Fine-grained Car Classification Using Transfer Learning

Group 7: Hala Abualsaud, Gates Zeng, Jialu Zhao

***Abstract* - This paper is aiming at improving car model classification tasks by evaluating several models to generate a high performance model that is faster and with high accuracy. We compare these results to that of baseline models and discuss the advantages and disadvantages of each model. We also try to improve the best model. We use the Stanford Cars Dataset.**

I. INTRODUCTION

CAR CLASSIFICATION is of crucial importance for public safety, in that it directly influences traffic monitoring, management, and control. Moreover, it's important for the field of intelligent transportation systems (ITSs) that cares about transport system automation. Recently, with the evolution of many computer vision and machine learning techniques, researchers have shown great interest in car classification problems [1].

Human eyes can differentiate between car types from several key features such as bodywork, hoods, logos, and other very fine details. To recognize these subtle differences from a car image is usually a very complex and challenging task for a computer. Several classical approaches have attempted to tackle this classification problem by using fine-grained classification (i.e. classification with extremely similar categories), a subfield of object recognition [2]. This type of classification is a very popular approach especially in classification of bird species. Moreover, scientists approached this problem by creating very high-quality and large datasets. However, creating and handling very large datasets is a time consuming and expensive approach in terms of computational power and data collection [3].

We use a deep learning framework to pre-train classifiers and we take advantage of transfer learning to make the learning process faster.

The input to our algorithm is a car image. We then use several machine learning models to output a predicted car model. Moreover, to increase the accuracy for our model, we try tuning learning rate, adding additional fully-connected layers, and adding dropout.

We conduct experiments on the car dataset [4], a dataset of various vehicle makes and models.

II. RELATED WORK

Stationary and real-time car classification methods have been studied. Scientists have used CNNs (convolutional neural networks) to learn features of the cars. They have also used transfer learning to benefit from the use of a large, pretrained network and to minimize overfitting. Fine-grained classification has also been used in car classification to enhance the accuracy by using fewer classifiers.

1) **CNN**: Widely used in the image classification field. It takes an input image, processes it and classifies it with some given categories [5,6]. The network learns salient features of the image that can then be used for classification.
2) **Transfer Learning**: This method allows the construction of deep neural networks, while using a limited dataset. It's effective in boosting accuracy of a deep neural network as the weights for the low-level features in the network have already been trained [7]. Furthermore, training time can also be reduced as the training task just involves fine-tuning the higher layers of the network.
3) **Fine-Grained Classification**: This type of classification focuses on distinguishing between visually similar object classes, such

as species of flowers or birds [8,9]. It also can be used for vehicle classifications of make, model, and even year. This is a very challenging task since it typically requires a very large dataset. Vehicles that differ in type but have similar viewpoints may look more similar than vehicles that differ in viewpoints but are actually of the same type.

III. DATASETS AND FEATURES

We used Stanford Cars Dataset: The dataset consists of 16,185 images with 196 classes of cars. We extracted 8,144 training images and used an 80:20 split (6,515 for training, 1,629 for validation). In this project, only the training images from the dataset were used for training and validation. The test set did not include labels and therefore some test images were used at the end only for visualizations and qualitative analysis.

We used some reference code to jump start training our models [10,11].

IV. METHODS

After investigating this car classification problem, we narrowed down our models to the following because they achieved relatively good performance in other experiments.

- AlexNet
- ResNet
  - ResNet34
  - ResNet50
  - ResNet101
  - ResNet152
- ResNeXt
  - ResNeXt50_32x4d

AlexNet was a breakthrough work in the ImageNet challenge. It used five convolutional layers, several max-pooling layers and three fully connected layers. ReLU was used as the activation function. [13]

Deeper networks tend to help with reducing overfitting and potentially reducing the overall network size [12]. ResNet was a significantly deeper network than AlexNet. Additionally, ResNet solved the vanishing gradient problem by adding the idea of “residual nodes”. The PyTorch framework provides a series of ResNet models and as the model number increases, the model is deeper. We intend to find out the relationship between model depth with its performance in our car classification task.

ResNeXt is a special case of ResNet50. In this case, the outputs of different paths are merged by adding them together rather than being depth-concatenated in ResNet. [12] We performed an experiment to analyze the difference between them.

On top of those pre-trained models, we added one fully connected layer and a softmax layer at the end in order to perform final classification.

Another experiment involved adding more fully connected layers. This was performed in order to understand how increasing model depth by using individual fully connected layers can affect the performance.

V. EXPERIMENTS AND RESULTS

Experimental Parameters:

- Optimizer: SGD
  - Momentum: 0.9
  - Decay: 1e-6
  - Nesterov: True
- Loss: Cross Entropy Loss
- Batch Size: 16

Table 1 shows the results of different models:

Model	**Training Accuracy**	**Validation Accuracy**
ResNet-34	92%	82%
ResNet-50	90%	81%
ResNet-101	89%	81%
ResNet-152	87%	80%
AlexNet	99%	62%

ResNeXt50_32x4d	96%	85%

Table 1: Models with training/ validation accuracy

After comparing the various pre-trained models, we explored changing the network architecture.

First, we compared the model performances between ResNet50 with one fully connected layer to that of two fully connected layers. The results from this experiments are as Table 2:

	Adding one FC	**Adding two FC**
Training Accuracy	92%	49%
Validation Accuracy	82%	44%

Table 2: ResNet50 adding one/two fully connected layers

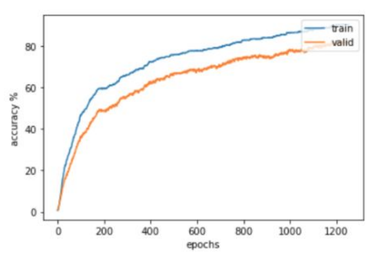

Figure 1: Resnet 50 with adding one fully connected layer

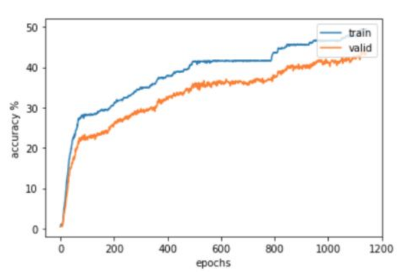

Figure 2: Resnet 50 with adding two fully connected layer

From the results above, we can see that adding one more fully connected layer significantly dropped the accuracy of the model in both training and validation cases. We realized that two fully connected layers have substantially more parameters to train than one FC layer (i.e. dataset size is relatively small for two FC layers) the model performance does not improve with more FC layers. Thus for the rest of our experiments, we only added one fully connected layer to our models.

Secondly, we noticed that AlexNet had high generalization error where the model reported 99% training accuracy but only 62% validation accuracy.

We tried to reduce the generalization error by adding a 30% dropout layer, Table 3 shows the result with the dropout layer and without the dropout layer:

	W/O dropout	**30% dropout**
Training Accuracy	99%	70%
Validation Accuracy	62%	62%

Table 3: Alexnet with without/ with 30% dropout

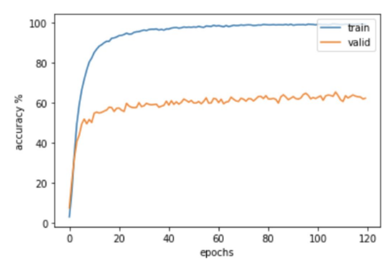

Figure 3: Alexnet w/o dropout

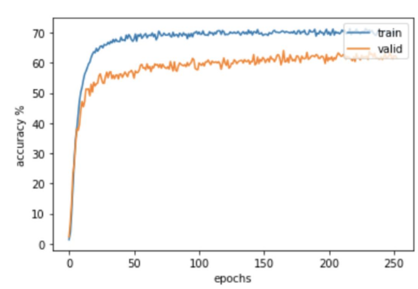

Figure 4: Alexnet w/ 30% dropout

Adding a 30% dropout layer reduced the generalization error while the validation accuracy for AlexNet was still low compared to other models.

Moreover, we also made a comparison between ResNet50 and ResNeXt50 models and as the result indicates (Table 3), ResNeXt-50 performs slightly better than ResNet-50.

	ResNet50	**ResNeXt50**
Training Accuracy	92%	95%
Validation Accuracy	82%	85%

Table 4: comparison between ResNet50 & ResNeXt50

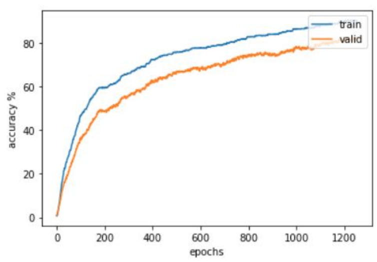

Figure 5: ResNet50

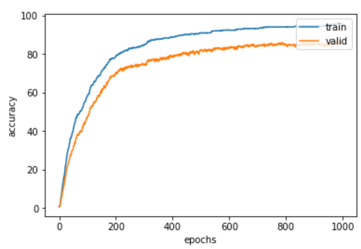

Figure 6: ResNeXt50

The last step was to analyze the relationship between model depth and model performance. We plotted training accuracy and validation accuracy for those models: ResNet34, ResNet50 and ResNet152 in the same graph for a better visualization as shown in Figure 7.

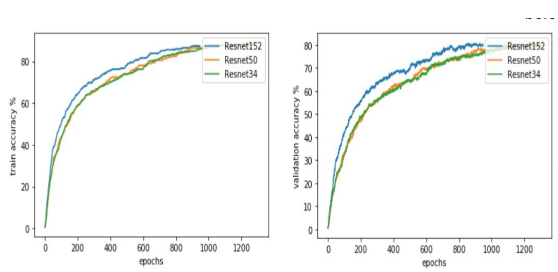

Figure 7: ResNet34, ResNet50, ResNet152 training and validation accuracy comparison

We noticed that as the model becomes deeper, it converges faster.
In addition, we created an application that took a vehicle image and gave the top 5 predictions of the car model with its confidence score. We also plotted the prediction results and their confidence scores in a bar chart for a better visualization.

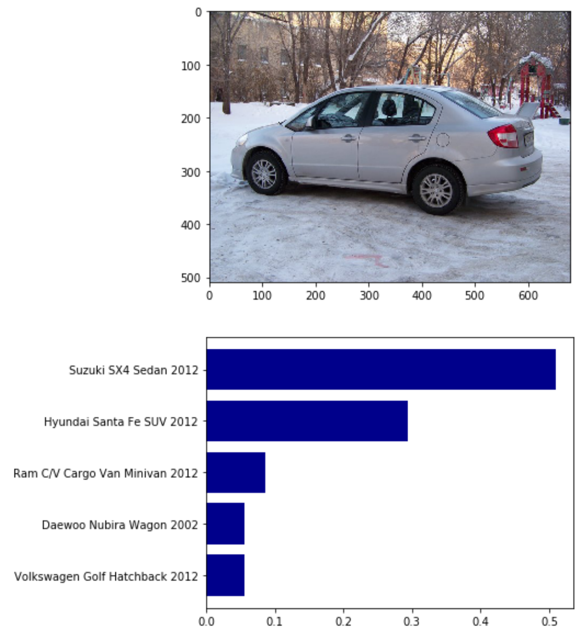

Figure 8: Application Example

V. CONCLUSION AND FUTURE WORK

We concluded that the AlexNet model had the lowest accuracy compared to other models we trained. Moreover, we realized that having more fully connected layers was not beneficial at improving our model. Dropout contributed effectively at reducing the bad generalization error in the AlexNet model. Additionally, we noticed that deeper ResNet networks converge faster. Lastly, we saw that for our car classification task, ResNeXt50 performed better than ResNet50. For future work, we believe that by adding other networks (i.e. apply an object detector or a salient detector before the classification network) and also data augmentation (i.e. grayscale the image for better recognition) could improve the accuracy of our model further.

VI. CONTRIBUTIONS

Hala Abualsaud: Summarized and organized literature, trained Resnet34, Resnet50, added dropout layers to different models.

Gates Zeng: Researched related literature, trained ResneXt50, created an application that can recognize the car model from testing images and visualize it by graph .

Jialu Zhao: Data preprocess, Data Visualization, Trained ResNet101, ResNet152 and AlexNet.

VII. REFERENCES/BIBLIOGRAPHY

[1] "Monza: Image Classification of Vehicle Make and Model ..." [Online]. Available: http://cs231n.stanford.edu/reports/2015/pdfs/lediurfinal.pdf . [Accessed: 27-May-2020].

[2] M. Hoser, "Modeling Adaptive Learning Agents for Domain Knowledge Transfer," *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2019.

[3] A. Ambardekar, M. Nicolescu, G. Bebis, and M. Nicolescu, "Vehicle classification framework: a comparative study," *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, Oct. 2014.

[4] "Cars Dataset," *Stanford Artificial Intelligence Laboratory*. [Online]. Available: https://ai.stanford.edu/~jkrause/cars/car_dataset.html. [Accessed: 27-May-2020].

[5] Krizhevsky, A., Sutskever, I. and Hinton, G.: ImageNet Classification with Deep Convolutional Networks. In: NIPS 2012

[6] B. Satar and A. E. Dirik, "Deep Learning Based Vehicle Make-Model Classification," *Artificial Neural Networks and Machine Learning – ICANN 2018 Lecture Notes in Computer Science*, pp. 544–553, 2018.

[7] Pan, S. and Qiang, Y.: A Survey on Transfer Learning. In: IEEE Transactions on Knowledge and Data Engineering 2010

[8] Angelova, A. and Zhu, S.: Efficient object detection and segmentation for fine-grained recognition. In: CVPR 2013

[9] Yang, Linjie, Luo, Ping, Loy, C. Change, Tang, and Xiaoou, "A Large-Scale Car Dataset for Fine-Grained Categorization and Verification," *arXiv.org*, 24-Sep-2015. [Online]. Available: https://arxiv.org/abs/1506.08959. [Accessed: 13-Jun-2020].

[10] Wengsengh, "wengsengh/Car-Models-Classifier," *GitHub*. [Online].Available:https://github.com/wengsengh/Car-Models-Classifier/blob/master/car_models_classifier.ipynb. [Accessed: 27-May-2020].

[11] Foamliu, "foamliu/Car-Recognition-PyTorch," *GitHub*, 09-Jan-2020. [Online]. Available: https://github.com/foamliu/Car-Recognition-PyTorch. [Accessed: 04-Jun-2020].

[12] V. Fung, "An Overview of ResNet and its Variants," Medium, 17-Jul-2017. [Online]. Available: https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035. [Accessed: 13-Jun-2020].

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.

Critique of group 7 – Fine-grained car classification using Transfer Learning
--Critiques by group 28

This presentation clearly explained the problem they intend to solve, dataset, they use, models they apply to solve the problem, insightful results.

Good parts:

1. They use diagram to explain the structure of their models, and comparison between similar models. The use of diagram helps me to understand the models more clearly.
Thanks!

2. The background of the project was explained thoroughly, and they also added the application page talking about how their project can be utilized in real world.
Thanks!

3. The style of the slides is very consistent.
Thanks!

Improvement suggestions:

1. The experiment on dropout only applied to AlexNet. Although the conclusion seems correspondent with the experiment results, I am curious will the results be same if dropout is applied to other models. In other words, how can we confirm dropout will improve the model generality when it was only tested on AlexNet?
Other models don't have generalization problems, so we didn't apply dropout to them because we believe even if we apply the dropout layer, we can't see a clear improvement on other models.

2. Similar concern for the experiment with adding more fully connected layer.
This is a valid concern. Due to the time restrictions, we were only able to test on only one of the models and extrapolate from that.

3. In the dataset, it was mentioned that there are were test data, but in the experiment results, they seem to forget mentioning test accuracy.
The dataset we found did not include labels for the test data. Thus, we were not able to evaluate the test accuracy and only reported the train and validation accuracy.

4. They did not explain the reason behind the parameters they chose.
We just tried different parameters and picked the best parameters.

5. Instead of stating the number of images for training and validation, I think it is clearer to show the percentages because a "80%" number seems easier to read than "8144 over 16185".
Thanks for this suggestion. We changed it in our report.

6. They explained the distribution of the dataset and it is clear one type of the car has significantly more data than the others. Would that affect the results? Will it make the results less reliable?
Yes, I think it would affect the results and in the future, we would try to balance out the dataset.

Critique of group 7 presentation - Fine-grained Car Classification using Transfer Learning

Critiques by group39.

This presentation shows clearly about training different models and shows us an impressive result of high accuracy in car classification. They demonstrated in an understandable way and easy to follow.

Some improvements/unclear:

Maybe you can try to show confusion matrixes to visualize the result?

Since we had over 196 classes, the confusion matrix would have been too hard to read to show any meaningful information. Therefore, we did not include it in the report.

Did you try adding a dropout layer in Resnet50 or Resnet152 or Resnext50?

These models don’t have a generalization problem, so we believe that adding dropout layers would not improve the generality greatly and we decided to only add dropout layers on Alexnet.

Why use SGD optimizer instead of others?

The decision was arbitrary.

Can you explain more about how do you use the test set?

We used the test set to provide some visuzaliations of car images in the end. Since we did not have test labels, we could not evaluate the accuracy of the model on the test set.

Critique of group 7 presentation - Fine-grained Car Classification using Transfer Learning
Critiques by group 11

The project experimented with several CNNs to classify car models. The experimental work seems really good and adequate for the scope and time constraint of the class. The team aspired to improve the performance of some of the models they tried, which is a good approach as they did not just seek to test and apply different models.

Some improvements and issues to address:

- From slide#3, we thought the objective of the project is to identify the type of the vehicle (private car, sedan, SUV, motorbike, .. etc) because the point of traffic management was brought up. However, it seems that the experiment and the work was on classifying the models of the cars - how does classifying cars based on make and model help with traffic control/management? Hence, we believe the presentation (and report) should be adjusted to clarify the background and motivation of the project/presentation (for e.g. this classification based on Model/Make could be used to identify cars as indicated in Amber Alerts).
  Knowing the model can help regulate traffic based on the type of cars that are on the road, for instance electric cars, hybrids or gasoline cars. If we know the model, we can dynamically setup certain lanes for more environmentally friendly vehicles during times of traffic congestion.

- It is better to compare ResNet with other advanced/modern architectures. Comparing ResNet to AlexNet might be unfair as AlexNet is from an older generation of networks than ResNet. This is also clear from the results section. I think the time and effort could be better spent by discussing how to improve the performance of ResNet rather than comparing it to AlexNet.
  You are absolutely right on this and actually the reason that we experimented on Alexnet is that we want to try different ways to improve the Alexnet and then compare with Resnet, Resnet here is more like a baseline.

- Provide justification on why you chose the architectures (more details on the model architecture: e.g. what was the level of dropout?).
  Those models that we chose have been experimented in other models and it has been shown that they have a good performance. We want to start from a good result and improve on it to get a better result.

- You talked about generalization but did not mention how the models performed on a test set, only on the training/validation.
  You're correct. We only had access to the train and validation set with labels. The test set did not have any labels to evaluate the generalization of the validation/test errors. However, we can still capture some of the generalization error by observing the performance of the generalization from train/test errors.

- In slide 16, the issue of overfitting was not fixed because only the training accuracy changed. We think the team should reconsider their approach on how to solve overfitting for this part.

  Yes, we should consider different ways to solve the overfitting problems but due to the time restrictions and we want to focus on other models and improvements, we didn't dig

into this part more.

● Important (Need to be checked): If the CNNs used have already been trained and transfer learning was used, we doubt that the training/validation accuracy will start from 0 after the 1st EPOCH (as shown in slides 15 to 18).
It actually could start from near 0. Transfer learning applied to the convolutional layers, not the fully connected layer that we used at the end for classification. While the model at the first epoch might be able to create a high-level representation of the car images after the convolutional layers, it doesn't know how to classify the images. After the first epoch, the model then has seen the labels and then we would see accuracy improving dramatically as shown in our plots.

Final Thoughts:
Overall really good work. The team tested several architectures and experimented with them. This is a really good approach especially for a class where the goal is learning and experimentation. Thank you for your presentation, which was designed beautifully, and hard work. We enjoyed reviewing it. All the best with your final report. We hope our comments and suggestions will help you with it.