Algorithm for food item recognition using VGG-16 and InceptionV3 CNN

Pulkit Jain* and Paras Chawla

Department of Electronics and Communication Engineering, Chandigarh University Mohali, India
E-Mail: pulkit.mech@cumail.in; drparaschawla.ece@cumail.in
** Corresponding Author*

Abstract.

Computer Vision is impacting the retail sector by bringing revolutionary changes in shopping experience for customers. It has not only raised the competition among several retailers but after its combination with Artificial Intelligence (AI) it has automated the manual processes thereby saving costs and time. Deep Learning is also one area along with computer vision which has done wonders in the area of many open research problems like food item recognition using Machine Learning algorithms. Convolution Neural Network (CNN) is of the approaches which can fulfil the pre-requisites for solving image recognition research problems. The objective of this paper is to utilise the proposed algorithm for automated identification of food items. The algorithm has been tested successfully using VGG-16 and InceptionV3 CNN models. The results of food item recognition show nearly 96.37% in VGG-16 and 98.4 % accuracy in InceptionV3 respectively. At the end real time testing results on food items and barcode items are also presented.

Keywords. Convolution Neural Network, VGG-16, InceptionV3, Machine Learning.

1. INTRODUCTION

An increase in number of health related problems, chronic diseases like obesity and cancer have raised alarms and created an urgent need for keeping a check on diet intake. A very common and an open research issue in the area of health and nutrition are to design appropriate mechanism for measuring accurate diet intake. To have a check on the food consumed per day, individuals generally keep a record of meals taken every day. But such an exercise is carried out manually utilising text based description but it is a tedious and dreary task. To beat this issue, there have been endeavours like image recognition using camera scanning. But due to wide variety of classes of food items available specially fruits and vegetables, the recognition accuracy becomes a major challenge. In this regard, many researchers have utilised image based recognition techniques. A smart phone based system was designed for food item recognition and logging using a small dataset which could

replace traditional methods of recording food items [1]. Similar work was carried out which achieved an accuracy of nearly 62.5 % for recognition of Japanese using machine learning approach [2]. Other related works talked about the usage of image retrieval, and related image processing techniques for not only food recording but also recording the nutritional content of food items [3]. Deep learning is one of the latest area which is also being used to address this open research problem [4]. It basically refers to collection of certain algorithms to solve such complex issues. The most particular trademark is that the distinguishing and important features of images are extricated automatically during the training of the models. CNN [5] is one of the approaches which can fulfil the pre-requisites of deep-learning methodology for solving image recognition problems. CNN is currently a best in class procedure for image recognition tasks especially after validation of its excellent performance in Visual Recognition Challenge. The entire paper is organized as below: Section II talks about the pre-processing steps involved in the algorithm used for food item recognition. Section III describes the main steps of algorithm used followed by description of hardware setup designed. The architecture of the two CNN models used is discussed in Section IV. The results obtained after testing are discussed in Section V.

2. ALGORITHM-PRE PROCESSING STEPS

Simple distortions like crop, flip and scale as pre-processing operations on images during model training can play a vital role in improving the results of the algorithm. These operations are simply analogous to natural variations and background noises present in the real world scenario. Hence, these make the models obtained after training very efficient. The first step starts with conversion of image data type to float32 type. This conversion followed by the cropping and scaling operation is depicted in Figure2.1.

```
decoded_image_as_float = tf.image.convert_image_dtype(decoded_image,
                                                      tf.float32)
decoded_image_4d = tf.expand_dims(decoded_image_as_float, 0)
margin_scale = 1.0 + (random_crop / 100.0)
resize_scale = 1.0 + (random_scale / 100.0)
margin_scale_value = tf.constant(margin_scale)
resize_scale_value = tf.random_uniform(shape=[],
                                       minval=1.0,
                                       maxval=resize_scale)
scale_value = tf.multiply(margin_scale_value, resize_scale_value)
precrop_width = tf.multiply(scale_value, input_width)
precrop_height = tf.multiply(scale_value, input_height)
precrop_shape = tf.stack([precrop_height, precrop_width])
precrop_shape_as_int = tf.cast(precrop_shape, dtype=tf.int32)
precropped_image = tf.image.resize_bilinear(decoded_image_4d,
                                            precrop_shape_as_int)
precropped_image_3d = tf.squeeze(precropped_image, axis=[0])
cropped_image = tf.random_crop(precropped_image_3d,
                               [input_height, input_width, input_depth])
```

Figure 2.1. Image Conversion, Cropping and Scale Operation

The crop operation starts with random placing of a bounding box in full sized image. Then crop parameter value decides the size of the box with respect to given input image. When zero, no crop i.e. the output and input image size are identical. When 0.5 then box size is half the dimensions (height x width) of input image. Next operation almost similar to crop is ‘scale’ with one difference that the bounding box is oriented at the centre initially. When scale % is zero bounding box and input image are of same size. When 50 % then box will be in any random range in between half and full dimensions of the image. For flip operation, ‘random_flip_left_right’ function available in tensor flow library is used as depicted in Figure2.2(a) It flips the image generally along second dimension i.e. width or will pass the image with same dimensions as input. In case of batch images, each image undergoes flip operation randomly independently of other images.

```
if flip_left_right:
  flipped_image = tf.image.random_flip_left_right(cropped_image)
else:
  flipped_image = cropped_image
```

(a)

```
brightness_min = 1.0 - (random_brightness / 100.0)
brightness_max = 1.0 + (random_brightness / 100.0)
brightness_value = tf.random_uniform(shape=[],
                                     minval=brightness_min,
                                     maxval=brightness_max)
brightened_image = tf.multiply(flipped_image, brightness_value)
distort_result = tf.expand_dims(brightened_image, 0, name='DistortResult')
```

(b)

Figure 2.2. Code snippet depicting a) flip left right b) brightness adjustment operation

The brightness can also be adjusted by varying brightness variable as depicted in Figure2.2(b) The steps involved in the algorithm for the same are: a) User inputs % value in brightness variable b) Calculation of minimum and maximum value c) Automatic selection of a random value between minima and maxima d) Multiplication of this value with input image resulting in output image with variable brightness.

3. ALGORITHM USED AND HARDWARE SETUP

The entire algorithm for training of CNN models comprises of seven main steps as shown in Figure3.1(a). The first phase involves calculation of bottleneck values after complete analysis of all input images. 'Bottleneck' is the name given to the layer just before the final output layer which mainly carries out the classification task. It outputs a set of values for the classifier to properly distinguish between the various classes of fruits and vegetables.

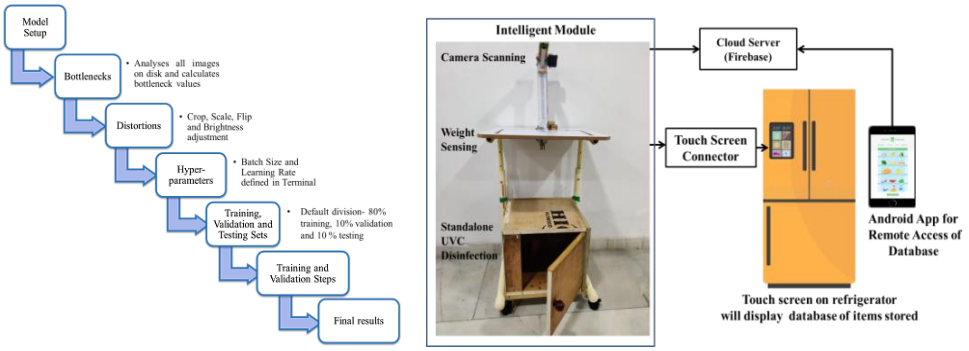

Figure 3.1 (a) Steps of Algorithm (b) Hardware Setup for food item identification

Next step is adding the distortions to improve the quality of results and getting the model trained for all worst case scenarios. Learning rate and batch size are certain hyper-parameters which play an important role to improve overall precision of model. Next step is division of the dataset into training (80%) and testing sets (20%), to avoid the problem of overfitting. The hardware setup [6] developed for applying the algorithm for food item identification comprises of intelligent module for camera scanning and weight sensing as depicted in Figure3.1(b).

4. VGG-16 AND INCEPTIONV3 CNN

VGG-16 [7] is one of the most preferred CNN architectures in the recent past. It has 16 convolution layers and has much more complex architecture than initial versions like LeNET. Only limitation is large number of parameters i.e. nearly 138 million which at times becomes difficult to handle. The flowchart depicted in the Figure4.1(a) shows

uniform structure of VGG-16. VGG-16 CNN model has been trained on standard dataset i.e. FRUITS 360 [8]. The programming language and platform used for training is Python and Google Colab.

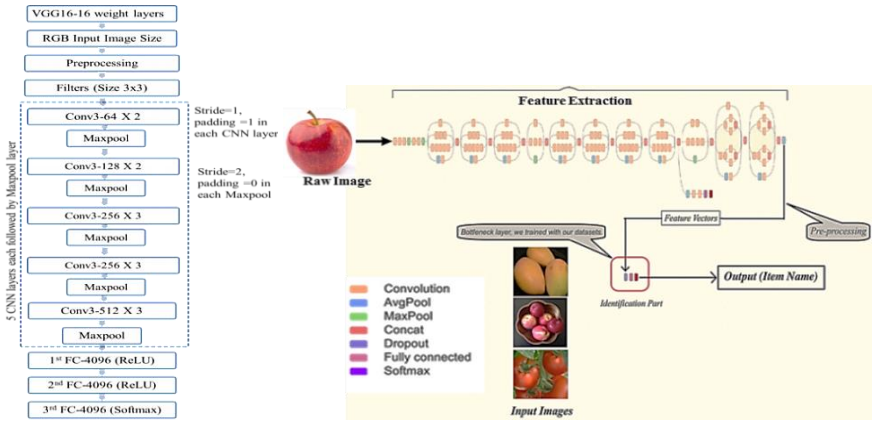

Figure 4.1 Flowchart depicting layers in (a) VGG-16 and (b) Inception-V3 CNN

InceptionV3 [9] is a 48 layer-network, which is pre-trained on ImageNet database for over 1000 classes as shown in Figure4.1(b). This model provides reduction in computational cost by 28% using factorization into smaller convolutions i.e. 5x5 convolution into two 3x3 CNN blocks. Moreover, the factorization into asymmetric convolutions i.e. decomposing 3x3 into 1x3 and 3x1 CNN blocks help in reduction of parameters by 33%.

5. TESTING AND RESULTS

The accuracy graph (orange line-validation, blue-line-training) and cross entropy loss graphs obtained after training of VGG-16 and Inception-3 CNN model is shown in Figure 5.1 and Figure5.2. The numbers of epochs are 10 and 4000 in both models respectively.

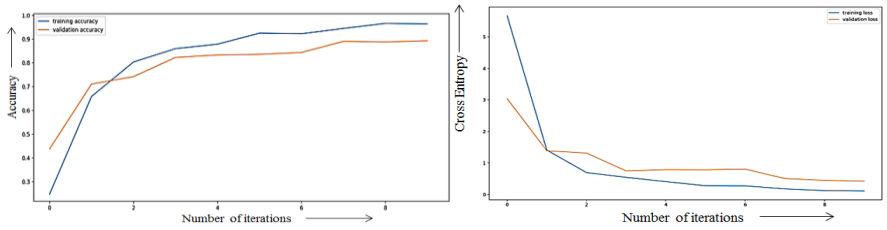

Figure 5.1 (a) Accuracy and (b) Cross Entropy of VGG-16 model v/s epochs

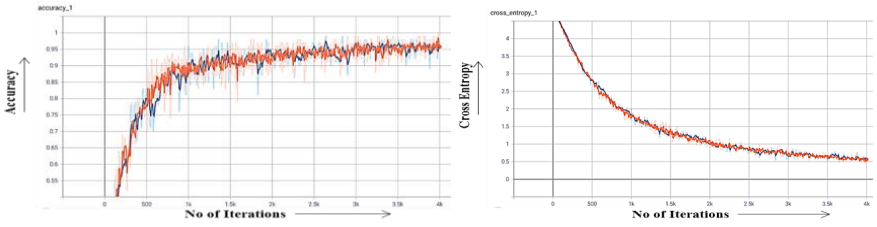

Figure 5.2 (a) Accuracy (b) Cross Entropy of InceptionV3 model v/s epochs

The model training results on test images and real time testing results are depicted in Figure5.3. The top five results are also depicted alongside which shows correct identification of different food items with high accuracy values. Table I depicts the comparison between training accuracy results of previous and the present work.

Figure 5.3 Model testing results on test and real-time images

TABLE 1 TRAINING ACCURACY RESULTS COMPARISON WITH PREVIOUS WORKS

CNN Model	Dataset	Training Accuracy (in %)
VGG-16-Previous Work [7]	FOOD-101	94.02%
InceptionV3-Previous Work [10]	FRUITS360	96.5%
VGG-16	FRUITS360	96.37%
InceptionV3	FRUITS360	98.4%

The testing with Quick Response (QR) and Barcode items are also depicted in Figure5.4.

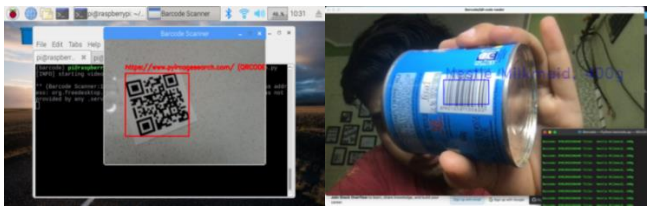

Figure 5.4 QR code and Bar code successful scan results

6. CONCLUSION

The testing of designed algorithm resulted in an excellent accuracy of 96.3% using VGG-16 and 98.4% using InceptionV3 CNN. The successful real time testing results carried using the designed hardware setup justified the performance of the algorithm proposed.

7. REFERENCES

[1] F. Zhu, et. al., 'The use of mobile devices in aiding dietary assessment and evaluation', IEEE J. Sel. Top. Signal Process., vol. 4, no. 4, pp. 756–766, 2010.

[2] H. Hoashi, T. Joutou, K. Yanai, 'Image recognition of 85 food categories by feature fusion', IEEE Int. Symp. Multimedia, pp. 296–301, 2009.

[3] K. Aizawa, Y. Maruyama, H. Li, C. Morikawa, G. C. De Silva, ' Food balance estimation by using personal dietary tendencies in a multimedia food log,' IEEE Trans. Multimed., vol. 15, no. 8, pp. 2176–2185, 2013.

[4] Q. V. Le, 'Building high-level features using large scale unsupervised learning', ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., pp. 8595–8598, 2013.

[5] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, 'Gradient-based learning applied to document recognition,' Proc. IEEE, vol. 86, no. 11, 1998.

[6] P. Jain, P. Chawla, 'Smart Module Design for Refrigerators based on Inception-V3 CNN Architecture,' Second Int. Conf. on Electon. and Sustainable Comm. Syst., pp. 1852-1859, 2021.

[7] S. Yadav, Alpana, S. Chand, 'Automated Food image Classification using Deep Learning approach,' Int. Conf. Adv. Comput. Commun. Syst., pp. 542–545, 2021.

[8] H. B. Unal, E. Vural, B. K. Savas, Y. Becerikli, 'Fruit Recognition and Classification with Deep Learning Support on Embedded System,' Innov. Intell. Syst. Appl. Conf. ASYU 2020.

[9] C. Szegedy, et al., 'Going deeper with convolutions,' IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, vol. 07-12-June-2015.

[10] Z. Huang, Y. Cao, T. Wang, 'Transfer learning with efficient convolutional neural networks for fruit recognition,' Proc. IEEE 3rd Inf. Technol. Networking, Electron. Autom. Control Conf., pp. 358–362, 2019.

Biographies

Pulkit Jain is currently a research scholar in ECE Department, Chandigarh University. His research areas include image processing, computer vision, and embedded systems.

Paras Chawla is currently working as an Associate Dean Academic Affairs and Professor, ECE Chandigarh University. He received the "Coventor Scholarship award" from MANCEF, New Mexico-USA. He has published more than 70 papers in various reputed National and International Journals/Conferences.