Transfer_Learning_assignment

May 5, 2021

0.0.1 Model-1

0.0.2 Model-2

0.0.3 Model-3

[ ]:

```
!curl --header "Host: doc-10-84-docs.googleusercontent.com" --header␣
 ↪"User-Agent: Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36␣
 ↪(KHTML, like Gecko) Chrome/83.0.4103.61 Safari/537.36" --header "Accept:␣
 ↪text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/
 ↪apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9" --header␣
 ↪"Accept-Language: en-IN,en-US;q=0.9,en;q=0.8" --header "Referer: https://
 ↪drive.google.com/drive/folders/1eY4pTqtGtXSAYVzaHwWAK2Cq4bHb1Zkq" --header␣
 ↪"Cookie: AUTH_p6auoatqfs07jahuj61ftbh1fhhqke22_nonce=91cot1ctonov8" --header␣
 ↪"Connection: keep-alive" "https://doc-10-84-docs.googleusercontent.com/docs/
 ↪securesc/hefrtae5da7uudg7909tvuvbvujep2jn/qss3i0qi2dq1k9h29322n5vgekgr8vip/
 ↪1591416900000/00484516897554883881/01214842798349466566/
 ↪1Z4TyI7FcFVEx8qdl4jO9qxvxaqLSqoEu?
 ↪e=download&authuser=0&nonce=91cot1ctonov8&user=01214842798349466566&hash=32f1oj749qld2aefoi
 ↪-L -o 'rvl-cdip.rar'
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 4444M    0 4444M    0     0  50.8M      0 --:--:--  0:01:27 --:--:-- 42.8M
```

[ ]:

```
get_ipython().system_raw("unrar x rvl-cdip.rar")
```

[ ]:

```
!ls
```

```
data_final  labels_final.csv  rvl-cdip.rar  sample_data
```

[ ]:

```
import tensorflow as tf
import os
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import␣
 ↪Dense,Input,Conv2D,MaxPool2D,Activation,Dropout,Flatten
```

```
from tensorflow.keras.models import Model,Sequential
import math
from tensorflow.keras.callbacks import TensorBoard
```

[ ]:
```
from tensorflow.keras.utils import plot_model
```

[ ]:
```
data = pd.read_csv("labels_final.csv")
data.head()
```

[ ]:
```
                                        path  label
0  imagesv/v/o/h/voh71d00/509132755+-2755.tif      3
1        imagesl/l/x/t/lxt19d00/502213303.tif      3
2       imagesx/x/e/d/xed05a00/2075325674.tif      2
3  imageso/o/j/b/ojb60d00/517511301+-1301.tif      3
4       imagesq/q/z/k/qzk17e00/2031320195.tif      7
```

[ ]:
```
data['label']= data['label'].astype(str)
```

[ ]:
```
train,test = train_test_split(data,test_size=0.33,random_state = 42)
```

[ ]:
```
print('train_size',train.shape)
print('test_size',test.shape)
```

```
train_size (32160, 2)
test_size (15840, 2)
```

[ ]:
```
tf.__version__
```

[ ]:
```
'2.2.0'
```

[ ]:
```
###Image data Generator class
datagen =tf.keras.preprocessing.image.ImageDataGenerator()

##We are fitting the data to Image data generator.
ImageGenerator_train = datagen.flow_from_dataframe(dataframe =train␣
 ↪,directory="data_final",x_col="path",
                                                y_col = "label",
                                                class_mode = "categorical",
                                               ␣
 ↪target_size=(224,224),batch_size=64)
```

```
Found 32160 validated image filenames belonging to 16 classes.
```

[ ]:
```
##We are fitting the data to Image data generator.
ImageGenerator_test = datagen.flow_from_dataframe(dataframe =test␣
 ↪,directory="data_final",x_col="path",
                                                y_col = "label",
                                                class_mode = "categorical",
                                               ␣
↪target_size=(224,224),batch_size=64)
```

```
Found 15840 validated image filenames belonging to 16 classes.
```

[ ]:
```
print(type(ImageGenerator_train))
```

```
<class 'keras_preprocessing.image.dataframe_iterator.DataFrameIterator'>
```

Model 1

[ ]:
```
%load_ext tensorboard
```

[ ]:
```
IMG_SHAPE = (224, 224, 3)
```

[ ]:
```
VGG16_model =VGG16(include_top=False,weights="imagenet",
                                       input_shape=IMG_SHAPE)
```

[ ]:
```
c = 0
for layers in VGG16_model.layers:
  layers.trainable = False
  c = c+1
```

[ ]:
```
VGG16_model.summary()
```

```
Model: "vgg16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 224, 224, 3)]     0

_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792

_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928

_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0

_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856

_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584

_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0

_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168

_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080

_________________________________________________________________
```

```
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
=================================================================
Total params: 14,714,688
Trainable params: 0
Non-trainable params: 14,714,688
_________________________________________________________________
```

[ ]:

```
[ ]:
```

```
[ ]: x = VGG16_model.get_layer('block5_pool').output
     x =␣
      ↪Conv2D(filters=32,kernel_size=(3,3),strides=(1,1),padding='valid',activation='relu',kernel_
      ↪keras.initializers.he_normal(seed=0),name='Conv1')(x)
     x = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='valid',name='pool1')(x)
     x = Flatten(name='Flatten')(x)
     x = Dense(units=1024,activation='relu',kernel_initializer=tf.keras.initializers.
      ↪he_normal(seed=10),name='FC1')(x)
     x = Dense(units=514,activation='relu',kernel_initializer=tf.keras.initializers.
      ↪he_normal(seed=10),name='FC2')(x)
     output_layer = Dense(units=16,activation='softmax',kernel_initializer=tf.keras.
      ↪initializers.glorot_normal(seed=5),name='output_layer')(x)

     model1 = Model(inputs=VGG16_model.input,outputs=output_layer )
```

```
[ ]: model1.summary()
```

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
```

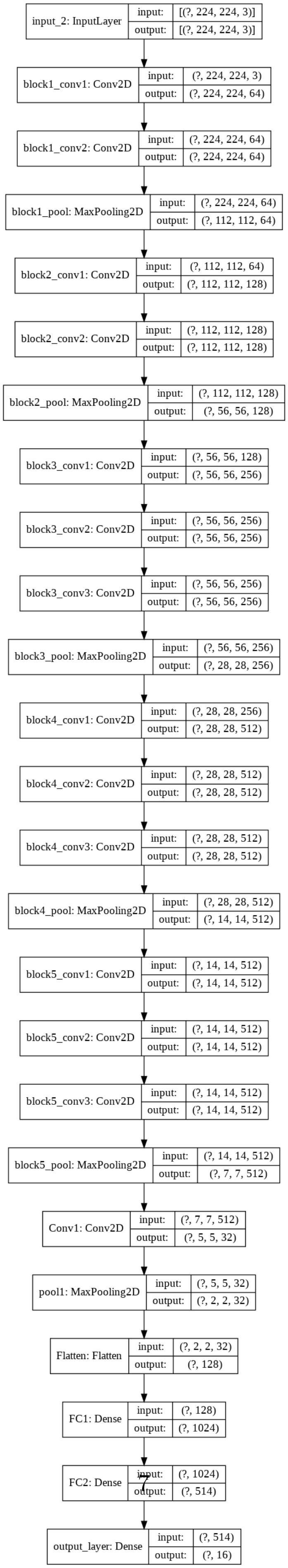
```
=================================================================
input_2 (InputLayer)         [(None, 224, 224, 3)]     0

_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792

_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928

_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0

_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856

_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584

_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0

_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168

_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080

_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080

_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0

_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160

_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808

_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808

_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0

_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808

_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808

_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808

_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0

_________________________________________________________________
Conv1 (Conv2D)               (None, 5, 5, 32)          147488

_________________________________________________________________
pool1 (MaxPooling2D)         (None, 2, 2, 32)          0

_________________________________________________________________
Flatten (Flatten)            (None, 128)               0

_________________________________________________________________
FC1 (Dense)                  (None, 1024)              132096

_________________________________________________________________
FC2 (Dense)                  (None, 514)               526850
```

```
_________________________________________________________________
output_layer (Dense)         (None, 16)                8240
=================================================================
Total params: 15,529,362
Trainable params: 814,674
Non-trainable params: 14,714,688
_________________________________________________________________
```

[ ]:

```
plot_model(model1,to_file='model1.png',show_shapes=True,show_layer_names=True,
           dpi=96)
```

[ ]:


input_2: InputLayer
input: [(?, 224, 224, 3)]
output: [(?, 224, 224, 3)]
block1_conv1: Conv2D
input: (?, 224, 224, 3)
output: (?, 224, 224, 64)
block1_conv2: Conv2D
input: (?, 224, 224, 64)
output: (?, 224, 224, 64)
block1_pool: MaxPooling2D
input: (?, 224, 224, 64)
output: (?, 112, 112, 64)
block2_conv1: Conv2D
input: (?, 112, 112, 64)
output: (?, 112, 112, 128)
block2_conv2: Conv2D
input: (?, 112, 112, 128)
output: (?, 112, 112, 128)
block2_pool: MaxPooling2D
input: (?, 112, 112, 128)
output: (?, 56, 56, 128)
block3_conv1: Conv2D
input: (?, 56, 56, 128)
output: (?, 56, 56, 256)
block3_conv2: Conv2D
input: (?, 56, 56, 256)
output: (?, 56, 56, 256)
block3_conv3: Conv2D
input: (?, 56, 56, 256)
output: (?, 56, 56, 256)
block3_pool: MaxPooling2D
input: (?, 56, 56, 256)
output: (?, 28, 28, 256)
block4_conv1: Conv2D
input: (?, 28, 28, 256)
output: (?, 28, 28, 512)
block4_conv2: Conv2D
input: (?, 28, 28, 512)
output: (?, 28, 28, 512)
block4_conv3: Conv2D
input: (?, 28, 28, 512)
output: (?, 28, 28, 512)
block4_pool: MaxPooling2D
input: (?, 28, 28, 512)
output: (?, 14, 14, 512)
block5_conv1: Conv2D
input: (?, 14, 14, 512)
output: (?, 14, 14, 512)
block5_conv2: Conv2D
input: (?, 14, 14, 512)
output: (?, 14, 14, 512)
block5_conv3: Conv2D
input: (?, 14, 14, 512)
output: (?, 14, 14, 512)
block5_pool: MaxPooling2D
input: (?, 14, 14, 512)
output: (?, 7, 7, 512)
Conv1: Conv2D
input: (?, 7, 7, 512)
output: (?, 5, 5, 32)
pool1: MaxPooling2D
input: (?, 5, 5, 32)
output: (?, 2, 2, 32)
Flatten: Flatten
input: (?, 2, 2, 32)
output: (?, 128)
FC1: Dense
input: (?, 128)
output: (?, 1024)
FC2: Dense
input: (?, 1024)
output: (?, 514)
output_layer: Dense
input: (?, 514)
output: (?, 16)

[ ]:

```
batch_size_points = math.ceil(32160/64)
print(batch_size_points)
```

```
503
```

[ ]:

```
15840/64
```

[ ]:

```
247.5
```

[ ]:

```
log_dir = "/content/logs/"
tensorboard_callback =  TensorBoard(log_dir=log_dir,histogram_freq =␣
 ↪1,write_graph = True,write_grads=True)
```

```
WARNING:tensorflow:`write_grads` will be ignored in TensorFlow 2.0 for the
`TensorBoard` Callback.
```

[ ]:

```
model1.compile(optimizer=tf.keras.optimizers.Adam(lr=0.
 ↪0001),loss='categorical_crossentropy',metrics=['accuracy'])
```

[ ]:

```
model1.
 ↪fit_generator(ImageGenerator_train,validation_data=ImageGenerator_test,steps_per_epoch=503,
```

```
Epoch 1/3
503/503 [==============================] - 269s 535ms/step - loss: 1.3102 -
accuracy: 0.5972 - val_loss: 1.6547 - val_accuracy: 0.5275
Epoch 2/3
503/503 [==============================] - 260s 516ms/step - loss: 1.1439 -
accuracy: 0.6428 - val_loss: 1.5810 - val_accuracy: 0.5539
Epoch 3/3
503/503 [==============================] - 269s 535ms/step - loss: 1.0130 -
accuracy: 0.6805 - val_loss: 1.5590 - val_accuracy: 0.5669
```

[ ]:

```
<tensorflow.python.keras.callbacks.History at 0x7f4918164668>
```

[ ]:

```
%tensorboard --logdir logs
```

```
<IPython.core.display.Javascript object>
```

Observation

Using bottle feature just gives accuracy of 52% , it means dataset is not similar to imagenet or some more tunning is required

Model 2

```
[ ]: VGG16_model2 =VGG16(include_top=False,weights="imagenet",
                                         input_shape=IMG_SHAPE)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-
applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58892288/58889256 [==============================] - 1s 0us/step
```

```
[ ]: for layers in VGG16_model2.layers:
       layers.trainable= False
```

```
[ ]: VGG16_model2.summary()
```

```
Model: "vgg16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
```

```
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
=================================================================
Total params: 14,714,688
Trainable params: 0
Non-trainable params: 14,714,688
_________________________________________________________________
```

[ ]:

```
x = VGG16_model2.get_layer('block5_pool').output

x =␣
 ↪Conv2D(4096,kernel_size=[7,7],strides=(1,1),padding='valid',activation='relu')(x)
x =␣
 ↪Conv2D(4096,kernel_size=[1,1],strides=(1,1),padding='valid',activation='relu')(x)
x = Flatten(name='flatten')(x)

output_layer = Dense(units=16,activation='softmax',kernel_initializer=tf.keras.
 ↪initializers.glorot_normal(seed=5),name='output_layer')(x)



model2 = Model(inputs=VGG16_model2.input,outputs=output_layer )
```

[ ]:

```
model2.summary()
```

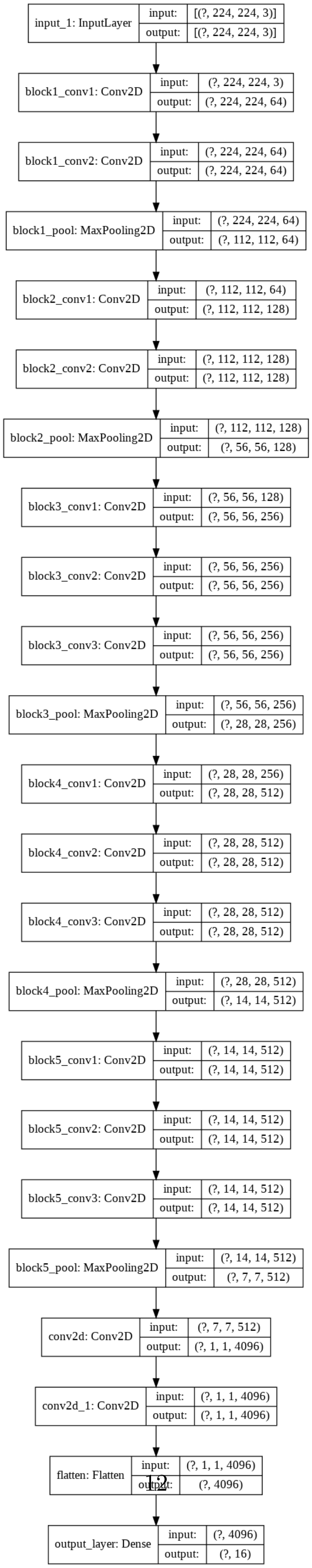
```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
```

```
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
conv2d (Conv2D)              (None, 1, 1, 4096)        102764544
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 1, 1, 4096)        16781312
_________________________________________________________________
flatten (Flatten)            (None, 4096)              0
_________________________________________________________________
output_layer (Dense)         (None, 16)                65552
=================================================================
Total params: 134,326,096
Trainable params: 119,611,408
Non-trainable params: 14,714,688
_________________________________________________________________
```

[ ]:

```
plot_model(model2,to_file='model1.png',show_shapes=True,show_layer_names=True,
           dpi=96)
```

[ ]:


input_1: InputLayer | input: | [(?, 224, 224, 3)] | output: | [(?, 224, 224, 3)]
block1_conv1: Conv2D | input: | (?, 224, 224, 3) | output: | (?, 224, 224, 64)
block1_conv2: Conv2D | input: | (?, 224, 224, 64) | output: | (?, 224, 224, 64)
block1_pool: MaxPooling2D | input: | (?, 224, 224, 64) | output: | (?, 112, 112, 64)
block2_conv1: Conv2D | input: | (?, 112, 112, 64) | output: | (?, 112, 112, 128)
block2_conv2: Conv2D | input: | (?, 112, 112, 128) | output: | (?, 112, 112, 128)
block2_pool: MaxPooling2D | input: | (?, 112, 112, 128) | output: | (?, 56, 56, 128)
block3_conv1: Conv2D | input: | (?, 56, 56, 128) | output: | (?, 56, 56, 256)
block3_conv2: Conv2D | input: | (?, 56, 56, 256) | output: | (?, 56, 56, 256)
block3_conv3: Conv2D | input: | (?, 56, 56, 256) | output: | (?, 56, 56, 256)
block3_pool: MaxPooling2D | input: | (?, 56, 56, 256) | output: | (?, 28, 28, 256)
block4_conv1: Conv2D | input: | (?, 28, 28, 256) | output: | (?, 28, 28, 512)
block4_conv2: Conv2D | input: | (?, 28, 28, 512) | output: | (?, 28, 28, 512)
block4_conv3: Conv2D | input: | (?, 28, 28, 512) | output: | (?, 28, 28, 512)
block4_pool: MaxPooling2D | input: | (?, 28, 28, 512) | output: | (?, 14, 14, 512)
block5_conv1: Conv2D | input: | (?, 14, 14, 512) | output: | (?, 14, 14, 512)
block5_conv2: Conv2D | input: | (?, 14, 14, 512) | output: | (?, 14, 14, 512)
block5_conv3: Conv2D | input: | (?, 14, 14, 512) | output: | (?, 14, 14, 512)
block5_pool: MaxPooling2D | input: | (?, 14, 14, 512) | output: | (?, 7, 7, 512)
conv2d: Conv2D | input: | (?, 7, 7, 512) | output: | (?, 1, 1, 4096)
conv2d_1: Conv2D | input: | (?, 1, 1, 4096) | output: | (?, 1, 1, 4096)
flatten: Flatten | input: | (?, 1, 1, 4096) | output: | (?, 4096)
output_layer: Dense | input: | (?, 4096) | output: | (?, 16)

[ ]:
```
mkdir logs
```

[ ]:
```
log_dir = "logs"
tensorboard_callback␣
 ↪=TensorBoard(log_dir=log_dir,histogram_freq=1,write_graph=True,write_grads=True)
```

```
WARNING:tensorflow:`write_grads` will be ignored in TensorFlow 2.0 for the
`TensorBoard` Callback.
```

[ ]:
```
model2.compile(optimizer=tf.keras.optimizers.Adam(lr=0.
 ↪001),loss='categorical_crossentropy',metrics=['accuracy'])
model2.fit_generator(ImageGenerator_train,validation_data␣
 ↪=ImageGenerator_test,steps_per_epoch=503,validation_steps = 247,␣
 ↪epochs=3,callbacks=[tensorboard_callback])
```

```
WARNING:tensorflow:From <ipython-input-40-3bef14f92f4f>:2: Model.fit_generator
(from tensorflow.python.keras.engine.training) is deprecated and will be removed
in a future version.
Instructions for updating:
Please use Model.fit, which supports generators.
Epoch 1/3
503/503 [==============================] - 336s 668ms/step - loss: 5.0102 -
accuracy: 0.4744 - val_loss: 1.6662 - val_accuracy: 0.5326
Epoch 2/3
503/503 [==============================] - 335s 665ms/step - loss: 1.3930 -
accuracy: 0.5907 - val_loss: 1.2908 - val_accuracy: 0.5969
Epoch 3/3
503/503 [==============================] - 341s 678ms/step - loss: 1.1142 -
accuracy: 0.6569 - val_loss: 1.2092 - val_accuracy: 0.6301
```

[ ]:
```
<tensorflow.python.keras.callbacks.History at 0x7f18900cdc18>
```

[ ]:
```
!kill
```

[ ]:
```
%tensorboard --logdir logs
```

```
<IPython.core.display.Javascript object>
```

[ ]:
```
Here we are using covnet as
```

Observation

1) Here we are using FC as covnet which and trainning only new layers , But still we are not very good accuracy , it means dataset is different than imagenet dataset , and need to train some more layers .

Model 3

[ ]:
```
VGG16_model3 =VGG16(include_top=False,weights="imagenet",
                                      input_shape=IMG_SHAPE)
```

[ ]:
```
for layers in VGG16_model3.layers[:-6]:
  layers.trainable = False
```

[ ]:
```
VGG16_model3.summary()
```

```
Model: "vgg16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
```

```
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
=================================================================
Total params: 14,714,688
Trainable params: 9,439,232
Non-trainable params: 5,275,456
_________________________________________________________________
```

```
[ ]: x = VGG16_model3.get_layer('block5_pool').output

     x =␣
      ↪Conv2D(4096,kernel_size=[7,7],strides=(1,1),padding='valid',activation='relu')(x)
     x =␣
      ↪Conv2D(4096,kernel_size=[1,1],strides=(1,1),padding='valid',activation='relu')(x)
     x = Flatten(name='flatten')(x)

     output_layer = Dense(units=16,activation='softmax',kernel_initializer=tf.keras.
      ↪initializers.glorot_normal(seed=5),name='output_layer')(x)




     model3 = Model(inputs=VGG16_model3.input,outputs=output_layer )
```

```
[ ]: model3.summary()
```

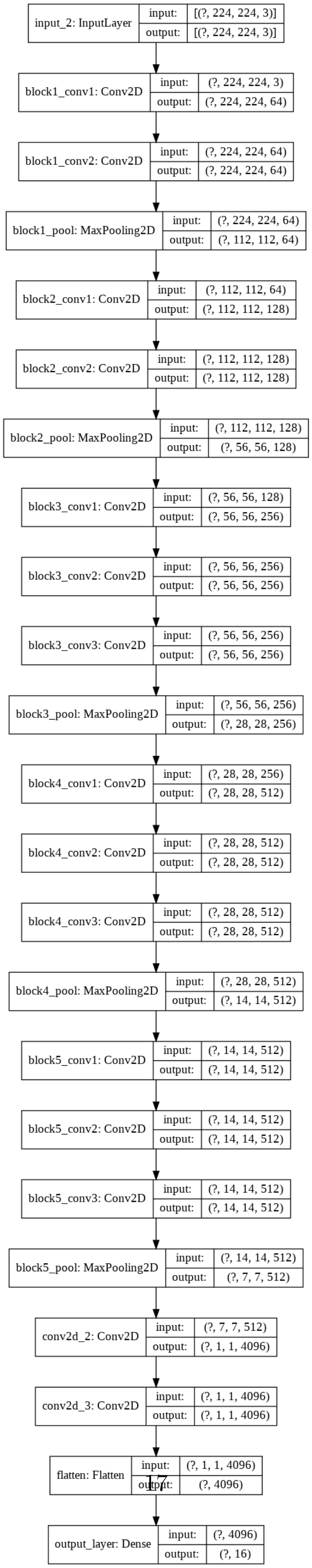
```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
```

```
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080

_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0

_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160

_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808

_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808

_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0

_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808

_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808

_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808

_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0

_________________________________________________________________
conv2d_2 (Conv2D)            (None, 1, 1, 4096)        102764544

_________________________________________________________________
conv2d_3 (Conv2D)            (None, 1, 1, 4096)        16781312

_________________________________________________________________
flatten (Flatten)            (None, 4096)              0

_________________________________________________________________
output_layer (Dense)         (None, 16)                65552
=================================================================
Total params: 134,326,096
Trainable params: 129,050,640
Non-trainable params: 5,275,456

_________________________________________________________________
```

[ ]:

```
plot_model(model3,to_file='model1.png',show_shapes=True,show_layer_names=True,
           dpi=96)
```

[ ]:


input_2: InputLayer | input: [(?, 224, 224, 3)] | output: [(?, 224, 224, 3)]
block1_conv1: Conv2D | input: (?, 224, 224, 3) | output: (?, 224, 224, 64)
block1_conv2: Conv2D | input: (?, 224, 224, 64) | output: (?, 224, 224, 64)
block1_pool: MaxPooling2D | input: (?, 224, 224, 64) | output: (?, 112, 112, 64)
block2_conv1: Conv2D | input: (?, 112, 112, 64) | output: (?, 112, 112, 128)
block2_conv2: Conv2D | input: (?, 112, 112, 128) | output: (?, 112, 112, 128)
block2_pool: MaxPooling2D | input: (?, 112, 112, 128) | output: (?, 56, 56, 128)
block3_conv1: Conv2D | input: (?, 56, 56, 128) | output: (?, 56, 56, 256)
block3_conv2: Conv2D | input: (?, 56, 56, 256) | output: (?, 56, 56, 256)
block3_conv3: Conv2D | input: (?, 56, 56, 256) | output: (?, 56, 56, 256)
block3_pool: MaxPooling2D | input: (?, 56, 56, 256) | output: (?, 28, 28, 256)
block4_conv1: Conv2D | input: (?, 28, 28, 256) | output: (?, 28, 28, 512)
block4_conv2: Conv2D | input: (?, 28, 28, 512) | output: (?, 28, 28, 512)
block4_conv3: Conv2D | input: (?, 28, 28, 512) | output: (?, 28, 28, 512)
block4_pool: MaxPooling2D | input: (?, 28, 28, 512) | output: (?, 14, 14, 512)
block5_conv1: Conv2D | input: (?, 14, 14, 512) | output: (?, 14, 14, 512)
block5_conv2: Conv2D | input: (?, 14, 14, 512) | output: (?, 14, 14, 512)
block5_conv3: Conv2D | input: (?, 14, 14, 512) | output: (?, 14, 14, 512)
block5_pool: MaxPooling2D | input: (?, 14, 14, 512) | output: (?, 7, 7, 512)
conv2d_2: Conv2D | input: (?, 7, 7, 512) | output: (?, 1, 1, 4096)
conv2d_3: Conv2D | input: (?, 1, 1, 4096) | output: (?, 1, 1, 4096)
flatten: Flatten | input: (?, 1, 1, 4096) | output: (?, 4096)
output_layer: Dense | input: (?, 4096) | output: (?, 16)

[ ]:

```
rm -r /content/logs/
```

[ ]:

```
mkdir logs
```

[ ]:

```
log_dir = "logs"
tensorboard_callback = TensorBoard(log_dir =␣
 ↪log_dir,histogram_freq=1,write_graph = True,write_grads = True)
```

```
WARNING:tensorflow:`write_grads` will be ignored in TensorFlow 2.0 for the
`TensorBoard` Callback.
```

[ ]:

```
model3.compile(optimizer=tf.keras.optimizers.SGD(learning_rate=0.
 ↪0001,momentum=0.
 ↪9,nesterov=True),loss='categorical_crossentropy',metrics=['accuracy'])
model3.fit_generator(ImageGenerator_train,validation_data =␣
 ↪ImageGenerator_test,steps_per_epoch=503,validation_steps  =␣
 ↪247,epochs=3,callbacks = [tensorboard_callback])
```

```
Epoch 1/3
503/503 [==============================] - 520s 1s/step - loss: 1.2265 -
accuracy: 0.6345 - val_loss: 0.9167 - val_accuracy: 0.7251
Epoch 2/3
503/503 [==============================] - 527s 1s/step - loss: 0.5933 -
accuracy: 0.8200 - val_loss: 0.8845 - val_accuracy: 0.7451
Epoch 3/3
503/503 [==============================] - 519s 1s/step - loss: 0.2931 -
accuracy: 0.9156 - val_loss: 0.8730 - val_accuracy: 0.7597
```

[ ]:

```
<tensorflow.python.keras.callbacks.History at 0x7f18426bd8d0>
```

[ ]:

```
!kill
```

```
kill: usage: kill [-s sigspec | -n signum | -sigspec] pid | jobspec … or kill
-l [sigspec]
```

[ ]:

```
%tensorboard --logdir logs
```

```
<IPython.core.display.Javascript object>
```

[ ]:

```

```

Observation

1) Here we are making same model as model 2 but here we are trainning new layers + Top most 6 layers and we are getting accuracy of 91% ,which means

data is very different than imagnet .