**CAR-BRAND IMAGE CLASSIFICATION**

**CSD Project submitted in fulfilment of the requirements for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IIN**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

## Aryan Shiv Siddhabathula, 121910313029

## Naveen Chowdary, 121910313046



## DEPARTMENT

## OF

## Computer Science Engineering

## GITAM

## (Deemed to be University)

**VISAKHAPATNAM**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GITAM SCHOOL OF TECHNOLOGY**

**GITAM**

**(Deemed to be University)**



# DECLARATION

I/We, hereby declare that the project report entitled **Car Brand Image Classification** is an original work done in the Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of B.Tech. in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree or diploma.

Date: 26th October, 2022

**Registration No(s). Name(s) Signature(s)**

**121910313029 Aryan Shiv Siddhabathula**

**121910313046 Naveen Chowdary**

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# GITAM SCHOOL OF TECHNOLOGY

**GITAM**

**(Deemed to be University)**



# CERTIFICATE

This is to certify that the project report entitled **Car Brand Image Classification** is a bonafide record of work carried out by **Aryan Shiv Siddhabathula (121910313029), Naveen Chowdary (121910313046)** students submitted in fulfilment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.

|  |  |
| --- | --- |
| **Project Guide** | **Head of the Department** |

**Abstract -**

Cars have been a major asset in influencing the growth of automobile sector, with an increase in automobile, it has eventually resulted in giving humans more freedom and accessibility to jobs and services. Car Brand Classification helps us in identifying vehicles from diversified brands. From this project, if a user wants to know which car belongs to which brand through images, he can be able to identify using this prediction model.

Let’s look into the process of building this prediction model:

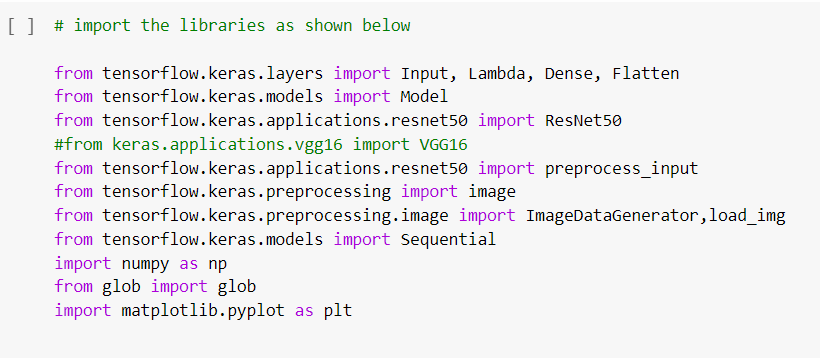
1. Build a Deep Learning Model
2. Using HTML, CSS, Java Script, we create a webpage.
3. Using FLASK, we build an interface between the prediction model and webpage for emphasizing on user-friendly features.

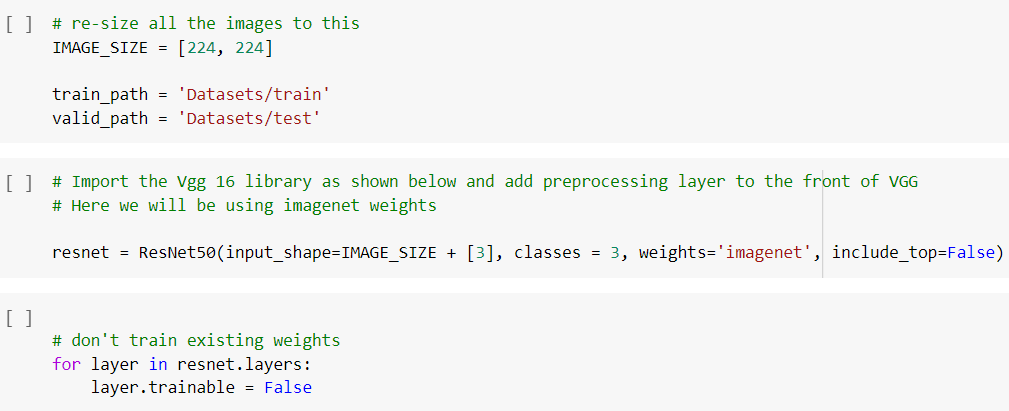
We import libraries such as TensorFlow, Keras, Matplotlib, Resnet etc.

Resnet-50 is a CNN Deep Learning Method which has around 50 layers to train our model. We can load more than a million images from various sources or from the ImageNet Database. This helps us to classify images into various categories.

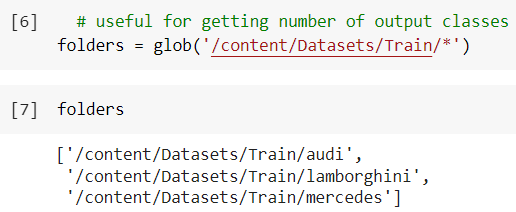
VGG-16 is a CNN (Convolutional neural network that is around 16 layers deep. Similar to Resnet-50, we can load more than a million images from various sources or from the ImageNet Database and classify based upon our needs.

TensorFlow is an end-to-end machine learning platform that can be used while working on a variety of tasks. It solely focused on training and inference of deep neural networks. Keras operates as an interface for the following TensorFlow library. It provides an interface for Artificial Neural Networks.



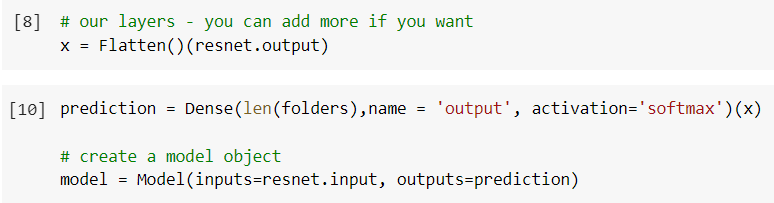


glob is useful for returning a file path or location if it matches a certain pattern. You can use glob to search for files whose filenames match a specific pattern using wildcard characters.

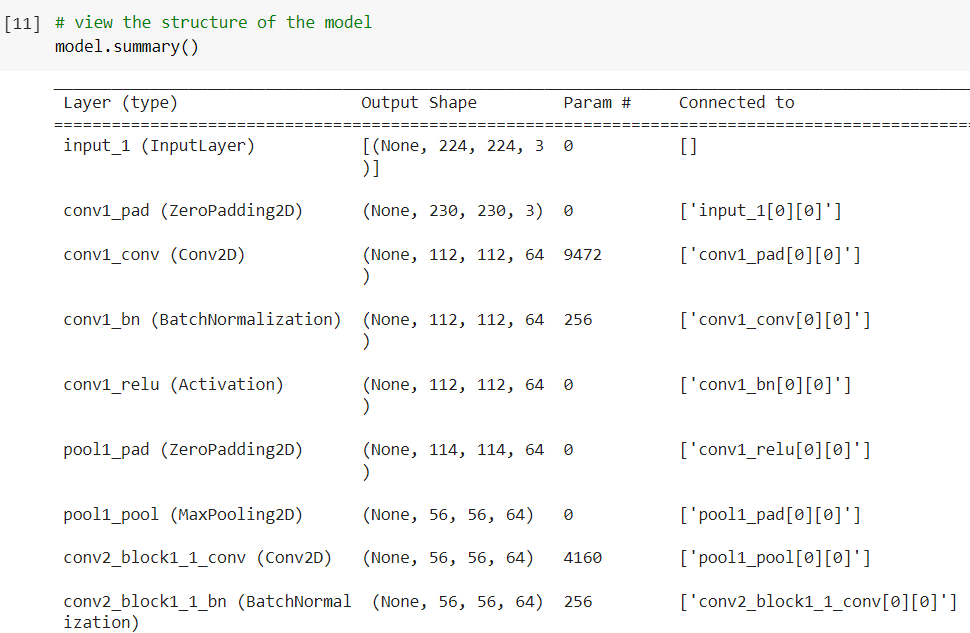


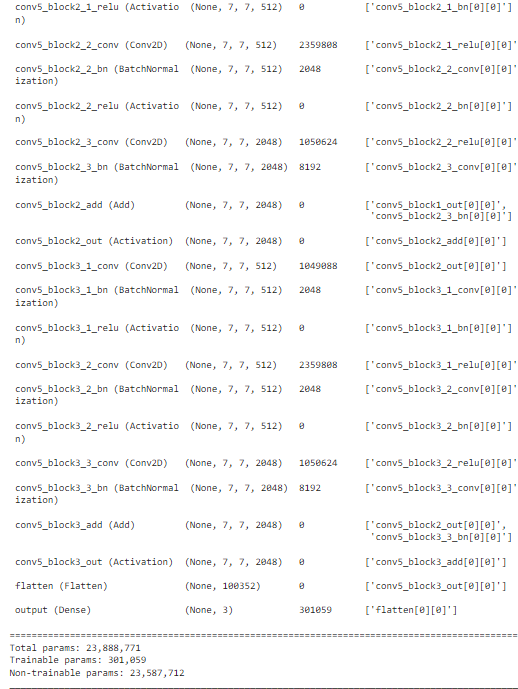
In general, the images you provide as input tend to have varying dimensions. Having a constant size input is important. For images, we need to concentrate on feeding the model with data that does not have various factors, so we use the Flatten() method to avoid various dimensions. Flattening converts the data into a one-dimensional array for input to the next level. Smooth the output of the convolutional layers to create a single long feature vector. and is connected to a final classification model called a fully connected layer.

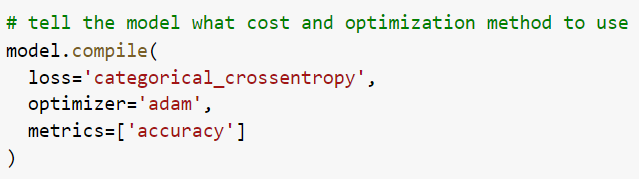
Dense function

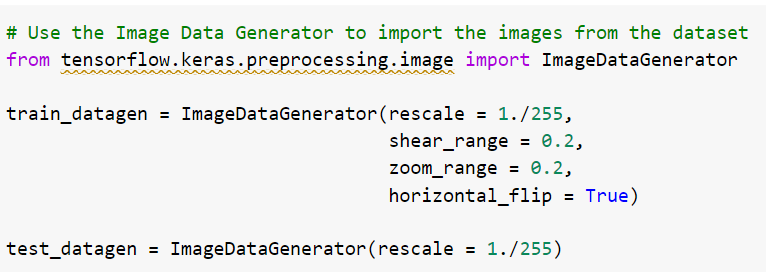


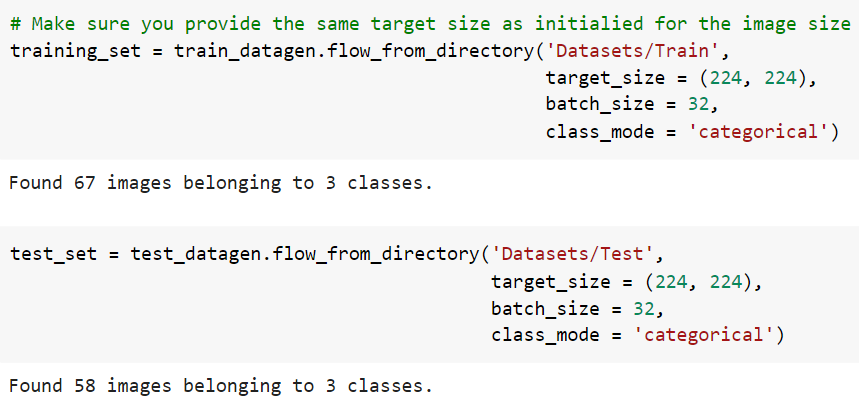
For Keras layers, the input form is typically the form of input data provided to the Keras model during training. The model cannot recognize the shape of the training data. The shape of other tensors (layers) is calculated automatically. We use  
pooling layers to reduce the dimensionality of the feature map. This reduces the number of parameters to learn and the amount of computations performed by the network.  
The pooling layer summarizes the features present in the region of the feature map produced by the convolutional layer. Therefore, more operations are performed on the merged features instead of the precisely placed features produced by the convolutional layer. This makes the model more robust to variations in feature positions in the input image.  
The activation function determines whether to activate the neuron. This means that the prediction process uses simpler mathematical manipulations to determine if a neuron's input to the network is significant.



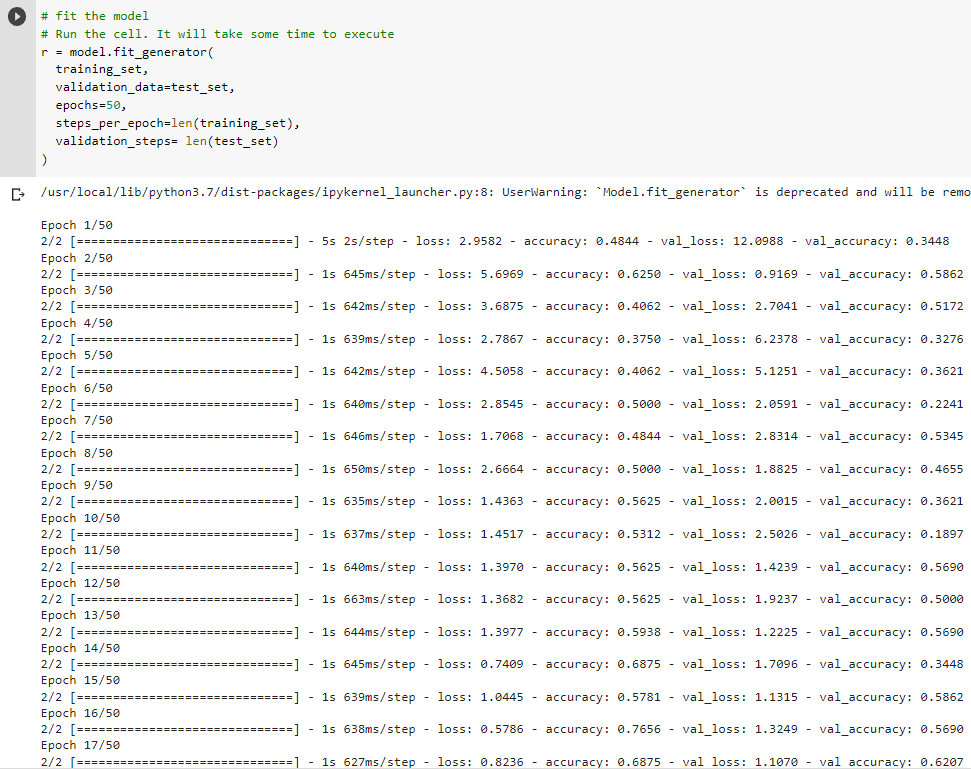


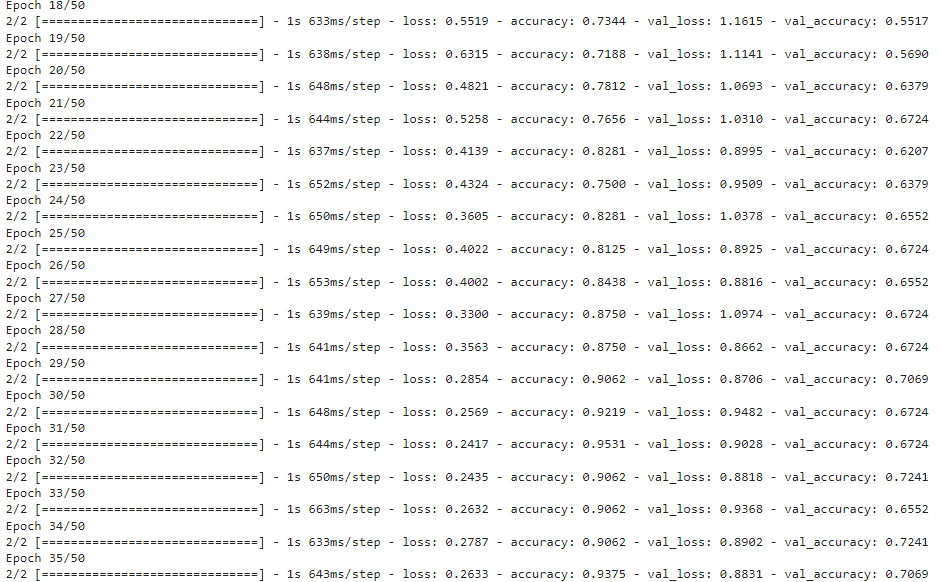


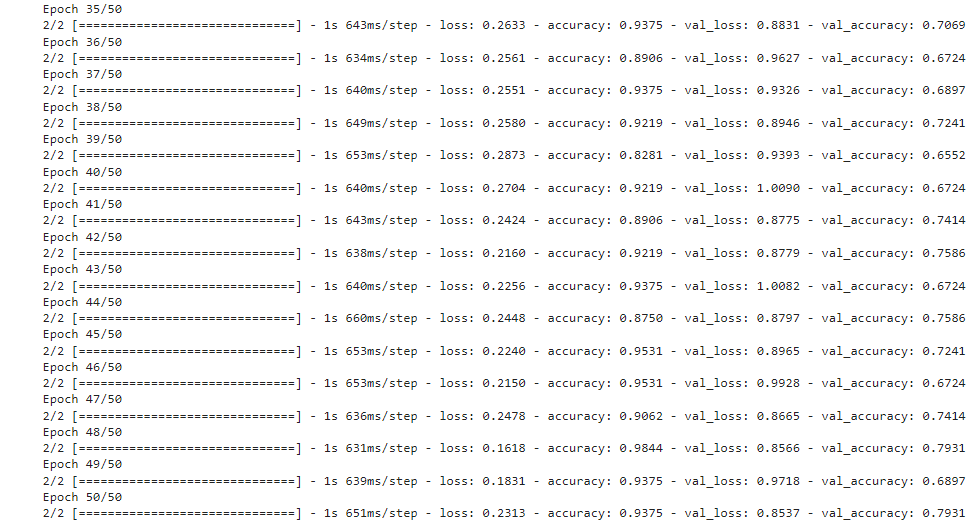
A loss function is a way of evaluating how well a machine learning algorithm models an imaginary data set. In other words, the loss function is a measure of how good the model is at predicting the expected outcome.  
Metrics are functions used to evaluate model performance. The metric function is similar to the loss function, except that the metric's evaluation results are not used when training the model. Note that any loss function can be used as a metric.

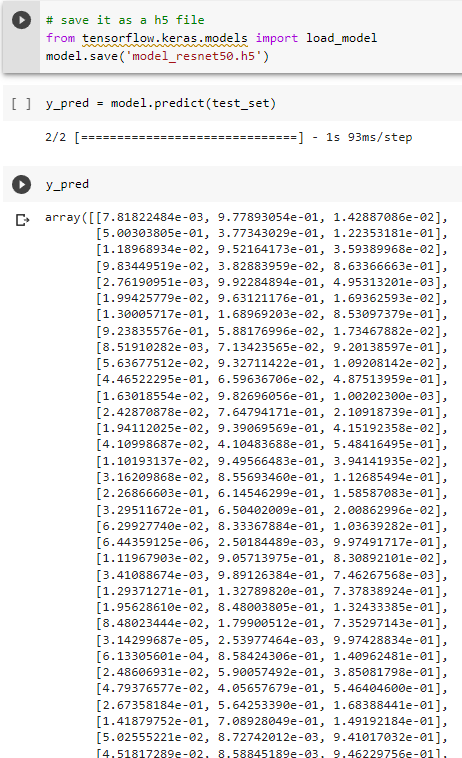
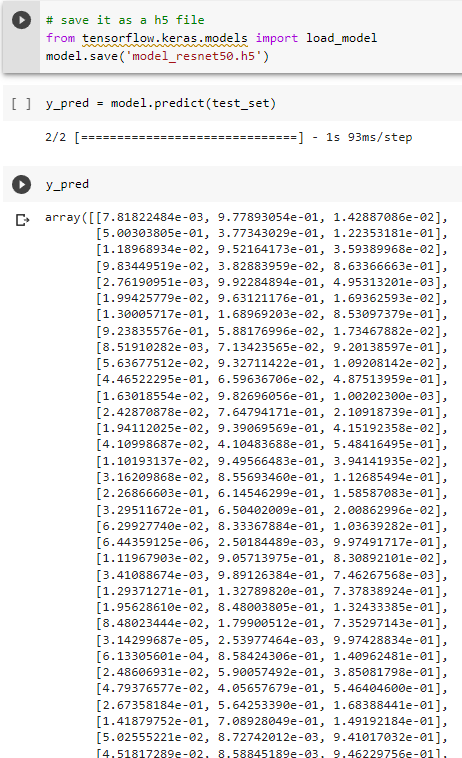


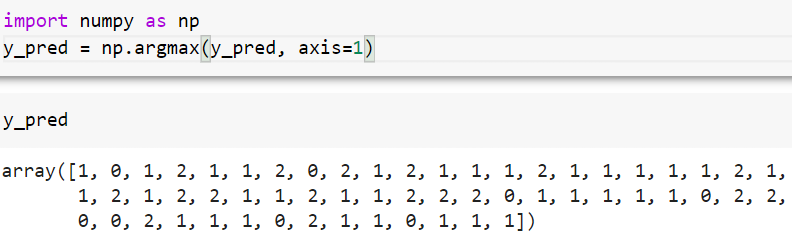
Function fitting is the process of training a neural network with a set of inputs to produce a set of related target outputs. After building the network with the desired hidden layers and training algorithm, you need to train the network with a set of training data.

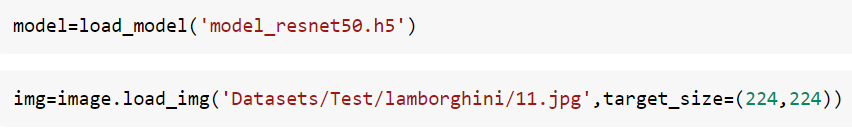


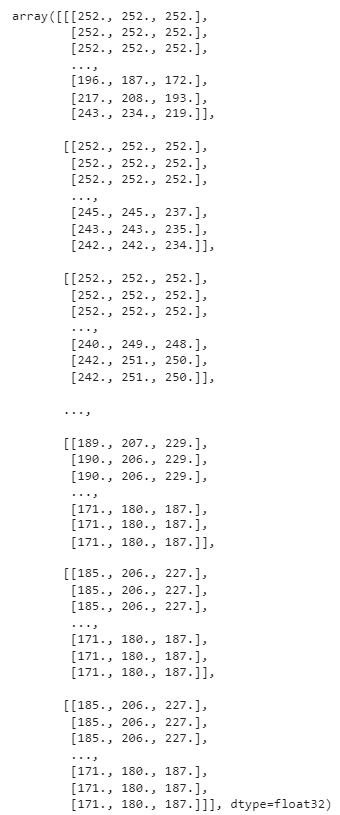
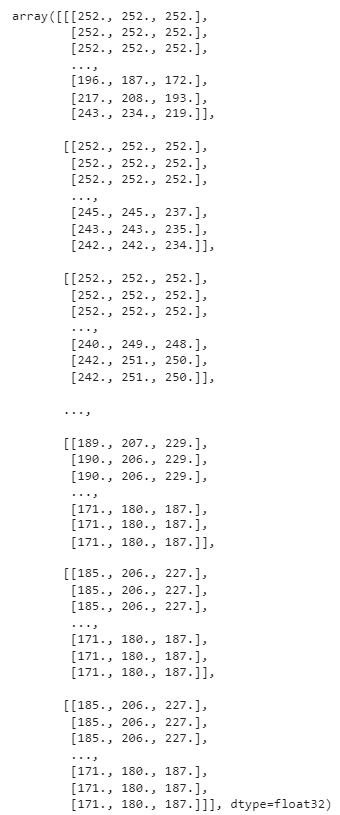


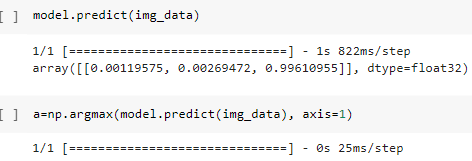
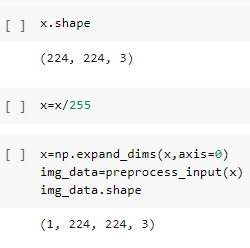


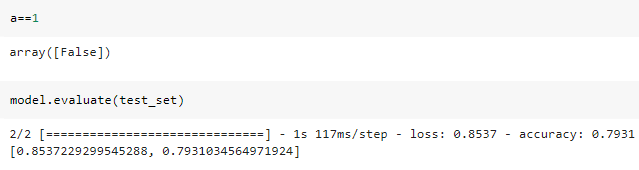




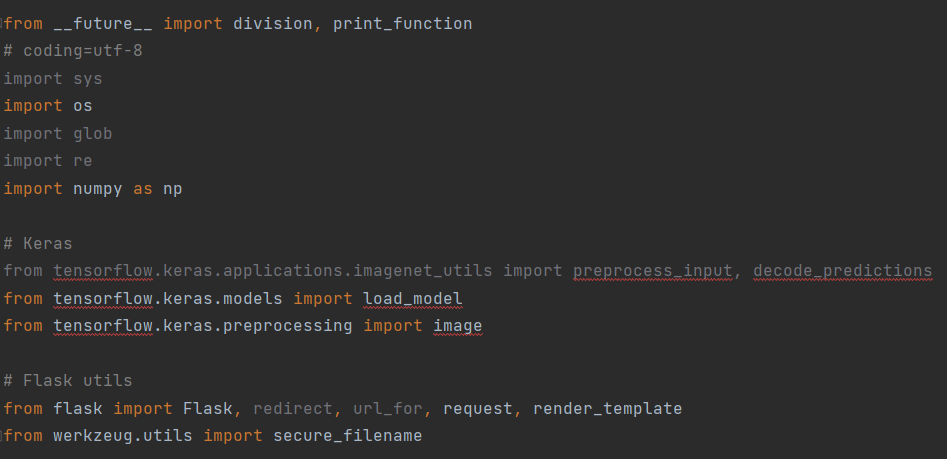
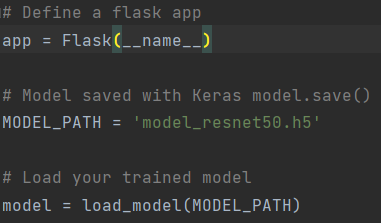


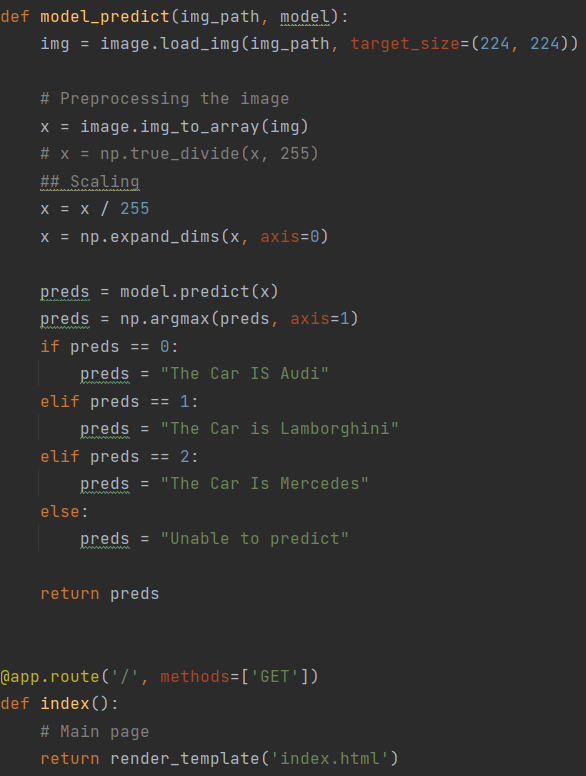
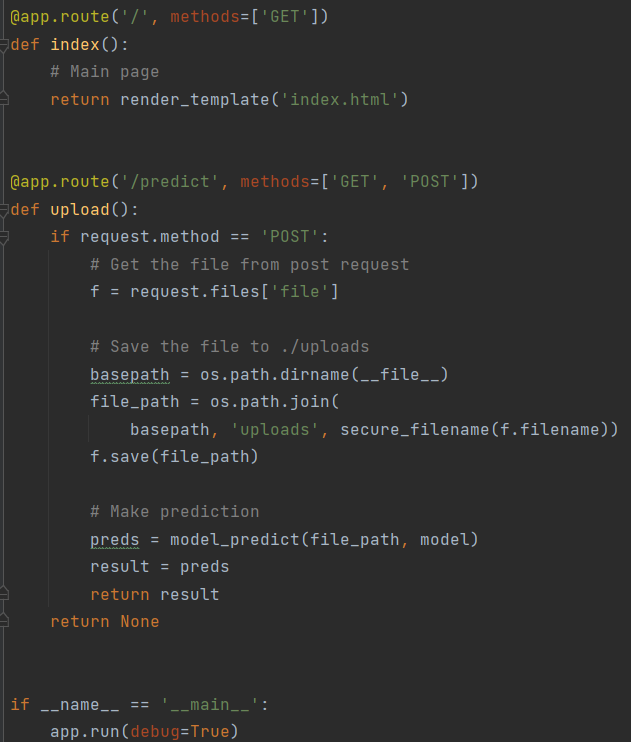






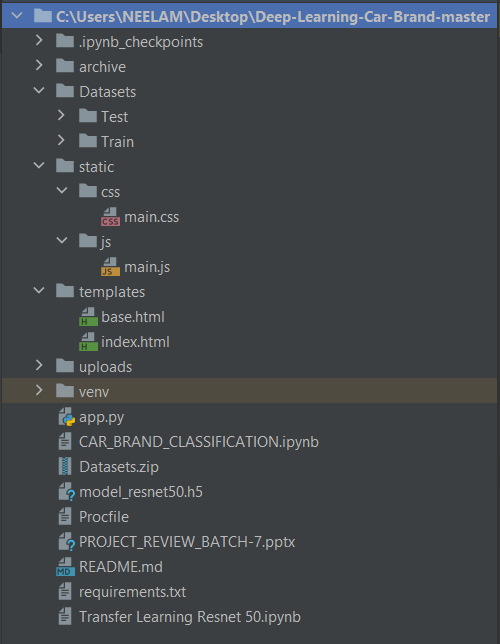
**APP.py – (Creating Interface using Flask ) -**

Now, before we start executing, we need to focus on saving our files in the specific format.

Using Command Prompt, we start executing our flask-related application python program, which eventually creates an application interface for executing the program. It provides us an IP Address, which when clicked in Google Chrome leads us to the following interface. On giving input through files, we observe that the model is able to predict that the following image is a Lamborghini, Audi or a Mercedes in a correct manner.



**OUTPUT IMPLEMENTATION AND RESULTS -**

