Food Nutrition and Calorie Estimation

Andrew McMullin
Huanchen Sun
James Wiaduck
Amanda (Yue) Yao
mcmullin, huanchen, jwiaduck, yueyao@umich.edu

Abstract

The goal of this project is to use machine learning and machine vision to implement a food recognition and calorie estimation system. This implementation is attempted by using a Mask-RCNN (Detectron2) to find food in an image and then using a Multi-task CNN to make calorie and nutrition predictions without knowing the food category. However, the implementation failed to train and was unable to make reasonable predictions with noisy datasets representative of the real world. With further demonstration of our work and analysis of what could have gone wrong with our system, we conclude several important findings to successfully develop such a system and give future research direction recommendations.

1 Introduction

As social media and internet blogs and websites become more popular, people are being overwhelmed online both by food content and a lack of nutritional information. The rise of eating disorders and anxieties might be exacerbated by fit looking influencers and aesthetic plates. Do these meals deliver the nutritional information they promises?

Despite seeing more food content than ever, we have never felt so far removed from the actual nutritional content of what we see online. We present a system to use machine learning and machine vision to implement a food nutrition and calorie estimation system. The goal of this project is to fill the gap between accurately detecting food and accurately estimating nutritional and caloric information for that food from an image.

There is a lack of currently available and viable systems that can accurately estimate nutritional and caloric information from an image alone. This problem entails object detection and recognition of the food, as well as prediction of the caloric and nutritional content. This draws mainly on machine learning techniques popular in computer vision.

Although there has been mild success with very strict datasets using Faster-RCNN for object detection and a Multi-task CNN for the calorie nutritional estimation (Ege and Yanai, 2017b,a), there is a lack of generalizability and reproducibility in currently proposed solutions. They require food categorization at the first step. Furthermore, there is a lack of accurate methods to quantify food amounts from an image alone.

We implemented a machine learning pipeline to estimate the caloric and nutritional content of food present in an image. We used a Mask-RCNN for object detection and Multi-task CNN with a custom loss function for food nutrition and calorie estimation without the requirement of food categorization.

Unfortunately, we found the system was unable to accurately train with any loss. With further evidence from our training experiments, we conclude that it is not possible to learn calorie information directly from a single image using only calorie information and nutrition facts as labels.

We first discuss related work. After an overview of the data we worked with, we will discuss our methods and models. We will then discuss our results and an analysis of our system and its failure. Finally, we present further research and ethical considerations that should be taken into account when addressing problems or deploying solutions such as this.

2 Related Work

With the rise of people's attention to healthy diets, people have been trying to develop tools for auto food image calorie estimation. (Wang et al.,

2021) In this era of computing, machine learning and computer vision models empower us to achieve automatic food calorie and nutrition estimation almost instantly.

Prior works focused on utilizing computer vision models for food categorization and volume estimation. Various model structures were developed, but most prior models depend on food categorization in their first stage. Therefore, prior works are mainly limited to analyzing food whose categories are easily detectable. Of most prior models, the food categorization stage was trained and tested on datasets of a fixed amount of categories. (Lu, 2016)

$$calorie = calorie/gram \times density \times volume \quad (1)$$

Moreover, prior works showed high accuracy in detecting food categories. With appropriate calorie-per-gram and food density information. According to equation 1, the only hardness of estimating food calories is to estimate food volumes. (Min et al., 2021)

2.1 Volume Estimation based on Calibration Object

Some prior works used calibration objects with fixed sizes to estimate the relative size of food. (Liang and Li, 2017) Example calibration objects include coins and ping-pong balls. Because trained models need to detect the calibration objects on images, both training and testing datasets need to have calibration objects on images. Moreover, the model could not be applied to scale due to the requirement of calibration objects.

2.2 Volume Estimation based on Multi-task CNN

To extend the use of the calibration-required calorie estimation models, one of the prior works proposed a Multi-task CNN stage. (Chauhan et al., 2018; Ege and Yanai, 2017b) As shown in Figure 1, the model skipped the explicit volume estimation stage. Given pre-processed images as inputs and trained on categories, the Multi-task CNN serves as a black box for implicit volume and calorie estimations. (Ege and Yanai, 2017b)

Inspired by the introduction of Multi-task CNN in the field of food calorie estimation, we decide to break the limit that food image calorie estimation must require food categorization. In this paper, we introduce an improved model that attempts to estimate calorie and nutrition information without the requirement of food categorization.

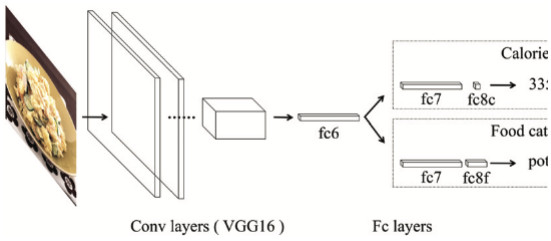

Figure 1: Overview of the Multi-task CNN in prior works

3 Data

As mentioned in prior parts, this paper focuses on innovating current food calorie estimation models and surpassing the requirement of food categorization. Therefore, prior datasets built strictly for food categorization are not applicable to our project. (Ege and Yanai, 2017b) Moreover, instead of regular CNNs, we apply a Multi-task CNN to supervise calorie information prediction. Therefore, we need more labels to train the model on. Among all possible labels, we picked the amounts of protein, fat, sugar, and carbohydrates as good indicators to supervise the calorie training process. (Ruede et al., 2020; Aiello et al., 2019; Wali, 2021)

3.1 Raw Data Resources

In searching for food images that corporates with calorie information and other nutrition information including the amounts of protein, fat, sugar, and carbohydrates, we focus on commercial cooking recipe sites on the Web.

For the experiments of this work, we collected data from 5 websites shown in Table 1. Each recipe presented on these sites contains food images, ingredients, cooking procedures, and nutrition information. To form our datasets, we web-scape 10,000 pairs of data and cleaned the data such that images are not defaults of websites and there are no missing entries in the nutrition information that we need.

3.2 Pre-Processing Raw Data

We need to pre-process the raw data we collected for our training process. Calorie and nutrition information is easily retrievable through text cleaning and editing. We store source URLs and filenames

Website	Number of Samples
eatingwell.com	4056
bbcgoodfood.com	2733
taste.com.au	1399
olivemagzine.com	1013
goodto.com	994

Table 1: The commercial cooking recipe sites we used for data scraping and the number of collected samples.

of images, calorie information, and nutrition information in neatly formatted JSON files. (Crockford, 2006)

Conversely, images are harder to pre-process: for the sake of artistic designs, food images do not always put food in the center. However, we need to align the food to the center of images for the training process of our Multi-task CNN layer. We perform such image transformation using the state-of-the-art object detection model Detectron 2, which we will discuss in the next section of this paper. (Wu et al., 2019a)

4 Methods

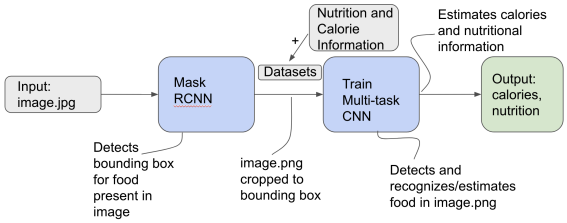

Figure 2: System diagram

Figure 2 shows our overall system diagram. After data pre-processing, we feed our images into a pretrained object detection network, Mask-RCNN, from Detectron2. We extract bounding box coordinates from the network and crop the image to get a centralized dataset. We then train and evaluate our Multi-task CNN network with nutrition calorie labels on the centralized dataset. The final trained model is then integrated into the back end of our web application which allows users to upload a food image and get calorie and nutrition information back. Algorithm details will be introduced in the next section.

5 Algorithms

5.1 Detectron2

Our Algorithm starts by finding where the food is in the image and obtaining a bounding box around it. We use Facebook AI Research Lab's next generation library Detectron2 (Wu et al., 2019b) which uses a Mask-RCNN and is pretrained on a COCO-InstanceSegmentation (Lin et al., 2014) dataset. This model having been pretrained on COCO gives us the ability to find different things in the image. This model finds objects such as Tabletops, Plates, Bowls, Spoons & Forks, Cups, and some types of food. (Wu et al., 2019b)

The Mask-RCNN was developed on top of Faster-RCNN which is a Region Based CNN. The Mask-RCNN makes advancements in image segmentation and instance segmentation. Segmentation classifies each pixel into a fixed set of categories, which classifies similar objects and allows for similar objects to be classified as 2 different objects instead of one large one.

There are two steps to the Mask-RCNN that we need to focus on. First, the model generates bounding boxes. Second, it classifies what's inside the bounding boxes. The bounding boxes are given as list of lists of size 4. Where each sub-list is an object detected in the image and the four values correspond to two (x, y) coordinates defining where the object is in the image. In another list, Detectron2 provides a classification of what each object is from it's list of possible objects, this list is the size of how many objects were detected.

Figure 3: Bounding Boxes

For this project we then take all objects that

are classified as a BOWL, as we can see in Figure 3, and crop the original image at all the points provided by the bounding boxes.

5.2 Multi-task CNN

After getting cropped images from Detectron2, we train our Multi-task CNN network. In this section, we will introduce our calorie estimation model in details. You can also find our code on GitHub (9).

5.2.1 Model Details

We mainly refer to these two papers when we are developing our Multi-task CNN model: (Ege and Yanai, 2017b) and (Ege and Yanai, 2017a). The authors introduced a VGG based model two heads added and trained separately: one of which represents the food class, the other represents the food calorie. They argued that food class helps learn import low level food image features, which with further training, calories can be learned from a single image.

However, as the dataset being used in the paper is pretty restricted and the food class in really hard to define and obtain for our scrapped dataset from online recipe, we decide to slightly modify our dataset and model in the following way: we first further scrape other nutrition facts including protein, fat, sugar, and carb. We add task heads for each nutrition fact and calorie information, which share the same backbone learning weights but are further tuned separately.

Figure 4 shows our model overview. We use VGG16/ResNet18/ResNet34 as our backbone and pick one with the best performance. We add 5 task heads to our backbone and each will estimate a nutrition fact, namely calorie, protein, fat, sugar, and carb. The five task heads share the same backbone since they share some common features that are extracted from the image and the fully connected layers in each head guarantees differences.

5.2.2 Loss

We implement custom loss from (Ege and Yanai, 2017a). The loss function we use is an average over all images of all 5 tasks. It includes two losses: relative error and absolute error. The relative error is given by:

$$L_{reij} = \frac{|ypred_{ij} - ytrue_{ij}|}{ytrue_{ij}}$$

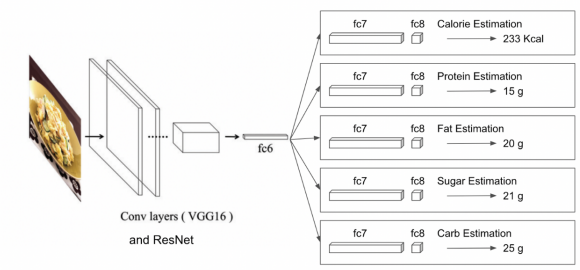

Figure 4: Multi-task CNN for Food Nutrition and Calorie Estimation

and the absolute error is given by:

$$L_{abij} = |ypred_{ij} - ytrue_{ij}|$$

Intuitively, we would like our model to give calorie estimation as accurate as possible within a reasonable error range based on the true calorie. We add a coefficient λ, default value 0.3, to weight the absolute error and the relative error.

Our overall loss function is as follows:

$$L = -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{5} L_{reij} + \lambda L_{abij}$$

where i represents 5 tasks heads, j represents the j^{th} image.

5.2.3 Training Details

In our training exploration, we tried the following variation:

- Backbones: VGG16, Resnet18, Resnet34
- Pretrained: each backbone with pretrained and not pretrained version
- Loss: our custom loss, L1 loss, MSE Loss
- Task Head: calorie head only, calorie head + nutrition heads
- Hyperparameters: Customized loss coefficient λ, num_epochs, learning_rate, batch_size

When training, we prioritize calorie estimation when saving the best model weights.

5.2.4 Evaluation

We use the following 2 metrics to evaluate our model:

- 20% Relative Error of each task: percentage of images where $L_{reij} \leq 0.2$
- 40% Relative Error of each task: percentage of images where $L_{reij} \leq 0.4$

5.3 Web Application

To provide results in a user accessible way and human interpretable way, we built a light weight web application that encapsulates our pre-trained models.

The web application provides a landing page with a simple form where users can submit an image containing food that they would like to know nutrition and calorie information about. The image is then transformed into the expected input format for the Detectron2 model and ran through the Mask-RCNN. The image is cropped according to the models outputs, and then passed into the Multi-task CNN. The caloric and nutritional results are then displayed back to the user, along with a masking picture from the Detectron2 model results. The system is able to deliver results in less than 10 seconds with computation done on the server, and no images or data is ever stored.

To accomplish this, we used a Flask REST API and Python and bundled the server into a Python package. This was then deployed along with the models to an Ubuntu virtual machine hosted on a popular cloud provider. The code for the web application can be found on the GitHub (9).

6 Results and Discussion

6.1 Training Loss and Accuracy

Unfortunately, we found that our model is not learning. Figure 5 shows the overall fluctuating training loss in different epochs. We can see that our model is not actually learning anything from the calorie label and nutrition labels, as the loss is not monotonically decreasing but fluctuating.

Figure 6 is a snippet of one training trial, with 20% error and 20% error metrics for each task head listed. We can see that we have less than 1% image that meets are within 20% and 40% error for each task.

6.2 Possible Reasons

The most possible reason account for the failure in our training process is that we have a noisy dataset. Prior datasets used for food categorization generally include dozens of images for a simple food category. (Bossard et al., 2014) On the contrary, almost every sample in our dataset is a unique category of food. As we designed it, our model cannot learn any categorical information about food.

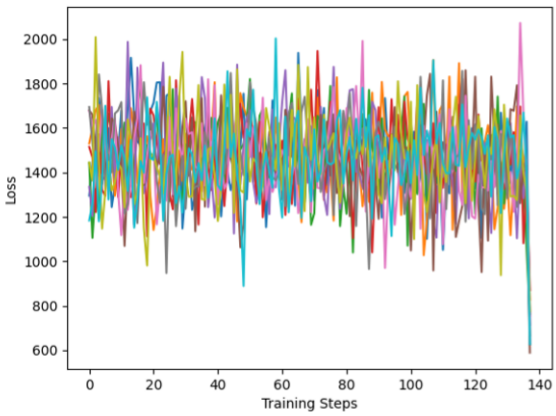

Figure 5: Training Loss through Epochs. Different colors in the plot represents different training epochs. Our training loss fluctuates and does not decrease through our training process.

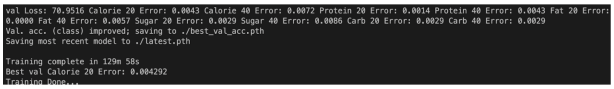

Figure 6: Training snippet with 20% error and 20% error metrics.

However, the results show that it is barely possible to conclude any calorie and nutrition information without a given category. A better illustration is shown in Figure 7. When two food images share similar graphical features but vary much calorie-wise and nutrition-wise, our model cannot distinguish between the two images without given categorical information.

Figure 7: On the left is a bowl of healthy vegetable soup with low calorie; on the right is a matcha cheesecake rich in calorie. Both foods are green and round, and thus might be hard for our model to distinguish.

6.3 Critiques and Discussions

Besides our noisy dataset problem, we carefully question the generalizability of the original paper we referred to (i.e. (Ege and Yanai, 2017b) and

(Ege and Yanai, 2017a)) with our training experiments.

First and foremost, the dataset from the original paper is a very restricted dataset from school lunch meals with standardized sizes and small number of food categories. Such datasets barely exist in real life. Manually labelling our dataset with food category is challenging, as it is difficult to define "food category" in real life. As this is a not super well-known paper, their method itself might be limited by their dataset, which could explain why we did not see similar results in our training.

Secondly, there is a possibility that the calorie estimation model itself has problems. The task that the Multi-task CNN had to solve is a really hard problem for human beings as it is. Yet, we are expecting a CNN model to learn some numbers which do not make any sense to the model from an image. Even with a pre-trained model, this is not like a classification problem where the models have a clearer path and direction to go, as calorie estimation depends on many more complicated and obscure things that are not easily extracted nor categorically defined. If it is the case that these techniques are not able to solve these problems, then our results confirmed this.

Thirdly, to see and understand this flaw in our calorie estimation model and to make an attempt at understanding our problem with a more simple approach or baseline, we used our model to try and classify food images without calorie estimation. This was done by using the FOOD-101 dataset and training using the VGG16 model. Through this model we were still unable to get a good food classification and only got around a 3% accuracy. Through this we understand that it would already be very difficult to classify real-world food with a substantially larger number of categories mentioned in the original paper and therefore almost impossible to use such model to learn calorie information.

6.4 Further Research

As a proof of concept to show that single images based direct calorie information learning is impossible, if given time, we would like to explore the possibility of calorie estimation in the following directions:

- Ingredient recognition based calorie estimation. With enough ingredients images and labels, we can adapt our Multi-task CNN model to an ingredients recognition network and then give calorie estimation based on ingredients. One pitfall would be sometimes it is hard to tell whether an ingredient exist when the food is too processes (e.g. oil in a burger), which was also an issue we suffered from in this paper.
- Volume estimation based calorie estimation. We will need more solid geometric knowledge and proper labels to estimate a 3D volume for food in the image. Some successful paper using this method have really limited lab-based dataset with pretty heavy hand-engineering, which is a real-world challenge.
- A combination of the first method and the second method.

7 Ethical Implications

Here we will go over some of the key ethical problems to consider when doing or using this project's product.

First, your food may not be properly present in the trained dataset. The system is likely to have limitations regarding the performance of the model in heterogeneous populations. Cuisines that are underrepresented in the training of the model are subject to inaccurate output. Cultural differences in food preparation, portions, and presentation may also limit the usability and accuracy of the model.

Second, the caloric prediction is prone to error and should not be misinterpreted, or trusted with 100% certainty. The system is limited to only providing an estimation of caloric and nutritional information, and even then is extremely novel in any sort of accuracy. The estimations are subject to variation and large error and will not provide ground truth information, which may be misleading or harmful to users. Incorrect output can mislead users on the nutritional and caloric content that they are consuming, which may have very negative consequences on the mental and physical health of the users both by perceived perception of diet and the resulting dietary or lifestyle changes that they may make. Examples may include malnutrition or eating disorders.

Third, the system will not provide any general dietary guidance over any time period and should

not be used for such. The system may not be approved by or endorsed by health care professionals, and different countries may have differing requirements for the deployment or use of the system as it relates to nutritional and caloric information. The system has no knowledge of the users current health and medical conditions, further validating that misuse of the system or incorrect output may have severe consequences for the user. For example, a vegetarian may be suggested to consume more meat or less tofu. Thus, the system is prone to limitations and implications arising from lack of context of the users.

8 Conclusion

From our exploration and methods, we conclude that it is not possible to learn calorie information directly from a single image using only calorie information and nutrition facts as labels. Since our project is broken down by modules, any part can be easily updated in the future if a better model is found, allowing us to maintain our model pipeline architecture. We also conclude that the dataset is one of the most important pieces to this puzzle and may have limited our ability to see better results. We saw that in datasets representative of real-world users, we were unable to understand the pictures enough to make accurate predictions, while with a simplified and cleaned dataset one could get closer. We learned a lot from this project and understand much more about a Neural Networks capabilities and flaws, and can see more clearly why our model failed and how we will be able to notice such things in our future projects.

9 Code and Presentation

Our Code Base: Github Link
Our Presentation (demo included): video recording slides

References

Luca Maria Aiello, Rossano Schifanella, Daniele Quercia, and Lucia Del Prete. 2019. Large-scale and high-resolution analysis of food purchases and health outcomes. *EPJ Data Science*, 8(1).

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101 – mining discriminative components with random forests. In *Computer Vision – ECCV 2014*, pages 446–461, Cham. Springer International Publishing.

Rahul Chauhan, Kamal Kumar Ghanshala, and R.C Joshi. 2018. Convolutional neural network (cnn) for image detection and recognition. In *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, pages 278–282.

Douglas Crockford. 2006. The application/json media type for javascript object notation (json). Technical report.

Takumi Ege and Keiji Yanai. 2017a. Estimating food calories for multiple-dish food photos. In *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 646–651. IEEE.

Takumi Ege and Keiji Yanai. 2017b. Simultaneous estimation of food categories and calories with multi-task cnn. In *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, pages 198–201.

Yanchao Liang and Jianhua Li. 2017. Computer vision-based food calorie estimation: dataset, method, and experiment.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2014. Microsoft coco: Common objects in context.

Yuzhen Lu. 2016. Food image recognition by using convolutional neural networks (cnns).

Weiqing Min, Zhiling Wang, Yuxin Liu, Mengjiang Luo, Liping Kang, Xiaoming Wei, Xiaolin Wei, and Shuqiang Jiang. 2021. Large scale visual food recognition.

Robin Ruede, Verena Heusser, Lukas Frank, Alina Roitberg, Monica Haurilet, and Rainer Stiefelhagen. 2020. Multi-task learning for calorie prediction on a novel large-scale recipe dataset enriched with nutritional information.

Milner A.J. Luk A.W.S. et al Wali, J.A. 2021. Impact of dietary carbohydrate type and protein–carbohydrate interaction on metabolic health.

Wenjie Wang, Ling-Yu Duan, Hao Jiang, Peiguang Jing, Xuemeng Song, and Liqiang Nie. 2021. Market2dish: Health-aware food recommendation. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 17(1):1–19.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. 2019a. Detectron2. https://github.com/facebookresearch/detectron2.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. 2019b. Detectron2. https://github.com/facebookresearch/detectron2.