Food Calories Estimation Using Deep Learning Neural Network

Ege Ucak
21527467
egeucak75@gmail.com

Ali Baran Tasdemir
21627653
alibaran@tasdemir.us

Asma AIOUEZ
21504074
asma.aiouez@gmail.com

Abstract

Understanding what's in the food we consume paves the way for a healthier lifestyle. People who check the food labels grant themselves the chance to compare the nutrient content of different options; hence, make healthier choices. A healthy diet is crucial throughout our lifetime and paying attention to nutrition labels is a major step toward enhancing the overall diet. This paper proposes a model which automatically estimates the calorie count of food as well as the nutrient information. Our method employs deep learning techniques for accurate food identification and other necessary information. In addition to image analysis, we extract the ingredients which will in turn be utilized for calories calculations. Our experiment has been performed with a dataset comprising of 100+ food types.

Keywords: food calories, health, computer vision, deep learning, neural network.

1. Introduction

Consuming food with high calorie intake is a harmful act and results in obesity, which is a preventable medical condition [1] that causes abnormal accumulation of fat in the body. It can result in numerous diseases such as obesity, diabetes, cholesterol, heart attacks, blood pressure, breast, colon and prostate cancers. Most of the world's population live in countries where overweight and obesity kills more people than underweight [1]. The primary cause of obesity is a combination of excessive food and lack of physical activities [2]. Diet and weight management is our major The aim is to raise awareness about the food intake and its nutritional value which would result in a significant decrease and prevention in earlier stages of the above mentioned diseases.

Since not all calories are the same, and for the most part they can be classified as good and bad, we decided to not only display the calories count of food but well also display its corresponding macronutrients information which include carbohydrate, protein, and fat values in grams (see Table 1). The motivation behind this is due to the fact that

Macros	Energy from macros per 1 g
Carbs	4 cal
Protein	4 cal
Fat	9 cal

Table 1. Energy obtained from macronutrients.

the trio are the 3 main suppliers of nutrients in the diet. In addition to providing the body with energy, macros serve a lot of other vital functions. In a diet, carbs are considered to be the most importantproviders of energy in terms of mental and physical activity whereas protein functions as ahormone, enzyme and an antibody in the immune system. Fats on the other hand regulate metabolismand maintain the elasticity of cell membranes. They also improve blood flow and are important for cell growth and regeneration. In such fashion, tracking he macros instead of just caloriesensures that theconsumed calories are all going to the right places in the body hence keeping a healthy body, preserving lean muscle mass along with removing unwanted body fat.

One thing that needs to be mentioned though is that for the volume size, we assume that all calculations will be for one portion size which depends on the food type. Hence our model wont have to deal with the volume estimation of the food displayed in the picture. Our main focus is to give a general idea of calories count as well as the corresponding macronutrients values.

This paper is trying to provide a more efficient way of estimating calories and nutrient facts more specifically macronutrients ones. Th process is as follows: first, we feed the different food images to the network to be processed. The pictures dont need to be taken in a certain angles as long as the food content is displayed in a clear manner. As a next step, the model will extract the ingredients of each food type which in turn will be used to calculate the calories as well as the macronutrients. Unlike the other related works, we rely on the ingredient types to produce the calorie count and nutrient facts rather than the volume of food under the assumptions that were previously mentioned.

The rest of the paper is organized as follows: In section 2 we present the related work. Section 3 explains the method we employ. In section 4, we describe the initial experimental settings.

2. Related Work

In this section, we will review the proposed methods from other research papers for automatic calorie estimation.

Liang and Li [3] proposed a computer-vision based calorie estimation method that estimates both volume and calorie. They are using calibration methods to make an accurate volume estimation. The image dataset used by this method taken with iPhone 4S and iPhone 7 with different lighting conditions. They used 2 different angles for images. One of them from the top, almost 0 degrees to the table and the other one is the side view, almost 90 degrees to the table. And there is a calibration object to estimate the volume of the food in the image. The proposed system is a deep learning model which includes Faster R-CNN. Faster R-CNN used for getting bounding boxes of foods and classifying the foods in that bounding boxes. The GrabCut algorithm used for getting more precious contour data from the bounding box provided with Faster R-CNN. According to calibration object from images, the true size of a pixel is known. After this phase, the volume of the food can be calculated with different mathematical formulations and calorie can be calculated by searching related tables. To estimating accurately, the calibration object and the shooting angles of photos are crucial.

Ege and Yanai [4] proposed another method that estimates calorie of the food with helping from food categories, ingredients and cooking directions. They are using a system based on several steps. The first step, food category estimation as a classification problem. After that, they are using food recipes to ingredient estimation. They are using Word2Vec to symbolize ingredients as real number vectors. Also, they are using cooking directions information and process them like ingredients. They propose a multitask CNN for the food calorie estimation with simultaneous learning of food calories, categories, cooking directions and ingredients. For this research, the quality of the dataset is very important.

Chen et al. [5] proposed an image-based food calorie estimation method using depth information with gathered with special depth cameras such as Kinect. Their system automatically identifies food categories by using feature vectorising such as SIFT and multi-class SVM and multi-class Adaboost algorithm. After that, they are using depth information to quantity and nutrition estimation. The main problem with this research is depth cameras. Because these special cameras like Kinect are expensive and rare for daily usage.

As described above, there are some different ways to perform this task. More common and standard approach for estimation is estimating both volume and calorie from food photo. The other approach is direct calorie estimation. With this approach without estimation category and volume of the food, the system estimates the calorie of the photo. The main issue with this approach is the lack of calorie-annotated food images.

3. Method

We are going to employ a 2 stages network. The first stage is going to be for extracting ingredient information, and the seconds is for estimating nutrient values.

First stage of the network is going to be CNN only, and it is going to be fine-tuned variant of ResNet18. We are going to use ResNet because it contains less parameters, which makes is faster to train. And its depth is also going to give us higher accuracies.

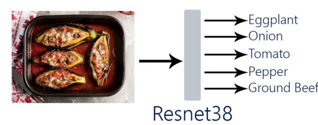

Figure 1. First section of the pipeline

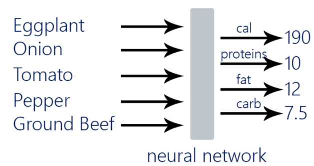

Figure 2. Second section of the pipeline

For the first stage, it is going to take images from the dataset we prepared. The dataset has around 2000 food type in each we have 50 images. We send the images as input into the CNN which will predict the ingredients that have been mined from a combination of EDAMAM's[6] Recipe Search API, and TextRazor's[7] NLP api. The final dataset for ingredient and nutrient info can be found at Google Drive[8]. In total, we come up with 957 different types of ingredients.

The seconds stage of the network is going to consist of fully connected layers, which is going to map ingredients to nutrition values. For getting the best accuracies, we are going to experiment with different layer topologies.

4. Experiments

4.1. Experimental Setup

For the images, at first we planned to use Food-101[9] dataset. However, since our problem is rather a regression problem instead of a classification one, the more food classes we had the better our model would learn. Hence, we constructed our own dataset which can be downloaded from Google Drive[8]. The dataset consists of the name of the food and an example image of the food, its ingredients which have been extracted from its recipe, and its nutrition info. Unfortunately, due to the huge size of our data and in order to speed up the process and obtain some results, we had to process only a portion of our dataset. We basically reduced the dataset size to 5000 images. As mentioned earlier, the first stage has the finetuned ResNet18 architecture in which we tried two models. For the first case only the last layer has been updated and for the second case, all layers have been frozen except the last three ones. Whereas for the second stage of the training, the architecture used is as follows:

```
n_in, m_in,  n_out = 957, 2048, 4
model_fc = model = torch.nn.Sequential(
    torch.nn.Linear(n_in, p_in),
    torch.nn.ReLU(),
    torch.nn.Linear(p_in, p_in),
    torch.nn.ReLU(),
    nn.Dropout(0, 5),
    torch.nn.Linear(p_in, p_in),
    torch.nn.ReLU(),
    torch.nn.Linear(p_in, p_in),
    torch.nn.ReLU(),
    nn.Dropout(0, 5),
    torch.nn.Linear(p_in, n_out),
)
```

4.2. Experimental Results

The following graphs represent the loss graphs when extracting the ingredients:

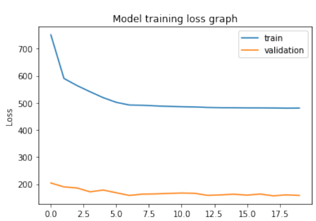

Figure 3. Loss graph for the architecture with last layer finetuned.

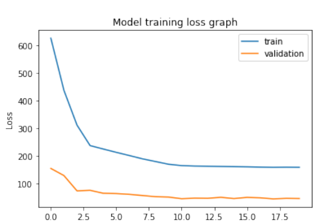

Figure 4. Loss graph for the architecture with the last three layers finetuned.

Freezing less layers of the architecture allows for better loss results since the model is able to learn more parameters of the dataset that is processing. The loss graph that we obtained from the second stage is as bellow:

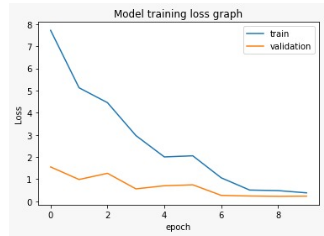

Figure 5. Loss graph for the fully connected architecture during the training.

With the model saved, we ran two test images and observed the following results:

Figure 6. Sample test food "Imam Beyildi".

```
Testing imam-bayildi.jpg
[0.11248967 0.12807585 0.04610611 0.04478601]
['Acini di pepe', 'Ale', 'Banana', 'Carrot',
'Cocoa bean', 'Corn kernel', 'Herb',
'Rib eye steak', 'Tart']
{'calories': 1228.143525474832,
'fat': 129.95600085029093,
'carbs': 60.29507010272733,
'protein': 49.682796556562295}
```

Figure 7. Sample test food "Nohut Yemegi".

```
Testing nohut-yemegi.png
[0.14840288 0.1047461  0.09721258 0.06526579]
['Bouillon cube', 'Cornbread', 'Foie gras',
'French fries', 'Garlic', 'Grits', 'Half and half',
'Mango', 'Nc chm', 'Plum tomato', 'Red cabbage',
'Royal icing', 'Spare ribs', 'Sriracha sauce',
'Vegetable oil', 'Yellow onion']
{'calories': 1620.2380155631333,
```

```
'fat': 106.28377175028066,
'carbs': 127.12932003173113,
'protein': 72.40178777032554}
```

As mentioned earlier, we have a total of 957 different ingredients and the way the ingredients are acquired from the CNN are selected by picking those whose probability is high. We set the threshold to be .95 which would tell us that the ingredient at that position is more likely to exist in that food. As we notice, the results and ingredients extracted are not as accurate and that's due to the reason that our model has been trained for a relatively small dataset.

5. Discussion and Conclusions

The idea behind this architecture is for better nutrient calculations based on the images it receives rather than a standard food definition. However, to achieve good results the dataset was huge in size and training was not carried out due to the time constraint we had. So we cut the dataset and ran the tests and as expected our model failed to generalize the instances it didn't see before.

References

[1] October) World Health Organization. (2011. Obesity study. [online]. http://www.who.int/mediacentre/factsheets/fs311/en/index.html.

[2] World Health Organization. (2012). World health statistics 2012. [online]. https://www.who.int/gho/publications/world_health_statistics/2012/en/index.html.

[3] Yanchao Liang and Jianhua Li. Computer vision-based food calorie estimation: dataset, method, and experiment, 2019.

[4] Takumi Ege and Keiji Yanai. Image-based food calorie estimation using knowledge on food categories, ingredients and cooking directions. *Proceedings of the on Thematic Workshops of ACM Multimedia 2017 - Thematic Workshops '17*, 2017.

[5] Mei-Yun Chen, Yung-Hsiang Yang, Chia-Ju Ho, Shih-Han Wang, Shane-Ming Liu, Eugene Chang, Che-Hua Yeh, and Ming Ouhyoung. Automatic chinese food identification and quantity estimation. *SIGGRAPH Asia 2012 Technical Briefs on - SA '12*, 2012.

[6] Recipe search api. https://developer.edamam.com/edamam-docs-recipe-api.

[7] Text razor nlp api. https://www.textrazor.com/.

[8] Ingredients dataset. https://drive.google.com/open?id=1dQBbRawtSoIxCsT0sVp4bBFmsIbrG3Tg.

[9] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.