

# pre-trained model 그대로 사용하기

박성호 (neowizard2018@gmail.com)

```
import tensorflow as tf
from tensorflow.keras.applications import Xception, InceptionV3, MobileNet, MobileNetV2
```

## MobileNet 사전학습모델 구축

MobileNet 사전학습모델을 그대로 사용할 경우 input\_shape=(224,224,3)

```
mobilenet_model = MobileNet(weights='imagenet', include_top=True, input_shape=(224,224,3))

mobilenet_model.summary()
```

## MobileNetV2 사전학습모델 구축

MobileNetV2 사전학습모델을 그대로 사용할 경우 input\_shape=(224,224,3)

```
mobilenetv2_model = MobileNetV2(weights='imagenet', include_top=True, input_shape=(224,224,3))

mobilenetv2_model.summary()
```

## Xception 사전학습모델 구축

Xception 사전학습모델을 그대로 사용할 경우 input\_shape=(299,299,3)

```
xception_model = Xception(weights='imagenet', include_top=True, input_shape=(299,299,3))  
  
xception_model.summary()
```

## InceptionV3 사전학습모델 구축

InceptionV3 사전학습모델을 그대로 사용할 경우 input\_shape=(299,299,3)

```
inceptionv3_model = InceptionV3(weights='imagenet', include_top=True, input_shape=(299,299,3))  
  
inceptionv3_model.summary()
```

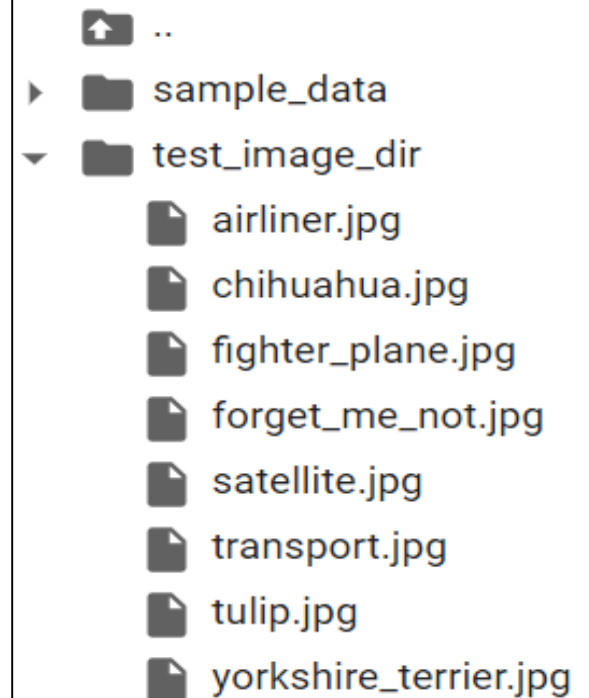
## 테스트 이미지 리스트 가져오기

```
import glob

# 테스트 이미지 가져오기

test_image_list = glob.glob('/content/test_image_dir/*.jpg')

print('total image # => ', len(test_image_list))
print(test_image_list)
```



## OpenCV 이용하여 테스트 이미지 로드 및 사이즈 변경

```
import cv2

dst_img_list_224 = []
dst_img_list_299 = []
label_str_list = []

for i in range(len(test_image_list)):

    src_img = cv2.imread(test_image_list[i], cv2.IMREAD_COLOR)

    dst_img_224 = cv2.resize(src_img, dsize=(224, 224)) # MobileNet, MobileNetV2
    dst_img_299 = cv2.resize(src_img, dsize=(299, 299)) # InceptionV3, Xception

    dst_img_224 = cv2.cvtColor(dst_img_224, cv2.COLOR_BGR2RGB)
    dst_img_299 = cv2.cvtColor(dst_img_299, cv2.COLOR_BGR2RGB)

    dst_img_224 = dst_img_224 / 255.0
    dst_img_299 = dst_img_299 / 255.0

    label_str = test_image_list[i].split('/')[ -1 ].split('.')[ 0 ].strip() # 이미지 정답 추출

    dst_img_list_224.append(dst_img_224)
    dst_img_list_299.append(dst_img_299)

    label_str_list.append(label_str) # 이미지 정답 저장
```

# 테스트 이미지 출력

```
import matplotlib.pyplot as plt

plt.figure(figsize=(8,6))

for i in range(len(dst_img_list_224)):

    plt.subplot(2,4,i+1)
    plt.axis('off')

    plt.title(label_str_list[i])

    plt.imshow(dst_img_list_224[i])

plt.tight_layout()
plt.show()
```

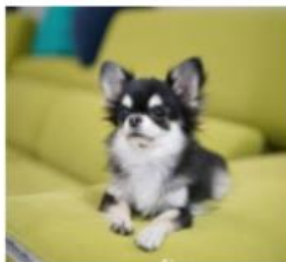
yorkshire\_terrier



airliner



chihuahua



tulip



fighter\_plane



transport



satellite



forget\_me\_not



## 각각의 사전학습 모델을 이용한 predict() 실행

```
mobilenet_pred = mobilenet_model.predict(dst_img_array_224) # MobileNet prediction

mobilenetv2_pred = mobilenetv2_model.predict(dst_img_array_224) # MobileNetV2 prediction

inceptionv3_pred = inceptionv3_model.predict(dst_img_array_299) # InceptionV3 prediction

xception_pred = xception_model.predict(dst_img_array_299) # Xception prediction

print(mobilenet_pred.shape)
print(mobilenetv2_pred.shape)
print(inceptionv3_pred.shape)
print(xception_pred.shape)
```

```
(8, 1000)
(8, 1000)
(8, 1000)
(8, 1000)
```

## decode\_predictions() 실행을 통해 top3 확인

```
from tensorflow.keras.applications.imagenet_utils import decode_predictions

mobilenet_prediction = decode_predictions(mobilenet_pred, top=3) # MobileNet
mobilenetv2_prediction = decode_predictions(mobilenetv2_pred, top=3) # MobileNetV2
xception_prediction = decode_predictions(xception_pred, top=3) # Xception
inceptionv3_prediction = decode_predictions(inceptionv3_pred, top=3) # InceptionV3
```

```
print(type(mobilenet_prediction))
print(type(mobilenetv2_prediction))
print(type(xception_prediction))
print(type(inceptionv3_prediction))
```

```
<class 'list'>
<class 'list'>
<class 'list'>
<class 'list'>
```



```
print('MobileNet Prediction Result')

for i in range(len(mobilenet_prediction)):

    print('=====')
    print('label=', label_str_list[i])
    print(mobilenet_prediction[i])
    print('=====')
```

```
print('MobileNetV2 Prediction Result')

for i in range(len(mobilenetv2_prediction)):

    print('=====')
    print('label=', label_str_list[i])
    print(mobilenetv2_prediction[i])
    print('=====')
```

```
print('Xception Prediction Result')

for i in range(len(xception_prediction)):

    print('=====')
    print('label=', label_str_list[i])
    print(xception_prediction[i])
    print('=====')
```

```
print('InceptionV3 Prediction Result')

for i in range(len(inceptionv3_prediction)):

    print('=====')
    print('label=', label_str_list[i])
    print(inceptionv3_prediction[i])
    print('=====')
```