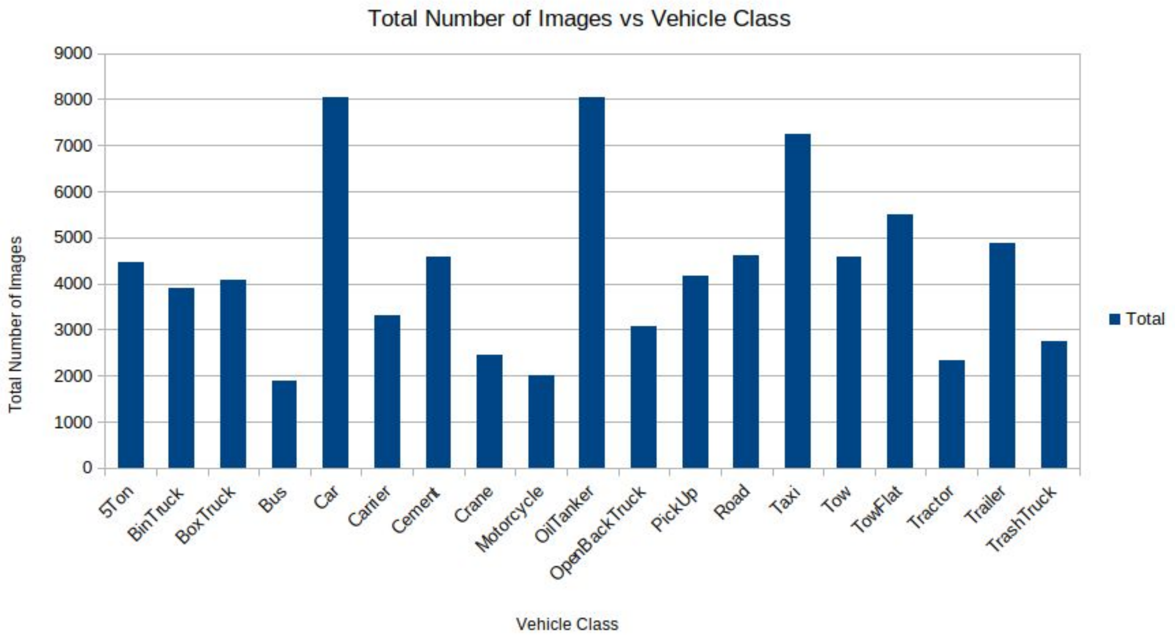
Vehicle Classification With VGG19

In a total of 82006 images of vehicles taken on the road, the images are labeled into their respective folders, which signifies the class that they belong to. There are a total of 19 classes of vehicles, namely, 5Ton, BinTruck, BoxTruck, Bux, Car, Carrier, Cement, Crane, Motorcycle, OilTanker, OpenBackTruck, PickUp, Road, Taxi, Tow, TowFlat, Tractor, Trailer, TrashTruck.

All the images in the dataset are coloured images cropped from the top view of a camera, and they all vary in dimensions. It is noted that all the images are back views of vehicles, and consist of either one of 3 angles: straight, rotated 45 degrees to the left, rotated 45 degrees to the right. Besides that, some images in the dataset have already been augmented by flipping, or have already been added noise, or blurred.

Before starting out, a total of 10 images per class was moved into a separate folder for validation purpose. The images are also moved into their respective folders to signify their class labels. This results in a remaining of 81816 images. The images that were moved out are proper images that have not yet been augmented. The distribution of the remaining images according to their classes are plot in the graph as follows.



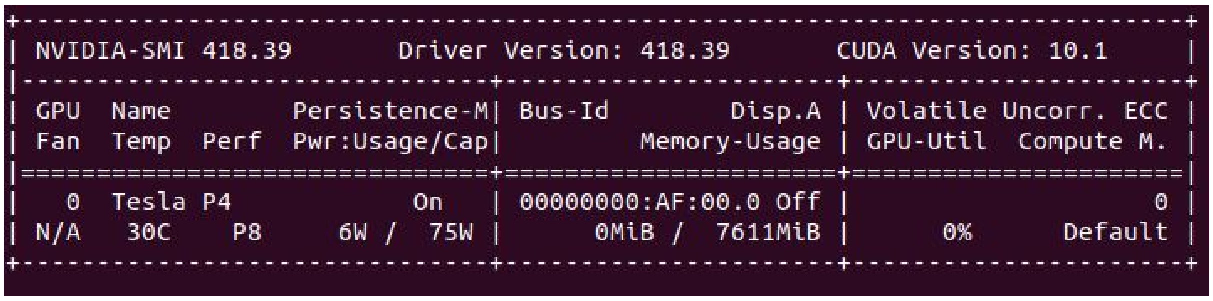
The 81816 images are then split into a ratio of 0.85:015, where 85% of the images were used for training the model while the remaining 15% were used to test the accuracy of the model. This results in a total of 69544 images for training and 12272 images for testing. The splitting method was done using scikit-learn library's train_test_split. It is noted that all the images are then resized to dimensions of 150 × 150 pixels, regardless for training or testing. Also, the training dataset was not shuffled, though shuffling the data may yield better generalization of the model, resulting in better predictions.

The model for the vehicle classification was done using Keras's built-in VGG19 model, a pre-trained Convolutional Neural Network (CNN) by Karen Simonyan and Andrew Zisserman. However, the original VGG19 model have been modified a little to cater to the classification of the 19 vehicles. The architecture of the model used is defined in the table as follows:

Convolution 64 (kernel size: 3) (activation: ReLU) (padding: same)
Convolution 64 (kernel size: 3) (activation: ReLU) (padding: same)
Max Pooling (pool size: 2) (strides: 2)
Convolution 128 (kernel size: 3) (activation: ReLU) (padding: same)
Convolution 128 (kernel size: 3) (activation: ReLU) (padding: same)
Max Pooling (pool size: 2) (strides: 2)
Convolution 256 (kernel size: 3) (activation: ReLU) (padding: same)
Convolution 256 (kernel size: 3) (activation: ReLU) (padding: same)
Convolution 256 (kernel size: 3) (activation: ReLU) (padding: same)
Convolution 256 (kernel size: 3) (activation: ReLU) (padding: same)
Max Pooling (pool size: 2) (strides: 2)
Convolution 512 (kernel size: 3) (activation: ReLU) (padding: same)
Convolution 512 (kernel size: 3) (activation: ReLU) (padding: same)
Convolution 512 (kernel size: 3) (activation: ReLU) (padding: same)
Convolution 512 (kernel size: 3) (activation: ReLU) (padding: same)
Max Pooling (pool size: 2) (strides: 2)
Convolution 512 (kernel size: 3) (activation: ReLU) (padding: same)
Convolution 512 (kernel size: 3) (activation: ReLU) (padding: same)
Convolution 512 (kernel size: 3) (activation: ReLU) (padding: same)
Convolution 512 (kernel size: 3) (activation: ReLU) (padding: same)
Max Pooling (pool size: 2) (strides: 2)
Flatten
Dense (output: 512) (activation: ReLU)
Dropout (0.5)

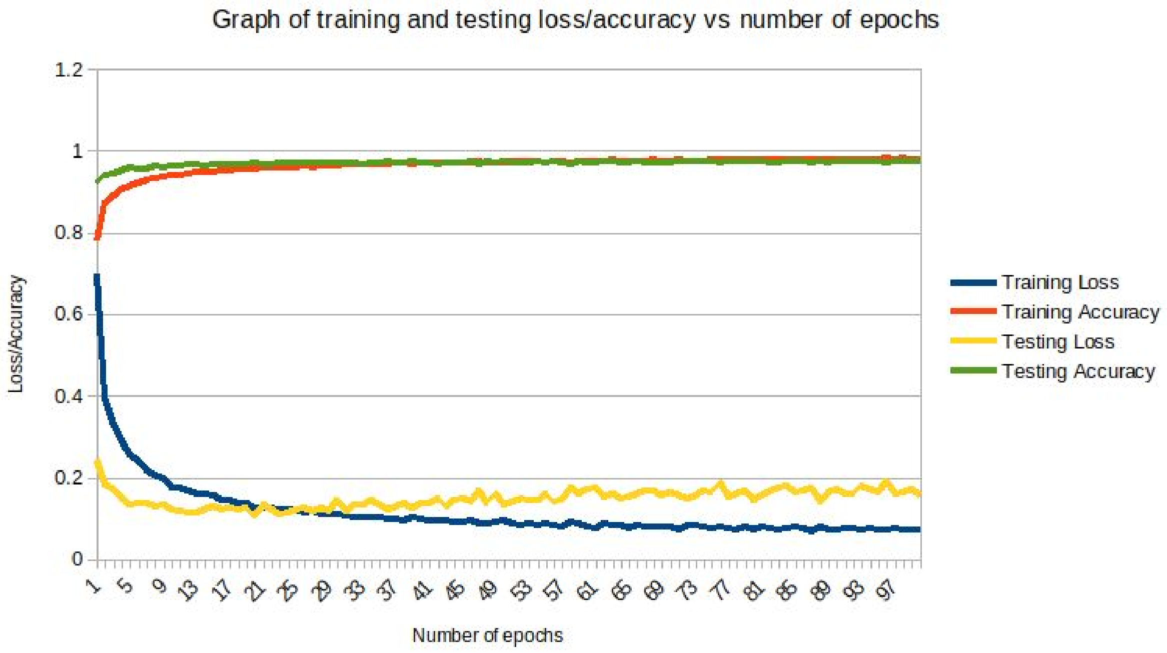
Dense (output: 19) (activation: Softmax)

The training was done using NVIDIA's Tesla P4 GPU, with a total memory of 7.61GB. The model was run in a Docker container using NVIDIA's CUDA 10 image. An overview of the GPU is shown in the figure below.

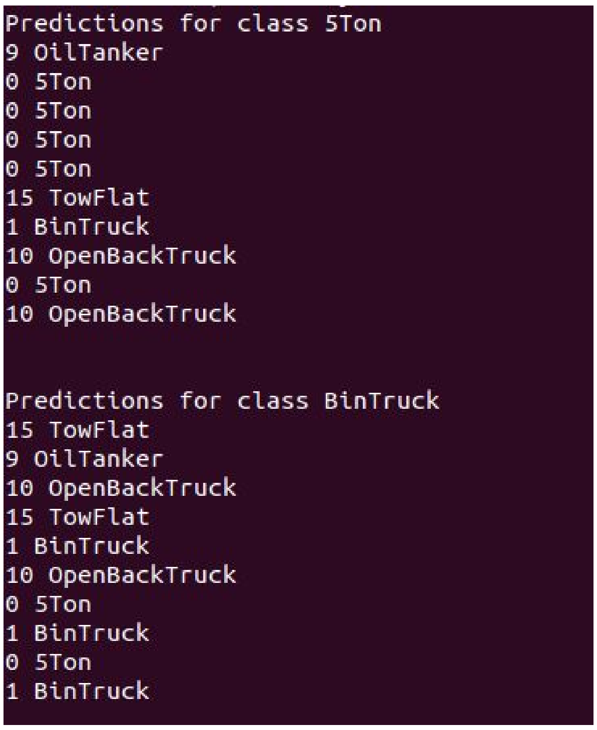


```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.39       Driver Version: 418.39       CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P4            On   | 00000000:AF:00.0 Off |                    0 |
| N/A   30C    P8     6W /  75W |      0MiB /  7611MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
```

A total of 100 epochs were run to train the model, with a batch size of 32. During the model fitting, the default settings for Adam was used as the model's optimizer, whereas sparse categorical cross entropy was used as the model's loss. The total run time per epoch ranges from 355 seconds to 359 seconds. After the training was done, the model was able to achieve a test accuracy of 97.56%. The training and testing accuracy and loss are shown in the graph below.



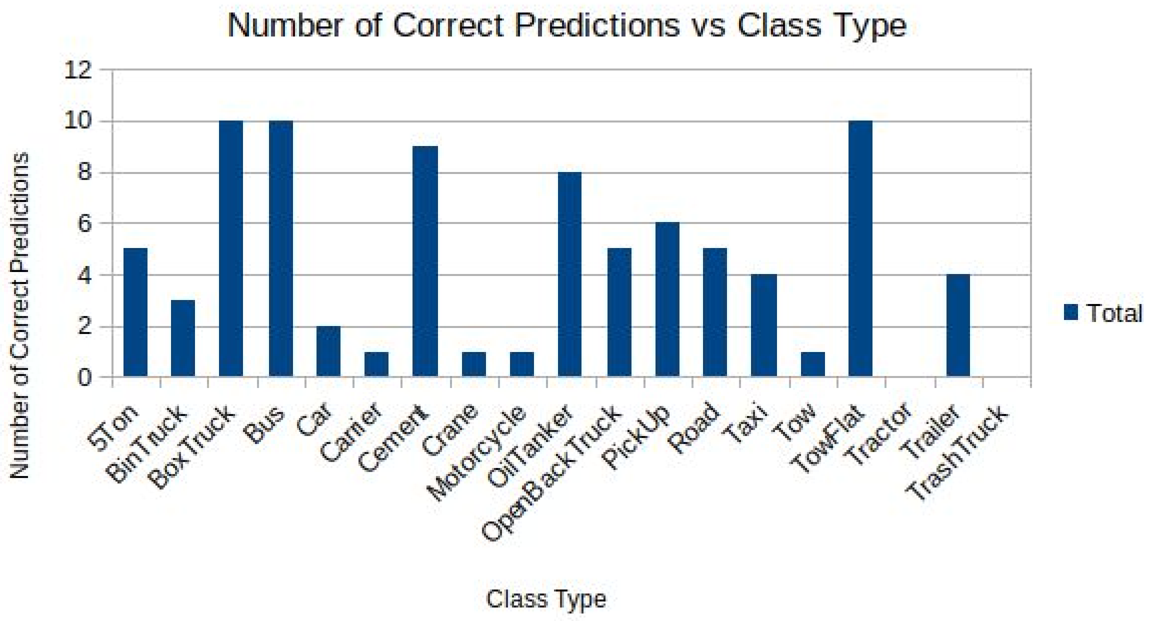
Once the training and testing accuracy of the model have been produced, the model can be validated by making predictions on the 190 images (10 images for 19 classes) that were moved out earlier. A portion of the predictions made is as shown in the screenshot below. It is noted that for each prediction, the number represents the index number of each class, beginning from 5Ton to TrashTruck, in an alphabetically increasing order.



```
Predictions for class 5Ton
9 OilTanker
0 5Ton
0 5Ton
0 5Ton
0 5Ton
15 TowFlat
1 BinTruck
10 OpenBackTruck
0 5Ton
10 OpenBackTruck


Predictions for class BinTruck
15 TowFlat
9 OilTanker
10 OpenBackTruck
15 TowFlat
1 BinTruck
10 OpenBackTruck
0 5Ton
1 BinTruck
0 5Ton
1 BinTruck
```

The prediction accuracy of all 19 classes are plot in the following graph.



Based on the graph, despite the Car and OilTanker class having the most number of training datasets, they are not among the classes with the highest number of correct predictions. This could be due to the amount of variations in the image dataset, especially for the Car class (i.e. there are cars of many different shapes, colours, and designs on the road as compared to other vehicles).