Food Recognition and Nutrition Estimation using Deep Learning

A project report submitted in partial fulfillment of the requirements
For the award of the degree of **Bachelor of Technology** in

Department Of Computer Science and Engineering

By

Sudipta Dandapat(16101104036)
Sudipta Sahana(16101104031)
Sanatan Nandi(16101104033)
Swarnakesar Mukherjee(17101104084)

Under the supervision of

Swalpa Kumar Roy
Assistant Professor
Department Of Computer Science and Engineering

Department Of Computer Science and Engineering
JALPAIGURI GOVERNMENT ENGINEERING COLLEGE JALPAIGURI, WEST BENGAL-735102.
April 2020

ABSTRACT

In today’s world a healthy lifestyle is a must for every individual and what they consume is of utmost importance in order to achieve the same. Our work focuses on creating software which gives the calorie of the fast food which the user is going to consume.

In order to achieve this, the software will take two images as input from the user, the top view and the side view. The image will have a probe object as well which is a coin whose volume will be known.

We used a Deep Convolutional Neural Network (DCNN) for the Food recognition model using Ensemble learning and compared the results with those of other neural network architectures: GoogLeNet, VGG and ResNet, for large-scale image recognition. The results showed that our Deep CNN model achieved better performance in detecting and recognizing fast food compared to other state-of-the-art models with accuracy 92.7% .

Contents

1. Introduction to the Chapter

Maintaining a healthy diet is an important goal for all people. Nowadays, fast food consumption has increased more and more and has become a daily diet of every individual. Fast food may be an easy option for us when we're in a rush, but its nutritional content definitely will not provide us the energy that we need for the rest of the day. So, it's important to estimate the nutritional content present in the fast food we eat.

We already have many different types of tools available online for Nutrition estimation, but, they assume that the user will enter some information about the food item consumed. For example, it might be expected that the user will enter the name of the food item or the ingredients, as well as the size of the food item and then run it against a static database of food items to be able to calculate the amount of calories in the user's consumed food item.

In this project, we came up with an approach to alleviate the user to enter all such details and have the same result with just a food image.

We propose a deep-learning based approach to calculate the calories from the food image through classification of type of food and knowing the weight of the food. Here, we use a pipeline approach by taking a few steps. First, we identify the type of food in the image. Second, we generate an estimated size of food item in grams. Then, by taking the first two intermediate results, we estimate different nutritional content from data-set.

2. Related Work

Automatically predicting the amount of calories in food items based on their images has received some attention in the computer vision domain. For example, Pouladzadeh et al. (2012)[1] proposed an approach to do this by dividing an image of a food item into multiple segments, such that all the pixels represented by one segment have the same characteristics in terms of color, texture, size and shape. After segmenting the image, an SVM classifier was used to predict the amount of calories in the food item. However, their experiments were done only on images of single ingredient food items, which limits the applicability of their approach for the type of food items we are concerned with here such as sandwiches or burgers that typically would consist of multiple ingredients. In addition, their experimental results showed inconsistent performance ranging between 58.13% and 88.34% in accuracy of the prediction based on the type of the food items.

Most Recently Calories Prediction from Food Images (2017) by Manal Chokr, Shady Elbassuoni[2] have more or less the same work as our, where their work was limited to only studio photos. But the scenario in real time is different and we have a bunch of external factors affecting the features which are absent in studio images. We considered both lab and real time images for mixed and good accuracy.

Moreover, a DCNN-based model, FoodNet, was proposed by Pandey et al. [3]. In that model, the dish image recognition system used a large dataset (ETH Food-101) that include 101 food categories.

Moreover, most of these approaches rely on additional information about the food items such as the restaurant that the food item belongs to, or the list of ingredients and cooking instructions of the food item. This all limits the applicability of these approaches.

3. Literature Review

3.1 Transfer Learning

Transfer learning generally refers to a process where a model trained on one problem is used in some way on a second related problem.
In deep learning, transfer learning is a technique whereby a neural network model is first trained on a problem similar to the problem that is being solved. One or more layers from the trained model are then used in a new model trained on the problem of interest.
Transfer learning has the benefit of decreasing the training time for a neural network model and can result in lower generalization error.
We can use Transfer Learning as:
➔ Classifier: The pre-trained model is used directly to classify new images.
➔ Standalone Feature Extractor: The pre-trained model, or some portion of the model, is used to pre-process images and extract relevant features.
➔ Integrated Feature Extractor: The pre-trained model, or some portion of the model, is integrated into a new model, but layers of the pre-trained model are frozen during training.
➔ Weight Initialization: The pre-trained model, or some portion of the model, is integrated into a new model, and the layers of the pre-trained model are trained in concert with the new model.

There are perhaps a dozen or more top-performing models for image recognition that can be downloaded and used as the basis for image recognition and related computer vision tasks.

Perhaps three of the more popular models are used in our work:

1. **VGG (e.g. VGG16 or VGG19).**
2. **GoogLeNet (e.g. InceptionV3).**
3. **Residual Network (e.g. ResNet50)**.

These models are both widely used for transfer learning both because of their performance, but also because they were examples that introduced specific architectural innovations, namely consistent and repeating structures (VGG), inception modules (GoogLeNet), and residual modules (ResNet).
Keras provides access to a number of top-performing pre-trained models that were developed for image recognition tasks.

3.2 Ensemble Learning

Neural network models are nonlinear and have a high variance, which can be frustrating when preparing a final model for making predictions.
Ensemble learning combines the predictions from multiple neural network models to reduce the variance of predictions and reduce generalization error.
Stacked generalization is an ensemble method where a new model learns how to best combine the predictions from multiple existing models.

4. Proposed Methodology

Our system will take an input image of a food item and outputs the nutritional content present in this food item like total fat, calories, carbs, sugar and protein. To be able to do this, the image is passed through a classifier which classifies the category of fast food it belongs to. This is explained in more detail in Section 4.2. Next, we will do the volume estimation using a calibration object for the calculation of calories and other nutritional content explained in Section 4.3. Finally, the predicted type and size of the food item are passed to another regressor that predicts the amount of nutritional content in the food item. This is described in detail in Section 4.4.
The process of calculating the calorie value from the food images has been depicted in Figure 1.

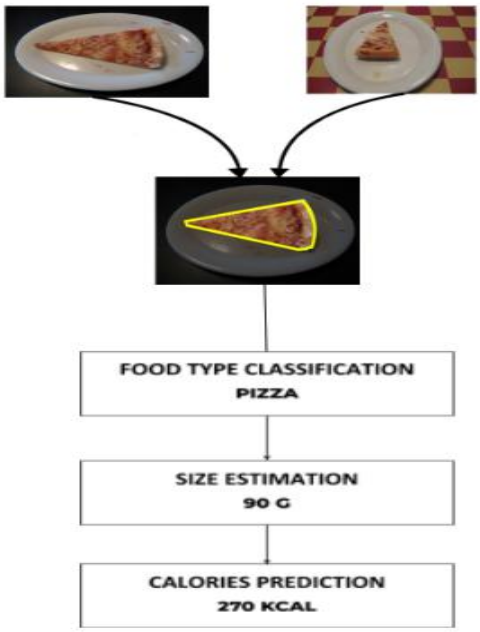


Fig. 1 Flowchart of the model developed

4.1 Data Collection

Our dataset is based on the Food-101 Data Set, which consists of images of food items belonging to over 101 categories.For each class, 250 manually reviewed test images are provided as well as 750 training images. In our dataset, we sampled 1,400 images from the Food-101 dataset, which were evenly distributed among 14 food types, namely: Muffins, Noodles,burger, Egg roll, cheese pizza, pepperoni pizza, french fries, sandwich, hot-dog, fried chicken, biscuits, shish kebab, doughnut and tacos.

We restricted the number of images used and the food types to make it less time-consuming process, which had to be done very carefully in order to produce high-quality ground truth data. And for the information of Nutritional content of these foods we use another dataset which is Nutritional Data for Fast Food 2017.

Item	Type	Serving Size (g)	Calories	Total Fat (g)	Saturated Fat (g)	Trans Fat (g)	Sodium (mg)	Carbs (g)	Sugars (g)	Protein (g)
Hamburger	Burger	98	240	8	3	0	480	32	6	12
Cheeseburger	Burger	113	290	11	5	0.5	680	33	7	15
Big Mac	Burger	211	530	27	10	1	960	47	9	24
Quarter Pounder with Cheese	Burger	202	520	26	12	1.5	1100	41	10	30
Bacon Clubhouse Burger	Burger	270	720	40	15	1.5	1470	51	14	39
Double Quarter Pounder with Cheese	Burger	283	750	43	19	2.5	1280	42	10	48
Chocolate Shake (12oz)	Milkshake	257	530	15	10	1	160	86	63	11
Premium Crispy Chicken Classic Sandwich	Breaded Chicken Sandwich	213	510	22	3.5	0	990	55	10	24
Premium Grilled Chicken Classic Sandwich	Grilled Chicken Sandwich	200	350	9	2	0	820	42	8	28
Chicken McNuggets® (4 piece)	Chicken Nuggets	65	190	12	2	0	360	12	0	9
Small French Fries	French Fries	75	230	11	1.5	0	130	30	0	2

4.2 Food type Classification

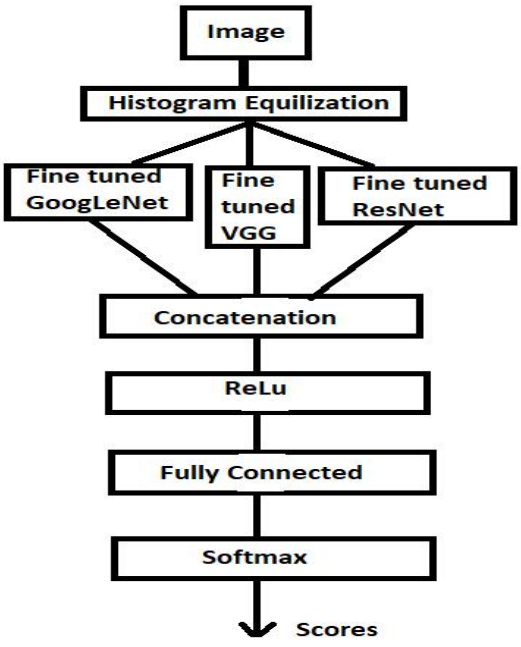
Given an image of a food item, our goal here is to predict the type of the food item.

For this task we use the concept of Transfer Learning in which some famous pre-trained models are used to classify new images. In our case, we first use three models:

1. **VGG (e.g. VGG16 or VGG19).**
2. **GoogLeNet (e.g. InceptionV3).**
3. **Residual Network (e.g. ResNet50)**

And then analyse their individual performance. After that, we develop a network using ensemble learning that would be deep enough, as it increases accuracy and yet have significantly less parameters to train. We choose VGG architecture as our baseline. GoogLeNet architecture uses the sparsity of the data to create dense representations that give information about the image with finer details. ResNet architecture addresses the problem of degradation of learning in networks that are very deep. In essence a ResNet is learning on residual functions of the input rather than unreferenced functions.

A Convolutional Neural Network (CNN) usually consists of convolutional layers and pooling layers [13]. Notations w and h represent width and height, ch is the RGB color channels of the input image I (w, h, ch). the convolutional layer and max-pooling are denoted as C and MP, the convolutional layer is C (k, cs, o) and takes kernel size (k), convolutional strides (cs), and the number of output feature maps is (o) as arguments. The pooling layer MP (r, ps) considers the side length of the pooling receptive field (r) and the pooling strides (ps). In addition, FC (c) and F (class) correspond to the fully-connected layer and the output layers, respectively, where (n) is the number of nodes and (c) is one of the food categories. Notation D stands for dropout. All convolutional layers use ReLU as an activation function. The DCNN model (M) is thus represented by:



M⇒ I (150, 150, 3) → C (9, 2, 32) → C (7, 2, 64) → [C (1, 1, 128), C (3, 2, 128), C (5, 2, 128)] → concat → [C (1, 1, 128), C (3, 2, 128), C (5, 2, 128)] → concat → MP (2, 2) → D → C (3, 2, 256) → MP (2, 2) → C (3, 2, 512) → MP (2, 2) → FC (2048) → D → FC (2048) → F (class).

Finally, there are 14 SoftMax neurons in the output layer, which corresponds to the 14 groups of fast food. This becomes an additional feature that is fed to the Nutrition predictor that we will describe later.

4.3 Food size estimation

In this step, we need two input images from top view and side view to estimate the volume of the food. And each image should include the calibration object which is a 10 Rs coin in our case (diameter 2.5cm). For doing this, we will perform the following steps:

1. **Deep Learning Based Objection Detection:** We use YOLOv3 model to perform object localization and detection on images and draw boundary box around the main food object.

2. **Removing background and unwanted noise:** After segmentation of each boundary box, we will replace the values of the background pixels being by zeros. This will remove the unwanted background and leave only the foreground pixels.

3. **Volume Estimation:** To estimate the volume, we calculate the scale factors based on calibration objects(10Rs coin of diameter 2.7cm). The side view's scale factor(aS) and top view's scale factor(aT) was calculated with Equation 1 and Equation 2 respectively.

$$\alpha_T = \frac{2.7}{(W_T + H_T)/2} \qquad \alpha_S = \frac{2.7}{(W_S + H_S)/2}$$

where, Ws and Hs are width and height of side view. Wt and Ht are width and height of top view.

Now, we estimate the volume using the following formula in Equation 3:

$$v = \beta \times (s_T + \alpha_T{}^2) \times \sum_{k=1}^{H_S} \left(\frac{L_S^k}{L_S^{MAX}}\right)^2 \times \alpha_S$$

HS is the height of side view PS and LkS is the number of foreground pixels in row k (k ∈ 1,2,…,HS). LMAX = max(Lk ,…,Lk), it records the maximum number of

foreground pixels in PS. ß is a compensation factor (default value = 1.0).

4. **Mass Estimation :** After estimating the volume, the next step is to estimate each food's mass. It can be calculated in Equation 4, Where v (cm^3) represents the volume of current food, and ρ (g/cm^3) represents its density value

$$m = \rho \times v \tag{4}$$

4.4 Nutritional content Estimation

Finally, we describe our main task, which is Nutrition content estimation of a food item. Given the input image we perform food type classification and size estimation and after that it passed to a regressor which outputs predicted amount of nutritional content in the food item. Our regressor is trained with the Nutritional Data for Fast Food 2017 dataset. This dataset contains information about Nutritional contents like calories, Fats, proteins and sugar in grams(g) of different types of fast food along with the serving size. Once the regressor is trained with the dataset, it will be used to predict the amount of Nutritional content in a food item.

5. Experimental Result

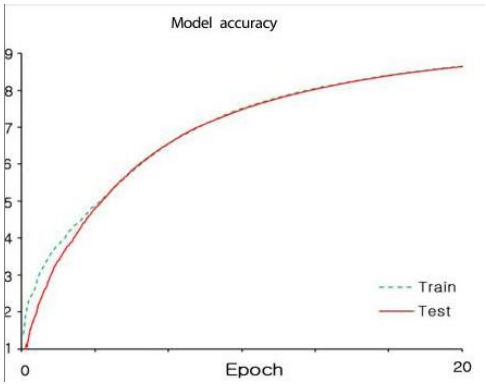
In this section, we present the experimental results of food type classification.. All our experiments were run on an Intel(R) Core(™) computer with a 2.30 GHz CPU and x64 based processor .

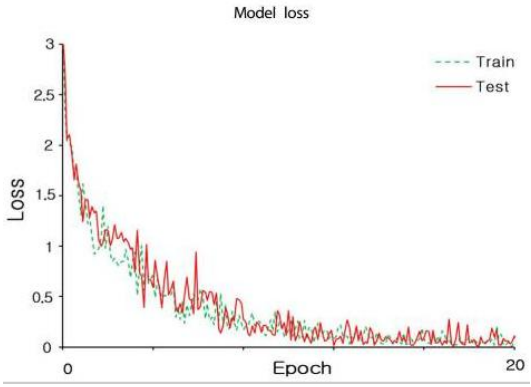
After training the model with our dataset, we compare the accuracy of different Models that are shown below:

Model Name	Test Accuracy	Prediction time

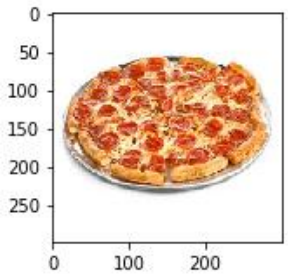
GoogLeNet	91.2%	0.87ms
ResNet	89.1%	1.2ms
VGG	84.6%	0.79ms
Ensemble Model	92.7%	0.42ms

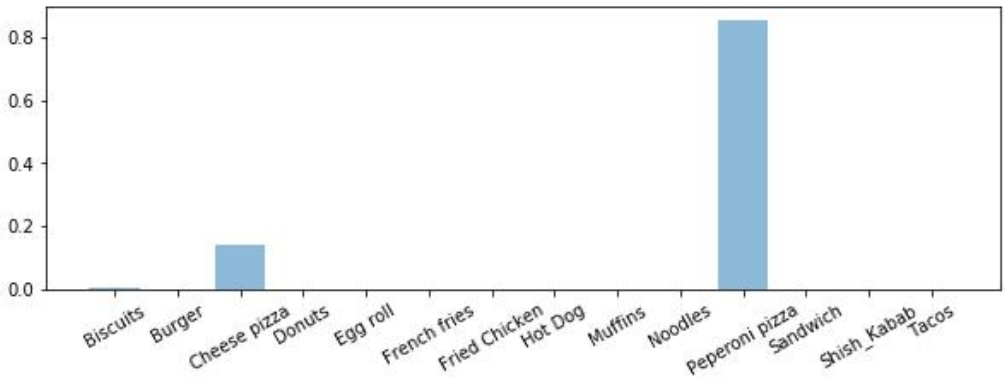
We had also shown the plots for **Accuracy and Loss of Ensemble Model.**





```
Out[54]: array([2.6111677e-03, 8.1516548e-05, 1.4271200e-01, 1.8771812e-05,
                1.3164656e-04, 1.1512225e-05, 3.9308568e-05, 1.8925935e-05,
                1.6672822e-04, 1.4358731e-04, 8.5369164e-01, 1.1417697e-04,
                6.7566441e-05, 1.9141388e-04], dtype=float32)
```





6. Future Scope

In future, we will extend our work with the following:

➔ We will increase our dataset to include more food types other than the 14 types we experimented with here.

➔ We will extend our system to handle the more realistic scenario where the user provides an image of a meal rather than just one individual food item as we assumed here.

➔ We will remove the restriction of two input images from top view and side view. So that we can estimate food size with only one single using 3D/2D model-to-image registration.

7. Conclusion

In this project work, we tackled the problem of predicting the amount of Nutritional contents in food items solely based on their images. To achieve this, we adapted a pipelined approach that first predicts the type and size of the food item in the image, then uses this information to predict the amount of calories , fats, carbs, protein and sugar in the food item. All our prediction tasks were performed using deep learning with some standard pre-trained model like InceptionV3 for object classification and YOLOv3 for object detection in fast food images. We compared our pipelined approach to a baseline approach that directly predicts the amount of calories based only on the image, and showed a reduction in prediction accuracy.

8. REFERENCES

[1] R. Almaghrabi, G. Villalobos, P. Pouladzadeh, and S. Shirmohammadi, "A Novel Method for Measuring Nutrition Intake Based on Food Image," in Proc. IEEE International Instrumentation and Measurement Technology Conference, Graz, Austria, 2012, pp. 366 - 370.

[2] Manal Chokr, Shady Elbassuoni:Calories Prediction from Food Images. AAAI 2017: 4664-4669

[3] Pandey P, Deepthi A, Mandal B, Puhan NB. FoodNet: recognizing food using ensemble of deep networks. IEEE Signal Process Lett. 2017;24:1758–1762.