**Classification of Vehicles using GoogLeNet Convolution Neural Network(CNN) Model**

Roopali Rajaram Sarode

Department of Computer Science

New Jersey Institute of Technology

Newark, NJ 07102

Contact: [rs366@njit.edu](mailto:rs366@njit.edu)

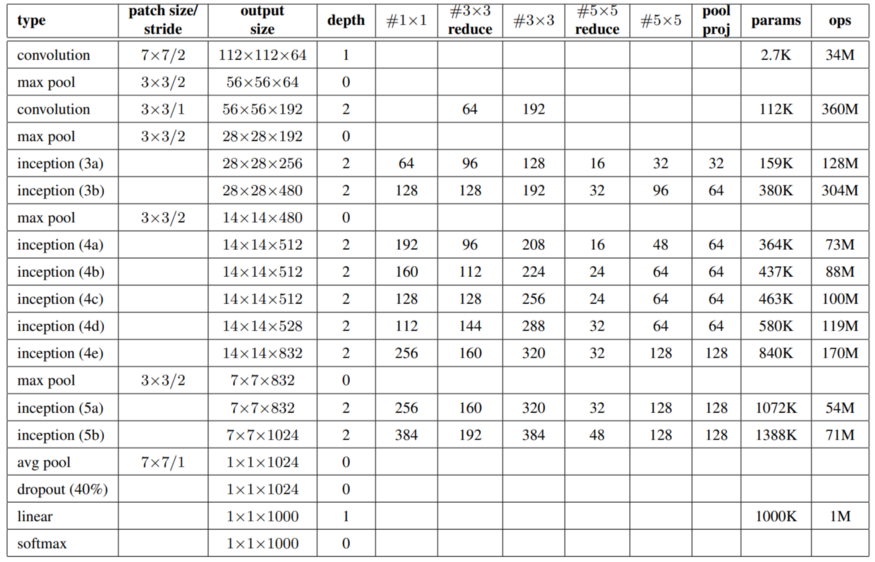
**Abstract**

Image classification consists of segmentation of images into various categories based on their features. A feature could be an edge in an image, a pixel intensity, the variation in pixel value and many more. Classifying vehicles based on an image can be tricky and difficult due to the pose of the vehicle, variation in angle, lighting conditions and lack of training dataset available. This is where the Convolution Neural Network comes into play.

It is easier to use pre-trained convolutional neural networks to classify images rather than building one from scratch. Another good thing about pre-trained CNN is they are well optimized. Transfer learning is used to simplify and make deployment processes faster. It takes long to design a convolutional neural network and even longer to optimize it. On the other hand, transfer learning can retrain an existing optimized convolutional neural network almost instantly.

**I. INTRODUCTION**

CNN is widely used in the field of pattern recognition and image processing as it is an effective, efficient algorithm. The GoogLeNet architecture unlike others uses different kinds of methods such as 1x1 convolution and global average pooling which enables it to create deeper neural networks. The overall architecture is 22 layers deep (27 layers including pooling layers). It is popularly used for computer vision tasks such as face detection, and recognition, adversarial training etc. The architecture leverages feature detection at different scales through convolutions with different filters and significant reduction in computational cost of training an extensive network. The table below shows the conventional GoogLeNet architecture.



[Figure 1: GoogLeNet configuration details, A variant of the Inception Architecture](https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43022.pdf)

## Characteristics and features of GoogLeNet configuration table (figure 1)

* The input layer of the GoogLeNet architecture takes in an image of the dimension 224 x 224.
* **Type**: This refers to the name of the current layer of the component within the architecture
* **Patch Size**: Refers to the size of the sweeping window utilized across conv and pooling layers. Sweeping windows have equal height and width.
* **Stride**: Defines the amount of shift the filter/sliding window takes over the input image.
* **Output Size**: The resulting output dimensions(height, width, number of feature maps) of the current architecture component after the input is passed through the layer.
* **Depth**: Refer to the number of levels/layers within an architecture component.
* **#1x1 #3x3 #5x5**: Refers to the various convolutions filters used within the inception module.
* **#3X3 reduce #5x5 reduce**: Refers to the numbers of 1x1 filters used before the convolutions.
* **Pool Proj**: This is the number of 1x1 filters used after pooling within an inception module.
* **Params**: Refers to the number of weights within the current architecture component.
* **Ops**: Refers to the number of mathematical operations carried out within the component.

In this paper, a vehicle classification model has been developed using GoogLeNet’s CNN. Below are the five steps to apply transfer learning for vehicle classification.

1. Loading dataset of various kinds of vehicles
2. Splitting the dataset into training, validation and testing datasets
3. Resizing the input images as per the GoogLeNet’s input image size
4. Modifying the feature learner and output layer of the GoogLeNet.
5. Re-train the feature learner layer using custom dataset

**II. DATASET**

For this paper, four classifications of vehicles have been created, labeled as airplane, bus, car, motorcycle. Each category contains 20-25 images for training, validation and testing purposes. Dataset images have been acquired from open source internet. Images of various sizes, lighting and taken from different angles have been chosen for training the neural network. GoogLeNet requires the input images to be of 224x224 pixels and have three channels. So the images have been resized accordingly.

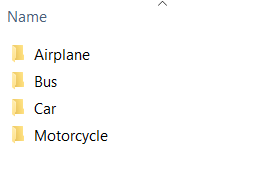


Figure 2: Dataset Labels

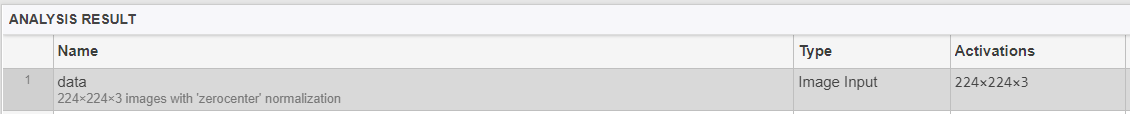
****

Figure 3: Input layer of GoogLeNet

**III. CREATING AND TRAINING THE NETWORK**

The dataset has been split into training, validation and testing dataset. For training of the network 70% of the dataset has been assigned, 15 % each has been assigned to validation and testing of the network. Since googLeNet has been trained to classify images into thousands of categories including bus, car, airplane etc. It will use the images given to it for training and classify them in appropriate categories.

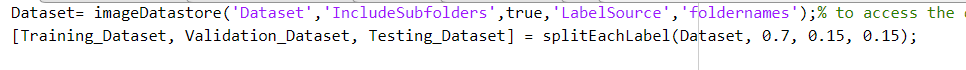


Figure 4 : Dataset split for training, validation and testing

We need to modify the googLeNet to retrain to classify vehicles according to custom dataset. In order to train the neural network we need to modify the feature learner layer of the GoogLeNet using a custom data set.

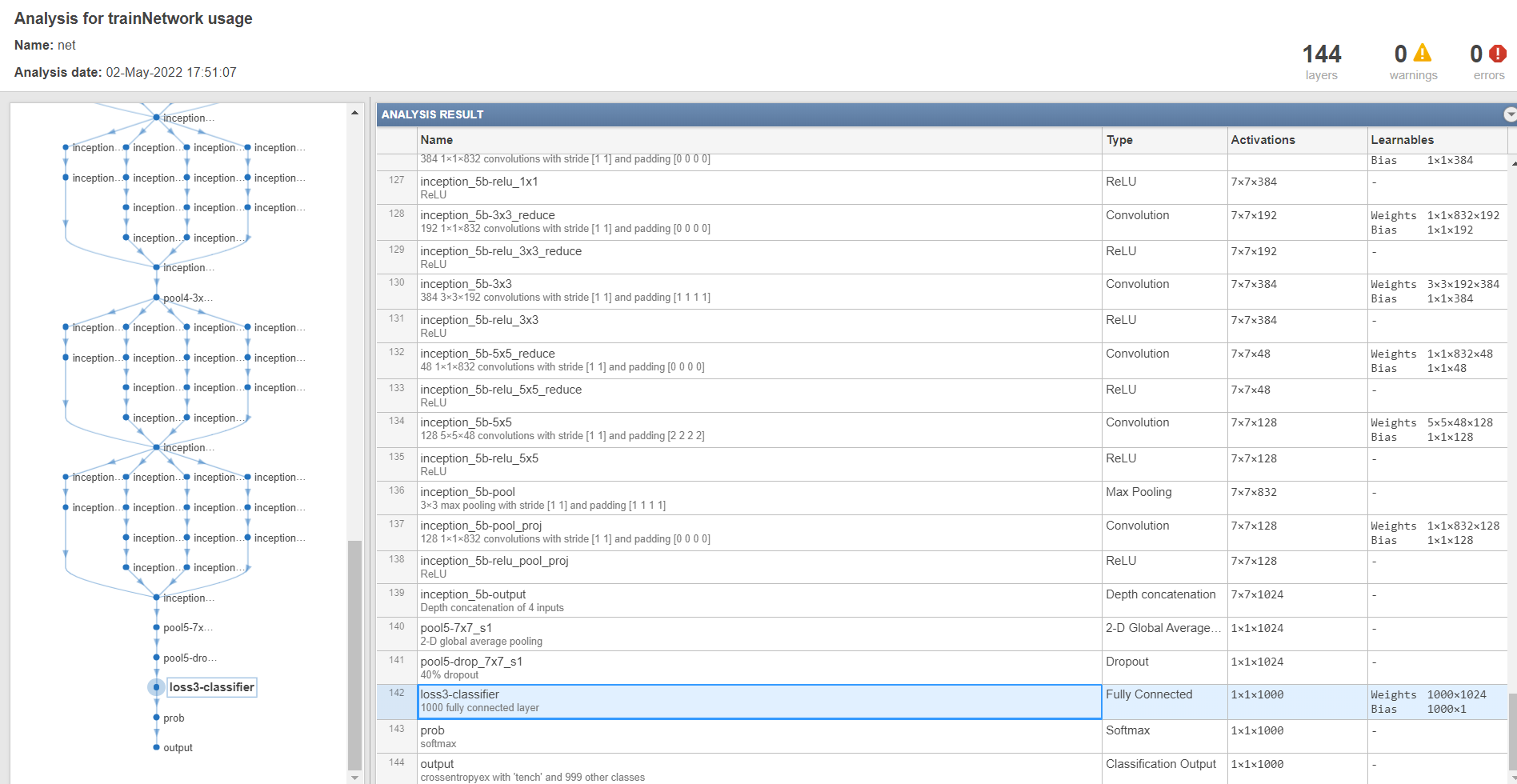


Figure 5 : Feature learner in GoogLeNet prior modification

The weights and bias are the most important concept of a neural network. The layer we are using to train the dataset uses weights and bias to classify images into categories. When the inputs are transmitted between neurons, the weights are applied to the inputs and passed into an activation function along with the bias.

When a neural network is trained on the training set, it is initialized with a set of weights. These weights are then optimized during the training period and the optimum weights are produced. A neuron first computes the weighted sum of the inputs.

**Y=Σ(weights \* input) +bias**

For instance, if the inputs are:

**x1,x2,x3,...**

And the weights are :

**w1,w2,w3,...**

Then a weighted sum is computed as:

**x1w1+ x2w2+x3w3+...+xnwn**

A bias constant is added to the weighted sum

**x1w1+ x2w2+x3w3+...+xnwn + bias**

Finally, it is fed into the activation function, which then gives an output.

**activation function(x1w1+ x2w2+x3w3+...+xnwn + bias)**

The weights reflect how important an input is. Bias is simply a constant value (or a constant vector) that is added to the product of inputs and weights. Bias is utilized to offset the result.

The bias is used to shift the result of activation function towards the positive or negative side.

The GoogLeNet has about 1000 classifications but for our paper we have considered only four (Figure 2). So the output layer of the network also needs to be modified. The labels of the folders would be considered as the 4 categories by the neural network to classify vehicles into their respective categories. The name of the new layer created is given as “Vehicle Feature Learner” which signifies the 4 categories mentioned above.

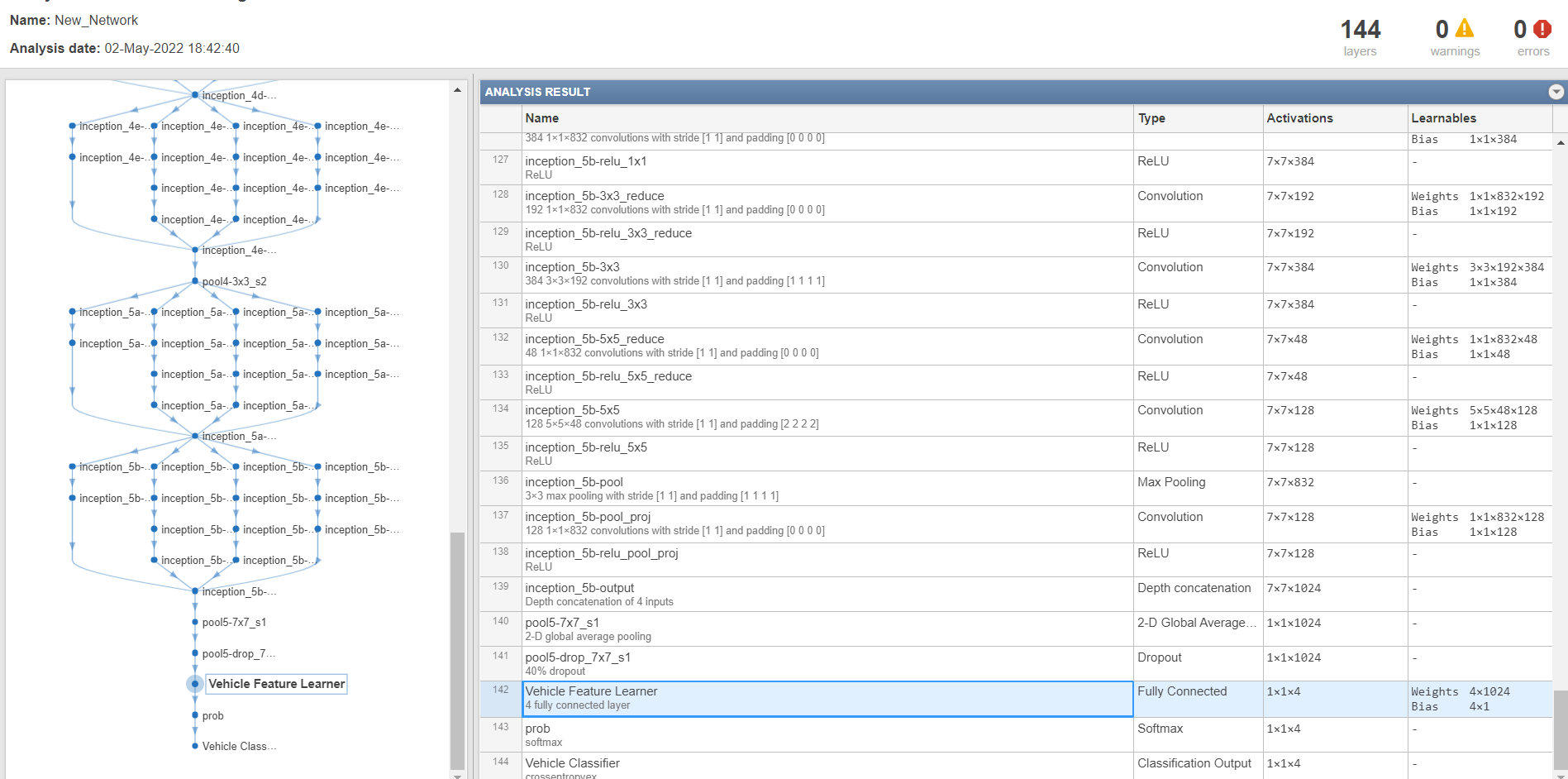


Figure 6: Feature learner layer renamed to Vehicle Feature Learner

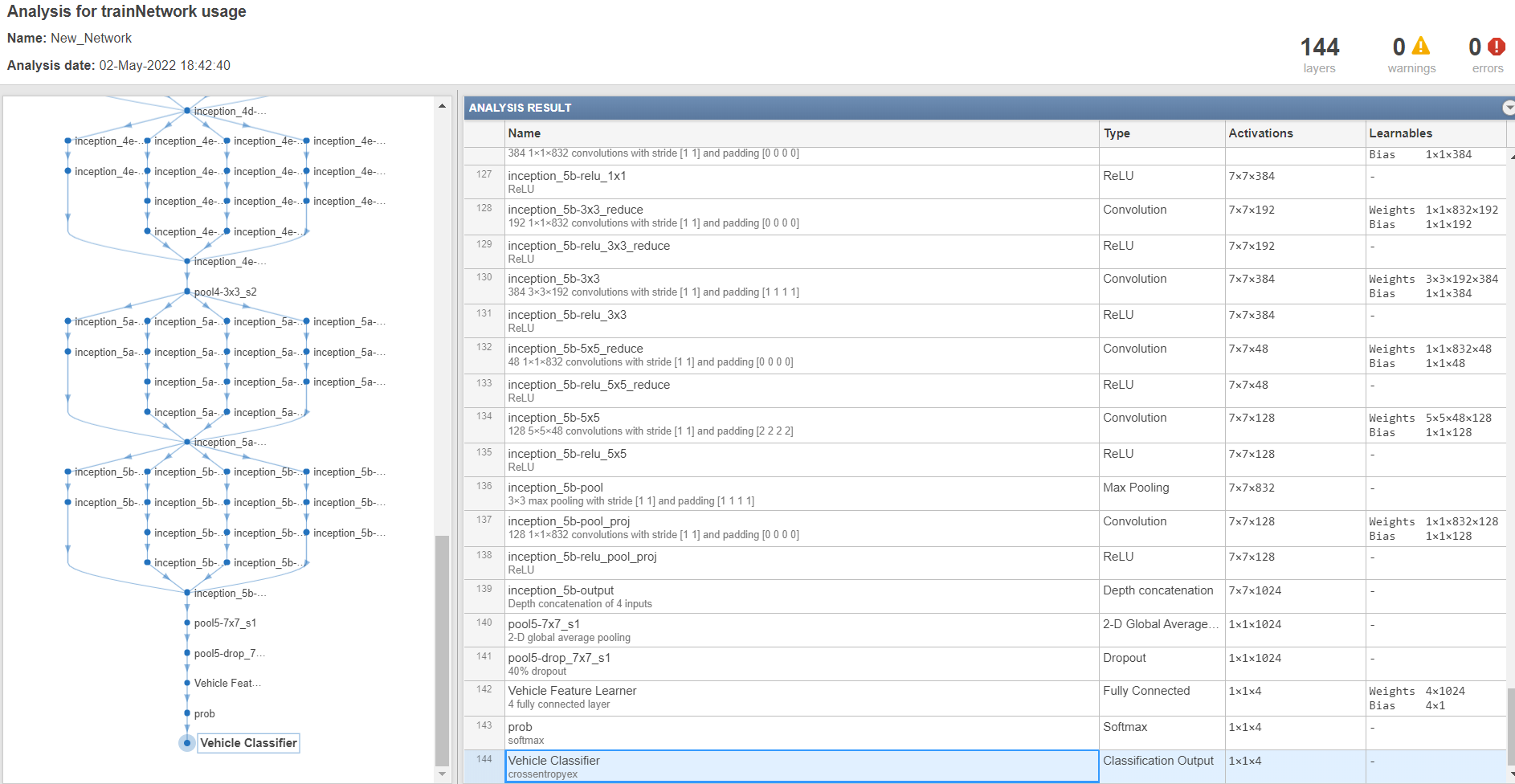


Figure 7: Output layer of GoogLeNet renamed as Vehicle Classifier

To train the network, a mini batch of size four has been used. Then the validation frequency is calculated by taking the floor value of the number of files in the resized training dataset divided by the mini batch size.

Training options function has been used to specify various training parameters. The parameters include learning mode, mini batch size, maximum epochs, initial learning rate, validation data, and validation frequency. The learning mode used is **sgdm or stochastic gradient descent with momentum.** SGDM helps accelerate gradient vectors in the right directions, thus leading to faster convergence.

Gradient descent is an optimization algorithm that iteratively reduces a loss function by moving in the direction opposite to that of steepest ascent. The direction of the steepest ascent on any curve, given the initial point, is determined by calculating the gradient at that point. The direction opposite to it would lead us to a minimum fastest.

The batch size as mentioned above has been given as four. Maximum cycles or epochs given is six, followed by a learning rate set of 3x10-4. The learning progress of the network is plotted on a graph to understand the validation progress.

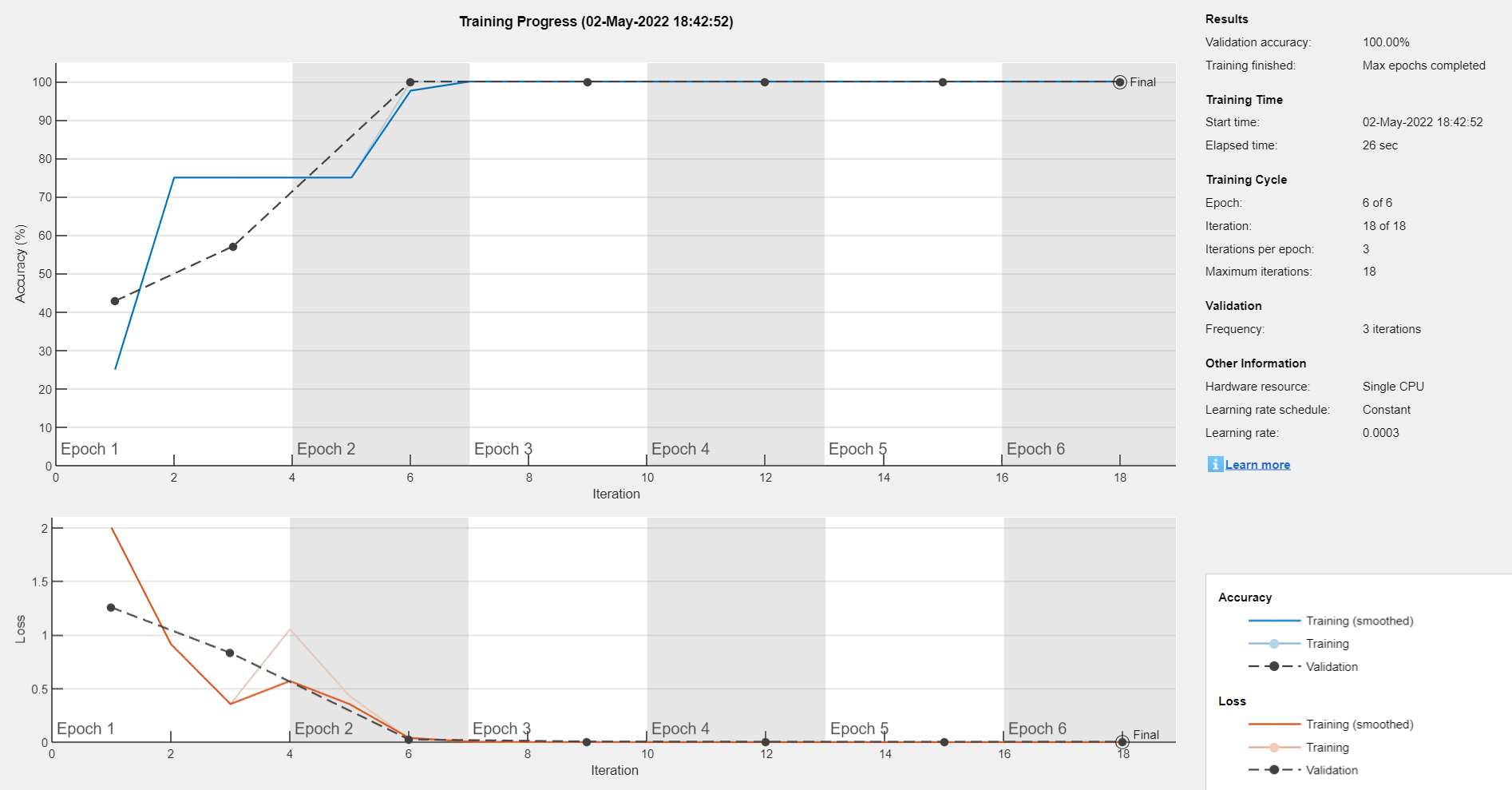


Figure 8: Learning progress graph of the network

As seen in the graph the validation accuracy keeps increasing with each epoch(cycle) and the validation loss keeps decreasing.

**IV. TESTING THE NETWORK**

To test the network created using GoogLeNet, a classify function is used. The function takes the parameters such as the network(once which we created and names as vehicle classifier) and the resized testing dataset (consisting of the 15 % of the overall dataset assigned at the beginning of the training process). The function returns the predicted label and the probability of prediction. To calculate the average accuracy of the prediction, mean is taken when the predicted label is equal to the testing dataset label. Four images are randomly selected from the testing dataset. Using a for loop the randomly chosen four images are displayed with their predicted labels as title, along with the probability of accuracy. For the given dataset, 100% accuracy was achieved. It means that the network was able to correctly identify the categories of the randomly selected images from the training dataset.

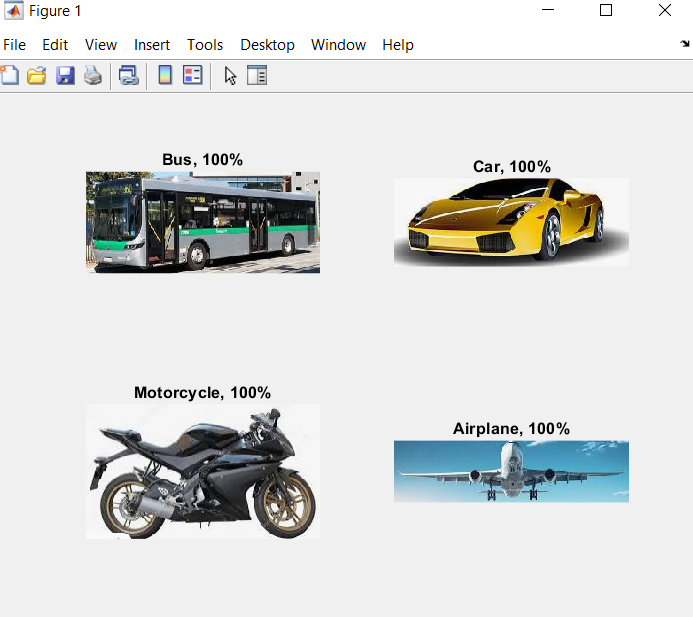


Figure 9: 1st random batch from Training dataset classified into categories

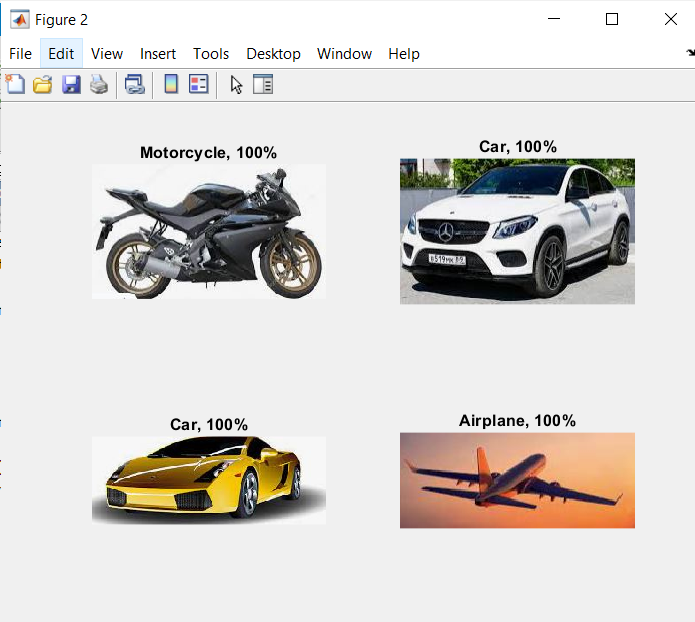


Figure 10: 2nd random batch from Training dataset classified into categories

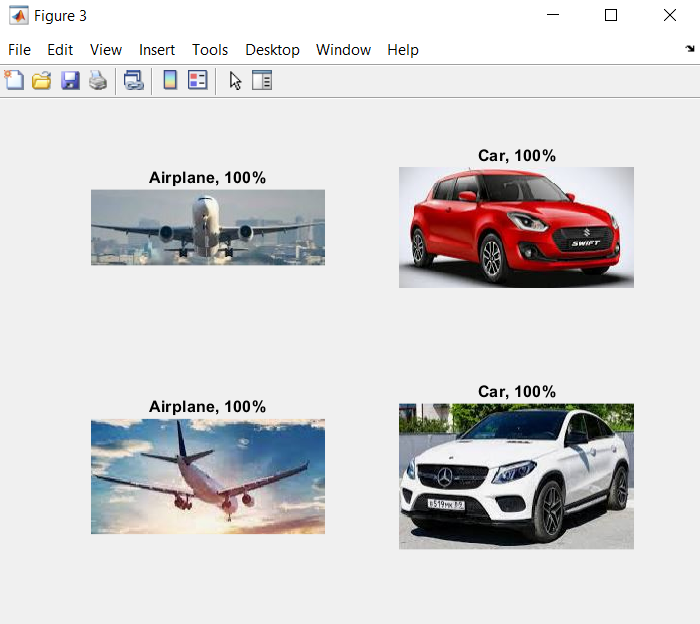


Figure 11: 3rd random batch from Training dataset classified into categories

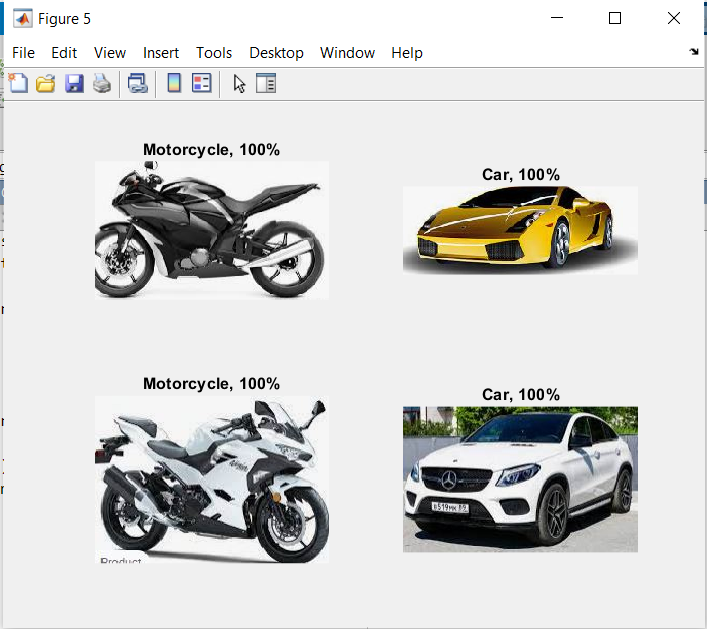


Figure 11: 4th random batch from Training dataset classified into categories

**V. CONCLUSION**

This paper aims to classify the images of vehicles into various categories using a pre-trained neural network, GoogLeNet. It takes long to design a convolutional neural network and even longer to optimize it. Using transfer learning, an existing optimized neural network can be retrained almost instantly. The dataset used for this paper was a collection of about 100 images belonging to one of the following categories : bus, car, airplane or motorcycle. Leveraging the GoogLeNet, the images were classified into appropriate categories with 100% accuracy. This network can be deployed on a video to be used in several intelligent transportation systems. By feeding a large number of images as training dataset the accuracy can be maintained and a better outcome is expected even when the images are not of the best quality.

**REFERENCES**

[1] Raja Durratun Safiyah , Zaid Abdul Rahim , Syamsul Syafiq , Zaidah Ibrahim and Nurbaity Sabri, “Performance Evaluation for VisionBased Vehicle Classification Using Convolutional Neural Network” : *International Journal of Engineering & Technology,, 7 (3.15) (2018) 86-90.*

[2] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik.” Rich feature hierarchies for accurate object detection and semantic segmentation”. In Computer Vision and Pattern Recognition, 2014. *CVPR 2014. IEEE Conference on, 2014.*

[3] Brahimi, M.; Boukhalfa, K.; Moussaoui, A. “Deep learning for tomato diseases: classification and symptoms visualization”. *Appl. Artif. Intell. 2017, 31, 299–315.*

[4] C. H. Samer, K. Rishi, and Rowen, “Image Recognition Using Convolutional Neural Networks,” *Cadence Whitepaper, pp. 1–12, 2015*

[5] J. T. Lee and Y. Chung, “Deep Learning-Based Vehicle Classification Using an Ensemble of Local Expert and Global Networks,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., vol. 2017–July, pp. 920–925, 2017.*

[6] G. Zhang, R.P. Avery, Y. Wang, “A Video-Based Vehicle Detection andClassification System for Real-Time Traffic Data Collection UsingUncalibrated Video Cameras,” Transportation Research Record: Journal of the Transportation Research Board, *1993: 138- 147, 2007*

[7] A. Krizhevsky and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” pp. 1–9