Your NutritionBro: Nutrition evaluation System

Qi Yin
qy652@nyu.edu

Mengxi Wu
mw4355@nyu.edu

1 Problem Statement

Food is essential for our lives. The fast development of social network platforms such as Facebook, Instagram and Twitter enables us to share the images of food. These food could be made at home or from restaurant. Most of time we are attracted by the appearance of the food in the images. However, it is not easy to identify the ingredients and nutrition of food in the images. Details such as nutrition of food inform us whether the food is healthy or suitable for us. Thus, this arouses a question: Could we obtain nutrition contents of food from a single image? Conventional neural network has demonstrated a major breakthrough in gathering information from images. Thus, we believe by exploring deep learning techniques, we could build a deep neural network to answer the details of food nutrition from a single image.

Our system contains two parts - ingredient detection and nutrient retrieval. For the first part, we found a suitable structure that can detection ingredients in image image, and then we designed a algorithm/structure to transform the list of ingredient into nutrition list. At last, we evaluated our system's performance and thinking future improvement.

2 Literature Survey

In last few years, people have achieved outstanding improvements in visual recognition tasks, such object detection, image classification, image caption. Currently, there is a vast literature in computer vision researching on food understanding. Such as estimating food quantities, find a the recipe for a given image. Some projects focus on ingredient and nutrition in the food. Im2Calories[4] is an excellent project with step of generic food detection, semantic image segmentation, column estimation, and finally calorie estimation. In inverse Cooking[6], they designed a system which firstly predict the ingredients from the image then generate recipe leveraging both image and ingredient list. And in 2018, a framework [7] which can estimate the ingredient, food attribute, food nutrition food in real time was proposed. It is implemented as a mobile platform which firstly detects food items in the images. On the other hand, they retrieve text data from Google and Common Crawl. And then computing vector representations with Word2Vec. So after recognize the food in image, the model can predict corresponding food attribute and food nutrition by comparing the similarity between feature vectors.

One important factor that boost research on food image is there are many powerful datasets. Food-101 proposed in 2014 gathers about 101000 images as well as GPS data and nutritional values. Later, two very large-scale dataset for food image tasks generated. Cookpad, proposed in 2017, contains additional information such as recipe, ingredients and process steps. Recipe1M [5] containing around 8000000 images as well as 1 million recipes. Beside, it involves structured extra information such as ingredients, instructions and recipe classes. In 2018, [4] create dataset Food201-Mutilabel and Food201-Segmented, which add segmentation tag and multi label based on dataset Food 101. These two datasets can be used in volume estimation tasks.

On the other hand, Transformer proposed in 2017 have achieved great advance in NLP area. So people naturally started to think if we can apply Transformer in computer vision tasks. [3] Researchers in Facebook AI put forwarded a vision version - Detection Transformer, which achieve competitive performance in objection detection and segmentation. And writer believes that with further parameter tuning the performance we can further improve the accuracy.

3 Model and Training

Our system mainly contains three modules - Image Encoder, Ingredient Decoder, Nutrition Retrieval.

3.1 Image Encoder

Image Encoder take an image ($360 \times 360 \times 3$) as input. Input images are prepossessed and output the representative vector. Usually people use CNN to extract features from image . Here we choose ResNet50 and ResNet18 as the base CNN network . In our model,we discard the last two layers of ResNet50/ ResNet18 (full connected layer) and instead add a convolution layer as well as drop out.

3.2 Ingredient Decoder:

The input of this decoder is the encoded information of a food images and its output is a series of predicted ingredients. We know the ingredients in the set are not necessarily independent. For instance, sugar and cream frequently appear together. To take care the dependencies of ingredients, we decide to model the set of ingredients as a list of ingredients, as a product of conditional probability. Thus, we decide to use a Transformer model as the ingredient decoder. However, with a normal transformer model, the order between ingredients will be taken into account. To avoid the order, we use a set Transformer. As suggested in previous set Transformer literature[6], to remove order, we can apply max-pooling on the softmax probabilities across time.

Ingredient Decoder Structure: The set transformer described above has of 4 blocks and we use 2 multi-head attentions. The structure of the block is shown in Figure 1.

3.3 Training

Training Setting We train the model to minimize the binary cross-entropy between the predicted ingredients and the ground truth. We select the mini-batch size to be 32. The optimizer we use is Adam Optimizer. The learning is set to be 0.0001 (which is the same as in ResNet). Due to the limitation of computation resource, we can not select very large epoch. We have trained with 6 epochs and 12 epochs for model with ResNet18, and with 6 epochs for model with ResNet50. We also apply early stopping during training with setting parameter patience to be 50. Our training set has 40905 images, validation set has 15150 images, and test set has 4545 images.

During the training we met some problems. Firstly, we find that after one epoch, the training error decrease as well as validation error increase greatly. It suffered from over fitting. And the reason is that the size of training data was too small. After we increase the number of training data, we overcome this problem. The training error and validation error both decrease gradually, which indicates that the model started to learn good features. Additionally, when we monitor the training process, we discover that our validation error is always less than training error. There are many possible reasons. We believe the major reason is that our dataset is too small and a little bit simple compared to the complexity and size of our model. Though we haven't implemented, the potential solution is to do multiple train-validation splits and average the results. Another possible reason could be dropout. Drop out was used during the training part, but not be used in the validation part. It is also possibly due to the time when the errors are measured. Training loss is measured during each epoch while validation loss is measured after each epoch. The model could be more mature at the end of each epoch. Besides these two possible reasons, the validation set that is easier than the training set may also cause this phenomenon.

3.4 Nutrition Retrieval:

There are no food image to nutrition dataset. Thus, in this paper, we first predict the ingredients and then use API to retrieve the nutrition contained in each ingredient. The API we use is from U.S. Department of Agriculture. The final output will be the nutrition contents retrieved from the website of U.S. Department of Agriculture according to the ingredient we predict.

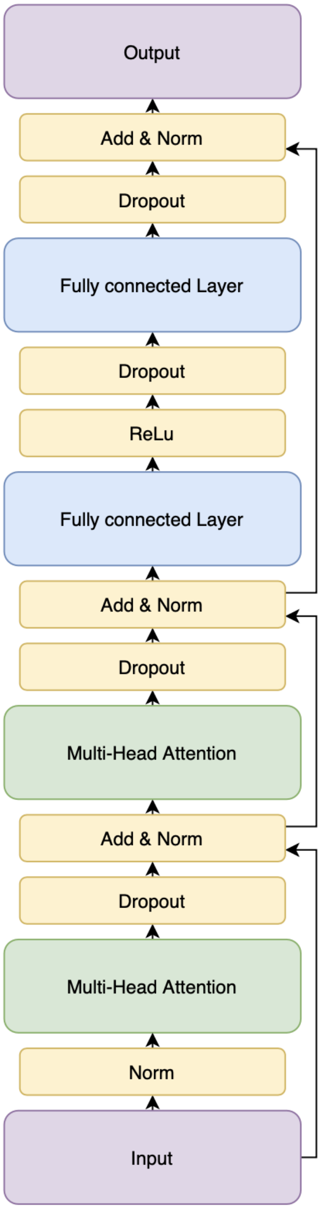


Figure 1: Structure of Block in Ingredient Decoder

Implement details: FoodData Central API provides REST access to FDC. In order to use that API, we need to firstly register on the website to get the access key(The API request is limited by 3600 request per hour per IP address). Then for each ingredient, we can query with REST API. And for each query, we will get a dataframe including all the related information for that ingredient. Then we distill the nutrient part, clean and merge it with other nutrient information related other ingredient in the list and finally will get a set of nutrient for each image.

4 Dataset

We plan to use **Recipe1M**[5] to train and test our encoder and decoder. Recipe1M contains recipes, ingredients and images, and we only use ingredients and image in it. However, the official website for downloading the dataset is shutdown for registrations. We used the unofficial dataset which we find from other website but lately we found the dataset has corrupted images. Thus, we gave up using this dataset.

Instead, we use **Food-101**[2] together with **Ingredients-101**[1]. **Food-101** contains 70000+ images of food from Foodspotting, organized by type of food(101 food classes). And this data set can be download through Kaggle API. **Ingredients-101** is a dataset for ingredients recognition. It contains the ingredients for each of the 101 classes in Food101 and can be download here. There are 446 unique ingredients. Besides adding caption of gradients list on images in Food101, **Ingredients-101** also divides the images in **Food-101** into three sets. The training set have 68175 images. The validation set has 7575 images. The test set has 25250 images

5 Results

5.1 Metric Evaluation

Model	Accuracy	F1
Res50 - 6 epoch	0.9415	0.4783
Res18 - 6 epoch	0.9302	0.4163
Res18 - 12 epoch	0.9445	0.4937

Our accuracy and F1 score are computed from true positive (TP), false positive (FP), true negative (TN) and false negative (FN). TP, FP, TN and FN are computed from one hot vector of predicted ingredients list and the ground truth ingredients list. The computation details can be referred to `metrics.py`.

The metric scores are evaluated on the test set from the dataset. Three models all achieve decent accuracy and F1 scores. Our model performance well on the dataset. Though intuitively ResNet50 should perform better than ResNet18, ResNet18 trained with 6 epochs performs as well as ResNet50 trained with 6 epochs on the test set. Additionally, training more epochs only slightly increases the accuracy and F1 scores. We concluded that ResNet18 is sufficient enough for mastering the dataset we choose. However, it may not imply that our model can generalize well. Generally speaking, transformer and ResNet which contain million of parameters need millions of images and hundred of epochs to generalize well. Our dataset only contains 446 types of ingredients and 101 classes of food. As we discuss in the training section, the dataset we select may be a little bit too simple for our model to master.

5.2 Sample Results and Analysis

In Figure 2, we summarize the prediction results for strawberry cheesecake and sushi from model with ResNet18 and ResNet50 respectively. The test images are from the test set of the dataset. Both network can predict most of ingredients of these two types of food. However, We can see that several ingredients at the beginning are very reasonable. Especially from the first one to fifth one, they are relatively accurate. However, sometimes the ingredients list is very long, and those ingredient that appear behind are not related to the test image. One of potential solutions we come up is to set a threshold for the softmax results. Once the probability is lower then a reasonable threshold, system will not display the prediction result.

Figure 3 shows the nutrition information retrieved from the API based on the model with ResNet18-6 epoch. In Figure 4, Information following the ingredient predictions is the raw nutrition contents retrieved via API provided by the U.S. Department of Agriculture. Not all the ingredients have the corresponding nutrition content. The underscore line for some ingredients may prevent the API returns the corresponding nutrition. To have better retrieved results we can remove the underscore line.

6 Conclusion

We planed to implement a system that can take image as input and out put a series of nutrient. We implemented the image encoder and ingredient decoder to detect a series of ingredients with the

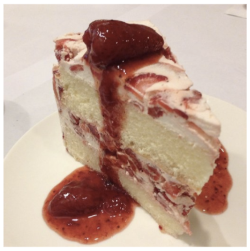
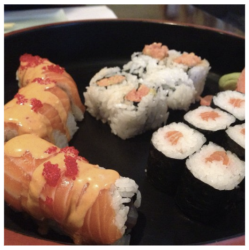
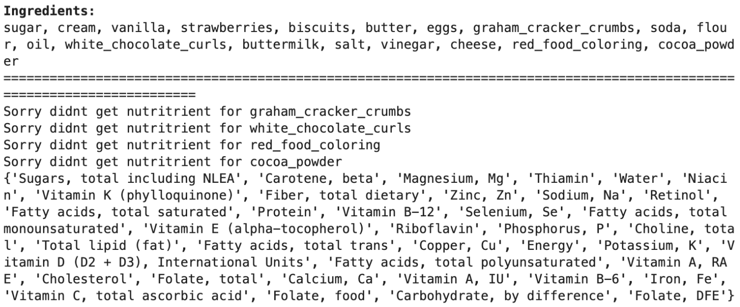
Image	Predicted Ingredients ResNet18-6 epoch	Predicted Ingredients ResNet50-6 epoch	True Ingredients
	**Sugar**, **Cream**, **Vanilla**, **Strawberries**, **Biscuits**, **Cheese**, Butter, Eggs, Graham cracker crumbs, Soda, Flour, Oil, White chocolate curls, Buttermilks, Salt, Vinegar, Red food coloring, Cocoa powder	**Sugar**, **Cream**, **Vanilla**, **Biscuits**, **Strawberries**, **Honey**, **Cheese**, Butter, Eggs, Graham cracker crumbs, Lettuce, Flour, Oil, Salt, Vinegar, pepper, parsley, pancake, chives, Yellow onion, Red food coloring,	**Strawberries, Sugar**, Heavy whipping **Cream, Vanilla**, Honey, **Biscuits, Cheese**
	**Salt, Sushi Rice, Vinegar, Water**, Oil, **Sashimi grade tuna**, carrot, cucumber, oregano, Purple onion, Chives, Cooking Spray	**Sugar, Sushi rice, Water, Vinegar, Sashimi grade tuna**, Mirin, Soda, Flour, Baking Powder, Icing, oil	**Sushi rice, Sashimi grade tuna, Water, Vinegar, Sugar, Salt**

Figure 2: Ingredient Predictions Sample with ResNet18 or with ResNet50

Nutrition for Sushi	Nutrition for Strawberry Cheesecake
Vitamin B-12 Cholesterol Calcium, Ca Fatty acids, total monounsaturated Zinc, Zn 22:5 n-3 (DPA) Carbohydrate, by difference Vitamin A, IU Carotene, beta Niacin Magnesium, Mg Sugars, total including NLEA Copper, Cu Carotene, alpha Folate, DFE Vitamin A, RAE Vitamin D (D2 + D3), International Units Vitamin E (alpha-tocopherol) Potassium, K Thiamin Selenium, Se Fatty acids, total polyunsaturated Sodium, Na Vitamin B-6 Fatty acids, total saturated Iron, Fe Riboflavin Water Ash Lutein + zeaxanthin Energy 22:6 n-3 (DHA) Manganese, Mn Cryptoxanthin, beta Folate, total Protein Folate, food Fiber, total dietary 20:5 n-3 (EPA) Phosphorus, P Total lipid (fat) Vitamin D (D2 + D3) Choline, total Vitamin C, total ascorbic acid Vitamin K (phylloquinone)	Vitamin B-12 Cholesterol Calcium, Ca Fatty acids, total monounsaturated Zinc, Zn Vitamin A, IU Carbohydrate, by difference Carotene, beta Niacin Magnesium, Mg Sugars, total including NLEA Copper, Cu Folate, DFE Vitamin A, RAE Vitamin D (D2 + D3), International Units Vitamin E (alpha-tocopherol) Potassium, K Thiamin Selenium, Se Fatty acids, total polyunsaturated Sodium, Na Vitamin B-6 Fatty acids, total saturated Iron, Fe Riboflavin Water Energy Folate, total Protein Folate, food Fiber, total dietary Fatty acids, total trans Total lipid (fat) Phosphorus, P Retinol Choline, total Vitamin C, total ascorbic acid Vitamin K (phylloquinone)

Figure 3: Nutrition Retrieved from API based on the Prediction from ResNet18 with 6 epochs

help of transformer (based on model from the paper [6]). And then based on the ingredients list



```
Ingredients:
sugar, cream, vanilla, strawberries, biscuits, butter, eggs, graham_cracker_crumbs, soda, flou
r, oil, white_chocolate_curls, buttermilk, salt, vinegar, cheese, red_food_coloring, cocoa_powd
er
===============================================================================================
=========================
Sorry didnt get nutritrient for graham_cracker_crumbs
Sorry didnt get nutritrient for white_chocolate_curls
Sorry didnt get nutritrient for red_food_coloring
Sorry didnt get nutritrient for cocoa_powder
{'Sugars, total including NLEA', 'Carotene, beta', 'Magnesium, Mg', 'Thiamin', 'Water', 'Niaci
n', 'Vitamin K (phylloquinone)', 'Fiber, total dietary', 'Zinc, Zn', 'Sodium, Na', 'Retinol',
'Fatty acids, total saturated', 'Protein', 'Vitamin B-12', 'Selenium, Se', 'Fatty acids, total
monounsaturated', 'Vitamin E (alpha-tocopherol)', 'Riboflavin', 'Phosphorus, P', 'Choline, tota
l', 'Total lipid (fat)', 'Fatty acids, total trans', 'Copper, Cu', 'Energy', 'Potassium, K', 'V
itamin D (D2 + D3), International Units', 'Fatty acids, total polyunsaturated', 'Vitamin A, RA
E', 'Cholesterol', 'Folate, total', 'Calcium, Ca', 'Vitamin A, IU', 'Vitamin B-6', 'Iron, Fe',
'Vitamin C, total ascorbic acid', 'Folate, food', 'Carbohydrate, by difference', 'Folate, DFE'}
```

Figure 4: Res18-6-epoch Model Test: Strawberry cheese

we retrieved corresponding nutrition from FoodData Central API. For the image encoder part, we tried different CNN models, Res150 and Res18. Also we applied the model to dataset Food101 and compared their performance. Our project can be found here Your NutritionBro.

7 Future Work

There are plenty of places we can make improvement. First, we can set up a more stable environment with more computation resources to run our experiments. Currently, we use Floydhub. The GPU resource is limited though with a convenient user interface. We can try Google Cloud Platform. Second, we need to gather larger dataset to train our model. Third, we may improve our nutrition retrieval by removing the underscore line in the ingredient names. We can also turn our project into a iOS app to make it viable to more people.

References

[1] Marc Bolaños, Aina Ferrà, and Petia Radeva. "Food Ingredients Recognition through Multi-label Learning". In: *In Proceedings of the 3rd International Workshop on Multimedia Assisted Dietary Management (ICIAP Workshops)*. 2017.

[2] Lukas Bossard, M. Guillaumin, and L. Gool. "Food-101 - Mining Discriminative Components with Random Forests". In: *ECCV*. 2014.

[3] Nicolas Carion et al. *End-to-End Object Detection with Transformers*. 2020. arXiv: 2005.12872 [cs.CV].

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

[5] A. Salvador et al. "Learning Cross-Modal Embeddings for Cooking Recipes and Food Images". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 3068–3076.

[6] Amaia Salvador et al. *Inverse Cooking: Recipe Generation from Food Images*. 2019. arXiv: 1812.06164 [cs.CV].

[7] R. Yunus et al. "A Framework to Estimate the Nutritional Value of Food in Real Time Using Deep Learning Techniques". In: *IEEE Access* 7 (2019), pp. 2643–2652.