ML_Caltech256_single_VGG16-Data_Agumentation

March 28, 2020

1 Final project Caltech256

– Network used VGG16

- **Pretrained VGG model (using weights of imageNet)**

Student Name/ID : Aloukik aditya(1115290) , Sarthak Rawat(1101124)

[1]:
```
from __future__ import absolute_import, division, print_function,␣
 ↪unicode_literals

import tensorflow as tf
import glob
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Flatten, Dense,␣
 ↪Dropout,concatenate,Activation
import numpy as np
from PIL import Image
from os import listdir
import sys
import matplotlib.pyplot as plt
from tensorflow.keras import optimizers
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.callbacks import CSVLogger
from sklearn.preprocessing import OneHotEncoder
import timeit
import gc
import random
from tensorflow.python.keras.callbacks import TensorBoard
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.regularizers import l1
from tensorflow.keras.callbacks import LearningRateScheduler
from time import time
from keras.preprocessing.image import ImageDataGenerator

print("Setup Done..")
```

```
Setup Done..
```

```
Using TensorFlow backend.
```

1.1 Loading test and train data

- we have split the data into 30 images of each category for Training
- And the rest for testing

[2]:
```python
path = './256_ObjectCategories/'
random.seed(28)
img_size = 224
test_dict = {}
train_dict = {}

train_imgs = []
test_imgs = []
train_labels = []
test_labels = []

ctr=0

start = timeit.default_timer()
categories = listdir(path)
for category in categories:
    image_files_list = glob.glob(path+category+ '/*.jpg')
    train_list = random.sample(image_files_list, k=30)
    test_list = [x for x in image_files_list if x not in train_list]
    test_dict[category] = test_list
    train_dict[category] = train_list


for key in test_dict:
    for img_name in test_dict[key]:
        file =  img_name
        test_imgs.append(np.array(image.load_img(file, target_size=(img_size,␣
 ↪img_size))))
        test_labels.append(key)


for key in train_dict:
    for img_name in train_dict[key]:
        file =  img_name
        train_imgs.append(np.array(image.load_img(file, target_size=(img_size,␣
 ↪img_size))))
        train_labels.append(key)
```

```
stop = timeit.default_timer()
print("Loading dataset Done")
print('Time taken to load data : ', stop - start)
test_imgs = np.array(test_imgs)
train_imgs = np.array(train_imgs)
test_labels = np.array(test_labels)
train_labels = np.array(train_labels)
```

```
Loading dataset Done
Time taken to load data :  269.1891710340001
```

1.2 Checking data shape and size

- We can see Train shape is of 7710 images (257*30)
- And rest for testing

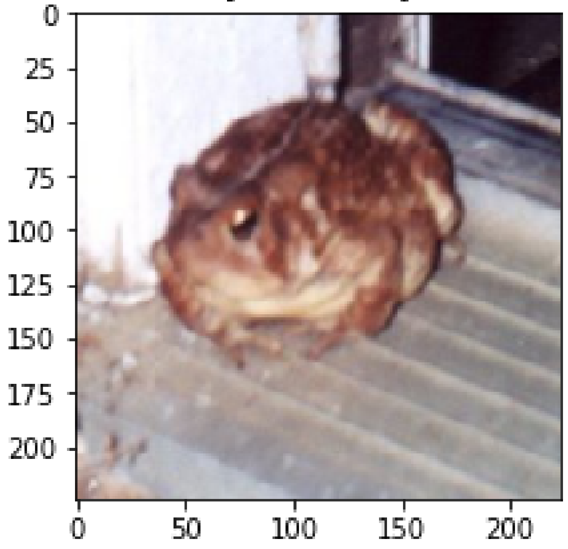
[3]:
```
gc.collect()
print("Shape of train data: ",train_imgs.shape)
print("Shape of test data: ",test_imgs.shape)
print("Shape of train labels: ",train_labels.shape)
print("Shape of test labels: ",test_labels.shape)
print("Total instances: ",train_labels.shape[0]+test_labels.shape[0])
```

```
Shape of train data:  (7710, 224, 224, 3)
Shape of test data:  (22897, 224, 224, 3)
Shape of train labels:  (7710,)
Shape of test labels:  (22897,)
Total instances:  30607
```

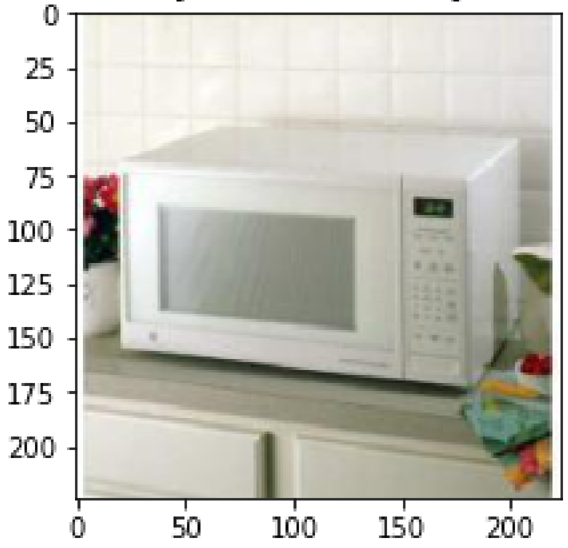
1.3 Plotting some random images

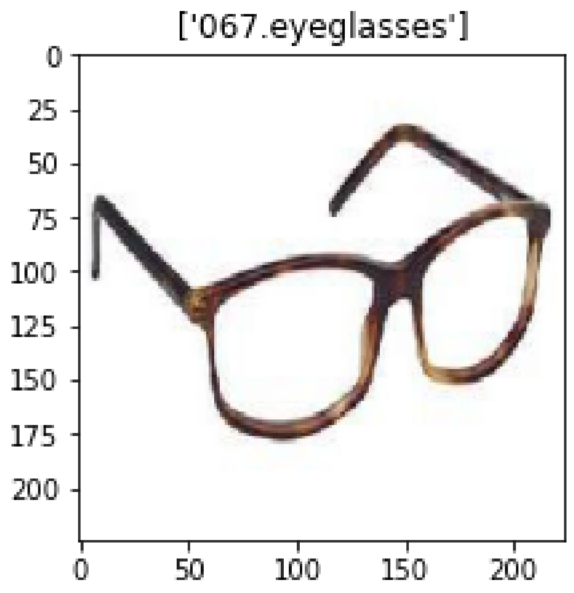
[4]:
```
val_index = []
random.seed(22)
for i in range(4):
    value = random.randint(0,3582)
    val_index.append(value)
    plt.figure( figsize=(15, 15))
    plt.subplot(1, 4, i+1)
    plt.title([train_labels[value]])
    plt.imshow(train_imgs[value])

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

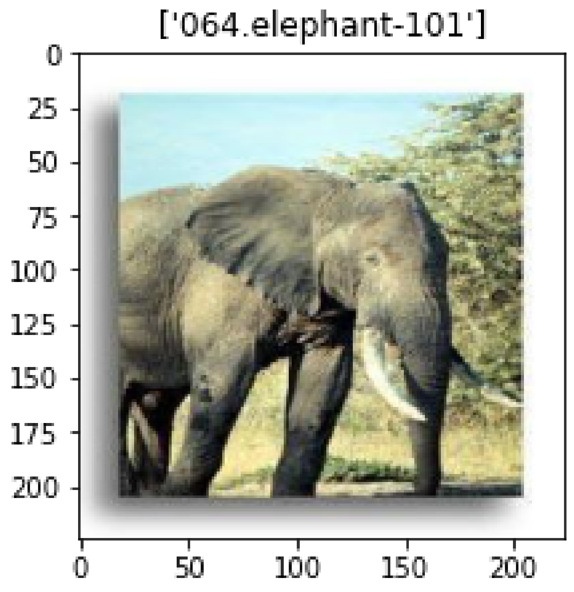
['256.toad']



['142.microwave']







2 Preprocess image using VGG function image

```
[5]: from tensorflow.keras.applications.vgg16 import preprocess_input
     start = timeit.default_timer()
     train_imgs = preprocess_input(train_imgs)
     test_imgs = preprocess_input(test_imgs)
```

```
stop = timeit.default_timer()
print('Time taken to preprocess/normalize data : ', stop - start)
gc.collect()
```

```
Time taken to preprocess/normalize data :  14.749757037999984
```

```
[5]: 9803
```

```
[6]: #check shape of  image
     #number of train images are now doubled due to image augmentation(we have only␣
      ↪used flip image for augmentation)
     print(train_imgs.shape)
     test_imgs.shape
```

```
(7710, 224, 224, 3)
```

```
[6]: (22897, 224, 224, 3)
```

2.1 Applying one hot encoding on our labels

```
[7]: onehot_encoder = OneHotEncoder(sparse=False)

     train_labels = train_labels.reshape(-1,1)
     test_labels = test_labels.reshape(-1,1)

     train_labels = onehot_encoder.fit_transform(train_labels)
     test_labels = onehot_encoder.transform(test_labels)
```

```
[8]: # test_labels1 = onehot_encoder.inverse_transform(test_labels)
```

```
[9]: print("Shape of train labels: ",train_labels.shape)
     print("Shape of test labels: ",test_labels.shape)
```

```
Shape of train labels:  (7710, 257)
Shape of test labels:  (22897, 257)
```

```
[10]: # Function to free up keras memory

      from keras.backend.tensorflow_backend import set_session
      from keras.backend.tensorflow_backend import clear_session
      from keras.backend.tensorflow_backend import get_session
      import tensorflow

      # Reset Keras Session
      def reset_keras():
          sess = get_session()
          clear_session()
```

```
    sess.close()
    sess = get_session()

    try:
        del classifier # this is from global space - change this as you need
    except:
        pass

    print(gc.collect()) # if it's done something you should see a number being␣
 ↪outputted

    # use the same config as you used to create the session
    config = tensorflow.ConfigProto()
    config.gpu_options.per_process_gpu_memory_fraction = 1
    config.gpu_options.visible_device_list = "0"
    set_session(tensorflow.Session(config=config))
```

2.2 Using model from keras–> VGG16

- we have used pretrained weights for imagenet and removed the top fully connected layers.

[11]:
```
#define variable learning rates

def learning_rate_schedule(epoch):
    if epoch <= 6:
        return 1e-5 # 0.00001
    elif epoch <= 8:
        return 1e-5
    elif epoch <= 15:
        return 1e-6
    else:
        return 1e-7
    return LR
```

[12]:
```
#Initilize vgg net
vgg_net = VGG16(include_top=True, weights='imagenet', input_shape=(224,224,3))

#Define our model
x = vgg_net.get_layer('fc2').output
x = Dense(900, activity_regularizer=l1(0.001), name='my_fc3')(x)
x = Activation('relu')(x)
x = Dropout(0.5)(x)
x = Dense(257, activation='softmax', name='predictions')(x)
model_updated = Model(inputs=vgg_net.input, outputs=x)
```

```
## save our model weights
model_updated.save_weights('CALTECH256_VGG16_model_updated_initial.h5')
print("Model_Updated saved")

history_fc_list = []
history_full_list = []
```

```
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-
packages/tensorflow_core/python/ops/resource_variable_ops.py:1630: calling
BaseResourceVariable.__init__ (from tensorflow.python.ops.resource_variable_ops)
with constraint is deprecated and will be removed in a future version.
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
Model_Updated saved
```

2.3 Using Keras image generator for image augmentation

[13]:
```
EPOCHS = 2
BATCH_SIZE = 128

train_generator = ImageDataGenerator(
                                        rotation_range = 20,
                                             horizontal_flip = True,
                                             zoom_range = .1 ,\
                                     fill_mode='nearest').flow(train_imgs,␣
 ↪train_labels, batch_size=BATCH_SIZE)
validation_generator = ImageDataGenerator().flow(test_imgs, test_labels,␣
 ↪batch_size=BATCH_SIZE)

validation_steps = test_imgs.shape[0]//BATCH_SIZE

steps_per_epoch = train_imgs.shape[0]//BATCH_SIZE
```

2.4 Run 1

[14]:
```
i=0
#Initilize vgg net
vgg_net = VGG16(include_top=True, weights='imagenet', input_shape=(224,224,3))

#Define our model
x = vgg_net.get_layer('fc2').output
x = Dense(900, activity_regularizer=l1(0.001), name='my_fc3')(x)
x = Activation('relu')(x)
x = Dropout(0.5)(x)
x = Dense(257, activation='softmax', name='predictions')(x)
model_updated = Model(inputs=vgg_net.input, outputs=x)
```

```
gc.collect()


model_updated.load_weights('CALTECH256_VGG16_model_updated_initial.h5')

print('------------------------New run started---------------------------------')
print('Initial model weights loaded , Started training run number ', i+1)
print('------------------------------------------------------------------------')

# Make the last Fully connected layers trainable and freezing the rest of VGG␣
 ↪model
for layer in model_updated.layers:
    layer.trainable = False
for layer in model_updated.layers[-4:]:
    layer.trainable = True

tensorboard = TensorBoard(log_dir="logs\{}".format('CALTECH256\FC_Layer'+str(i)))

model_updated.compile(loss='categorical_crossentropy', optimizer=optimizers.
 ↪Adam(lr=0.0001), metrics=['accuracy'])


print('--------------------------Starting Fully connected layer␣
 ↪training------------------------------------')

# history_fc = model_updated.fit(train_imgs, train_labels, batch_size=128,\
# shuffle=True, epochs=2, validation_data=\
#                     (test_imgs, test_labels),callbacks =[tensorboard])

history_fc = model_updated.fit_generator(train_generator,␣
 ↪steps_per_epoch=steps_per_epoch, epochs=2,
                    validation_data=validation_generator,␣
 ↪validation_steps=validation_steps,
                    shuffle=True, callbacks=[tensorboard])


gc.collect()
##Unfreeze weights ------ Now full model will be trained
for layer in model_updated.layers:
    layer.trainable = True


no_epochs = 20
tensorboard = TensorBoard(log_dir="logs\{}".
 ↪format('CALTECH256\FULL_Model'+str(i)))
```

```
opt2 = optimizers.Adam(lr=0.00001)

### Variable learning rate -----------------------
lrate = LearningRateScheduler(learning_rate_schedule)
callbacks_list = [lrate,tensorboard]




model_updated.compile(loss='categorical_crossentropy', optimizer=opt2,␣
 ↪metrics=['accuracy'])


print('--------------------------Starting Full model training -->after␣
 ↪unfreez------------------------------------')

## Training full model
# history = model_updated.fit(train_imgs, train_labels, batch_size=128,\
# shuffle=True, epochs=no_epochs, validation_data=\
#                       (test_imgs, test_labels),callbacks =callbacks_list)
history = model_updated.fit_generator(train_generator,␣
 ↪steps_per_epoch=steps_per_epoch, epochs=no_epochs,
                    validation_data=validation_generator,␣
 ↪validation_steps=validation_steps,
                    shuffle=True, callbacks=[tensorboard])



## Timer for checking time taken
stop = timeit.default_timer()
print('Time taken for one run is : ', stop - start)

model_updated.save_weights('CALTECH256_VGG16_model_updated_fin.h5')
print("Model_Updated weights saved")

## saving history for plots
history_fc_list.append(history_fc)
history_full_list.append(history)
```

```
------------------------New run started--------------------------------
Initial model weights loaded , Started training run number  1
-----------------------------------------------------------------------
--------------------------Starting Fully connected layer
training------------------------------------
Epoch 1/2
59/60 [============================>.] - ETA: 1s - loss: 6.6416 - acc:
```

```
0.0553Epoch 1/2
178/60 [========================================================================
=================] - 53s 297ms/step - loss: 5.0138 - acc: 0.3369
60/60 [==============================] - 129s 2s/step - loss: 6.6255 - acc:
0.0561 - val_loss: 4.9545 - val_acc: 0.3369
Epoch 2/2
59/60 [============================>.] - ETA: 1s - loss: 4.5909 - acc:
0.2962Epoch 1/2
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 3.4648 - acc: 0.5646
60/60 [==============================] - 126s 2s/step - loss: 4.5810 - acc:
0.2975 - val_loss: 3.4254 - val_acc: 0.5646
--------------------------Starting Full model training -->after
unfreez------------------------------------
Epoch 1/20
59/60 [============================>.] - ETA: 1s - loss: 3.4686 - acc:
0.4771Epoch 1/20
178/60 [========================================================================
=================] - 53s 297ms/step - loss: 2.9034 - acc: 0.6373
60/60 [==============================] - 128s 2s/step - loss: 3.4642 - acc:
0.4774 - val_loss: 2.8850 - val_acc: 0.6373
Epoch 2/20
59/60 [============================>.] - ETA: 1s - loss: 2.7852 - acc:
0.6138Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 2.6283 - acc: 0.6781
60/60 [==============================] - 126s 2s/step - loss: 2.7851 - acc:
0.6142 - val_loss: 2.5984 - val_acc: 0.6781
Epoch 3/20
59/60 [============================>.] - ETA: 1s - loss: 2.4059 - acc:
0.6892Epoch 1/20
178/60 [========================================================================
=================] - 53s 295ms/step - loss: 2.4348 - acc: 0.7070
60/60 [==============================] - 126s 2s/step - loss: 2.4066 - acc:
0.6887 - val_loss: 2.4157 - val_acc: 0.7070
Epoch 4/20
59/60 [============================>.] - ETA: 1s - loss: 2.0979 - acc:
0.7561Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 2.3293 - acc: 0.7264
60/60 [==============================] - 125s 2s/step - loss: 2.0963 - acc:
0.7560 - val_loss: 2.2974 - val_acc: 0.7264
Epoch 5/20
59/60 [============================>.] - ETA: 1s - loss: 1.8558 - acc:
0.8091Epoch 1/20
178/60 [========================================================================
=================] - 53s 295ms/step - loss: 2.2265 - acc: 0.7394
60/60 [==============================] - 124s 2s/step - loss: 1.8539 - acc:
```

```
0.8095 - val_loss: 2.2014 - val_acc: 0.7394
Epoch 6/20
59/60 [============================>.] - ETA: 1s - loss: 1.6770 - acc:
0.8480Epoch 1/20
178/60 [========================================================================
=================] - 53s 295ms/step - loss: 2.1485 - acc: 0.7507
60/60 [==============================] - 125s 2s/step - loss: 1.6755 - acc:
0.8477 - val_loss: 2.1285 - val_acc: 0.7507
Epoch 7/20
59/60 [============================>.] - ETA: 1s - loss: 1.5235 - acc:
0.8785Epoch 1/20
178/60 [========================================================================
=================] - 53s 295ms/step - loss: 2.0900 - acc: 0.7560
60/60 [==============================] - 123s 2s/step - loss: 1.5238 - acc:
0.8781 - val_loss: 2.0674 - val_acc: 0.7560
Epoch 8/20
59/60 [============================>.] - ETA: 1s - loss: 1.3887 - acc:
0.9082Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 2.0575 - acc: 0.7590
60/60 [==============================] - 124s 2s/step - loss: 1.3876 - acc:
0.9091 - val_loss: 2.0391 - val_acc: 0.7590
Epoch 9/20
59/60 [============================>.] - ETA: 1s - loss: 1.2886 - acc:
0.9318Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 2.0117 - acc: 0.7637
60/60 [==============================] - 123s 2s/step - loss: 1.2891 - acc:
0.9319 - val_loss: 1.9987 - val_acc: 0.7637
Epoch 10/20
59/60 [============================>.] - ETA: 1s - loss: 1.2042 - acc:
0.9445Epoch 1/20
178/60 [========================================================================
=================] - 53s 295ms/step - loss: 1.9692 - acc: 0.7670
60/60 [==============================] - 123s 2s/step - loss: 1.2021 - acc:
0.9449 - val_loss: 1.9556 - val_acc: 0.7670
Epoch 11/20
59/60 [============================>.] - ETA: 1s - loss: 1.1059 - acc:
0.9642Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.9286 - acc: 0.7743
60/60 [==============================] - 123s 2s/step - loss: 1.1054 - acc:
0.9644 - val_loss: 1.9129 - val_acc: 0.7743
Epoch 12/20
59/60 [============================>.] - ETA: 1s - loss: 1.0607 - acc:
0.9717Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.9181 - acc: 0.7717
```

```
60/60 [==============================] - 123s 2s/step - loss: 1.0617 - acc:
0.9715 - val_loss: 1.9029 - val_acc: 0.7717
Epoch 13/20
59/60 [============================>.] - ETA: 1s - loss: 1.0020 - acc:
0.9780Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.8711 - acc: 0.7757
60/60 [==============================] - 123s 2s/step - loss: 1.0021 - acc:
0.9781 - val_loss: 1.8658 - val_acc: 0.7757
Epoch 14/20
59/60 [============================>.] - ETA: 1s - loss: 0.9446 - acc:
0.9855Epoch 1/20
178/60 [========================================================================
=================] - 53s 295ms/step - loss: 1.8593 - acc: 0.7783
60/60 [==============================] - 122s 2s/step - loss: 0.9447 - acc:
0.9856 - val_loss: 1.8466 - val_acc: 0.7783
Epoch 15/20
59/60 [============================>.] - ETA: 1s - loss: 0.9130 - acc:
0.9873Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.8308 - acc: 0.7798
60/60 [==============================] - 121s 2s/step - loss: 0.9126 - acc:
0.9871 - val_loss: 1.8248 - val_acc: 0.7798
Epoch 16/20
59/60 [============================>.] - ETA: 1s - loss: 0.8846 - acc:
0.9909Epoch 1/20
178/60 [========================================================================
=================] - 53s 295ms/step - loss: 1.8274 - acc: 0.7796
60/60 [==============================] - 120s 2s/step - loss: 0.8845 - acc:
0.9908 - val_loss: 1.8182 - val_acc: 0.7796
Epoch 17/20
59/60 [============================>.] - ETA: 1s - loss: 0.8467 - acc:
0.9921Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.7968 - acc: 0.7818
60/60 [==============================] - 120s 2s/step - loss: 0.8468 - acc:
0.9922 - val_loss: 1.7946 - val_acc: 0.7818
Epoch 18/20
59/60 [============================>.] - ETA: 1s - loss: 0.8176 - acc:
0.9952Epoch 1/20
178/60 [========================================================================
=================] - 53s 295ms/step - loss: 1.7794 - acc: 0.7846
60/60 [==============================] - 119s 2s/step - loss: 0.8174 - acc:
0.9953 - val_loss: 1.7753 - val_acc: 0.7846
Epoch 19/20
59/60 [============================>.] - ETA: 1s - loss: 0.7817 - acc:
0.9967Epoch 1/20
178/60 [========================================================================
=================] - 52s 295ms/step - loss: 1.7719 - acc: 0.7848
60/60 [==============================] - 120s 2s/step - loss: 0.7823 - acc:
0.9966 - val_loss: 1.7673 - val_acc: 0.7848
Epoch 20/20
59/60 [============================>.] - ETA: 1s - loss: 0.7671 - acc:
0.9970Epoch 1/20
178/60 [===============================================================================
=================] - 53s 295ms/step - loss: 1.7615 - acc: 0.7858
60/60 [==============================] - 119s 2s/step - loss: 0.7666 - acc:
0.9971 - val_loss: 1.7509 - val_acc: 0.7858
Time taken for one run is :  2745.758847087
Model_Updated weights saved
```

2.5 Run 2

[15]:
```
i=1
#Initilize vgg net
vgg_net = VGG16(include_top=True, weights='imagenet', input_shape=(224,224,3))

#Define our model
x = vgg_net.get_layer('fc2').output
x = Dense(900, activity_regularizer=l1(0.001), name='my_fc3')(x)
x = Activation('relu')(x)
x = Dropout(0.5)(x)
x = Dense(257, activation='softmax', name='predictions')(x)
model_updated = Model(inputs=vgg_net.input, outputs=x)

gc.collect()


model_updated.load_weights('CALTECH256_VGG16_model_updated_initial.h5')

print('------------------------New run started-----------------------------------')
print('Initial model weights loaded , Started training run number ', i+1)
print('------------------------------------------------------------------------')

# Make the last Fully connected layers trainable and freezing the rest of VGG␣
↪model
for layer in model_updated.layers:
    layer.trainable = False
for layer in model_updated.layers[-4:]:
    layer.trainable = True

tensorboard = TensorBoard(log_dir="logs\{}".format('CALTECH256\FC_Layer'+str(i)))

model_updated.compile(loss='categorical_crossentropy', optimizer=optimizers.
↪Adam(lr=0.0001), metrics=['accuracy'])
```

```
print('--------------------------Starting Fully connected layer␣
 ↪training-----------------------------------')

# history_fc = model_updated.fit(train_imgs, train_labels, batch_size=128,\
# shuffle=True, epochs=2, validation_data=\
#                      (test_imgs, test_labels),callbacks =[tensorboard])

history_fc = model_updated.fit_generator(train_generator,␣
 ↪steps_per_epoch=steps_per_epoch, epochs=2,
                    validation_data=validation_generator,␣
 ↪validation_steps=validation_steps,
                    shuffle=True, callbacks=[tensorboard])


gc.collect()
##Unfreeze weights ------ Now full model will be trained
for layer in model_updated.layers:
    layer.trainable = True


no_epochs = 20
tensorboard = TensorBoard(log_dir="logs\{}".
 ↪format('CALTECH256\FULL_Model'+str(i)))

opt2 = optimizers.Adam(lr=0.00001)

### Variable learning rate -----------------------
lrate = LearningRateScheduler(learning_rate_schedule)
callbacks_list = [lrate,tensorboard]




model_updated.compile(loss='categorical_crossentropy', optimizer=opt2,␣
 ↪metrics=['accuracy'])


print('--------------------------Starting Full model training -->after␣
 ↪unfreez------------------------------------')

## Training full model
# history = model_updated.fit(train_imgs, train_labels, batch_size=128,\
# shuffle=True, epochs=no_epochs, validation_data=\
#                      (test_imgs, test_labels),callbacks =callbacks_list)
```

```
history = model_updated.fit_generator(train_generator,␣
↪steps_per_epoch=steps_per_epoch, epochs=no_epochs,
                    validation_data=validation_generator,␣
↪validation_steps=validation_steps,
                    shuffle=True, callbacks=[tensorboard])



## Timer for checking time taken
stop = timeit.default_timer()
print('Time taken for one run is : ', stop - start)

model_updated.save_weights('CALTECH256_VGG16_model_updated_fin.h5')
print("Model_Updated weights saved")

## saving history for plots
history_fc_list.append(history_fc)
history_full_list.append(history)
```

```
------------------------New run started---------------------------------
Initial model weights loaded , Started training run number  2
------------------------------------------------------------------------
--------------------------Starting Fully connected layer
training------------------------------------
Epoch 1/2
59/60 [============================>.] - ETA: 1s - loss: 6.6341 - acc:
0.0561Epoch 1/2
178/60 [===============================================================
=================] - 53s 297ms/step - loss: 4.9489 - acc: 0.3495
60/60 [==============================] - 126s 2s/step - loss: 6.6130 - acc:
0.0590 - val_loss: 4.8997 - val_acc: 0.3495
Epoch 2/2
59/60 [============================>.] - ETA: 1s - loss: 4.5688 - acc:
0.2997Epoch 1/2
178/60 [===============================================================
=================] - 53s 296ms/step - loss: 3.4735 - acc: 0.5593
60/60 [==============================] - 124s 2s/step - loss: 4.5632 - acc:
0.2997 - val_loss: 3.4300 - val_acc: 0.5593
--------------------------Starting Full model training -->after
unfreez------------------------------------
Epoch 1/20
59/60 [============================>.] - ETA: 1s - loss: 3.4597 - acc:
0.4752Epoch 1/20
178/60 [===============================================================
=================] - 53s 297ms/step - loss: 2.9304 - acc: 0.6347
60/60 [==============================] - 127s 2s/step - loss: 3.4578 - acc:
0.4756 - val_loss: 2.8873 - val_acc: 0.6347
```

```
Epoch 2/20
59/60 [============================>.] - ETA: 1s - loss: 2.7762 - acc:
0.6130Epoch 1/20
178/60 [==============================================================================
=================] - 53s 296ms/step - loss: 2.6157 - acc: 0.6792
60/60 [==============================] - 127s 2s/step - loss: 2.7814 - acc:
0.6128 - val_loss: 2.6062 - val_acc: 0.6792
Epoch 3/20
59/60 [============================>.] - ETA: 1s - loss: 2.3953 - acc:
0.6904Epoch 1/20
178/60 [==============================================================================
=================] - 53s 296ms/step - loss: 2.4521 - acc: 0.7017
60/60 [==============================] - 126s 2s/step - loss: 2.3967 - acc:
0.6899 - val_loss: 2.4389 - val_acc: 0.7017
Epoch 4/20
59/60 [============================>.] - ETA: 1s - loss: 2.0768 - acc:
0.7587Epoch 1/20
178/60 [==============================================================================
=================] - 53s 296ms/step - loss: 2.3147 - acc: 0.7215
60/60 [==============================] - 127s 2s/step - loss: 2.0726 - acc:
0.7600 - val_loss: 2.3051 - val_acc: 0.7215
Epoch 5/20
59/60 [============================>.] - ETA: 1s - loss: 1.8517 - acc:
0.8069Epoch 1/20
178/60 [==============================================================================
=================] - 53s 296ms/step - loss: 2.2356 - acc: 0.7344
60/60 [==============================] - 125s 2s/step - loss: 1.8528 - acc:
0.8061 - val_loss: 2.2095 - val_acc: 0.7344
Epoch 6/20
59/60 [============================>.] - ETA: 1s - loss: 1.6645 - acc:
0.8510Epoch 1/20
178/60 [==============================================================================
=================] - 53s 296ms/step - loss: 2.1548 - acc: 0.7466
60/60 [==============================] - 127s 2s/step - loss: 1.6657 - acc:
0.8507 - val_loss: 2.1429 - val_acc: 0.7466
Epoch 7/20
59/60 [============================>.] - ETA: 1s - loss: 1.5138 - acc:
0.8855Epoch 1/20
178/60 [==============================================================================
=================] - 53s 296ms/step - loss: 2.0816 - acc: 0.7543
60/60 [==============================] - 125s 2s/step - loss: 1.5128 - acc:
0.8858 - val_loss: 2.0759 - val_acc: 0.7543
Epoch 8/20
59/60 [============================>.] - ETA: 1s - loss: 1.3757 - acc:
0.9176Epoch 1/20
178/60 [==============================================================================
=================] - 53s 296ms/step - loss: 2.0366 - acc: 0.7589
60/60 [==============================] - 124s 2s/step - loss: 1.3776 - acc:
```

```
0.9173 - val_loss: 2.0163 - val_acc: 0.7589
Epoch 9/20
59/60 [============================>.] - ETA: 1s - loss: 1.2806 - acc:
0.9310Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 2.0059 - acc: 0.7632
60/60 [==============================] - 125s 2s/step - loss: 1.2810 - acc:
0.9309 - val_loss: 1.9882 - val_acc: 0.7632
Epoch 10/20
59/60 [============================>.] - ETA: 1s - loss: 1.1973 - acc:
0.9490Epoch 1/20
178/60 [========================================================================
=================] - 53s 297ms/step - loss: 1.9748 - acc: 0.7657
60/60 [==============================] - 124s 2s/step - loss: 1.1970 - acc:
0.9488 - val_loss: 1.9633 - val_acc: 0.7657
Epoch 11/20
59/60 [============================>.] - ETA: 1s - loss: 1.1078 - acc:
0.9646Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.9288 - acc: 0.7697
60/60 [==============================] - 126s 2s/step - loss: 1.1090 - acc:
0.9645 - val_loss: 1.9175 - val_acc: 0.7697
Epoch 12/20
59/60 [============================>.] - ETA: 1s - loss: 1.0484 - acc:
0.9740Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.9033 - acc: 0.7720
60/60 [==============================] - 123s 2s/step - loss: 1.0497 - acc:
0.9734 - val_loss: 1.8928 - val_acc: 0.7720
Epoch 13/20
59/60 [============================>.] - ETA: 1s - loss: 0.9995 - acc:
0.9779Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.8723 - acc: 0.7767
60/60 [==============================] - 123s 2s/step - loss: 0.9990 - acc:
0.9781 - val_loss: 1.8627 - val_acc: 0.7767
Epoch 14/20
59/60 [============================>.] - ETA: 1s - loss: 0.9588 - acc:
0.9820Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.8557 - acc: 0.7760
60/60 [==============================] - 123s 2s/step - loss: 0.9589 - acc:
0.9817 - val_loss: 1.8491 - val_acc: 0.7760
Epoch 15/20
59/60 [============================>.] - ETA: 1s - loss: 0.9122 - acc:
0.9862Epoch 1/20
178/60 [========================================================================
=================] - 53s 297ms/step - loss: 1.8350 - acc: 0.7780
```

```
60/60 [==============================] - 124s 2s/step - loss: 0.9124 - acc:
0.9863 - val_loss: 1.8226 - val_acc: 0.7780
Epoch 16/20
59/60 [============================>.] - ETA: 1s - loss: 0.8687 - acc:
0.9911Epoch 1/20
178/60 [================================================================================
=================] - 53s 297ms/step - loss: 1.8225 - acc: 0.7792
60/60 [==============================] - 123s 2s/step - loss: 0.8686 - acc:
0.9913 - val_loss: 1.8148 - val_acc: 0.7792
Epoch 17/20
59/60 [============================>.] - ETA: 1s - loss: 0.8445 - acc:
0.9918Epoch 1/20
178/60 [================================================================================
=================] - 53s 297ms/step - loss: 1.7945 - acc: 0.7822
60/60 [==============================] - 123s 2s/step - loss: 0.8446 - acc:
0.9918 - val_loss: 1.7880 - val_acc: 0.7822
Epoch 18/20
59/60 [============================>.] - ETA: 1s - loss: 0.8115 - acc:
0.9936Epoch 1/20
178/60 [================================================================================
=================] - 53s 296ms/step - loss: 1.7811 - acc: 0.7858
60/60 [==============================] - 122s 2s/step - loss: 0.8122 - acc:
0.9936 - val_loss: 1.7750 - val_acc: 0.7858
Epoch 19/20
59/60 [============================>.] - ETA: 1s - loss: 0.7841 - acc:
0.9956Epoch 1/20
178/60 [================================================================================
=================] - 53s 296ms/step - loss: 1.7661 - acc: 0.7838
60/60 [==============================] - 122s 2s/step - loss: 0.7844 - acc:
0.9955 - val_loss: 1.7627 - val_acc: 0.7838
Epoch 20/20
59/60 [============================>.] - ETA: 1s - loss: 0.7655 - acc:
0.9976Epoch 1/20
178/60 [================================================================================
=================] - 53s 296ms/step - loss: 1.7625 - acc: 0.7853
60/60 [==============================] - 122s 2s/step - loss: 0.7653 - acc:
0.9976 - val_loss: 1.7556 - val_acc: 0.7853
Time taken for one run is :  5491.758908227999
Model_Updated weights saved
```

2.6 Run 3

[17]:
```
reset_keras()
i=2
#Initilize vgg net
vgg_net = VGG16(include_top=True, weights='imagenet', input_shape=(224,224,3))
```

```
#Define our model
x = vgg_net.get_layer('fc2').output
x = Dense(900, activity_regularizer=l1(0.001), name='my_fc3')(x)
x = Activation('relu')(x)
x = Dropout(0.5)(x)
x = Dense(257, activation='softmax', name='predictions')(x)
model_updated = Model(inputs=vgg_net.input, outputs=x)

gc.collect()


model_updated.load_weights('CALTECH256_VGG16_model_updated_initial.h5')

print('------------------------New run started---------------------------------')
print('Initial model weights loaded , Started training run number ', i+1)
print('------------------------------------------------------------------------')

# Make the last Fully connected layers trainable and freezing the rest of VGG␣
 ↪model
for layer in model_updated.layers:
    layer.trainable = False
for layer in model_updated.layers[-4:]:
    layer.trainable = True

tensorboard = TensorBoard(log_dir="logs\{}".format('CALTECH256\FC_Layer'+str(i)))

model_updated.compile(loss='categorical_crossentropy', optimizer=optimizers.
 ↪Adam(lr=0.0001), metrics=['accuracy'])


print('--------------------------Starting Fully connected layer␣
 ↪training------------------------------------')

# history_fc = model_updated.fit(train_imgs, train_labels, batch_size=128,\
# shuffle=True, epochs=2, validation_data=\
#                     (test_imgs, test_labels),callbacks =[tensorboard])

history_fc = model_updated.fit_generator(train_generator,␣
 ↪steps_per_epoch=steps_per_epoch, epochs=2,
                    validation_data=validation_generator,␣
 ↪validation_steps=validation_steps,
                    shuffle=True, callbacks=[tensorboard])


gc.collect()
##Unfreeze weights ------ Now full model will be trained
for layer in model_updated.layers:
    layer.trainable = True


no_epochs = 20
tensorboard = TensorBoard(log_dir="logs\{}".
 ↪format('CALTECH256\FULL_Model'+str(i)))

opt2 = optimizers.Adam(lr=0.00001)

### Variable learning rate -----------------------
lrate = LearningRateScheduler(learning_rate_schedule)
callbacks_list = [lrate,tensorboard]





model_updated.compile(loss='categorical_crossentropy', optimizer=opt2,␣
 ↪metrics=['accuracy'])


print('--------------------------Starting Full model training -->after␣
 ↪unfreez------------------------------------')

## Training full model
# history = model_updated.fit(train_imgs, train_labels, batch_size=128,\
# shuffle=True, epochs=no_epochs, validation_data=\
#                     (test_imgs, test_labels),callbacks =callbacks_list)
history = model_updated.fit_generator(train_generator,␣
 ↪steps_per_epoch=steps_per_epoch, epochs=no_epochs,
                    validation_data=validation_generator,␣
 ↪validation_steps=validation_steps,
                    shuffle=True, callbacks=[tensorboard])




## Timer for checking time taken
stop = timeit.default_timer()
print('Time taken for one run is : ', stop - start)

model_updated.save_weights('CALTECH256_VGG16_model_updated_fin.h5')
print("Model_Updated weights saved")

## saving history for plots
history_fc_list.append(history_fc)
history_full_list.append(history)
```

40

```
------------------------New run started---------------------------------
Initial model weights loaded , Started training run number  3
------------------------------------------------------------------------
--------------------------Starting Fully connected layer
training------------------------------------
Epoch 1/2
59/60 [============================>.] - ETA: 1s - loss: 6.6055 - acc:
0.0570Epoch 1/2
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 4.9102 - acc: 0.3612
60/60 [==============================] - 131s 2s/step - loss: 6.5894 - acc:
0.0588 - val_loss: 4.8751 - val_acc: 0.3612
Epoch 2/2
59/60 [============================>.] - ETA: 1s - loss: 4.5562 - acc:
0.3013Epoch 1/2
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 3.4262 - acc: 0.5654
60/60 [==============================] - 127s 2s/step - loss: 4.5449 - acc:
0.3031 - val_loss: 3.3969 - val_acc: 0.5654
--------------------------Starting Full model training -->after
unfreez------------------------------------
Epoch 1/20
59/60 [============================>.] - ETA: 1s - loss: 3.4444 - acc:
0.4844Epoch 1/20
178/60 [========================================================================
=================] - 53s 297ms/step - loss: 2.9116 - acc: 0.6360
60/60 [==============================] - 129s 2s/step - loss: 3.4437 - acc:
0.4846 - val_loss: 2.8832 - val_acc: 0.6360
Epoch 2/20
59/60 [============================>.] - ETA: 1s - loss: 2.8072 - acc:
0.6063Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 2.6494 - acc: 0.6752
60/60 [==============================] - 127s 2s/step - loss: 2.8075 - acc:
0.6056 - val_loss: 2.6196 - val_acc: 0.6752
Epoch 3/20
59/60 [============================>.] - ETA: 1s - loss: 2.4265 - acc:
0.6806Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 2.4279 - acc: 0.7071
60/60 [==============================] - 128s 2s/step - loss: 2.4209 - acc:
0.6811 - val_loss: 2.4193 - val_acc: 0.7071
Epoch 4/20
59/60 [============================>.] - ETA: 1s - loss: 2.1048 - acc:
0.7564Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 2.3034 - acc: 0.7259
60/60 [==============================] - 126s 2s/step - loss: 2.1023 - acc:
```

```
0.7567 - val_loss: 2.2960 - val_acc: 0.7259
Epoch 5/20
59/60 [============================>.] - ETA: 1s - loss: 1.8599 - acc:
0.8100Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 2.2049 - acc: 0.7410
60/60 [==============================] - 126s 2s/step - loss: 1.8528 - acc:
0.8105 - val_loss: 2.2055 - val_acc: 0.7410
Epoch 6/20
59/60 [============================>.] - ETA: 1s - loss: 1.6706 - acc:
0.8487Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 2.1370 - acc: 0.7471
60/60 [==============================] - 127s 2s/step - loss: 1.6706 - acc:
0.8486 - val_loss: 2.1358 - val_acc: 0.7471
Epoch 7/20
59/60 [============================>.] - ETA: 1s - loss: 1.5215 - acc:
0.8842Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 2.0958 - acc: 0.7544
60/60 [==============================] - 126s 2s/step - loss: 1.5221 - acc:
0.8843 - val_loss: 2.0867 - val_acc: 0.7544
Epoch 8/20
59/60 [============================>.] - ETA: 1s - loss: 1.4056 - acc:
0.9058Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 2.0427 - acc: 0.7587
60/60 [==============================] - 124s 2s/step - loss: 1.4066 - acc:
0.9057 - val_loss: 2.0282 - val_acc: 0.7587
Epoch 9/20
59/60 [============================>.] - ETA: 1s - loss: 1.2944 - acc:
0.9306Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.9971 - acc: 0.7656
60/60 [==============================] - 124s 2s/step - loss: 1.2946 - acc:
0.9305 - val_loss: 1.9879 - val_acc: 0.7656
Epoch 10/20
59/60 [============================>.] - ETA: 1s - loss: 1.2013 - acc:
0.9465Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.9740 - acc: 0.7655
60/60 [==============================] - 125s 2s/step - loss: 1.1991 - acc:
0.9471 - val_loss: 1.9579 - val_acc: 0.7655
Epoch 11/20
59/60 [============================>.] - ETA: 1s - loss: 1.1308 - acc:
0.9579Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.9370 - acc: 0.7698
```

```
60/60 [==============================] - 123s 2s/step - loss: 1.1318 - acc:
0.9574 - val_loss: 1.9288 - val_acc: 0.7698
Epoch 12/20
59/60 [============================>.] - ETA: 1s - loss: 1.0583 - acc:
0.9715Epoch 1/20
178/60 [========================================================================
=================] - 53s 295ms/step - loss: 1.9111 - acc: 0.7722
60/60 [==============================] - 124s 2s/step - loss: 1.0591 - acc:
0.9714 - val_loss: 1.9020 - val_acc: 0.7722
Epoch 13/20
59/60 [============================>.] - ETA: 1s - loss: 1.0093 - acc:
0.9766Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.8821 - acc: 0.7737
60/60 [==============================] - 122s 2s/step - loss: 1.0105 - acc:
0.9762 - val_loss: 1.8747 - val_acc: 0.7737
Epoch 14/20
59/60 [============================>.] - ETA: 1s - loss: 0.9569 - acc:
0.9846Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.8635 - acc: 0.7778
60/60 [==============================] - 122s 2s/step - loss: 0.9564 - acc:
0.9846 - val_loss: 1.8486 - val_acc: 0.7778
Epoch 15/20
59/60 [============================>.] - ETA: 1s - loss: 0.9151 - acc:
0.9870Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.8450 - acc: 0.7798
60/60 [==============================] - 121s 2s/step - loss: 0.9143 - acc:
0.9871 - val_loss: 1.8290 - val_acc: 0.7798
Epoch 16/20
59/60 [============================>.] - ETA: 1s - loss: 0.8744 - acc:
0.9906Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.8225 - acc: 0.7814
60/60 [==============================] - 122s 2s/step - loss: 0.8746 - acc:
0.9905 - val_loss: 1.8134 - val_acc: 0.7814
Epoch 17/20
59/60 [============================>.] - ETA: 1s - loss: 0.8442 - acc:
0.9922Epoch 1/20
178/60 [========================================================================
=================] - 53s 295ms/step - loss: 1.8001 - acc: 0.7811
60/60 [==============================] - 120s 2s/step - loss: 0.8449 - acc:
0.9922 - val_loss: 1.7930 - val_acc: 0.7811
Epoch 18/20
59/60 [============================>.] - ETA: 1s - loss: 0.8161 - acc:
0.9945Epoch 1/20
178/60 [========================================================================
=================] - 53s 296ms/step - loss: 1.7768 - acc: 0.7852
60/60 [==============================] - 120s 2s/step - loss: 0.8152 - acc:
0.9946 - val_loss: 1.7692 - val_acc: 0.7852
Epoch 19/20
59/60 [============================>.] - ETA: 1s - loss: 0.7856 - acc:
0.9960Epoch 1/20
178/60 [==========================================================================
=================] - 53s 296ms/step - loss: 1.7759 - acc: 0.7833
60/60 [==============================] - 120s 2s/step - loss: 0.7853 - acc:
0.9960 - val_loss: 1.7666 - val_acc: 0.7833
Epoch 20/20
59/60 [============================>.] - ETA: 1s - loss: 0.7659 - acc:
0.9969Epoch 1/20
178/60 [==========================================================================
=================] - 53s 296ms/step - loss: 1.7456 - acc: 0.7848
60/60 [==============================] - 119s 2s/step - loss: 0.7649 - acc:
0.9970 - val_loss: 1.7470 - val_acc: 0.7848
Time taken for one run is :  8965.158857645
Model_Updated weights saved
```

2.7 Tensorboard logs have been save to logs folder

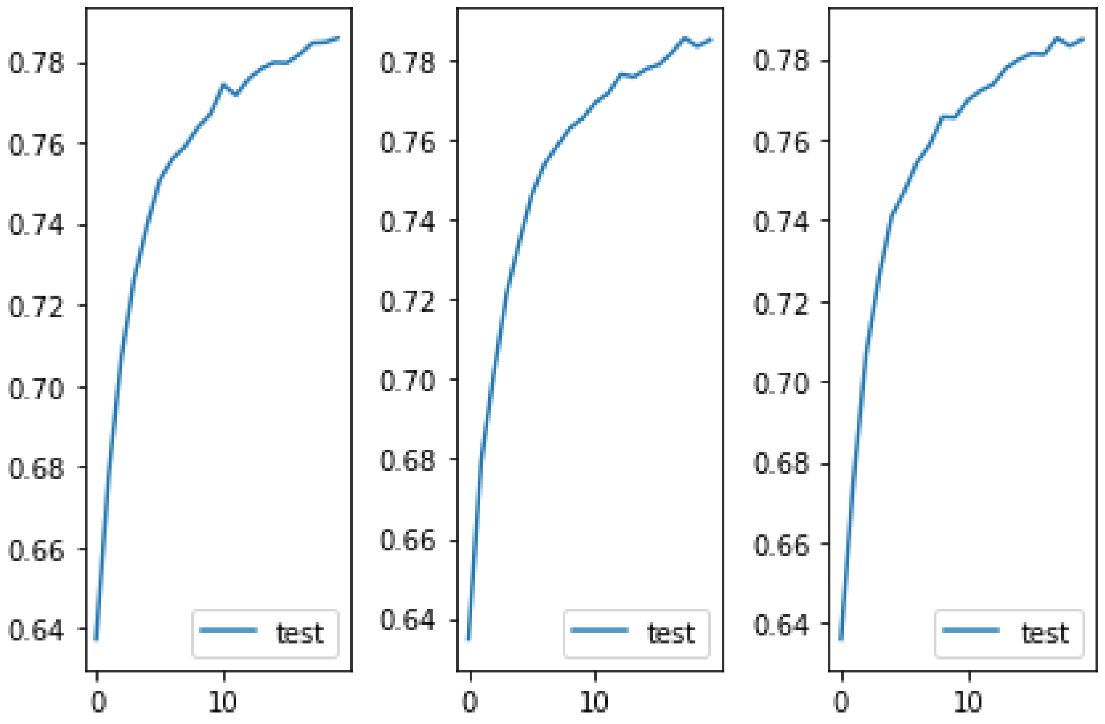
- where "*_Layer" folder means logs where we train only our FC layers
- where "*_Model" folder means logs where we train all layers in our model

2.8 Plotting the graphs of testing accuracy

- Test accuracy
- Plotted for only FC_layer training
- Plotted for full model training

[18]:
```
plt.subplot(1, 3, 1)
plt.plot(history_full_list[0].history['val_acc'])
plt.legend(['test'], loc='lower right')

plt.subplot(1, 3, 2)
plt.plot(history_full_list[1].history['val_acc'])
plt.legend(['test'], loc='lower right')

plt.subplot(1, 3, 3)
plt.plot(history_full_list[2].history['val_acc'])
plt.legend(['test'], loc='lower right')
plt.tight_layout()

plt.show()

avg_train_acc = 0
avg_test_acc = 0
```

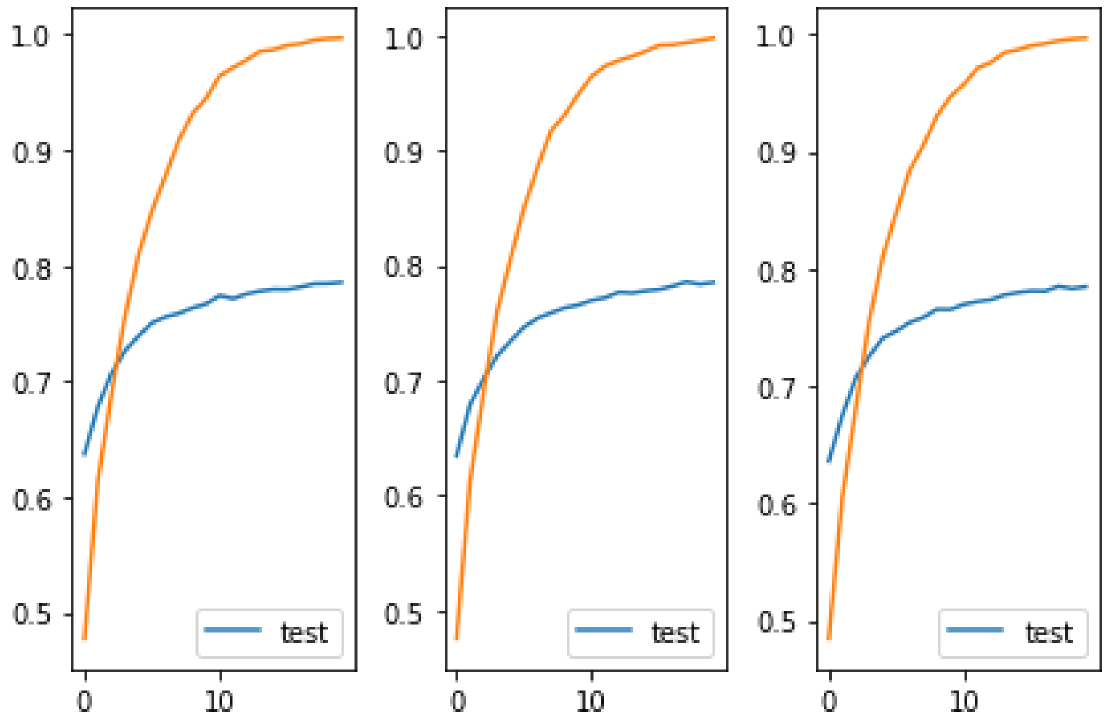
```
final_train_loss = []
final_test_loss = []

for his in history_full_list:
    final_train_loss.append(his.history['loss'][-1])
    final_test_loss.append(his.history['val_loss'][-1])
    avg_train_acc = avg_train_acc + his.history['acc'][-1]
    avg_test_acc = avg_test_acc + his.history['val_acc'][-1]

avg_train_acc = avg_train_acc/len(history_fc_list)
avg_test_acc = avg_test_acc/len(history_fc_list)


print("Average testing accuracy: {}".format(avg_test_acc))
```



```
Average testing accuracy: 0.7853317856788635
```

2.9 Train acc vs Test acc

[19]:
```
plt.subplot(1, 3, 1)
plt.plot(history_full_list[0].history['val_acc'])
plt.plot(history_full_list[0].history['acc'])
plt.legend(['test'], loc='lower right')
plt.subplot(1, 3, 2)
plt.plot(history_full_list[1].history['val_acc'])
plt.plot(history_full_list[1].history['acc'])
plt.legend(['test'], loc='lower right')

plt.subplot(1, 3, 3)
plt.plot(history_full_list[2].history['val_acc'])
plt.plot(history_full_list[2].history['acc'])
plt.legend(['test'], loc='lower right')
plt.tight_layout()

plt.show()
print("Average train accuracy: {}".format(avg_train_acc))
```

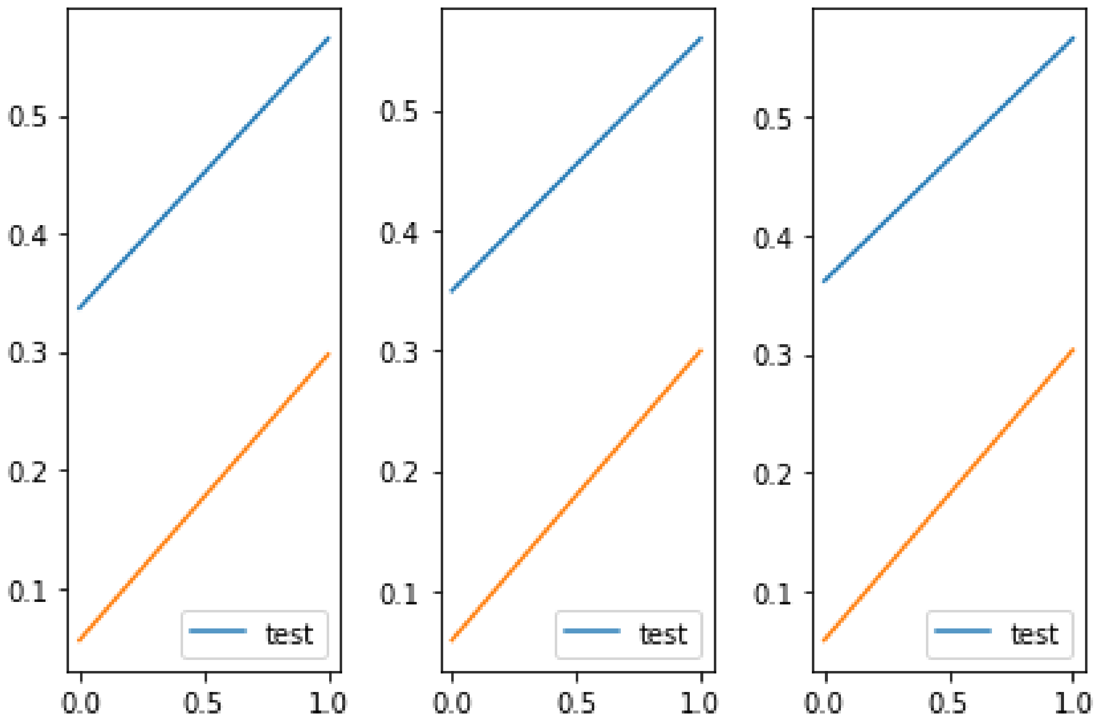


```
Average train accuracy: 0.9972302714983622
```

2.10 FC layers test and train accuracy

- Fc layers training accuracy
- graphs look linear since we have only 2 epoch
- accuracy graph of full model training is plotted above

[20]:
```
plt.subplot(1, 3, 1)
plt.plot(history_fc_list[0].history['val_acc'])
plt.plot(history_fc_list[0].history['acc'])
plt.legend(['test'], loc='lower right')

plt.subplot(1, 3, 2)
plt.plot(history_fc_list[1].history['val_acc'])
plt.plot(history_fc_list[1].history['acc'])
plt.legend(['test'], loc='lower right')

plt.subplot(1, 3, 3)
plt.plot(history_fc_list[2].history['val_acc'])
plt.plot(history_fc_list[2].history['acc'])
plt.legend(['test'], loc='lower right')
plt.tight_layout()

plt.show()
```



3 Final model average accuracy on test set

[21]:
```
print("Average Model Train accuracy is  : ",avg_test_acc*100,"%")
```

```
Average Model Train accuracy is  :  78.53317856788635 %
```