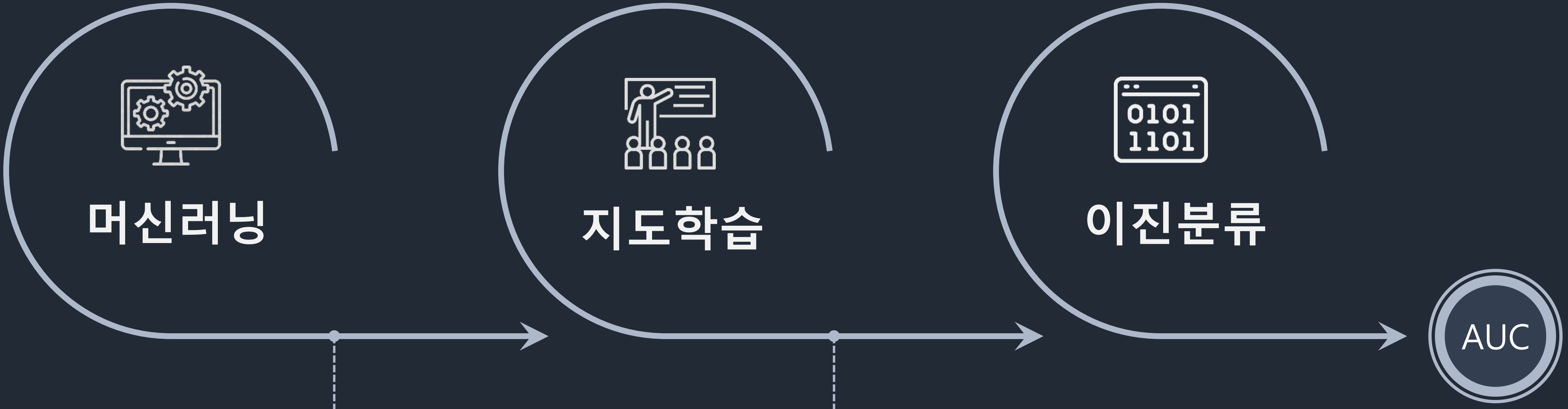
22-2 기계학습기초

딥러닝 경진대회 진행사항

인공지능학부 215001 서가연

01. 문제 정의



move_out

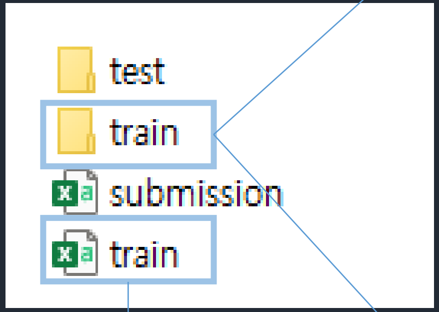
정답데이터 존재

0 : 퇴거, 1: 미퇴거

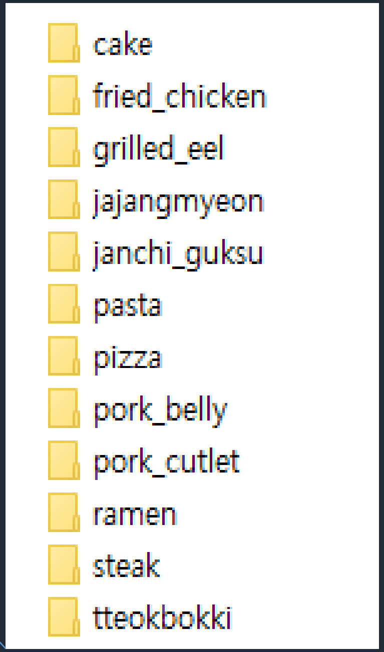
target value는 0 또는 1

대구시_영구임대아파트
입주자 퇴거여부 예측

02. 데이터 탐색

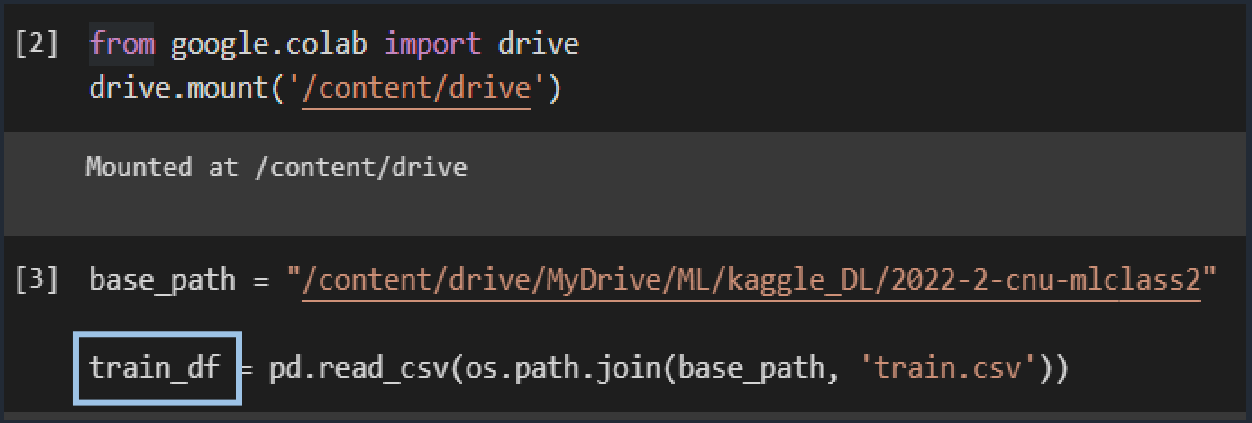

test
train
submission
train

데이터 정보

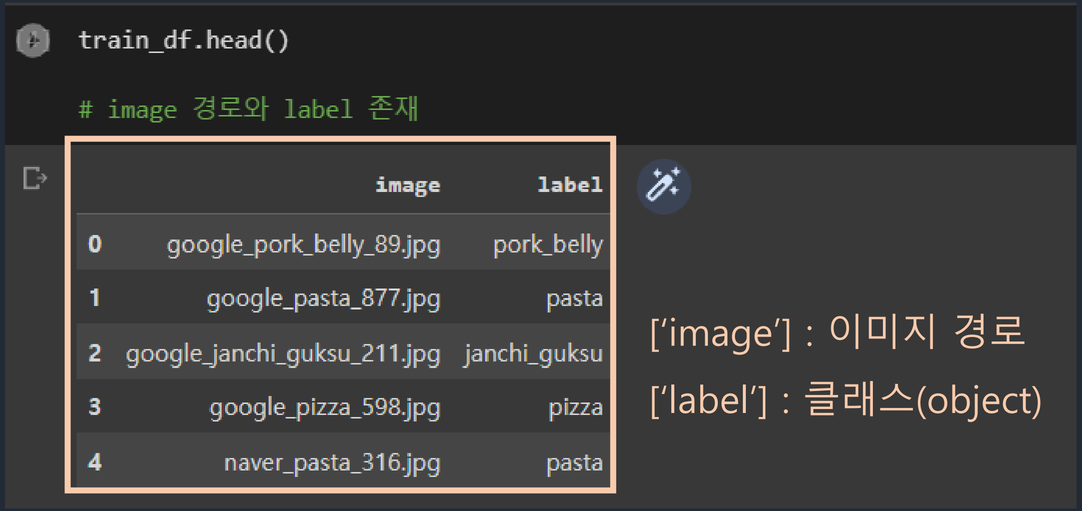

cake
fried_chicken
grilled_eel
jajangmyeon
janchi_guksu
pasta
pizza
pork_belly
pork_cutlet
ramen
steak
tteokbokki

12개의 클래스

데이터 로드


```
[2]  from google.colab import drive
     drive.mount('/content/drive')

     Mounted at /content/drive

[3]  base_path = "/content/drive/MyDrive/ML/kaggle_DL/2022-2-cnu-mlclass2"

     train_df = pd.read_csv(os.path.join(base_path, 'train.csv'))
```

02. 데이터 탐색

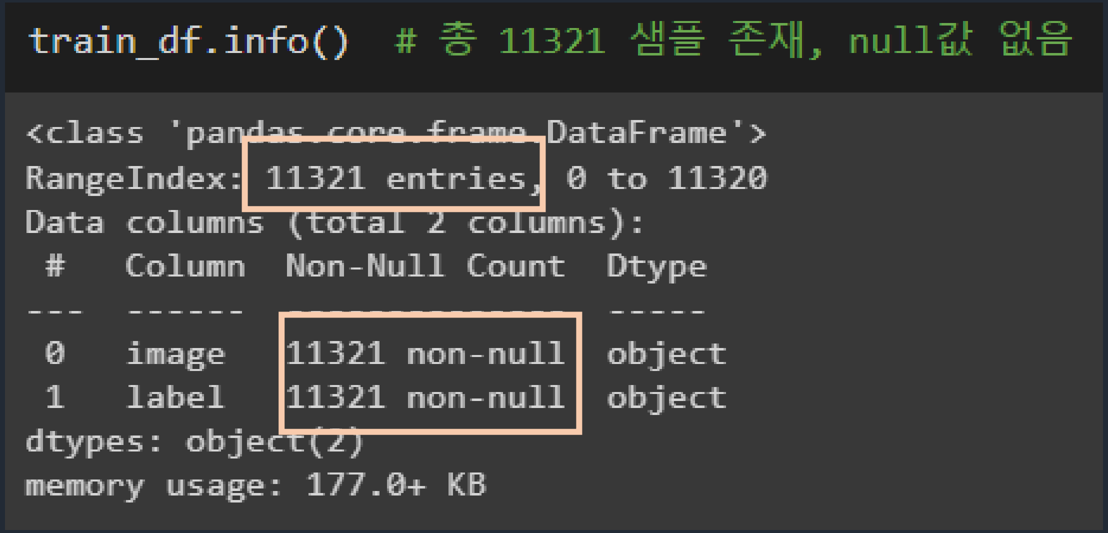


```
train_df.head()

# image 경로와 label 존재
```

	image	label
0	google_pork_belly_89.jpg	pork_belly
1	google_pasta_877.jpg	pasta
2	google_janchi_guksu_211.jpg	janchi_guksu
3	google_pizza_598.jpg	pizza
4	naver_pasta_316.jpg	pasta

['image'] : 이미지 경로

['label'] : 클래스(object)



```
train_df.info()  # 총 11321 샘플 존재, null값 없음
```

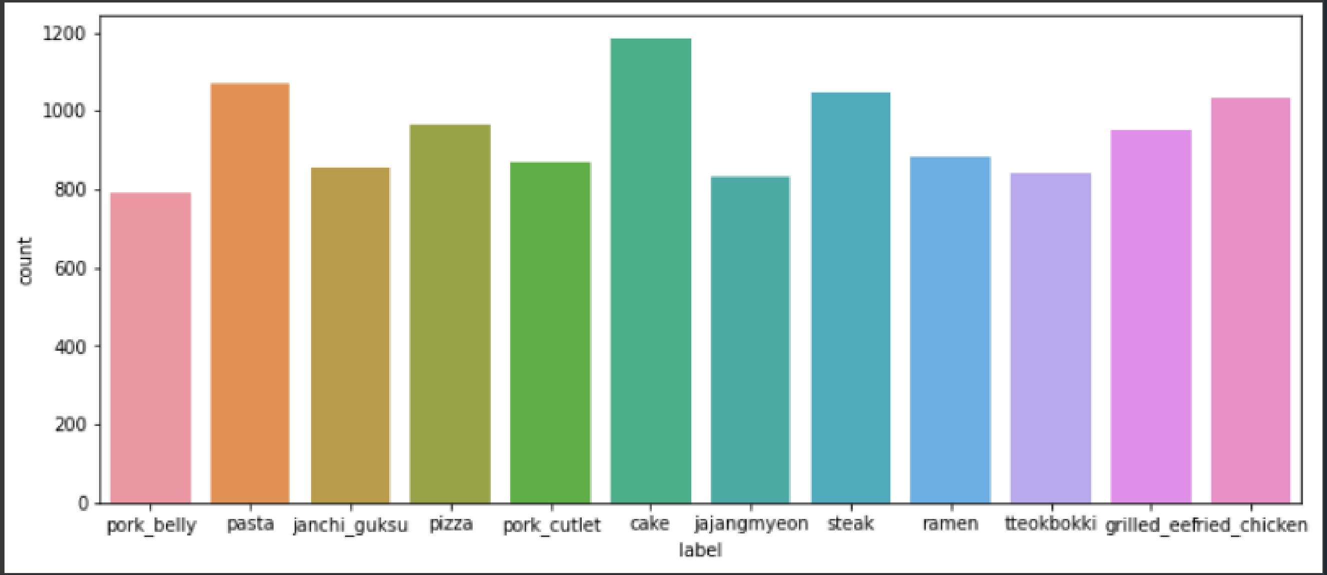
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11321 entries, 0 to 11320
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   image   11321 non-null  object
 1   label   11321 non-null  object
dtypes: object(2)
memory usage: 177.0+ KB
```

02. 데이터 탐색

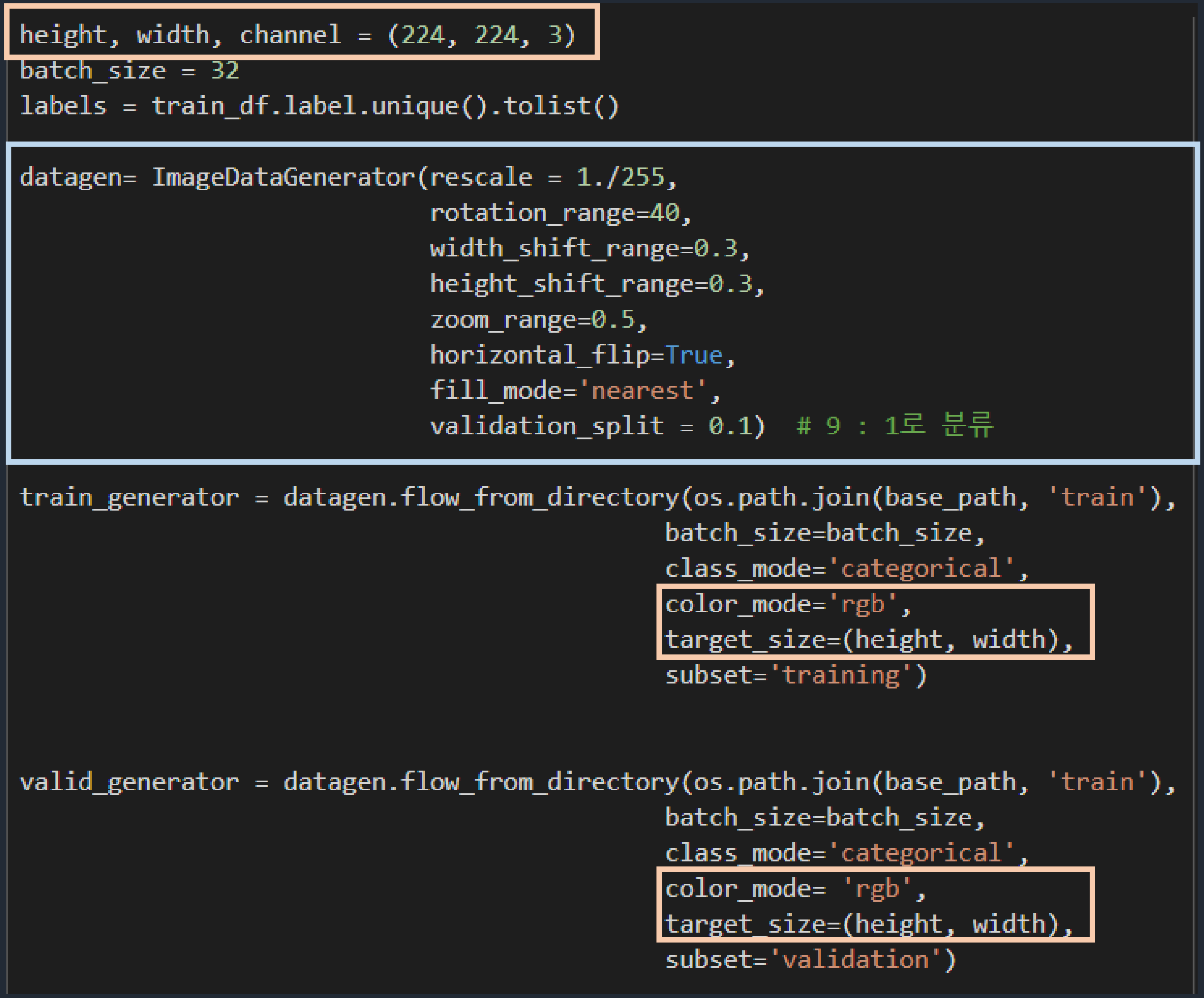
```python
train_df['label'].value_counts()
```

```
cake            1184
pasta           1072
steak           1048
fried_chicken   1034
pizza            963
grilled_eel      951
ramen            881
pork_cutlet      869
janchi_guksu     856
tteokbokki       841
jajangmyeon      833
pork_belly       789
Name: label, dtype: int64
```

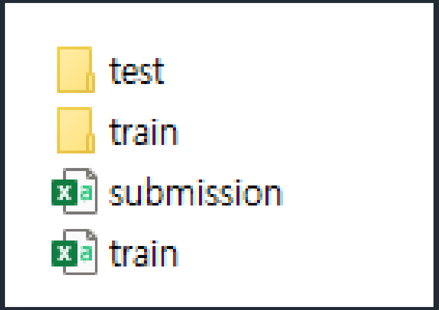
```python
# 데이터 분포 => 거의 비슷하게 분포

figure = plt.figure(figsize=(12, 5))
sns.countplot(data=train_df, x='label')
plt.show()
```



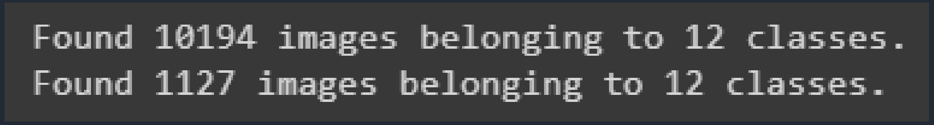
03. 데이터 전처리



```
height, width, channel = (224, 224, 3)
batch_size = 32
labels = train_df.label.unique().tolist()

datagen= ImageDataGenerator(rescale = 1./255,
                            rotation_range=40,
                            width_shift_range=0.3,
                            height_shift_range=0.3,
                            zoom_range=0.5,
                            horizontal_flip=True,
                            fill_mode='nearest',
                            validation_split = 0.1)  # 9 : 1로 분류

train_generator = datagen.flow_from_directory(os.path.join(base_path, 'train'),
                                              batch_size=batch_size,
                                              class_mode='categorical',
                                              color_mode='rgb',
                                              target_size=(height, width),
                                              subset='training')


valid_generator = datagen.flow_from_directory(os.path.join(base_path, 'train'),
                                              batch_size=batch_size,
                                              class_mode='categorical',
                                              color_mode= 'rgb',
                                              target_size=(height, width),
                                              subset='validation')
```



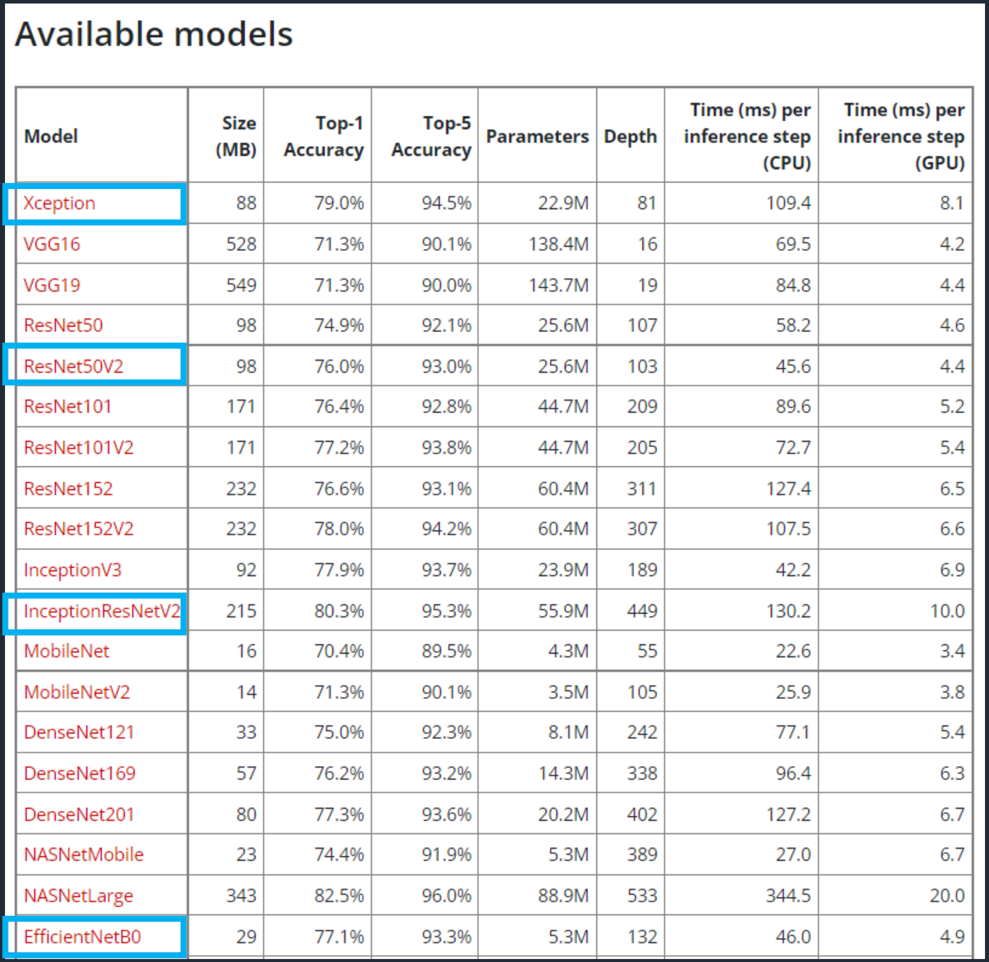
test
train
submission
train



```
Found 10194 images belonging to 12 classes.
Found 1127 images belonging to 12 classes.
```

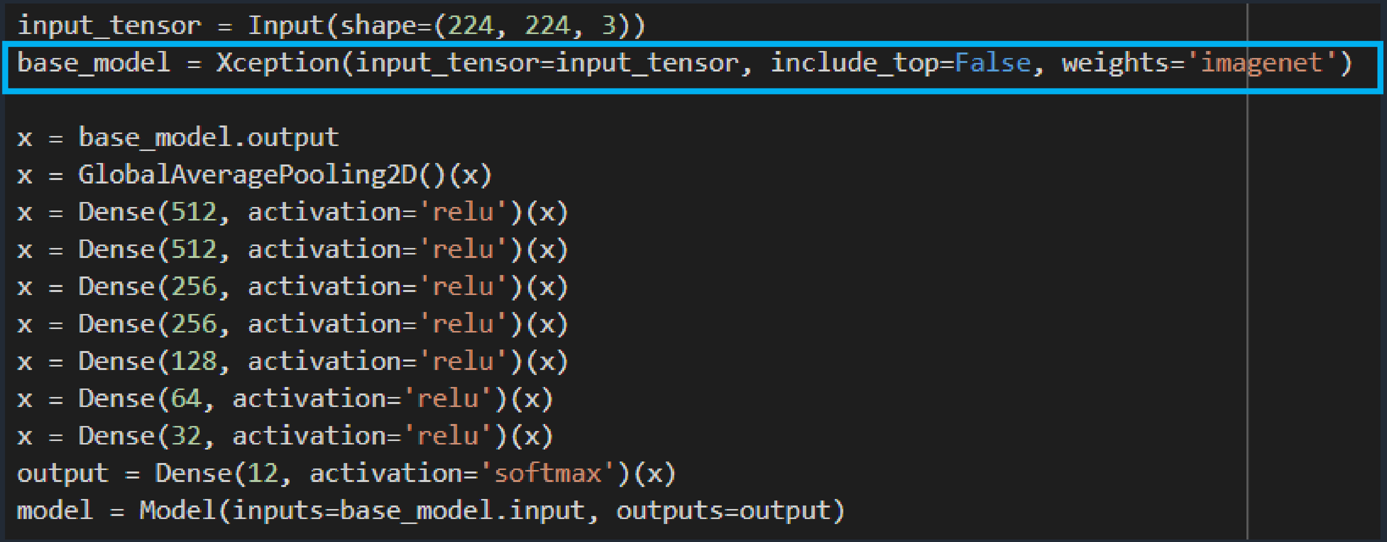
04. 모델 선택

사전학습 모델 사용



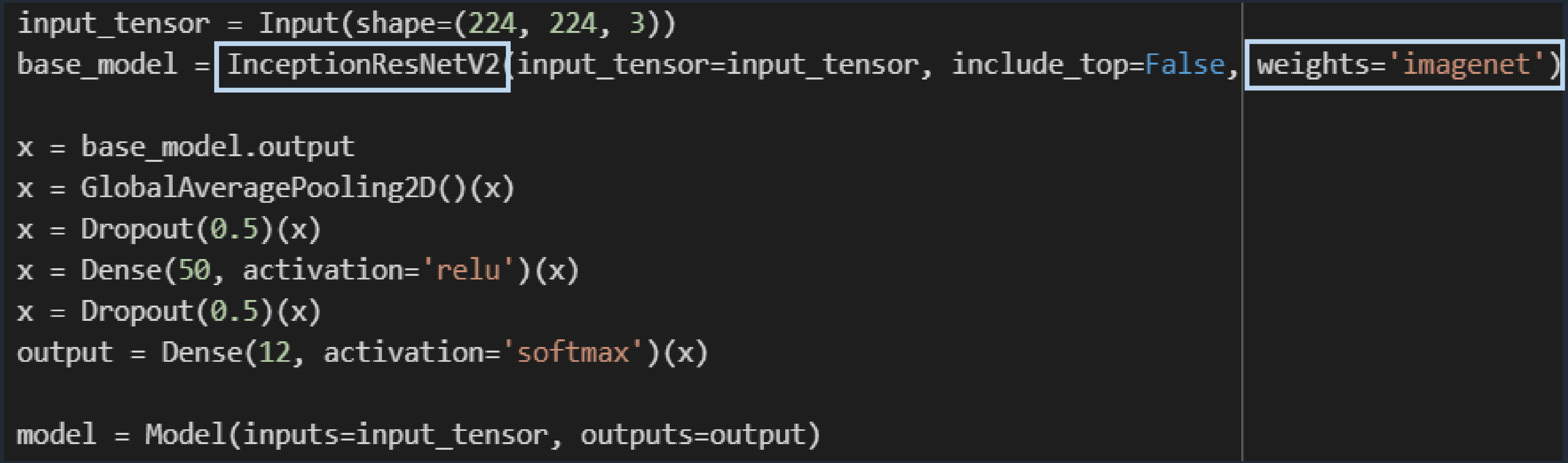
Available models

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
VGG19	549	71.3%	90.0%	143.7M	19	84.8	4.4
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6
ResNet50V2	98	76.0%	93.0%	25.6M	103	45.6	4.4
ResNet101	171	76.4%	92.8%	44.7M	209	89.6	5.2
ResNet101V2	171	77.2%	93.8%	44.7M	205	72.7	5.4
ResNet152	232	76.6%	93.1%	60.4M	311	127.4	6.5
ResNet152V2	232	78.0%	94.2%	60.4M	307	107.5	6.6
InceptionV3	92	77.9%	93.7%	23.9M	189	42.2	6.9
InceptionResNetV2	215	80.3%	95.3%	55.9M	449	130.2	10.0
MobileNet	16	70.4%	89.5%	4.3M	55	22.6	3.4
MobileNetV2	14	71.3%	90.1%	3.5M	105	25.9	3.8
DenseNet121	33	75.0%	92.3%	8.1M	242	77.1	5.4
DenseNet169	57	76.2%	93.2%	14.3M	338	96.4	6.3
DenseNet201	80	77.3%	93.6%	20.2M	402	127.2	6.7
NASNetMobile	23	74.4%	91.9%	5.3M	389	27.0	6.7
NASNetLarge	343	82.5%	96.0%	88.9M	533	344.5	20.0
EfficientNetB0	29	77.1%	93.3%	5.3M	132	46.0	4.9

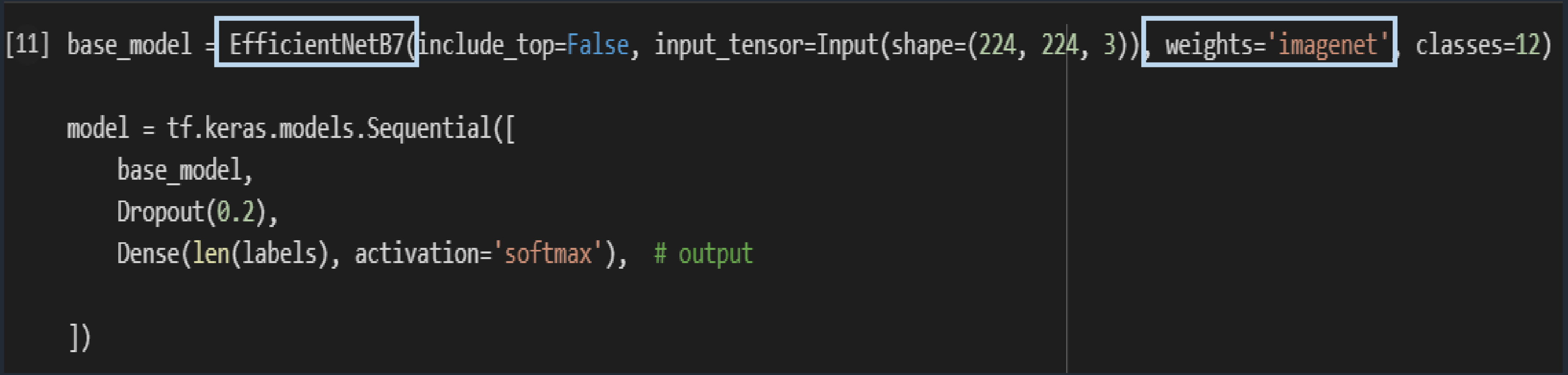


```python
input_tensor = Input(shape=(224, 224, 3))
base_model = Xception(input_tensor=input_tensor, include_top=False, weights='imagenet')

x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(512, activation='relu')(x)
x = Dense(512, activation='relu')(x)
x = Dense(256, activation='relu')(x)
x = Dense(256, activation='relu')(x)
x = Dense(128, activation='relu')(x)
x = Dense(64, activation='relu')(x)
x = Dense(32, activation='relu')(x)
output = Dense(12, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=output)
```

04. 모델 선택

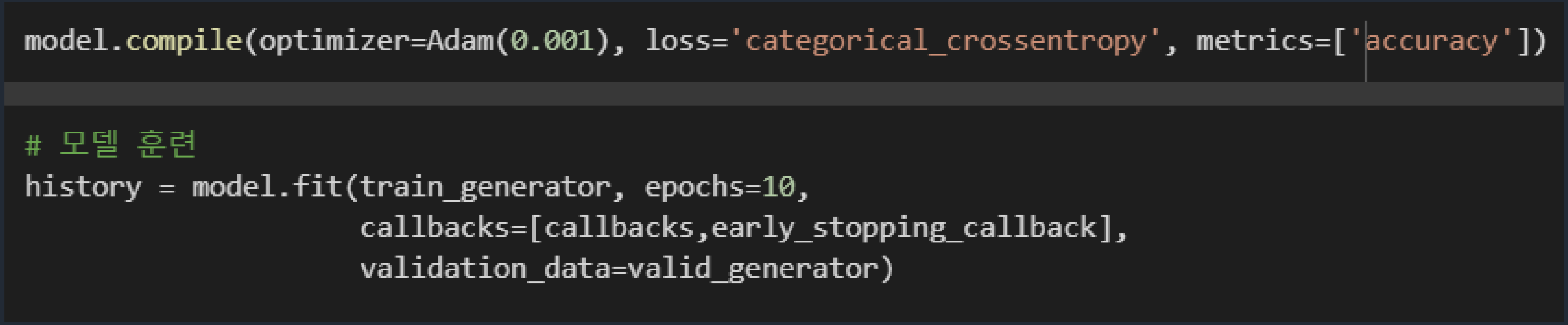


```
input_tensor = Input(shape=(224, 224, 3))
base_model = InceptionResNetV2(input_tensor=input_tensor, include_top=False, weights='imagenet')

x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dropout(0.5)(x)
x = Dense(50, activation='relu')(x)
x = Dropout(0.5)(x)
output = Dense(12, activation='softmax')(x)

model = Model(inputs=input_tensor, outputs=output)
```



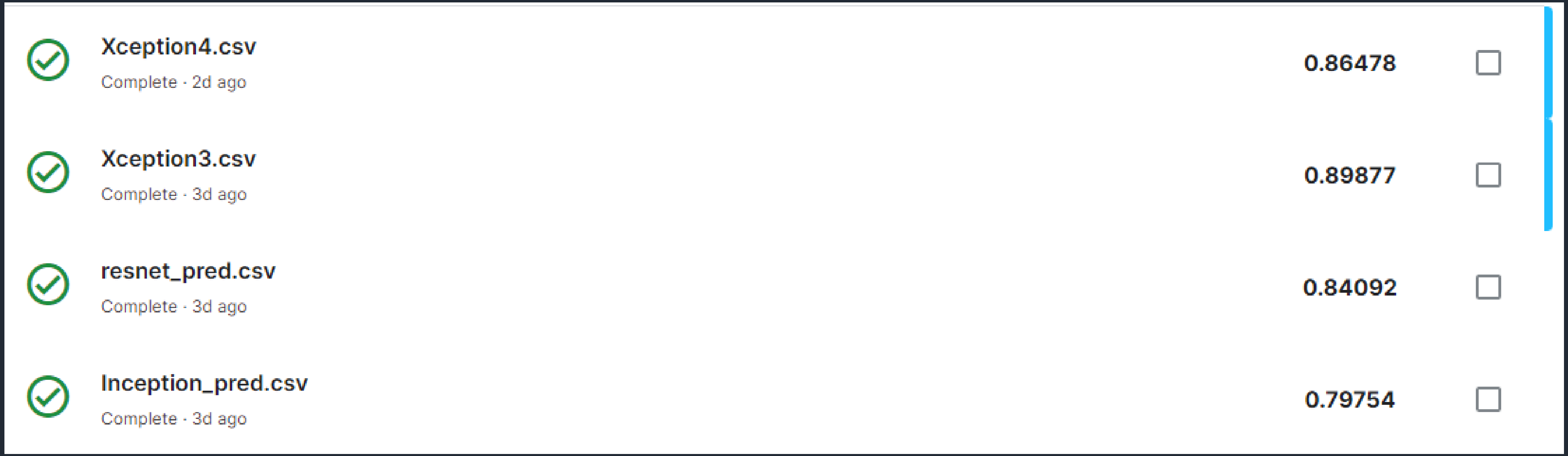
```
[11] base_model = EfficientNetB7(include_top=False, input_tensor=Input(shape=(224, 224, 3)), weights='imagenet', classes=12)

     model = tf.keras.models.Sequential([
         base_model,
         Dropout(0.2),
         Dense(len(labels), activation='softmax'),  # output

     ])
```

05. 모델링



```python
model.compile(optimizer=Adam(0.001), loss='categorical_crossentropy', metrics=['accuracy'])
```

```python
# 모델 훈련
history = model.fit(train_generator, epochs=10,
                    callbacks=[callbacks,early_stopping_callback],
                    validation_data=valid_generator)
```



Xception4.csv Complete · 2d ago	0.86478	☐
Xception3.csv Complete · 3d ago	0.89877	☐
resnet_pred.csv Complete · 3d ago	0.84092	☐
Inception_pred.csv Complete · 3d ago	0.79754	☐

06. 계획

- 전처리 기법
- Fine tuning

감사합니다