BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

COURSE NO : EEE 212

NAME OF THE PROJECT : Food Image Recognition and Food Value Determination by Using CNN(Convolutional Neural Networks)

Submitted to :

Md. Mukhlasur Rahman Tanvir

Lecturer, Department of EEE

Shoilie Chakma

Lecturer, Department of EEE

Submitted By :

M Zafir Sadik Khan	ID : 1606135
Shihab Uddin Ahmed	ID : 1606136
Partho Ghosh	ID : 1606161
Sudipta Chandra Sarkar	ID : 1606162

LEVEL	: 2	TERM	: 1
DEPT.	: EEE	SECTION	: C1

Table of Contents

Topic Page No.

1. Objective:

Food safety and health is increasingly attracting attentions. An effective computer vision method to recognize the food category can efficiently help evaluate the food nutrition. It provides a simple means to estimate the dietary caloric intake and evaluate people's eating habits, by using cameras to keep track of their food consumption. We mainly focus on using the CNN (Convolutional Neural Networks) to identify an image of a specific food then print out the food value of that specific food, i.e. how much calories the food contain and how much protein, fat and other minerals the food possess. We define each food as a single class and there will be a picture containing the information of that food to the corresponding class. In this study, a small-scale dataset consisting of 10000 images of ten categories. We proposed a CNN-based food recognition method on the food recognition problem: the transfer learning and the fine-tuning on the whole architecture based on the VGG-19 model. Our algorithm is performed on the Food-101 dataset and obtained impressive recognition results. VGG-19 converges much faster and achieves accuracy of 81.6%. Our future work includes optimizing the network architecture and yielding a much higher leaning result. What's more, we will try to implement the recognition algorithm on the mobile devices and make it available in our practical daily lives.

2. Background:

Food is the cornerstone of people's life. Nowadays more and more people care about the dietary intake since unhealthy diet leads to numerous diseases, like obesity and diabetes. Accurately labelling food items is significantly essential to help us keep fit and live a healthy life. However, currently referring to nutrition experts or Amazon Mechanical Turk is the only way to recognize the food category.

To stay healthy we not only need all of the 5 nutrients (Protein, Carbohydrate, Fat, Vitamins and Minerals) in our diet but we also need them in the correct quantities - this is what we mean by a balanced diet. The consequences of not having a balanced diet are numerous: if you do not eat enough protein, you will not be able to grow properly; if you do not eat enough energy containing foods (e.g. carbohydrates and fat), you will feel very tired; and if you eat too much energy containing foods you will become overweight. Many people in the developed world eat too much of some types of food, for example a lot of saturated fats, and become overweight. Obesity is becoming a big problem in the developed world. One third of all Americans are obese. Being obese has serious health implications including increasing your chances of heart disease, diabetes, high blood pressure, having a stroke or getting a number of forms of cancer. In the developing world, on the other hand, many people suffer from: Hunger, or under-nutrition, whereby they do not have enough food or Malnutrition, which means 'badly nourished' and is as much about what you eat as how much. Malnutrition is characterized by inadequate intake of protein, energy and/or micronutrients and by frequent infection and disease. Food Aid is a great opportunity for all to learn more about these issues and to help the hungry and malnourished overseas.

Due to the widespread use of low-cost imaging devices like smartphone cameras, more and more applications are being developed in computer vision to facilitate automatic object recognition, among which food image recognition has recently gained much attention. Nowadays, people, especially diabetes patients, are increasingly cautious about their diet for improved health care. Food image recognition provides a simple means to estimate the dietary caloric intake and evaluate people's eating habits, by using cameras to keep track of their food consumption. In recent years, Convolutional neural networks (CNN) have enjoyed great popularity as a means for image classification. In implementing CNN for image classification, researchers have to collect such a large-scale dataset that contains more than one million images, for network training because of the need for learning a large number of parameters involved in the network, which, however, is not a trivial task. However, how to derive the food information (e.g., food type and portion size) from food image effectively and efficiently remains a challenging and open research problem. We propose a new Convolutional Neural Network (CNN)-based food image recognition algorithm to address this problem. We applied our proposed approach to real-world food image data sets (Food-101) and achieved impressive results. Our experiments have demonstrated that the proposed approach is a promising solution for addressing the food image recognition problem. Our future work includes further improving the performance of the algorithms and integrating our system into a real-world mobile and cloud computing-based system to enhance the accuracy of current measurements of dietary intake.

3. Methodology:

3.1. Datasets:

A dataset of food images is required to evaluate the performance of the different feature extraction and classification algorithms proposed. Currently, the largest dataset available is Food-101. It contains 101,000 images divided into 101 food categories.

Name	Year	Images	Classes	Type	Task	Annotation
Food-101	2014	101,000	101	101	Food Recognition	Public

But we have taken only 10 classes of food. Each class contains 1000 images of food. The classes are: Burger, Chicken Curry, Chicken wings, Sandwich, Donut, French fry, Omelet, Pizza, Samucha and roll.We have taken only 10 classes because of time limitation and limited CPU power. The dataset has been taken from the kaagle.com. Each class contains different types of images of same food. These images had a large variations in quality and size. Some of them have a neat, uniform background while some have cluttered background

Fig: Some images from Food-101 Dataset.

3.2. Train, Test & Split:

To Train our model, we split the data in each class. Then randomly divided into two parts for training and test with a 3:1 ratio, resulting in 750 training and 250 test images. Totally now we have 7500 test images and 2500 test images to train our model.

3.3. Image Processing:

We do not use any image preprocessing in our project. Prior to image analysis, all the images were down-sampled to a fixed resolution of 64×64. Because all the images in the dataset are not same in size. So we have to resize them.

3.4. Model:

Convolutional Neural Networks (CNN) are very similar to ordinary Neural Networks, they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function: from the raw image pixels on one end to class scores at the other. And they still have a loss function (e.g. SVM/Softmax) on the last (fully-connected) layer and all the tips/tricks we developed for learning regular Neural Networks still apply.

In this project, first we start with a basic two layer convolutional neural networks. It was not a powerful neural network. Then we move to LeNet model by which we achieve accuracy only 41.6%.

After that, we move to ImageNet architecture. We start with a VGG-16 model and fit the data. It has 5 block of convolutional and maxpooling layer. It has 3 fully connected layer also. As there is ten prediction in output layer, so we add 10 neurons in output layer. But ImageNet model architecture has 1000 neurons in output layer. So we have to change it. We only train our last layer and freeze the weights of other layer. By this way, we train our model and we achieved accuracy of 80.4%. Then we also look for other types of model but no model can increase the accuracy. Then we finally moved to the VGG-19 model. We run 100 epoch and achieved the accuracy of 81.6%.

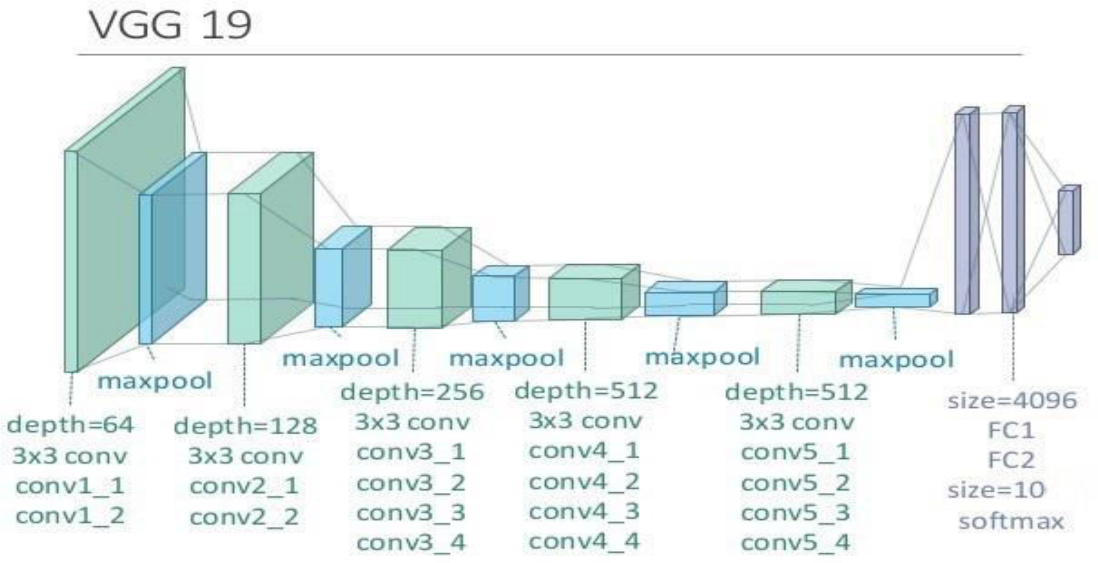
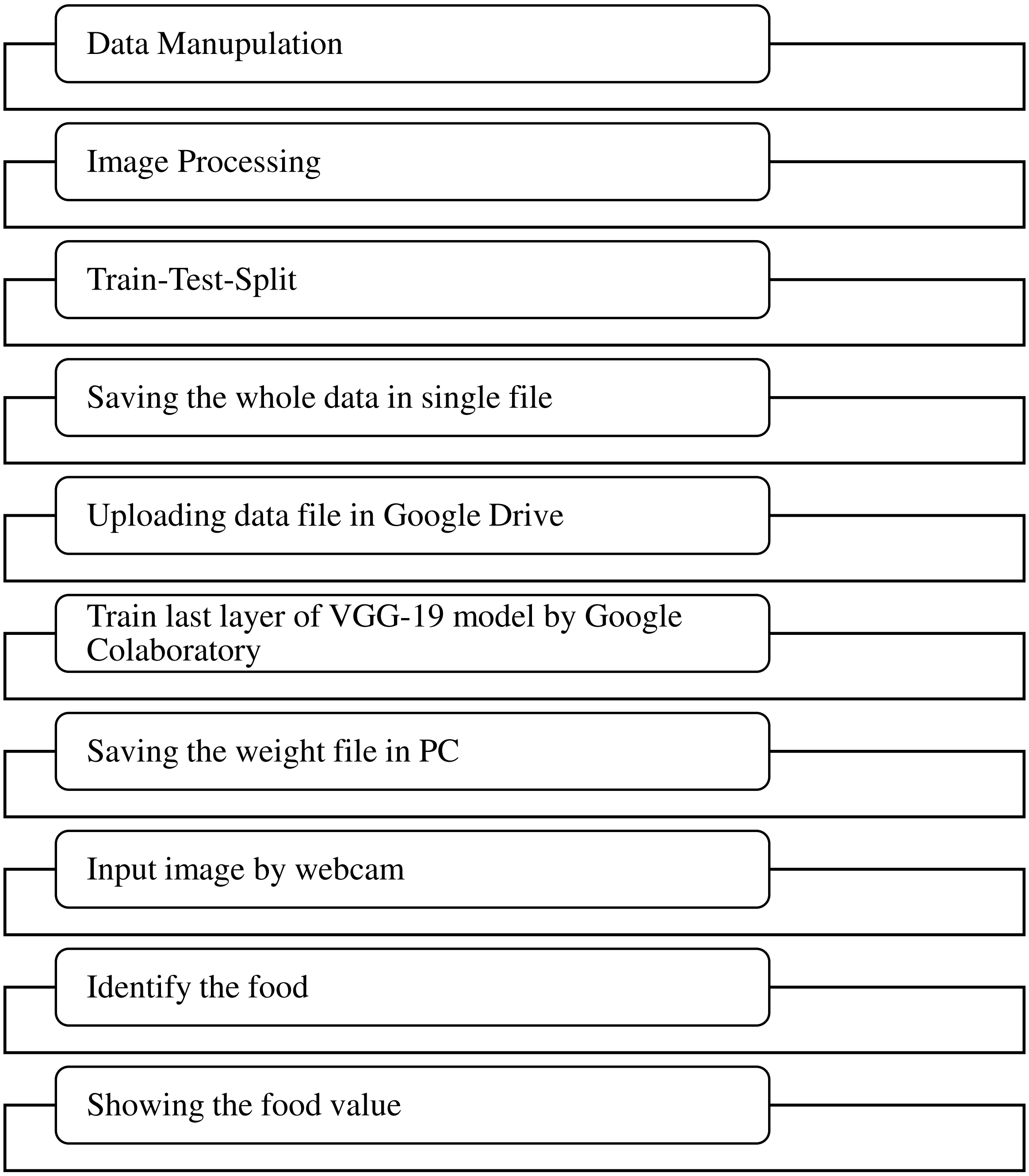


Fig: VGG-19 Model.

3.5. Google Colaboratory:

Colaboratory is a research tool for machine learning education and research. It's a Jupyter notebook environment that requires no setup to use. Firstly we use our CPU power which was very slow. For each epoch, about one hour needed. Then we turn into GPU power which was very time consuming and help us to run more epoch fastly. So we can test more model in our dataset and check the accuracy.

3.6. Flow-chart on Overall Procedure:


Data Manupulation
Image Processing
Train-Test-Split
Saving the whole data in single file
Uploading data file in Google Drive
Train last layer of VGG-19 model by Google Colaboratory
Saving the weight file in PC
Input image by webcam
Identify the food
Showing the food value

4. Result:

Model Accuracy:

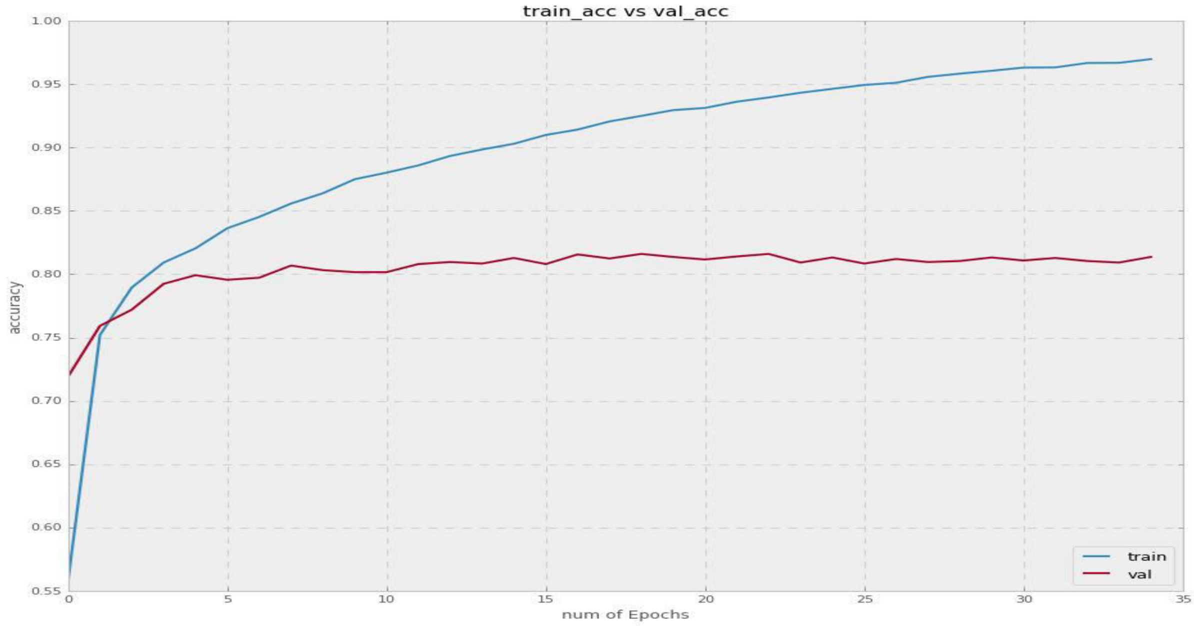


Fig: Accuracy vs No. of Epoch.

Model Loss:

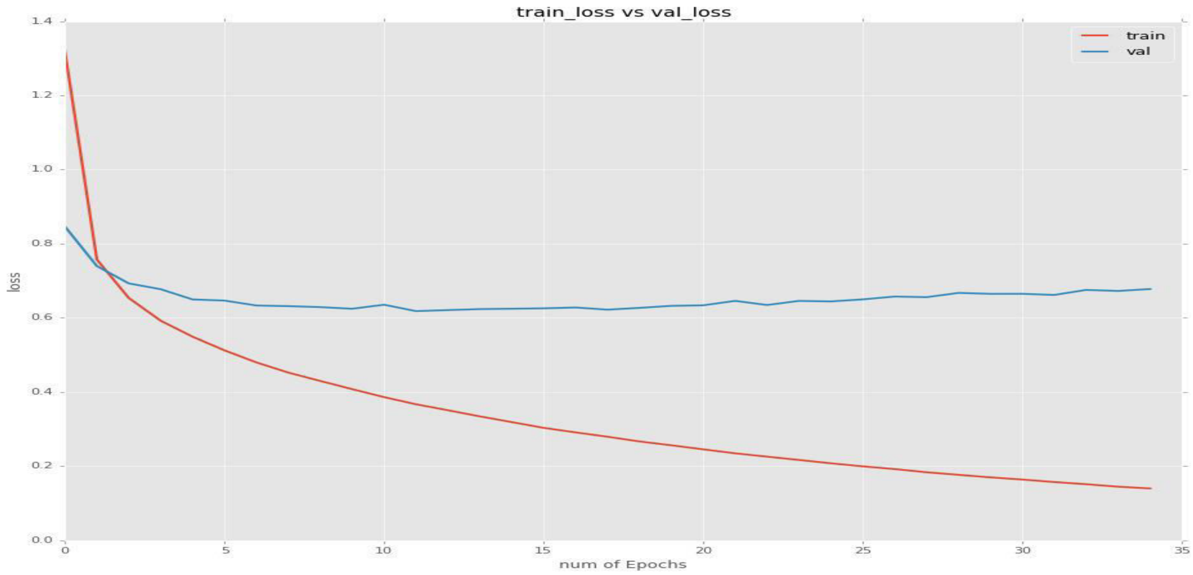
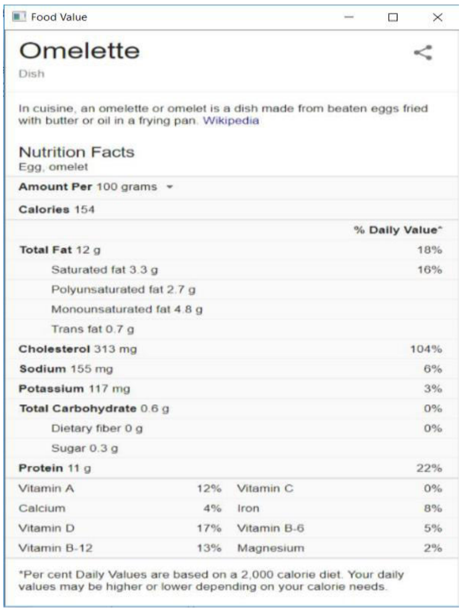


Fig: Loss vs No. of Epoch.

Input & Output:

Fig : Input an image (Omelette) captured by webcam



Omelette

Dish

In cuisine, an omelette or omelet is a dish made from beaten eggs fried with butter or oil in a frying pan. Wikipedia

Nutrition Facts

Egg, omelet

Amount Per 100 grams	
Calories 154	
	% Daily Value*
Total Fat 12 g	18%
Saturated fat 3.3 g	16%
Polyunsaturated fat 2.7 g	
Monounsaturated fat 4.8 g	
Trans fat 0.7 g	
Cholesterol 313 mg	104%
Sodium 155 mg	6%
Potassium 117 mg	3%
Total Carbohydrate 0.6 g	0%
Dietary fiber 0 g	0%
Sugar 0.3 g	
Protein 11 g	22%

Vitamin A	12%	Vitamin C	0%
Calcium	4%	Iron	8%
Vitamin D	17%	Vitamin B-6	5%
Vitamin B-12	13%	Magnesium	2%

*Per cent Daily Values are based on a 2,000 calorie diet. Your daily values may be higher or lower depending on your calorie needs.

Fig : Showing the food value of Omelette as output

5. Discussion:

We divide our project in 2 parts: i. image recognition & ii. Food value determination. At first we have to make a dataset or find a dataset. For the chosen of dataset, it was kept in mind that we needed various types of data. We needed the data of the various classes of food in various angles, the data of ripe fruits, unripe fruits, detach part of fruits and many others. As we needed various types of data for every classes it was not possible for us to make dataset on this. So we started finding dataset. It was very difficult for finding out data with enough information and also we had to make sure that resolution was better. Firstly we found out dataset with information of 30-40 classes each of having 100-200 pictures. Finally we found out in Kagle.com a dataset named Food 101 where 101 classes of food information was given with 1000 pictures for each class. But there were many classes which not found in our country and so we started the work with 10 classes which is familiar to us. These are Burger, Chicken Curry, Chicken wings , Sandwich , Donut , French fry , Omelet , Pizza , Samucha and roll.

After the data collection we started the work with a basic neural network named ‘2 Layers Neural Network’ which was customized by us and as the data was very noisy we got dissatisfied results. We firstly got only 20-25% accuracy.

Next we moved to Li-Net. Li-net is a first generation convolution neural network and by working with this architecture we got the highest accuracy about 41%. Here we did not do any kind of image pre- processing. Instead we just invert the pictures in 64 * 64. As the images were not in the same size we made all in 64*64 and just fitted the images with our system. And by our system we got the highest accuracy with 41%.

As 41% accuracy was not good enough so we tried further to improve the system. We tried by changing learning rate but we did not get better result. At first we tried with train data 750 & test data 250. Neither increasing nor decreasing the data was able to change the result. As the processing was getting in CPU we were not able to test out different models. For solving the problem we used GPU in Google Collaborate where they provide free GPU. But in Google Collaborate there is a problem in data upload. Normally we upload data from our disk, read data from here and do further processing. But for the Google Collaborate directly we can not access our disk. For this reason we can not upload data, read data and process it. So we would not be able to fit any kind of model, give input and also would not find any output. So the system would not work.

So we had to find an alternative way. At first the data was pre-possessed the data. At the same way in imagenet in keras dataset are pre-processed and if we provide image size there the image would be pre-processed. So we did the same and by this way our data was pre-processed. Then the data was conceded next and saved as one file. Although there were only 10 thousand pre-processed images, the size of the file was about 5.6GB and uploaded that file in Google drive. Then we had to merge Google drive and collaborate with specific code. And by this way the file was ready and we took the file in main function.

After that we applied transfer learning method. At first we worked with VGG-16. Here train data was 750 & test data was 250. There were 5 blocks of convolution line layers and 3 connected layers. Last layer was for 1000 prediction but as we worked with 10 classes so we changed the layer for 10 predictions. One thing to mention that we just worked with last layer and the other layers were kept frizzed. By this procedure we got huge success. We got about 80.4 % accuracy. Although before this our best accuracy was about 40-41% but this procedure gave us double success. Then we tried further to improve this process further more. We tried with Inception, Exception and Resnet-50 model. But these models did not give us as good result as VGG-16 model gave us. In these models we found about 70-75% accuracy. At last we tried with another model named VGG-19 and got a satisfactory result of 81.6% accuracy.

We are satisfied with the result. But if we got fewer times more, we could do rather effect in this field. We could do image augmentation like horizontal flip, zoom in- zoom out, Z-C, Y-T augmentation for dark and bright images. Then we expect that we could find much better result like 85-90% success.

6. Appendix:

1. Folder (Overall Matlab Project):

https://drive.google.com/open?id=1ynzHafg69L3ksOqkIsIpIJzeI8oJ3k_8

2. Dataset link:

https://www.kaggle.com/dansbecker/food-101

3. Code link:

i. Data extraction and data manipulation

https://drive.google.com/open?id=1bp15cNSxYYJqBRlfSVw4PAZf7QbH_dmp

ii. Weight File Producing by Jupyter Notebook (Google Colaboratory file)

https://drive.google.com/open?id=1C9LdSf4QikZmlggFm-t31n_uMClyo5Sr

iii. Main classifier

https://drive.google.com/open?id=1aJwFaAxuNCDhEncU2E3P8PKZ8SCWtS_0