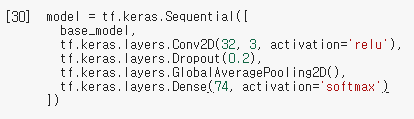
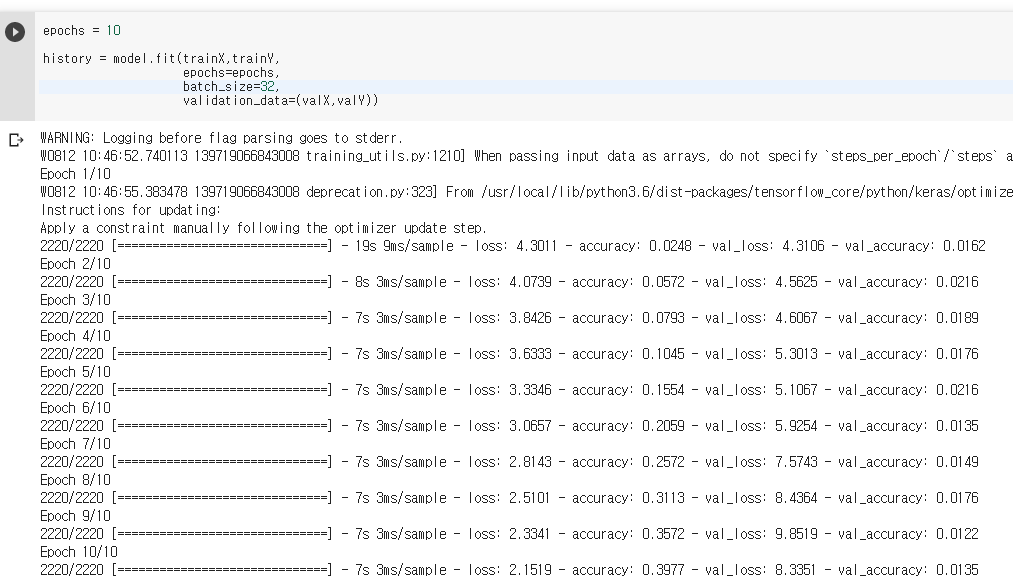
**Image Classification을 위한 MobileNetV2 transfer-learning**

**<첫번째 트레이닝>**





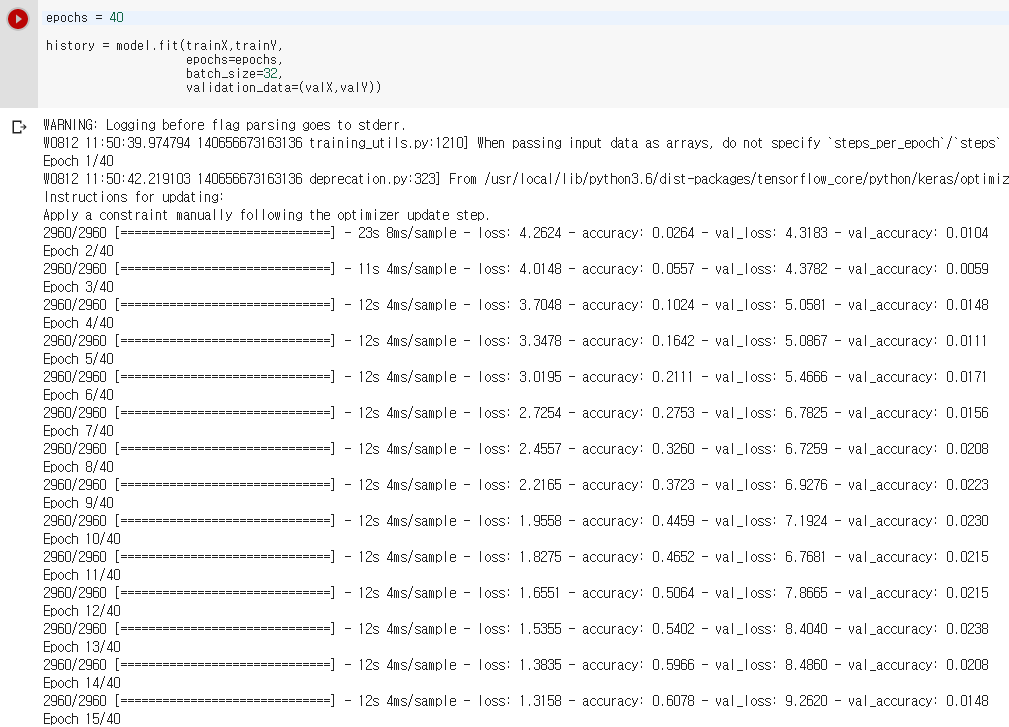
(GPU쓰니까 속도가 확실히 장난없다…ㄷㄷ1분정도 걸렸나?)

10번 학습시켰을때, accuracy가 대략 40퍼센트. Val\_loss는 점점 증가하고 있다.

**overfitting되고 있는 상태.-> 데이터는 늘리고, 정규화 추가**

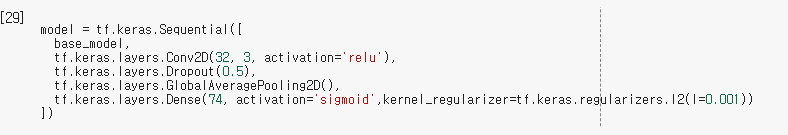
**<두번째 트레이닝>**

**(train,val)=(30,10) -> (40,10)**

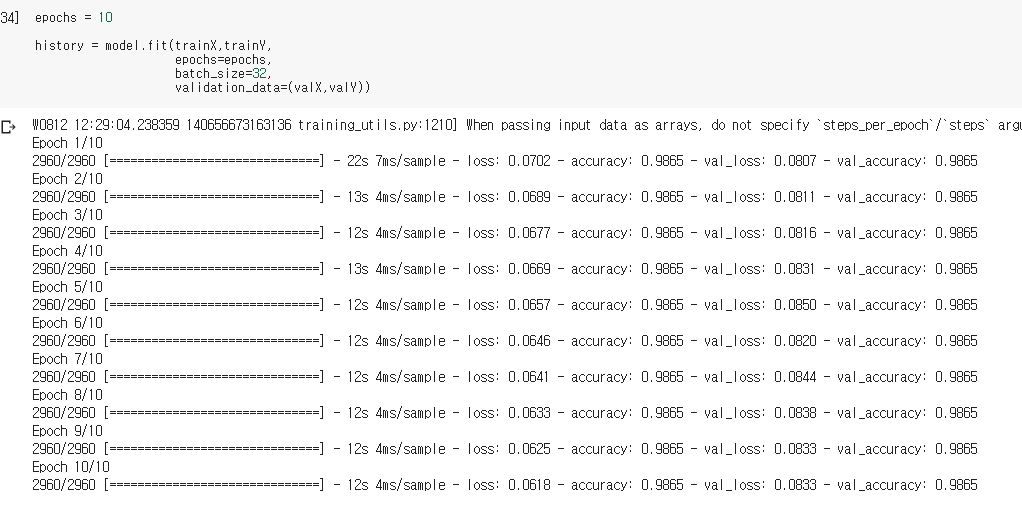
****

**val\_accuracy가 별 진전이 없음. Train\_acc는 점점 증가**

**<세번째 트레이닝>**

****

**Dropout, l2 regularization 추가**

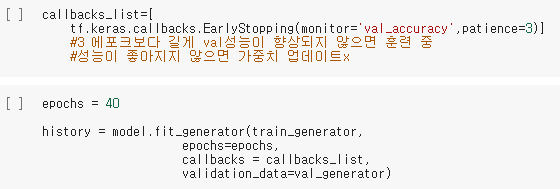
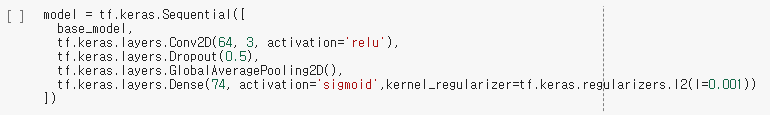
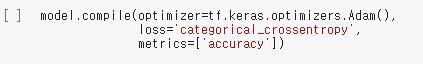
****

**train\_acc, Val\_acc: 98.5%**

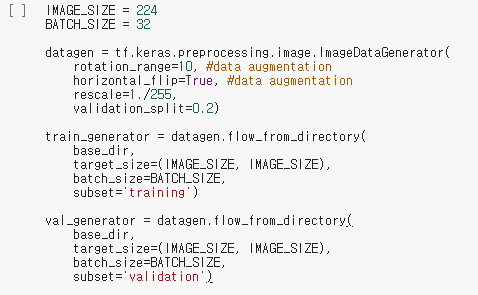
**<네번째 트레이닝- 다시 원점으로>**

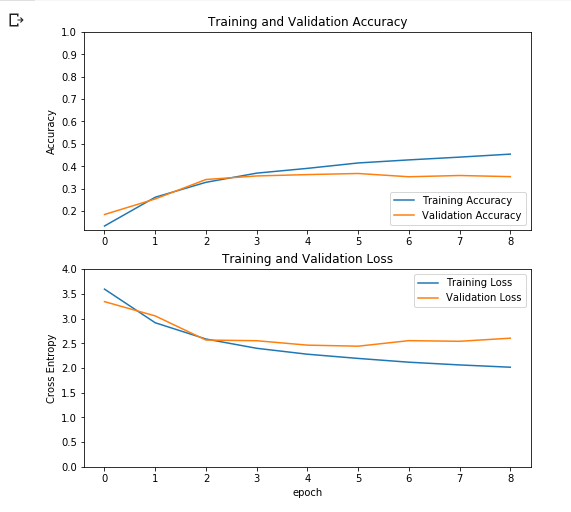
**안드로이드 스튜디오에 올리는 과정에서 정확도가 너무 낮게 나와서 트레이닝을 다시했다.**

**이번엔 내가 원래 적었던 코드가 아니라, image generator를 사용하였고 binary crossentropy가 아니라 categorical crossentropy를 사용하였다. Categorical을 사용하니 정확도가 현저히 떨어졌는데 사실 이제 맞는 것이였다.**

****

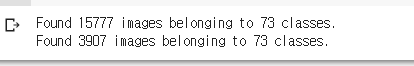
**<다섯번째 트레이닝-데이터 수를 늘리기>**

**원래 각 40개의 트레이닝 데이터를 200개 정도로 재수집하여 늘려주고, data augmentaion을 통해 또 한번 늘려주었다.**



**결과는…처참…. 정확도가 전혀 오르지 않았다…**

**<여섯번째 시도>**

****

**Augmentaion이 부질없는 것 같아 그냥 없이 해주었다. **

**밑져야 본전이다. 그냥 원래 모델에서 했던 것처럼 dense 2개만 추가해 주었다.**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

mobilenetv2\_1.00\_224 (Model) (None, 7, 7, 1280) 2257984

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_5 (Dense) (None, 7, 7, 128) 163968

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_6 (Dense) (None, 7, 7, 73) 9417

=================================================================

Total params: 2,431,369

Trainable params: 173,385

Non-trainable params: 2,257,984

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**그런데 fit\_generator에서 categorical crossentropy에서 문제가 발생했다. 다시 돌려놓았다..**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

mobilenetv2\_1.00\_224 (Model) (None, 7, 7, 1280) 2257984

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_4 (Conv2D) (None, 5, 5, 64) 737344

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_4 (Dropout) (None, 5, 5, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

global\_average\_pooling2d\_5 ( (None, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

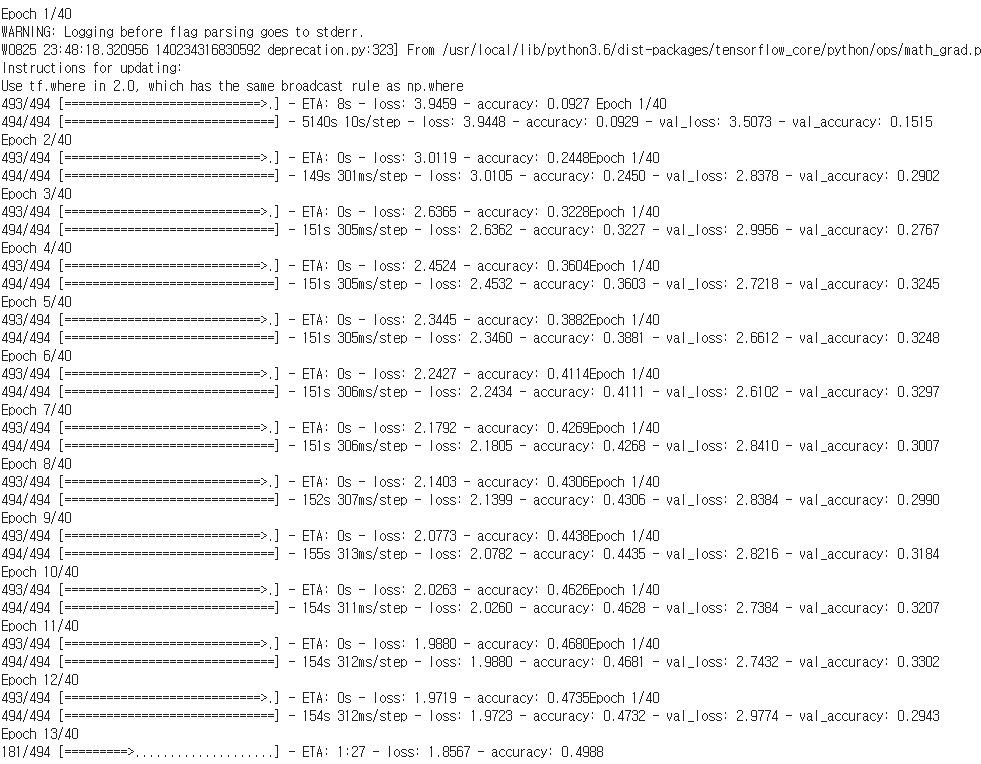
dense\_8 (Dense) (None, 73) 4745

=================================================================

Total params: 3,000,073

Trainable params: 742,089

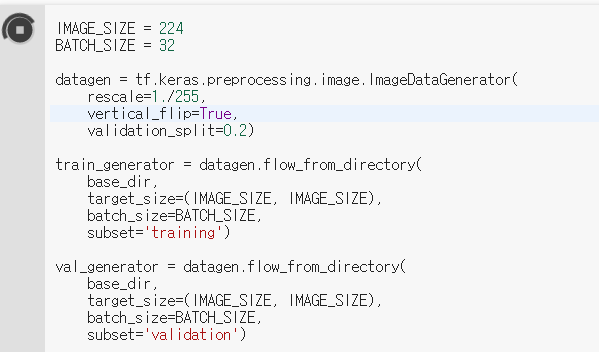
Non-trainable params: 2,257,984

****

**여전히…성능은 좋아지지 않는다ㅠ**

**<일곱번째 시도>**

**그래도 포기할 수 없다. Data augmentaion을 기존의 horizontal\_flip이 아니라 vertical flip으로 해주고, 여기에 다른 옵션을 추가하면 학습시간이 너무 오래 걸릴 것 같아 한가지 augmentaion만 적용해 주었다.**

****

****

Layer (type) Output Shape Param #

=================================================================

mobilenetv2\_1.00\_224 (Model) (None, 7, 7, 1280) 2257984

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d (Conv2D) (None, 5, 5, 32) 368672

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout (Dropout) (None, 5, 5, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

global\_average\_pooling2d (Gl (None, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

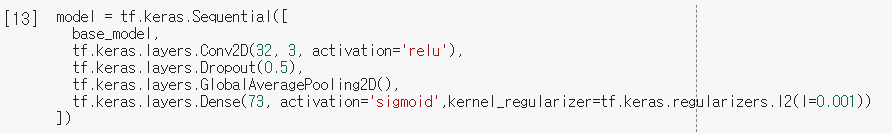
dense (Dense) (None, 73) 2409

=================================================================

Total params: 2,629,065

Trainable params: 371,081

Non-trainable params: 2,257,984

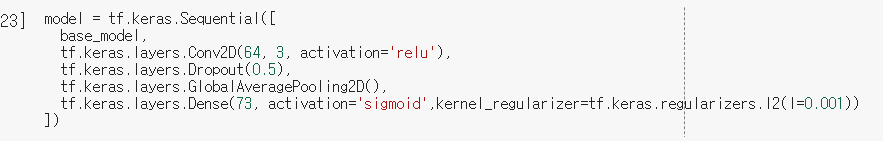
****

**파라미터가 너무 많아서 정확도가 낮게 나오나 싶어서 Conv2D의 depth를 64에서 32로 바꾸어주었다.**

****

**뭐가 문제일까… 일단 성능향상이 너무 느리게 된다.**

**<여덟번째 시도>**

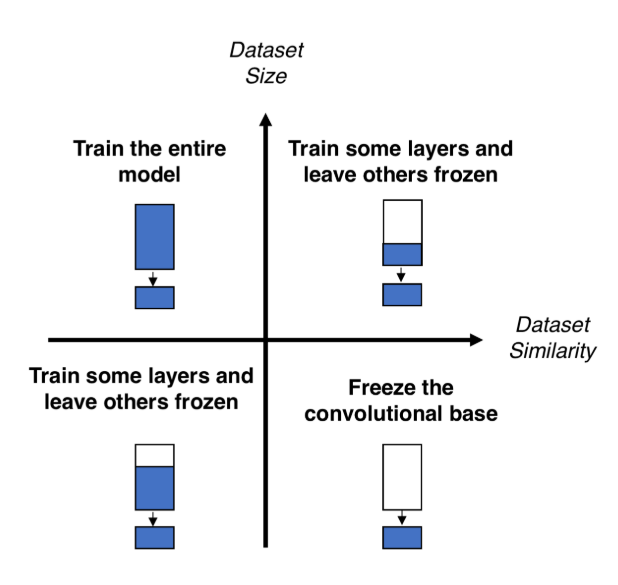
****

**32의 값을 다시 64로 돌리고 data augmenation을 없앴다.**

**같은 조건에서 예전에 Dropout이 0.2 였을때 학습속도는 빨랐다. 다만, val\_acc가 늘지 않았으므로 이번엔 Dropout만 0.5로 해주었다.**

**결과는….매한가지**

**<아홉번째 시도>**

****

**왜이렇게 성능이 안나오고, 학습속도가 느릴까 고민해보았다.**

**Optimize 함수 혹은 transfer learning의 특성에서 문제가 생긴 것이 아닌가 하는 생각이 들었다.**

**우선, optimize인 adam은 거의 모든 모델에서 일반적으로 좋은 함수라고 하였기에 일단은,**

**Dataset similarity가 다르다고 생각하고 기존에 freeze해서 모델을 학습시켰는데 일부 layer를unfreeze 해서 학습시켜보기로 하였다.**

**<안드로이드 스튜디오에 올리기>**

1. **Main 밖에 있는** build.gradle

dependencies {  
    implementation 'org.tensorflow:tensorflow-lite:0.0.0-nightly'  
}

android {  
    defaultConfig {  
        ndk {  
            abiFilters 'armeabi-v7a', 'arm64-v8a'  
        }  
    }  
}

**추가. (ndk Option)**

aaptOptions {  
 noCompress "tflite"  
}

**압축하면 파일 제대로 읽어들일 수 없음**

<https://pythonkim.tistory.com/133?category=703510>

1. **Bazel 설치(이거 하다가 포기ㅋㅋㅋ)**

**TensorflowLite 예제 활용**

<https://codelabs.developers.google.com/codelabs/recognize-flowers-with-tensorflow-on-android/?hl=ko#6>

protected String getModelPath() {  
 // you can download this file from  
 // see build.gradle for where to obtain this file. It should be auto  
 // downloaded into assets.  
 return "mymodel.tflite";  
}  
  
@Override  
protected String getLabelPath() {  
 return "labels.txt";  
}

**assets 폴더에 labels.txt와 mymodel.tflite 를 넣어준뒤 코드 변경**

<https://webnautes.tistory.com/1141>

<https://recipes4dev.tistory.com/145>

**에뮬레이터 설치**

* **오픈소스 구조**

**Tflite 폴더**

* **Classifier**
* **ClassifierFloatMobileNet**
* **ClassifierQuantizedMobileNet**
* **Classifier에서 Float 혹은 Quantiazed model을 부름**

**CameraActivity (UI)**

**CameraConnectionFragment**

**ClassifierActivity**

**LegacyCameraConnectionFragment**

<https://codelabs.developers.google.com/codelabs/recognize-flowers-with-tensorflow-on-android/?hl=ko#8> 모델작동방법

<https://codeasy.tistory.com/6>

안드로이드 기본 구조

-> 모델이 동작이 잘 안됨 (flower 기본 제공되어 있는 것 활용한 것도 실패..)

-> 데이터의 양이 부족해서가 아님. 모델을 올리는 과정에서 문제가 있는 것으로 보임

-> 내 코드로 변경했었는데 연동과정에서 다른점이 있어서 원래 모델의 코드를 기반으로 model.fit -> model.fit\_generator 로 변경하여 flower 모델을 다시 트레이닝 시킨 결과, 정확도가 높아짐.

-> 파라미터 설정은 유지한 채로, trainX,trainY 생성 방식을 image generator을 사용하여 만드는 것으로 변경

->TIP) GPU 용량에 한계가 있기 때문에 batch\_size를 작게 하거나, 구글 코랩을 이용하거나, (로컬의 주피터에서는 실행이 안되는 경우도 존재. 실행환경이 다르기 때문)