Group 16: [FoodPhone]

Lijun Chen, Mingyang Yan, Weipeng Wang,Yibin Zhang
{me4ever,myyan,weipeng,zhangyb}@bu.edu

Fig. 1: Example of images of four food categories: apple pie, baby back rib, waffle, and sushi

1. Project Task

The goal of our project is to recognize and classify the type of food given 101 food types by analyzing images of food/meals, like the ones shown in Fig.1. We need to develop a machine learning model to accurately label the major kind of food in a single image. The input of the model we used is an image shown in Fig.1 and the output is a number corresponding to its label.

2. Related Work

According to F. Zhu, A. Myers, Zhang, etc. [1] [2] [3], the food recognition problem usually starts with an image segmentation process, where from each segment low and high levels of the features indicating the type of food are extracted.

Although different projects take different datasets and variant image processing methods such as active contour, SIFT, salient/region based sampling are applied, CNN and SVM are the most commonly used machine learning models to classify the type of food.

3. Dataset and Metric

There are 75750 training images provided on Kaggle, where 80% of it will become the training set and 20% will be used as the validation set to optimize the performance of the training model. The other 25250 provided test images will be used as the test set. There are 101 food categories in total.

Training	Validation	Test	Total
60600	15150	25250	101000

Table 1. Dataset summary

The classification accuracy will be using the simplest metric by dividing the number of correct predictions by the total number of predictions over the test set:

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of prediction}}$$

we hope to show that our method can achieve accuracy higher than the 80% achieved by the baseline method.

4. Approach

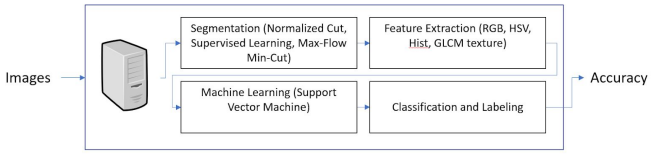

Fig. 2: Project Overall System Architecture

As is shown in Figure 2, our team first pre-processed the images. To do this we segmented the images using SLIC (Simple Linear Iterative Clustering) library to get superpixels and used normalized cut to smoothen the boundaries in the image and merged those smaller segments with similar color.

Then, we manually labeled 200 segmented images, which is a small subset of our training images and contains food and non-food area, to train a simple supervised model to separate food and non-food area. We used a calibrated support vector machine based on average rgb color, average hsv color, 3-bin histogram and relative position to the center of the image of regions. A simplified min-cut max-flow method is used to cut through the largest probability gap between regions thus the image is separated into food and non-food area. Examples shown in Fig.5-8.

After we obtained the segmented imagines, that supposedly only contain area of food, we extracted information like RGB value, HSV value, Histogram, and GLCM texture from them by using skimage feature library.

We used joblib from sklearn externals library to load and store the data.

Our machine learning model is based on SVM. We used LinearSVC library, that applies the one-vs-rest scheme for multiple classes and has a squared hinge loss function:

$$L(f(x), y) = \sum_i max(0, 1 - y_i * h(x_i))^2$$

where y is the dataset and h(x) is the hypothesis function.

Pre-processing Result -- Image Segmentation

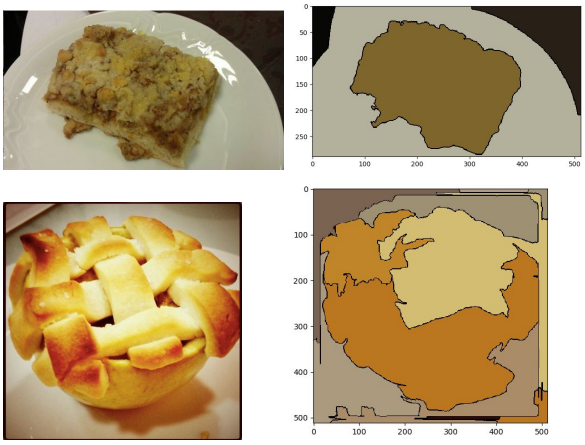

Fig. 3: images and segmentation results of two kinds of pie

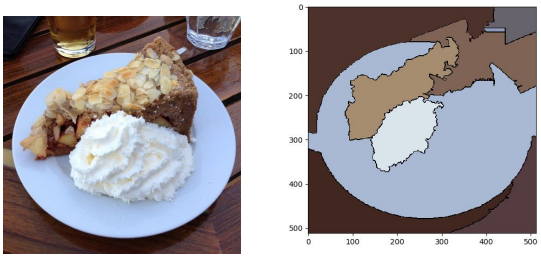

Fig. 4: image and segmentation result of a cake and ice cream

The result image is average rgb color, as shown in the right of Fig. 3. Although the segmentation in Fig. 4 does not cover all of the ice cream, it still includes the texture and color feature of the ice cream.

Pre-training Result -- Supervised Learning for Food Area Identification and Better Feature Extraction for Training

Unfortunately, the entire process does take quite a lot of processing power (mainly brought by the SLIC and greedy normalized cut); however, with only 200 training images, 400 samples, this does not look bad at all and it can certainly be improved greatly by feeding more training data. From this point, we can get the features of a certain kind of food more precisely. As a preparation for the final training on the training set, we have 3 cues of mean rgb color, 3 cues of mean hsv color, 3-bin histogram, 5-bin histogram, 8 cues of GLCM dissimilarity in 4 directions, 2 different distances and 8 cues of GLCM correlation in 4 directions, 2 different distances.

Fig. 5-8: image segmentation after supervised learning

5. Final Result -- Training in Support Vector Machine

The first experiment was using the feature hog to represent the texture. It is fed together with the color features and histogram. It gave out a few thousands dimensional feature vector which was not very compatible with the other features plus the information of the shape of the food in a image is highly likely to had already been lost in the segmentation step.

In the second experiment, the texture features extracted from the grey-level co-occurrence matrix, the dissimilarity and correlation properties in 4 directions and 2 different distances were used to replace the hog descriptor. 9 patches evenly located in the segmented food region were selected to represent the texture of this certain type of food. These features were fed into a SVC model, which does not scale well enough to the training set with over 70000 samples.

Our last experiment before submitting the report, with the feature space scaling issue fixed, applied the svm

model called LinearSVC. This model is fairly fast under a one-vs-rest schema but has a lot of mis-classification issues.

For the food object detection, there are definitely some better and more efficient methods out there that can keep features of the food like its shape and filter the noise of the image at the same time. The segmentation step does not necessarily improve the performance of the trained model under the same training model. For the training model itself, one-vs-one schema may be tried in the future by downsizing the number of samples first aiming to better separate the training samples.

6. Conclusion

In this project, the team explores the possibility of applying image segmentation method to improve the result of the training result of food type identification. A supervised training method using SGDC Classifier as is mentioned in Section 4 was used to help us decide the most possible border between food and no-food parts in an image. Total of 30 features including colors and texture were fed into a SVM to get our final model.

Although the expected result was not reached in the certain period of time and there exists method using CNN which gives about 85% or higher accuracy in food classification, with better features and supervised training model, our accuracy can be improved to a similar level.

7. Timeline and Roles

Task	Deadline	Lead
Image Pre-Processing: Segmentation	11/13/18	Weipeng Wang
Image Pre-Processing: Feature Extraction	11/19/18	Mingyang Yan
Design and initially train the machine learning model	11/19/18	Yibin Zhang
Continuously train the model to get a successful result	12/11/18	Lijun Chen
Run tests	12/11/18	Yibin Zhang
Prepare report and poster	12/11/18	all

References

[1] F. Zhu *et al*., "The Use of Mobile Devices in Aiding Dietary Assessment and Evaluation," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 4, pp. 756-766, Aug. 2010.

[2] A. Myers *et al*., "Im2Calories: Towards an Automated Mobile Vision Food Diary," *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, 2015, pp. 1233-1241.

[3] Zhang, W., Yu, Q., Siddiquie, B., Divakaran, A., & Sawhney, H. (2015). “Snap-n-Eat”: Food Recognition and Nutrition Estimation on a Smartphone. *Journal of Diabetes Science and Technology, 9*(3), 525-533.

[4] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274-2282, Nov. 2012.

[5] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” Int. J. Comput. Vis., vol. 1, pp. 321–331, 1998.