Report for Food Recognition

HU ZUYING, Hong Kong University of Science and Technology, HK
YUAN DEMING, Hong Kong University of Science and Technology, HK
ZHANG HAOZE, Hong Kong University of Science and Technology, HK
ZHANG TIANXIANG, Hong Kong University of Science and Technology, HK

Additional Key Words and Phrases: Food Recognition, Image Classification, Text Classification, CNN, SVM

ACM Reference Format:
HU Zuying, YUAN Deming, ZHANG Haoze, and ZHANG Tianxiang. 2017. Report for Food Recognition.

1 INTRODUCTION

Given an image of food or some descriptive context of food, a person could easily figure out what kind of food it is and speak out its name. To recognize food by the given image or context, machines lack the abilities to do the same things as it is impossible for machines to taste food and accumulate relative experiences. However, based on the mechanisms of machine learning food recognition could be done by utilizing text classification and image recognition. On the other hand, Food recognition enjoyed massive applications including food recommendation, nutrition management etc. Therefore, the accurate recognition of food is worth academic and commercial exploring. In this experiment, we focus on food recognition based on texts and images respectively. After that we tried to combine results produced by text classification and image classification to get an ensemble result for one food based on its recipe and image.

2 DATASET, SCHEMA AND METRICS

To explore an accurate algorithm of food recognition, we utilize dataset UPMC Food-101 with about 100,000 items, consisting of images and HTML webpages, of food recipes classified in 101 categories. Referring to the descriptive file of UPMC Food-101, there are 86,574 images with corresponding text with other items either being image or text[7].

Given the characteristics of UPMC food-101, food recognition task could be subdivided into two parts: image recognition by image input and text classification by text input. For an input item, the recognition model should predict its category. And top 1 accuracy and top 5 accuracy are adopted to measure the performance of the model.

3 RELATED WORKS

The food recognition task is subdivided into two subtasks including image recognition and text classification, therefore, we explore related works on these two subtasks.

Authors' addresses: HU Zuying, Hong Kong University of Science and Technology, HK, collin.hu@connect.ust.hk; YUAN Deming, Hong Kong University of Science and Technology, HK, demingyuan3094@gmail.com; ZHANG Haoze, Hong Kong University of Science and Technology, HK, hzhangbl@connect.ust.hk; ZHANG Tianxiang, Hong Kong University of Science and Technology, HK, 853122604@qq.com.

XXXX-XXXX/2017/12-ART $15.00
https://doi.org/xxx.xxx

3.1 Image Classification

Image classification is to classify an input image by its content, which is one the main tasks in computer vision. Recently computer vision has witnessed dramatic progresses with widely applying Convolutional Neural Networks (CNN) in computer vision tasks.

In ImageNet Large Scale Visual Recognition Competitions (ILSVRC), the winning teams have proposed several states-of-the-art methods in image recognition tasks. To some extend some methods even outperform the human beings. Some widely used models proposed by winning teams includes VGG Net [3] and ResNet [2]. These models show excellent performance in ILSVRC. In 2015 ILSVRC, ResNet has achieved 3.57% of top-5 error over ImageNet dataset. However, these CNN models always adopt very deep layer structure and a large scale of parameters to do image classification leading to high requirements on computing devices and long-term training period. These CNN models perform well in ILSVRC on the condition of massive training dataset being available, which may not be satisfied in some tasks. CNN models perform far better than any other traditional methods despite these drawbacks. To overcome these limitations on CNN, researchers have come up several methods such as data augmentation and transfer learning. Based on CNN structure some more sophisticated models, such as Fast R-CNN Ross [6], have been proposed by researchers to do more challenging tasks in computer vision.

3.2 Text Classification

Text classification is to classify text documents given to their content. The researchers in this filed have proposed several methods including Naive Bayes, SVM, Neural Networks, etc. to accomplish this task.

3.2.1 Naive Bayes Method. Naive Bayes classifiers perform well in text classification. Differing in the handling of world occurrences in the document, two main approaches, Multi-variate Bernoulli model and multinomial model, are developed during past several decades [1]. According to Andrew McCallum and Kamal NigamâĂŹs experiments, the performance of Naive varies given different sizes of vocabulary and different sources of vocabularies. Though Naive Bayes performs surprisingly well in text classification task, an intrinsic flaw resides in this method as this approach is based on the assumption that attributes of text are independent. This method ignores the information of the relevance between words and sentence structures, which may lead to a bottleneck in this method though the limitation of Naive Bayes in text classification requires further researches.

3.2.2 Support Vector Machine. Support Vector Machine(SVM) is another method that is widely applied in text classification. Larry M. Manevitz and Malik Yousef have shown in their research that one-class SVM is superior to many other methods including one-class versions of the algorithms prototype (Rocchio), nearest neighbor, naive Bayes except of the neural network one [4]. As pointed by Larry M. Manevitz and Malik Yousef, SVM approach is sensitive to the choices of word representation and kernel function. The careful decisions on these choices determining significantly on the performance of the SVM make SVM not a good method in text classification for robust performance.

3.2.3 Neural Networks. Neural Network gives a superior and robust performance over other methods [5]. By projecting the words in the document into a m-dimension vector, the text classification could be done by Neural Network.

4 EXPERIMENTAL PROCEDURES

In this section, we will give the details of the methods we used in classifying food based on texts and images and our exploration on combining the above two results by an ensemble model.

4.1 Text classification

4.1.1 Data cleaning and stemming. In dataset UPMC-101, we first convert HTLM file into text file which contain many noises such as meaningless marks, punctuations. To remove these meaningless marks, data cleaning is needed before further processing the data. And for one word there may be many forms, for example, word ŕeadḿay appear as ŕeading; ŕeads: The computer may treat these different forms of one word as different words even though they refer to the same meaning, which may increase the dimension of features of each instance dramatically. To eliminate the effects of various forms of one word, stemming could be applied on the text file to reduce dimension of the features. After stemming, various forms of one word will be converted to the same form, therefore, ŕeads; ŕeadingánd ŕeadáll will be converted to be ŕead; which could be treated as the same word by computer.

4.1.2 Data cleaning and stemming. We mainly utilized two techniques in our experiments which are count vector and word vector to present text data. To present text file in form of count vector, we first build up a dictionary to collect all the words appearing the text files with several rules: 1. all words will be converted into stemmed form; 2. the words appearing in less than 11 documents will not be included in the dictionary; 3. stop words defined in snowball stemmer such as t́his; t́hatánd etc will not be included in the dictionary. Then each word in the dictionary will be treated as one feature of the text file. For a text file, the value of one feature will be the occurrence of the corresponding word in the text file. The count vector presenting method will encounter one problem that the words with frequent occurrences in whole dataset may dominate the results of classification. Therefore, when counting the frequency of word in one text file, term frequency inverse document frequency(TF-IDF) technique are used to solve this problem which gives more favors to word less occurred in whole dataset but more occurred in particular documents. We called this process as text file count-vectorization.

To present text file in form of word vector, similarly we first build up a dictionary which contain all the words of whole dataset, then we used word embedding method which represents each word with a unique 200-dimension vector. Meanwhile, we maintain the importance of each word by multiplying each word vector with it corresponding TF-IDF score got from count-vectorization. For a text file which contains m different words with discarding words not in both dictionaries, we will get a $m \times 200$ matrix, then we do an average on each dimension. After this process, we could use a 200-d vector to represent a text file.

4.1.3 Classification. In text classification tasks, we utilized classification methods: Multinomial Naive Bayes(Multinomial NB), SVM and Neural Networks (NN) provided by python package sklearn and pytorch.

Naive Bayes: We apply Multinomial NB on the count vector of the text file. In our experiment, we set the model λ with as 0.01 and make the fit-priori as False.

SVM: For text file word vector representing, we first used SVM with linear kernel to do classification.We standardized the input 200-d vector then applied SVM classifier with linear kernel on the data.

NN: We also applied Neural Network classifier on text file word vector representing. Similarly we first standardized the input data.

For classifier,firstly, we use PyTorch to build our NN classifier. According to text file word vector representing and the number of food category, we set our data's input dimension as 200, and output dimension as 101 with a hidden layer of 500 nodes. We use Mean Squared Error as our loss function and set learning rate = 1e-3 at first. Then we select Adam schema as our optimizer with halving learning rate every 150 iterations.

The results of the three classification methods are shown in table 1. SVM and NN got close result as both used linear mechanism to do classification while Naive Bayes gave a relatively poor accuracy.

Model	Accuracy
Multinomial NB	68.3%
SVM	77.6%
NN	78.6%

Table 1. Accuracies of Three Text Classifications

4.2 Image Classification Module

In this step, we try to use an image classification model using transfer learning to product a prediction according to the photo in the dataset.

To achieve this goal, we shall pre-process the photos to fit the request of transfer learning. First of all, we have to do some preprocessing. The photos given are not in the same size, resolution, and aspect ratio. First we crop at the center of the image until it gets a specific size. Then we will transform into a tensor and finally normally it. And we separate the image into three parts: train, validation and test with the ratio of 7:1.5:1.5.

We select transfer learning as the model in our project. It has two main steps.We use convolutional network to extract the feature.Comparing with fully-connected Neural Network, the input is connected to every hidden neuron. In CNN, however, neurons in the first hidden layer will only be connected to small region of inputs. The values of the first layer will be the results of a convolution between the input layer and filters.This method ensures that it is good at extracting image features which always exists in some pixels close to each other. So, we can use model trained by others to extracted and remove the final fully connected layer to fit in our own dataset.

After that, we get a classifier for the food recognition. But we still need to further improve its accuracy. We need to fine-tune the weights of the model using back-propagation. Considering the factor of overfitting, weâĂŹd better freeze the earlier layers which are usually responsible of recognize edges, shapes and colors. The later layers, respectively, will concentrate on details of the image which is not so suitable when the dataset is not as same as the one it trained with.

To build the training environment, we need high performance GPU to accelerate the speed of training, we use Amazon AWS. Comparing with CPU, GPU is good at parallel computing due to its structure, which has lots of cores which can only do very simple computation while CPU can deal with complex commands but with lower amounts of core. In a much smaller size of transfer learning, the time with only CPU is thousands times longer than the one with GPU accelerated. In fact, considering that it actually takes 12 hours on the cloud platform with Nvidia Tesla k80, we donâĂŹt think that it is possible to end the training using CPU on personal computer. And to save time, we also use pre-trained model to save time and recognized some common patterns or features in the images. There are many models we can select. We try to test them and find a best among them. The Top-1 error, top-5 error is shown in the Table 2.

Model	Top-1	Top-5
Resnet-18	48.2%	73.4%
Resnet-34	49.2%	73.3%
Resnet-152	64.9%	83.2%
vgg-16	64.4%	82.4%

Table 2. Top-1 and Top-5 Errors of the Four CNN Models

And we selected Resnet-152 according to the result, The result is not as good as text classification. There exist several reasons for the result. First, texts are usually more detailed than the image. For example, on a website introducing a recipe, the texts may refer to the name of the food recipe for times, but the images may not be all related to the recipe. Also, the images may not be clear enough to distinguish from each other even with eyes of human beings. And some food may share same features, which confuse the network,

4.3 Ensemble Module

In this step, we tried to use an ensemble model to produce a more accurate prediction by combining results got by text classification and image classification. According to the accuracies of all the method, we used top 5 predictions of Neural Networks Classification for each text file, and top 5 predictions of Resnet-152 for each corresponding image.

It is assumed that if two sets of predictions vote on the same prediction, this prediction is more likely to be the right prediction. To combine the two set of results, we assign each prediction a weight as follow. As the accuracy of text classification is higher than image classification, we assign prediction of text classification a little more weight.

For the top 5 predictions of text classification result for each text file, if the prediction ranks at j-th position, its weight is $W_j = 5 - j + \frac{1}{j+1}$ .For the top 5 predictions of image classification result for each text file, if the prediction ranks at j-th position, its weight is $W_j = 5 - j$.

After the assignment, we combine the two sets of predictions together with adding the weights whose corresponding prediction are same. Then we choose the prediction with largest weight as the final prediction. The result of the ensemble model is shown as follow. According to the table 3 above, we could improve the accuracy slightly even though the improvement is not significantly. It is worth more exploration on this ensemble model to combine the results of image classification and text classification in a better way.

Model	Accuracy
NN	78.6%
Resnet-152	64.9%
Ensemble-Model	80.2%

Table 3. Accuracies based on Text Classification, Image Classification and Ensemble Model

5 CONCLUSION

In this experiment, we train text-image classification model from the dataset UPMC Food-101, which contains 101 food categories. Presenting the text data by TF-IDF, we train text classification models by Multinomial NB, SVM and NN. For image classification, Resnet18, Resnet34, Resnet152 and VGG 16 are utilized for transfer learning. Based on the best performance models, NN and Resnet152, of these two parts, the ensemble model obtains a better accuracy, 80.2%.

REFERENCES

[1] McCallum Andrew, Nigam Kamal, et al. 1998. A comparison of event models for naive bayes text classification. *AAAI-98 workshop on learning for text categorization* (1998).

[2] He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian. 2016. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016).

[3] Simonyan Karen and Zisserman Andrew. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (Sept. 2014).

[4] Manevitz Larry M and Yousef Malik. 2001. One-class SVMs for document classification. *Journal of Machine Learning Research* 2 (Dec. 2001).

[5] Manevitz Larry M and Yousef Malik. 2007. One-class document classification via neural networks. *Neurocomputing* 70 (2007).

[6] Girshick Ross. 2015. Fast R-CNN. *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015).

[7] Wang Xin, Kumar Devinder, Thome Nicolas, Cord Matthieu, and Frédéric Precioso. 2015. RECIPE RECOGNITION WITH LARGE MULTIMODAL FOOD DATASET. *Multimedia & Expo Workshops (ICMEW), 2015 IEEE International Conference* (2015).