**APS360**

**Artificial Intelligence Fundamentals**

Final Report

Team 4

| Student name | Student number |
| --- | --- |
| Varun Sampat | 1003859602 |
| Eric Geerts | 1004319720 |
| Atharva Datar | 1004123254 |
| Vineet Kaushik | 1004151963 |

Word Count: 2496 /2500

Penalty: 0%

**1.0 Introduction**

E-commerce is a rapidly growing market in the world today, posting year over year ~15% gains in retail market share in the United States since 2010[1]. In conjunction vehicle e-commerce is a market sector that is primed for rapid expansion, if current growth rates continue 1.3 million vehicles in the United States alone will be bought and sold online[2]. Therefore there is a distinct need (and competitive advantage) for new technologies that improve existing vehicle e-commerce business processes. This is the **motivation** behind our project.

The **goal** of the project is to develop a program that will utilize artificial intelligence to classify the make of a vehicle from a photograph. Initially, the group had aimed to also determine the color of the car. But due to a few constraints, that was not proceeded with. A successful implementation of this project would be **useful** in improving the data analysis capabilities of vehicle e-commerce companies due to the high quality of data it would provide (current automated methods of analysis are often unreliable due to human error at the data input stage[3]).Machine learningis an appropriate tool for this project since **image recognition (computer vision)** is an area where **machine learning has a history of being successful**[4].

The report highlights the key steps taken in the completion of the project. It includes a discussion of previous projects completed in the field, how data processing was carried out, the architecture of the final model and a baseline model developed for comparison. The report concludes with a discussion of results and finally outlines some of the challenges and difficulties we overcame during the course of this project.

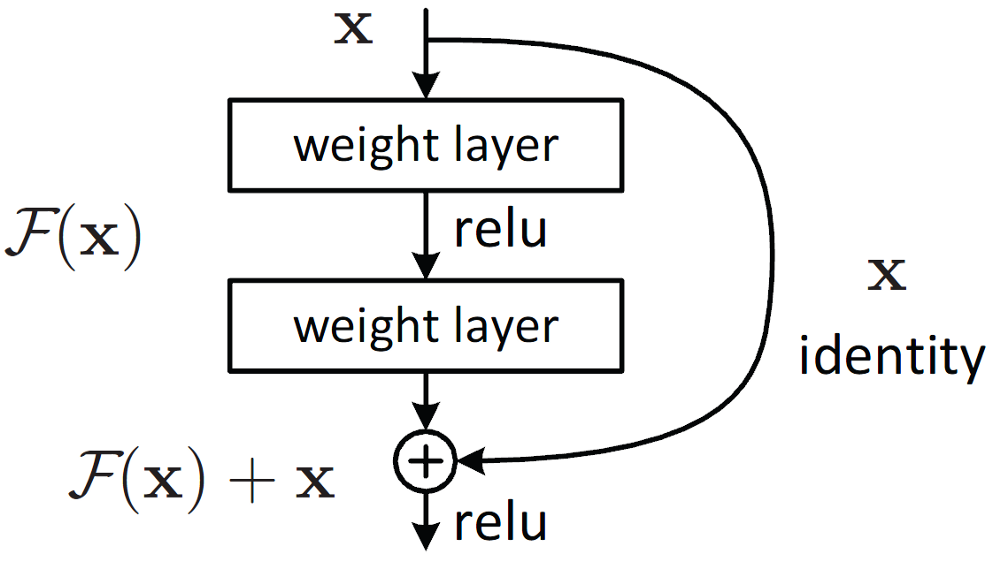
**2.0 Illustration**

Our team decided to implement a model that will predict the brand (make) of a car given its image. Like any image classification problem, this task also consisted of two stages:

1. **Image/data cleaning or augmentation**: this includes random rotating, random flipping horizontally, resizing and making the image color invariant, so that the features of the car could be extracted and noise can be minimized
2. **Brand Prediction**: processed image is passed though a trained neural network which predicts the brand of the car

**Fig 1: Diagram of the brand prediction process our model will follow**

**Model:** the first model tested was a basic 3-layers CNN with a Relu Activation and a fully connected layer with an output of 50, corresponding to our 50 different brands. This model resulted in a low validation accuracy of 12.3%. Afterwards a pretrained resnet model with 34 layers was implemented. This model makes use of 34 convolutional layers, with a batch normalization in between each. This model uses a Relu activation with fully connected final layers to predict the car brand.

****

**Fig 2: Diagram of our machine learning model**

**3.0 Background & Related Work**

Similar projects have been completed in the past for example, the Vehicle Make and Model Recognition (VMMR) System at the University of Regina[5]. This project was meant to be useful in the field of automatic vehicle surveillance, traffic management; the goal is to take variations in illuminations and weather, occlusions, shadows, and reflections into account when recognizing the vehicle. Another similar project is the Car Make and Model Recognition project completed by Hashir Yaqoob[6]. This project is also geared towards the traffic monitoring and traffic surveillance systems–however it recognizes and classifies cars in real time.

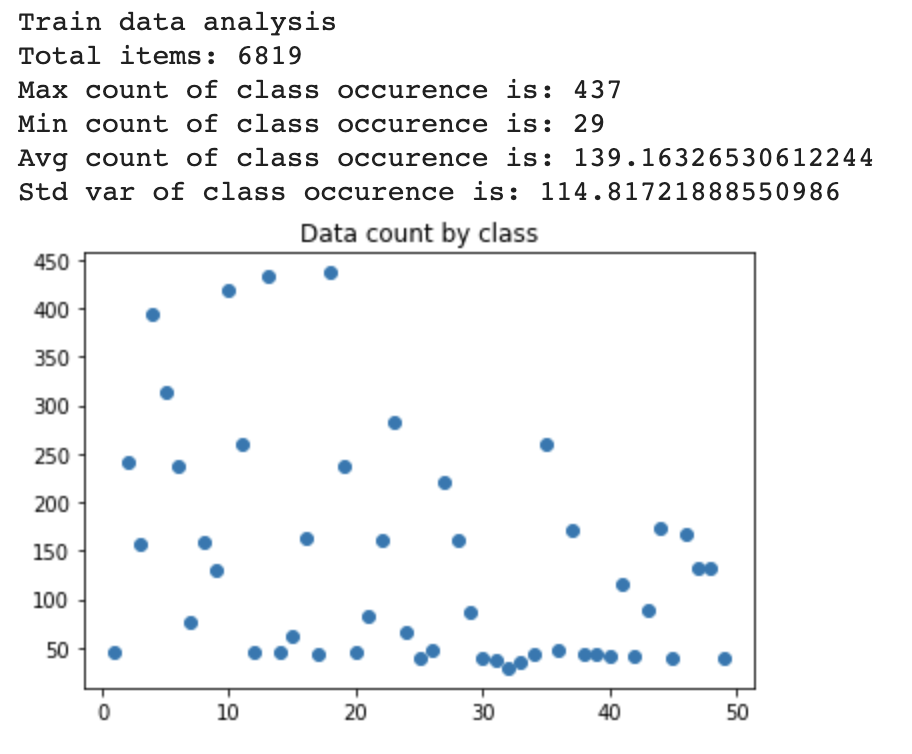
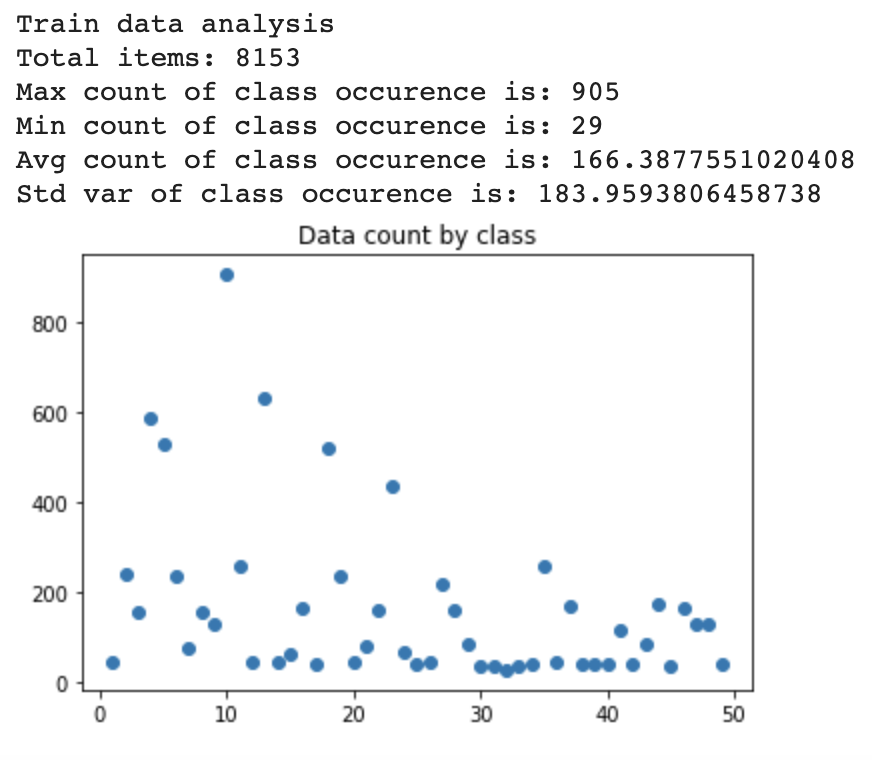
**4.0 Data Processing**

The most important part of an image recognition model is collecting sufficient data that we can train the model on. Through careful consideration, the dataset chosen was the ‘Car Dataset’[7] provided by Stanford University. This dataset has over 16,000 images with 196 car classes. A ‘class’ is defined as: **‘Car Brand + Model + Model Year’.** However, for the purpose of this project, we classified the cars based on only the **‘Car Brand’**. This reduced the number of classes to 50 from 196, providing more images per class, allowing for better results.

Data processing is comprised of two main sub tasks outlined below:

* *Data loading*

We reduced the number of classes from 196 to 50. We did that using a descriptor file that contained the class names for all original labels. To reduce the classes, unique brand names were extracted from that file and stored in a data structure.

****

**Fig 3, 4: Analysing class distribution before (left) and after (right) applying our data processing methodology**

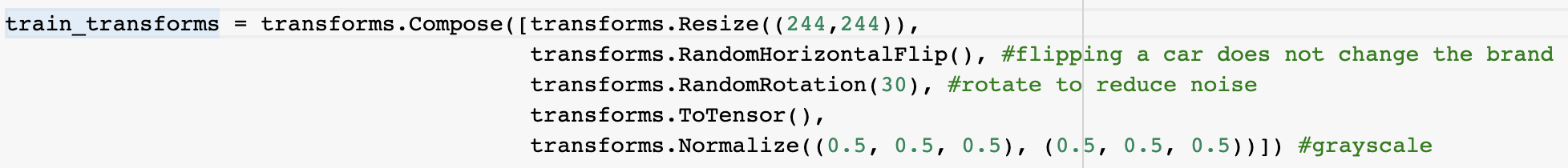
This change impacted the even count distribution of our data, as seen on the left. There were many ways to tackle this new issue, either duplicate images for all low count classes or getting rid of a few images of classes with high count.

We removed images of high count classes, as there were fewer classes with high counts as you can see on the diagram. Making that adjustment resulted in a distribution as seen on the right. We did not duplicate images of low counts as the first change resulted in a good performance overall, which will be discussed later on.

* *Data Augmentation*

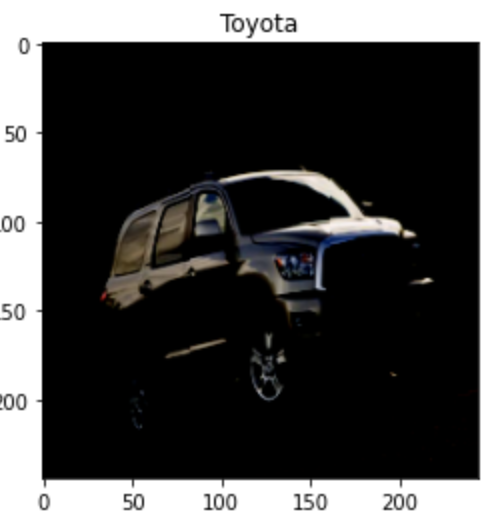
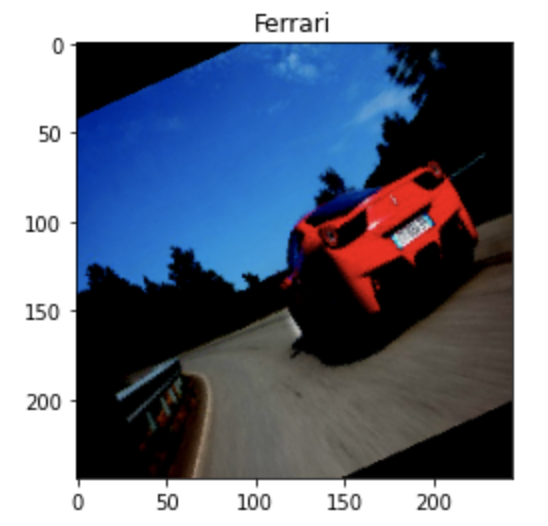
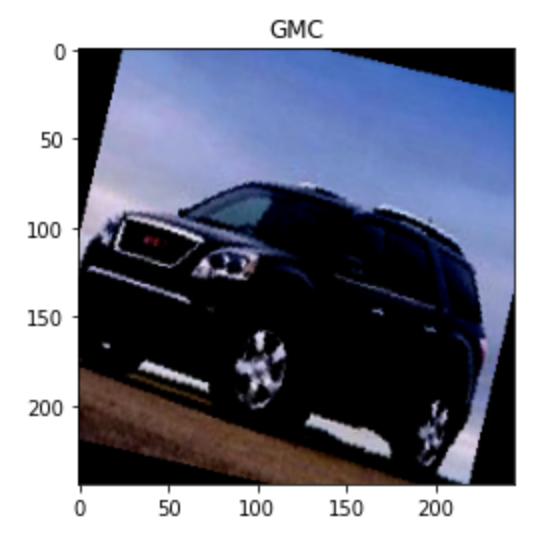
Given the nature of the problem, a few manipulations had to be done to the training data to make the model more robust to noise. These changes do not affect the car brand, which is why they were important:

* Color invariance
* Randomly flipping the image horizontally
* Randomly rotating in both directions by small angles



**Fig 5: Data augmentation code**

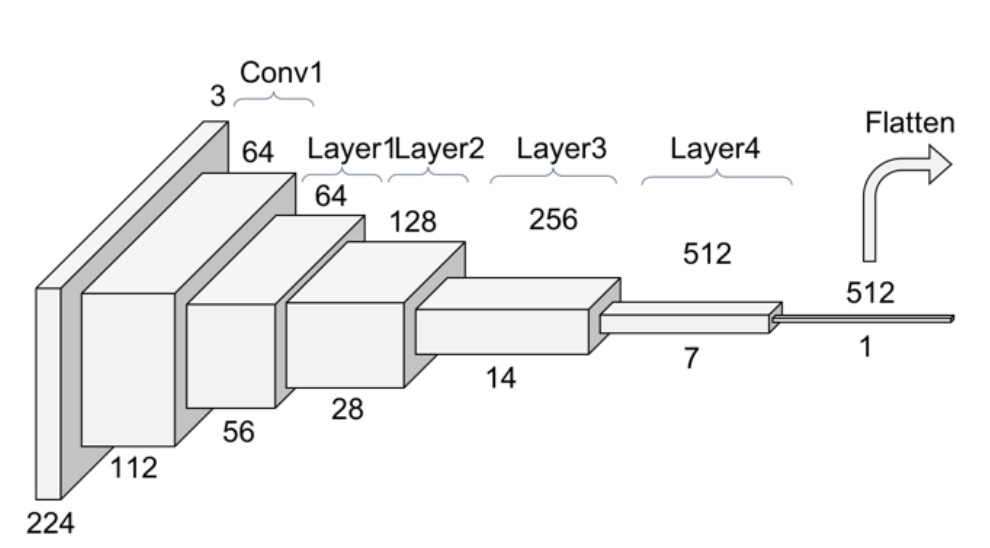
The manipulations as discussed were integrated in the *transforms* for the *train data loader* object as seen in the code above. The result transformation is seen in the images below, which has removed a lot of noise, rotated the images randomly. The images are not completely black & white, but still better than the original dataset



**Fig 6, 7, 8: Examples from our dataset after our data augmentation methodology was applied**

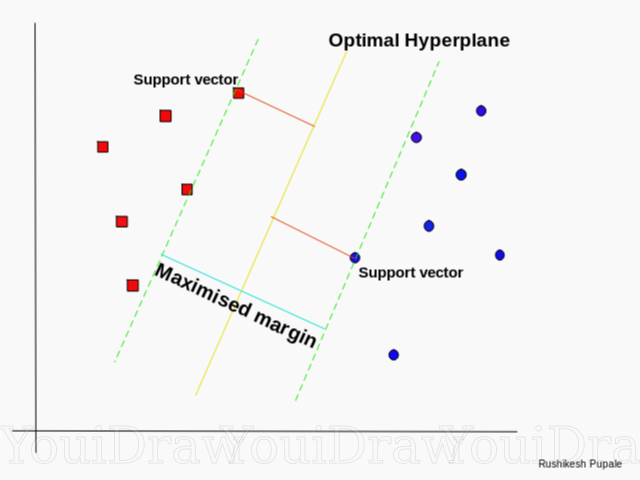
**5.0 Architecture**

The model chosen is a pretrained resnet34 model. This model has 34 convolutional layers divided between four sequential layers. The first sequential layer takes in an input of size 3, and has an output of size 64. The first sequential layer normalizes the batch in between each convolutional layer. The second sequential layer takens in an input of 64 and has an output of 128, and similar to the first layer, performs batch normalization in between each layer. The third and fourth layers take in an input of size 128 and 256 and have outputs of size 256 and 512 respectively. Both layers normalize the batch in between the convolutional layers. All of the convolution layers have a kernel size of 3x3. This model has a relu activation function and an adaptive average pooling. There is also a fully connected layer that has an input of size 512 and an output of 50, corresponding to the number of classes in our dataset.



**Fig 9: Architecture present in our pretrained resnet34 model [8]**

**6.0 Baseline Model**

****

**Fig 10: Illustration of SVM Model Hyperplane Search [5]**

To ensure that the accuracy in our model cannot be easily replicated by simpler methods we compare the results of our model to that of a baseline. For this baseline we select a Support-vector machine (SVM) model because it is intuitive, a low amount of hyper parameters, and has a high level of success when applied to classification problems [5].

The goal of an SVM Model is to find the optimal hyperplane which separates each class from each other by constructing support vectors from data points, selecting hyperplanes when support vector length is maximized.

Our SVM baseline model resizes all images to 244 \* 244 and condenses all 3 RGB colour channels into one. From this we then extract 150 feature components and then train the model, using a grid search to determine optimal hyper parameters.

With this model we were able to achieve an accuracy of only 8% [Appendix C]. We determine this is because of the large number of classes (49) and the large amount of similarities between the classes interfering with the models ability to create effective Hyperplanes. (Toyota vs Honda is very similar, compared to Apple vs Bicycle where SVM would be more effective).

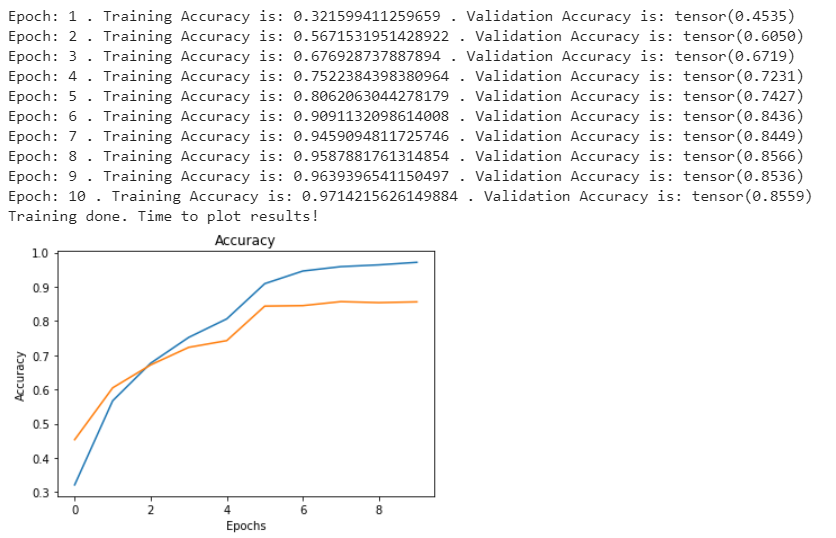
Although this model has low accuracy, it is a more effective baseline than other models because replacing it with an ANN or CNN would be time intensive, and would likely result in overfitting to the data set (which would skew the effectiveness of our primary model in comparison).

**7.0 Quantitative Results**

After the model architecture was decided hyperparameter tuning was carried out. As our model is a pre-trained Resnet, changing the model parameters was not required. In the first 3 attempts, the learning rate and the number of epochs were kept constant, with the batchsize being modified (values tested: 512, 128, 64, 32). The best result was obtained with a batch size of 32. For the fifth attempt, the learning rate was increased to 0.01, the batch size set to 32 and the number of epochs remained constant at 10. This attempt resulted in the highest accuracy of 85.59%.

| Quantitative | Best Training Accuracy: **97.14%** (batch\_size = 32, lr = 0.01)  Worst Training Accuracy: **88.47%** (batch\_size = 64, lr = 0.001)  Best Validation Accuracy: **85.59%** (batch\_size = 32, lr = 0.01)  Worst Validation Accuracy: **72.15%** (batch\_size = 64, lr = 0.001)  The training and Validation curves for each of the five models can be found in Appendix-B |
| --- | --- |

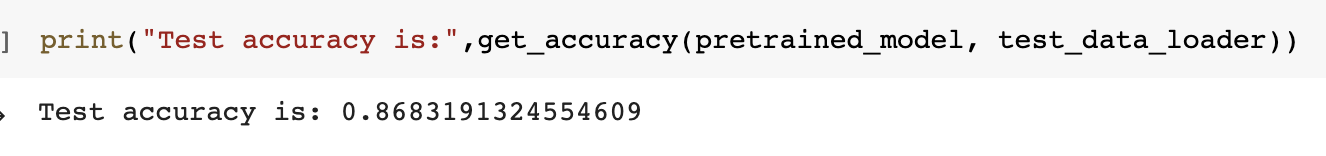
**Table 11: Hyperparameter tuning key data points**



**Figure 12**: **Fifth Attempt**

**Epochs = 10, Batch Size = 32, Learning Rate = 0.01**

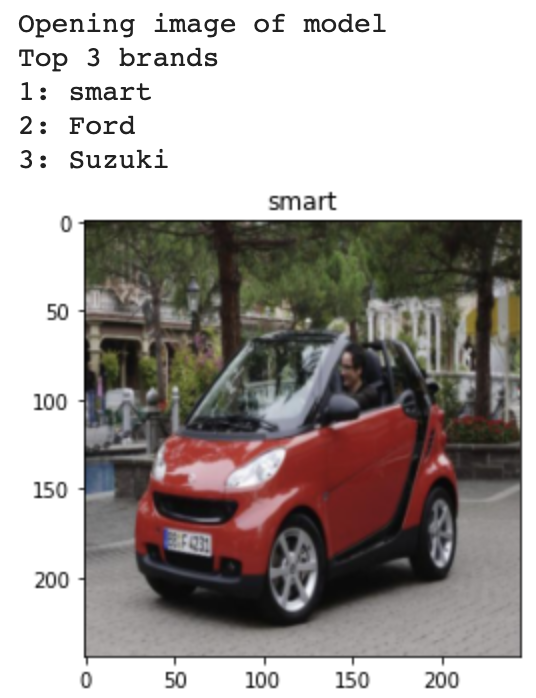
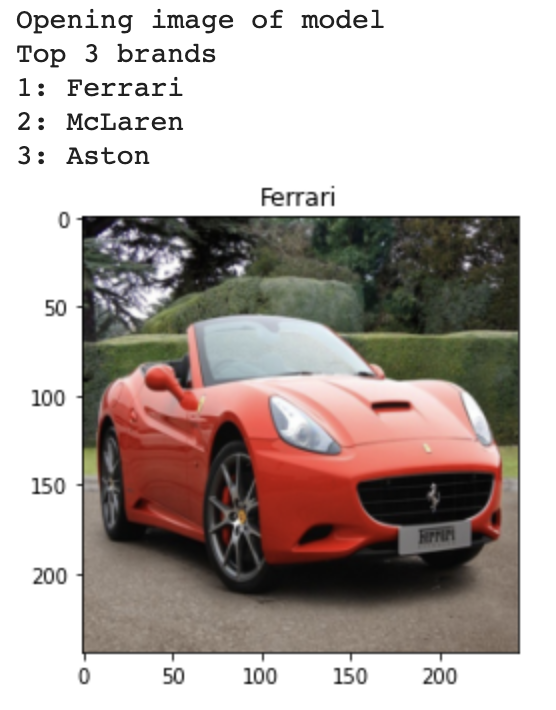
From the figure above, it can be seen that the validation and training curve for the fifth set of hyperparameters stabilizes itself around 6 epochs, therefore increasing the number of epochs was unnecessary. Since the fifth model resulted in the highest accuracy of 85.59% it was chosen as our final model. To ensure that our model was generalizing for all car brands, we tested our final model on a new testing dataset containing over 6000 car images. Our model attained an accuracy of 86.83% on this testing set.

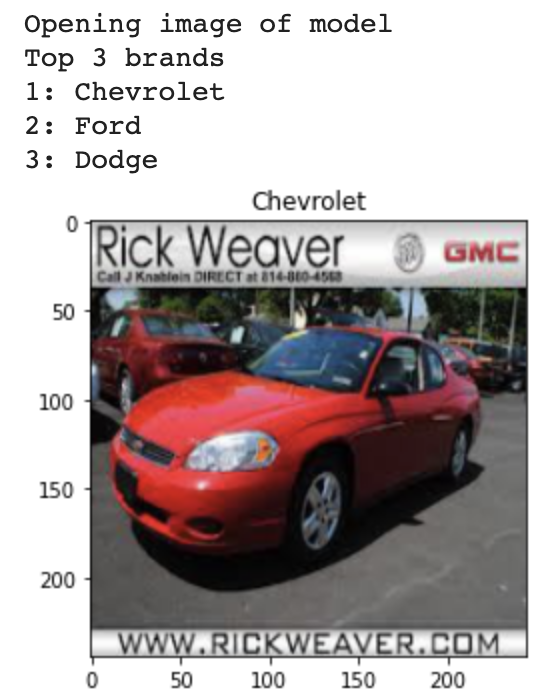
****

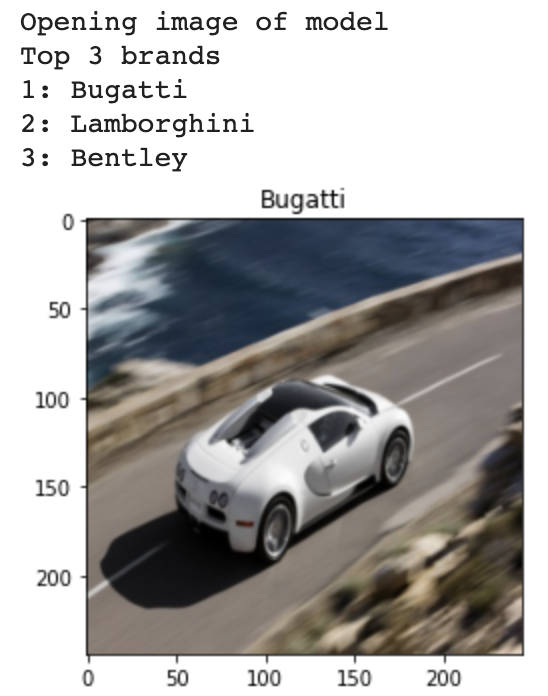
**Figure 13: Final model results on new testing data**

**8.0 Qualitative Results**

For model attempts 2-5 (Appendix B) it can be seen that the training accuracy as well as the validation accuracy plateau around 8 epochs. This shows that the model is not **overfitting** or **underfitting** to the training dataset. A few images taken from different classes from the test data set and were used as input, for visualization purposes. The 4 results can be seen below, where the model’s top 3 predictions are printed, whereas the true result was added to the image’s title. Further evaluation is continued in section 10.





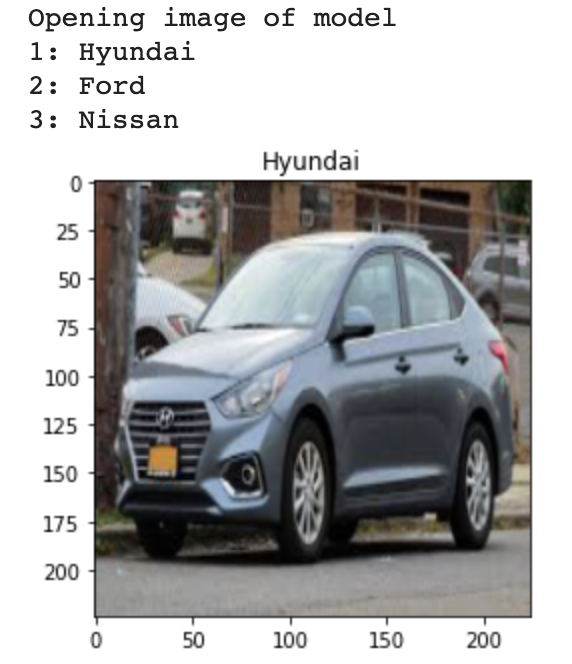


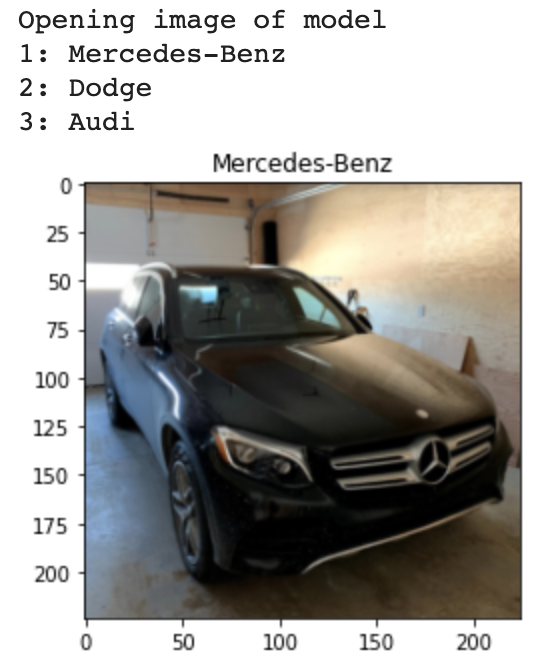
**Figure 14, 15, 16, 17: Car Image with our Model’s top 3 predictions**

**9.0 Model Evaluation on New Data**

The Stanford University ‘Car Dataset’ we used to train, validate and test our model has images of cars from before 2012[7]. Most of these car models have now been discontinued. Secondly, we did not use online images of cars from before 2012 since there was no way of verifying if these images appeared in our training data. However, it was still possible to test our model, since we had taken a general approach towards classifying the cars, by just their make and not their model or year.

The team decided to test the model on recent car pictures. For this purpose, we took pictures of cars on the street and also some online images. We were able to find a set of around 30 new images, never seen before by our model, but belonging to the brands that our model was trained to classify.





**Figure 18, 19, 20, 21: Car Image (New Data) with our Model’s top 3 predictions**

The model correctly classified 26 images, 3 of which are shown above. A deeper look into this behaviour of the model can be described by the frequency at which companies change their car designs. High-end luxury cars such as Mercedes do not change their design often, but when they do they are still pretty similar to old designs. Additionally, core design across models for such cars are also similar. On the other hand, more economical brands/models like Chevrolet and Honda are quick to change from model-to-model and year-to-year. This explains why the model performed well on high end brands and the 4 ones it got wrong were more economical brands (as in figure 21).

**10.0 Discussion of Results**

In section 8.0 (Qualitative Results), the model was passed in images of different brands. The selection of brands was crucial. As discussed in data processing, classes were not split evenly. So, to evaluate the model’s performance over different classes with different class-counts becomes important, which is why Chevrolet (most populated), Smart (least populated) and Bugatti & Ferrari (mediumly populated) were chosen. Despite these differences, the model correctly predicted 11/12 images (3 per brand). Although this is not a high number, for the sake of visualization it certainly helps.

The model has also picked up features/traits across similar models/brands. For example, while predicting Bugatti or Ferrari, the model predicts them correctly. But notice how the model’s top 3 predictions also mention other brands such as McLaren, Aston Martin, Lamborghini and Bentley. These brands are also into manufacturing spots/luxury cars. This indicates that while training, the model was able to recognize similar traits/designs across cars/brands. A similar pattern can be seen in Smart & Chevrolet, and also on the results for the new data.

**11.0 Ethical Considerations**

The dataset we are utilizing is the Stanford Cars dataset. Although the dataset has seen widespread use, it still contains legible license plates. This is a potential ethical issue as license plates can be considered personally identifiable information [6] and therefore we must ensure we are not displaying this information freely.

Another ethical consideration comes with the possible use cases for our model. Technology that is capable of identifying car brand and colour from images is a privacy issue as it can be utilized by nefarious actors to track and identify the locations of individuals without their consent. Therefore, when we apply this technology in real world scenarios, we must ensure we have the explicit permission of the vehicle owners to take a photo of their car and apply our machine learning algorithm.

Both of the aforementioned ethical issues fall under the ethical category of privacy and data governance [9] which states that data collected by AI systems should be secure and private. Throughout the course of creating this model we continue to strive to uphold this standard.

**12.0** **Project Difficulty**

Though the Stanford University ‘Car Dataset’ had a large number and variety of images, not all images were ideal for training the model. Below, are examples of two types of images that made it hard for the model to learn and predict the car brand accurately.



**Fig 22: Text Present in Car Image Fig 23: Car is not a Subject of the Image**

As shown in Figure 22, many images in the dataset had text present along with a photo of the car. Also, the car present in the image is actually a Volkswagen, however there are two Ford logos present at the top. This poses further challenges. In Figure 23, the car is not present as the subject of the image. Additionally, there are a variety of features (such as the tent, trees, mountains, clouds) in the backdrop. Such background noise present in the images makes it harder for the model to correctly predict the car make.

For the reasons provided above, the car classification problem is harder than it seems at first. These reasons are probably why a simple CNN proved ineffective and a deeper network such as ResNet34 was required to achieve a higher accuracy and better results.

**13.0 References**

**[1]** J. Young, J. Risley, J. Young, J. Risley, and F. Ali, “US ecommerce sales grow 14.9% in 2019,” Digital Commerce 360. [Online]. Available: https://www.digitalcommerce360.com/article/us-ecommerce-sales/ . [Accessed: 21-Feb-2020].

**[2]** M. Brohan, “How ecommerce will change automotive retail,” Digital Commerce 360, 19-Nov-2019. [Online]. Available: https://www.digitalcommerce360.com/2019/11/19/how-ecommerce-will-change-automotive-retail/ . [Accessed: 21-Feb-2020].

**[3]** K. A. Barchard and L. A. Pace, “Preventing human error: The impact of data entry methods on data accuracy and statistical results,” Computers in Human Behavior, vol. 27, no. 5, pp. 1834–1839, 2011.

**[4]** G. Seif, “Deep Learning for Image Recognition: why it's challenging, where we've been, and what's next,” Medium, 04-May-2019. [Online]. Available: https://towardsdatascience.com/deep-learning-for-image-classification-why-its-challenging-where-we-ve-been-and-what-s-next-93b56948fcef . [Accessed: 23-Feb-2020].

**[5]** R. Gandhi, “Support Vector Machine - Introduction to Machine Learning Algorithms,” Medium, 05-Jul-2018. [Online]. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47. [Accessed: 05-Apr-2020].

**[6]** G. Zvulony, “Gil Zvulony,” Zvulony & Co., 07-Jul-2015. [Online]. Available: https://zvulony.ca/2011/articles/privacy/privacy-law-license-plates/. [Accessed: 05-Apr-2020].

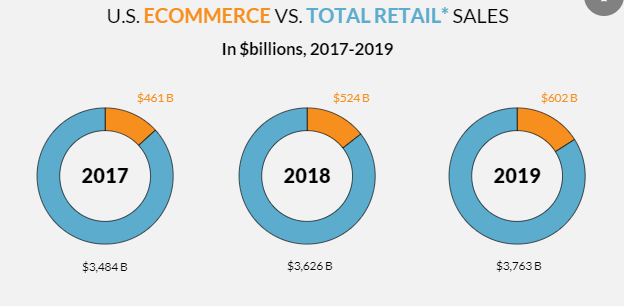
**[7]** 3D Object Representations for Fine-Grained Categorization Jonathan Krause, Michael Stark, Jia Deng, Li Fei-Fei 4th IEEE Workshop on 3D Representation and Recognition, at ICCV 2013 (3dRR-13). Sydney, Australia. Dec. 8, 2013. Available: https://ai.stanford.edu/~jkrause/cars/car\_dataset.html

**[8]**"Understanding and visualizing ResNets", Medium, 2020. [Online]. Available: https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8. [Accessed: 14- Apr- 2020].

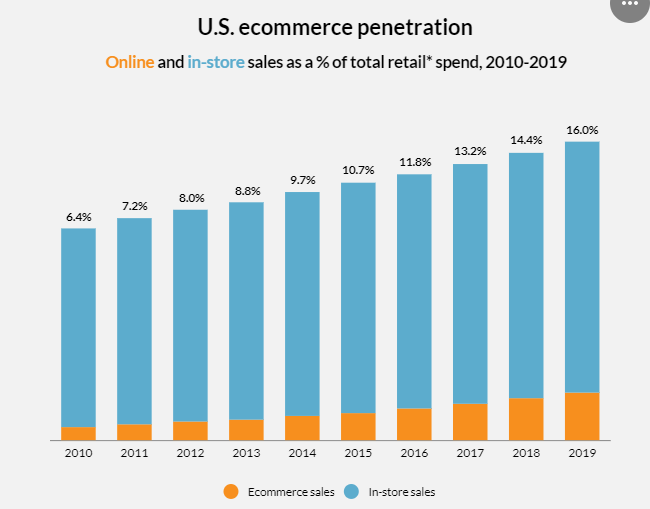
**[9]** J. Vincent, “AI systems should be accountable, explainable, and unbiased, says EU,” The Verge, 08-Apr-2019. [Online]. Available: https://www.theverge.com/2019/4/8/18300149/eu-artificial-intelligence-ai-ethical-guidelines-recommendations. [Accessed: 09-Apr-2020].

**14.0 Appendices**

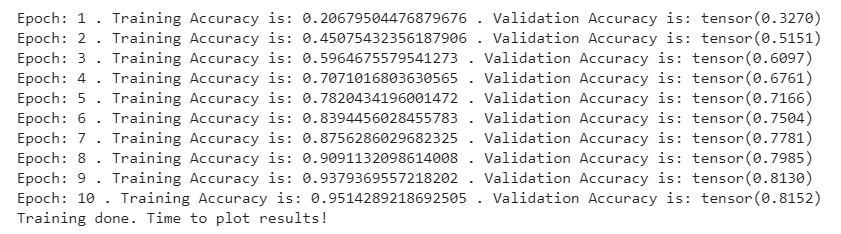
Appendix **A**: Graphs showing a rise in e-commerce sale in the United States

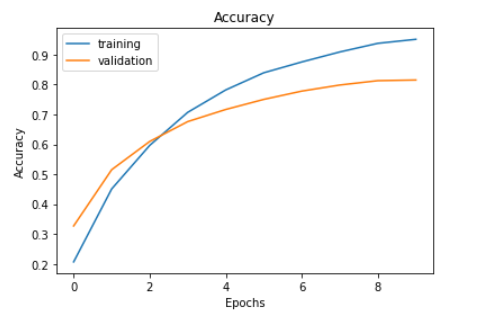


**Fig A.1: Ecommerce vs Total Retail Sales (USA) 2017 - 2019 [1]**

**Fig A.2: U.S. Ecommerce Penetration (2010 - 2019) [1]**

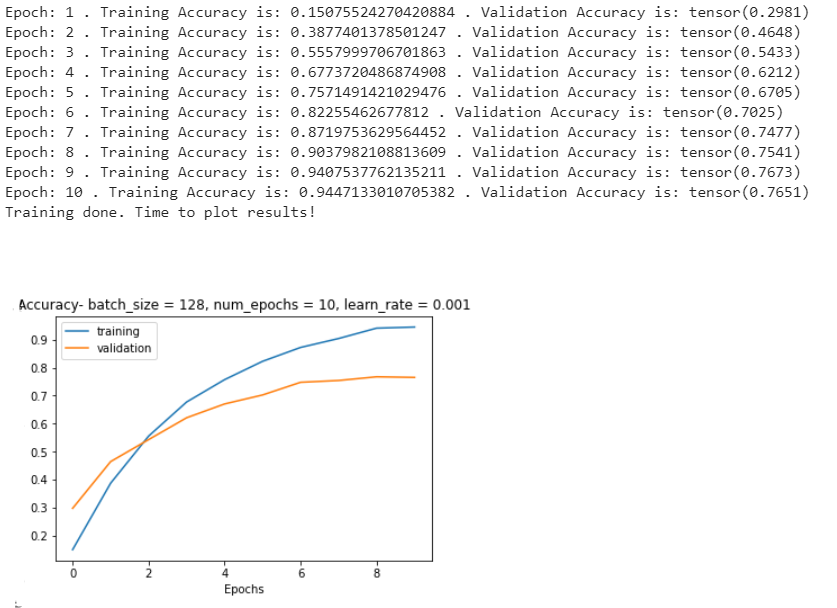
Appendix **B:** Some accuracy curves for the hyperparameters we tested





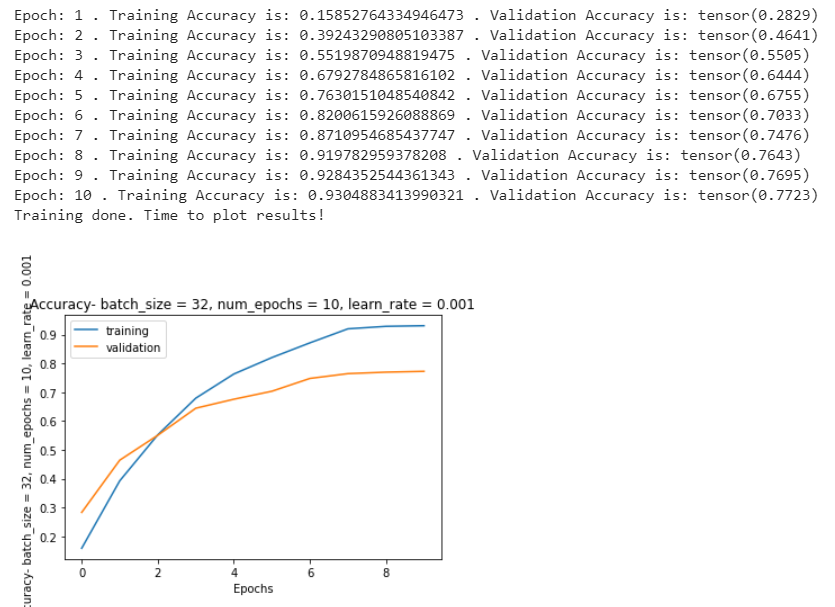
**Fig B.1: Accuracy of our Model w/ HyperParameters**

**(Batch\_size = 51, Learning Rate = 0.001,Num\_Epoch = 10)**



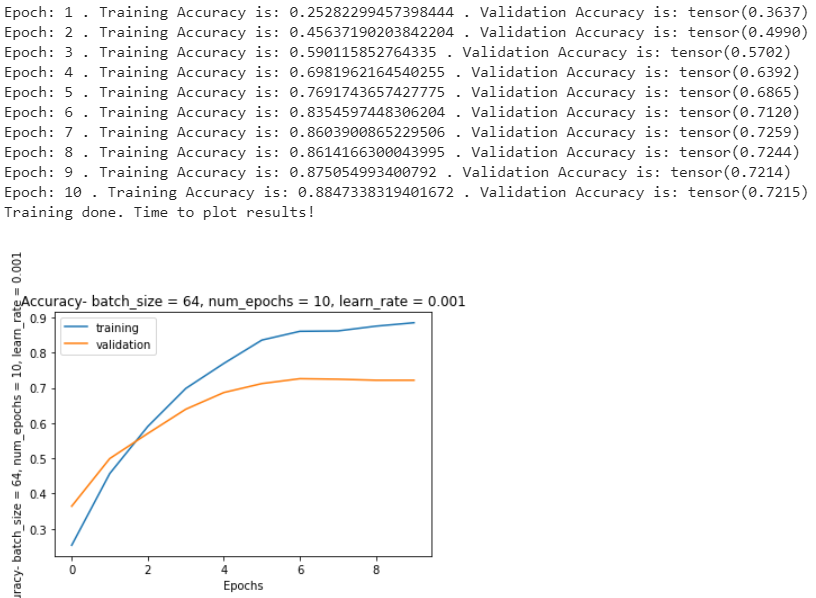
**Fig B.2: Accuracy of our Model w/ HyperParameters**

**(Batch\_size = 128, Learning Rate = 0.001,Num\_Epoch = 10)**



**Fig B.3: Accuracy of our Model w/ HyperParameters**

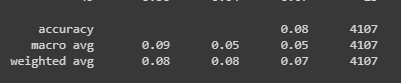
**(Batch\_size = 32, Learning Rate = 0.001,Num\_Epoch = 10)**



**Fig B.4: Accuracy of our Model w/ HyperParameters**

**(Batch\_size = 64, Learning Rate = 0.001,Num\_Epoch = 10)**

Appendix **C**: Baseline model details



**Fig C.1: Accuracy of The Baseline Model**