In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
import h5py
import time
import random
import os
from datetime import datetime, timedelta
np.random.seed(1)
```

In [2]:
```
import tensorflow as tf
```

In [3]:
```
# Tensorflow 버전 확인
tf.__version__
```

Out[3]:
```
'2.1.0'
```

In [4]:
```
# GPU 사용 여부 확인
print(tf.test.is_built_with_cuda())
print(tf.config.list_physical_devices('GPU'))
```

```
True
[PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
```

이미지 데이터 처리

이미 훈련된 VGG16 모델을 사용한 전이 학습 및 특성 추출

In [5]:
```
from PIL import Image
from tensorflow.keras import Input
from tensorflow.keras import models
from tensorflow.keras import layers
from tensorflow.keras import optimizers, initializers, regularizers, metrics
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPool2D, Flatten, Dense, Dropout
```

In [6]:
```
from tensorflow.keras.applications import VGG16
conv_base = VGG16(weights='imagenet',
                  include_top=False,
                  input_shape=(150, 150, 3))
```

In [7]:
```
conv_base.summary()
```

```
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: 14,714,688
Non-trainable params: 0
_________________________________________________________________
```

In [10]:

```
# 수집된 총 이미지 개수 : 153,576
# 이미지가 존재하지 않는 이미지 파일 : 1
# 제거 후 총 이미지 개수 : 153,575
# 데이터 양이 충분히 많기에 image augmentation은 할 필요가 없다고 판단
# 특성추출이 내가 수집한 이미지 데이터에 더 적합하도록 맞추기 위해 fine-tuning방식 사용
```

In [8]:

```
# 미세조정(fine-tuning) 시도
conv_base.layers
```

Out[8]:

```
[<tensorflow.python.keras.engine.input_layer.InputLayer at 0x7fcb84d71c90>,
 <tensorflow.python.keras.layers.convolutional.Conv2D at 0x7fcbb43b9ad0>,
 <tensorflow.python.keras.layers.convolutional.Conv2D at 0x7fcb0abde950>,
 <tensorflow.python.keras.layers.pooling.MaxPooling2D at 0x7fcb0abdeed0>,
 <tensorflow.python.keras.layers.convolutional.Conv2D at 0x7fcb00120710>,
 <tensorflow.python.keras.layers.convolutional.Conv2D at 0x7fcb00120110>,
 <tensorflow.python.keras.layers.pooling.MaxPooling2D at 0x7fcb001339d0>,
 <tensorflow.python.keras.layers.convolutional.Conv2D at 0x7fcb00139890>,
 <tensorflow.python.keras.layers.convolutional.Conv2D at 0x7fcb00144f90>,
 <tensorflow.python.keras.layers.convolutional.Conv2D at 0x7fcb0014ac50>,
 <tensorflow.python.keras.layers.pooling.MaxPooling2D at 0x7fcb0014ffd0>,
 <tensorflow.python.keras.layers.convolutional.Conv2D at 0x7fcb00156e50>,
 <tensorflow.python.keras.layers.convolutional.Conv2D at 0x7fcb000e2d50>,
 <tensorflow.python.keras.layers.convolutional.Conv2D at 0x7fcb000e5cd0>,
 <tensorflow.python.keras.layers.pooling.MaxPooling2D at 0x7fcb000eba50>,
 <tensorflow.python.keras.layers.convolutional.Conv2D at 0x7fcb000efc50>,
 <tensorflow.python.keras.layers.convolutional.Conv2D at 0x7fcb000fbb50>,
 <tensorflow.python.keras.layers.convolutional.Conv2D at 0x7fcb000feb10>,
 <tensorflow.python.keras.layers.pooling.MaxPooling2D at 0x7fcb00105c50>]
```

In [9]:

```
# block5_conv1, block5_conv2, block5_conv3 --> fine-tuning
conv_base.trainable = True

set_trainable = False
for layer in conv_base.layers:
    if layer.name == 'block5_conv1':
        set_trainable = True
    if set_trainable:
        layer.trainable = True
    else:
        layer.trainable = False
```

In [10]:

```
conv_base.summary()
```

```
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: 7,079,424
Non-trainable params: 7,635,264
_________________________________________________________________
```

In [11]:

```
#디렉토리에서 이미지 로드 및 generator 생성
from tensorflow.keras.preprocessing.image import ImageDataGenerator

datagen = ImageDataGenerator(rescale=1./255)
batch_size = 512

def extract_features(directory, sample_count):
    features = np.zeros(shape=(sample_count, 4, 4, 512)) #추출된 이미지 특성들을 저장할 ndarray변수
    generator = datagen.flow_from_directory(directory,
                                        target_size=(150, 150),
                                        batch_size=batch_size,
                                        class_mode=None)

    i = 0 #predict한 이미지 수 저장
    for inputs_batch in generator:
        #VGG16을 통해 나오는 특성맵 저장
        feature_map_batch = conv_base.predict(inputs_batch)
        features[i * batch_size : (i + 1) * batch_size] = feature_map_batch #index

        i += 1
        if i * batch_size >= sample_count:
            break
    return features, generator.filenames
```

In [ ]:

```
# image feature extraction
try:
    image_dir = './image_2015_2019_folder'
    features, filenames = extract_features(image_dir, 153575)
    image_data = features.reshape(153575, 4*4*512)
    df_image = pd.DataFrame(image_data)
    try:
        df_image['index'] = [filename.split('/')[1].split('.')[0] for filename in filenames]
        df_image['index'] = df_image['index'].astype('int64')
        df_image = df_image.set_index('index').sort_index().reset_index()
        df_image.to_pickle('image_feature_df.pkl')
    except Exception as e:
        df_image['index'] = filenames
        df_image.to_pickle('image_feature_df.pkl')

        filename_split_error = []
        filename_split_error.append(e)
        split_error = pd.DataFrame(filename_split_error, columns=['error'])
        split_error.to_csv('filename_split_error.csv', index=False)

except Exception as ex:
    error_list = []
    error_list.append(ex)
    error = pd.DataFrame(error_list, columns=['error'])
    error.to_csv('image_feature_extraction_failed.csv', index=False)
```

```
Found 153575 images belonging to 1 classes.

/home/eodud0582/anaconda3/envs/tensorflow/lib/python3.7/site-packages/PIL/Image.py:932: UserWarning: Palette images with Trans
parency expressed in bytes should be converted to RGBA images
  "Palette images with Transparency expressed in bytes should be "
```

이미지 데이터 전처리 결과

- Shape : (153575, 8192)
- 다른 데이터셋과의 결합을 위해 Index 번호 추가 : (153575, 8193)

In [ ]:

메타데이터, 텍스트데이터와 결합 후 적용 분류 모델

In [ ]:

```
model = Sequential()
#model.add(Dense(256, activation='relu', input_shape=(4*4*512)))
model.add(Dense(1, activation='sigmoid'))
```

In [ ]:

```
model.compile(loss='binary_crossentropy',
              optimizer=optimizers.Adam(lr=0.0001),
              metrics=['accuracy'])

history = model.fit_generator(
          train_generator,
          steps_per_epoch=200,
          epochs=30,
          validation_data=validation_generator,
          validation_steps=50)
```