**DATA MINING – CS634**

**­Final Project**

**TRANSFER LEARNING – ResNet50**

**­­­(TensorFlow + Keras)**

**Submitted By – PARTH SHARMA**

**UCID – 31543562 Mail ID –** [**ps75@njit.edu**](mailto:ps75@njit.edu)

Brief Summary -

Dataset that I am having is having following structure:

1. Train Data 2) Test Data

* Audi - Audi
* Lamborghini - Lamborghini
* Mercedes - Mercedes

I have divided this project working into 2 parts.

In Part-1,

1. ResNet50 model has been built without using existing weights that comes with Transfer Learning model.
2. Later, model has been created and trained on Train Data.
3. Model has been saved.

In Part-2,

Saved model has been loaded again and then used to predict brand class name of a new example image.

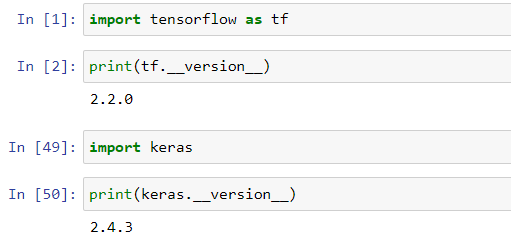
**Steps Included**

1. Importing Tensorflow & Keras
2. Importing necessary modules from Keras
3. Importing other libraries
4. Setting Path for Train data & Test Data
5. Setting up ResNet 50 network pre-trained over ImageNet weights
6. Form our model by providing our own prediction layers to pre-trained network.
7. Summary of model
8. Compile the model
9. Fit the model using training data
10. Plot ‘Train Loss’ vs ‘Validation Loss’
11. Plot ‘Training Accuracy’ vs ‘Validation Accuracy’
12. Saving the model
13. Load the saved model
14. Load a new sample image
15. Convert img into an array
16. Normalize array
17. Preprocess your array
18. Predict the class of Car-Brand of test image using the saved model.

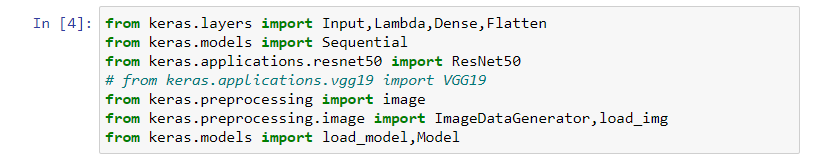
ENTIRE WORKFLOW

Step 1 – Import TensorFlow & Keras

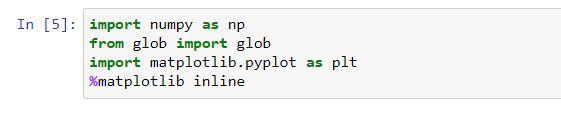
* Check version of TensorFlow & Keras
* (NOTE – TensorFlow version should be atleast 2.2)



Step 2 – Import necessary modules from Keras.

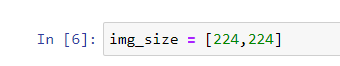


Step 3 - Importing other libraries

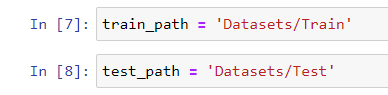


* Glob stands for Global
* It is used to return all the file paths that match a specific pattern.

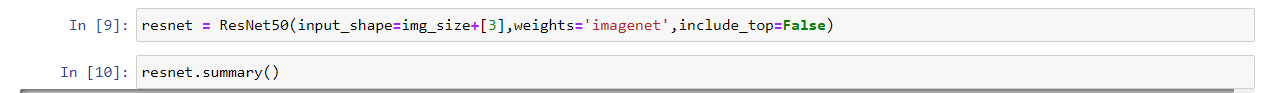
Step 4 – Setting up size for an image



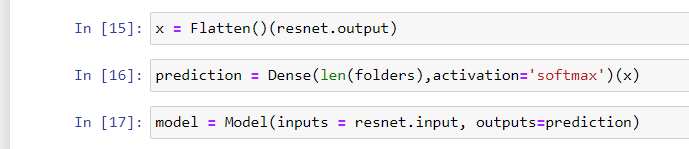
Step 5 - Setting Path for Train data & Test Data



Step 6 - Setting up ResNet 50 network pre-trained over imagenet weights



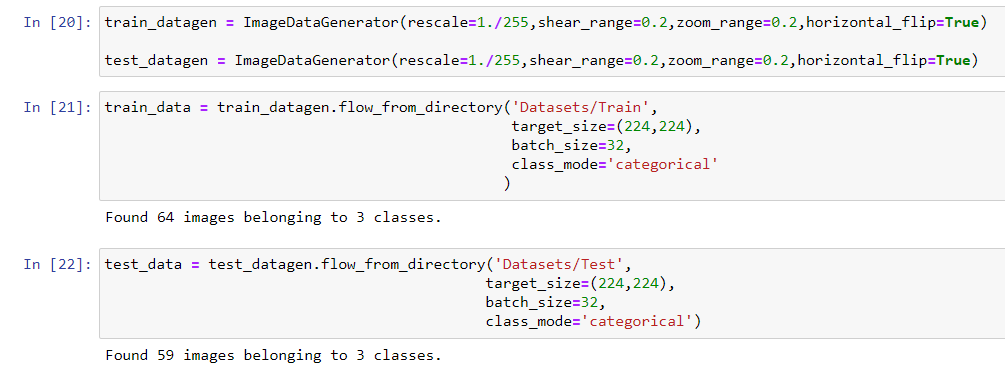
Step 7 – Form our model by providing our own prediction layers to pre-trained network.



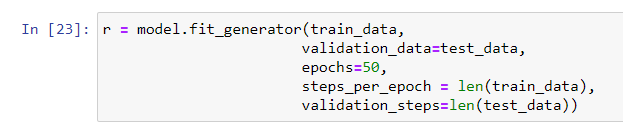
Step 8 – Compile Model

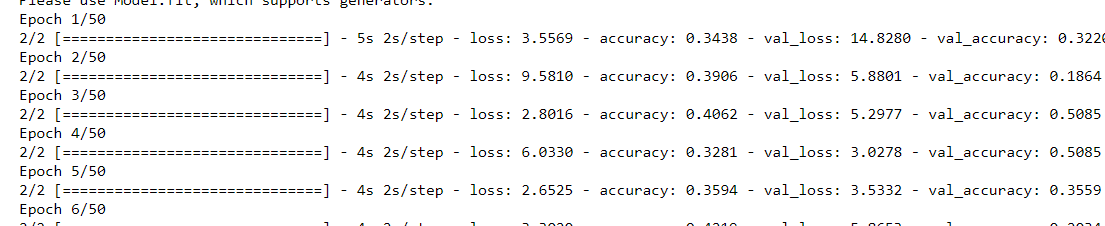


Step 9 – Setting up Training Data & Test Data



Step 10 – Fit the model over Training Data

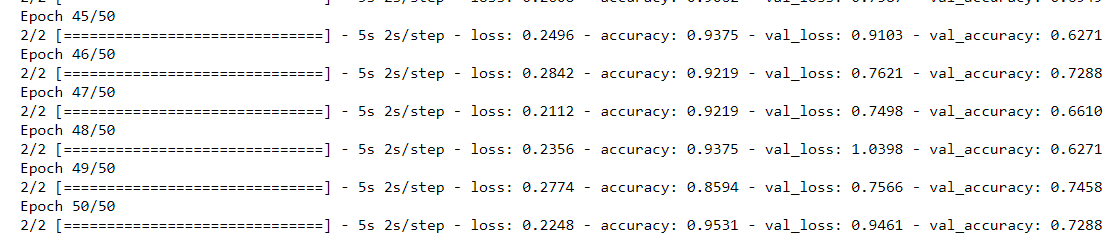




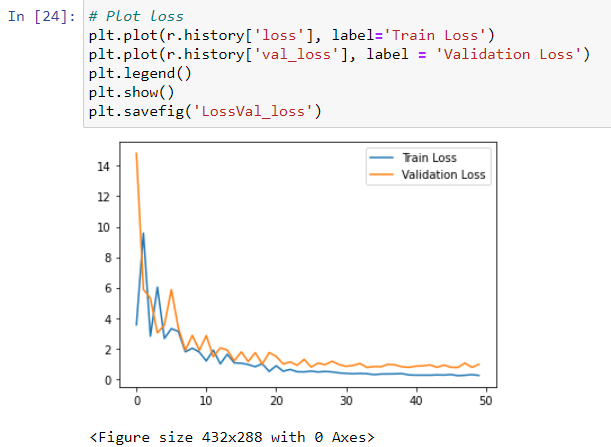
………

………

……..



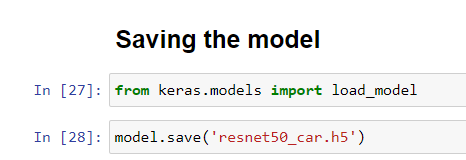
Step 11- Plot Loss



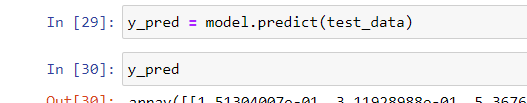
Step 12 – Plot Accuracy

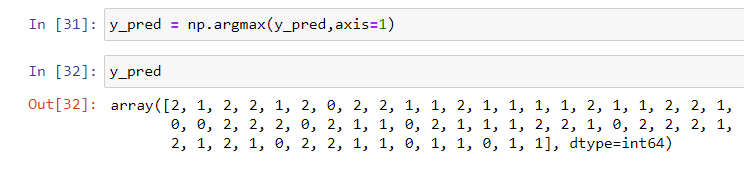


Step 13 – Save the Model

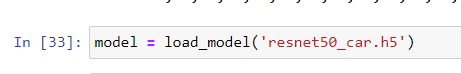


Step 14 – Model Prediction using Test data

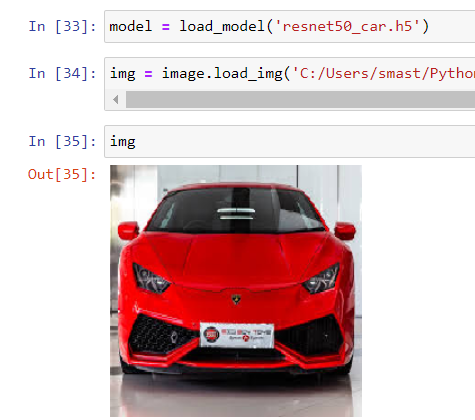




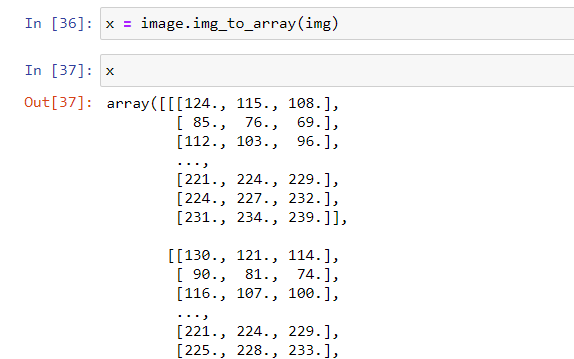
Step 15 – Loading saved model



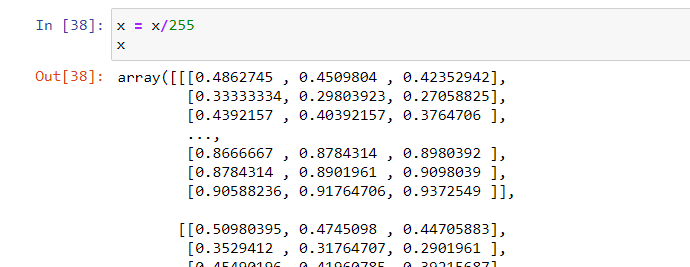
Step 16 – Loading a sample image to make a prediction

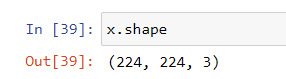


Step 17 – Converting ‘img’ into array

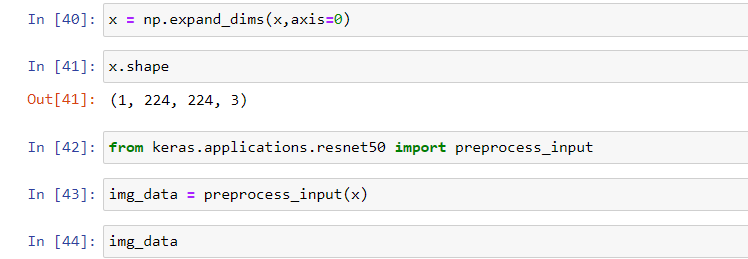


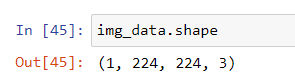
Step 18 – Normalize img\_array to keep values in (0-1) range.



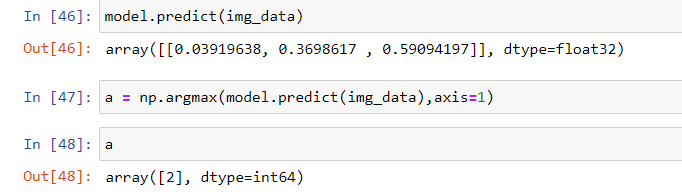


Step 19 – preprocessing image array





Step 20 – Predict brand Class for car image

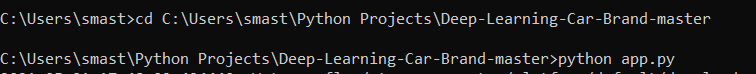


Step 21 – Created Flask App (app.py file)

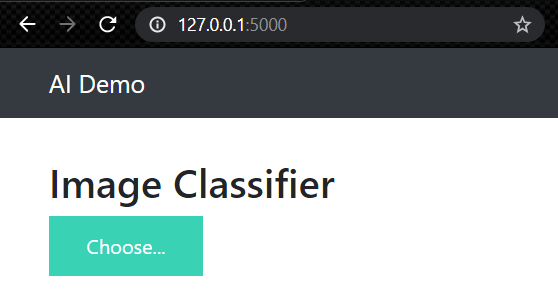
Here we will be choosing a random image file on our Flask App and our saved model will be giving its prediction over chosen image.

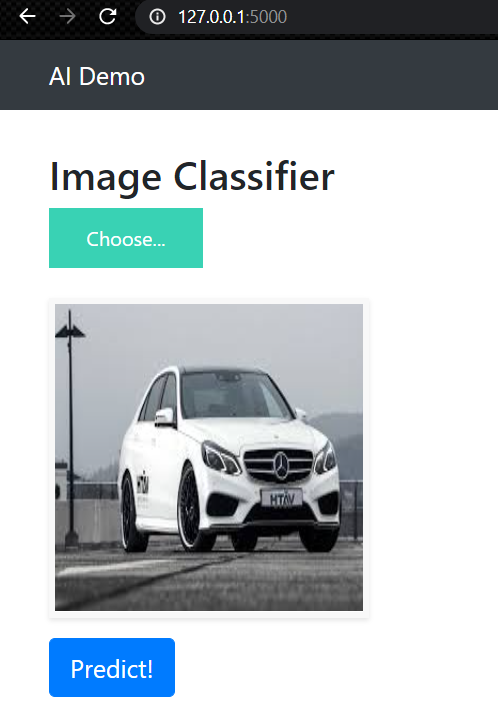
( Build this just for expanding my skills, otherwise there was no need of this Flask App to be included in this project.)

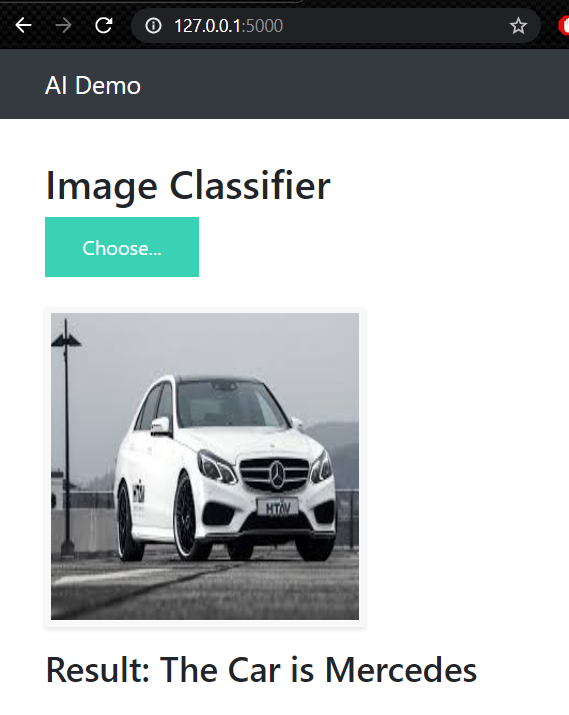
I felt this Flask App will provide more user-centric approach to the created model.

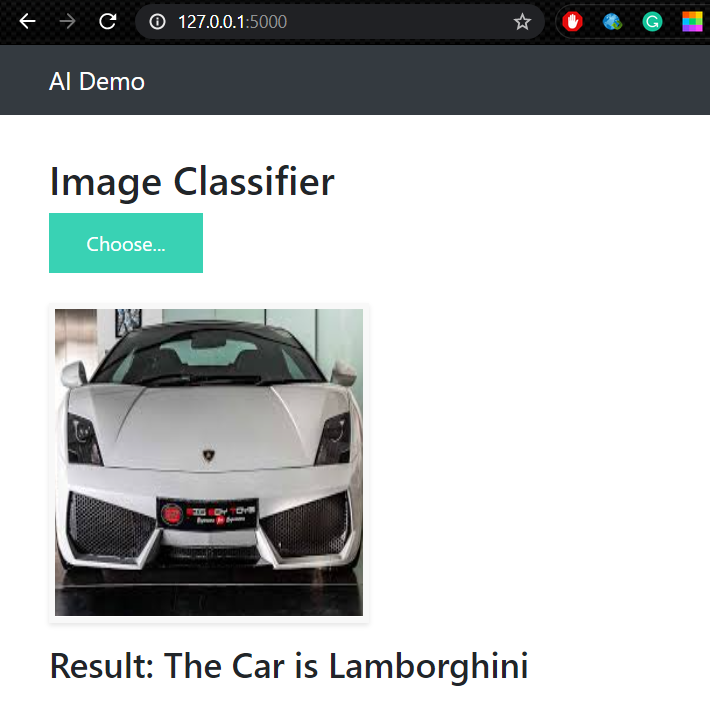


**RESULTS**









NOTE – 1) To run this program as a python file in your local env use – Resnet50\_Final\_Project.py

Command – python Resnet50\_Final\_Project.py

NOTE – 2) To run Flask app, use command – python app.py

1. Command to create Virtual Env with TensorFlow & Keras.



2. Install Tensorflow in your environment

- conda install tensorflow

- conda install keras

3. To Check all your libraries present in virtual env

- pip freeze

