CSEE5590/490:

Python and Deep Learning Programming

(2018 Fall)

*Python Project Report*

Submitted On:

04 December, 2018

Submitted By:

Fatema Hasta (Class ID: 10)  
Vinay Jaibheem (Class ID: 11)  
Farid Uddin Ahmed (Class ID: 02)  
Zarin Tasnim Sandhie (Class ID: 26)



LIST OF THE DOCUMENTATIONS:

1. Author
2. Objective
3. Features
4. Input/ Output Screenshots
5. Implementation & Code Snippet
6. Evaluation
7. Conclusion
8. Links
9. References

AUTHORS

This report contains all the documents for the Group project. The assignment was done by Fatema Hasta (Class ID: 10), Vinay Jaibheem (Class ID: 11), Farid Uddin Ahmed (Class ID: 02) and Zarin Tasnim Sandhie (Class ID: 26), all of them are students for the course Python/Deep learning (CSEE5590/490, Fall 2018) at the University of Missouri-Kansas City (UMKC).

OBJECTIVE

The objectives of this Project is:

* To design a web application for the classification of the required type of food.
* It will take input in the form of an image.
* This image can be fed to the model that detects the type of food and classifies the image either into Soup, Bread, Dairy product, etc.
* Based on this category, particular feature will be appeared to the user.

METHODOLOGY

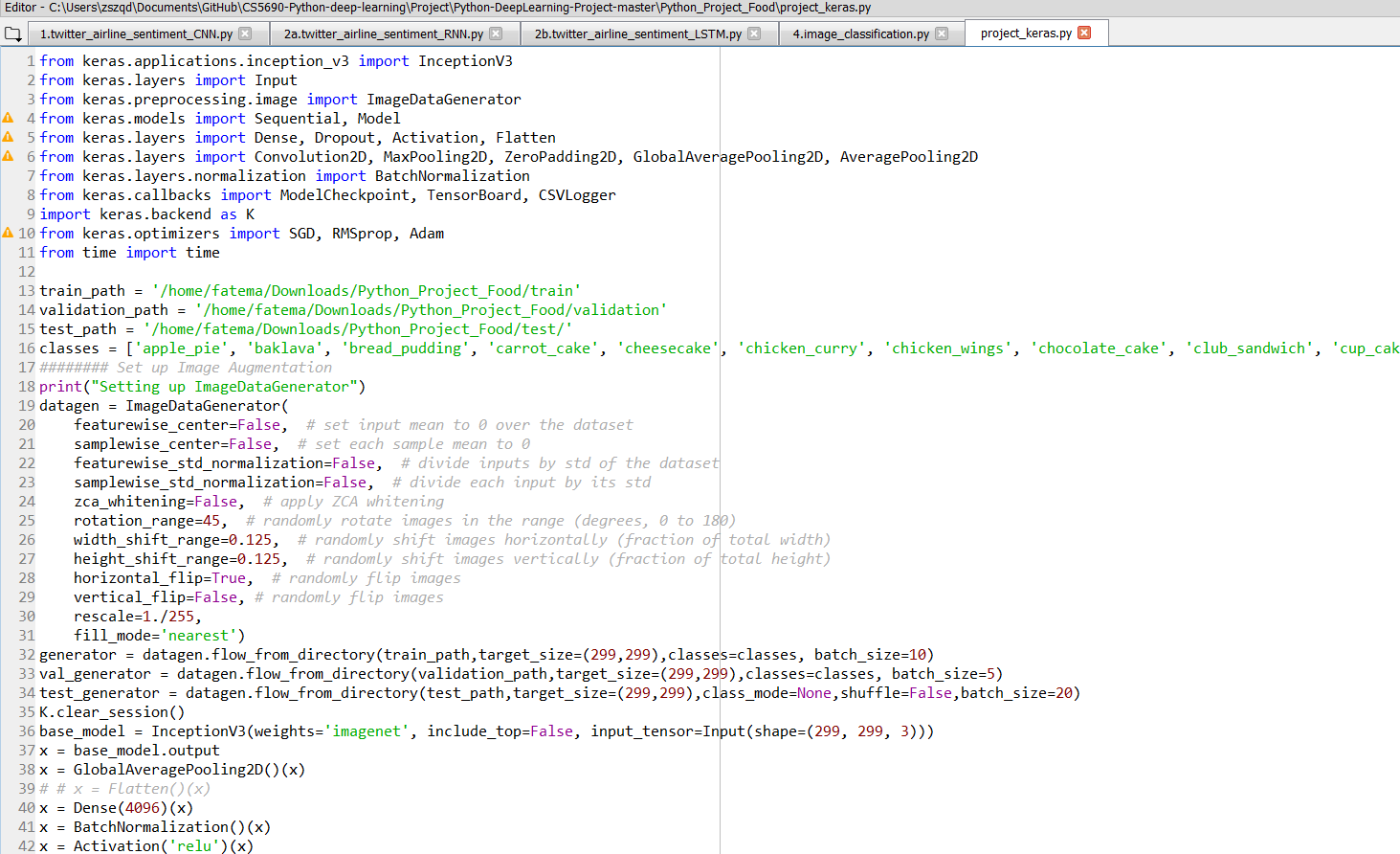
* In order to detect the type of food, there will be a food dataset categorized into Soup, Bread, Dairy product, etc. For our case,
* Convolutional Neural Networks (CNN) can be used to classify the new image of a food into one of these classes.
* Based on the class, Google Places API can be used to find places or restaurants where the user can go have this food. By utilizing Edamam Food API, nutrition details of the food is identified.
* The datasets can later be extended into a more elaborate dataset.

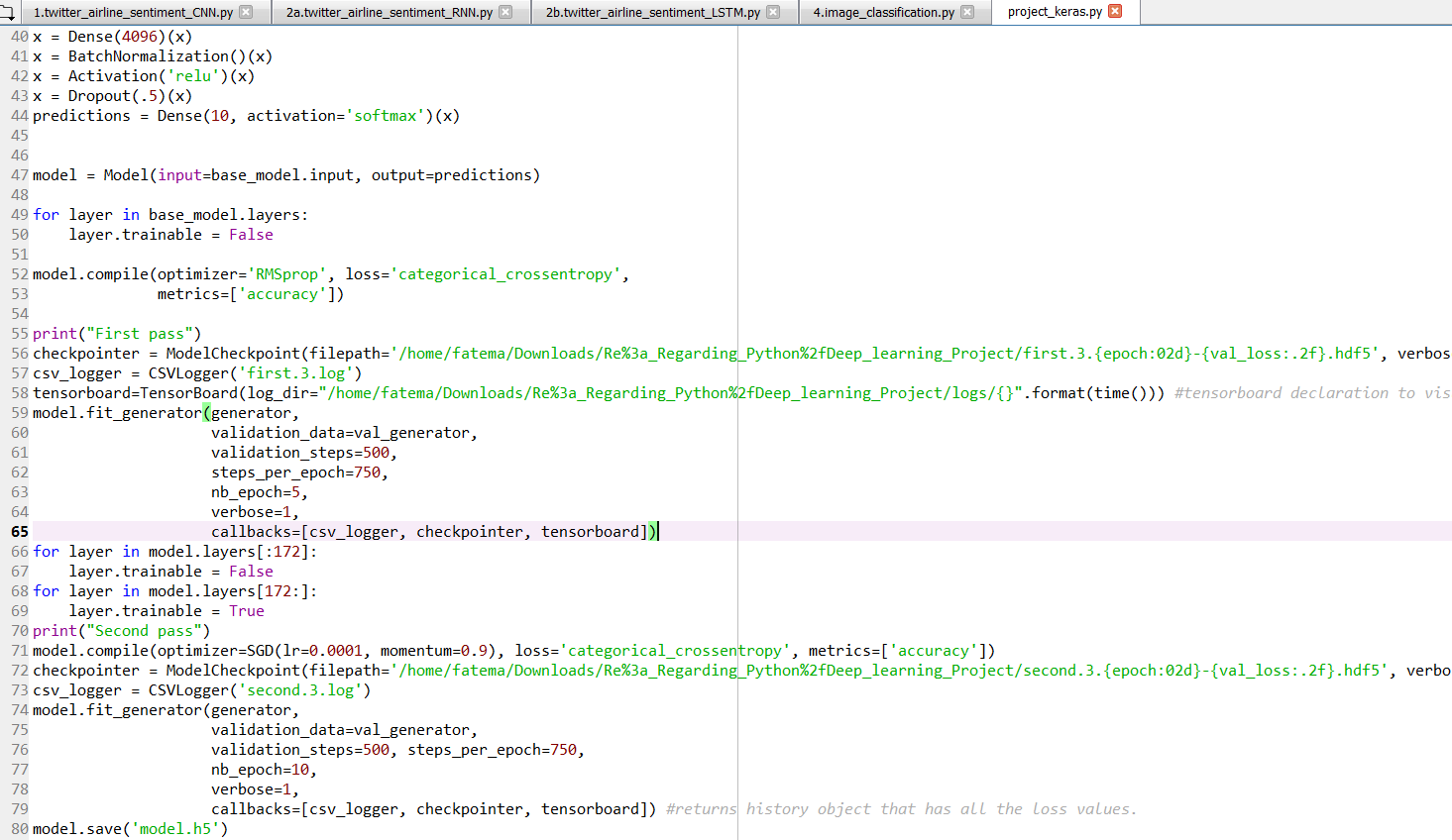
CODE EXPLANATION

* For our project, we have taken 10 classes from “Food-101” dataset.
* The classes we have used are: ‘apple\_pie', 'baklava', 'bread\_pudding', 'carrot\_cake', 'cheesecake', 'chicken\_curry', 'chicken\_wings', 'chocolate\_cake', 'club\_sandwich' and 'cup\_cakes'.
* Each class has 1000 total pictures.
* We have created 2 different folders named “train” and “validation” each of those contain 10 folders for each classes.
* The train folder is used for training dataset and each of the class folders inside train folder contains 750 pictures.
* The validation folder is used for validating the dataset and each of the class folders inside the validation folder contain 250 pictures.
* The test folder contain 20 random pictures from different classes which are later used for testing the model.
* All the images are augmented at first in which procedure the pictures are rotated, shifted, whitened etc. to increase the number of data in our dataset.
* Then inception is used as a pre-trained model.
* For the optimization, two different types of optimizers are used in two different pass.
* For the first pass, we used RMSprop and 5 epochs. The optimization is run for the whole model.
* For the second pass, SGD optimizer is used and 10 epochs is used. Here, expect the last two layers, all the other layer are frozen by making it false. The optimization is done only for the last two layers.
* Then the model is saved.
* This model can later be loaded and tested with test images.
* When we predict the name of an unknown food class from the model, this predicted name is saved in a .csv file.
* Then we pass this food name from the csv file to Edamam Api.
* Edamam Api has two features: one is to extract recipe and another is to extract calorie and nutrition chart from the recipe.
* At first, we extracted the recipe and then using that recipe, we extracted the calorie and nutrition.

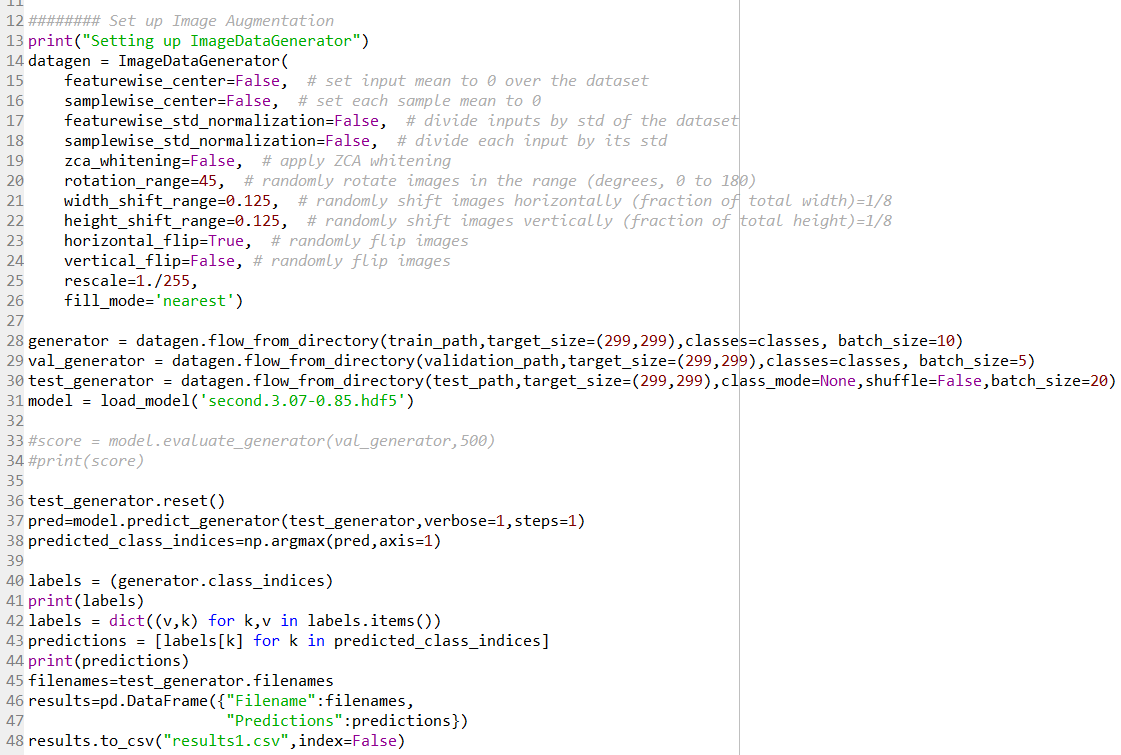
CODE SNIPPETS

Code for Image Classification with Inception Model on Food 101 dataset:

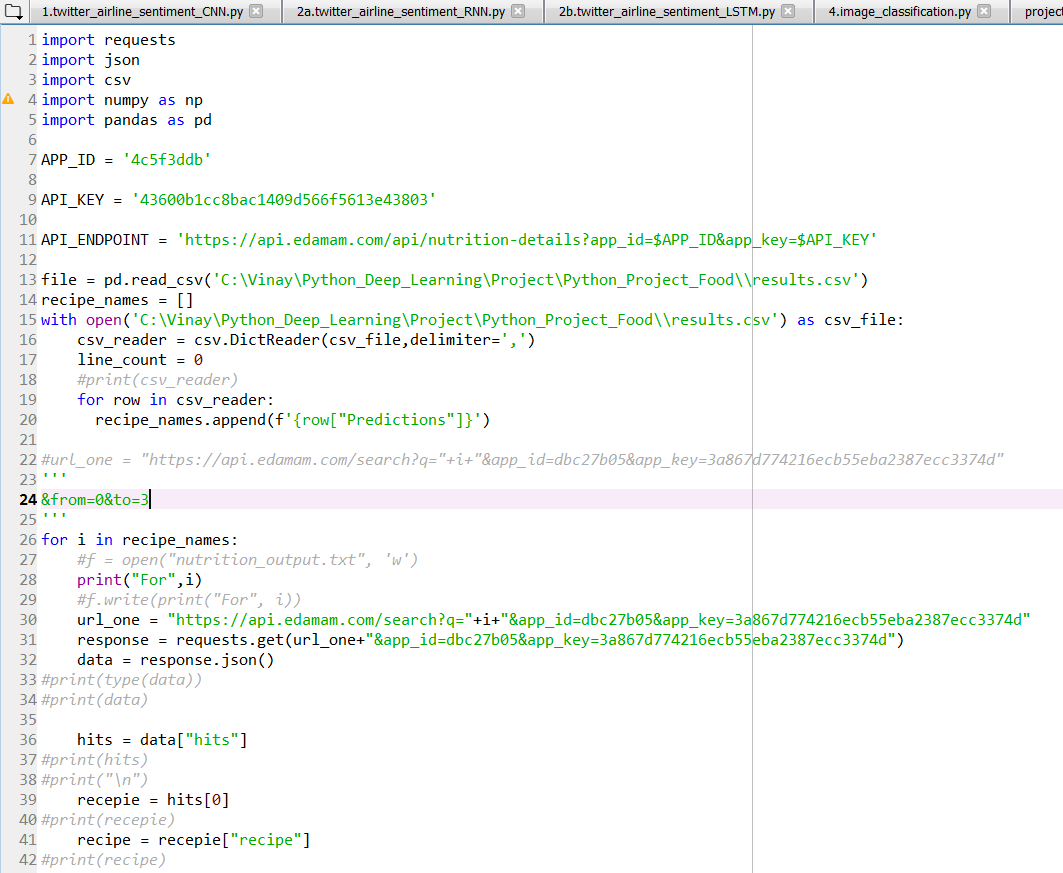


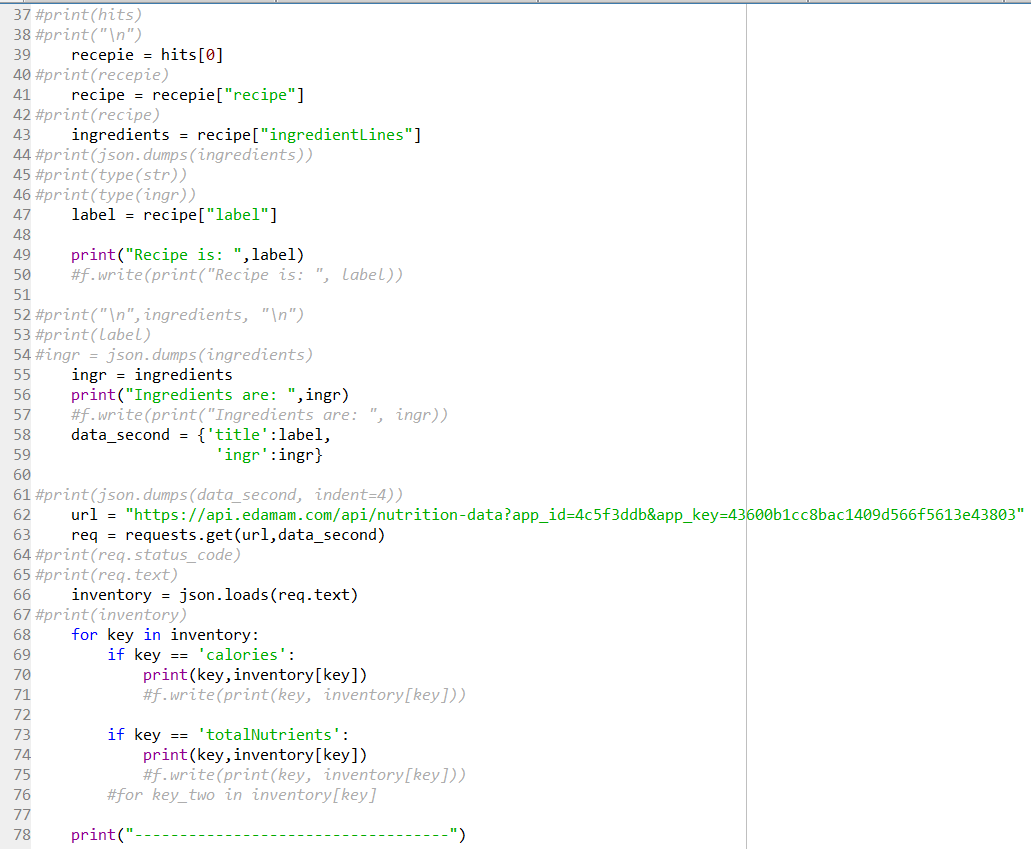


Code for loading the saved model and testing with test files:



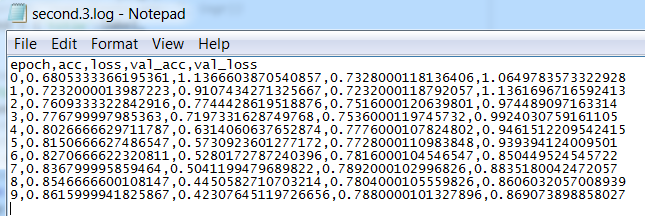
Code for Nutrition Api:



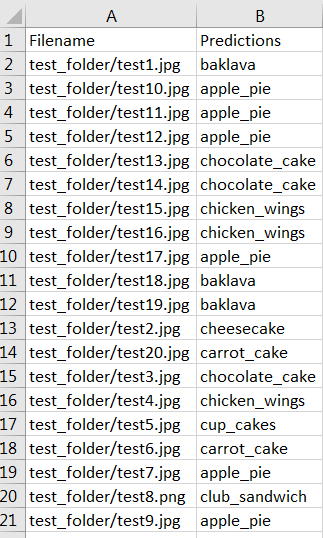


OUTPUT SNIPPET

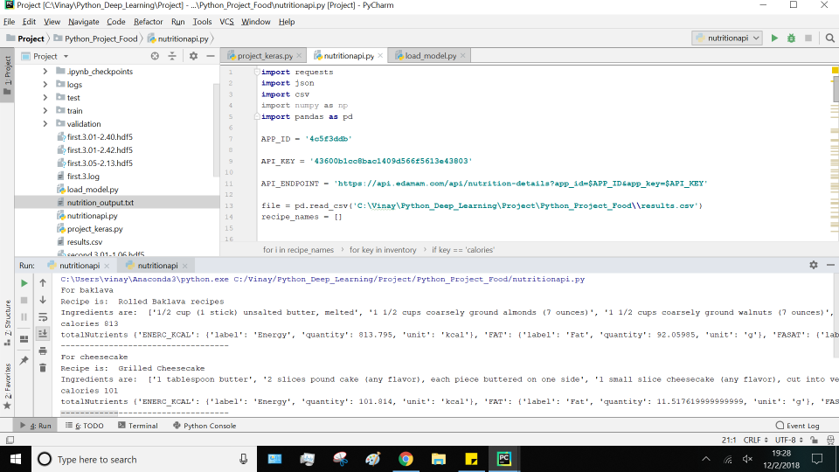
Log file for the second pass with 10 epochs:



Predicated result for the test files:



Output from the nutrition Api:



EVALUATION

* The accuracy for the code is quite good. With a total of 15 epochs, we reached a training accuracy of 86% and validation accuracy of 79%. With more time and more epochs, the accuracy can be increased more.
* We used two different kinds of optimizer which had a very positive result on our code.
* When implemented on 20 random test files, most of the predictions were good. Only 3 predictions were wrong.
* The nutrition of the predicted food was calculated accurately with the nutrition api.

CONCLUSION

During this project, we learned about the different kinds of models like AlexNet, ResNet, LeNet, VGG16, VGG19, inception etc. We also learned the used of different optimizers in same model. Due to time shortage, we only worked on 10 classes from food dataset. With proper time and appropriate resources, it will be possible to implement our project with all the classes in the food dataset.

LINKS

Github Link:

<https://github.com/fatemahasta/Python-DeepLearning-Project>

REFERENCES

1. <https://github.com/stratospark/food-101-keras>
2. <https://github.com/matterport/Mask_RCNN>
3. http://blog.stratospark.com/deep-learning-applied-food-classification-deep-learning-keras.html