In [ ]:
```
!unzip "/content/drive/My Drive/ADM/dataset3.zip" -d "/content/sample_d
ata/dataset"
```

```
Archive:  /content/drive/My Drive/ADM/dataset3.zip
replace /content/sample_data/dataset/dataset3/test_images_from_train/
00fb450622785388.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename:
```

In [ ]:
```
# Make sure we are pointing to the directory that has all the files nec
essary
import numpy as np
import pandas as pd
import sys, requests, shutil, os
import numpy as np
from shutil import copyfile
import urllib
from tensorflow.keras.preprocessing.image import ImageDataGenerator, im
g_to_array, load_img
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dropout, Flatten, Dense, GlobalAver
agePooling2D
from tensorflow.keras import applications
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras import optimizers
#from tensorflow.keras.utils.np_utils import to_categorical
from tensorflow.keras.utils import to_categorical

from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.models import Model
import csv
import os
import cv2
from tensorflow.keras.models import load_model
import matplotlib.pyplot as plt
import math
from tensorflow.keras.optimizers import Adam
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt
import numpy as np
import argparse
import random
import tensorflow as tf
import tensorflow.keras
from keras.optimizers import SGD, Adam
```

In [ ]:
```
cd "/content/sample_data/dataset/dataset3"
```

```
/content/sample_data/dataset/dataset3
```

In [ ]:
```
# This is where we fine tune the pretrained model according to our data
set
img_width, img_height = 96, 96
save_model_weights = "VGG16_weights.h5"
train_data_dir = 'train_images_model'
validation_data_dir = 'validation_images_model'
batch_size = 200
epochs = 100
def train_VGG16():



    base_model = applications.VGG16(weights='imagenet',include_top= Fal
se,input_shape=(96,96,3))
    top_model = Sequential()
    top_model.add(Flatten(input_shape=base_model.output_shape[1:]))
    top_model.add(Dense(256, activation='relu'))
    top_model.add(Dense(256, activation='relu'))
    n_class = 6000
    top_model.add(Dense(n_class, activation='softmax'))

    model = Model(base_model.input, top_model(base_model.output))
    # set the first 16 layers to non-trainable (weights will not be upd
ated)
    # 1 conv layer and three dense layers will be trained
    for layer in model.layers[:16]:
        layer.trainable = False
    #model.load_weights("/content/sample_data/dataset/dataset3/VGG16_we
ights.h5")
    model.compile(loss='categorical_crossentropy',
                  optimizer=optimizers.Adam(lr=0.001, beta_1=0.9,beta_2
=0.999,epsilon=1e-8, decay=0.0),
                  metrics=['accuracy'])
    print ('Compilation done.')

    train_datagen = ImageDataGenerator(rescale=1. / 255,
                                       rotation_range=90,
                                        width_shift_range=0.2,
                                        height_shift_range=0.2,
                                        zoom_range = 0.5)

    valid_datagen = ImageDataGenerator(rescale=1. / 255)

    train_generator = train_datagen.flow_from_directory(
        train_data_dir,
        target_size=(img_height, img_width),
        batch_size=batch_size,
        class_mode='categorical')

    #np.save('class_indices.npy', train_generator.class_indices)

    validation_generator = valid_datagen.flow_from_directory(
        validation_data_dir,
        target_size=(img_height, img_width),
        batch_size=batch_size,
```

```
        class_mode='categorical')

    print ('Model fit begins...')
    history1=model.fit_generator(
        train_generator,
        steps_per_epoch=150,
        epochs=epochs,
        validation_data=validation_generator,
        validation_steps=75,
        callbacks=[ModelCheckpoint(filepath="vgg16_weights_tf_dim_order
ing_tf_kernels-notop.h5",
                                   save_best_only=True, save_weights_on
ly=True)]
        )

    model.save_weights(save_model_weights)
    # final weights are saved in bottleneck_fc_model.h5 file
    return history1
history1=train_VGG16()
```

```
Compilation done.
Found 246794 images belonging to 6000 classes.
Found 54916 images belonging to 6000 classes.
Model fit begins...
```

```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:932: UserWarning:
Palette images with Transparency expressed in bytes should be convert
ed to RGBA images
  "Palette images with Transparency expressed in bytes should be "
```

```
Epoch 1/100
150/150 [==============================] - 108s 722ms/step - loss: 7.
4300 - accuracy: 0.0090 - val_loss: 7.1718 - val_accuracy: 0.0139
Epoch 2/100
150/150 [==============================] - 109s 729ms/step - loss: 6.
9714 - accuracy: 0.0192 - val_loss: 6.9208 - val_accuracy: 0.0205
Epoch 3/100
150/150 [==============================] - 108s 722ms/step - loss: 6.
6725 - accuracy: 0.0275 - val_loss: 6.6716 - val_accuracy: 0.0251
Epoch 4/100
150/150 [==============================] - 109s 726ms/step - loss: 6.
5231 - accuracy: 0.0335 - val_loss: 6.4537 - val_accuracy: 0.0350
Epoch 5/100
150/150 [==============================] - 109s 726ms/step - loss: 6.
4007 - accuracy: 0.0410 - val_loss: 6.3077 - val_accuracy: 0.0434
Epoch 6/100
150/150 [==============================] - 109s 727ms/step - loss: 6.
2400 - accuracy: 0.0529 - val_loss: 6.3571 - val_accuracy: 0.0443
Epoch 7/100
150/150 [==============================] - 110s 731ms/step - loss: 6.
1298 - accuracy: 0.0607 - val_loss: 6.1201 - val_accuracy: 0.0615
Epoch 8/100
150/150 [==============================] - 110s 731ms/step - loss: 5.
9918 - accuracy: 0.0681 - val_loss: 6.0070 - val_accuracy: 0.0675
Epoch 9/100
150/150 [==============================] - 109s 727ms/step - loss: 5.
8889 - accuracy: 0.0771 - val_loss: 5.9660 - val_accuracy: 0.0698
Epoch 10/100
150/150 [==============================] - 110s 731ms/step - loss: 5.
7831 - accuracy: 0.0857 - val_loss: 5.6975 - val_accuracy: 0.0823
Epoch 11/100
150/150 [==============================] - 109s 728ms/step - loss: 5.
7154 - accuracy: 0.0882 - val_loss: 5.6943 - val_accuracy: 0.0902
Epoch 12/100
150/150 [==============================] - 109s 726ms/step - loss: 5.
6435 - accuracy: 0.0931 - val_loss: 5.5814 - val_accuracy: 0.0971
Epoch 13/100
150/150 [==============================] - 109s 728ms/step - loss: 5.
5917 - accuracy: 0.0958 - val_loss: 5.5662 - val_accuracy: 0.1003
Epoch 14/100
150/150 [==============================] - 109s 729ms/step - loss: 5.
5090 - accuracy: 0.0989 - val_loss: 5.4818 - val_accuracy: 0.1073
Epoch 15/100
150/150 [==============================] - 109s 728ms/step - loss: 5.
4479 - accuracy: 0.1081 - val_loss: 5.4121 - val_accuracy: 0.1115
Epoch 16/100
150/150 [==============================] - 109s 724ms/step - loss: 5.
4146 - accuracy: 0.1098 - val_loss: 5.3636 - val_accuracy: 0.1151
Epoch 17/100
150/150 [==============================] - 109s 725ms/step - loss: 5.
3553 - accuracy: 0.1113 - val_loss: 5.3788 - val_accuracy: 0.1195
Epoch 18/100
150/150 [==============================] - 110s 730ms/step - loss: 5.
3490 - accuracy: 0.1151 - val_loss: 5.4474 - val_accuracy: 0.1133
Epoch 19/100
150/150 [==============================] - 109s 728ms/step - loss: 5.
```

```
2971 - accuracy: 0.1209 - val_loss: 5.3283 - val_accuracy: 0.1272
Epoch 20/100
150/150 [==============================] - 109s 726ms/step - loss: 5.
2446 - accuracy: 0.1248 - val_loss: 5.2383 - val_accuracy: 0.1331
Epoch 21/100
150/150 [==============================] - 110s 730ms/step - loss: 5.
2162 - accuracy: 0.1228 - val_loss: 5.1954 - val_accuracy: 0.1388
Epoch 22/100
150/150 [==============================] - 110s 730ms/step - loss: 5.
2105 - accuracy: 0.1253 - val_loss: 5.1831 - val_accuracy: 0.1293
Epoch 23/100
150/150 [==============================] - 109s 728ms/step - loss: 5.
1667 - accuracy: 0.1315 - val_loss: 5.1527 - val_accuracy: 0.1325
Epoch 24/100
150/150 [==============================] - 109s 726ms/step - loss: 5.
1345 - accuracy: 0.1336 - val_loss: 5.2340 - val_accuracy: 0.1307
Epoch 25/100
150/150 [==============================] - 109s 725ms/step - loss: 5.
1241 - accuracy: 0.1340 - val_loss: 5.2166 - val_accuracy: 0.1418
Epoch 26/100
150/150 [==============================] - 108s 722ms/step - loss: 5.
1025 - accuracy: 0.1361 - val_loss: 5.0651 - val_accuracy: 0.1508
Epoch 27/100
150/150 [==============================] - 109s 728ms/step - loss: 5.
0498 - accuracy: 0.1392 - val_loss: 5.1113 - val_accuracy: 0.1483
Epoch 28/100
150/150 [==============================] - 109s 726ms/step - loss: 5.
0443 - accuracy: 0.1408 - val_loss: 5.1296 - val_accuracy: 0.1529
Epoch 29/100
150/150 [==============================] - 108s 723ms/step - loss: 5.
0499 - accuracy: 0.1398 - val_loss: 5.0342 - val_accuracy: 0.1507
Epoch 30/100
150/150 [==============================] - 108s 721ms/step - loss: 4.
9938 - accuracy: 0.1462 - val_loss: 5.0420 - val_accuracy: 0.1569
Epoch 31/100
150/150 [==============================] - 109s 726ms/step - loss: 4.
9983 - accuracy: 0.1471 - val_loss: 5.0137 - val_accuracy: 0.1583
Epoch 32/100
150/150 [==============================] - 109s 724ms/step - loss: 4.
9832 - accuracy: 0.1482 - val_loss: 5.0043 - val_accuracy: 0.1606
Epoch 33/100
150/150 [==============================] - 110s 730ms/step - loss: 4.
9514 - accuracy: 0.1527 - val_loss: 4.9065 - val_accuracy: 0.1719
Epoch 34/100
150/150 [==============================] - 109s 725ms/step - loss: 4.
9365 - accuracy: 0.1472 - val_loss: 4.9690 - val_accuracy: 0.1697
Epoch 35/100
150/150 [==============================] - 110s 732ms/step - loss: 4.
9152 - accuracy: 0.1490 - val_loss: 4.8987 - val_accuracy: 0.1738
Epoch 36/100
150/150 [==============================] - 110s 731ms/step - loss: 4.
8905 - accuracy: 0.1535 - val_loss: 4.8897 - val_accuracy: 0.1688
Epoch 37/100
150/150 [==============================] - 109s 727ms/step - loss: 4.
8836 - accuracy: 0.1577 - val_loss: 5.0219 - val_accuracy: 0.1615
Epoch 38/100
```

```
150/150 [==============================] - 109s 728ms/step - loss: 4.
8446 - accuracy: 0.1567 - val_loss: 4.9855 - val_accuracy: 0.1672
Epoch 39/100
150/150 [==============================] - 109s 724ms/step - loss: 4.
8623 - accuracy: 0.1564 - val_loss: 4.9100 - val_accuracy: 0.1745
Epoch 40/100
150/150 [==============================] - 108s 722ms/step - loss: 4.
8319 - accuracy: 0.1631 - val_loss: 4.9599 - val_accuracy: 0.1672
Epoch 41/100
150/150 [==============================] - 108s 723ms/step - loss: 4.
8018 - accuracy: 0.1641 - val_loss: 4.8263 - val_accuracy: 0.1813
Epoch 42/100
150/150 [==============================] - 109s 724ms/step - loss: 4.
8038 - accuracy: 0.1625 - val_loss: 4.8219 - val_accuracy: 0.1865
Epoch 43/100
150/150 [==============================] - 109s 727ms/step - loss: 4.
8120 - accuracy: 0.1608 - val_loss: 4.8227 - val_accuracy: 0.1813
Epoch 44/100
150/150 [==============================] - 109s 724ms/step - loss: 4.
8053 - accuracy: 0.1621 - val_loss: 4.7299 - val_accuracy: 0.1912
Epoch 45/100
150/150 [==============================] - 108s 723ms/step - loss: 4.
7757 - accuracy: 0.1666 - val_loss: 4.9076 - val_accuracy: 0.1785
Epoch 46/100
150/150 [==============================] - 108s 723ms/step - loss: 4.
7433 - accuracy: 0.1683 - val_loss: 4.8026 - val_accuracy: 0.1888
Epoch 47/100
150/150 [==============================] - 109s 724ms/step - loss: 4.
7614 - accuracy: 0.1668 - val_loss: 4.7838 - val_accuracy: 0.1893
Epoch 48/100
150/150 [==============================] - 109s 729ms/step - loss: 4.
7155 - accuracy: 0.1739 - val_loss: 4.8190 - val_accuracy: 0.1835
Epoch 49/100
150/150 [==============================] - 109s 728ms/step - loss: 4.
7411 - accuracy: 0.1712 - val_loss: 4.7230 - val_accuracy: 0.1908
Epoch 50/100
150/150 [==============================] - 109s 729ms/step - loss: 4.
7027 - accuracy: 0.1734 - val_loss: 4.7255 - val_accuracy: 0.1964
Epoch 51/100
150/150 [==============================] - 110s 732ms/step - loss: 4.
7041 - accuracy: 0.1739 - val_loss: 4.7440 - val_accuracy: 0.1863
Epoch 52/100
150/150 [==============================] - 111s 738ms/step - loss: 4.
6709 - accuracy: 0.1782 - val_loss: 4.9017 - val_accuracy: 0.1861
Epoch 53/100
150/150 [==============================] - 110s 734ms/step - loss: 4.
6747 - accuracy: 0.1765 - val_loss: 4.6901 - val_accuracy: 0.2022
Epoch 54/100
150/150 [==============================] - 110s 735ms/step - loss: 4.
6791 - accuracy: 0.1747 - val_loss: 4.6284 - val_accuracy: 0.2114
Epoch 55/100
150/150 [==============================] - 111s 739ms/step - loss: 4.
6236 - accuracy: 0.1842 - val_loss: 4.8191 - val_accuracy: 0.1897
Epoch 56/100
150/150 [==============================] - 110s 735ms/step - loss: 4.
6483 - accuracy: 0.1785 - val_loss: 4.8465 - val_accuracy: 0.1816
```

```
Epoch 57/100
150/150 [==============================] - 111s 737ms/step - loss: 4.
6349 - accuracy: 0.1788 - val_loss: 4.6319 - val_accuracy: 0.1984
Epoch 58/100
150/150 [==============================] - 111s 740ms/step - loss: 4.
6469 - accuracy: 0.1795 - val_loss: 4.8030 - val_accuracy: 0.1935
Epoch 59/100
150/150 [==============================] - 111s 738ms/step - loss: 4.
6271 - accuracy: 0.1807 - val_loss: 4.6667 - val_accuracy: 0.2045
Epoch 60/100
150/150 [==============================] - 111s 737ms/step - loss: 4.
5969 - accuracy: 0.1824 - val_loss: 4.7158 - val_accuracy: 0.1982
Epoch 61/100
150/150 [==============================] - 111s 739ms/step - loss: 4.
6065 - accuracy: 0.1827 - val_loss: 4.7126 - val_accuracy: 0.2009
Epoch 62/100
150/150 [==============================] - 111s 739ms/step - loss: 4.
5993 - accuracy: 0.1829 - val_loss: 4.8235 - val_accuracy: 0.1881
Epoch 63/100
150/150 [==============================] - 111s 739ms/step - loss: 4.
5754 - accuracy: 0.1884 - val_loss: 4.6701 - val_accuracy: 0.2016
Epoch 64/100
150/150 [==============================] - 111s 739ms/step - loss: 4.
6165 - accuracy: 0.1832 - val_loss: 4.6406 - val_accuracy: 0.2095
Epoch 65/100
150/150 [==============================] - 110s 734ms/step - loss: 4.
5615 - accuracy: 0.1873 - val_loss: 4.6796 - val_accuracy: 0.2053
Epoch 66/100
150/150 [==============================] - 110s 737ms/step - loss: 4.
5861 - accuracy: 0.1844 - val_loss: 4.7908 - val_accuracy: 0.2001
Epoch 67/100
150/150 [==============================] - 110s 734ms/step - loss: 4.
5303 - accuracy: 0.1902 - val_loss: 4.6051 - val_accuracy: 0.2157
Epoch 68/100
150/150 [==============================] - 110s 734ms/step - loss: 4.
5441 - accuracy: 0.1920 - val_loss: 4.6863 - val_accuracy: 0.2141
Epoch 69/100
150/150 [==============================] - 110s 734ms/step - loss: 4.
5555 - accuracy: 0.1838 - val_loss: 4.6632 - val_accuracy: 0.2169
Epoch 70/100
150/150 [==============================] - 109s 728ms/step - loss: 4.
5475 - accuracy: 0.1881 - val_loss: 4.6371 - val_accuracy: 0.2157
Epoch 71/100
150/150 [==============================] - 109s 730ms/step - loss: 4.
5341 - accuracy: 0.1900 - val_loss: 4.6168 - val_accuracy: 0.2135
Epoch 72/100
150/150 [==============================] - 110s 735ms/step - loss: 4.
5506 - accuracy: 0.1891 - val_loss: 4.6198 - val_accuracy: 0.2141
Epoch 73/100
150/150 [==============================] - 109s 729ms/step - loss: 4.
4899 - accuracy: 0.1952 - val_loss: 4.6055 - val_accuracy: 0.2180
Epoch 74/100
150/150 [==============================] - 109s 730ms/step - loss: 4.
5188 - accuracy: 0.1936 - val_loss: 4.7596 - val_accuracy: 0.2003
Epoch 75/100
150/150 [==============================] - 110s 730ms/step - loss: 4.
```

```
5118 - accuracy: 0.1942 - val_loss: 4.7052 - val_accuracy: 0.2155
Epoch 76/100
150/150 [==============================] - 108s 720ms/step - loss: 4.
5407 - accuracy: 0.1928 - val_loss: 4.4593 - val_accuracy: 0.2325
Epoch 77/100
150/150 [==============================] - 109s 725ms/step - loss: 4.
5090 - accuracy: 0.1949 - val_loss: 4.7124 - val_accuracy: 0.2168
Epoch 78/100
150/150 [==============================] - 108s 718ms/step - loss: 4.
4910 - accuracy: 0.1924 - val_loss: 4.6655 - val_accuracy: 0.2123
Epoch 79/100
150/150 [==============================] - 108s 721ms/step - loss: 4.
4860 - accuracy: 0.1936 - val_loss: 4.5779 - val_accuracy: 0.2245
Epoch 80/100
150/150 [==============================] - 110s 731ms/step - loss: 4.
4877 - accuracy: 0.1953 - val_loss: 4.5497 - val_accuracy: 0.2271
Epoch 81/100
150/150 [==============================] - 109s 726ms/step - loss: 4.
4845 - accuracy: 0.1953 - val_loss: 4.7317 - val_accuracy: 0.2072
Epoch 82/100
150/150 [==============================] - 108s 722ms/step - loss: 4.
4688 - accuracy: 0.1976 - val_loss: 4.5545 - val_accuracy: 0.2273
Epoch 83/100
150/150 [==============================] - 111s 743ms/step - loss: 4.
4392 - accuracy: 0.1977 - val_loss: 4.5983 - val_accuracy: 0.2251
Epoch 84/100
150/150 [==============================] - 110s 731ms/step - loss: 4.
4504 - accuracy: 0.1988 - val_loss: 4.6340 - val_accuracy: 0.2145
```

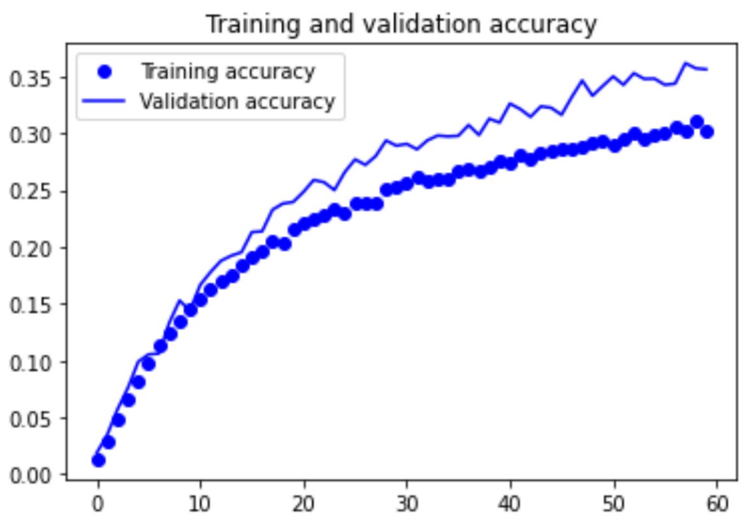
In [ ]:

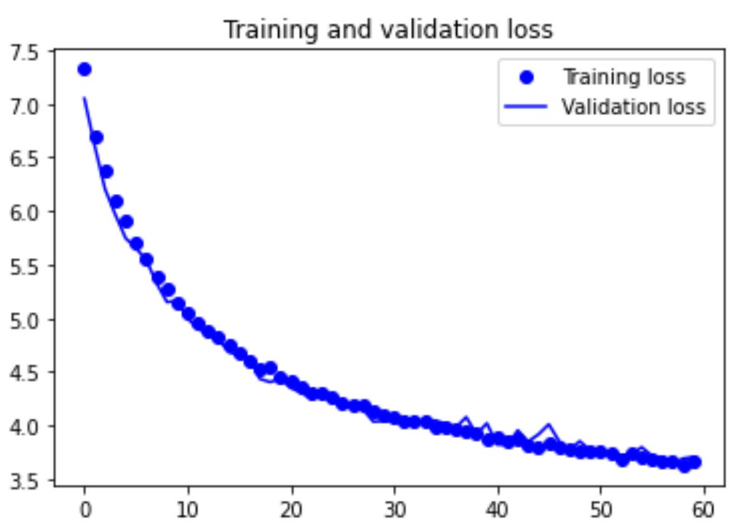
```
history1
```

Out[ ]:

```
<tensorflow.python.keras.callbacks.History at 0x7f04acdab198>
```

In [ ]:
```
acc = history1.history['accuracy']
val_acc = history1.history['val_accuracy']
loss = history1.history['loss']
val_loss = history1.history['val_loss']
epochs = range(60)
plt.figure()
plt.plot(epochs, acc, 'bo', label='Training accuracy')
plt.plot(epochs, val_acc, 'b', label='Validation accuracy')
plt.title('Training and validation accuracy')
plt.legend()
plt.show()
epochs = range(60)
plt.figure()
plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()
plt.show()
```





In [ ]: