AI 출발프로젝트2_4

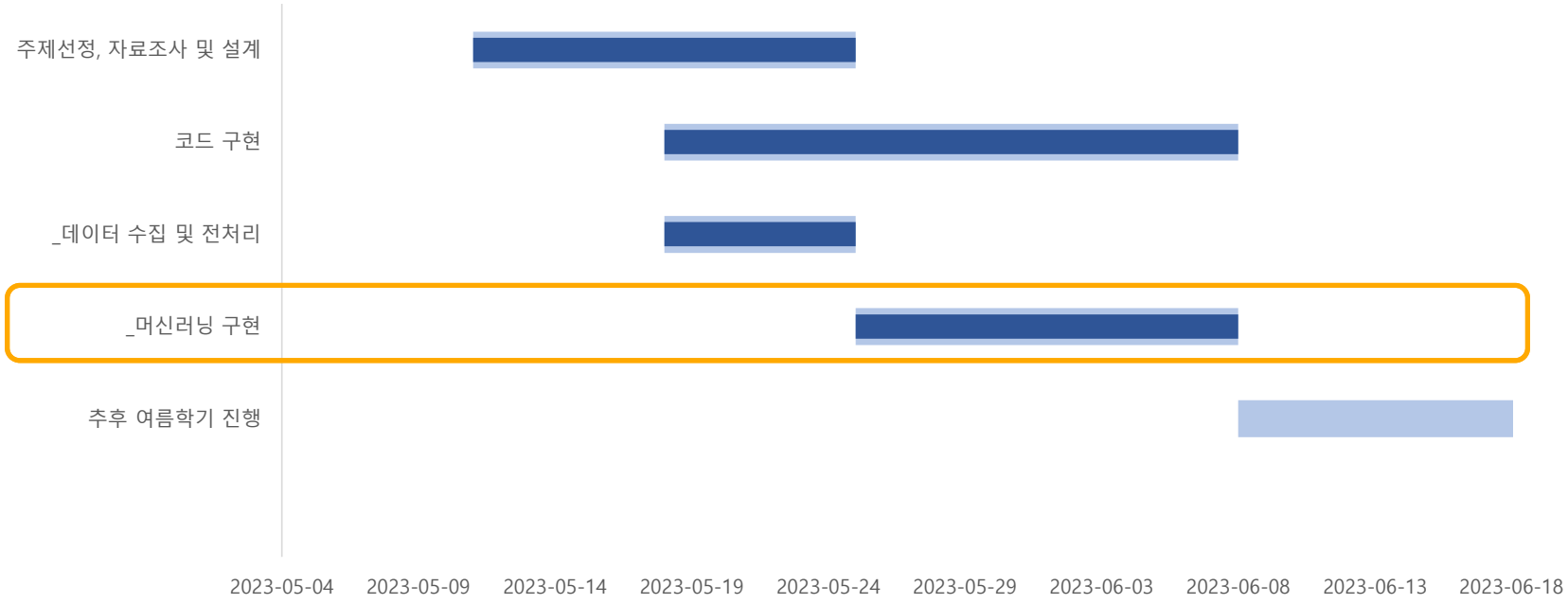
2017211836 김지훈, 2020213246 조성은, 2020215727 신재욱, 2020213297 오아연

Contents

Contents

일정 간트 차트

출발 프로젝트_AI	시작일	작업 일수	종료날짜	진행률
주제선정, 자료조사 및 설계	2023-05-11	14	2023-05-25	100%
코드 구현	2023-05-18	21	2023-06-08	100%
_데이터 수집 및 전처리	2023-05-18	7	2023-05-25	100%
_머신러닝 구현	2023-05-25	14	2023-06-08	100%
추후 여름학기 진행	2023-06-08	???		0%



Contents

이번주 예정 진행 사항

코드 구현

1. 모델 완성 및 테스트

Contents

모델피팅에 걸리는 과도한 시간

```
# Train the model with data from 3 classes
n_classes = 101
epochs = 10
nb_train_samples = train_files
nb_validation_samples = test_files

history, class_map_101 = train_model(n_classes,epochs, nb_train_samples,nb_validation_samples
print(class_map_101)
```

```
Found 75750 images belonging to 101 classes.
Found 25250 images belonging to 101 classes.
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_
87910968/87910968 [==============================] - 1s 0us/step
WARNING:absl:`lr` is deprecated in Keras optimizer, please use `learning_rate` or use the lega
<ipython-input-9-3963c06562db>:76: UserWarning: `Model.fit_generator` is deprecated and will b
  history = model.fit_generator(train_generator, #훈련 데이터 셋
Epoch 1/10
   3/1515 [..............................] - ETA: 22:15:34 - loss: 5.2506 - accuracy: 0.0133
```

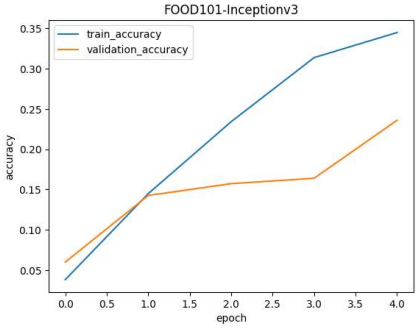
Contents

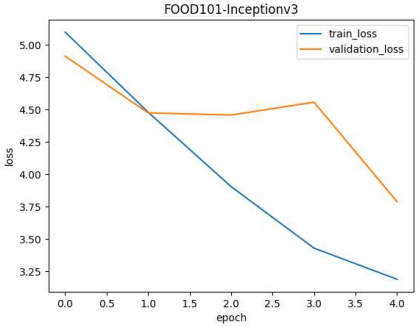
해결 시도

```
img_width, img_height = 299, 299
train_data_dir = 'food-101/train_mini'
validation_data_dir = 'food-101/test_mini'
batch_size = 16
```

```
img_width, img_height = 299, 299
train_data_dir = 'food-101/train_mini'
validation_data_dir = 'food-101/test_mini'
batch_size = 50
```

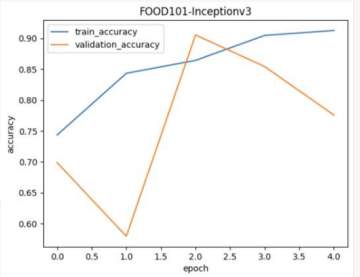
```
img_width, img_height = 299, 299
train_data_dir = 'food-101/train_mini'
validation_data_dir = 'food-101/test_mini'
batch_size = 100
```

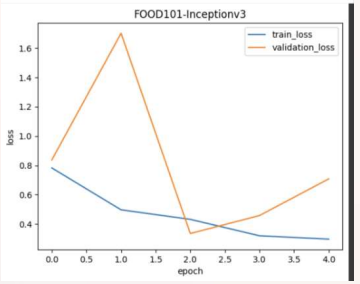





```
train_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input)

test_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
```





Contents

모델 피팅

```
n_classes = 3
epochs = 30
nb_train_samples = train_files
nb_validation_samples = test_files

history, class_map_3 = train_model(n_classes,epochs, nb_train_samples,nb_validation_samples)
print(class_map_3)
```

```
Found 2250 images belonging to 3 classes.
Found 750 images belonging to 3 classes.
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering_tf
87910968/87910968 [==============================] - 1s 0us/step
/usr/local/lib/python3.10/dist-packages/keras/optimizers/legacy/gradient_descent.py:114: UserWarning: The `lr` argument is deprecated,
  super().__init__(name, **kwargs)
<ipython-input-16-1a074316ca40>:76: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please us
  history = model.fit_generator(train_generator, #훈련 데이터 셋
Epoch 1/30
140/140 [==============================] - ETA: 0s - loss: 1.0618 - accuracy: 0.4785
Epoch 1: val_loss improved from inf to 0.86844, saving model to bestmodel_3class.hdf5
140/140 [==============================] - 94s 520ms/step - loss: 1.0618 - accuracy: 0.4785 - val_loss: 0.8684 - val_accuracy: 0.7038
Epoch 2/30
140/140 [==============================] - ETA: 0s - loss: 0.8320 - accuracy: 0.6858
Epoch 2: val_loss improved from 0.86844 to 0.68143, saving model to bestmodel_3class.hdf5
140/140 [==============================] - 66s 473ms/step - loss: 0.8320 - accuracy: 0.6858 - val_loss: 0.6814 - val_accuracy: 0.7908
Epoch 3/30
140/140 [==============================] - ETA: 0s - loss: 0.6936 - accuracy: 0.7525
Epoch 3: val_loss improved from 0.68143 to 0.54119, saving model to bestmodel_3class.hdf5
140/140 [==============================] - 67s 479ms/step - loss: 0.6936 - accuracy: 0.7525 - val_loss: 0.5412 - val_accuracy: 0.8492
Epoch 4/30
140/140 [==============================] - ETA: 0s - loss: 0.5781 - accuracy: 0.8039
Epoch 4: val_loss improved from 0.54119 to 0.45043, saving model to bestmodel_3class.hdf5
140/140 [==============================] - 71s 506ms/step - loss: 0.5781 - accuracy: 0.8039 - val_loss: 0.4504 - val_accuracy: 0.8655
Epoch 5/30
140/140 [==============================] - ETA: 0s - loss: 0.4950 - accuracy: 0.8380
Epoch 5: val_loss improved from 0.45043 to 0.37958, saving model to bestmodel_3class.hdf5
140/140 [==============================] - 66s 470ms/step - loss: 0.4950 - accuracy: 0.8380 - val_loss: 0.3796 - val_accuracy: 0.8845
```

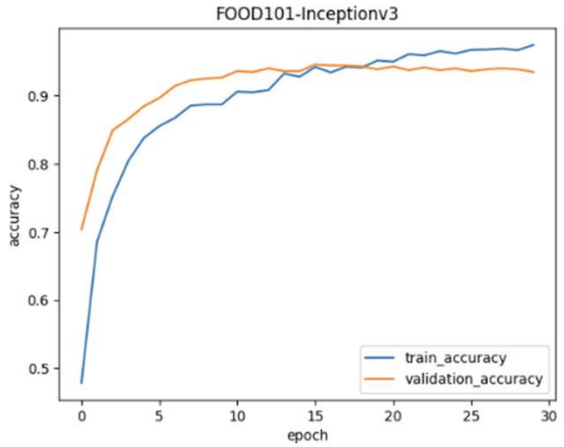
Contents

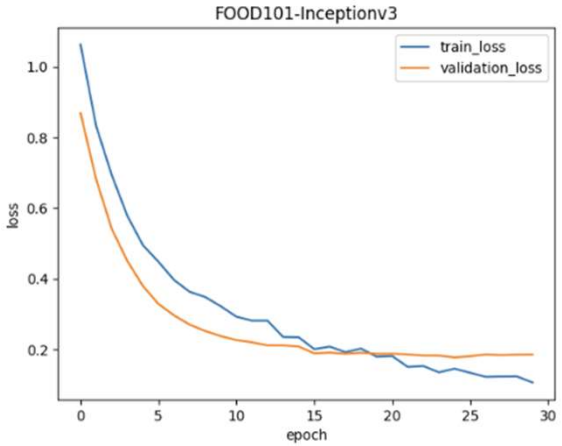
성능 그래프

```
def plot_accuracy(history,title):
    plt.title(title)
    plt.plot(history.history['accuracy']) # change acc to accuracy if testing TF 2.0
    plt.plot(history.history['val_accuracy']) # change val_accuracy if testing TF 2.0
    plt.ylabel('accuracy')
    plt.xlabel('epoch')
    plt.legend(['train_accuracy', 'validation_accuracy'], loc='best')
    plt.show()


def plot_loss(history,title):
    plt.title(title)
    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.ylabel('loss')
    plt.xlabel('epoch')
    plt.legend(['train_loss', 'validation_loss'], loc='best')
    plt.show()


plot_accuracy(history,'FOOD101-Inceptionv3')
plot_loss(history,'FOOD101-Inceptionv3')
```





Contents

예측 및 테스트

```
K.clear_session()
model_best = load_model('bestmodel_3class.hdf5',compile = False)
```

```
CPU times: user 3.76 s, sys: 89.2 ms, total: 3.85 s
Wall time: 3.9 s
```

```
def predict_class(model, images, show = True):
  for img in images:
    img = image.load_img(img, target_size=(299, 299))
    img = image.img_to_array(img)
    img = np.expand_dims(img, axis=0)
    img = preprocess_input(img)

    pred = model.predict(img)
    index = np.argmax(pred)
    food_list.sort()
    pred_value = food_list[index]
    #print(pred)
    if show:
        plt.imshow(img[0])
        plt.axis('off')
        plt.title(pred_value)
        plt.show()
```

```
uploaded = files.upload() ## 추가, 파일 업로드 기능 실행
for fn in uploaded.keys():
  print('User uploaded file "{name}" with length {length} bytes'.format(
      name=fn, length=len(uploaded[fn])))
```

파일 선택 upload.jpg

- **upload.jpg**(image/jpeg) - 100000 bytes, last modified: 2023. 6. 8. - 100% done

```
Saving upload.jpg to upload.jpg
User uploaded file "upload.jpg" with length 100000 bytes
```

```
images.append('upload.jpg')

predict_class(model_best, images, True)
```

```
1/1 [==============================] - 3s 3s/step
WARNING:matplotlib.image:Clipping input data to the valid range
```

apple_pie

Contents

여름방학 진행 예정 사항

여름학기 진행예정

1. 유저 인터페이스 구성
2. DB 구성
3. 서버 구축
4. 모델 최적화

감사합니다