

# Food Recognition and Calorie Estimation using Convolutional Neural Networks

Rajni Jindal<sup>1</sup>, Sachin Singla<sup>2</sup>, Priyesh Agarwal<sup>3</sup> and Rishabh Singh<sup>4</sup>

<sup>1-4</sup>Department of Computer Science & Engineering, Delhi Technological University, New Delhi, India  
rajnijindal@dce.ac.in, sachindtu289@gmail.com, priyeshagarwal468@gmail.com, rishabhsingh66@gmail.com

**Abstract**—We propose a Convolutional Neural Network (CNN) based food classification and detection model which uses TensorFlow for calorie estimation, in an image of food taken. The calorie count is calculated on the basis of the calorie value for each identified category of fruit or vegetable.

**Index Terms**— Deep Learning, Food Recognition, Food Detection, Neural Network, CNN, TensorFlow, Keras, Calorie Estimation.

## I. INTRODUCTION

With the urbanization and busy lifestyle, obesity rates are increasing at an alarming rate. People need to control their calorie intake to maintain a fit and healthy body. Monitoring calorie intake is a very complex process since it requires huge volumes of data and calculations. So, it would be convenient to have a system that could automatically identify the calorie count using the images of the food item. In this paper, an efficient method to classify food objects and identify the calorie count is proposed, which is based on object recognition and deep learning neural networks.

### A. Motivation

Food is crucial for the proper functioning of the human body. In recent times, the number of individuals taking the utmost care of their dietary intake has increased manifold because of the consequences faced on taking inadequate nutrition such as obesity [1]. A diet scheme must consider the complete count of calories to be consumed for retaining a robust life.

However, in the majority of situations, individuals face complications in identifying the calorie count of the food item they take, because they do not have proper nutritional information about the food item, and also because of the complex calculations involved in manually measuring the calorie count. This process takes a lot of time, and because of their busy lifestyle, it becomes really difficult for individuals to carry out this herculean task. Therefore, it is practical to have a system that can automatically measure the number of calories present in a food item. Such a system can help people maintain proper health, and prevent various diseases later.

### B. Problem Description

In the last few years, advancements in deep learning and convolutional networks have led to improvements in the accuracy of object classification and detection based models. Incorporating this advanced technology to improve the accuracy of food detection and classification is remarkably crucial for helping the population lead a

robust life. In this paper, we propose a food recognition and calorie estimation algorithm which is based on deep learning neural networks and improves the accuracy of the assessment.

### *C. Background*

Prior work in the field of food image classification was primarily focused on the simple algorithms of machine learning such as Support Vector Machines (SVM) [2] and Random Forest.

These algorithms mostly use the comparative or spatial relations of various attributes or features. However, they require greater computational resources, if performed on a large scale. Our convolutional neural network-based model was able to provide higher accuracy than traditional methods, which at its peak was 97.5%. We also record the changes in efficiency and accuracy on modifying the hyperparameters, such as the count of neurons in every convolutional layer or the learning rate. This recorded data can help us determine the performance of the model and the ideal values of associated parameters that give the best accuracy.

In this paper, we demonstrate the way in which the model acts on changing the associated features, and characteristics. The resulting model is dependent on TensorFlow's Object Detection API, which can be useful in detecting multiple fruit items in a single image, and then the final calorie count is found by using the number of calories of each detected fruit or vegetable in the image, by performing some mathematical calculations.

### *D. Algorithm*

In this paper, we have shown different algorithms for classifying objects like Support Vector Machines and Random Forest, along with deep learning neural networks such as Convolutional Neural Networks (CNN) [3], which is implemented on TensorFlow API [4] and Keras framework. We have classified 131 classes of fruits and vegetables. For that, we incorporated LabelImage's graphical user interface for manually binding input boxes on the images that were collected from google, and then employed the COCO model's [5] weights to train the weights of neural network using the approach of transfer learning on different categories of food classes. Finally, the layers of the neural net are modified using the object detection API of TensorFlow [4] to improve the model for our own dataset training. The evaluation of our trained model is carried out with the help of TensorBoard. Finally, the detection is used in measuring the count of Food objects for the purpose of calorie estimation in the respective image.

## II. RELATED WORK

There is an increasing interest in object recognition and classification based algorithms.

Researchers have already begun to employ AI based algorithms, deep learning [6] and computer vision which solve the task of estimating the calorie intake of food using its image. The addressed problem can be solved using different algorithms such as YOLO, CNN, and SSD [7]. However, the obtained accuracy varies for each algorithm. There can also be different techniques for volume estimation. Although these models have satisfactory outcomes, the majority of algorithms are either too slow or have less accuracy. CNN based food classification is relatively faster, and also has very high accuracy of upto 97%.

The primary motive of this research was to use a novel method for addressing the calorie measurement problem, and to achieve better results, with greater accuracy. We provide a model that makes use of deep learning neural networks as a means of making the food detection model highly accurate. It has been shown by experimental results that on making use of various methods for segmentation like color, texture, the accuracy rate is greatly improved for images containing parts of the food item.

## III. PROPOSED WORK

For obtaining a good classifier, it becomes indispensable to have a reliable dataset containing exceptional quality as well as quantity.

### *A. Data Exploration*

In this paper, we have used the latest version of the Fruits-360 [8] dataset which consists of numerous images of fruits and vegetables. The latest update of the dataset which is version 5, contains 90,483 images of fruits and vegetables which belong to 131 categories, and is continuously updated with new images.

Noisy background, and inadequate number of images are the major issues with the prior datasets. On using such datasets for training, the classifier may be unable to classify the correct food item with change in background.

Each image of the Fruit-360 [8] dataset consists of an individual fruit or a single vegetable. For calorie

estimation, data related to the nutritional information and per unit calorie for different food items is required. For obtaining this information, we have used the data from the Food and Drug Administration (FDA). FDA is an organization, which plays a notable role in assisting public health by stimulating development and innovation in the field of medicine. FDA has its headquarters in Maryland, and its 223 offices are located throughout 50 states of USA.

### B. Data Description

The latest update of the Fruits-360 dataset [8] contains 90,483 images of fruits and vegetables which belong to 131 categories. Out of these 90483 images, the total number of training set images are 67692, and the number of testing set images are 22688. The remaining 103 images are the ones which contain more than one fruit or vegetable.

This Fruit-360 dataset is available on both Kaggle[8] and GitHub[9]. For obtaining the images from different angles, a motor was used to rotate the fruits and vegetables, and the images were captured.

Fig. 1 denotes the proportion of the training and testing set images in the dataset.

### C. Training of Classifier

The CNN model is built using the following steps:

1. Sequence-create a linear layer stack: The constructor is called by passing the list of layers by the sequential model. After that the tuple-based input shape is passed to the first layer. The input form contains useful batch size dimensions that specifies a fixed batch size.
2. Convolution: In the convolution layer, all neurons perform convolution operations on the input. The resultant power of the layer is determined by the size of the filter where convolutional neurons are the most important parameter. The size of the file determines the output power.
3. Grouping: The grouping layer used is the largest grouping used to reduce the space size. This shrinks the count of parameters, thereby reducing the length of calculation.
4. Fully connected: After receiving the input of the previous neuron, the last layer is fully connected, also known as the Dense layer.

Fig. 2 shows the representation of the CNN model used for training. Fig. 3 denotes the steps the image undergoes when it is passed through the CNN model.

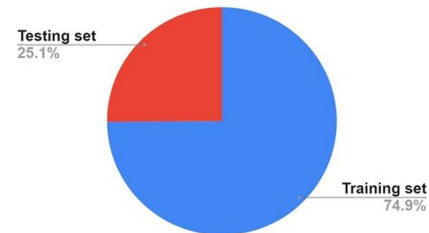


Figure 1. Training set vs. Testing set

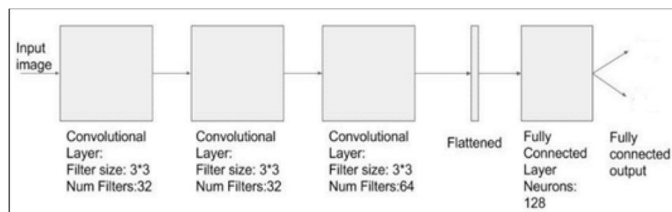


Figure 2. Representation of CNN

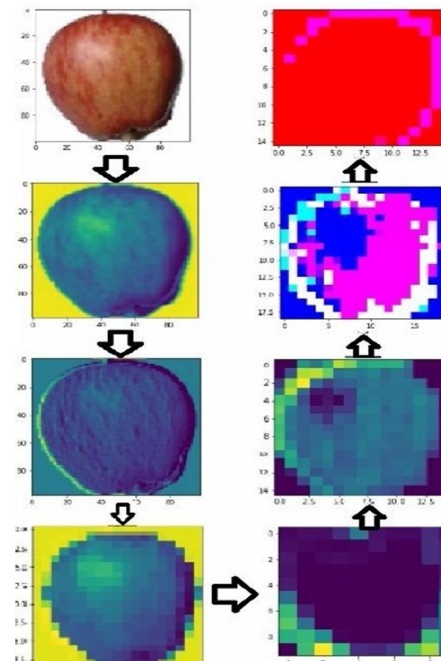


Figure 3. Steps of image when passed through CNN

We tried with different configurations of the models to obtain the best possible CNN layer configuration. We observed that the best results were obtained for the 8 Convolutional Neural network with 6 Conv2D layers and 2 Dense layers. The accuracies obtained for the various configurations are depicted in Table I.

TABLE I. ACCURACIES FOR DIFFERENT CONFIGURATIONS OF THE CNN MODEL

| S. No. | CNN layer configuration |      | Accuracy on the training set | Accuracy on the test set | S. No.        | CNN layer configuration |               | Accuracy on the training set | Accuracy on the test set |    |
|--------|-------------------------|------|------------------------------|--------------------------|---------------|-------------------------|---------------|------------------------------|--------------------------|----|
| 1      | Convolutional           | 16   | 0.9636                       | 0.9347                   | 6             | Convolutional           | 16            | 0.9734                       | 0.9423                   |    |
|        | Convolutional           | 32   |                              |                          |               | Convolutional           | 32            |                              |                          |    |
|        | Dense layer             | 512  |                              |                          |               | Convolutional           | 16            |                              |                          |    |
|        | Dense layer             | 60   |                              |                          |               | Convolutional           | 32            |                              |                          |    |
| 2      | Convolutional           | 32   | 0.9733                       | 0.9466                   |               | Convolutional           | 64            |                              |                          |    |
|        | Convolutional           | 32   |                              |                          |               | Convolutional           | 128           |                              |                          |    |
|        | Dense layer             | 512  |                              |                          |               | Dense layer             | 1024          |                              |                          |    |
|        | Dense layer             | 60   |                              |                          |               | Dense layer             | 60            |                              |                          |    |
| 3      | Convolutional           | 32   | 0.9705                       | 0.9628                   | 7             | Convolutional           | 32            | 1                            | 1                        |    |
|        | Convolutional           | 32   |                              |                          |               | Convolutional           | 32            |                              |                          |    |
|        | Convolutional           | 64   |                              |                          |               | Convolutional           | 64            |                              |                          |    |
|        | Convolutional           | 64   |                              |                          |               | Convolutional           | 64            |                              |                          |    |
|        | Dense layer             | 512  |                              |                          |               | Convolutional           | 64            |                              |                          |    |
|        | Dense layer             | 60   |                              |                          |               | Dense layer             | 512           |                              |                          |    |
| 4      | Convolutional           | 32   | 0.9504                       | 0.9447                   |               | Dense layer             | 2             |                              |                          |    |
|        | Convolutional           | 32   |                              |                          |               | 8                       | Convolutional |                              |                          | 32 |
|        | Convolutional           | 64   |                              |                          | Convolutional |                         | 32            |                              |                          |    |
|        | Convolutional           | 64   |                              |                          | Convolutional |                         | 64            |                              |                          |    |
|        | Convolutional           | 64   |                              |                          | Convolutional |                         | 64            |                              |                          |    |
|        | Convolutional           | 64   |                              |                          | Convolutional |                         | 64            |                              |                          |    |
|        | Convolutional           | 64   |                              |                          | Convolutional |                         | 64            |                              |                          |    |
|        | Convolutional           | 64   |                              |                          | Dense layer   |                         | 512           |                              |                          |    |
|        | Dense layer             | 512  |                              |                          | Dense layer   |                         | 60            |                              |                          |    |
|        | Dense layer             | 60   |                              |                          | 9             | Convolutional           | 32            | 0.9835                       | 0.9746                   |    |
| 5      | Convolutional           | 16   | 0.9687                       | 0.9489                   |               | Convolutional           | 32            |                              |                          |    |
|        | Convolutional           | 16   |                              |                          |               | Convolutional           | 64            |                              |                          |    |
|        | Convolutional           | 32   |                              |                          |               | Convolutional           | 64            |                              |                          |    |
|        | Convolutional           | 32   |                              |                          |               | Convolutional           | 64            |                              |                          |    |
|        | Convolutional           | 64   |                              |                          |               | Convolutional           | 64            |                              |                          |    |
|        | Convolutional           | 128  |                              |                          |               | Dense layer             | 512           |                              |                          |    |
|        | Dense layer             | 1024 |                              |                          |               | Dense layer             | 131           |                              |                          |    |
|        | Dense layer             | 60   |                              |                          |               |                         |               |                              |                          |    |

After trying on 2 categories and then on 60 categories, we obtained highly encouraging results. Thus we used the same CNN model on the complete dataset Fruit-360 [8] with the latest version 5 that contains 131 classes of fruits and vegetables.

#### IV. RESULTS AND ANALYSIS

The Fig. 4 shows the metrics of the CNN model used for training the latest version 5 of the Fruits-360 [8] dataset with 131 categories of fruits and vegetables. The accuracy increased rapidly for the first five epochs and afterwards, the improvements are slow. We obtain the accuracy of 97.46% from the model. We observe that the accuracy improves, and the model loss decreases as we increase the epochs, as depicted in the graph shown in Fig. 6. The rate of change decreases as we increase epochs, and after a certain point, the loss becomes nearly constant.

Model: "sequential\_6"

| Layer (type)                  | Output Shape         | Param # |
|-------------------------------|----------------------|---------|
| conv2d_36 (Conv2D)            | (None, 100, 100, 32) | 896     |
| conv2d_37 (Conv2D)            | (None, 98, 98, 32)   | 9248    |
| max_pooling2d_18 (MaxPooling) | (None, 49, 49, 32)   | 0       |
| dropout_24 (Dropout)          | (None, 49, 49, 32)   | 0       |
| conv2d_38 (Conv2D)            | (None, 49, 49, 64)   | 18496   |
| conv2d_39 (Conv2D)            | (None, 47, 47, 64)   | 36928   |
| max_pooling2d_19 (MaxPooling) | (None, 23, 23, 64)   | 0       |
| dropout_25 (Dropout)          | (None, 23, 23, 64)   | 0       |
| conv2d_40 (Conv2D)            | (None, 23, 23, 64)   | 36928   |
| conv2d_41 (Conv2D)            | (None, 21, 21, 64)   | 36928   |
| max_pooling2d_20 (MaxPooling) | (None, 10, 10, 64)   | 0       |
| dropout_26 (Dropout)          | (None, 10, 10, 64)   | 0       |
| flatten_6 (Flatten)           | (None, 6400)         | 0       |
| dense_12 (Dense)              | (None, 512)          | 3277312 |
| dropout_27 (Dropout)          | (None, 512)          | 0       |
| dense_13 (Dense)              | (None, 131)          | 67203   |
| Total params: 3,483,939       |                      |         |
| Trainable params: 3,483,939   |                      |         |
| Non-trainable params: 0       |                      |         |

Figure 4. CNN model metrics for 131 classes

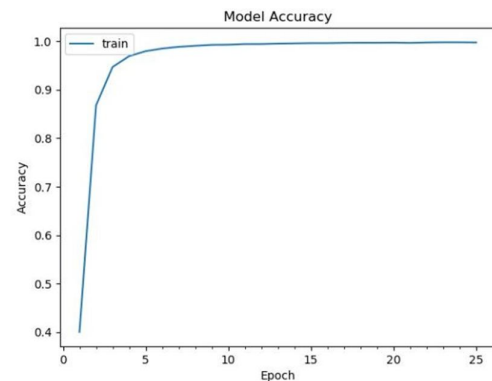


Figure 5. Model accuracy vs number of epochs

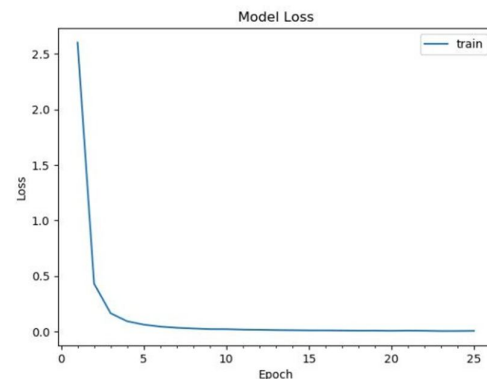


Figure 6. Model loss vs number of epoch

The model classifies the fruits and vegetables from the images given to it, and gives the output. It is evident that the model is able to correctly classify the food image as shown in the Fig. 7. After recognising the image of the food item, and using information about calorie per unit food item from existing sources, calorie count was recorded and is presented as shown in Fig. 8.

#### V. CONCLUSION

In this paper, we have proposed a Convolutional Neural Network-based Deep Learning approach for building a food detection and classification system. By using a different number of epochs and neural net layers, and fine-tuning the hyperparameters, we improved the test accuracy. We have also determined the calorie count of the food, by using the average calorie value of that particular fruit, which was detected. This research work has a multitude of benefits and opens doors for future work, as it could be used in smartphone applications for providing dietary information to the users in a convenient way.

This could prove to be a useful engineering solution. More features such as calculating BMI, water alarm, and diet plan recommendations could be added to the existing solution to build a complete health application. Also, whether an individual should take a particular food item or not could also be recommended, based on the calorie count and the person's current medical history. This shows that future work is based on creating more distributed and complex structures in an automated way.

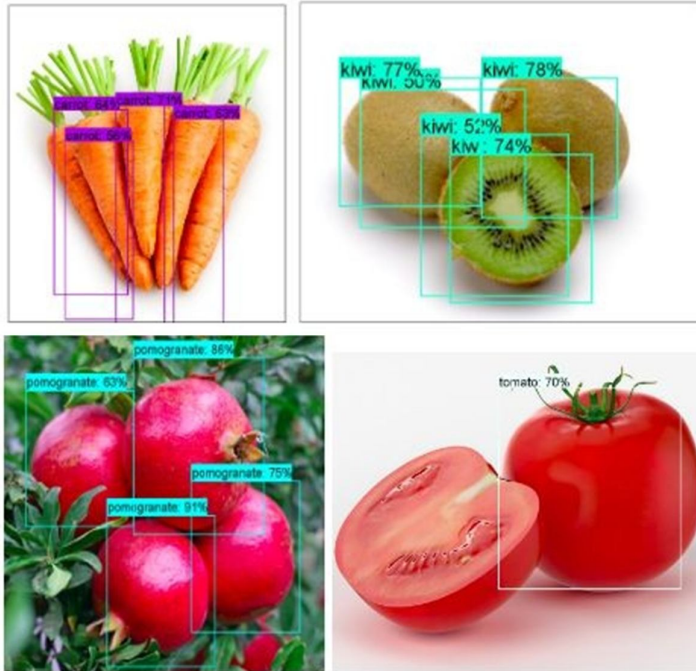


Figure 7. Output of CNN food recognition model

```
TOTAL CALORIES IN IMAGE:210
{'tomato': 1}
TOTAL CALORIES IN IMAGE:22
{'kiwi': 1}
TOTAL CALORIES IN IMAGE:42
{'kiwi': 3}
TOTAL CALORIES IN IMAGE:126
{'kiwi': 1}
TOTAL CALORIES IN IMAGE:42
{'tomato': 4}
TOTAL CALORIES IN IMAGE:88
{'kiwi': 2}
TOTAL CALORIES IN IMAGE:84
{'banana': 1}
TOTAL CALORIES IN IMAGE:105
{'banana': 1}
```

Figure 8. Calorie Estimation Output

## REFERENCES

- [1] World Health Organization. (2019, October) Obesity Study. [Online]. <https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight>
- [2] Pouladzadeh, Parisa & Villalobos, Gregorio & Almaghrabi, Rana & Shirmohammadi, Shervin. (2012). A Novel SVM Based Food Recognition Method for Calorie Measurement Applications. Proceedings of the 2012 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2012. 495-498. 10.1109/ICMEW.2012.92.
- [3] Yu, Liang & Li, Binbin & Jiao, Bin. (2019). Research and Implementation of CNN Based on TensorFlow. IOP Conference Series: Materials Science and Engineering. 490. 042022. 10.1088/1757-899X/490/4/042022.
- [4] TensorFlow. Tensorflow. [Online; accessed 19.05.2020].
- [5] Lin, Tsung-Yi & Maire, Michael & Belongie, Serge & Hays, James & Perona, Pietro & Ramanan, Deva & Dollár, Piotr & Zitnick, C.. (2014). Microsoft COCO: Common Objects in Context.
- [6] Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., and McCool, C. Deep-fruits: A fruit detection system using deep neural networks. Sensors 16, 8 (2016).
- [7] Jia, Songmin & Dia, Chentao & Zhang, Guoliang & Dun, Ao & Sun, Yanjun & Li, Xiuzhi & Zhang, Xiangyin. (2019). Object Detection Based on the Improved Single Shot MultiBox Detector. Journal of Physics: Conference Series. 1187. 042041. 10.1088/1742-6596/1187/4/042041.
- [8] <https://www.kaggle.com/moltean/fruits>
- [9] Oltean, M., and Muresan, H. Fruits 360 dataset on github. [Online; accessed 19.05.2020].