

```
In [ ]: !wget --header="Host: storage.googleapis.com" --header="User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit
```

```
In [ ]: !unzip /content/TL_DATA
```

```
In [3]: import tensorflow as tf
import os
import numpy as np
import pandas as pd
from keras.models import Sequential
from keras import applications
from tensorflow.keras.layers import Dense, Conv2D, MaxPool2D, Activation, Dropout, Flatten, GlobalAveragePooling2D
from tensorflow.keras import Input
import random as rn
from tensorflow.keras.models import Model
from keras.callbacks import Callback, EarlyStopping
import datetime
```

```
In [7]: #https://vijayabhaskar96.medium.com/tutorial-on-keras-flow-from-dataframe-1fd4493d237c
```

```
traindf=pd.read_csv('/content/labels_final.csv', dtype=str)

def append_content_dataFinal(inp):
    return "/content/data_final/"+inp

traindf["path"]=traindf["path"].apply(append_content_dataFinal)

from keras_preprocessing.image import ImageDataGenerator
datagen = ImageDataGenerator(rescale=1./255., validation_split=0.25
                             zoom_range=[0.5,1.0],
                             brightness_range=[0.2,1.0],
                             rotation_range = 90,
                             horizontal_flip = True
                             )
```

```
In [8]: #Train Test split
```

```
train_generator = datagen.flow_from_dataframe(dataframe=traindf, directory=None, x_col="path", y_col="label", subset="train",
class_mode="categorical", target_size=(224,224))

val_generator = datagen.flow_from_dataframe(dataframe=traindf, directory=None, x_col="path", y_col="label", subset="validation",
class_mode="categorical", target_size=(224,224))
```

Found 36000 validated image filenames belonging to 16 classes.

Found 12000 validated image filenames belonging to 16 classes.

Model 1

```
In [ ]: basemodel = applications.VGG16(include_top=False, weights='imagenet')
        basemodel.trainable = False

        input_layer = Input(shape=(224,224,3),name='Input_Layer')

        x = basemodel(input_layer , training = False)

        Conv1 = Conv2D(filters=128,kernel_size=(3,3),strides=(2,2),padding='valid', activation='relu',kernel_initializer=tf.keras
        Pool1 = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='valid')(Conv1)
        flat = Flatten()(Pool1)
        FC1 = Dense(units=64,activation='relu',kernel_initializer=tf.keras.initializers.glorot_normal(seed=3))(flat)
        FC2 = Dense(units=32,activation='relu',kernel_initializer=tf.keras.initializers.glorot_normal(seed=3))(FC1)
        Out = Dense(units=16,activation='softmax',kernel_initializer=tf.keras.initializers.glorot_normal(seed=3))(FC2)

        finalmodel = Model(input_layer , Out)

        finalmodel.compile(optimizer=tf.keras.optimizers.Adam(lr=0.01),loss='categorical_crossentropy',metrics=['accuracy'])
        tf.keras.backend.clear_session()

        logdir = os.path.join("logs", datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))

        earllystop = EarlyStopping (monitor='val_accuracy', min_delta=0.01, patience=1, verbose=1 , mode = 'auto', restore_best_we
        tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=logdir,histogram_freq=1, write_graph=True)

        finalmodel.fit(train_generator,epochs=10,steps_per_epoch=None, callbacks = [tensorboard_callback,earllystop],validation_da

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf
_kernels_notop.h5
58892288/58889256 [=====] - 3s 0us/step
Epoch 1/10
563/563 [=====] - 283s 489ms/step - loss: 2.2406 - accuracy: 0.2977 - val_loss: 1.4707 - val_acc
uracy: 0.5293
Epoch 2/10
563/563 [=====] - 257s 456ms/step - loss: 1.4116 - accuracy: 0.5497 - val_loss: 1.4510 - val_acc
uracy: 0.5537
Epoch 3/10
563/563 [=====] - 226s 401ms/step - loss: 1.2681 - accuracy: 0.6016 - val_loss: 1.2882 - val_acc
uracy: 0.6039
Epoch 4/10
```

```
563/563 [=====] - 204s 362ms/step - loss: 1.1677 - accuracy: 0.6384 - val_loss: 1.2739 - val_acc
uracy: 0.6206
Epoch 5/10
563/563 [=====] - 184s 326ms/step - loss: 1.1071 - accuracy: 0.6560 - val_loss: 1.2838 - val_acc
uracy: 0.6228
Restoring model weights from the end of the best epoch.
Epoch 00005: early stopping
```

```
Out[ ]: <tensorflow.python.keras.callbacks.History at 0x7fc6c01fdd30>
```

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

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

Output hidden; open in <https://colab.research.google.com> to view.

Model 2

```
In [11]: basemodel = applications.VGG16(include_top=False, weights='imagenet')
          basemodel.trainable = False

          input_layer = Input(shape=(224,224,3),name='Input_Layer')

          x = basemodel(input_layer , training = False)

          # for layer in vgg_conv.layers[:]:
          #     layer.trainable = False
          # for layer in basemodel.layers:
          #     print(layer, layer.trainable)

          #https://stackoverflow.com/questions/37213388/keras-accuracy-does-not-change
          adam = tf.keras.optimizers.Adam(learning_rate=0.0001)

          Conv1 = Conv2D(filters=4096,kernel_size=7,strides=1,padding='valid', activation='relu',kernel_initializer=tf.keras.initia
          Conv2 = Conv2D(filters=4096,kernel_size=1,strides=1,padding='valid' , activation='relu',kernel_initializer=tf.keras.initi
          flat = Flatten()(Conv2)
          Out = Dense(units=16,activation='softmax')(flat)

          finalmodel = Model(input_layer , Out)

          finalmodel.compile(optimizer=adam,loss='categorical_crossentropy',metrics=['accuracy'])

          tf.keras.backend.clear_session()

          logdir = os.path.join("logs", datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
```

```

earlystop = EarlyStopping(monitor='val_accuracy', min_delta=0.01, patience=1, verbose=1, mode = 'max')
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=logdir, histogram_freq=1, write_graph=True)

finalmodel.fit(train_generator, epochs=20, steps_per_epoch=None, callbacks = [tensorboard_callback, earlystop], validation_data=

```

```

Epoch 1/20
563/563 [=====] - 264s 468ms/step - loss: 75.2200 - accuracy: 0.5078 - val_loss: 8.2673 - val_accuracy: 0.6407
Epoch 2/20
563/563 [=====] - 227s 404ms/step - loss: 6.0428 - accuracy: 0.6599 - val_loss: 2.6923 - val_accuracy: 0.6889
Epoch 3/20
563/563 [=====] - 233s 413ms/step - loss: 2.3906 - accuracy: 0.6721 - val_loss: 1.7650 - val_accuracy: 0.6793
Epoch 00003: early stopping

```

Out[11]: <tensorflow.python.keras.callbacks.History at 0x7f0c0cc80cf8>

```

In [12]: %load_ext tensorboard
         %tensorboard --logdir logs

```

Output hidden; open in <https://colab.research.google.com> to view.

Model 3

```

In [16]: basemodel = applications.VGG16(include_top=False, weights='imagenet')
         basemodel.trainable = False

         input_layer = Input(shape=(224,224,3), name='Input_Layer')

         x = basemodel(input_layer, training = False)

         for layer in basemodel.layers[:]:
             layer.trainable = False

         #setting trainable = true for last 6 layers of VGG16
         for layer in basemodel.layers[13:]:
             layer.trainable = True

         for layer in basemodel.layers:
             print(layer, layer.trainable)

         adam = tf.keras.optimizers.Adam(learning_rate=0.0001)

```

```

Conv1 = Conv2D(filters=4096,kernel_size=7,strides=1,padding='valid', activation='relu',kernel_initializer=tf.keras.initializer.RandomNormal(stddev=0.01))
Conv2 = Conv2D(filters=4096,kernel_size=1,strides=1,padding='valid', activation='relu',kernel_initializer=tf.keras.initializer.RandomNormal(stddev=0.01))
flat = Flatten()(Conv2)
Out = Dense(units=16,activation='softmax')(flat)

finalmodel = Model(input_layer , Out)

finalmodel.compile(optimizer=adam,loss='categorical_crossentropy',metrics=['accuracy'])

tf.keras.backend.clear_session()

logdir = os.path.join("logs", datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))

earlystop = EarlyStopping(monitor='val_accuracy', min_delta=0.01, patience=1, verbose=1, mode = 'max')
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=logdir,histogram_freq=1, write_graph=True)

finalmodel.fit(train_generator,epochs=20,steps_per_epoch=None,callbacks = [tensorboard_callback,earlystop],validation_data=val_generator)

<tensorflow.python.keras.engine.input_layer.InputLayer object at 0x7f0d153d0e10> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7f0d153ca198> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7f0c0df2d160> False
<tensorflow.python.keras.layers.pooling.MaxPooling2D object at 0x7f0d15474828> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7f0d1544af28> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7f0d153ca1d0> False
<tensorflow.python.keras.layers.pooling.MaxPooling2D object at 0x7f0d153b7438> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7f08eaf91630> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7f0d153caa58> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7f0d15464f60> False
<tensorflow.python.keras.layers.pooling.MaxPooling2D object at 0x7f0d153d3160> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7f0d153dcb70> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7f0d153dc7b8> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7f0d153dc828> True
<tensorflow.python.keras.layers.pooling.MaxPooling2D object at 0x7f0d153747f0> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7f0d153749e8> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7f0d1537e7f0> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7f0d1537eeb8> True
<tensorflow.python.keras.layers.pooling.MaxPooling2D object at 0x7f0d153dc588> True
Epoch 1/20
563/563 [=====] - 270s 477ms/step - loss: 74.6087 - accuracy: 0.5221 - val_loss: 7.9329 - val_accuracy: 0.6722
Epoch 2/20
563/563 [=====] - 255s 453ms/step - loss: 5.8349 - accuracy: 0.6565 - val_loss: 2.6198 - val_accuracy: 0.6871
Epoch 3/20
563/563 [=====] - 267s 474ms/step - loss: 2.3357 - accuracy: 0.6716 - val_loss: 1.7885 - val_accuracy: 0.6672
Epoch 00003: early stopping
<tensorflow.python.keras.callbacks.History object at 0x7f0d1539ad30>

```

Out[16]:

```
In [18]: %load_ext tensorboard  
          %tensorboard --logdir logs
```

Output hidden; open in <https://colab.research.google.com> to view.