Vehicle Make and Model Classification Using Convolutional Neural Networks

Syed Hasib Akhter Faruqui, Rajitha Meka
Department of Mechanical Engineering, San Antonio, Texas
{syed-hasib-akhter.faruqui,rajitha.meka@utsa.edu}

***Abstract* - In recent times, for the purpose of object detection in large-scale generic database, Deep Convolution Neural Network (D-CNN) has gained popularity. This has become the core of most state of art computer vision solutions for wide variety of tasks. The main aim of this report is to detect the make and model of the vehicals. This is important in the area of traffic control management, identification, stolen APB etc. In this project, state of art Convolution Network along with one of our own constructed network are compared. The architecture of our network is a deep convolution network with very small convolution filter, which showed a significant improvement in computation. Although, GoogleNet Net shows a significant accuracy in classification compared to our model.**

***Keywords* - CNN, Vehicle Classification, Inception(V3) Net**

I. INTRODUCTION

Vehical identification is an important aspect in intelligent traffic systems. There are wide variety of car models available in the market. In the competetive environment , every car maker is striving to release new car models every year with improved car styling which suit the majority of the customer taste. Identification of car models has become a topic of interest for enthusiasts and hobbiests. The problem with identifying car models is the appearance in different views as shown in Figure 1. Many of the car models carry some of the old features from the previous model while taking in a new design approach. For humans this is a fairly easy task but however due to complexity in design as asthetics this is a hard task for computers. These are basically identified as fine-grained problems.

Figure 1: Different view point of cars in the dataset

The fine grained images are basically objects that are both semantically and visually similar to each other with some minor changes in features that differentiates the objects from one another [3]. There have been a lot of investigation for the purpose of classifying such object classes like birds [13] In cases like this use of Deep Convolution Neural Network (D-CNN) have proven to be quite efficient. It is to be noted that the presence of large-scale labeled data is one of the key reasons behind the success of CNN's [2][4]. But it becomes difficult to train a D-CNN if the dataset is small. As small dataset may cause the overfitting of the weights in the network. This problem

can be solved using the methods of image augmentation. It can be seen from the previous works that image augmentation for the purpose of increasing training set has been proven helpful [5][6].

A. Project Description

The project is to classify the make and model of the cars using the supervised techniques. The experiments on the cars was conducted on cars dataset [1] also known as stanford cars dataset. The dataset has 16,185 images with 196 different classes of cars. All the images of the cars are of different views , sizes and shapes. The presented data has a hierarchy of the catagories. They are given in a manner of make, model and year of the release. This hierarchy can be seen in Figure 2. Due to limitation of time and computational resources we have run one of the tests using the GoogleNet [2] with all 196 classes and a second test with only 25 classes from the 196 classes. We used GoogleNet and our model for the tests. From hereafter we will be referring to our model as NhL Net. The number of images per class varied for each class. The 25 classes were chosen from the database in a way that each class will have greater than equal to 90 images in each class. Although image augmentation methods were later introduced to increase the number of training images. Tuning some of the network parametres on the D-CNN's we were able to obtain a very high quality result from the dataset.

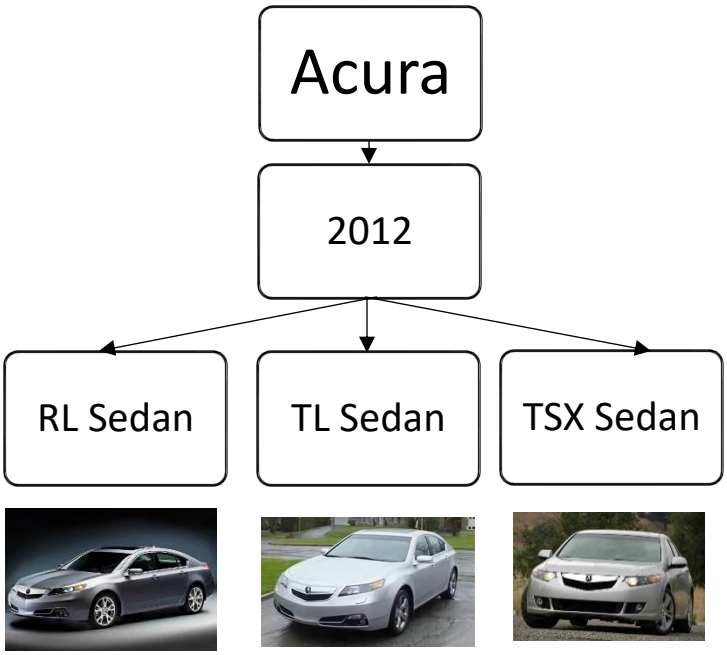

Figure 2: Tree Sructure of The Car dataset

B. Project Impact

The concept of vehicle detection is very useful in terms of security and traffic control. This concept when clubbed with other algorithms to read the licence plates will be helpful to detect the car and also the number from security cameras can be helpful in time of need. This system can also be used to search nation wide for stolen vehicals. The traffic control departments can make much use of this classification.

C. Previous Work

Most recent work on the same dataset has been done by Liu et al [7]. They have provided a coparative comparision between different state of art convolution models like CaffeNet [8], VGGNet [5], GoogleNet [2], AlexNet [3]. There has been work on leraning different features and parts of the car with the help feature desccriptors like HOG [10] features and CNN's [9]. Zhang et al [11] fitted an wireframe model to the car structures of an image. These wirefeames were then used to recognize the model of the car based on the training parametres. These 3D wireframe method was later changed and improved to construct 3D space curves based on the 2D training images. Using the 3D alignment the curves are matched to testing images to alignment to determine the make and model of the cars [12].

However it's still an open problem on differentiating fine grained object classification tasks. The recent improvements of CNN's since the winning model of Krizhevsky et al. [3] their developed network "AlexNet" has been widely used in different vision related problems like object detection, segmentation, pedastrian identification, human pose estimation, video classification, autonomous driving, object tracking and supper resolution. VGGNet and GoogleNet were the successors of AlexNet. But regardless of their performance Alexnet needed much more computational effort compared to its successors. AlexNet used 60 million parametres where VGGNet used three times more parameters than AlexNet. Whereas, GoogleNet employed only 5 million parametres which is 12x reduction in comperision to AlexNet. Thus the decision to use the GoogleNet for the classification work. The architecture of NhLNet and its structure reasoning will be explained in later sections.

The dataset description, pre-processing of the data, implementation of the network and analysis are explained in detail in a structured way in the next section.

II. PROPOSED METHOD AND APPROACH

A. Dataset

This dataset, consists of 196 classes and 16,185 images and is larger than the existing fine-grained car dataset [13] (14 classes, 1,904 images). This data set is also known as Stanford Cars Dataset [1], which is available at the following website (http://ai.stanford.edu/~jkrause/cars/car_dataset.html). Althoug the classes are fine grained, each of these classes can be identified as visually distinct from one another. For example the 1991 & 2012 Volkswagen Golf are different in apperance. 2011 Volkswagen Golf is identical to 2012 model, so is not added in the dataset. Each of the images consists of cars with noises in the background. Some of the images were professionally shot while some of them were collected from classified ads and internet resources. The annotation file included with the dataset contains the location of each car in an image. This facilitates identification of the cars for the purpose of easy calculation.Sample images from the dataset with different angles of view are shown in Figure 3.

Figure 3: A sample of images from the cars dataset demonstrating a range of cars from different makes and model

B. Pre-Processing

All the images in the dataset were jpg format which are converted to png format for ease of use. The preprocessing of images are done in three steps- i) Cropping of Images according to the bounding boxes provided in to the annotation files. This is done to get rid of the extraneous noise from the training data. ii) Now the set contains car images of different dimensions and aspect ratios. Resizing all the images to fit our network is a necessary step. For this project we have resized all the images in to size "227x227x3". While doing this some of the images were compressed to fit our desired size without considering the aspect ratio of the original images. iii) Data augmentation is done by adding some gitters, rotating, mirroring, increasing/decreasing saturation of the images. This is done to increase the number of training images. As we designed the networks to take RGB images, Black & White images were also removed from the dataset. This will help us to avoide any future errors. We created our preliminary training and testing sets by splitting the dataset into 70-30 section. Figure 4 shows the black and white images that were excluded from the computation.

Figure 4: Sample Black and White images in Dataset

C. Implementation of Network

While designing the NhL net we followed the same philosophy as Simonyan et al [5]. We tried to develop a very deep convolution network that uses smaller filter windows and smaller stride sizes for convolution layers [15,16]. We have tested the NhL Net for two different configurations. The First configuration have 10 convolution network (6 of them are 3x3, 2 of them are 2x2 and 2 of them are 1x1 in size). The inclusion of 1x1 filters is done to add the effect of non linearity in the form of transforming the input channels without affecting the receptiive fields of the convolution layer. Padding is used for the images during convolution operations thus the image size remains the same through out the layers. A total of 5 spatial pooling is carried out in the form of max-pooling. Max pooling is performed for a filter size of 2x2 with stride 2. And the second configuration have 10 convolution network of the same size (3x3). This configuration has one less maxpooling than the previous model.

The convolution layer is later followed by a single Fully-Connected Neural network layer. Before passing the image inputs to the neural network we pass it through an operation of Local Response Normalization (LRN) similar to Krizhevsky et al [3] and then to dropout filter to avoide overfitting [17].

It is to be mentioned before passing the input images of fixed size to the network we did a small image normalization task. We took the mean of an image and substracted it from each pixel of that image to reduce some of the common noise from the picture. Thus we hypothesized that only the pixels with high intensity along with the features of the object in the image will remain. That will make the computation of feature detection easier in the ocnvolution stage. The two configurations that we used are showed on Figure 5.

We also implemented GoogleNet for the purpose of comparision with NhL Net. GoogleNet is a direct improvement of the AlexNet. It has an architecture of 22 layers. The model uses Inception module which is a different level of organization to process the input data. Its classification capabilities are experimentally varified on the ILSVRC 2014 classification and detection challenges [2]. It is worthwile to mention that GoogleNet is 12x smaller in comparison to AlexNet. GoogleNet generates 3 output for each layer. In this project we have only taken the output from the last layer as it is proved to giving the best resulting output.

D. Analysis

In the past increasing the depth of convolution network led to better performance [2,5,18]. The training procedure we followed is similar to AlexNet [3]. The optimization is carried out using Adam gradient Optimizer [19]. With an initial learning rate of 0.001, this is later decreased by a factor of 10 whenever the accuracy of validation set stopped improving. The initial training processes a batch of 45 which was later reduced to a set of 20. In the total training process the learning rate was changed four times and batch size was changed only once.

For the same number of iterations it was observed that the second network with 5 layers was performing well in comperision to the formar one. It seemed to us that as we hypothetized that the addition of 1x1 filters will add some non-linearity to the calculation that may help identifying the objects even better was not as affective as we have expected. The comperisional chart of accuracy for the models is shown in table 1 of chapter III.

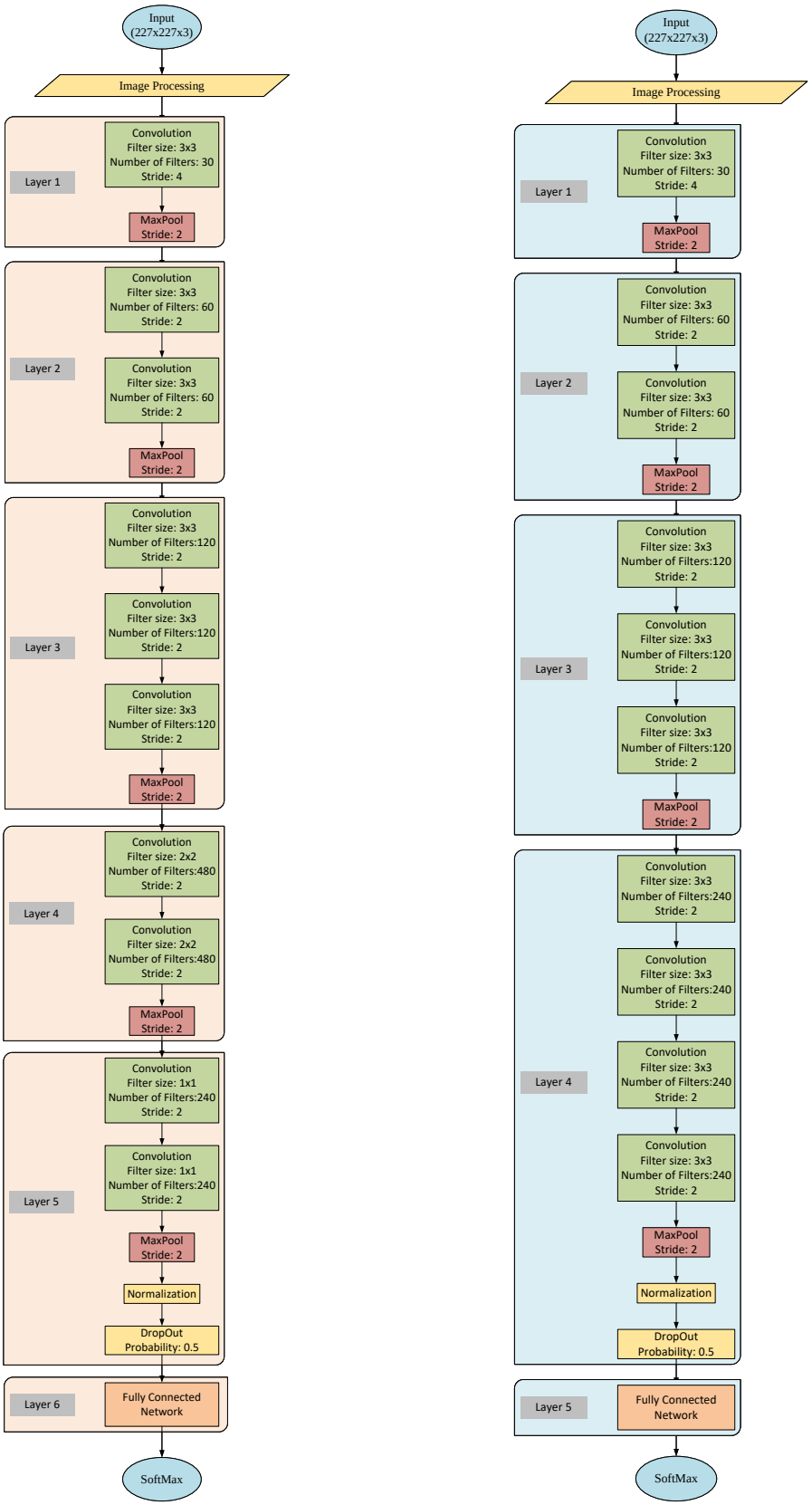

Figure 5: Flow Chart of NhL Net

III. RESULTS AND ERROR ANALYSIS

Table 1 shows the individual results obtained from the tests. We have run two different tests for the purpose of training. One with the total 196 classes from the dataset and second one for the 25 classes from the dataset. The confusion matrix of NhLNet_B for both 196 and 25 classes are shown in Figure 10. From the confusion matrices it is evident that the NhL Net tends to misclassify some classes more than others (e.g. Ford GT Coupe 2006, Mercedes Benz 300-class convertible, Mercedes Benz C class Sedan). Whereas in case of GoogleNet it only misclassified one of them (Ford Focus sedan 2007).

Table 1: Top 1 accuracy for all the experiments

Model	Classes/ Labels	Top 1 Accuracy	Initial Learning Rate
GoogleNet (InceptionNet)	196	0.773	0.001
GoogleNet (InceptionNet)	25	0.981	0.001
NhLNet_A	25	0.183	0.001
NhLNet_B	25	0.951	0.001
NhLNet_B	196	0.753	0.001

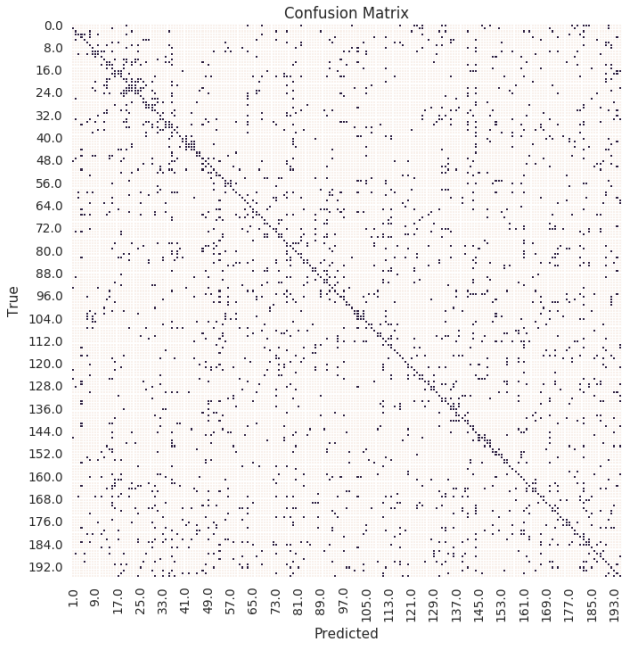

Figure 6: GoogleNet confusion matrix for 196 classes

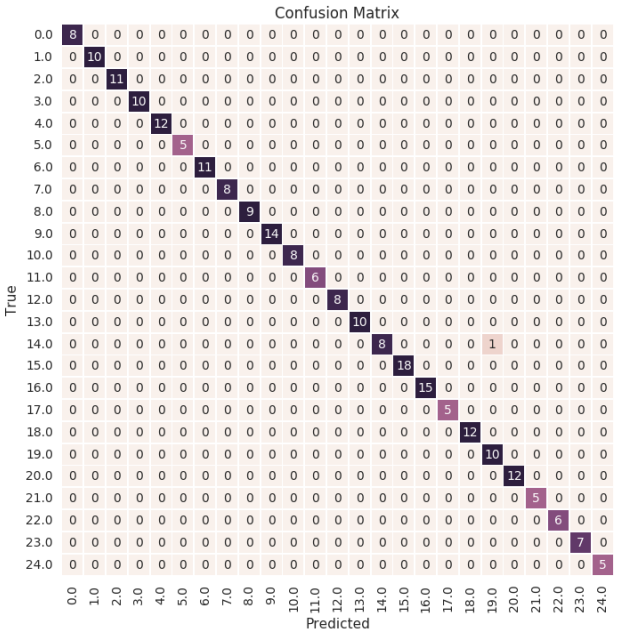

Figure 7: GoogleNet confusion matrix for 25 classes

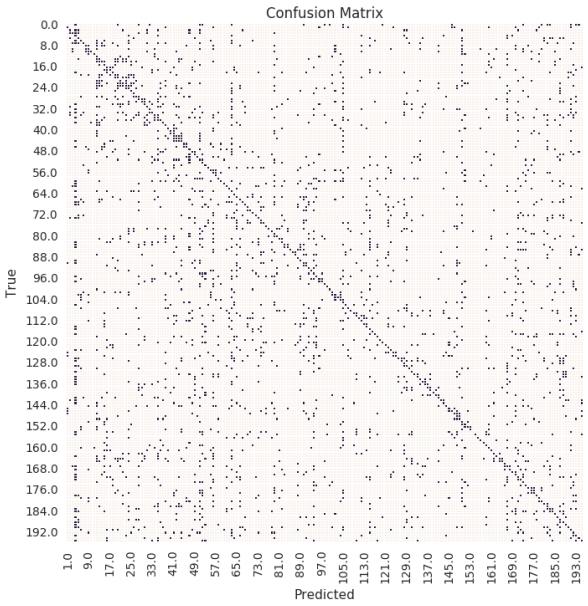

Figure 8: NhLNet_B confusion matrix for 196 classes

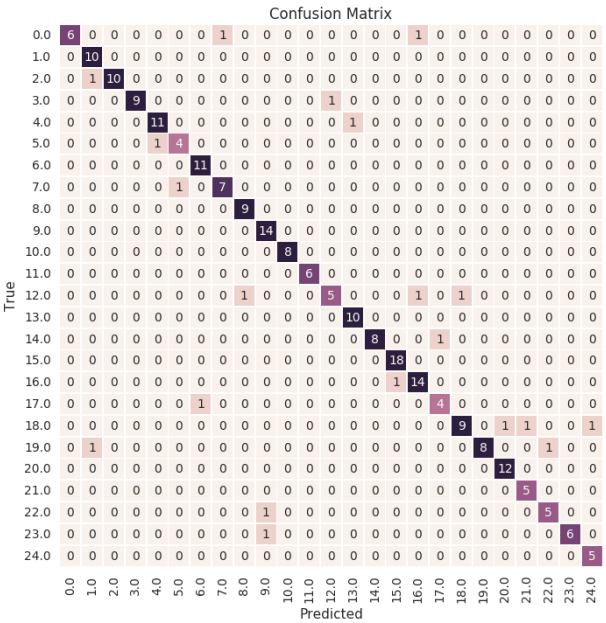

Figure 9: NhLNet_B confusion matrix for 25 classes

When we looked into the misclassified cars we found that both of the networks misclassified the same car to be of the different class. This is shown in Figure 10. After checking it through we hypothetized that the reason is due to the compressed images which got compressed during the pre-processing step of re-sizing all images to same size.

(a) Original Image (Ford Focus sedan 2007)

(b) Misclassified as (Mercedes Benz C class Sedan2012)

Figure 10: GoogleNet Misclassification

GoogleNet was able to classify most of the images perfectly. "Mercedes Benz C class Sedan2012" which have a similar appearance from the side with "Ford Focus sedan 2007" was not classified correctly. In case of NhLNet it would seem that some of the misclassification was related to images being visully similar. Though the classes were reasonalby diverse in the database, some of the models are virtually similar which might prove hard even for human

to distinguish. Some examples of such cars may include Chevorlet corvette ZR1 and Chevorlet corvette ZR1, Audi and BMW sedan etc.

Figure 11: NhLNet Misclassification

Other than visual similarity there may be other source of errors such as enough training data, data split distribution etc. Due to time constraint of this project not much effort was put throuh to those issues. Although data augmentation to some level is used to increase the number of images in training set of 25 class set. Apparently, the models for classifying are performing somewhat well. We also observed that some compressed images after image resizing may also add to some of the skewing of weights in the network.

IV. CONCLUSION

We have tried to present a useful framework for car classification. Within the scope of the two D-CNN models were compared. With the smaller dataset of 25 classes we were able to reach higher classification accuracy. Given time and computational effort the same can be achieved for the bigger class (196 classes) dataset too. A possible future work for this dataset can be the use of transfer learning from other trained models to speed up the training process. Also preprocessing of images (Bounding Box, Cropping, Scalling) can be done in a manner to avoid compression of the original images.

It is to mention that both models (GoogleNet and Our model) used in this project were very deep convolution networks. It was demonstrated that the increasing the depth of convolution network can be benificial for both computation and classification.

Acknowledgment

We would like to thank Dr. Peyman Najafirad, for his advice and guidance during the project and Ali Miraftab, Teaching Assistant for the course EE-6973-001-Fall-2016-SP:MachineLearningw/BigData for his constant help during the time of coding and training the models. Results presented in this paper were obtained using the Chameleon testbed supported by the National Science Foundation.

REFERENCES

[1] Krause, Jonathan, et al. "3d object representations for fine-grained categorization." *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2013.

[2] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[3] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[4] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." European Conference on Computer Vision. Springer International Publishing, 2014.

[5] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[6] Zagoruyko, Sergey, and Nikos Komodakis. "Learning to compare image patches via convolutional neural networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[7] Liu, Derrick, and Yushi Wang. "Monza: Image Classification of Vehicle Make and Model Using Convolutional Neural Networks and Transfer Learning."

[8] Jia, Y. Q. C. "An Open Source Convolutional Architecture for Fast Feature Embedding." (2013).

[9] Krause, Jonathan, et al. "Learning Features and Parts for Fine-Grained Recognition." ICPR. Vol. 2. No. 7. 2014.

[10] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 1. IEEE, 2005.

[11] Zhang, Zhaoxiang, et al. "Three-dimensional deformable-model-based localization and recognition of road vehicles." IEEE transactions on image processing 21.1 (2012): 1-13.

[12] Ramnath, Krishnan, et al. "Car make and model recognition using 3D curve alignment." IEEE Winter Conference on Applications of Computer Vision. IEEE, 2014.

[13] M. Stark, J. Krause, B. Pepik, D. Meger, J. J. Little,B. Schiele, and D. Koller. Fine-grained categorization for 3d scene understanding. In BMVC, 2012.

[14] TFLearn: Deep learning library featuring a higher-level API for TensorFlow.

[15] Sermanet, Pierre, et al. "Overfeat: Integrated recognition, localization and detection using convolutional networks." arXiv preprint arXiv:1312.6229 (2013).

[16] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." European Conference on Computer Vision. Springer International Publishing, 2014.

[17] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." Journal of Machine Learning Research 15.1 (2014): 1929-1958.

[18] Goodfellow, Ian J., et al. "Multi-digit number recognition from street view imagery using deep convolutional neural networks." arXiv preprint arXiv:1312.6082 (2013).

[19] Duchi, John, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." Journal of Machine Learning Research 12.Jul (2011): 2121-2159.