KT GenieLabs

Dev-Challenge 2022

Task 3. Food Image Classification Model Design

JazzIsHorse

Contents

- Project Members
- Project Plan
- Model details

Project Members

JazzIsHorse

- Yoo Seung-hun
- Depart. of Physio, Dong-Eui University

- Kim Jeong-do
- Depart. of Philosophy, Pusan National University

- Lee Sang-min
- Depart. of Ecological engineering, Pukyong National University

Task

과제 3 음식 이미지 분류 모델 설계

	1차 심사 (예선) : ~9월 21일(수) 13시	2차 심사 (본선) : 9월 23일(금)~9월 29일(목)
개발 환경	자체 개발환경 사용 예) Google Colab	KT GenieMars GPU제공 (NGC PyTorch Docker기반)
학습모델 또는 데이터셋	KT 제공 - 제공 데이터량 50종 X 200장	KT 제공 - 제공 데이터량 • Train: 100종X 200장 • Validation: 100종X 40장 • Test셋은 비공개 (평가에만 사용)
산출물	- 문제해결방법을 기술한 PPT ※ 별도 테스트가 없기 때문에 주어진 데이터에 대한 Accuracy 포함 (5 Cross Validation결과)	- 문제해결방법을 기술한 PPT - 학습 및 평가 때 사용했던 스크립트 (도커환경 구성 시 사용한 requirements.txt 파일 포함)
평가방법	1. 평가방법 - 문제해결방식(100%) 2. 세부내용 - 문제해결방식 • 모델 구현 방법의 창의성 • 모델 학습 방법의 효율성 • 별도 검증/테스트 셋이 주어지지 않기 때문에 주어진 데이터에 대해 5 Cross Validation평균 Accuracy 제출 • 자체 개발환경 리소스 제한으로 인한 이미지 사이즈 변경이 있을 경우 (예> 256x256 -> 32x32) PPT에 해당 내용을 기술 할 것	1. 평가방법 - 문제해결방식(40%), Accuracy(60%) 2. 세부내용 - 문제해결방식 • 모델 구현 방법의 창의성 • 모델 학습 방법의 효율성 • 모델 크기에는 제한이 없으나 주어진 인프라 리소스내에서 학습이 가능해야 함 - Accuracy (Top 1) • 제공된 validation데이터는 모델 학습 시 포함 금지 • 최종 평가 시 동일 클래스의 공개되지 않은 데이터 기반 평가

Task

주의사항

사전 학습된 모델 (ImageNet pre-trained model 및 해당 플랫폼 외부에서 학습된 모델 등) 사용 불가하고 제공된 학습 데이터(KT 제공: 100종의 음식물 데이터) 이외의 추가 데이터를 이용한 모델 학습 시 탈락

제공된 인프라 환경에서 실험이 재현 가능해야 하며 학습 및 평가 스크립트 실행 시 결과 동일해야 함

- 평가 시 학습 결과를 재현 할 수 있도록 학습 및 평가 스크립트 제출 할 것 (하이퍼 파라미터 (ex> learning rate, warm-up 등등) 포함 필수)
- 모델 크기의 제한은 없으나, 복수의 분류기를 사용하는 ensemble방법 금지

Focus on task

Creative thinking for problem solving

- way to reduce the background.
- Image emphasize featuring

Submission

- PPT
- Accuracy
- 5 cross validation

Data checking

Project plan

Image preprocessing

- Image resize due to resource limitation 256 to 64.
- Keep 3 channels bcs, especially, food picture.

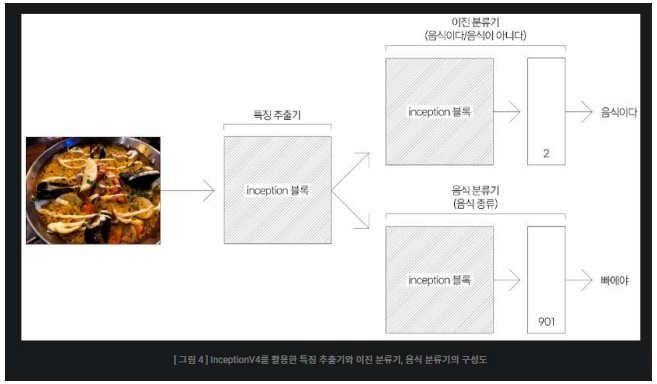
Data Augmentation

- Using Image “ImageDataGenerator” from Keras.
- rotation, flip etc.

Project plan

Decision Tree

- Among the limitations presented in the assignment, the ensemble method was limited, so it was attempted to be connected in a serial form.
- However, This way already tried by Kakao Enterprise.
- We could learn a lot from this website.
  - https://tech.kakaoenterprise.com/84#footnote_link_84_6

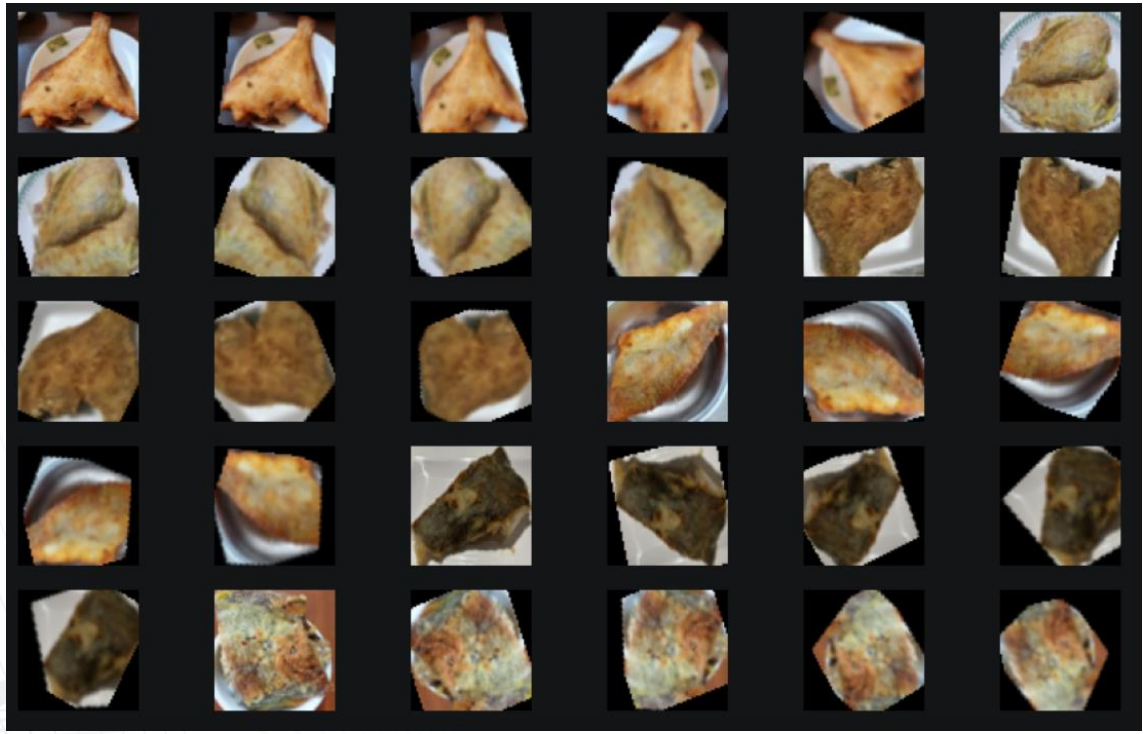


[그림 4] InceptionV4를 활용한 특징 추출기와 이진 분류기, 음식 분류기의 구성도

Project plan

Data Augmentation

- Using Image "ImageDataGenerator" from Keras
- rotation, flip etc.



Project plan

Data loader

- Even we reduced image size, we still still running out of resources.
- So reference here.
  - https://minimin2.tistory.com/100

Category array mean Square for

Model details

model compile summary

- Optimizer = Adam (Learning rate = 0.001)
- Activation = LeakyReLU
- Conv2d layer
- MaxPooling 2D
- Flatten
- Dense(128, 64)
- Output nodes = 50

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_1 (Conv2D)           (None, 64, 64, 64)        1792

 activation_1 (Activation)   (None, 64, 64, 64)        0

 max_pooling2d_1 (MaxPooling  (None, 32, 32, 64)       0
 2D)

 dropout_5 (Dropout)         (None, 32, 32, 64)        0

 flatten_2 (Flatten)         (None, 65536)             0

 dense_6 (Dense)             (None, 128)               8388736

 dropout_6 (Dropout)         (None, 128)               0

 dense_7 (Dense)             (None, 64)                8256

 dropout_7 (Dropout)         (None, 64)                0

 dense_8 (Dense)             (None, 50)                3250

=================================================================
Total params: 8,402,034
Trainable params: 8,402,034
Non-trainable params: 0
_________________________________________________________________
```

Model details

Model.fit

```
    model_history.append(history)
```

```
y: 0.9452
Epoch 47/1000
157/157 [==============================] - 9s 55ms/step - loss: 0.4803 - accuracy: 0.8462 - val_loss: 0.2425 - val_accurac
y: 0.9343
Epoch 48/1000
157/157 [==============================] - 9s 55ms/step - loss: 0.4804 - accuracy: 0.8468 - val_loss: 0.2373 - val_accurac
y: 0.9367
Epoch 49/1000
157/157 [==============================] - 9s 56ms/step - loss: 0.4905 - accuracy: 0.8443 - val_loss: 0.2445 - val_accurac
y: 0.9351
Epoch 50/1000
157/157 [==============================] - 9s 57ms/step - loss: 0.4888 - accuracy: 0.8464 - val_loss: 0.2754 - val_accurac
y: 0.9195
Epoch 51/1000
157/157 [==============================] - 9s 55ms/step - loss: 0.4798 - accuracy: 0.8474 - val_loss: 0.2594 - val_accurac
y: 0.9260
Epoch 52/1000
157/157 [==============================] - 9s 55ms/step - loss: 0.4835 - accuracy: 0.8488 - val_loss: 0.2716 - val_accurac
y: 0.9238
```

```
average = 0
for i in range(0,5):
    average += model_history[i].history['accuracy'][-1]

print("정확도:", average/5)
```

```
정확도: 0.7899499893188476
```

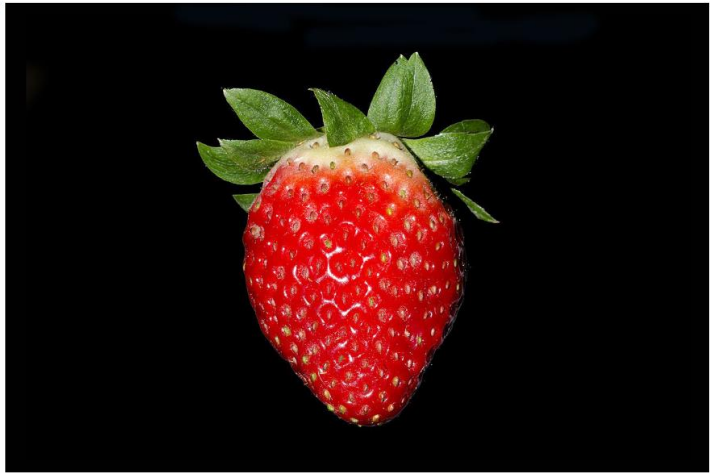
Prediction_Correct

```
# 답을 출력한다.
print(f'₩n predict : {categories[np.argmax(pred)]} (Category no.{np.argmax(pred)}) ₩n')
test_org
```

predict : 비빔밥 (Category no.44)

```
# 답을 출력한다.
print(f'₩n predict : {categories[np.argmax(pred)]} (Category no.{np.argmax(pred)}) ₩n')
test_org
```

predict : 딸기 (Category no.30)



```
# 답을 출력한다.
print(f'₩n predict : {categories[np.argmax(pred)]} (Category no.{np.argmax(pred)}) ₩n')
test_org
```

predict : 고구마 (Category no.4)

Prediction_Wrong

```
# 답을 출력한다.
print(f'₩n predict : {categories[np.argmax(pred)]} (Category no.{np.argmax(pred)}) ₩n')
test_org
```

```
predict : 달걀볶음밥 (Category no.18)
```

```
# 답을 출력한다.
print(f'₩n predict : {categories[np.argmax(pred)]} (Category no.{np.argmax(pred)}) ₩n')
test_org
```

```
predict : 떡국 (Category no.32)
```

Thank you