Neural Network for Food Image Classification and Recipe Retrieval with automatic un-supervised labeling

Yuwei Chen
College of Information and Computer Science
140 Governors Dr, Amherst, MA 01002
yuweichen@umass.edu

Chuanpu Luo
College of Information and Computer Science
140 Governors Dr, Amherst, MA 01002
chuanpuluo@umass.edu

Abstract

Food image recognition is playing an important part in enhancing Internet users online searching experience. It is more intuitive to search by images rather than long sentences of description. However, due to the nature of food images, their recognition is a particularly challenging task, which is why traditional machine learning approaches have achieved a low classification accuracy. In this project, we fine-tuned two CNNs initially trained on a large natural image recognition dataset(Imagenet ILSVRC) and transferred the learned feature representations to the recipes crawled from allrecipes.com. The dataset consisting of 41240 images and recipes of 100 categories were clustered and labeled by Non-negative Matrix Factorization(NMF) and TF-IDF topic model. Data augmentation was applied to increase the size of the data. In this project, we first conduct horizontal comparison on pre-trained ResNet-18 and pre-trained VGG-16 models. We found that pre-trained VGG-16 out-performed ResNet in predictive percision. Then we conduct vertical comparison by modifying the last three layers respectively. Lastly, based on our best classification model, we obtain 'feature vector' for every image in our dataset. By matching 'feature vector' of every image, we are able to retrieve several candidate recipes.

1. Introduction

1.1. Motivations

In current digital era, people like sharing pictures of food on the social media [5]. "How to cook that" is one of the ten most popular "how to" questions on YouTube, followed by "how to draw" and "how to kiss" [1]. Whereas, when people come across an attracting food image, since the food image may not always be attached with dish names and ingredients, not to mention, instruction. Therefore, it would be hard for people to retrieve recipes and cook what they want. In such circumstance, an automatic food image recognition and recipe retrieval system is favored by all food lovers.

1.2. Challenges

There are challenges of building this system. One challenge is that food image is not deform-able and there are various forms of a certain kind of food, let's say, potato can be made as smashed potato, French fries and baked potato. These dishes may not be in the same recipes categories, they sometimes look very similar only in terms of image information, which would make it difficult for previous supervised classification. Another challenge lies in the training of neural network. Since training an entire Covolutional Neural Network from scratch with random initialization needs an exceptional large dataset and requires high computational strength[3]. Generally, we can use two pre-trained CNN models with different structures and have been trained on a large dataset such as ImageNet, remove the last fully connected layer, freeze the parameters from the rest of the CNN as a fixed feature extractor for the new dataset. Similarly, it is possible to fine-tune any layers in pre-trained CNN by fixing some of the earlier layers unchanged and adjust some later layers of the network.

1.3. Structures

Our project deals with the problem of automated recognition of a photographed cooking dish and its appropriate recipe by applying fine-tuned pre-trained CNN aided with data augmentation to 100 categories food image data crawled from Allrecipes.com.

The remainder of this paper is organized as follows. Section 2 is a short review of related work. Section 3 explains details of the methodology we applied from data pre-processing, food image classification to recipe retrieval. Section 4 presents the dataset we used, the preprocessing performed, and fine-tuning performance with discussion. Section 5 shares our conclusions based on the result we obtained and future works we may conduct.

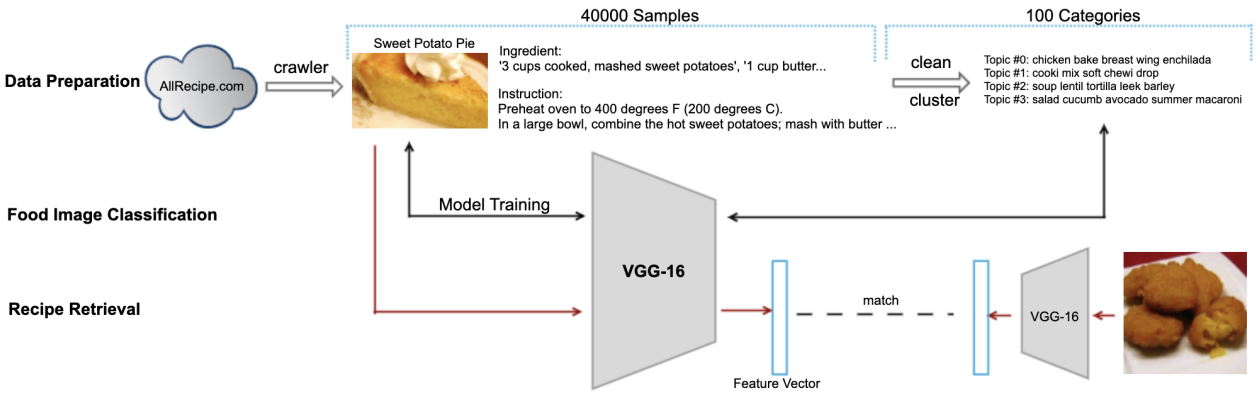


Figure 1. Structure of Model

2. Related Work

To recognize food image, previous method [11] try to make statistics local features in food images. Since food items are deform-able objects with various appearance, traditional image features such as SIFT and HOG doesn't work well. With the emergence of deep learning, more advanced neural network frameworks such as VGG-16 [9] and ResNet [2] have been used in image classification. Since 2015, Deep Neural Network has been applied in food image classification [10]. The author Yanai tried to fine-tune model pre-trained from ImageNeet. They achieved around 70% accuracy on UEC-FOOD100 and UEC-FOOD256 dataset.

After that, Liu et all. [4] designed and implemented a real-time food-recognization system, which automatically preprocess and segments images and then classifies images with neural network similar to GoogleNet. They achieved 77% Top1 accuracy. Pan et al. [6] designed multi-class classification of food ingredients. The author designed a multi-ingredient classification algorithm by integrating ResNet deep feature sets, Information Gain (IG) feature selection, and the SMO classifier. Based on a small multi-class dataset that includes 41 classes of food ingredients and 100 images for each class. The author achieves 87% accuracy.

The above works done by other researchers are really impressive and give us ideas about what to do in this project. However, something can still be improved from the aspect of dataset. First, all of the images are labeled by human beings, which are accurate but expensive. Second, categories in these dataset are too general, for example, 'rice' and 'beef curry'. Third, there are no recipes in most of these datasets. These reasons encourage us to build our own dataset and make categories by un-supervised clustering methods.

There are two main ways to obtain recipe from food image: retrieve recipe from data-set or generate new recipe. In a this paper [8], the author trains a neural network to find a joint embedding of recipes and images that yields impressive results on an image-recipe retrieval task (retrieval approach). From the latest Facebook research [7], the author recreates cooking recipes given food images, by combining CNN feature extraction and Natural Language Generation Model (recreation approach). Based on these two papers, the recreation approach is able to generate new recipes that have never been tried before. However, the recreation approach can not guarantee the correctness of each recipe it generated. For example, numerical numbers such as cooking time and quantity of ingredients in each recipe may be wrong, while these numbers are critical in cooking process.

3. Approach

The complete structure of our model is shown in Figure1. There are three main challenges we have to deal with in this project: a) Scrape food images and recipes from Allrecipe.com, clean and categorize them properly in 100 classes. b) Transfer, train and fine-tune pre-trained neural network model such as VGG-16 and ResNet-18 on food image classification task. c) Use trained neural network to extract 'feature vector' from food image, match 'feature vector' between food images, and retrieval recipes for input food image. In the following part of this section, I will describe each part in detail.

Data preparation: First, we crawl recipes displayed on top 3000 pages from Allrecipes.com (we modified web-crawler tool originated by developer Ryan [1] and run it on Google Virtual Machine). Each recipe includes title, ingredients, instructions and cooked food image. In order

[1] https://github.com/rtlee9/recipe-summarization

to classify all recipes into 100 food categories, we extract key ingredients from all recipe titles (excluding stop words and ingredient-unrelated words such as 'best', 'amazing'), use term frequency–inverse document frequency(TF-IDF) to calculate how close the relationship between each ingredient and each recipe. Then we use non-negative matrix factorization (NMF) to simultaneously do dimension reduction and clustering. In this way, we cluster ingredients and calculate the probability of possible category for each recipe.

Food image classification: In order to speed up the process of model training, we decide to transfer pre-trained model such as VGG-16 and ResNet-18 on our food image classification task. We revise its last fully-connected layer from 1000 output dimension (classes of ImageNet) into 100 output dimension (100 food categories). In our first modified model, we freeze all convolutional layers, keep only the last fully-connected layer un-freezed. And later, we try to un-freeze last two or three fully-connected layer, and compare their performance. Also, we compare the performance of VGG-16 and ResNet-18. We use negative log likelihood loss as our loss function and Adam as our optimization algorithm.

Recipe Retrieval: In order to retrieve recipe for an input food image, we extract feature vector from food image by our trained VGG-16. After that, we will match feature vector between input food images and all other images in our training set (using KNN). Finally, we will retrieve top-5 most likely recipes for user.

4. Experiment

In 4.1, we are going to introduce what kind of dataset we are going to use in experiment. In 4.2, details about how we train neural network model for food image classification will be displayed. In 4.3, we will explain how to retrieve recipe given an input food image. At last, we discuss some reasons in generating these results.

4.1. Dataset

The datset includes 41240 recipes with each containing images, ingredients and instructions. Figure 2 is an overview of a single recipe, we extract the photographed cooking dishes on top right, the ingredients in the middle, and the instructions in the bottom.

Figure 3 shows some categories after we cluster these 41240 recipes into 100 classes. In order to train our food image classification model, we divided our dataset into training set, testing set and validation set with 80%, 10%, and 10% ratio of the original dataset. There are 32992 images in training set, and 4124 images in testing set and validation set respectively.

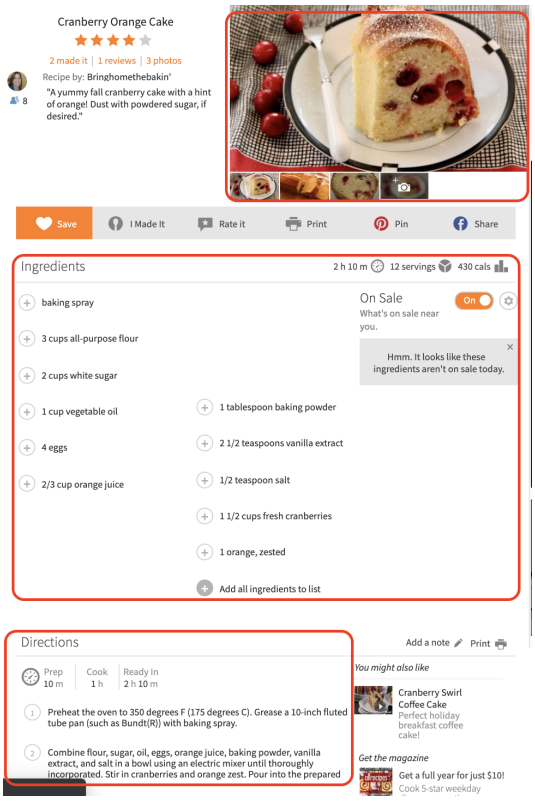


Figure 2. sample page of a recipe

```
Topic #0:
chicken bake breast wing enchilada
Topic #1:
cooki mix soft chewi drop
Topic #2:
soup lentil tortilla leek barley
Topic #3:
salad cucumb avocado summer macaroni
Topic #4:
pie shepherd crust rhubarb custard
Topic #5:
cake pound coffe upsid bundt
Topic #6:
potato bake mash scallop twice
Topic #7:
chees bake mac macaroni blue
```

Figure 3. Sample of 100 Categories

4.2. Training the Models

Before feeding images into our model, we apply image pre-processing and normalization on each batch. We resized them into 224 x 224 pixels and normalized each color channel by subtracting a mean value and dividing by

a standard deviation. Because we only extracted surface image for each recipe, and not all categories are sharing decent amount of images. So, we need to increase the amount of images in some categories. In this project, we use image augmentation to artificially increase the number of images in training set by resizing, cropping, and flipping horizontally. A different random transformation is applied each epoch (while training), so the network effectively sees many different versions of the same image. All of the data is also converted to Torch Tensors before normalization. The validation and testing data is not augmented but is only resized and normalized. The normalization values are standardized for Imagenet so that our new data can be transfered to the pre-trained model well.

Parameters of Training Model	
Model	VGG-16 v.s. ResNet-18
Freeze FC-Layer	1/2/3 v.s. 1
Optimization Algorithm	Adam
Loss Function	Negative Log Likelihood
Accuracy	Top5
Learning Rate	1e-3
Epochs	14
Batch	100
Beta1	0.9
Beta2	0.999

Table 1. Parameters of Training Model

In table 1, we listed all parameters we used to train VGG-16 and ResNet-18 in detail. For VGG-16, we revise it's last fully-connected layer from 1000 output dimension (classes of ImageNet) into 100 output dimension (100 food categories). In order to achieve higher accuracy,we try to unfreeze its last one fully-connected layer, its last two fully-connected layers and its last three fully-connected layers. We use Adam as our optimization algorithm, the reason it that Adam is able to decrease it's learning rate after each epoch, which is more stable and more likely to converge than SGD. We set the learning rate as 1e-3, under which learning rate the model can converge smoothly. We use negative log likelihood as our loss function, because we wish our model is able to distinguish the right category as much as possible. One thing has to be mention is that, instead of using Top1 accuracy, we use Top5 accuracy instead. The reason is that our final purpose is doing recipe retrieval, we train this classification model is for extracting 'feature vector' from food images. Also, in practical, we just need the correct recipe appear in the first returned page, no need to be the first one.

4.3. Recipe Retrieval

Below shows one example of our recipe retrieval model, Figure 4 shows the given image, Figure 5,6,7 shows three retrieved recipes from our model. Below figure 5,6,7 are those retrieved ingredient and instrctions.

Figure 4. Input Image of Recipe Retrieval

Figure 5. Retrieved Recipe 1

Figure 5 Title: Garlic Chicken with Orzo Noodles

Ingredients: *['1 cup uncooked orzo pasta', '2 tablespoons olive oil', '2 cloves garlic', '1/4 teaspoon crushed red pepper', '2 skinless, boneless chicken breast halves - cut into bite-size pieces', 'salt to taste', '1 tablespoon chopped fresh parsley', '2 cups fresh spinach leaves', 'grated Parmesan cheese for topping']*

Instructions: *Bring a large pot of lightly salted water to a boil. Add orzo pasta, cook for 8 to 10 minutes, until al dente, and drain. Heat the oil in a skillet over medium-high heat, and cook the garlic and red pepper 1 minute, until garlic is golden brown. Stir in chicken, season with salt, and cook 2 to 5 minutes, until lightly browned and juices run clear. Reduce heat to medium, and mix in the parsley*

and cooked orzo. Place spinach in the skillet. Continue cooking 5 minutes, stirring occasionally, until spinach is wilted. Serve topped with Parmesan cheese.

Figure 6. Retrieved Recipe 3

Figure 6 Title: Migas

Ingredients: *['6 eggs', '1/4 cup milk', '3 (6 inch) corn tortillas, coarsely chopped', '1 tablespoon butter', '1 tablespoon chopped green onion', '1/2 cup diced cooked ham', 'salt and pepper to taste']*

Instructions: *In a bowl, beat eggs and milk together. Soak tortilla pieces in egg mixture for 5 to 10 minutes. Meanwhile, heat butter in a skillet over medium heat. Saute green onion and ham until onion is soft, about 5 minutes. Pour egg mixture into skillet and cook over medium/low heat until eggs are set.*

4.4. Result and Discussion

Figure 7 includes the training and validation performance regarding the first epoch. The plot on the left shows that the VGG-16 model with the last fully-connected layer un-freezed achieves both the highest training accuracy(37.73%) and validation accuracy(43.62%) at the first epoch. And the lowest falls on the Overall, VGG-16 models start out better than ResNet-18 model at the beginning stage. The plot on the right shows that all training loss are slightly lower than validation loss. It implies that the Dropout layers take effect to avoid over-fitting the training images. On the other hand, the differences of these two losses are not very large that causes under-fitting.

Figure 8 shows the trend of all four models when data being trained for more epochs. VGG-16 model with the last fully-connected layer un-freezed stays relatively high compared to the other three models. Although VGG-16 model with the last two fully-connected layers un-freezed has a lower accuracy than VGG-16 with last layers un-freezed, it gradually increases after 12 epochs, and later transcent to

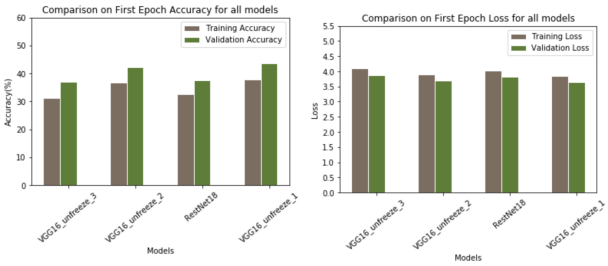


Figure 7. Comparison Among Four Models on First Epoch

be the highest accuracy among all models in epoch 14. In addition, We can see there is a large jump from a low accuracy start out for ResNet-18. The performance of VGG-16 model with the last three fully-connected layer un-freezed. Figure 8 shows one of the wrong classified food images.

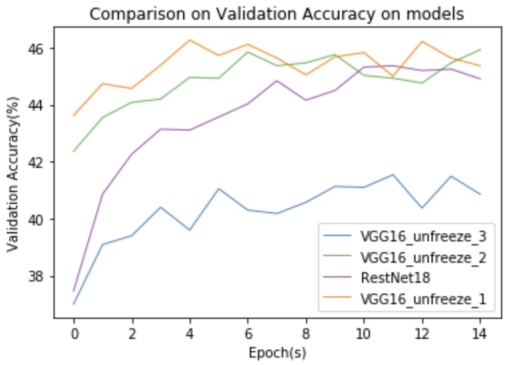


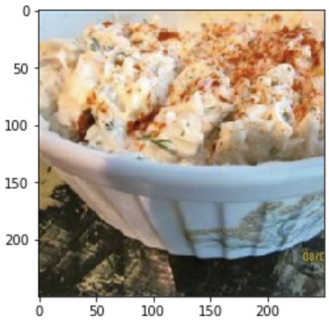
Figure 8. Comparison of Validation Accuracy Among Four Models for 15 Epochs

The correct category: 'salad cucumber avocado' is not appeared in the Top5 prediction and we are wondering how this happens. From Figure 9, we can observe that there exists overlapping ingredients between categories, for example, 'bake' appears both in Top 2 and Top 3 prediction. The reason is that TF-IDF and NMF calculates the possibilities between each ingredient and each category, which means they allow one ingredient exists in two different categories. Thus, the test image 'crab salad' is close to 'cupcak' in Top1 prediction, close to 'bake' in Top2 and Top3 prediction.

5. Conclusion

In this project, our main purpose is building a system which enables food lovers to retrieve recipe by food images. In order to do so, we first scrawl food images and recipes from Allrecipe.com, clean and categorize them properly in 100 classes. Then we transfer, train and fine-tune pretrained neural network model such as VGG-16 and ResNet-18 on food image classification task. Third, we use trained neural network to extract 'feature vector' from food image, match 'feature vector' between food images, and retrieve

```
Ground Truth Title: Soundview Crab Salad
Ground Truth Category: 3 : salad cucumb avocado summer macaroni quinoa greek pea beet kale
```




```
tensor([[14,  0, 56, 23,  4]], device='cuda:0')
Top  1 predict category: 14 : chocol frost cupcak mouss doubl fudg cover dark hot german
Top  2 predict category: 0 : chicken bake breast wing enchilada buffalo parmesan thigh thai dumpl
Top  3 predict category: 56 : oatmeal bake overnight cinnamon cooki nut crispi butterscotch chewi mapl
Top  4 predict category: 23 : roast oven rosemari herb brussel sprout beet rib pork pan
Top  5 predict category: 4 : pie shepherd crust rhubarb custard cherri pot meringu key mini
```

Figure 9. A sample of false classification

recipes for input food image.

From this project, we start from data processing , label construction, neural network selection to recipe retrieval. We experience the whole process in turning a raw image to a human readable recipe and a piece of doable instruction. We learn that although food images are deform-able, we can use features combined with ingredients to label, and use different image transformation method including flipping, resizing to enrich our categories and later help machine to learn from the image. In addition to that, we compare different neural network structure by experimenting with layer modification and we also find that a slight change in parameter such as Dropout ratio would cause a 3 4% improvement in validation accuracy. However, also through the experiment, we find that there is not always the case to more layers the better. We have to take joint effect into considerations.

In the future, there are some aspect we can do to improve the performance of this model. First, we can scrawl more recipe from the Internet to enlarge our training set. In this project, we trained our model with a comparatively small training set(30000 training images and 100 categories, which means each category has less than 300 samples in average). And we need to find a better way to cluster and define our label. Second, from the view of system, it would be nicer to build our system on the website, for users from all over the world. We could further train our model by collecting images from users, which could be beneficial both to users and our system. Third, we use KNN for recipe retrieval, which is fine for small amount of data but slow for larger amount of data. We could replace K-Means of KNN to accelerate the matching process of 'feature vectors'.

References

[1] Jenny Cooper. "Cooking trends among millennials: Welcome to the digital kitchen". In: *Think with Google* (2015).

[2] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[3] Ahmad Babaeian Jelodar Kaoutar Ben Ahmed. "Fine-Tuning VGG Neural Network For Fine-grained State Recognition of Food Images". In: *arXiv:1809.09529* (2018).

[4] Chang Liu et al. "A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure". In: *IEEE Transactions on Services Computing* 11.2 (2017), pp. 249–261.

[5] Sara McGuire. "Food photo frenzy: inside the Instagram craze and travel trend". In: *Business. com, February* 22 (2017).

[6] Lili Pan et al. "Deepfood: Automatic multi-class classification of food ingredients using deep learning". In: *2017 IEEE 3rd international conference on collaboration and internet computing (CIC)*. IEEE. 2017, pp. 181–189.

[7] Amaia Salvador et al. "Inverse cooking: Recipe generation from food images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10453–10462.

[8] Amaia Salvador et al. “Learning cross-modal embeddings for cooking recipes and food images”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3020–3028.

[9] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[10] Keiji Yanai and Yoshiyuki Kawano. “Food image recognition using deep convolutional network with pre-training and fine-tuning”. In: *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE. 2015, pp. 1–6.

[11] Shulin Yang et al. “Food recognition using statistics of pairwise local features”. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pp. 2249–2256.