

음식 이미지 분류기

전체 개요

데이터 전처리

모델 학습

결과

서론 / 필요성

배경



현대 사회에서 건강한 식단 관리의 중요성이 증가하고 있으며, 많은 사람들이 섭취하는 음식의 종류와 영양소를 알고 싶어함

필요성



음식 이미지를 자동으로 분류하여 영양 성분을 예측하면 개인 맞춤형 영양 관리가 가능하며, 이는 건강 증진과 식습관 개선에 기여할 수 있음

관련 연구 / 내용

관련 연구

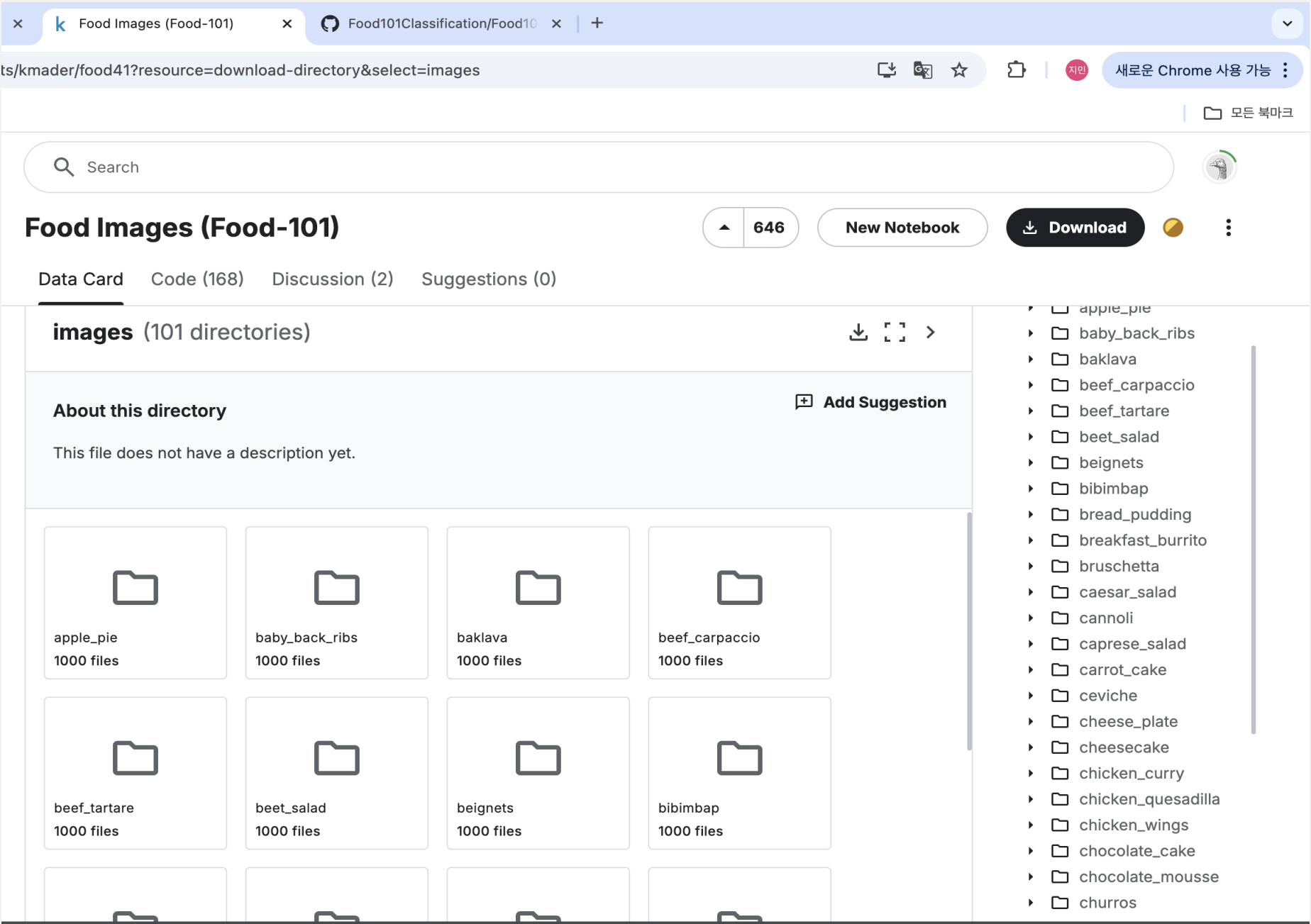
Food-101 데이터셋을
사용한 음식 분류 연구

내용

이데이터셋을 활용한 연구들은
CNN(Convolutional Neural Network)을
주로 사용하여 높은 성능을 달성

목표

딥러닝을 사용해 음식 이미지를
보고 음식 종류를 인식하며, 영양
정보를 제공



프로젝트 목표 및 개요

데이터 수집



데이터 수집, 전처리

모델 학습



모델 학습 및 평가

결과 분석



최종 결과 및 분석

데이터셋

Data Explorer

Version 5 (5.82 GB)

- ▶ images
- ▶ meta
 - food_c101_n1000_r384x...
 - food_c101_n10099_r32x...
 - food_c101_n10099_r32x...
 - food_c101_n10099_r64x...
 - food_c101_n10099_r64x...
 - food_test_c101_n1000_r...
 - food_test_c101_n1000_r...
 - food_test_c101_n1000_r...
 - food_test_c101_n1000_r...
 - food_test_c101_n1000_r...
 - food_test_c101_n1000_r...

train.txt (1.47 MB)



This preview is truncated due to the large file size. Create a Notebook or download this file to see the full content.

Download

Create Notebook

apple_pie/1005649
apple_pie/1014775
apple_pie/1026328
apple_pie/1028787
apple_pie/1043283
apple_pie/1050519
apple_pie/1057749
apple_pie/1057810
apple_pie/1072416
apple_pie/1074856
apple_pie/1074942
apple_pie/1076891
apple_pie/1077610
apple_pie/1077964
apple_pie/1088809
apple_pie/1097378
apple_pie/1103795
apple_pie/1109597
apple_pie/1111062
apple_pie/1112300
apple_pie/1112838
apple_pie/1121884
apple_pie/112378
apple_pie/1133267

Data Explorer

Version 5 (5.82 GB)

- ▶ images
- ▼ meta
 - meta
 - classes.txt
 - labels.txt
 - test.json
 - test.txt
 - train.json
 - train.txt
 - food_c101_n1000_r384x...
 - food_c101_n10099_r32x...
 - food_c101_n10099_r32x...
 - food_c101_n10099_r64x...
 - food_c101_n10099_r64x...
 - food_test_c101_n1000_r...
 - food_test_c101_n1000_r...
 - food_test_c101_n1000_r...
 - food_test_c101_n1000_r...
 - food_test_c101_n1000_r...
 - food_test_c101_n1000_r...

데이터 셋 종류: Food-101

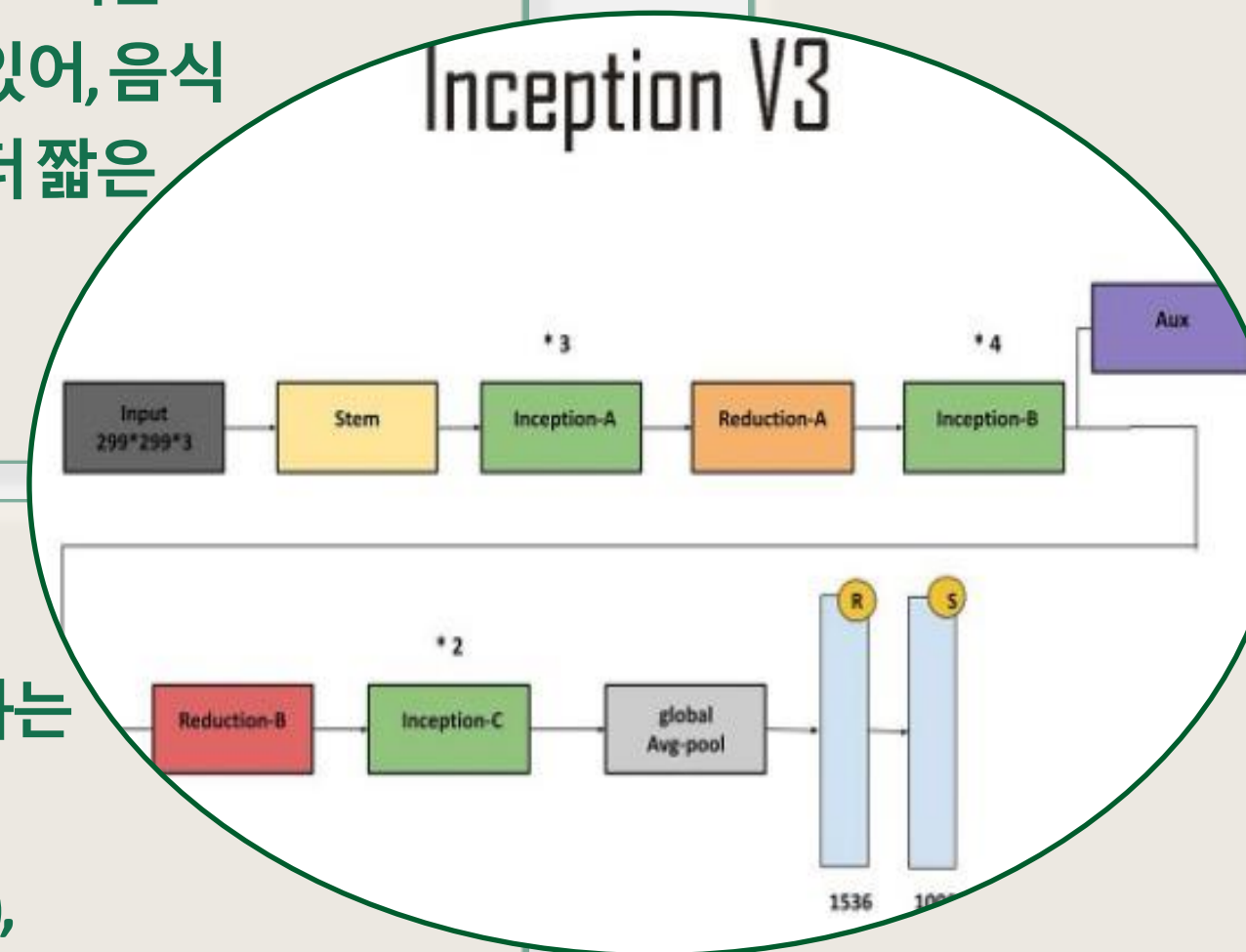
클래스 수: 101개 음식 종류

이미지 개수: 총 101,000장

인공지능 모델 선택 및 구조

Inception V3는 구글에서 개발한 이미지 분류를 위한 딥러닝 모델로, 매우 높은 정확도를 보여줌
대규모 데이터셋에서 미리 학습이 되어 있어, 음식 데이터셋에 맞게 전이 학습을 적용하면 더 짧은 시간 안에 높은 성능을 낼 수 있음
-> 음식 종류별 특성을 인식하는데 유리

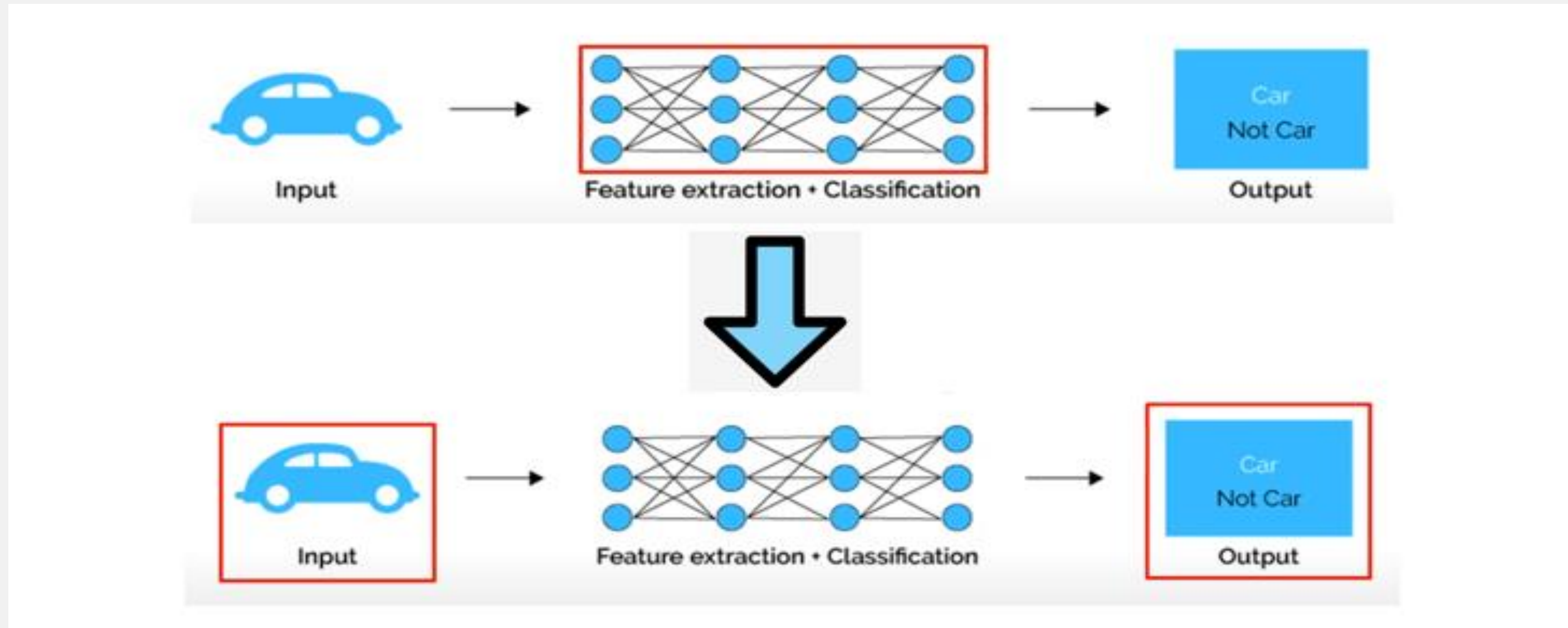
Inception V3 모델은 특징을 추출하는 레이어들과, 마지막으로 이미지를 분류하는 레이어들로 구성
주요 레이어: Convolutional Layer(특징 추출), Pooling Layer(차원 축소 및 특징 강화), Fully Connected Layer(최종 분류)



입력과 출력 형태:
224 x 224 이미지 입력 ->
여러 Conv, Pooling, Fully Layers ->
101개 음식 클래스 출력

모델 구조 전체 흐름도:
입력 이미지 -> 특징 추출 레이어(Conv, Pooling) -> Fully Connected 레이어 -> Softmax로 최종 분류

전이 학습(Transfer Learning)



대규모 데이터셋으로 미리 학습된 모델을 가져와, 새로운 데이터셋(음식 이미지)으로 다시 학습시키는 방법
이를 통해 데이터가 적더라도 빠르게 높은 정확도를 얻을 수 있음

Inception V3 모델의 전이 학습 적용 과정

기존 모델의 기본 레이어를 고정하고, 음식 분류를 위한 새로운 레이어를 추가
최종 레이어만 학습시키는 방식으로 새로운 데이터셋에 빠르게 적응하도록 함

```
27
28 # 모델 생성 함수
29 def create_model(): 1 usage
30     base_model = InceptionV3(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
31     x = base_model.output
32     x = GlobalAveragePooling2D()(x)
33     x = Dense(1024, activation='relu')(x)
34     predictions = Dense(train_generator.num_classes, activation='softmax')(x)
35     model = Model(inputs=base_model.input, outputs=predictions)
36     #사전 학습된 기본 레이어는 고정
37     for layer in base_model.layers:
38         layer.trainable = False
39
40     model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
41     return model
42
```


데이터 전처리의 목표 및 과정

데이터 전처리의 필요성 및 목표

이미지 크기와 색상 등을 표준화하여 모델이 더 쉽게 학습할 수 있도록 도움

데이터 증강을 통해 적은 데이터를 다양한 형태로 변형하여 더 많은 데이터를 학습하는 효과를 얻음

데이터 전처리 과정-이미지 크기 조정

모든 이미지 크기를 224 X 224 크기로 맞춰, 모델 입력에 적합하도록 함.

데이터 전처리 과정-데이터 증강

회전-음식 이미지를 20도 범위에서 무작위로 회전하여 다양한 각도 학습

수평 이동, 수직 이동-20% 범위 내에서 이동하여 중심이 다소 변한 이미지 학습

확대/축소-20% 확대 또는 축소 || 수평 뒤집기-이미지 좌우 반전을 통해 양방향 학습

데이터 전처리의 목표 및 과정

```
# 이미지 증강 설정
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True
)
```

데이터 전처리를 위한
ImageDataGenerator 코드

- # 픽셀 값을 0-1 사이로 조정
- # 회전
- # 수평 이동
- # 수직 이동
- # 확대 / 축소
- # 수평 반전

학습 데이터 준비 및 학습 과정

학습 데이터 준비 및 설정

```
# 이미지 데이터 로딩
train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(224, 224), # 이미지 크기
    batch_size=32,
    class_mode='categorical' # 다중 클래스 분류
)
```

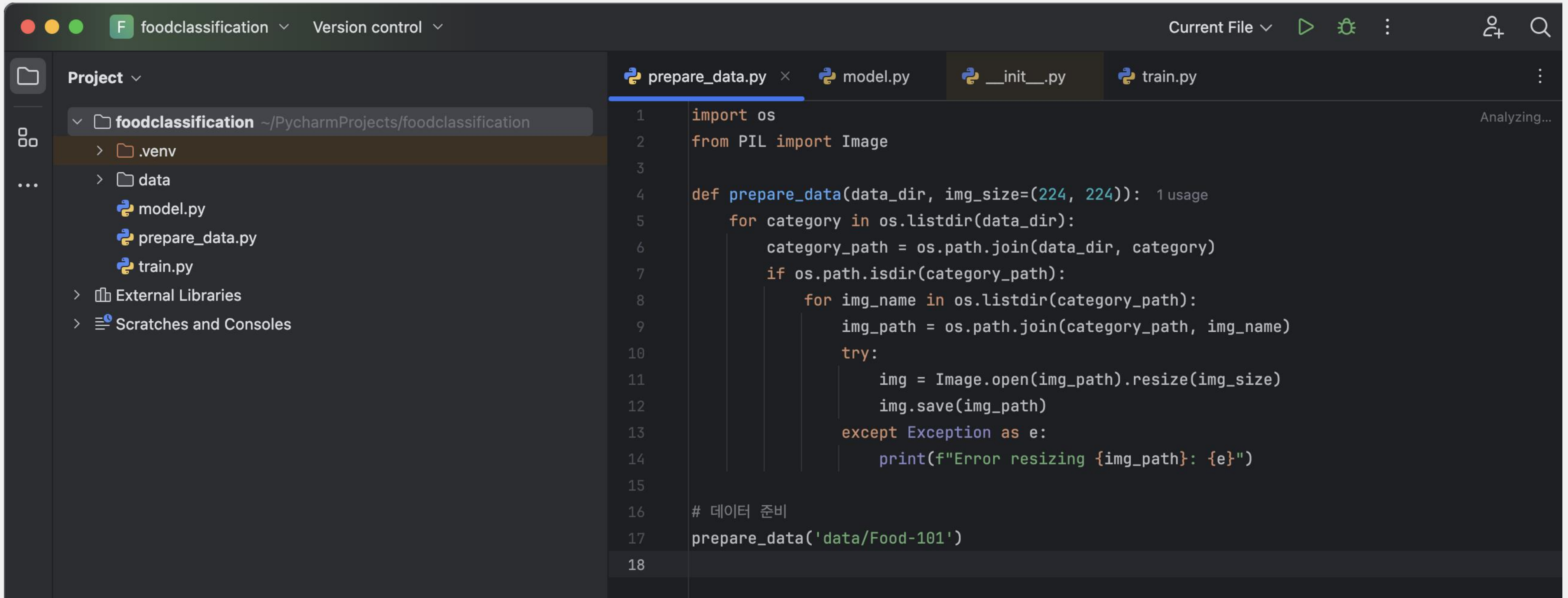
flow_from_directory 함수로 이미지 데이터를 생성하고 배치 단위로 모델에 공급하여 학습
배치 크기 32, 에포크 10으로 학습 설정

모델 학습 과정 설명

```
# 모델 학습
epochs = 10
model.fit(train_generator, epochs=epochs, steps_per_epoch=len(train_generator))
```

```
Epoch 1/10
3157/3157 ————— 2077s 657ms/step - accuracy: 0.2995 - loss: 2.9593
Epoch 2/10
805/3157 ————— 25:51 659ms/step - accuracy: 0.4199 - loss: 2.2908
```

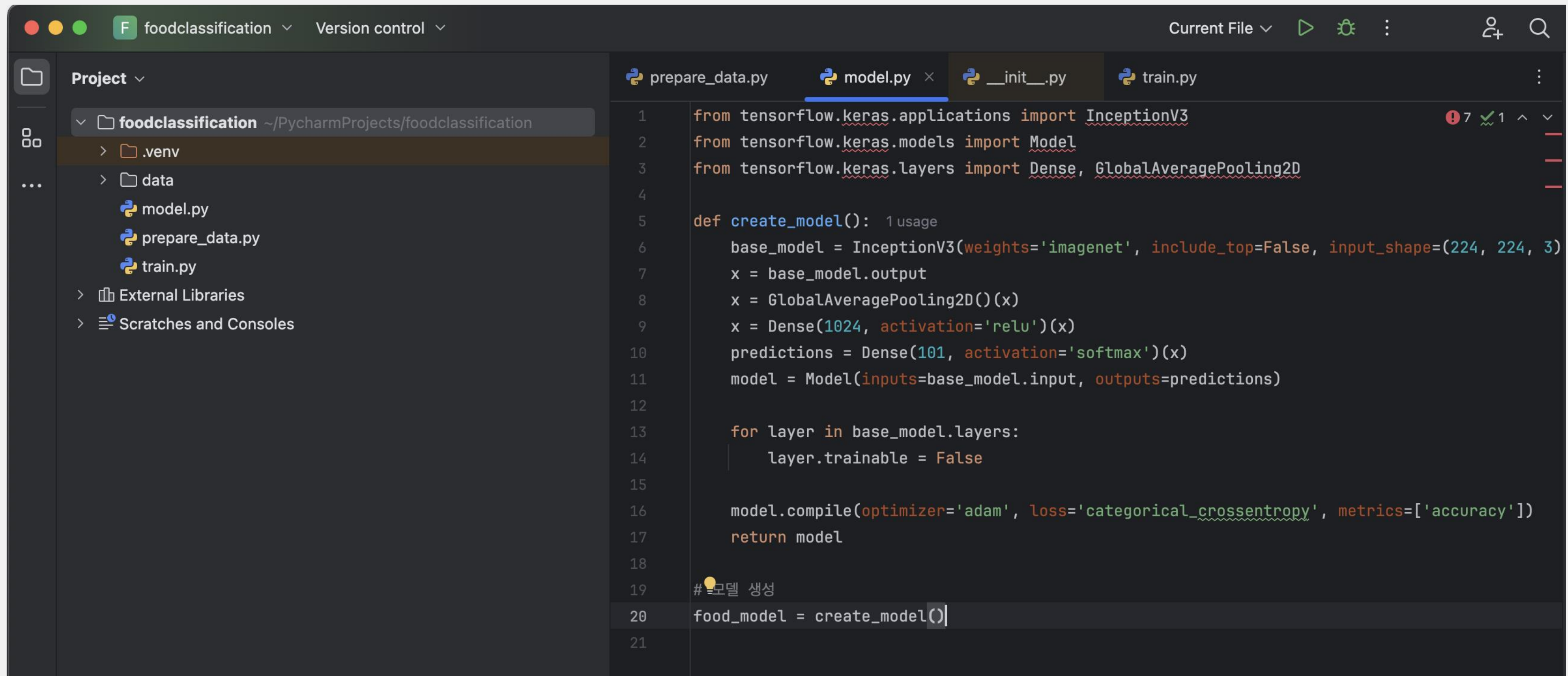
학습 목표는 손실(loss)을 줄이고, 정확도(accuracy)를 높이는 것
학습이 진행되면서 손실 값은 낮아지고, 정확도는 높아지도록
모델을 최적화함 -> model.fit 함수 사용



Project foodclassification ~ /PycharmProjects/foodclassification

- .venv
- data
 - model.py
 - prepare_data.py
 - train.py
- External Libraries
- Scratches and Consoles

```
1 import os
2 from PIL import Image
3
4 def prepare_data(data_dir, img_size=(224, 224)): 1 usage
5     for category in os.listdir(data_dir):
6         category_path = os.path.join(data_dir, category)
7         if os.path.isdir(category_path):
8             for img_name in os.listdir(category_path):
9                 img_path = os.path.join(category_path, img_name)
10                try:
11                    img = Image.open(img_path).resize(img_size)
12                    img.save(img_path)
13                except Exception as e:
14                    print(f"Error resizing {img_path}: {e}")
15
16 # 데이터 준비
17 prepare_data('data/Food-101')
```



```
1 from tensorflow.keras.applications import InceptionV3
2 from tensorflow.keras.models import Model
3 from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
4
5 def create_model(): 1 usage
6     base_model = InceptionV3(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
7     x = base_model.output
8     x = GlobalAveragePooling2D()(x)
9     x = Dense(1024, activation='relu')(x)
10    predictions = Dense(101, activation='softmax')(x)
11    model = Model(inputs=base_model.input, outputs=predictions)
12
13    for layer in base_model.layers:
14        layer.trainable = False
15
16    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
17    return model
18
19 # 모델 생성
20 food_model = create_model()
21
```


ersion control ▾

Current File ▾

prepare_data.py model.py __init__.py train.py ×

1 from tensorflow.keras.preprocessing.image import ImageDataGenerator

2 from tensorflow.keras.applications import InceptionV3

3 from tensorflow.keras.models import Model

4 from tensorflow.keras.layers import Dense, GlobalAveragePooling2D

5 from tensorflow.keras.callbacks import ModelCheckpoint

6 import os

7

8 # 데이터 경로 설정

9 train_data_dir = 'data/images'

10

11 # 이미지 증강 설정

12 train_datagen = ImageDataGenerator(

13 rescale=1./255,

14 rotation_range=20,

15 width_shift_range=0.2,

16 height_shift_range=0.2,

17 zoom_range=0.2,

18 horizontal_flip=True

19)

20

21 # 이미지 데이터 로딩

22 train_generator = train_datagen.flow_from_directory(

23 train_data_dir,

24 target_size=(224, 224),

25 batch_size=32,

26 class_mode='categorical'

27)

28

29 # 모델 생성 함수

30 def create_model(): 1 usage

31 base_model = InceptionV3(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

32 x = base_model.output

33 x = GlobalAveragePooling2D()(x)

34 x = Dense(1024, activation='relu')(x)

53:1 LF UTF-8 4 spaces Python 3.12 (foodclassification)

Current File ▾

prepare_data.py model.py __init__.py train.py ×

28

29 # 모델 생성 함수

30 def create_model(): 1 usage

31 base_model = InceptionV3(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

32 x = base_model.output

33 x = GlobalAveragePooling2D()(x)

34 x = Dense(1024, activation='relu')(x)

35 predictions = Dense(train_generator.num_classes, activation='softmax')(x)

36 model = Model(inputs=base_model.input, outputs=predictions)

37 #사전 학습된 기본 레이어는 고정

38 for layer in base_model.layers:

39 layer.trainable = False

40

41 model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

42 return model

43

44 # 모델 생성

45 model = create_model()

46

47 # 모델 학습

48 epochs = 10

49 model.fit(train_generator, epochs=epochs, steps_per_epoch=len(train_generator))

50

51 # 모델 저장

52 model.save('food_classification_model.h5')

53

54

55 # 체크포인트 설정: 'accuracy'가 최고일 때 모델을 저장

56 checkpoint = ModelCheckpoint(

57 'model_checkpoint.h5', # 저장할 파일명

58 monitor='accuracy', # 성능 기준 지표

59 save_best_only=True, # 가장 좋은 성능일 때만 저장

60 mode='max' # 모니터링 지표가 클수록 좋은 경우 'max'

61)

61:2 LF UTF-8 4 spaces Python 3.12 (foodclassification)