Unlabeled Food Image based Nearby Restaurant Recommendation System

Yoojin Oh, Sujin Kim, Seoyeon Ye, Ahyun Ji
Ewha Womans University
{mydianaoh, sml09181, s25y25n, nolli77}@ewhain.net

Abstract

An inherent limitation of conventional map applications is that it relies on users inputting specific food names for restaurant recommendations. The focal challenge lies in scenarios where users possess only images of food without knowledge of their names, rendering traditional search methods ineffective. In this paper, we propose the innovative restaurant recommendation system that identifies the depicted food and subsequently recommends nearby restaurants that serve that specific dish to address this gap. Specifically, we trained Convolutional Neural Networks (CNN) with pre-trained weights learned on a larger image dataset, achieving an accuracy up to 90.2% on InceptionV3.

1. Introduction

With the growing convenience of location-based recommendation systems, an increasing number of users are turning to applications such as Naver Map or KakaoMap to explore nearby restaurants and cafes. However, existing applications require users inputting the names of specific dishes. This poses a challenge when users cannot recall the names of the foods they desire, making the search process less straightforward.

Most food images with same or similar category have similar visual features. Based on this knowledge, this project introduces a solution utilizing Convolutional Neural Networks (CNN), or deep learning models, for an image classification task, which may provide efficient mechanism for discovering nearby restaurants for users. The project specifically targets the limitations observed in existing applications, offering users a more intuitive and visually-driven experience as they embark on their culinary exploration.

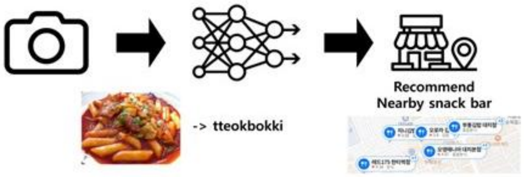


Figure 1: The overall pipeline of our proposed food image based nearby recommendation system.

2. Related Work

ImageNet-1K. ImageNet-1K is a large visual database with 1.4M color images in 1,000 classes. ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) was held, and various models were proposed for the ImageNet Classification. Most of pre-trained models used in the original paper was also used in this project [1]. The proposed models in the paper showed good performance in classification on ImageNet dataset. VGG [2] was the runner-up of ILSVRC 2014, using multiple 3x3 convolutional filters. Despite its simplicity, VGG16 demonstrated strong performance on image classification tasks. ResNet was proposed by He et al. (2015) for image recognition and the winner of ILSVRC 2015 [3]. It solves vanishing gradient problem by Residual Learning. InceptionNet [4] increased depth with few parameters of 4M required, using Inception module. DenseNet [5] was introduced by Huang et al. (2016). DenseNet connects each layer to every other layer in a feed-forward fashion to alleviate the vanishing-gradient problem.

Transfer Learning. Transfer Learning is a technique in machine learning (ML) in which knowledge learned from a task is re-used in order to boost performance on a related task [7]. As our task and ImageNet dataset, a large dataset, has similar label and example space, we decided to implement transfer learning in order to boost performance significantly. Specifically, four models, VGG16, ResNet50, InceptionV3, and DenseNet161 were used.

3. Data

Food images were derived from AI Hub and Naver Place. Restaurant information was gathered from Naver Place. As AI Hub food datasets are well preprocessed, we decided to add crawled images from Naver Place to add some noise to make more robust models. We focused on restaurants near Yangjae, Gangnam, and Samseong station. The crawled dataset from Naver Place specifically contains the restaurant's ID and name, as well as supercategory and the name of each food.

AI Hub dataset was composed of 400 classes with over 2,000 images each, totally 800K images. We decided to reduce the number of classes and samples by selecting the classes which is in the Naver Place data. Therefore, the number of classes decreased into 64 classes. Also, for the reduction of the samples, the bad examples of images in the figure 3 were deleted as they have too much noise or strange form. Also, the images were resized to 224x224 except for InceptionV3. In InceptionV3, images were resized to 299x299.

The created dataframe of Naver Place dataset consists of the supercategory column and the menu name column which is revised into the food names in AI Hub data. The information of AI Hub data was consolidated into a CSV file containing ID, image path, category number, and image name. We performed label encoding on the name column to train our model. Thus, the number of supercategories is 15 and the number of the food names is 64.

We tested performance of Resnet50 on partial dataset to get some basic configures including transformations. Specifically, we loaded images, split them into train and test sets, and applied normalization and data augmentation techniques such as resizing, color jitter, and random flips. As a result, default epochs was set as 10, learning rate as 5e-5, batch size as 8. (Additional tunings were applied for each model based on these configures.)

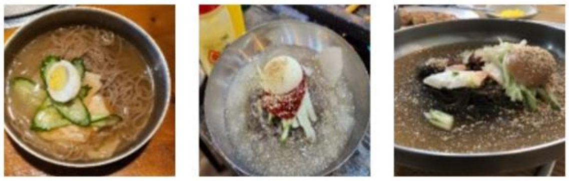


Figure 2: 'Naengmyeon' images in AI Hub

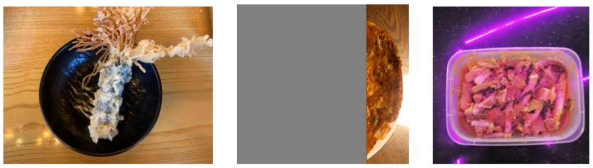
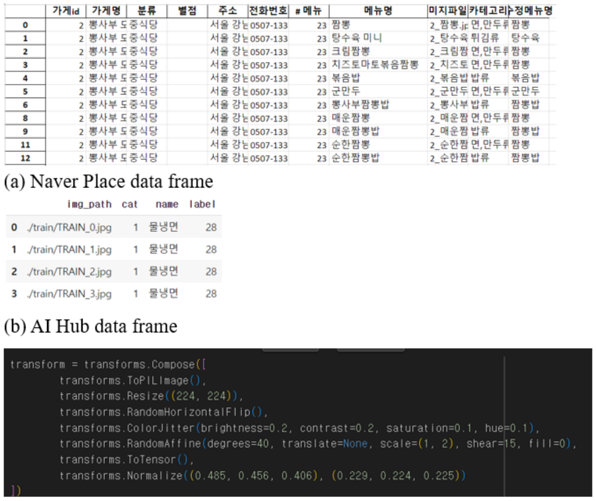


Figure 3: Bad images (left : strange shape of ‘Gimmaritwigim’, middle : partially obscured, right : galaxy included)



	가게id	가게명	분류	별점	주소	전화번호	# 메뉴	메뉴명	미지파일	카테고리	수정메뉴명
0	2	뽕사부 도	중식당		서울 강ㄴ	0507-133	23	짬뽕	2_짬뽕.jp	면,만두류	짬뽕
1	2	뽕사부 도	중식당		서울 강ㄴ	0507-133	23	탕수육 미니	2_탕수육	튀김류	탕수육
2	2	뽕사부 도	중식당		서울 강ㄴ	0507-133	23	크림짬뽕	2_크림짬	면,만두류	짬뽕
3	2	뽕사부 도	중식당		서울 강ㄴ	0507-133	23	치즈토마토볶음짬뽕	2_치즈토	면,만두류	짬뽕
4	2	뽕사부 도	중식당		서울 강ㄴ	0507-133	23	볶음밥	2_볶음밥	밥류	볶음밥
5	2	뽕사부 도	중식당		서울 강ㄴ	0507-133	23	군만두	2_군만두	면,만두류	군만두
6	2	뽕사부 도	중식당		서울 강ㄴ	0507-133	23	뽕사부짬뽕밥	2_뽕사부	밥류	짬뽕밥
8	2	뽕사부 도	중식당		서울 강ㄴ	0507-133	23	매운짬뽕	2_매운짬	면,만두류	짬뽕
9	2	뽕사부 도	중식당		서울 강ㄴ	0507-133	23	매운짬뽕밥	2_매운짬	밥류	짬뽕밥
11	2	뽕사부 도	중식당		서울 강ㄴ	0507-133	23	순한짬뽕	2_순한짬	면,만두류	짬뽕
12	2	뽕사부 도	중식당		서울 강ㄴ	0507-133	23	순한짬뽕밥	2_순한짬	밥류	짬뽕밥

(a) Naver Place data frame

	img_path	cat	name	label
0	./train/TRAIN_0.jpg	1	물냉면	28
1	./train/TRAIN_1.jpg	1	물냉면	28
2	./train/TRAIN_2.jpg	1	물냉면	28
3	./train/TRAIN_3.jpg	1	물냉면	28

(b) AI Hub data frame

```
transform = transforms.Compose([
        transforms.ToPILImage(),
        transforms.Resize((224, 224)),
        transforms.RandomHorizontalFlip(),
        transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.1, hue=0.1),
        transforms.RandomAffine(degrees=40, translate=None, scale=(1, 2), shear=15, fill=0),
        transforms.ToTensor(),
        transforms.Normalize((0.485, 0.456, 0.406), (0.229, 0.224, 0.225))
])
```

(c) Default settings for transformations

Figure 4: Created dataframe of Naver Place and AI Hub raw dataset and obtained default settings for transformations.

4. Methods

4.1. Setup

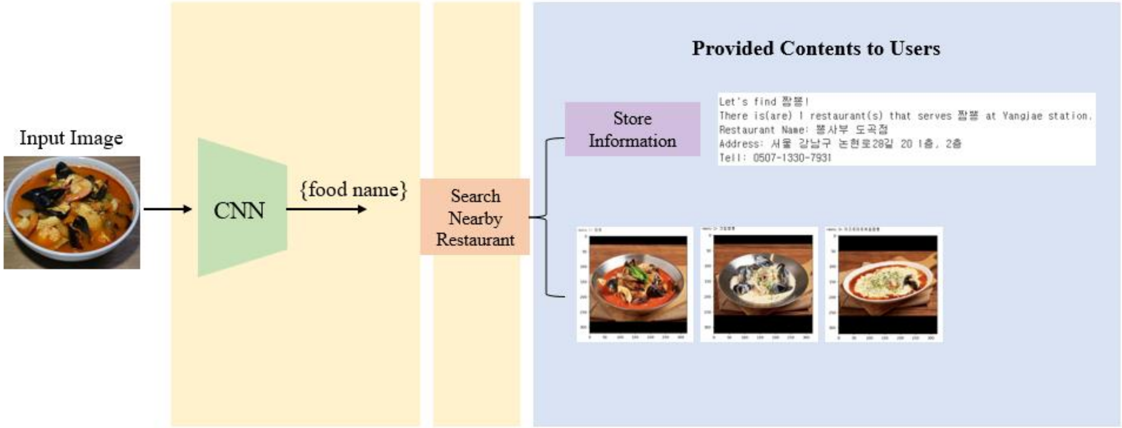
Models were run on Colab with T4 GPU and written in Pytorch, a high-level library for deep learning. Models were saved on each epoch when the validation accuracy increased. The goal of each model was minimizing the validation loss. The scheduler and optimizer used across all models were StepLR and Adam.

4.2. VGG16

In VGG16, different transformations were applied to the train and test sets and all of the layers were frozen. Also customed classifier was used, which consists of a fully connected activation layer with ReLU, a dropout layer with a 40% drop rate, and a fully connected output layer with log softmax. Max epochs for stop was 3.

4.3. ResNet50

In ResNet50, train and test sets had different transformations like VGG16. Other hyperparameters were default since it served as our base model.



```
Let's find 짬뽕!
There is(are) 1 restaurant(s) that serves 짬뽕 at Yangjae station.
Restaurant Name: 뽕사부 도곡점
Address: 서울 강남구 논현로28길 20 1층, 2층
Tell: 0507-1330-7931
```

Figure 5: Simulation of restaurant recommendation system. The input image given is 'Jjamppong'. The CNN model classifies the taken image as 'Jjamppong' and returns the predicted class. Then the result is given to nearby restaurant recommending model. The model returns a close restaurant that serves the menu with several information.

4.4. DenseNet161

Default transformations obtained from the previous stage was adopted. Also, early termination was implemented in DenseNet161. Specifically, hyperband algorithm with Bayesian inference was applied. In this case, eta, the bracket multiplier schedule, was 2 and the iteration for the first bracket was set as 2 for configurations [6].

4.5. InceptionV3

Different transformations to the train and test datasets were applied unlike DenseNet161. Concretely, various transformations, including resizing, random horizontal flipping, color jittering were used in the training set, while the test set only used normalization and resizing.

Model	Image size	Epochs	Batch	Learning rate	Loss
VGG16	224x224	10	8	2e-4	NLL
ResNet50	224x224	10	8	5e-5	CE
DenseNet161	224x224	10	5	5e-5	CE
InceptionV3	299x299	16	8	5e-5	CE

Table 1: Model settings

4.6. Nearby Restaurant Recommendation System

Users input a food image, and the trained CNN model classifies it. The recommendation system then suggests nearby restaurants based on the subway station selected by the user—Gangnam, Yangjae, or Samseong. If specific restaurants serving the identified food aren't found, the system recommends the top three highly rated restaurants in the super category or, if unavailable, the top three nearby highly rated restaurants.

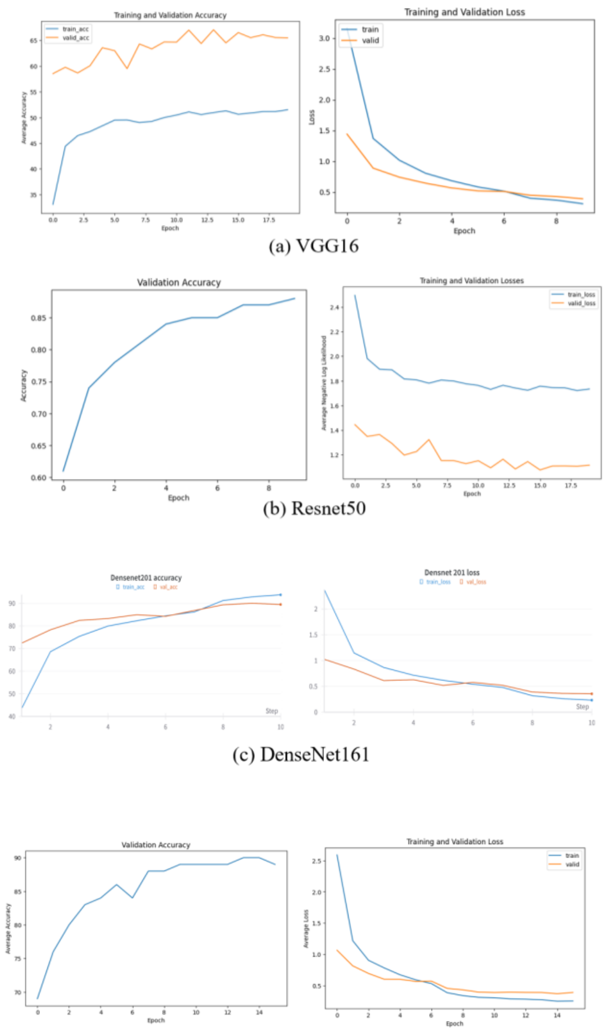
5. Results and Discussion

Evaluation metrics, including accuracy and F1 scores (micro, macro, and weighted) were utilized. Both DenseNet161 and InceptionV3 got top performances among four models, each has 90% and 90.2% as accuracy. VGG16 had the lowest score with 61.7%. It seems that the model was relatively simple for this image classification task compared to other three models, and the fine-tuned layers are relatively shallow in depth.

A single input image of ‘Jjamppong’ was given as an input to the fine-tuned InceptionV3 model, and the predicted class was given to nearby restaurant recommendation system. The model correctly classified the image of 'Jjamppong' as itself and successfully recommended related foods, as shown in Figure 4 above. Also, ‘Yangjae’ station was given as an input of recommendation model to specify the location.

Model	Accuracy	F1 score (micro)	F1 score (macro)	F1 score (weighted)
VGG16	0.617	0.617	0.609	0.626
Resnet50	0.872	0.872	0.870	0.870
DenseNet161	0.900	0.900	0.899	0.899
InceptionV3	0.902	0.902	0.900	0.900

Table 2: Model performance results



(a) VGG16

(b) Resnet50

(c) DenseNet161

(d) InceptionV3

Figure 6: Accuracy and loss graph for each model. Left graph indicates accuracy, and right one indicates loss graph of train and validation set.

6. Conclusion and Future Work

We tried various model architectures against AI Hub and Naver place image dataset classification. Models were trained from transfer learning with VGG16, ResNet50, DenseNet161, and InceptionV3. The highest performance was shown in InceptionV3 with total accuracy of 90.2%.

Our system enhances the restaurant search experience, particularly when users have only images of desired food. Due to the limited resources, there were limitations in utilizing large amount of image datasets. Therefore, by incorporating an enhanced environment, such as larger and well-preprocessed datasets, sophisticated preprocessing techniques, and detailed fine-tuning, the overall performance of the models could be further improved.

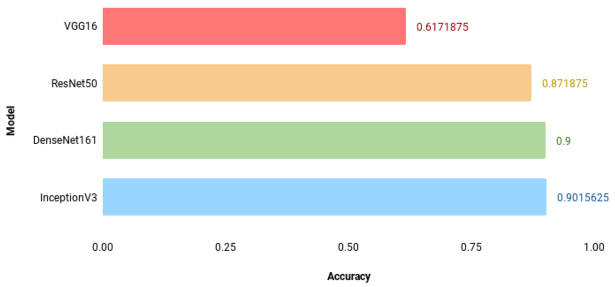


Figure 7: Summary of Accuracy of each model

7. Contributions and Code

Each teammate trained different CNN based models, including VGG16(Sujin Kim), Resnet50(Ahyun Ji), DenseNet161(Yoojin Oh), and InceptionV3(Seoyeon Ye).

References

[1] Malina Jiang. Food Image Classification with Convolutional Neural Networks, 2019.Retrieved October 13, 2023, from https://cs230.stanford.edu/projects_fall_2019/reports/26233496.pdf.

[2] Karen Simonyan and Andrew Zisserman. Very Deeep Convolutional Networks for Large-Scale Image Recognition. In ICLR, 2015.

[3] Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun. Deep Residual Learning for Image Recognition. In CVPR, 2015.

[4] Christian Szegedy and Vincent Vanhoucke and Sergey Ioffe, Jonathon Shlens and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In CVPR, 2016.

[5] Gao Huang and Zhuang Liu and Laurens van der Maaten and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In CVPR, 2017.

[6] Docs.wandb.ai. Define sweep configuration | Weights & Biases Documentation. Retrived December 15, 2023, from https://docs.wandb.ai/guides/sweeps/define-sweep-configuration

[7] Jeremy West and Dan Ventura and Sean Warnick, Sean. Spring Research Presentation: A Theoretical Foundation for Inductive Transfer, 2007. Retrieved December 15, 2023, from https://en.wikipedia.org/wiki/Transfer_learning#cite_note-1