Malnad College of Engineering

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

Hassan – 573 202

Subject:Deep Learning
Subject Code:18IS864

ACTIVITY 01

VEHICLE CLASSIFICATION USING GOOGLENET

Submitted by
Harshavardhan T P (4MC18IS011)
Sanjith B V (4MC18IS041)
Abdul Azeem (4MC19IS400)

Submitted to
Dr.Balaji Prabhu
Associate Professor

Department of Information Science & Engineering
Malnad College of Engineering
Hassan - 573 202

2021-22

INTRODUCTION

It is easier to use pre-trained convolutional neural network to classify images than to build one from scratch. The best thing about pre-trained CNN is they are well optimized. For example – GoogleNet. It is one of the most popular convolutional neural networks. It has been trained to recognize 1000 different objects. What if you need to train it to classify your own data? Well, this is exactly what has been covered in this article. This article shows the process of vehicle classification using GoogleNet convolutional neural network (CNN).
GoogLeNet is a convolutional neural network that is 22 layers deep. You can load a pretrained version of the network trained on either the ImageNet or Places365 data sets. The network trained on ImageNet classifies images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. The network trained on Places365 is similar to the network trained on ImageNet, but classifies images into 365 different place categories, such as field, park, runway, and lobby. These networks have learned different feature representations for a wide range of images.
GoogLeNet is a 22-layer deep convolutional neural network that's a variant of the Inception Network, a Deep Convolutional Neural Network developed by researchers at Google.The GoogLeNet architecture presented in the ImageNet Large-Scale Visual Recognition Challenge 2014(ILSVRC14) solved computer vision tasks such as image classification and object detection find out how well it performed at the conclusion section of this article.
Today GoogLeNet is used for other computer vision tasks such as face detection and recognition, adversarial training etc.

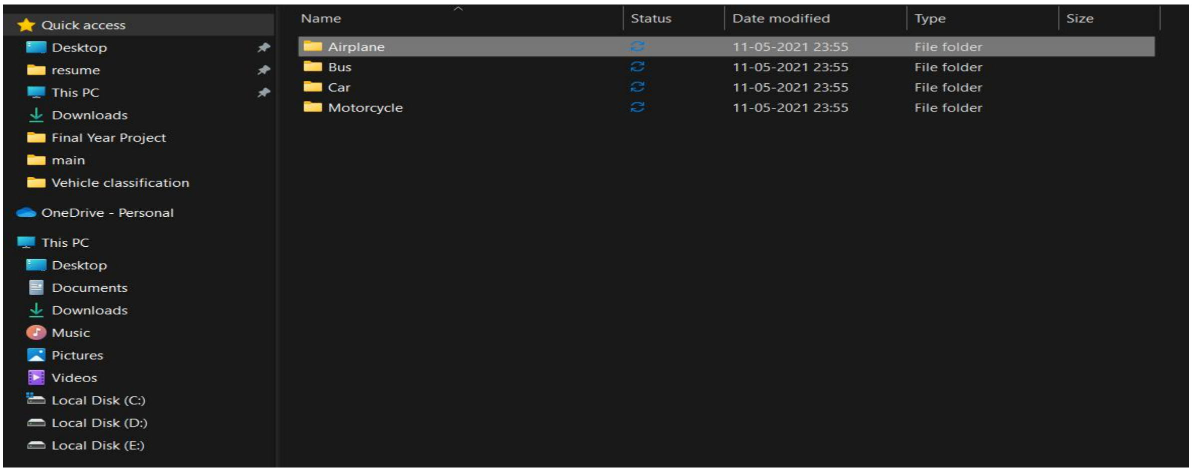
Procedure

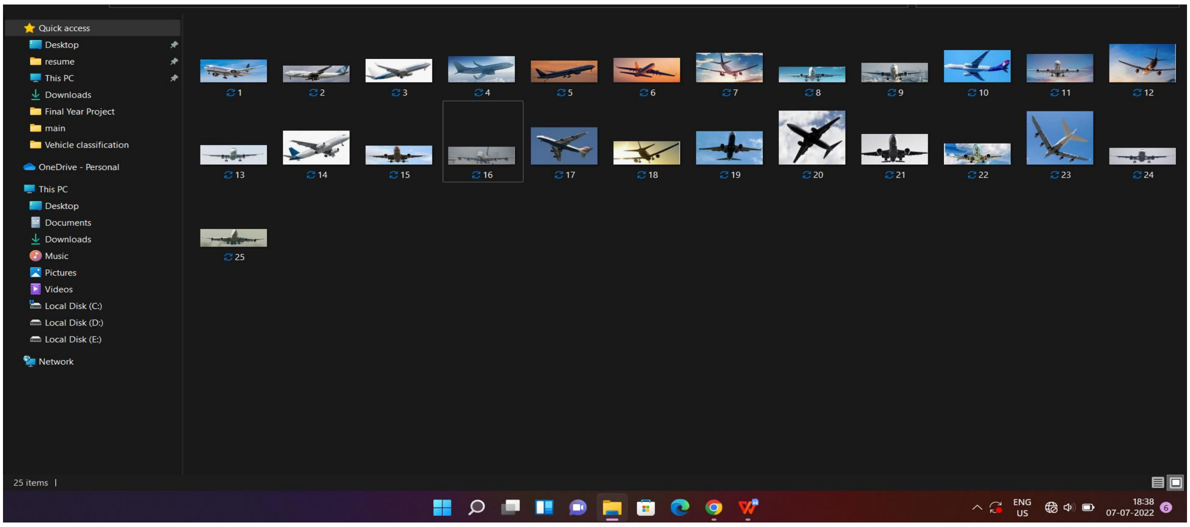
The procedure is pretty simple. Actually, the origin of idea of transfer learning has been originated from the need of simplicity and faster deployment. It takes long to design a convolutional neural network and even longer to optimize it. On the other hand, using transfer learning, you can retrain an existing optimized convolutional neural network almost instantly. Here are the five steps to apply transfer learning to classify vehicle using GoogleNet

Load your vehicle dataset,

1. Split the dataset into training, validation and testing dataset,
2. Resize the images according to the input layer of the network,
3. Modify the feature learner and output layer of the GoogleNet,
4. Re-train the feature learner layer using your dataset.

Dataset





We have made 4 classes of data which are airplane,bus,car,motorcycle

Contains the images of there respective class which are downloaded by google.

Code

The concept of vehicle classification using convolutional neural network (CNN) has been implemented using two separate scripts. They are:

1. Training – To train the network
2. Test Network – To test the network

Code of Training Script:

```
Dataset = imageDatastore('Dataset', 'IncludeSubfolders',
true, 'LabelSource', 'foldernames');
[Training_Dataset, Validation_Dataset, Testing_Dataset] =
splitEachLabel(Dataset, 0.7, 0.15, 0.15);

net = googlenet;
analyzeNetwork(net)

Input_Layer_Size = net.Layers(1).InputSize(1:2);
Resized_Training_Dataset =
augmentedImageDatastore(Input_Layer_Size ,Training_Dataset);
Resized_Validation_Dataset =
augmentedImageDatastore(Input_Layer_Size ,Validation_Dataset);
Resized_Testing_Dataset =
augmentedImageDatastore(Input_Layer_Size ,Testing_Dataset);

Feature_Learner = net.Layers(142).Name;
Output_Classifier = net.Layers(144).Name;

Number_of_Classes = numel(categories(Training_Dataset.Labels));

New_Feature_Learner = fullyConnectedLayer(Number_of_Classes, …
            'Name', 'Vehicle Feature Learner', ...
             'WeightLearnRateFactor', 10,...
             'BiasLearnRateFactor', 10);

New_Classifier_Layer = classificationLayer('Name', 'Vehicle Classifier');

Network_Architecture = layerGraph(net);
```

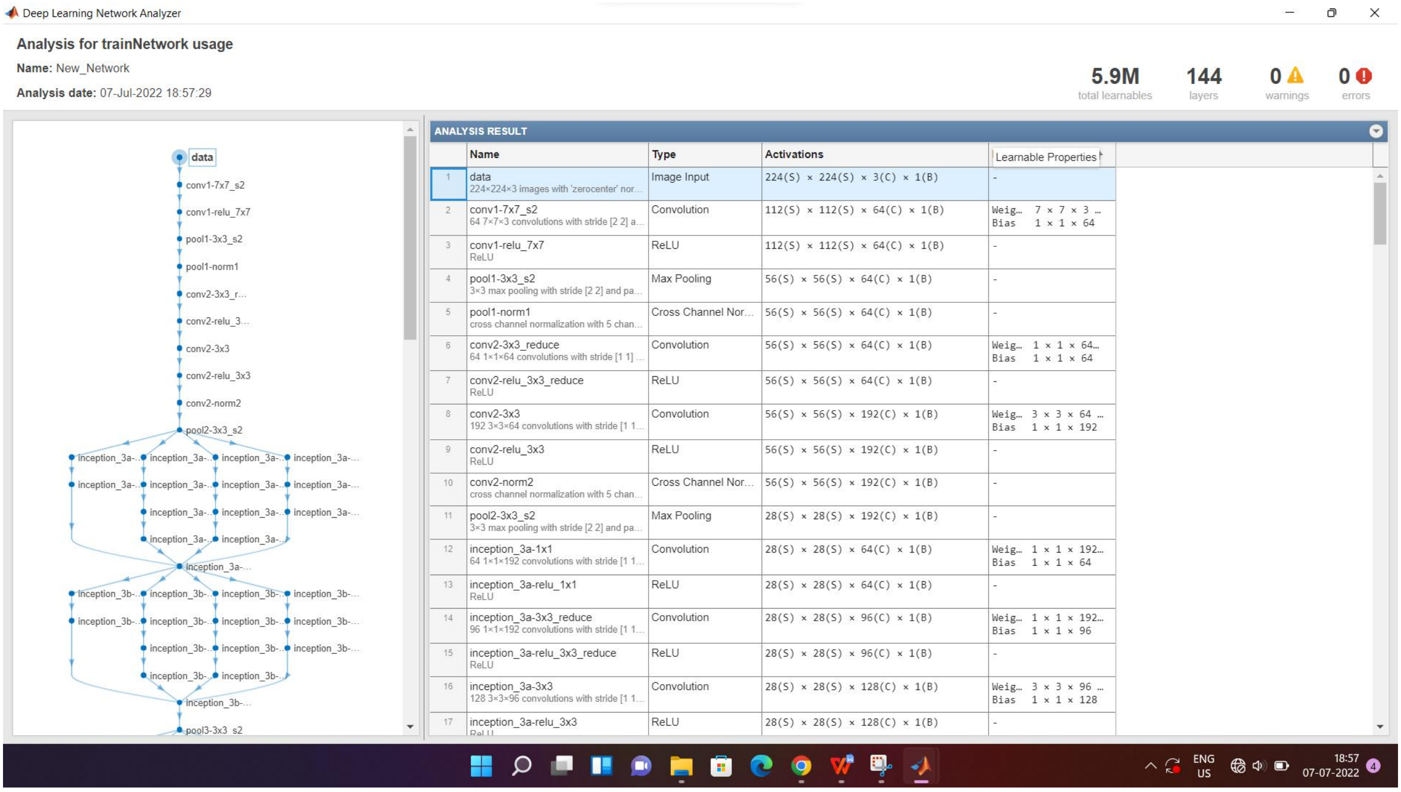
```
New_Network = replaceLayer(Network_Architecture, Feature_Learner,
New_Feature_Learner);

New_Network = replaceLayer(New_Network, Output_Classifier,
New_Classifier_Layer);

analyzeNetwork(New_Network)

Minibatch_Size = 4;
Validation_Frequency =
floor(numel(Resized_Training_Dataset.Files)/Minibatch_Size);
Training_Options = trainingOptions('sgdm', ...
        'MiniBatchSize', Minibatch_Size, ...
        'MaxEpochs', 6, ...
        'InitialLearnRate', 3e-4, ...
        'Shuffle', 'every-epoch', ...
        'ValidationData', Resized_Validation_Dataset, ...
        'ValidationFrequency', Validation_Frequency, ...
        'Verbose', false, ...
        'Plots', 'training-progress');

net = trainNetwork(Resized_Training_Dataset, New_Network,
Training_Options);
```

Code of Test Network Script:

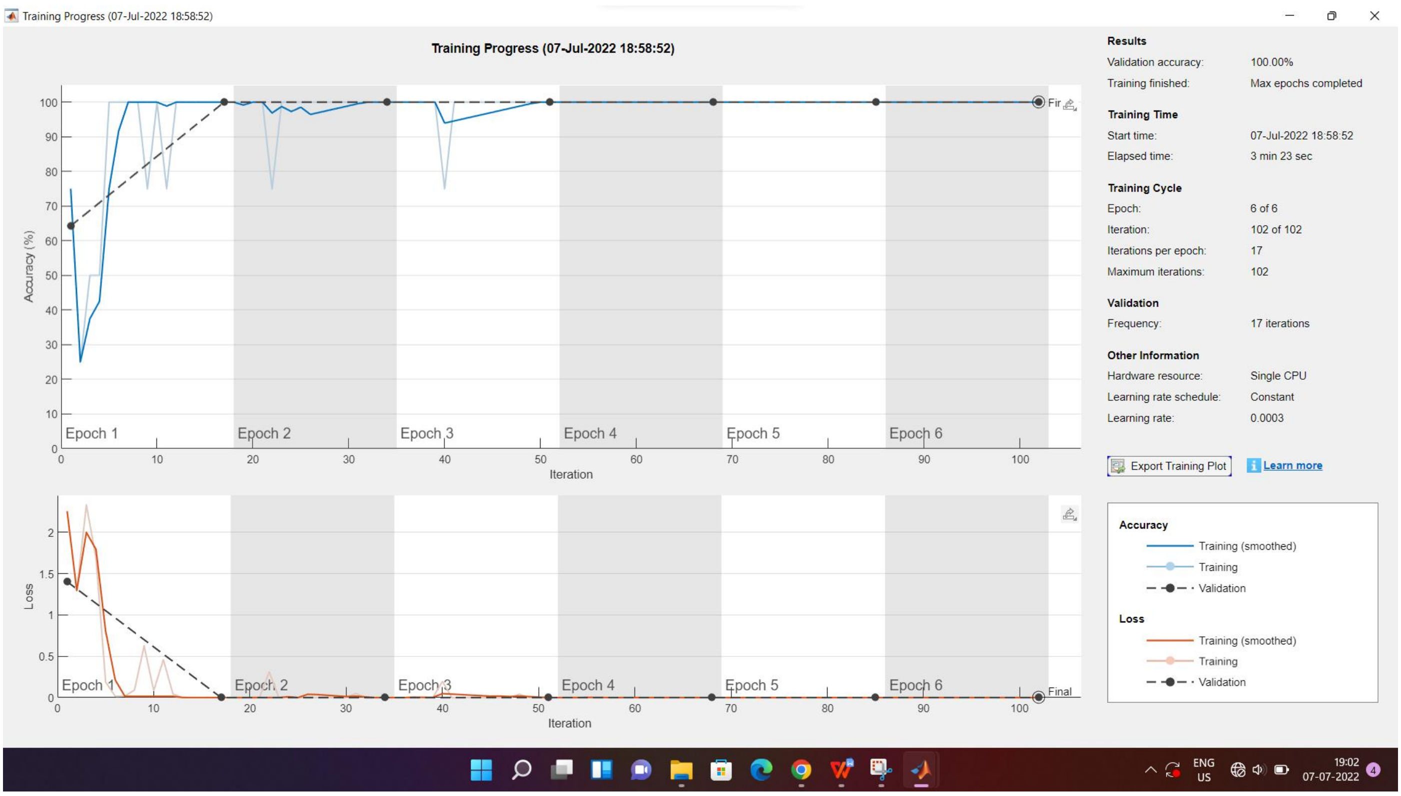
```
[Predicted_Label, Probability] = classify(net, Resized_Testing_Dataset);
accuracy = mean(Predicted_Label == Testing_Dataset.Labels);

index = randperm(numel(Resized_Testing_Dataset.Files), 4);
figure

for i = 1:4
    subplot(2,2,i)
    I = readimage(Testing_Dataset, index(i));
    imshow(I)
    label = Predicted_Label(index(i));
    title(string(label) + ", " + num2str(100*max(Probability(index(i), :)), 3) +
"%");
end
```

Screenshots:

Analysis Network



Training Progress



OUTPUT

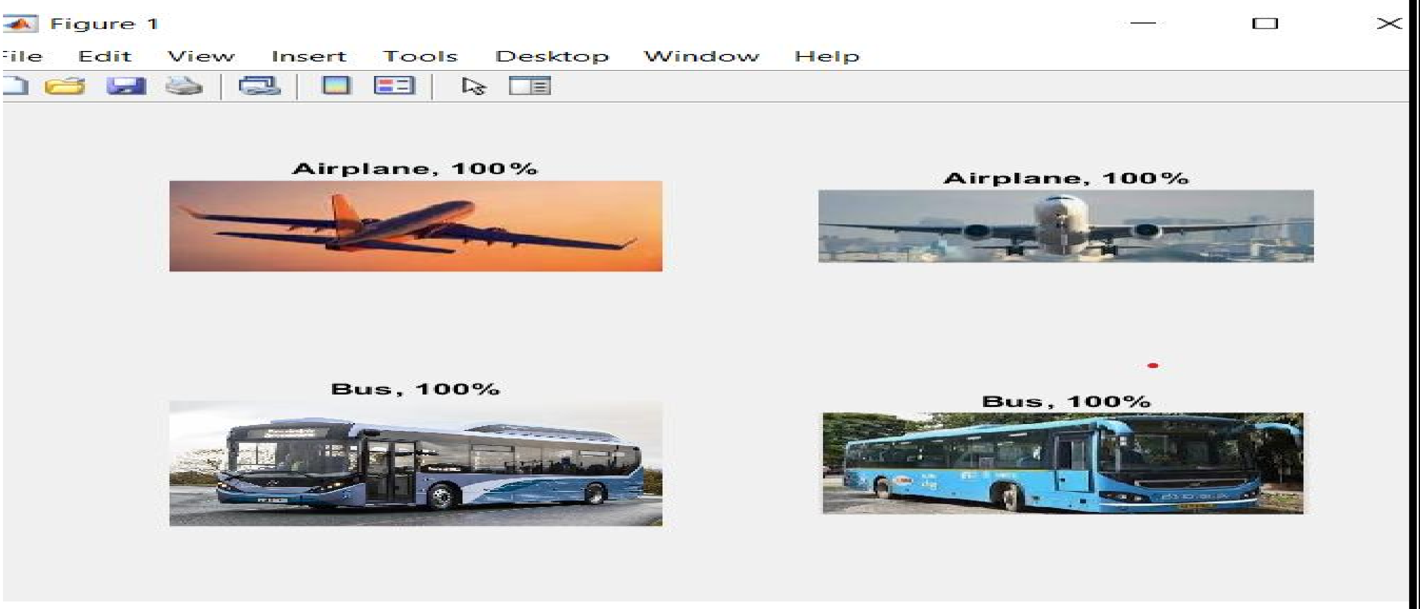


Figure 1
File Edit View Insert Tools Desktop Window Help
Airplane, 100%
Airplane, 100%
Bus, 100%
Bus, 100%

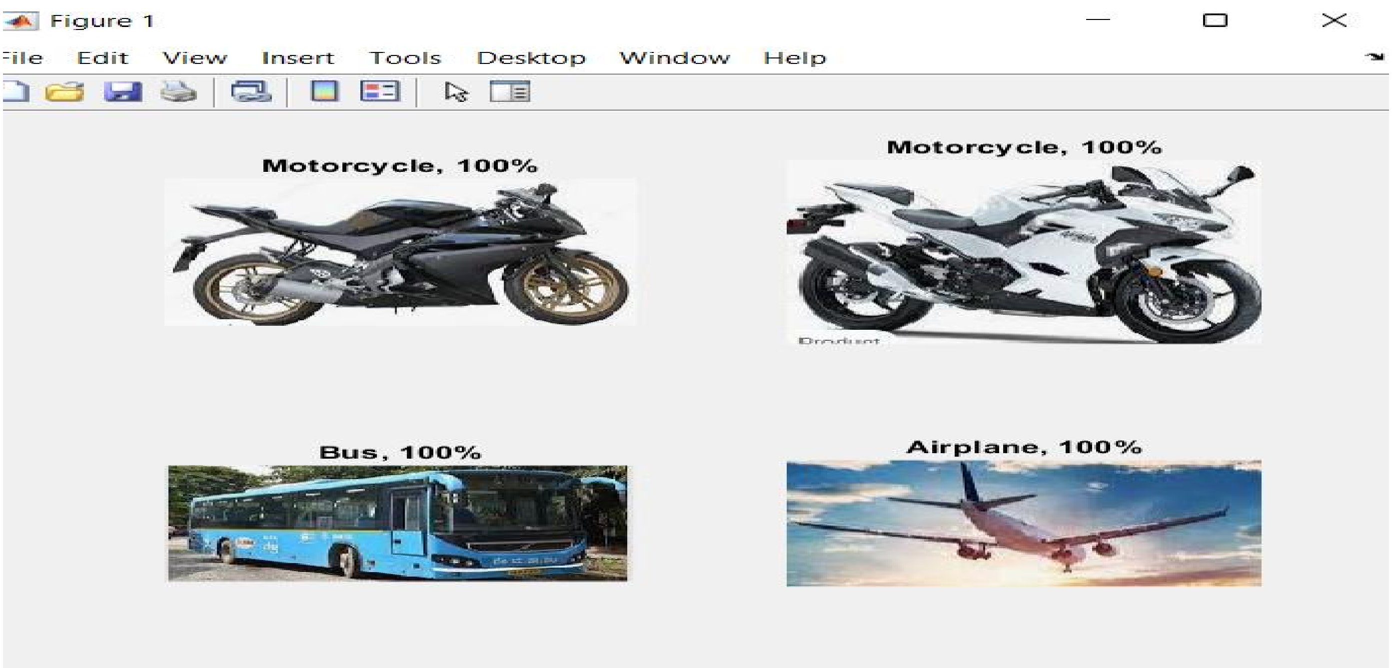


Figure 1
File Edit View Insert Tools Desktop Window Help
Motorcycle, 100%
Motorcycle, 100%
Bus, 100%
Airplane, 100%