internet, to perform predictions. This approach was chosen because the more data you have to train your model, the higher the learning accuracy, 20% for validation is a fair split to validate the model against trained model.

# Architecture

## ResNet50

We used a pretrained resnet50 architecture, pretrained on ImageNet weights that are available on the Internet. We then fine-tuned the model like below.

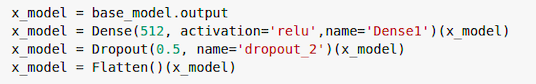


We added a global average pool to reduce the number of parameters and computation in the network. We also added two dense layers, two dropout layers to prevent overfitting and a flatten layer to convert the data into a 1-dimensional array to input into the layer that follows it. We used a regularizer of 0.0001 to further prevent overfitting.

We trained this model on 10 epochs, but after 10 epochs, it overfits (achieves an accuracy of 100%) for training data, so 5-7 epochs would also be ideal. We used ReLu for activation because it is, according to research, quick to evaluate.

## VGG16

VGG16 followed a similar fine-tuning architecture as the ResNet50 model, except for the omission of a dense layer and a dropout layer, so it looked like the following.



This model was also trained on 10 epochs and achieved an accuracy of 90.51% on the validation data.

## LeNet

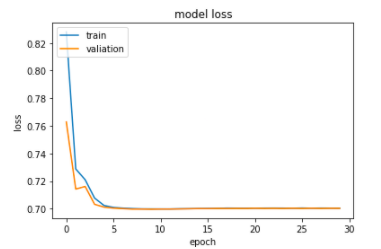
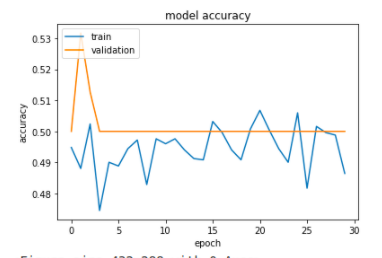
LeNet was the poorest performing model amongst the four models we trained. It had two convolutional layers, two average pooling layers, two dropout layers, a flatten layer and three dense layers. It achieved an accuracy of 50% on the training and validation data

## Basic CNN model

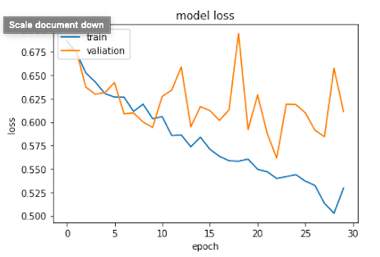
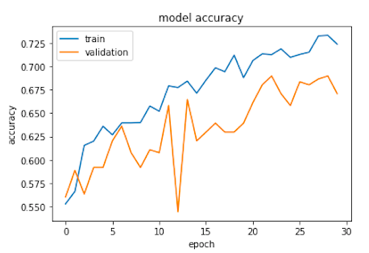
This model had four convolutional layers, one average pooling layer, three max pooling layers, two dense layers, a flatten layer, and batch normalization to improve speed and stability of the CNN.

# VISUALISATION OF THE RESULTS

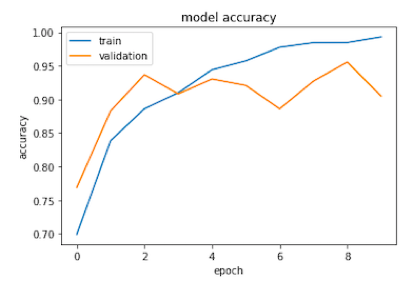
## LeNets:



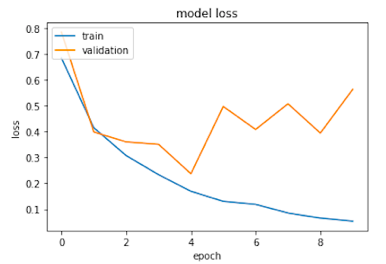
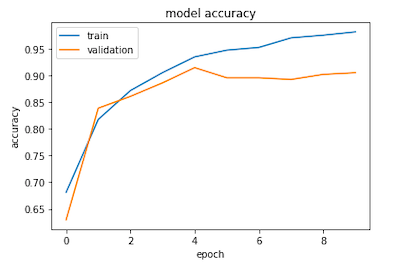
## CNN Model



## VGG16

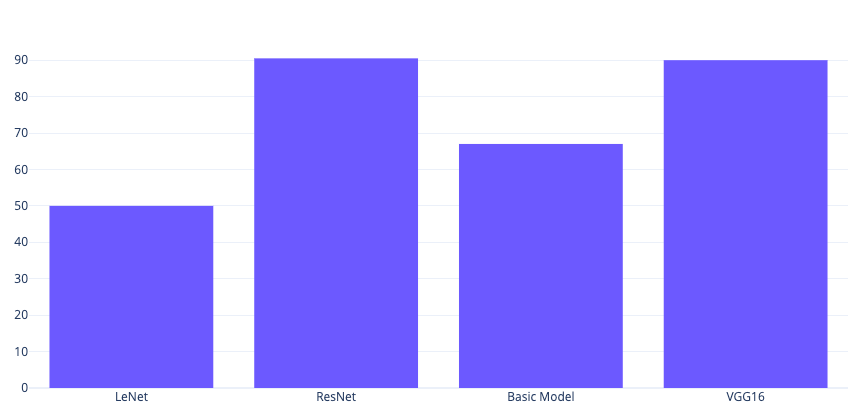


## ResNet50



## Model Accuracies

This is a representation of the accuracies of our different models, with ResNet Having the highest accuracies, followed by VGG16 and our proposed model.



# Evaluation of the Results

On the original unaugmented testing data, we decided to use all the models we trained and then predict on the best two. The two sequential models, LeNet and the basic CNN model had an expected difficulty classifying the images. LeNet classified the cars and the trucks as trucks while the CNN model classified most of the trucks as cars.

ResNet50 and VGG16, however, were able to classify most if not all correctly and performed well even on the augmented testing data.

# Discussion and Outlook

The issues we encountered include limited computational power, it would have been ideal to use augmented data and play around with the models to get a high accuracy in the training phase of our project as well but because of limited RAM space and the amount of time it takes to train models, we could not do that. In the future, perhaps with unlimited computational power we could do augmentation on the images, and maybe even expand our project to be able to classify images based on make and model of the cars.

The takeaway from this particular project is that transfer learning models, like ResNet50 and VGG16 are the best for image classification, based on our dataset.

##### References

1. N. Garcia and G. Vogiatzis, “Learning non-metric visual similarity for image retrieval,” Image Vis. Comput., vol. 82, pp. 18–25, 2019, doi: 10.1016/j.imavis.2019.01.001.
2. [2] Q. Hu, H. Wang, T. Li, and C. Shen, “Deep CNNs with Spatially Weighted Pooling for Fine-Grained Car Recognition,” IEEE Trans. Intell. Transp. Syst., vol. 18, no. 11, pp. 3147–3156, 2017, doi: 10.1109/TITS.2017.2679114.
3. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” Int. Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, 2015
4. Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification. In Proc. IEEE Conf. Comp. Vis. Patt. Recogn., pages 4749–4757, 2015.
5. N. Shvai, A. Hasnat, A. Meicler, and A. Nakib, “Accurate classification for automatic vehicletype recognition based on ensemble classifiers,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 3, pp. 1288–1297, 2020, doi: 10.1109/TITS.2019.2906821.
6. L. C. Chen, J. W. Hsieh, Y. Yan, and D. Y. Chen, “Vehicle make and model recognition using sparse representation and symmetrical SURFs,” Pattern Recognit., vol. 48, no. 6, pp. 1979–1998, 2015, doi: 10.1016/j.patcog.2014.12.018.
7. J. Fang, Y. Zhous, Y. Yu, and S. Du, “Fine-Grained Vehicle Model Recognition Using,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1782–1792, 2017, doi: 10.1109/TITS.2016.2620495.
8. Y. Gao and H. J. Lee, “Vehicle make recognition based on convolutional neural network,” *2015 IEEE 2nd Int. Conf. InformationScience Secur. ICISS 2015*, pp. 1–4, 2016, doi: 10.1109/ICISSEC.2015.7371039.
9. C. C. Chang and C. J. Lin, “LIBSVM: A Library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–39, 2011, doi: 10.1145/1961189.1961199.
10. E. Hsiao, S. N. Sinha, K. Ramnath, S. Baker, L. Zitnick, and R. Szeliski, “Car make and model recognition using 3D curve alignments,” *2014 IEEE Winter Conf. Appl. Comput. Vision, WACV 2014*, pp. 285–292, 2014, doi: 10.1109/WACV.2014.6836126.
11. T. Y. Lin, A. Roychowdhury, and S. Maji, “Bilinear CNN models for fine-grained visual recognition,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 International Conference on Computer Vision, ICCV 2015, pp. 1449–1457, 2015, doi: 10.1109/ICCV.2015.170.
12. Q. Kang, H. Zhao, D. Yang, H. S. Ahmed, and J. Ma, “Lightweight convolutional neural network for vehicle recognition in thermal infrared images,” *Infrared Phys. Technol.*, vol. 104, p. 103120,2020,doi:10.1016/j.infrared.2019.103120.
13. N. Boonsim and S. Prakoonwit, “Car make and model recognition under limited lighting conditions at night,” *Pattern Anal. Appl.*, vol. 20, no. 4, pp. 1195–1207, 2017, doi: 10.1007/s10044-016-0559-6.