# FoodNet: Transfer Learning for Food Classification

**Yunhan Tang A53244500**
**Wentao Wei A53243471**
**Yuyi Tan A53220110**

## Abstract

In this work, we use CNN transfer learning to classify 101 kinds of food. Pre-trained Convolutional Networks VGG16, VGG19 and double CNN networks were used as the feature extractors, with fully connected network as transfer learning classifier. Top-N method were implemented to analysis the results more deeply and profoundly. In order to better understand the performance difference, we use a novel visualization technique that gives insight into the function of intermediate feature layers and the operation of the classifier. we also discussed the application of this project on recommendation systems which enable business to make promotions based on the classification result.

## 1 Introduction

### 1.1 Background

Food plays an import role in our daily life, taking pictures of food and sharing them on the social networks has become part of people's life in recent years, and is has become a kind of culture. Accordingly, food image classification has been needed by multiple commercial fields, all kinds of applications, such as measure the calorie, make recommendations for customer.Food image classification has become a key issue for intelligent food materials receiving and food supply chain applications. Thus, how to efficiently, accurately and quickly detect the image of food materials is a challenging research topic and will have high value of scientific research and business[1].

Traditional methods based on principal component analysis and local binary features are restricted by hardware performance and cannot deal with large amount of image data and has low accuracy in recognition effect[2]. However, Convolutional Neural Network(CNN) is a powerful tool for feature extraction and for handling massive data which enjoys a great success in image and video recognition and outperforms human beings. In this project, Convolutional Neural Network(CNN) was implemented based on two modules. One module used to extract features is VGG (Very deep convolutional neural network for large-scale image recognition), which was proposed in 2014.and trained by Imagenet. And another module is used to do the classification.

### 1.2 Convolutional Neural Network

Artificial neural network is the model which simulates functions of neural network in the biology. The artificial neural network is powerful and plays the important role in the artificial intelligence. The basic unit in artificial neural network is the neuron.Each neuron is connected by other neurons with different weights and it has its own activation function which gives output after receiving input from previous layers

Traditional neural network is fully connected which each neuron is connected with all neurons in the next layer. There is only one version of weights between neighbor layers. When the input is noisy, like scaling, variance, the fully connected neural network performs not well. In 1998, Lecunn presented Lenet-5, which brings us the convolutional network. [3]

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

Convolutional Neural Network is very similar to the traditional artificial neural network which consists of neurons, weighted connections and activation functions. Basically, the CNN is divided into feature extraction layer and classifier layer. The neurons of feature extraction layer are partially connected where each neuron only connect neurons within specific area of last layer. The neurons' inputs are computed by convolution operation. In this case, each neuron only focuses on specific area and extract corresponding lower level features. The classifier layer is consist of several fully connected layer which combines previous lower level feature.

Over years, the CNN becomes deeper and deeper because it is shown that the deeper the network is, the better the performance is. However, training a CNN from the scratch costs too much time and computing resources. Instead, it is more convenient to use pre-trained model and re-train it in order to apply on the new task.

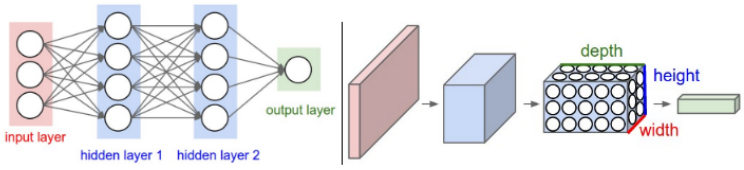


Figure 1: CNN structure

### 1.3 VGG16

VGG16, proposed by Simonyan and Zisswerman, is a deep Convolutional Neural Network model which can achieve 92.7% top-5 accuracy in ImageNet database.

The detailed structure is shown as Figure 2. The input of training VGG16 is fixed-size 224*224 RGB images, and the output is the classification results contains 1000 channels(one for each class). The image is passed through a stack of convolution layers with a very small 3 x 3 receptive field. A 1x1 convolution filters is also used in one of the configurations and can be seen as a linear mapping of the input channels. The task of Spatial pooling has been carried out using five max-pooling layers, which follow some of the convolutional layers but not all the convolutional layers.[1]

The stack of convolutions layers have different depth, and each followed by a Max-pooling, performed over the window of 2x2 pixel, stride 2. After the convolution layers, there exist three Fully-Connected(FC) layers with the first two have 4096 channels each and the third one have 1000 channels.

The stack of convolutional layers is followed by three fully- connected layers: the first two layers have a total of 4096 channels each, and the third layer performs a 10 0 0-way ILSVRC classi- fication. The final layer is the softmax layer. All hidden layers are equipped with the rectification(ReLU)nonlinearity.[4]

### 1.4 VGG19

The structure of VGG19 is very similar to the structure of VGG16, the details is shown as Figure 3.In VGG19, there is an added convolutional layer in each of the last three stacks, which is of the same depth as the original existing layers in the stack, equals to 256, 512 and 512. The remaining fully connected layers of the network has no change compared to VGG16.

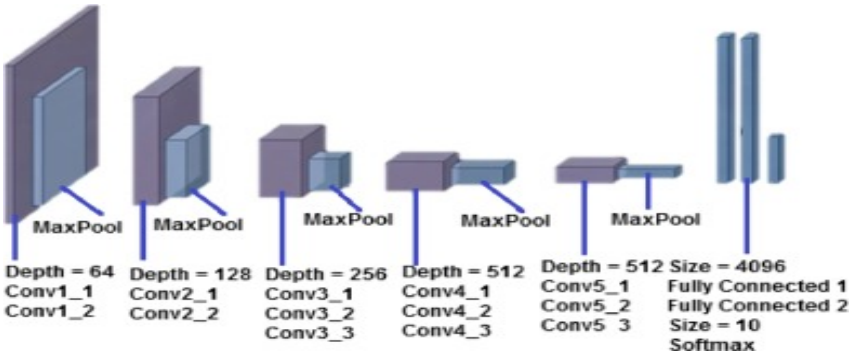


Figure 2: VGG 16 structure

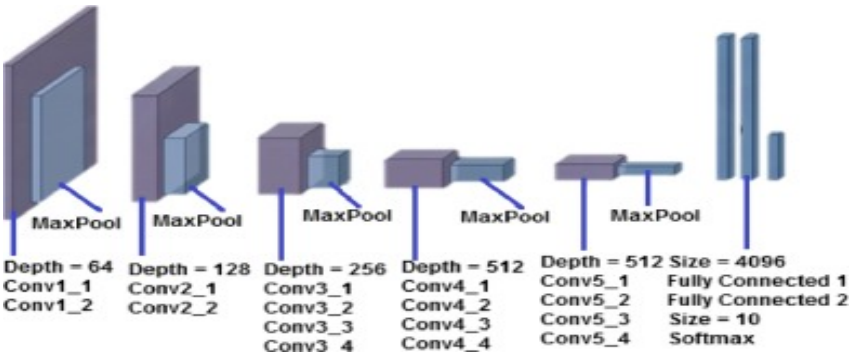


Figure 3: VGG 19 structure

## 2 Methods

### 2.1 Artitecture

Our models is based on pretrained models which are divided into feature extraction part and classifier part.

**Feature extraction**: The feature extraction part is the partial model of pretrained VGG16 and VGG19.[5][1]. The partial models are the copies of these pretrained model except for their last classifier layer. The inputs of these models are RGB images of 224 * 224 size and their outputs are one dimensional flat tensors. Since VGG16 and VGG19 are trained by huge amount of images of different classes, their convolutional layers are good tools to extract both low level features and high level features. In this case, we can use these abstract 1*4096 or 1*8192 size tensors as the representation of raw images.

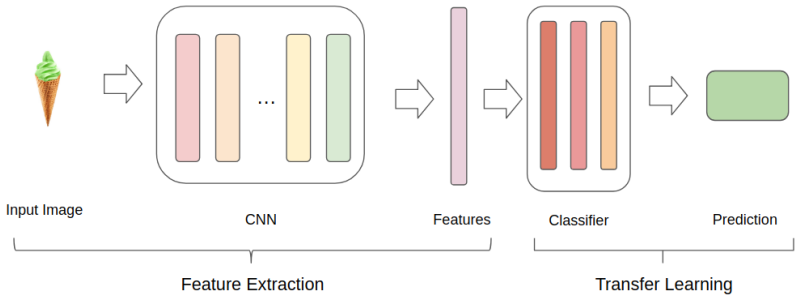


Figure 4: FoodNet V1 structure

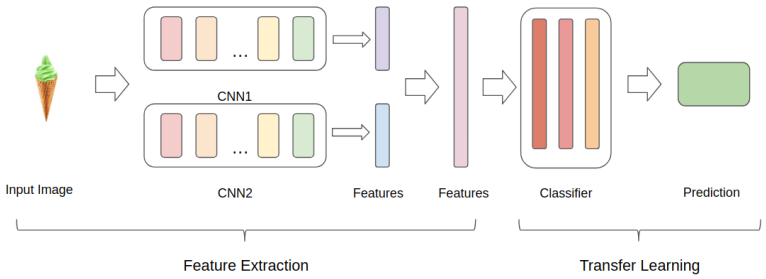


Figure 5: FoodNet V2 structure

**Classifier**: After extracting the features from the input, we use 3 fully connected linear layers to classify food into 101 classes. The input is a tensor with the abstract feature from the feature extraction part and the output is a 1 * 101 tensor where the last layer is the Softmax layer that specify the probabilities of 101 classes the input tensor belongs to. Then we choose the first $N$ biggest class probability as top $N$ result.

### 2.2 Convolutional Neural Network Visualizations

Our code uses pretrained AlexNet or VGG from the model zoo. Some of the code also assumes that the layers in the model are separated into two sections; features, which contains the convolutional layers and classifier, that contains the fully connected layer (after flatting out convolutions).

## 3 Experiment

### 3.1 Dataset

Our dataset is food-101[6], which was presented in 2014 with 101'000 images. The images are divided into training set and test set. Both training set and test set have 101 classes of food. In the training set, there are 750 pictures per class in training set and there are 250 pictures per class in the test set.

Figure 6: Food-101 dataset

### 3.2 Evaluation

Since we are intend to know the top $N$ classification result, we use the top 1 results and training label as the cross entropy loss function.

### 3.3 Hyperameters

Table 1: Hyperameters overview

| Feature | Classifier | Learning Rate | Optimizer |
|---|---|---|---|
| VGG16 | 4096+2048+101 | 0.0001 | Adam |
| VGG19 | 4096+2048+101 | 0.0001 | Adam |
| VGG16 + VGG19 | 8192+2048+101 | 0.0001 | Adam |

### 3.4 Convolutional Neural Network Visualizations

In this paper, large Convolutional Network models have demonstrated impressive classification performance on some categories of food but behave not that well on some other categories. However we don't have clear understanding of why they perform so well, or how they might be improved. In order to better understand the performance difference, we use a novel visualization technique that gives insight into the function of intermediate feature layers and the operation of the classifier.

## 4 Results

We have implemented VGG16, VGG19 and VGG16+VGG19 transfer learning experiment.

### 4.1 VGG16 Top N result

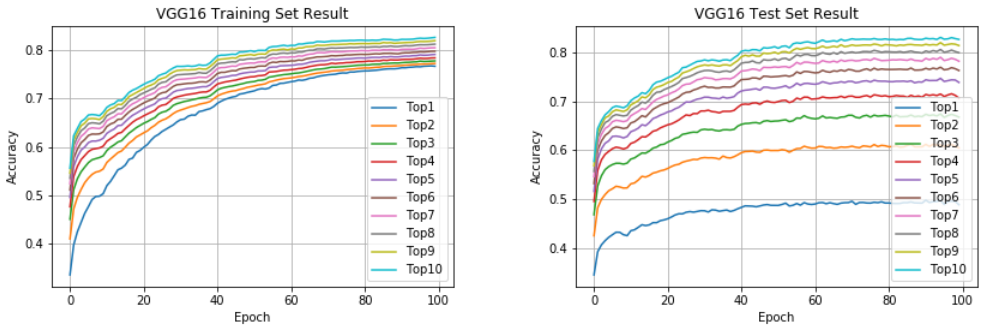


Figure 7: VGG 16 result

### 4.2 VGG19 Top N result

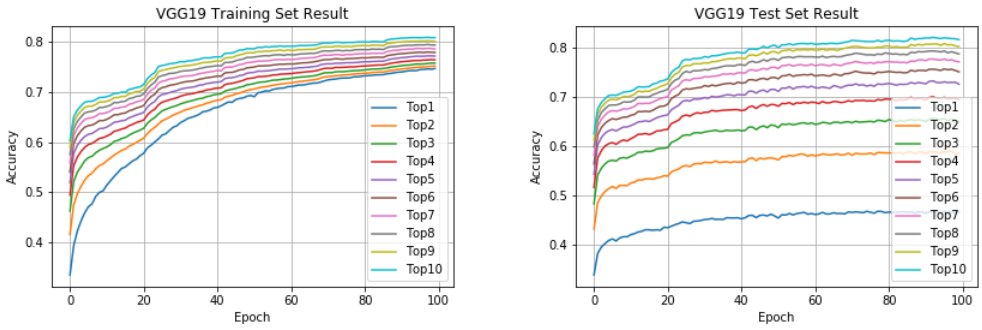


Figure 8: VGG 19 result

### 4.3 VGG16+VGG19 Top N result

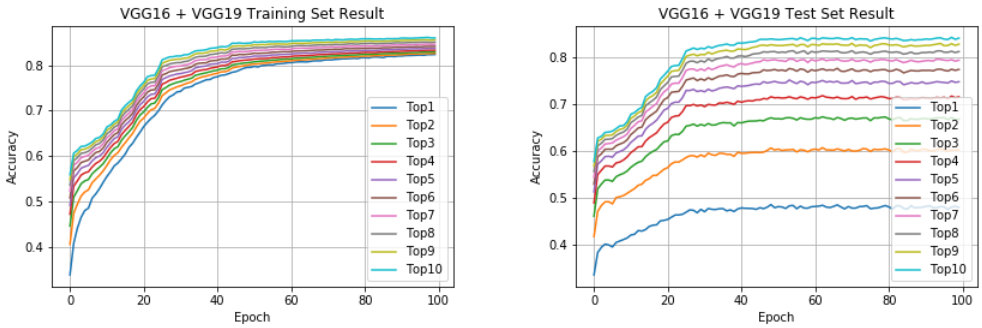


Figure 9: VGG16 + VGG19 result

### 4.4 Food Category Classification Accuracy

| | food_name | accuracy | | food_name | accuracy | | food_name | accuracy | | food_name | accuracy |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 33 | edamame | 0.944 | 52 | gyoza | 0.628 | 19 | chicken_quesadilla | 0.480 | 50 | grilled_salmon | 0.368 |
| 63 | macarons | 0.832 | 83 | red_velvet_cake | 0.624 | 17 | cheese_plate | 0.480 | 67 | omelette | 0.368 |
| 32 | dumplings | 0.828 | 41 | french_onion_soup | 0.620 | 20 | chicken_wings | 0.480 | 47 | gnocchi | 0.360 |
| 64 | miso_soup | 0.784 | 27 | creme_brulee | 0.620 | 59 | lasagna | 0.468 | 36 | falafel | 0.348 |
| 54 | hot_and_sour_soup | 0.784 | 23 | churros | 0.616 | 21 | chocolate_cake | 0.468 | 39 | foie_gras | 0.340 |
| 70 | pad_thai | 0.752 | 7 | bibimbap | 0.608 | 3 | beef_carpaccio | 0.456 | 26 | crab_cakes | 0.336 |
| 51 | guacamole | 0.748 | 55 | hot_dog | 0.604 | 48 | greek_salad | 0.452 | 9 | breakfast_burrito | 0.328 |
| 40 | french_fries | 0.744 | 29 | cup_cakes | 0.600 | 38 | fish_and_chips | 0.452 | 96 | tacos | 0.324 |
| 86 | sashimi | 0.736 | 43 | fried_calamari | 0.596 | 53 | hamburger | 0.444 | 18 | chicken_curry | 0.316 |
| 69 | oysters | 0.732 | 78 | poutine | 0.592 | 85 | samosa | 0.440 | 77 | pork_chop | 0.312 |
| 75 | pho | 0.728 | 61 | lobster_roll_sandwich | 0.584 | 46 | garlic_bread | 0.436 | 87 | scallops | 0.308 |
| 91 | spaghetti_carbonara | 0.720 | 34 | eggs_benedict | 0.560 | 80 | pulled_pork_sandwich | 0.428 | 4 | beef_tartare | 0.300 |
| 88 | seaweed_salad | 0.708 | 11 | caesar_salad | 0.560 | 28 | croque_madame | 0.424 | 15 | ceviche | 0.296 |
| 31 | donuts | 0.696 | 25 | club_sandwich | 0.548 | 62 | macaroni_and_cheese | 0.424 | 22 | chocolate_mousse | 0.288 |
| 65 | mussels | 0.692 | 92 | spring_rolls | 0.540 | 14 | carrot_cake | 0.420 | 5 | beet_salad | 0.276 |
| 68 | onion_rings | 0.688 | 97 | takoyaki | 0.536 | 72 | pancakes | 0.416 | 37 | filet_mignon | 0.268 |
| 6 | beignets | 0.688 | 42 | french_toast | 0.528 | 95 | sushi | 0.416 | 82 | ravioli | 0.260 |
| 24 | clam_chowder | 0.672 | 98 | tiramisu | 0.528 | 84 | risotto | 0.412 | 56 | huevos_rancheros | 0.252 |
| 60 | lobster_bisque | 0.672 | 94 | strawberry_shortcake | 0.512 | 13 | caprese_salad | 0.412 | 8 | bread_pudding | 0.236 |
| 35 | escargots | 0.664 | 71 | paella | 0.512 | 74 | peking_duck | 0.404 | 57 | hummus | 0.232 |
| 76 | pizza | 0.664 | 2 | baklava | 0.504 | 73 | panna_cotta | 0.396 | 0 | apple_pie | 0.220 |
| 100 | waffles | 0.660 | 44 | fried_rice | 0.500 | 81 | ramen | 0.388 | 99 | tuna_tartare | 0.200 |
| 30 | deviled_eggs | 0.656 | 12 | cannoli | 0.496 | 66 | nachos | 0.384 | 10 | bruschetta | 0.004 |
| 90 | spaghetti_bolognese | 0.640 | 1 | baby_back_ribs | 0.492 | 79 | prime_rib | 0.384 | 16 | cheesecake | 0.000 |
| 45 | frozen_yogurt | 0.632 | 58 | ice_cream | 0.484 | 89 | shrimp_and_grits | 0.372 | 93 | steak | 0.000 |
| | | | | | | | | | 49 | grilled_cheese_sandwich | 0.000 |

Figure 10: Food Category Classification Accuracy

### 4.5 Convolutional Neural Network Visualizations

We pick three kind of food from 101 categories, respectively are ice cream, hot dog and spaghetti. For each type of food, we pick three different shape and color images to compare their differences. For each image, we implemented the Convolutional Neural Network visualizations in four different ways. For each row, the first image is the original image. And for the remaining images are respectively colored vanilla backpropagation, guided backpropagation saliency, gradient-weighted class activation map and gradient-weighted class activation heatmap. The last image in each row is gradient-weighted class activation heatmap on image, it's clear to see the which part of the image is used for classification. The deeper the color is, the corresponding area is giving more weight to classify this image.

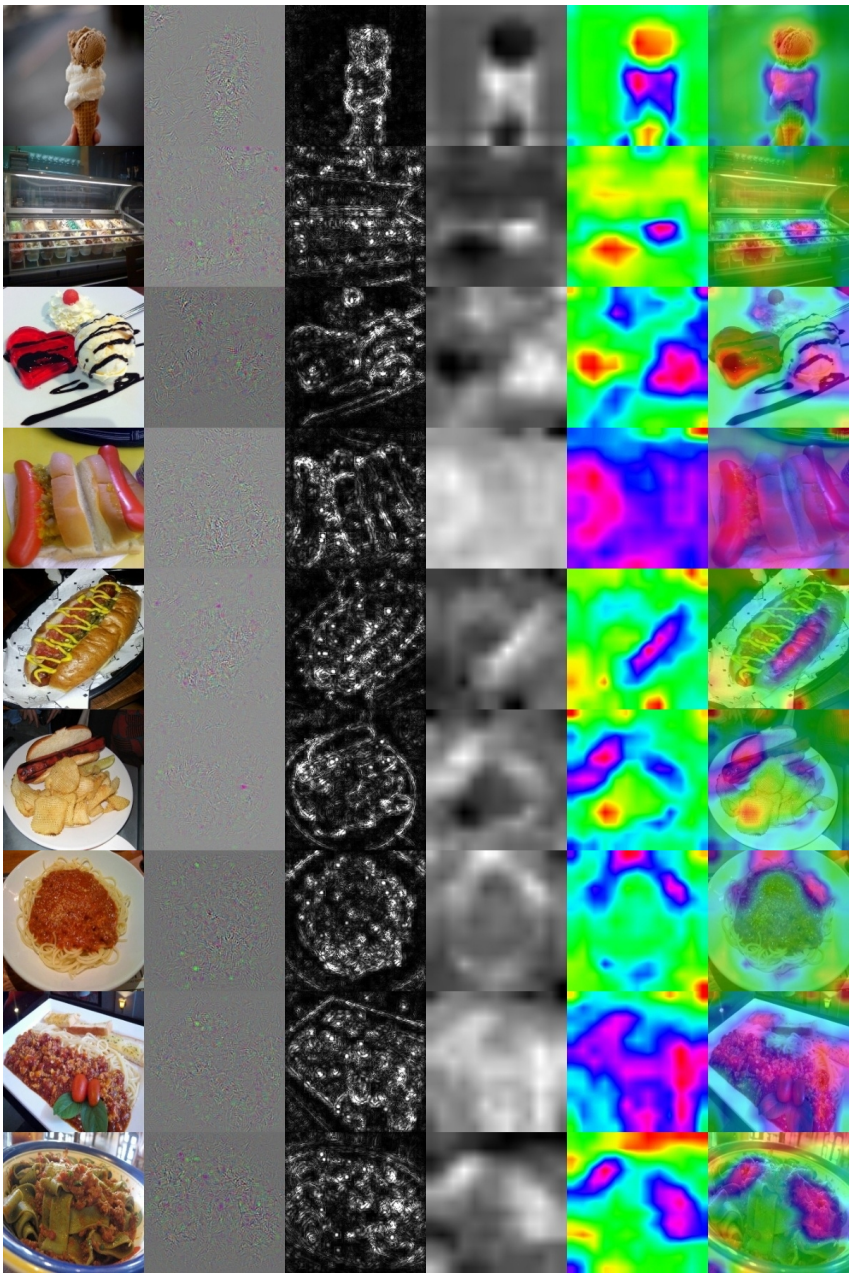

Figure 11: Convolutional Neural Network Visualizations

# 5 Discussion

## 5.1 Accuracy Analysis

We can see from Fig 7 to Fig 9 that we get best results using the VGG16 and VGG19 combined model. The Top10 accuracy of VGG16 and VGG19 are eventually about 80% but the Top1 accuracy is around 77%. In the VGG16 and VGG19 combined model, all the accuracies are above 80% and this is clearly better than a single VGG16 or VGG19 model. Also, the combined model converges faster than a single model.

## 5.2 Error Analysis

As it is shown on the graph above, for **Top-1**, both of the training error and test error has been decreasing during the training, but the final accuracy shows that test error decreased far less than the training error. Also,**Top-1** accuracy for training set and test set are much lower than **Top-2** and **Top-3**. These might because of the following reasons.

First, the reason why the **Top-1** only has around 50% accuracy not only because the food images used to train were taken under poor light condition which have unclear edges and not obvious color information, but also because there always exists some decorations on the food that influenced the model to extract features.

Second, some categories substantially have several similar features, they are made from the same ingredients and are made as very similar shapes, all with some decorations, such as the "baby back ribs", "pork chop" and "steak", there are all meat that cooked by gilling. Accordingly, if **Top-2** or **Top-3** were considered, the tasks actually can be seen as classify by classes.

## 5.3 Food Category Classification Accuracy Analysis

We can see from the Fig 10, the top 5 accuracy of food category classification are edamame, macarons, dumplings, miso soup and hot and sour soup. It's not hard to conclude that the common features of these food. Edamame, macarons and dumplings are in block shape which means they are separate with each other. And the second to fifth food are all soups, it means that the liquid food is easy to classify.

The last 5 accuracy of food category classification are grilled cheese sandwich, steak, cheesecake, bruschetta and tuna tartare. Fig 12 is a table including these food and their top 5 incorrect classification (250 test images). The number stands for the times when they are incorrectly classified as another type of food. For steak, they are most likely to be classified as pork chop. As we can see from Fig 13, the differences between pork chop and steak are so tiny, even human can't tell the difference. Another reason is that the definition of one food is not that precise, there may be some overlap between two type of food.

| Grilled_cheese_sandwich | | Steak | | Cheesecake | | Bruschetta | | Tuna_tartare | |
|---|---|---|---|---|---|---|---|---|---|
| garlic_bread | 62 | pork_chop | 112 | tiramisu | 85 | lobster_roll_sandwich | 61 | ceviche | 18 |
| pulled_pork_sandwich | 59 | baby_back_ribs | 94 | chocolate_cake | 71 | beef_carpaccio | 39 | beef_tartare | 18 |
| club_sandwich | 56 | filet_mignon | 70 | panna_cotta | 64 | caprese_salad | 38 | panna_cotta | 13 |
| hamburger | 48 | grilled_salmon | 45 | strawberry_shortcake | 56 | sushi | 27 | foie_gras | 12 |
| french_toast | 44 | french_toast | 41 | red_velvet_cake | 52 | takoyaki | 27 | guacamole | 12 |

Figure 12: Incorrect Classification Analysis

Figure 13: Pork Chop and Steak Comparison

### 5.4 Convolutional Neural Network Visualizations Analysis

From Fig 6 we can have some insights into the function of intermediate feature layers and the operation of the classifier. First, we can look at the three ice cream images. The first ice cream image is clear without any background disturbance. Thus we can see from the Gradient-weighted Class Activation Heatmap, the most "hot" area of Gradient-weighted Class Activation is the cream itself. The second image is a freezer inside which are ice cream. Intuitively, this image is definitely harder to classify compare with the first image, because it's not only the ice cream itself and the ice cream are in the freezer which increases the difficulty to classify the image. According to the heatmap, we can see that the network can still find the correct ice cream area to classify the food. The third image is three ice cream balls with some chocolate cream on it in a plate. To be noticed, this image is classified as cheese cake instead of ice cream which makes sense by just looking at their appearances, they do look alike. And we also can see that the network use the chocolate cream area to classify the ice cream which is actually the common feature between the two items.

For the images of hot dog, the first and third image are right classifications. And the second image is a wrong classification. We see that both the first and third classification are based on the ham sausage. However, in the second image, it's based on the bread. Because there are cream on the ham sausage in the second image.

For the image of spaghetti, the last two images are correct classification. It's clear to see from the heat map, the classification is based on the meat paste on top of the noddles. However, the classification of the first image is based on some marginal noddles rather than the biggest meat paste part in the middle. This is really a surprising result. Notice that the meat paste is on one side or scattered around in the last two images rather than in the middle. This may explain why the first classification is incorrect.

## 6 Application

The result of this project actually has be applied to commercial and business. In recent years, people focus more and more on the food and diet, and are more willing to seek great foods through multiple web Apps or mobile Apps. Thus, the classification results based on top-N task can be used by business to make recommendations for their customers, attracting more potential customers and maintain the old customers.

For example, if someone share a "steak" picture to the social networks with a comment "it's really good", then the business can do the classification of this picture. After the customer use the App in the future, the business can recommend some other food that similar to the steak.

## References


[1] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[2] Hao Chen, Jianglong Xu, Guangyi Xiao, Qi Wu, and Shiqin Zhang. Fast auto-clean cnn model for online prediction of food materials. *Journal of Parallel and Distributed Computing*, 117:218 – 227, 2018.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[4] Sarfaraz Masood, Abhinav Rai, Aakash Aggarwal, M.N. Doja, and Musheer Ahmad. Detecting distraction of drivers using convolutional neural network. *Pattern Recognition Letters*, 2018.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[6] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.