Robotic Inference: Motorist Classification Using Deep-CNN

Smruti Panigrahi, Ph.D.
Udacity Robotics Nandegree

***Abstract*—In this paper a classification or a robotic inference model is provided to categorize different types of motorists on the roadways to aid in autonomous vehicle's decision making. In particular, classification of the cars, motorcycles, bicycles, and background is performed in order to distinguish one from the others. A subset of publicly available dataset (MIO-TCD) is used in order to train and test the inference model. Using a well-known CNN deep-learning model, GoogLeNet, with an Adam optimizer of learning rate 0.001, the model was trained for 16 epochs with a final validation score of 95% and test scores of 100% for background, 95% for cars, 90% for motorcycles, and 80% for bicycles.**

Keywords—Classification, Deep-CNN, LeNet, AlexNet, GoogLeNet

I. Introduction

As the autonomous vehicle (AV) race intensifies, it is imperative that the AVs be able to detect and classify various types of transport systems on the road so as to make informed decision on the vehicle control. Classifying various types of motorists would aid in the decision-making process of the central intelligence of the vehicle. Inference of objects using deep-CNN is a powerful method in computer vision. However, there are various models and parameters involved in training a model that can significantly affect the final classification performance.

Various image classification models have been developed over the years, such as LeNet[1], AlexNet[2], GoogLeNet[3]. While LeNet uses a stochastic gradient decscent optimization technique on grayscale images, the AlexNet and GooLeNet are suited for high resolution color images.

The first DNN architecture to win the ImageNet [4] classification challenge was AlexNet. Better DNNs such as ResNet, Inception and GoogLeNet have been recently developed using the ImageNet benchmark dataset. In this paper various trained models using LeNet, AlexNet, and GoogLeNet will be discussed.

Using MIOvision Traffic Camera Dataset (MIO-TCD) [5, 6], the models in this paper are trainined using the Nvidia DIGITS, an online platform for training deep learning models. It utilizes transfer learning from existing pretrained models such as LeNet, AlexNet, and GoogLeNet.

II. Data Acquisition and Model Slecetion

A. Data Acquisition

Two different datasets are trainined and evaluated. The first dataset is a supplied dataset containing images of candy boxes, bottles, and empty conveyor belt taken from a Jetson mounted over a conveyor belt. The images of are uploaded into DIGITS workspace and trained for the purpose of real time sorting.

The second dataset used in this paper is extracted from the MIOvision Traffic Camera Dataset and contains colored images taken from various angles of various types of tranportation systems. Though the dataset contains upto 11 different classes of objects, in this paper we examined only 4 classes, cars, motorcycles, bicycles, and background, in order to keep the data upload and training time reasonable.

For this robotic inference project, a phone is used to collect the dataset of images which are rectangular images at first. A system requires a constant input dimensionality. Therefore, the images are down-sampled to a fixed resolution of 256 x 256.

B. Model Selection

There are three choices of the CNN models for the classification task, namely, LeNet, AlexNet, and GoogLeNet. Since the LeNet model uses very low-resolution 28x28 images, the model was not used for training the provided dataset. Also AlexNet uses grayscale images where some of the features from the colored images could get lost. Hence, based on 256x256 color images, the GoogLeNet model was chosen for the classfication task.

C. Learning Rate

The learning rate and the choice of the optimization model is crucial for obtaining a well-trainined model. Extra care must be taken to avoid choosing a leraning rate that is too high or too low. Too high learning rate might not be able to converge to the minimum and at times might even skip the minimum and blow up. Too low learining rate might result in the training getting stuck in a local minimum. This problem can be addressed using the Adam optimizer since it dynamically adjusts the training rate as the training progresses. Total 16 epochs were chosen for the traning and validation of the model. For the training, the initial learning rate for the Adam optimiser was chosen to be 0.001 which was further reduced by the model to 0.0001 and 0.00001 within the 16 ecpochs.

D. Data Selection

A classification dataset was first created using NVIDIA DIGITS, with 25% of the training data set aside for validation. An extra set of test data set was created to test the model performance after training and validation process was completed.

Due to large amount of data contained in the MIO-TCD dataset, a subset of the entire dataset was chosen in the interest of upload time and training time in the DIGITS workspace. This subset was chosen by selecting the lowest size (in KB) images in each of the four classes. This dataset is referred to as the "Motorist Dataset" throughout the article.

E. Example Dataset

The Motorist dataset contains four classes, namely car, bicycle, motorcycle, and background. An example of each class of dataset is provided below.

1. Car: Total 1980 images of cars were used for training and validation.

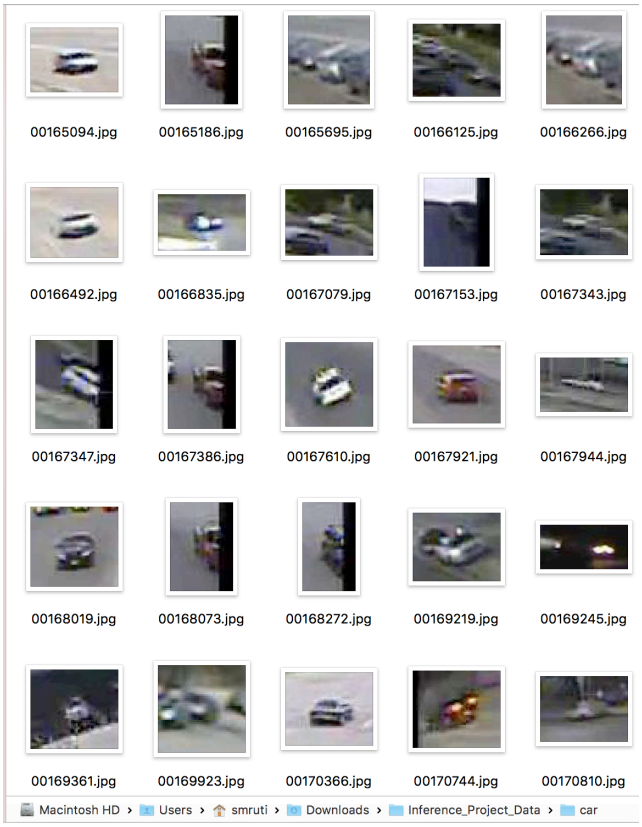


Fig. 1: An example car dataset.

2. Bicycle: Total 1980 images of the bicycles were used for training and validation.

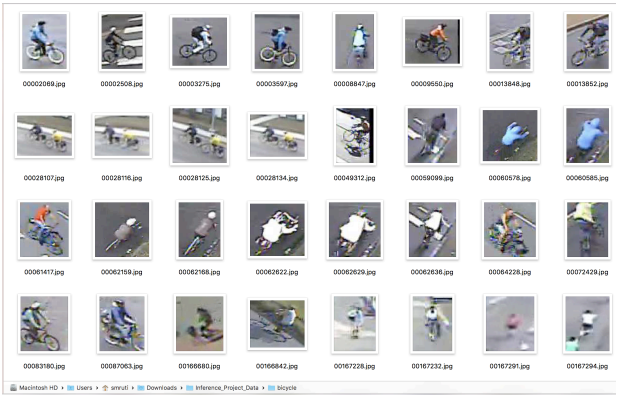


Fig. 2: An example bicycle dataset.

3. Motorcycle: Total 805 images of motor cycles from different angles were used for training and validation.

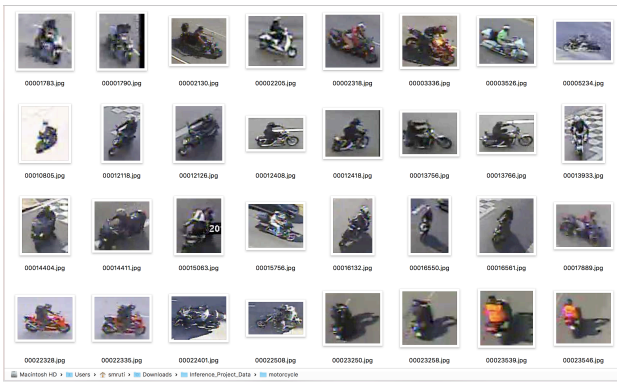


Fig. 3: An example motorcycle dataset.

4. Background: Total 1096 images of the background were used for training and validation.

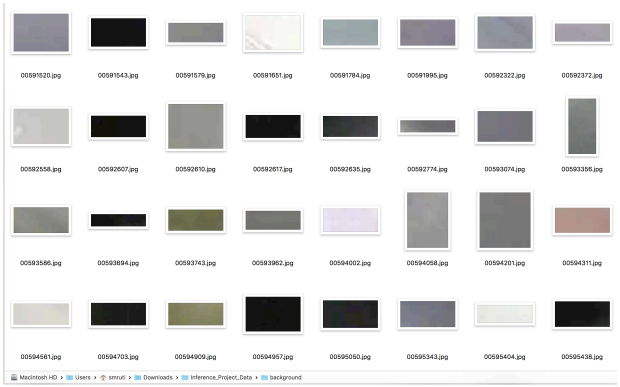


Fig. 4: An example background dataset.

Initially 1980 images from each class were uploaded to the DIGITS workspace as training and validation images. However, some of the images were currupted and were not used during training the model. Hence those corrupted images were removed from the dataset (from the motorcycle and background datasets). After the model was tranined with the clean dataset, a set of images containing 20 images from each class were used to test the accuracy of the model.

III. Results

A. *Conveyor Dataset*

Using the conveyor image dataset, GoogLeNet model is used for training and validation of the dataset. Using Adam optimization with a learning rate of 0.001, the network was trained for 8 epochs as shown in Fig. 5 and Fig. 6. The model was tasked with classifying between bottles, candy boxes and the background conveyor belt. Evaluation of the trained model results show 75.41% accuracy (meets the targeted accuracy of 75%) with an inference time of about 5.5 ms (below the targeted inference time of 10 ms), as shown in Fig. 7.

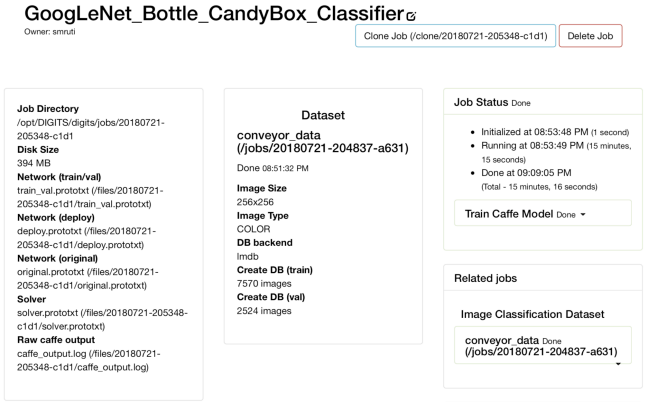


Fig. 5: GoogLeNet Model applied to the Conveyor dataset

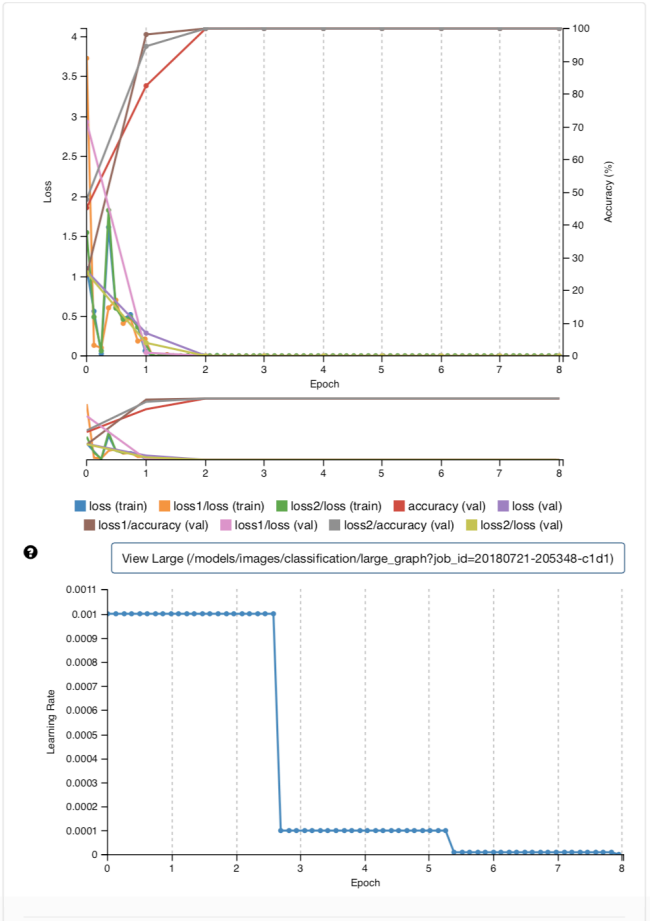


Fig. 6: GoogLeNet model trained for 8 epochs with an Adam optimizer, and an initial learning rate of 0.001.

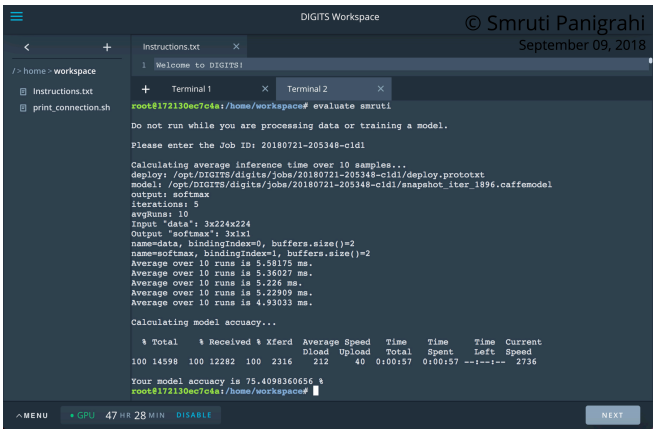


Fig. 7: Model accuracy of the conveyor dataset using a trained AlexNet model with SGD learning rate 0.001.

B. *Motorist Dataset*

Three different models were used to train and evaluate the inference performance on the extracted Motorist dataset from the MIO-TCD dataset. The LeNet model was used by converting all images into 28x28 grayscale images in the Digits workspace through the classification data preparation. The model was trained for 16 epochs with SGD solver with learning rate of 0.001. Evaluating with the test dataset resulted in accuracy of ??90%. Further retraining the saved LeNet model for another 16 epochs did not improve the accuracy. The model training and validation result for each epoch is shown in the plot below in Fig. 8.

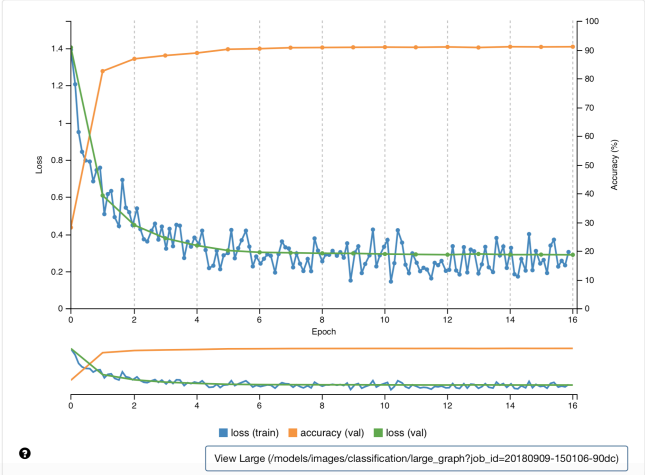


Fig. 8: LeNet on the Motorist dataset after 16 epochs using SGD optimizer of initial learning rate 0.001.

Next, the AlexNet was used to train the model with an Adam optimizer of initial learning rate 0.001. In this case, the dataset was converted into 256x256 resolution while retaining the RGB color format of the dataset. After 16 epochs, the evaluation result with the test dataset showed lower accuracy than the LeNet with SGD as seen in Fig. 9. When the pre-trained model was used to train for another 16 epochs the validation accuracy improved significantly as seen in Fig. 10. The resulting final accuracy for the test dataset was also improved. However, the bicycle was classified as motorcycle almost 40% of the bicycle test images.

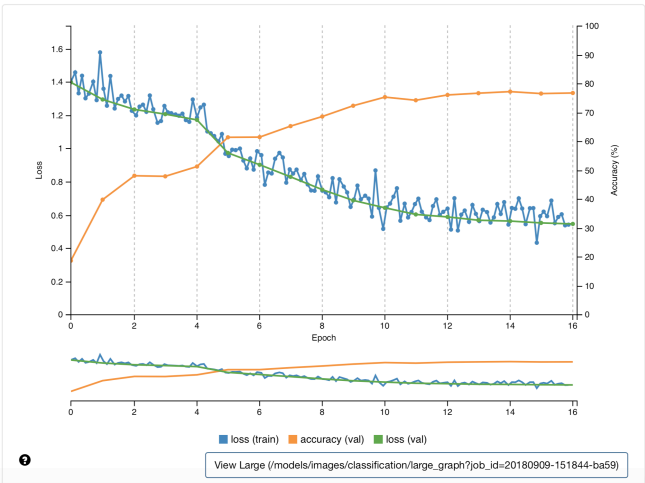


Fig. 9: AlexNet on the Motorist dataset after 16 epochs using Adam optimizer of initial learning rate 0.001.

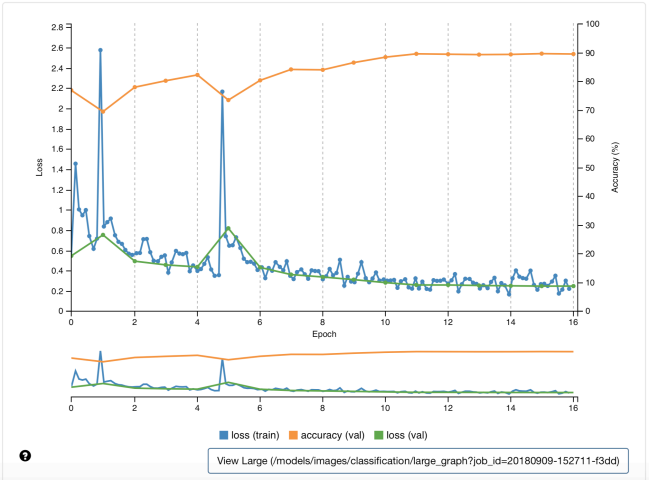


Fig. 10: AlexNet on the pre-trained model after 16 more epochs using Adam optimizer of initial learning rate 0.001.

Finally the most complex DNN was used on the Motorist dataset. Using GoogLeNet to train the dataset with an Adam optimizer of initial learning rate 0.001, the Motorist dataset was trained for 16 epochs. The learning rates are dynamic and are reduced every one third of the epochs as shown in Fig. 11. Retaining the RGB color format, the images were converted into uniform 256x256 images though the digits data classification. The validation performance improved to 95.8%, as seen in Fig. 12. Evaluating the test images, GoogLeNet produced much improved classification result for the bicycle at 80% accuracy. The background was classified correctly 100% of the test images, while the cars and motorcycles had 95% and 90% accuracy as shown in the confusion matrix in Fig. 13.

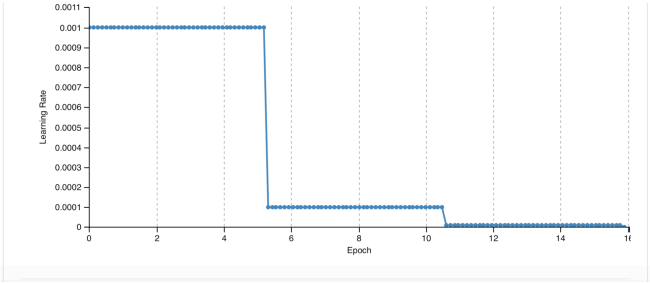


Fig. 11: Learning rate changing from 0.001 to 0.0001 at epoch 5, and to 0.00001 at epoch 11.

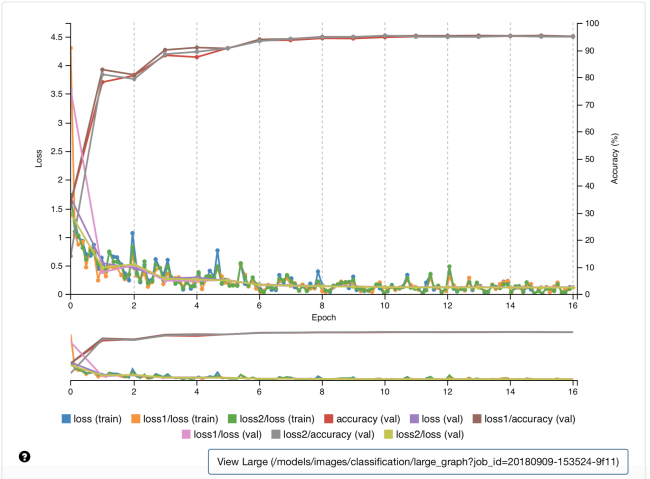
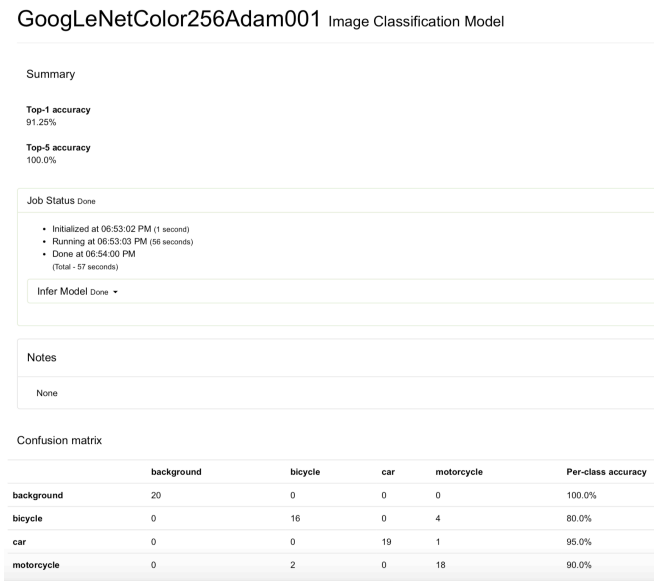


Fig. 12: GoogLeNet on the pre-trained model after 16 epochs using Adam optimizer of initial learning rate 0.001.



	background	bicycle	car	motorcycle	Per-class accuracy
background	20	0	0	0	100.0%
bicycle	0	16	0	4	80.0%
car	0	0	19	1	95.0%
motorcycle	0	2	0	18	90.0%

Fig. 13: Summary of the GoogLeNet results on the Motorist dataset containing the confusion matrix for the test data evaluation using the GoogLeNet model.

Examples of a bicycle classification and misclassification are shown in the figures 14 and 15 respectively.

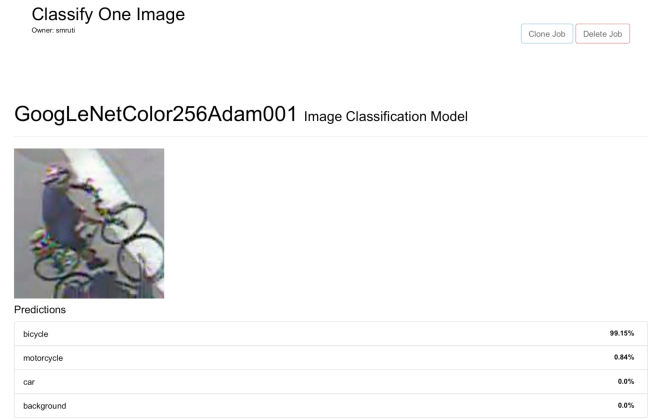


Fig. 14: Correctly classified bicycle test image.

Fig. 15: Bicycle misclassified as a motorcycle.

IV. Discussion

The Conveyor dataset was evaluated to be 75.41% accurate after training for 8 epochs with GoogLeNet model using a Adam optimization, and intial learning rate of 0.001, which meets the targated requirement.

For the Motorist dataset, all three models were used to improve the accuracy of the classification model. The GoogLeNet model was the most accurate in terms of classifying all four classes. LeNet took a fraction of the time to complete same number of training epochs compared to AlexNet and GoogLeNet.

It was observed that the misclassification of the bicycle class as motorcycle was due to the fact that the number of training images for the motorcycles were less than 50% of the bicycle images. lso using the pre-trained GoogLeNet model to train for another 16 epochs could increase the accuracy of the classification. However, since in many images the bicycles look more like motorcycles, at some point, the classification accuracy would be difficult to improve even with additional training. Hence more data for the motorcycle would be required, if classification accuracy is to be improved. Since AlexNet is much faster to train than the GoogLeNet, and the results showed 100% accuracy for all classes except for the bicycle class, using more motorcycle data and using AlexNet could be the most efficient way to improve inference accuracy.

V. Conclusion

In this Robotic Inference project a deep-CNN model on the Motorist data obtained from MIO-TCD [5, 6] was trained and validated using the Nvidia DIGITS workspace. It was extremely beneficial to have existing models formulated in the Nvidia DIGITS, especially LeNet, AlexNet and GoogLeNet, in order to quickly train with new dataset. Also, having the flexibility of using a pre-trained model and training from there is very useful to check for model accuracy improvement with further traning.

Two different datasets, Conveyor Dataset and Motorist Dataset, were trained using various deep-CNN models and their accuracies were compared in this paper. Once a well-trained model is established using the desired dataset, the model can be saved and used for realtime application. However, it is important that the image quality/resolution be taken into account when the model is intended to be used in realtime. For example, if the Motorist model is to be used in realtime autonomous vehicles, it is important that the dataset be populated with images taken from the camera similar to the ones equiped with the AV.

Improvements can be made to the models trained in this paper, by augmenting data and using additional data for motorcycles. Also it is possible to improve the model accuracy by training for more epochs and using different learning rates.

Acknowledgment

The authour would like to thank Udacity Inc. for providing certain GPU times for training and testing various deep-CNN models.

References

[1] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

[2] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[3] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[4] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 248-255).

[5] Luo, Z., Frederic, B., Lemaire, C., Konrad, J., Li, S., Mishra, A., Achkar, A., Eichel, J., & Jodoin, P. M. (2018). MIO-TCD: A new benchmark dataset for vehicle classification and localization. *IEEE Transactions on Image Processing*.

[6] http://podoce.dinf.usherbrooke.ca