Food 101: A Healthy-eating Camera for Calorie Detection

Jiahe Tan, Jiajia Liang, Yike Guo
Department of Computer Science, University of Virginia, Charlottesville, VA 22904
[jt7gu, jl9pg, yg7jg]@virginia.edu

Abstract

In contemporary society, a healthy lifestyle is becoming more and more popular among the public. However, aside from occasionally strict diet, making health choices is not always easy to maintain when people are placed in a fast-paced and overwhelmed society. Therefore, a reliable and easy-to-use calorie detection camera will have a high demand. User simply need to take a picture of the food, they can now make informed decisions on food intake based on the calories output from our program. In this work, we will use reduced-size Food101 data set. We will first apply AlexNet[3] model designed by Alex Krizhevsky in 2012 to extract features. Then, we will use fully connected layers to classify images into 55 food categories. We will discuss different activation functions performance for our data set, and experiments how the food classification and calorie detection works for real image taken in our daily life. By using Convolutional Neural Network(CNN), we show that it is plausible to create a smart camera that use computer vision technology to classify specific range of food and identify its calorie.

1. Introduction

2000 calories a day is used as a general nutrition advice for adult[2]. However, most people find it hard to get information about food calories and calories intake. Therefore, we proposed to create a Healthy-eating Camera for Calorie detection, which instantly output the calories for the food image took by users.

We choose to use the Food-101 data set, which is originally a data set used by Lukas Bossard, Matthieu Guillaumin, Luc Van Gool for a novel method to mine discriminative parts using Random Forests[1]. The data set has 101 food classes, and 1000 images per class. Within each food class, 750 images are designed for training, and the rest 250 images are designated for validation. Considering the limited time and limited computing resources we can access, we decided to keep the first 55 classes of food, resulting with 41248 training images and 13750 testing images in total. Besides using Food-101 data set, we also upload photos taken in our real life to evaluate how our calorie detection camera works.

2. Related Work

Early work on food detection include using Random Forests to mine discriminative parts that makes it possible to mine for parts simultaneously for all classes and to share knowledge among them[1]. In 2016, Lu applied CNN models for food image classification and data augmentation techniques based on geometric transformation[4].

More recent works on food detection that uses deep learning and image classification technique include a deep convolutional neural network merging with YOLO, a state-of-the-art detection strategy, was built to achieve simultaneous multi-object recognition and localization[5]. Our current work does not put emphasis on multi-object recognition, i.e. recognize the ingredient in the image, but rather we focus on the dishes as a whole. But we acknowledge that the ability to perform multi-object recognition will provide more detailed information about the dishes, and this will be our future goals.

3. Model

We first prepared a relatively clean set of images from the Food-101 Dataset by extracting 55 classes of food through the pretrained AlexNet model. Then we preprocessed the data using AlexNet.

AlexNet was introduced with the publication of the paper, "ImageNet Classification with Deep Convolutional Neural Networks" by Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. It consists of 5 convolutional layers and 3 fully connected layers. The first two Convolutional layers are followed by Max Pooling layers. The third and the fourth Convolutional layers are connected together directly. The fifth Convolutional layer is again followed with a Max Pooling layer. The output of the fifth Convolutional layer is then connected to two fully connected layers, and the output then goes into a Softmax classifier with 1000 out features. Relu nonlinearity activation function is applied to

all of the layers.

Since our data set has 55 classes of food in total. We finetuned the pretrained Alexnet by taking out the last linear layer and added three fully connected linear layers to classify image into 55 classes. The inputs are a tensor with feature extracted by Alexnet, and the output is a softmax layer giving the probabilities of the 55 classes.

We will use two methods to evaluate our network. First, we will track the top 1 prediction accuracy, which we only look at the label with the highest probability. Since we want to display the correct food calories as accurate as possible, at the same time avoid information overhead, we choose to also look at top 5 prediction accuracy rate. If the true label is in the top 5 predictions given by the network, we considered this trail as a successful prediction.

4. Experiments Results and Discussions

We used Pytorch to implement our fully connected layers. We first started with Relu, and then experimented using Sigmoid and Tanh activation function[6]. Afterward, to visualize the performance results, we plotted out the training lost figure, training accuracy figure, and test accuracy figure for each activation function, shown in Figure 2. To briefly compare the results, we came to the conclusion that Tanh have the highest training accuracy as well as test accuracy in general. While Relu did a worse job at both training accuracy and testing accuracy. In terms of execution time, Tanh took more than twice of Relu and Sigmoid's execution time. However, in order to better classfy image and output corresponding calories, we will be using tanh activation functions in the following analysis.

From Figure 2, two observations we found are that 1) there is a huge gap between top 5 prediction accuracy and top 1 prediction accuracy, and the gap is only appeared in testing senario. 2) Only for top 1 prediction accuracy, the difference between training accuracy and testing accuracy is large (68 percent vs 39 percent).

Usually, the gap between training and testing accuracy are caused by the fact that the training data is not large enough to cover the entire distribution of our data set. However, in this case, our data set contains 750 pictures for each class, which we considered to be sufficient. Therefore, we suggest that this discrepancy is due to the nature of images in our data set. Some image are taken in poor lighting conditions, some are mixed with many other food, and some correct label is even tricky and hard for human to classify. Also, some food classes are very similar in terms of ingredient and its looks. For example, the images for Greek salads looks very similar to Caesar salads. Therefore, the network trained for training set might not be a representative network for testing set, since there are too many uncertainty and confounding elements. The influence is most prevailed if we considered only top 1 prediction, since top 5 predictions allow more room for the error caused by the above limitations of the images.

```
('creme_brulee', 0.8)
('french_onion_soup', 0.8)
('grilled_cheese_sandwich', 0.808)
('fried_rice', 0.816)
('bibimbap', 0.82)
('french_fries', 0.832)
('frozen_yogurt', 0.836)
('dumplings', 0.852)
('cup_cakes', 0.856)
('clam_chowder', 0.864)
('hot_and_sour_soup', 0.884)
('edamame', 0.912)
```

Figure 1. 10 highest accuracy food classes

In order to build a stable calorie detection camera, we will be using top five predictions given by Tanh activation functions. For our training data, network using Tanh function achieve 80 percent accuracy and 68 percent accuracy, which we considered to be sufficient. To have a better idea of how our network perform for each type of food, we printed out the accuracy results for each food class, among which the "edamame" class had the highest train and testing accuracy, 96 percent and 91.2 percent respectively. Food classes with the top ten testing accuracy are shown in Figure 1.

To test the usefulness of our prediction model, we uploaded some of the food images we took in our daily life and see how well our network's prediction worked. In Figure 3, We showed that if the images are very representatives, the predictions are very robust and accurate. However, there are few images that our model are failed to identified correctly in top 5 predictions. We noticed that most incorrect predictions are caused by either 1) there are too many ingredients in the image, or 2) the mispredictions are very close to the actual labelling. For example, for a chocolate cake image, all predictions are some forms of baking products.

After testing the robustness of our modal, we loaded in calories information of each food category. Calories information are searched on fitbit.com and manually encoded. In Figure 4, we showed some of the the successful predictions and calories output associated with it. We also tested other different classes of food.

Given the limited time and amount of computing resources at hand, there are many limitations in our results and many improvements that can be done in the future. Specifically, our accuracy rate is not high enough, even using top 5 predictions. Also, since many images in the Food-101 data set are taken poorly, we might need to use a deeper Convolutional network to better extract features and classify data. Second, we can consider doing multi-object recognition tasks to identify ingredients in the image, thus providing more complete calories information to people who want to be informed.

References

[1] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

[2] D. N. Goyal R. Food label reading: Read before you eat. 2018.

[3] A. Krizhevsky. Imagenet classification with deep convolutional neural networks. 2012.

[4] Y. Lu. Food image recognition by using convolutional neural networks (cnns). 2016.

[5] J. Sun, K. Radecka, and Z. Zilic. Foodtracker: A real-time food detection mobile application by deep convolutional neural networks. In *Computer Vision and Pattern Recognition*, 2019.

[6] A. S. V. Understanding activation functions in neural networks. 2017.

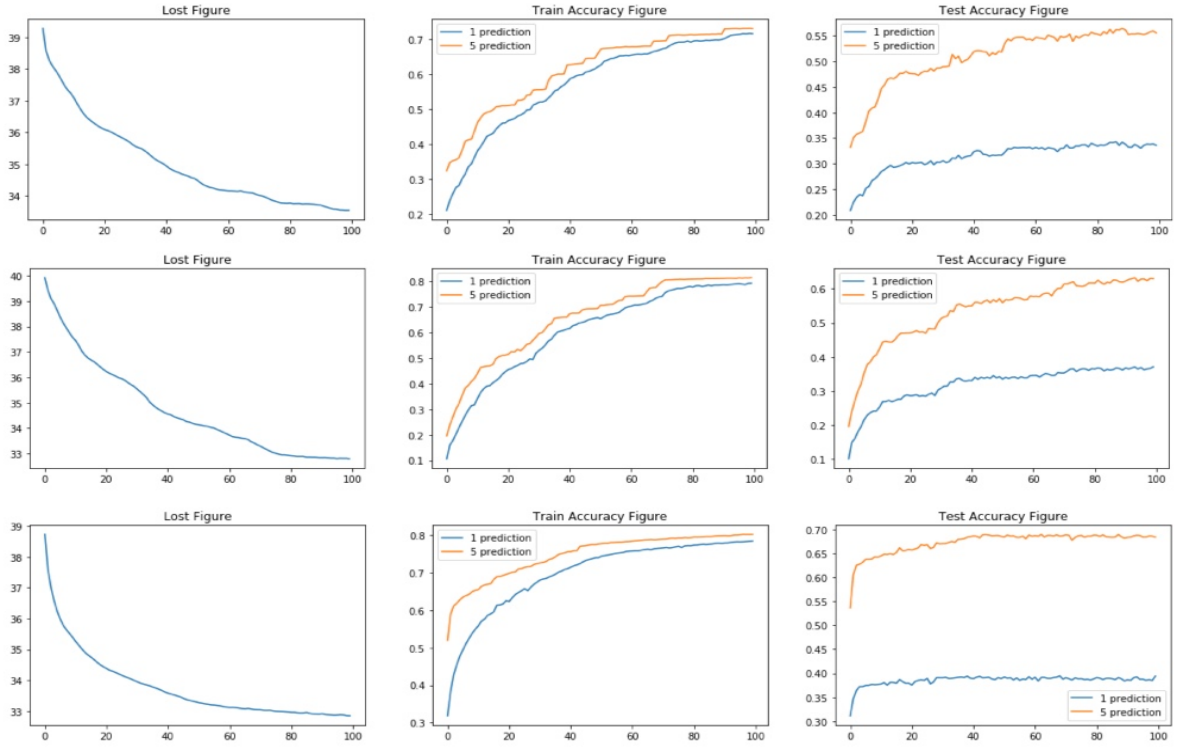

Figure 2. Here we show the corresponding lost and training and test accuracy figures in sequence of the activation functions (Relu, Sigmoid, Tanh).

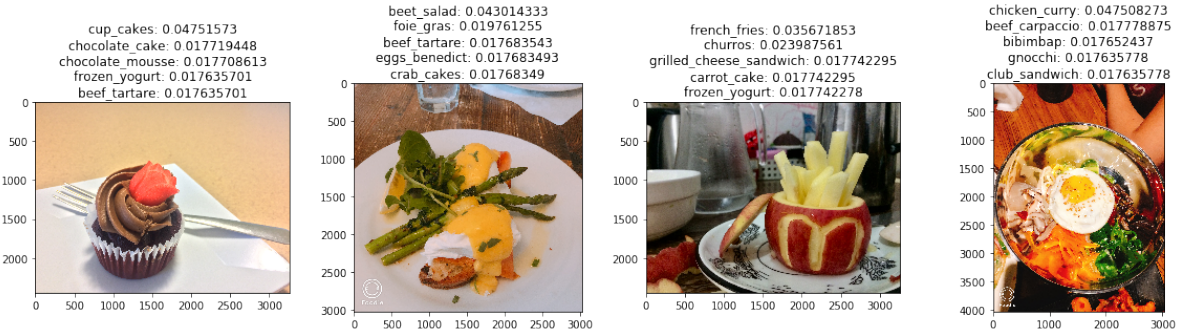

Figure 3. Success prediction sample

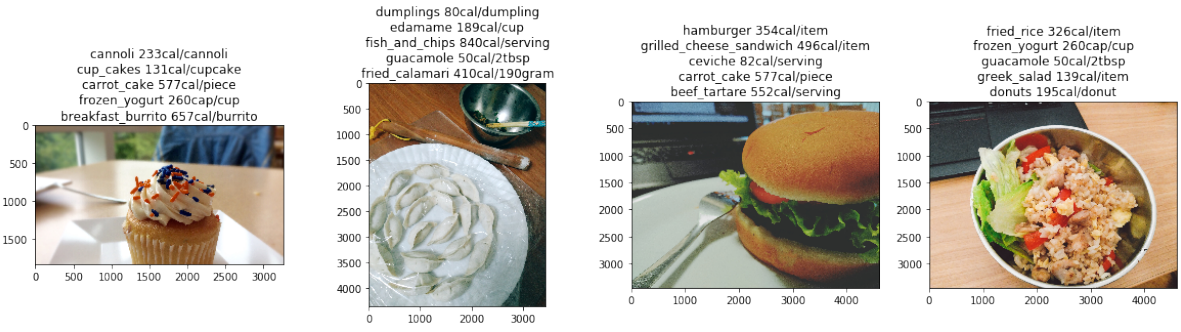

Figure 4. Sample calories output for cupcake, dumpling, hamburger, and fried rice.