

VGG16 을 활용한 KFood 이미지 분류

Park Sang Hee

Department of Applied Data Science

Sungkyunkwan University

parksanghee@g.skku.edu

ABSTRACT

본 연구에서는 한국을 대표하는 여러 음식들을 분류하는 이미지 분류 모델을 생성하고자 한다. 150 종의 한국 음식에 대하여 전이 학습을 이용해서 VGG16 모델을 사용한다. 가중치는 ImageNet 의 가중치를 그대로 사용하고, 모델 끝부분에 Fully Connected Layer 만 연결하여 커스텀 모델을 만든다.

KEYWORDS

Deep Learning, VGG16, KFood, Image Classification

ACM Reference format:

G. Gubbiotti, P. Malagò, S. Fin, S. Tacchi, L. Giovannini, D. Bisero, M. Madami, and G. Carloti. 1997. SIG Proceedings Paper in word Format. In *Proceedings of ACM Woodstock conference, El Paso, Texas USA, July 1997 (WOODSTOCK '97)*, 4 pages. <https://doi.org/10.1145/1234>

1 INTRODUCTION

최근 젊은이들은 다른 나라로 여행을 갈 때 여행을 떠나기 며칠 전부터 SNS, 웹서핑, 유튜브 등의 다양한 채널을 이용해 맛집, 관광지 등을 검색한다. 특히 음식의 경우, 현지에서만 먹을 수 있는 음식을 먹기 위해 관광지에 있는 음식점이 아닌 한적한 곳으로 찾아가는 노력을 기울이기도 한다. 그러나 이런 음식점들은 관광객들을 많이 받는 곳이 아니기 때문에 외국인들을 위한 메뉴판이 따로 준비가 되어있지 않다.



그림 1: 일본어로만 이루어진 현지 메뉴판

그렇기 때문에, 현지 메뉴판에서 자신이 원하는 음식을 제대로 시키지 못하거나, 혹여나 알레르기나 종교 등의 이유로 먹지 못하는 음식의 재료가 들어갈 수도 있지만, 메뉴판의 있는 정보만으로도 부족하고, 있는 메뉴판조차 현지 언어를 못하면 읽을 수 없기 때문에 특정 재료를 넣거나 빼달라는 등의 주문도 하기 매우 어렵다.

이러한 음식점에서 문제들은 위와는 정반대로 한국에서도 종종 일어나기도 한다. 특히 한국 음식 중에는 한국인들이 먹어 보아도 맛이 헛갈리는 음식들이 많고, 하나의 방법을 가지고 다양한 재료들을 써서 모양이나 생김새가 구분이 힘든 음식들이 많다.



그림 2: 외국인들이 구분하기 힘든 비빔밥과 알밥

2 Data Description & Data Pre-Processing

2.1 Data Description

이번 이미지 분류 연구에 사용되는 이미지 데이터는 AI Hub(<http://www.aihub.or.kr/>)에서 제공하는 한국 음식 이미지 데이터이다. AI Hub의 이미지 제공 목적은 미국의 ImageNet 처럼 고품질의 이미지를 무료로 개방하여 한국의 시각 지능 기술 발전의 기초를 제공하기 위함이라고 한다. 데이터를 구축한 수행기관은 한국과학기술연구원(KIST)이다.

음식의 종은 한식재단의 음식분류 및 한국인이 즐겨 먹는 음식통계를 참조하여 선정된 150 종의 음식으로 구성되며,

한식메뉴외국어표기 길라잡이(한식재단, 200 International Korean Menu Guide, 2014)를 참고하여 음식의 종류를 대분류(밥, 면, 국 등) 및 소분류를 결정하고 ID 를 부여하고 구조화하였다.

대분류	소분류	대분류	소분류
구이	갈비구이, 갈치구이, 고등어구이, 송장구이, 닭갈비, 떡갈비, 떡갈비, 불고기, 삼겹살, 갈비구이, 초가구이, 갈비구이, 훈제오리	국	계란국, 떡국/만두국, 우국, 미역국, 복쌈국, 소고기국, 시래기국, 육개장, 콩나물국
김치	김치, 떡두기, 나박김치, 무생채, 배추김치, 떡김치, 부추김치, 멸치김치, 오이소박이, 총각김치, 파김치	나물	가지볶음, 고사리나물, 미역줄기볶음, 숙주나물, 시금치나물, 애호박볶음
떡	장단	만두	만두
면	떡국수, 물냉면, 비빔냉면, 수제비, 양국수, 잔치국수, 팔면, 팔국수, 중국수, 라면, 자장면, 함흥	우침	고추된장우침, 피이코우침, 도토리묵, 감제, 도라지우침, 콩나물우침, 홍어우침
밥	김밥, 김치볶음밥, 비빔밥, 새우볶음밥, 밀밥, 잡곡밥, 주먹밥, 유부초밥	볶음	견세우볶음, 오징어볶음, 감자볶음, 고추장진미볶음, 두부김치, 떡볶이, 라볶이, 말치볶음, 소세지볶음, 어묵볶음, 제육볶음, 우무미볶음
탕	보쌈	음향류	수정과, 식혜

그림 3: AI Hub 에서 제공한 음식 분류표

이미지의 규격은 통일되어 있지 않으며, 400x400 크기부터 1000x1000 크기의 이미지로 다양하다. 포함된 이미지는 크게 27 개의 대분류로 이루어져 있으며, 150 종의 소분류로 이루어져 있다. 각각의 대분류에 포함된 소분류의 개수는 제각각이며, 한 개의 소분류된 이미지에 대하여 1,000 장의 이미지가 존재하며, 총 $150 \times 1,000 = 150,000$ 장의 이미지가 존재한다.

2.2 Data Pre-Processing

데이터 전처리는 파이썬 딥러닝 라이브러리인 tf.keras 의 ImageDataGenerator 를 활용해 데이터 제네레이터 객체를 생성한다. 제네레이터 객체를 생성할 때 사용하는 파라미터값은 다음과 같다.

파라미터명	값
rescale	1./255
rotation_range	90
width_shift_range	0.1
height_shift_range	0.1
shear_range	0.5
zoom_range	0.1
horizontal_flip	True
fill_mode	"nearest"

표 1: 데이터 제네레이터 파라미터

Rescale = 1./255 옵션은 이미지를 3 차원(가로 x 세로 x 채널)의 벡터값으로 변환할 때 값의 범위가 0~255 이기 때문에 255 로 나눠주어 모든 벡터값이 0 ~1 사이의 값을 갖도록 만들어준다.

Rotation_range=90 옵션은 지정된 각도 내에서 원본 이미지를 회전시키는 옵션이다.

width_shift_range=0.1 은 지정된 수평방향 이동 범위 내에서 임의로 원본 이미지를 이동시킨다. 수치는 전체 넓이의 비율로 나타낸다. 예를 들어 0.1 이고 전체 넓이가 100 이라면, 10 픽셀 내외로 좌우로 움직인다.

height_shift_range = 0.1 은 지정된 수직방향 이동 범위 내에서 임의로 원본 이미지를 이동시킨다. 수치는 전체 넓이의 비율로 나타낸다. 예를 들어 0.1 이고 전체 넓이가 100 이라면, 10 픽셀 내외로 위아래로 움직인다.

shear_range = 0.5 은 밀림 강도 범위내에서 임의로 원본이미지를 변형시킨다. 수치는 시계반대방향으로 밀림 강도를 라디안으로 나타냅니다. 예를 들어 0.5 이라면, 0.5 라디안내외로 시계반대방향으로 변형시킨다.

zoom_range = 0.1 은 지정된 범위 내에서 원본 이미지를 확대하거나 축소시켜 이미지를 변형시킨다. 예를 들어 0.1 이라면 원본이미지를 0.9 ~ 1.1 사이로 줌한다.

horizontal_flip = "True" 옵션은 원본이미지를 수평방향으로 뒤집기를 한다.

fill_mode = "nearest" 옵션은 이미지 사이즈를 변경할 때 부족한 부분에 대하여 어떤 방향으로 채울 지에 대한 옵션 값이다.

위와 같은 옵션값으로 이미지 데이터 제네레이터를 만든 후에 전체 이미지 사이즈를 150 x 150 픽셀의 크기로 통일시킨다.

3 Model

모델은 VGG16 을 사용하고, VGG16 의 가중치값은 ImageNet 의 가중치값을 사용한다.

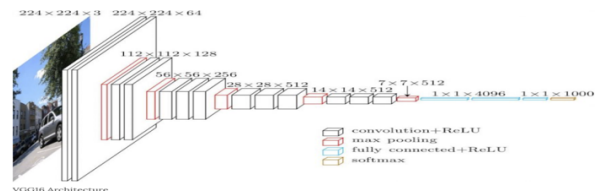


그림 4: VGG16 구조 1

Model: "vgg16"		
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 150, 150, 3)	0
block1_conv1 (Conv2D)	(None, 150, 150, 64)	1792
block1_conv2 (Conv2D)	(None, 150, 150, 64)	36928
block1_pool (MaxPooling2D)	(None, 75, 75, 64)	0
block2_conv1 (Conv2D)	(None, 75, 75, 128)	73856
block2_conv2 (Conv2D)	(None, 75, 75, 128)	147584
block2_pool (MaxPooling2D)	(None, 37, 37, 128)	0
block3_conv1 (Conv2D)	(None, 37, 37, 256)	295168
block3_conv2 (Conv2D)	(None, 37, 37, 256)	590080
block3_conv3 (Conv2D)	(None, 37, 37, 256)	590080
block3_pool (MaxPooling2D)	(None, 18, 18, 256)	0
block4_conv1 (Conv2D)	(None, 18, 18, 512)	1180160
block4_conv2 (Conv2D)	(None, 18, 18, 512)	2359808
block4_conv3 (Conv2D)	(None, 18, 18, 512)	2359808
block4_pool (MaxPooling2D)	(None, 9, 9, 512)	0
block5_conv1 (Conv2D)	(None, 9, 9, 512)	2359808
block5_conv2 (Conv2D)	(None, 9, 9, 512)	2359808
block5_conv3 (Conv2D)	(None, 9, 9, 512)	2359808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0
Total params: 14,714,688		
Trainable params: 14,454,528		
Non-trainable params: 260,160		

그림 5: VGG16 구조 2

VGG16 은 총 파라미터의 개수가 14,714,688 개이다. 그러나 이 모델을 처음부터 학습시키게 되면 많은 시간이 걸리기 때문에 마지막 레이어에 Flatten Layer 와 Dense Layer 를 추가한다.

VGG16 의 전체 2 번째 Block 까지를 사용하고, 추가한 Flatten 과 Dense 2 개의 Layer 에 대해서만 사용하는 커스텀 모델을 만들고, VGG16 의 Block 에 대해서는 가중치를 고정시킨다.

Model: "model_1"		
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 150, 150, 3)	0
block1_conv1 (Conv2D)	(None, 150, 150, 64)	1792
block1_conv2 (Conv2D)	(None, 150, 150, 64)	36928
block1_pool (MaxPooling2D)	(None, 75, 75, 64)	0
block2_conv1 (Conv2D)	(None, 75, 75, 128)	73856
block2_conv2 (Conv2D)	(None, 75, 75, 128)	147584
block2_pool (MaxPooling2D)	(None, 37, 37, 128)	0
flatten_1 (Flatten)	(None, 175232)	0
dense_1 (Dense)	(None, 27)	4731291
Total params: 4,991,451		
Trainable params: 4,731,291		
Non-trainable params: 260,160		

그림 6: Custom Model

커스텀 모델의 경우 파라미터의 수가 4,991,451 개로 VGG16 을 그래도 사용했을 때 보다 1/3 가량 파라미터의 수가 줄어들었다. Loss 함수는 Categorical_crossentropy 를 사용하고, 옵티마이저는 Adam Optimizer 를 사용하였다.

4 Result

실험 결과는 에폭크를 더 진행하면 결과가 더 좋은 것이라 예상되지만, 실험 환경 등의 이유와 현재 상태의 정확도도 충분한 값이기 때문에 더이상 학습하지 않았다.

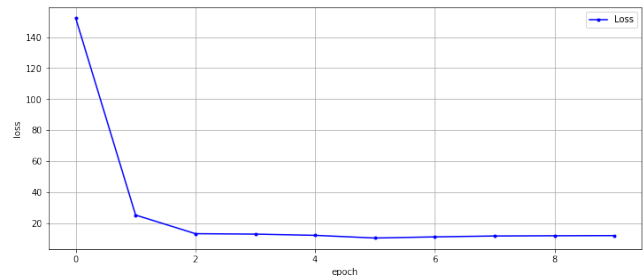


그림 6: epoch - loss 그래프

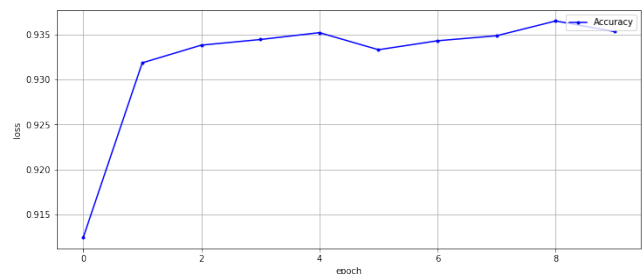


그림 7: epoch-acc 그래프

5 Summary

정확도는 93.5% 이다. 물론 이 값은 전체 150,000 개의 데이터에 대해 어느 정도 전처리 변환을 시키고, 전부 학습시켰을 때의 결과이기 때문에 Train 과 Test 로 데이터셋을 분리하였을 때의 결과보다는 부정확할 수 있다.

추후 연구에는 데이터를 분리시키고, VGG16 구조를 일부만 사용하는 것이 아닌, 전체 구조를 사용했을 때의 모델에 대한 연구 예정이다. 또한 VGG16 뿐만 아니라, EfficientNet, Xception 등의 최신 알고리즘 등을 사용할 예정이다.

또한 완성된 모델을 이용하여 이미지에 대하여 레이블만 제공하는 것이 아니라 음식에 대한 정보를 제공하는 서비스까지 연결할 예정이다. 식품에 대한 상세 정보 및 영양 성분은 식품의약품안전처 식품영양성분 데이터베이스의 OpenAPI 를 활용하여 DB 를 구축한다. 이미지가 주어지면 150 개의 음식에 대하여 매핑을 하고, 그 매핑된 레이블에 대하여 주요 재료와 성분들을 표시하는 서비스를 구축한다.

Reference

- [1] 이한수, 김종근, 유정원, 정영상, 김성신 (2018). 전이학습 기반의 합성곱 신경망을 이용한 다중클래스 분류에 관한 연구. 한국지능시스템학회 논문지, 28(6), 531-537.
- [2] Hokuto Kagaya, Kiyoharu Aizawa, Makoto Ogawa. (2014). Food Detection and Recognition Using Convolutional Neural Network. MM '14: Proceedings of the 22nd ACM international conference on Multimedia. 1085-1088.
- [3] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen. (2019). MobileNetV2: Inverted Residuals and Linear Bottlenecks. The IEEE Conference on Computer Vision and Pattern Recognition. 4510-4520.
- [4] Ashutosh Singla, Lin Yuan, Touradj Ebrahimi. (2016). Food/Non-food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model. MADiMa '16: Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management. 3-10.
- [5] Paritosh Pandey, Akella Deepthi, Bappaditya Mandal, N. B. Puan. (2017). FoodNet: Recognizing Foods Using Ensemble of Deep Networks. IEEE Signal Processing Letters. 1758 - 1762.