Image Based Food Calories Estimation Using Various Models of Machine Learning

Haoyu Hu1

641709573@qq.com

Computer Science

Nanjing University of Posts and Telecommunications

Nanjing, 210023, China

Zihao Zhang2

Penn State University, State College, PA, 16803, US

Yulin Song3

College of Liberal Arts & Sciences

UIUC, Urbana, IL, 61801, US

##### 

##### *ABSTRACT:* For the last few decades, it has been the popular trend in China that people are putting more attention on improving their healthiness and regulating calorie intake for every meal, so that we build a model for calorie estimation of Chinese food. In an attempt to express our concerns on this issue, and with our great interests, we used object detection to estimate the calories count of some famous Chinese dishes as well as that of Western dishes. Based on the food images and previously defined calorie data, we built some image-based calorie estimation models, which we hoped can accurately identify the name of the Chinese and Western foods and provide their calorie intake and recipe, and finally offer meal plan advice for different groups of people. To identify the dish, we used the SSD (Single Shot MultiBox Detector) for real-time processing of object detection and classification. We also used a computer application called “labelImg” to manually label our dishes with their respective original dish names. Using our models, users can easily calculate the calorie intake of their desired foods by taking photos, saving a lot of time compared to their conventional methods.

***KEYWORDS: Object detection***

##### INTRODUCTION

This paper will propose a food recognition and calorie estimation. Firstly, the chosen dataset for algorithm implementation will be introduced, and we label the images using a github project called “labelImg”. Procedure of data clean will follow the analysis of the raw data set. In the second part, we use SSD object detection. In order to speed up the process, we used SSD to eliminate the need for a regional proposal network. SSD has applied a number of improvements, including multi-scale features and default boxes, to compensate for the loss of accuracy. With these improvements, SSD is able to match the accuracy of R-CNN which uses lower resolution images faster, which further improves the detection speed. Finally, suggestions for future work will be discussed.

##### DATASETS ACQUISITION AND HANDLING

The dataset we acquired containing Western foods, called Food-101, which was posed on Kaggle, was, according to the uploader K Scott Mader, “repackaged from the original source (gzip) available at <https://www.vision.ee.ethz.ch/datasets_extra/food-101/>”. We downloaded and used the repackaged version on Kaggle, and it contains 101 categories with a thousand images for each category [1]. We used the two categories that we believed to yield the most accurate results, namely “steak” and “clam\_chowder”.

In addition, we retrieved another dataset containing Chinese Foods, called VireoFood-172 dataset, published by “VIREO - Video Retrieval Group” on their website. The dataset contains 110,241 food images from 172 categories [2]. After getting the permission from the publisher to download their data, we followed their instructions and acquired the dataset. Again, we chose four categories that have their original respective file folder names as “1”, “5”, “59”, “61”.

In order to yield better results for estimation of the calories data of the respective food dishes, we manually labeled the images with an application called “labelImg”[3]. The specific procedures of labeling the images are as follows. First, we agreed on labeling each image in the six categories with six different names that are both simple and accurate enough. We used the name “steak” for “steak”, “soup” for “clam\_chowder”, “Braised Pork” for “1”, “Pork with salted vegetable” for “5”, “Beef noodles” for “59” and “Beef Kebab” for “61”. Then, we identified the min and max of horizontal and vertical values in the images that contain food; in other words, we labeled the parts of images with rectangles that do not contain anything not related to the food, and assigned the “new names” to food in each category. After that, for each category, we saved both the original image (.jpg) files and the resulting labels (.xml) files in the same folder to prepare for calories estimation.

##### METHOD AND MODEL

**CNN** **(Convolutional neural network)：**

Convolutional neural network includes three layers: input layer, hidden layer and output layer.

1. *Input layer*

The input layer of the convolutional neural network can handle multi-dimensional data. Since this project uses convolutional neural networks for computer vision applications, RGB channels as well as two-dimensional pixels on the image are taken as input data in this project.

1. *Hidden layer*

The convolutional neural network ‘s hidden layer consists of three parts: the convolutional layer, the pooling layer and the fully connected layer.

*1) The convolutional layer*

The convolutional layer containing multiple convolution kernels can be used to extract features from input data. Similar to feedforward neural network neurons, each element of convolution kernel corresponds to a weight coefficient and a bias. Each neuron connects to multiple neurons near the preceding layer. At work, the convolution kernel will scan the input features regularly, multiply as well as sum the input features by matrix elements and add the bias, in order to obtain reliable output.

*2) The pooling layer*

After the feature extraction of the convolutional layer is completed, the pooling layer will perform feature selection and information filtering of output feature mapping. The pooling layer contains the preset pooling function, which replaces the results of a single point in the feature map with the statistics of the feature map of its adjacent areas. At the same time of pooling layer selection, the feature map controlled by pool size, step size and filling is scanned by convolution kernel.

*3) The fully connected layer*

The position of the fully connected layer in the convolutional neural network is similar to that of the hidden layer in the traditional feedforward neural network. The last part of the hidden layer of a convolutional neural network only sends signals to the other fully connected layer. In the fully connected layer, the feature map expands the spatial topology into a vector and transfers the activation function.

*4) Output layer*

Generally, the fully connected layer is the upstream of the output layer of a convolutional neural network, so its structure and working principle are similar to the output layer of a traditional feedforward neural network. In this project, the object's central coordinates, size, and classification are output as a result of the output layer.

**SSD (Single Shot MultiBox Detector)：**

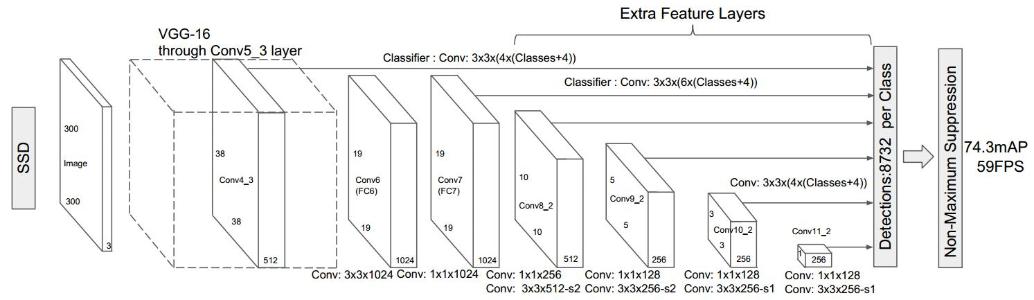


Figure 1: Structure of SSD

1. *The structure and characteristics of SSD*

SSD adds a series of convolutional layers on the base of VGG-16. The default box it generates is multi-scale, because the feature map of the default box generated by the SSD is both the last layer of the CNN output and the default box generated by the relatively shallow feature map. Since shallow features are more suitable for identifying small objects, the SSD algorithm has better detection effects for small targets than other algorithms. As shown in the figure, after the default boxes of each layer are assembled, there are a total of 8732 default boxes, that is, the SSD algorithm can generate 8732 detections per class. The non-maximum suppression algorithm ensures that the multi-scale default boxes generated by SSD have a higher probability of finding the candidate boxes closer to the ground reality.

However, the large number of default boxes will also reduce the training speed.

1. *The process of SSD*

First, enter a picture, let the picture go through a convolutional neural network (CNN) to extract features, and generate a feature map.

Then, extract the feature map of six layers, and generate the default box on each pixel of the feature map. While training, the Ground Truth box as well as the default Bounding box are paired as follows:

First of all, in order to make sure that every ground truth box corresponds to a unique default bounding box, we use each Ground Truth box to find the default bounding box which has the largest IoU.

The rest of the default bounding boxes that have not yet been paired will then attempt to pair with any Ground Truth box and match if their IoU is greater than the threshold value which is 0.5 for SSD.

Obviously, the default bounding box that is paired to the ground truth box is positive, and the default bounding box that is not paired to the ground truth box is negative.

Finally, collect all the generated default boxes and use the non-maximum suppression algorithm to output the default box with the best matching effect.

1. *The calculation of loss*

The SSD‘s objective function is composed of two different parts: One is to calculate the confidence loss of the corresponding default box and the target category. The other is to calculate the corresponding location loss.

It is defined as follows:

In the formula, the number of pre-selected boxes matched to the Ground Truth box is represented as N. Besides, the parameter used to adjust the ratio of confidence loss to position loss is represented as α, which defaults to 1. In addition, c is the confidence, l is the prediction box, and g is the ground truth box.

The confidence loss is defined as follows:

The location loss is defined as follows:

In the above formula, i represents the i-th default bounding box, j represents the j-th ground truth box, and p represents the p-th class. In addition, represents the i-th default bounding box matches the j-th ground truth box of class p. What’s more, g is the ground truth box, l is the predicted box, d is the default bounding box, w is the width, and h is the height.

##### EXPERIMENTAL PROCESS

We label a total of 5650 pictures of six types of Chinese food, and divide each type of image into a training set and test set according to the ratio of 7 to 3. Then we modify the corresponding configuration file of the official model to change the number of training steps, object category, training set as well as test set parameters, mapping relationship, and so on[4]. We use the official model provided by tensorflow as a pre-training model to continue training based on our own labeled data set[5].

##### DISCUSSION AND RESULT

For image recognition tasks with SSD object detection methods, we tried our model in 5000 to 50000 steps respectively. It runs 0.5 seconds for each step. Finally, loss stabilized between 1 and 2.

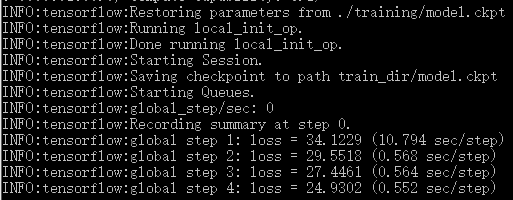


Figure 2: Effect when starting training

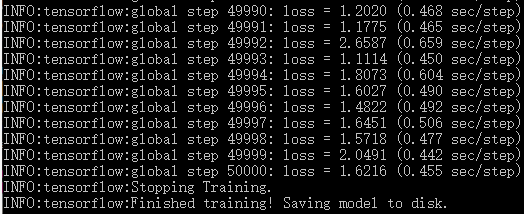


Figure 3: Effect at the end of training

The final accuracy results are as follow:

|  |  |
| --- | --- |
| 5000 | 56.10% |
| 10000 | 65.85% |
| 15000 | 75.61% |
| 20000 | 85.37% |
| 25000 | 68.29% |
| 30000 | 97.56% |
| 50000 | 95.12% |

For 15,000 steps, the steak and the Pork with salted vegetable will always be confused, and the recognition rate is low.

For 20,000 steps, there is no longer a problem of mixing steak and Pork with salted vegetable, and the recognition effect is very good.

For 25000 steps, the model labels different boxes on the same object, which is obviously a recognition error. This problem leads to a decrease in recognition accuracy.

For 50,000 steps, there was a problem of confusing Pork with salted vegetable with braised pork, but the identification frame was more precise.

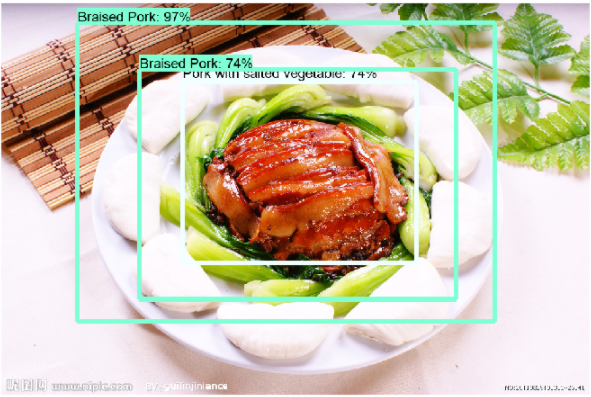


Figure 4: Recognition error in 25000 steps

It can be seen from the table that at first, as the number of steps increases, the recognition accuracy will gradually increase, but after a certain number of steps, the recognition accuracy will fluctuate due to the influence of various factors.

The final effect is as follows:



Figure 5: Final effect

##### 

##### CONCLUSION, LIMITATIONS AND PROSPECTS

Our training set has about 700 images for each category, a total of 3955 images, and the test set has a total of 1695 images. For object detection algorithms, our training set size is actually not large. But by using SSD, the marked objects in each picture can be used to the greatest extent, so as to achieve a good recognition effect. In addition, the accuracy of the model trained using the SSD algorithm is less affected by each image itself. When we train more than 20,000 steps, the recognition of similar foods will be more accurate. In other words, the data set we use may contain many noise factors, such as multiple objects in an image, large differences in resolution between images, and high similarities between different food categories. The model can still achieve better recognition results. Through the SSD algorithm, object detection can effectively solve many of the above problems.

By comparison, we find that the effect of the 30,000-step method is significantly better than the latter. Therefore, we choose 30,000 steps as our image recognition model. In the future, we will increase the types of identifiable food and make the model more powerful. In addition, based on this model, we will combine OpenCV to obtain the device camera, dynamically identify the type of food and give the corresponding calories.

##### REFERENCES

[1] Mader, K. S. (2018, May 15). *Food Images (Food-101)*. Kaggle. Retrieved September 30, 2020, from <https://www.kaggle.com/kmader/food41>

[2] *VireoFood-172 dataset*. (n.d.). VIREO-Video Retrieval Group. Retrieved October 6, 2020, from <http://vireo.cs.cityu.edu.hk/VireoFood172/>

[3] Enicck. LabelImg. https://github.com/tzutalin/labelImg

[4] Jonathan Huang & TF Object Detection Team, TensorFlow Object Detection API from https://github.com/tensorflow/models/tree/master/research/object\_detection

[5] Yiming Shi & vivek rathod, TensorFlow 1 Detection Model from https://github.com/tensorflow/models/blob/master/research/object\_detection/g3doc/tf1\_detection\_zoo.md