```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remo
```

```
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import ResNet50
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from tqdm import tqdm
from keras.layers import BatchNormalization
from keras.layers import Input, Lambda, Dense, Flatten
from keras.models import Model
from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator
from glob import glob
import matplotlib.pyplot as plt
```

```
# Unzip the file
!unzip "/content/drive/MyDrive/Selfie-dataset.tar.zip" -d "/content/drive/MyDrive/destination_folder(selfie)"
```

```
Archive:  /content/drive/MyDrive/Selfie-dataset.tar.zip
  inflating: /content/drive/MyDrive/destination_folder(selfie)/Selfie-dataset.tar.gz
```

```
import tarfile
with tarfile.open("/content/drive/MyDrive/Selfie-dataset.tar.gz",'r:*') as tar:
  csv_path=list(n for n in tar.getnames() if n.endswith('.txt'))
  df=pd.read_csv(tar.extractfile(csv_path[0]),header=0,sep=" ")
```

Double-click (or enter) to edit

```
df.head()
```

| | 00a454da495e11e28a7322000a1fa414_6 | 3.901 | 1 | 1.1 | -1 | -1.1 | -1.2 | 1.2 | -1.3 | -1.4 |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 00cddb96ac4c11e3a30212279ba1b65f_6 | 4.385 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 |
| **1** | 01cdd7aa1a1a11e2aaa822000a1fb0dd_6 | 4.243 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 |
| **2** | 024696bead0c11e389d50ec42b3b1b1c_6 | 4.169 | -1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 |
| **3** | 026df048221a11e2b52122000a1fa4b5_6 | 3.873 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | -1 |
| **4** | 02b511eed26911e2851d22000a1fb71f_6 | 4.458 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 |

```
score","1":"partial_faces","1.1":"is_female","-1":"baby","-1.1":"child","-1.2":"teenager","1.2":"youth","-1.3":"middleage","
```

```
df.head()
```

| | image | score | partial_faces | is_female | baby | child | teenager | youth | middleage | senio |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 00cddb96ac4c11e3a30212279ba1b65f_6 | 4.385 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | - |
| **1** | 01cdd7aa1a1a11e2aaa822000a1fb0dd_6 | 4.243 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | - |
| **2** | 024696bead0c11e389d50ec42b3b1b1c_6 | 4.169 | -1 | -1 | -1 | -1 | 1 | -1 | -1 | - |
| **3** | 026df048221a11e2b52122000a1fa4b5_6 | 3.873 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | - |
| **4** | 02b511eed26911e2851d22000a1fb71f_6 | 4.458 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | - |

```
df=df.iloc[:2000]
```

```
!tar -xvf  '/content/drive/MyDrive/destination_folder(selfie)/Selfie-dataset.tar.gz' -C '/content/drive/MyDrive/cell_images'
```

```
Selfie-dataset/images/1170045_693757937348103_1549391212_a.jpg
```

```
Selfie-dataset/images/10012573_231568337050125_1706051455_a.jpg

Selfie-dataset/images/10005709_467883560011375_1670896738_a.jpg
Selfie-dataset/images/1661927_228596544008105_684865188_a.jpg
Selfie-dataset/images/10009910_740160099338963_731167508_a.jpg
Selfie-dataset/images/10261022_257568864414688_977484631_a.jpg
Selfie-dataset/images/10005582_1415191278741495_953986968_a.jpg
Selfie-dataset/images/928487_1515113792049350_410196779_a.jpg
Selfie-dataset/images/1172134_650102058361010_88252759_a.jpg
Selfie-dataset/images/10254290_776403229037379_286983984_a.jpg
Selfie-dataset/images/10013194_1401592026783203_1386952190_a.jpg
Selfie-dataset/images/925229_1489023201318607_972962151_a.jpg
Selfie-dataset/images/1799638_482311871894747_521153364_a.jpg
Selfie-dataset/images/10009921_501440839959825_1474353175_a.jpg
Selfie-dataset/images/927640_531513403624343_448637030_a.jpg
Selfie-dataset/images/10245952_1481992535349761_749784743_a.jpg
Selfie-dataset/images/10175161_274370092730193_1757993163_a.jpg
Selfie-dataset/images/10251429_679667042095777_1344221422_a.jpg
Selfie-dataset/images/1171674_230385217170058_1886386055_a.jpg
Selfie-dataset/images/10175321_220018201526686_658004582_a.jpg
Selfie-dataset/images/10011244_1481587075390754_1643133014_a.jpg
Selfie-dataset/images/10175409_466912923442527_1562831364_a.jpg
Selfie-dataset/images/10261153_544365585681842_1248577112_a.jpg
Selfie-dataset/images/1168756_1411637795769261_1994150638_a.jpg
Selfie-dataset/images/10175310_504551812982788_1349697104_a.jpg
Selfie-dataset/images/10261207_1403069069966975_913027073_a.jpg
Selfie-dataset/images/10011145_610382789037431_198816674_a.jpg
Selfie-dataset/images/1172108_1483830398501744_1261882285_a.jpg
Selfie-dataset/images/924330_1407895696148830_404485472_a.jpg
Selfie-dataset/images/10246022_266216820223667_909033495_a.jpg
Selfie-dataset/images/1527585_513574698753296_248689837_a.jpg
Selfie-dataset/images/10175248_742306999143238_1132026306_a.jpg
Selfie-dataset/images/1388918_483589691767930_1134250942_a.jpg
Selfie-dataset/images/10254096_230103673851041_449431829_a.jpg
Selfie-dataset/images/1742216_624340240987478_2142275494_a.jpg
Selfie-dataset/images/1516990_618133034946812_1463025505_a.jpg
Selfie-dataset/images/1171207_355645297912762_1716626234_a.jpg
Selfie-dataset/images/10254123_671815509545416_342391770_a.jpg
Selfie-dataset/images/1530844_1507585446135572_1387881321_a.jpg
Selfie-dataset/images/10251503_519209288188580_1423482137_a.jpg
Selfie-dataset/images/927637_1423581211228758_1863134812_a.jpg
Selfie-dataset/images/927269_762202957157992_1823797539_a.jpg
Selfie-dataset/images/10249089_1423298967924215_2087281240_a.jpg
Selfie-dataset/images/10009266_564914623607828_1266114427_a.jpg
```

```
Selfie-dataset/images/928236_1469130203304494_1417563548_a.jpg
Selfie-dataset/images/10246123_809645345731926_640122272_a.jpg
Selfie-dataset/images/10013114_625499334196702_86707009_a.jpg
Selfie-dataset/images/1941050_619024631515838_1892055960_a.jpg
Selfie-dataset/images/10004350_278987988934901_1464910350_a.jpg
Selfie-dataset/images/927944_684406841621052_1940581209_a.jpg
Selfie-dataset/images/10005514_868086223208834_181683222_a.jpg
Selfie-dataset/images/1172182_681632578549948_77562453_a.jpg
Selfie-dataset/images/10009256_264618983712081_872164325_a.jpg
Selfie-dataset/images/1170967_1437678063140327_1722044_a.jpg
Selfie-dataset/images/1597255_672082299518755_1108449002_a.jpg
Selfie-dataset/images/5500550a94e611e28b2822000aa80213_6.jpg
Selfie-dataset/selfie_dataset.txt
Selfie-dataset/README.txt
```

```
df.shape
```

```
(2000, 10)
```

```
image_directory = '/content/drive/MyDrive/cell_images/Selfie-dataset/images/'

SIZE = 224
X_dataset = []
for i in tqdm(range(df.shape[0])):
    img = image.load_img(image_directory +df['image'][i]+'.jpg', target_size=(SIZE,SIZE,3))
    img = image.img_to_array(img)
    img = img/255.
    X_dataset.append(img)

X = np.array(X_dataset)
```

```
100%|██████████| 2000/2000 [00:06<00:00, 304.48it/s]
```

```
y = np.array(df.drop(['image','partial_faces'], axis=1))
y.shape
```

```
(2000, 8)
```

```
IMAGE_SIZE = [224, 224,3]

# adding preprocessing layer to the front of VGG
resnet = ResNet50(input_shape=(224, 224,3), include_top=False, weights="imagenet")

# To prevent training of existing weights
for layer in resnet.layers:
  layer.trainable = False


x = Flatten()(resnet.output)
prediction = Dense(8, activation='sigmoid')(x)

# create a model object
model = Model(inputs=resnet.input, outputs=prediction)

# To view the structure of the model
model.summary()


model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
conv3_block1_1_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block1_1_conv[0][0]
__________________________________________________________________________________________________
conv3_block1_1_relu (Activation (None, 28, 28, 128)  0           conv3_block1_1_bn[0][0]
__________________________________________________________________________________________________
conv3_block1_2_conv (Conv2D)    (None, 28, 28, 128)  147584      conv3_block1_1_relu[0][0]
__________________________________________________________________________________________________
conv3_block1_2_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block1_2_conv[0][0]
__________________________________________________________________________________________________
conv3_block1_2_relu (Activation (None, 28, 28, 128)  0           conv3_block1_2_bn[0][0]
__________________________________________________________________________________________________
conv3_block1_0_conv (Conv2D)    (None, 28, 28, 512)  131584      conv2_block3_out[0][0]
__________________________________________________________________________________________________
conv3_block1_3_conv (Conv2D)    (None, 28, 28, 512)  66048       conv3_block1_2_relu[0][0]
__________________________________________________________________________________________________
conv3_block1_0_bn (BatchNormali (None, 28, 28, 512)  2048        conv3_block1_0_conv[0][0]
```

| | | | |
|---|---|---|---|
| conv3_block1_3_bn (BatchNormali | (None, 28, 28, 512) | 2048 | conv3_block1_3_conv[0][0] |
| conv3_block1_add (Add) | (None, 28, 28, 512) | 0 | conv3_block1_0_bn[0][0]<br>conv3_block1_3_bn[0][0] |
| conv3_block1_out (Activation) | (None, 28, 28, 512) | 0 | conv3_block1_add[0][0] |
| conv3_block2_1_conv (Conv2D) | (None, 28, 28, 128) | 65664 | conv3_block1_out[0][0] |
| conv3_block2_1_bn (BatchNormali | (None, 28, 28, 128) | 512 | conv3_block2_1_conv[0][0] |
| conv3_block2_1_relu (Activation | (None, 28, 28, 128) | 0 | conv3_block2_1_bn[0][0] |
| conv3_block2_2_conv (Conv2D) | (None, 28, 28, 128) | 147584 | conv3_block2_1_relu[0][0] |
| conv3_block2_2_bn (BatchNormali | (None, 28, 28, 128) | 512 | conv3_block2_2_conv[0][0] |
| conv3_block2_2_relu (Activation | (None, 28, 28, 128) | 0 | conv3_block2_2_bn[0][0] |
| conv3_block2_3_conv (Conv2D) | (None, 28, 28, 512) | 66048 | conv3_block2_2_relu[0][0] |
| conv3_block2_3_bn (BatchNormali | (None, 28, 28, 512) | 2048 | conv3_block2_3_conv[0][0] |
| conv3_block2_add (Add) | (None, 28, 28, 512) | 0 | conv3_block1_out[0][0]<br>conv3_block2_3_bn[0][0] |
| conv3_block2_out (Activation) | (None, 28, 28, 512) | 0 | conv3_block2_add[0][0] |
| conv3_block3_1_conv (Conv2D) | (None, 28, 28, 128) | 65664 | conv3_block2_out[0][0] |
| conv3_block3_1_bn (BatchNormali | (None, 28, 28, 128) | 512 | conv3_block3_1_conv[0][0] |
| conv3_block3_1_relu (Activation | (None, 28, 28, 128) | 0 | conv3_block3_1_bn[0][0] |
| conv3_block3_2_conv (Conv2D) | (None, 28, 28, 128) | 147584 | conv3_block3_1_relu[0][0] |
| conv3_block3_2_bn (BatchNormali | (None, 28, 28, 128) | 512 | conv3_block3_2_conv[0][0] |
| conv3_block3_2_relu (Activation | (None, 28, 28, 128) | 0 | conv3_block3_2_bn[0][0] |
| conv3_block3_3_conv (Conv2D) | (None, 28, 28, 512) | 66048 | conv3_block3_2_relu[0][0] |
| conv3_block3_3_bn (BatchNormali | (None, 28, 28, 512) | 2048 | conv3_block3_3_conv[0][0] |

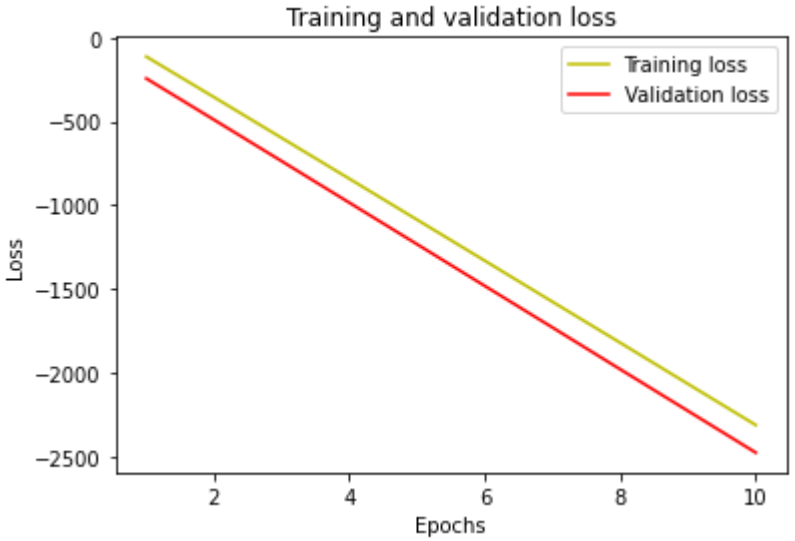
```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=20, test_size=0.3)
```

```
history = model.fit(X_train, y_train, epochs=10, validation_data=(X_test, y_test), batch_size=64)
```

```
Epoch 1/10
22/22 [==============================] - 9s 399ms/step - loss: -113.7039 - accuracy: 0.9450 - val_loss: -243.9999 - va
Epoch 2/10
22/22 [==============================] - 7s 319ms/step - loss: -355.5009 - accuracy: 1.0000 - val_loss: -490.2747 - va
Epoch 3/10
22/22 [==============================] - 7s 319ms/step - loss: -598.7076 - accuracy: 1.0000 - val_loss: -737.1755 - va
Epoch 4/10
22/22 [==============================] - 7s 319ms/step - loss: -842.2648 - accuracy: 1.0000 - val_loss: -985.0171 - va
Epoch 5/10
22/22 [==============================] - 7s 320ms/step - loss: -1086.6611 - accuracy: 1.0000 - val_loss: -1232.7355 -
Epoch 6/10
22/22 [==============================] - 7s 320ms/step - loss: -1330.9397 - accuracy: 1.0000 - val_loss: -1481.2738 -
Epoch 7/10
22/22 [==============================] - 7s 319ms/step - loss: -1576.0028 - accuracy: 1.0000 - val_loss: -1729.2463 -
Epoch 8/10
22/22 [==============================] - 7s 320ms/step - loss: -1820.1232 - accuracy: 1.0000 - val_loss: -1978.4656 -
Epoch 9/10
22/22 [==============================] - 7s 320ms/step - loss: -2065.5710 - accuracy: 1.0000 - val_loss: -2226.5164 -
Epoch 10/10
22/22 [==============================] - 7s 319ms/step - loss: -2310.4136 - accuracy: 1.0000 - val_loss: -2474.9949 -
```

```
#plot the training and validation accuracy and loss at each epoch
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(loss) + 1)
plt.plot(epochs, loss, 'y', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

```
plt.show()
```

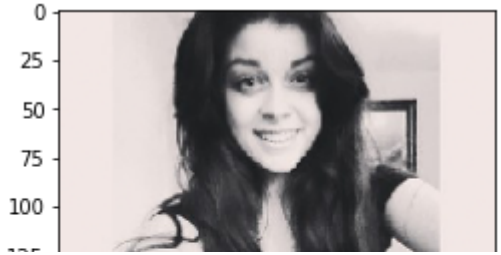



```
#Validation on an image

img = image.load_img('/content/0dabcf40c0e611e388890002c9ce57aa_6.jpg', target_size=(SIZE,SIZE,3))

img = image.img_to_array(img)
img = img/255.
plt.imshow(img)
img = np.expand_dims(img, axis=0)

classes = np.array(df.columns[2:]) #Get array of all classes
proba = model.predict(img)  #Get probabilities for each class
sorted_categories = np.argsort(proba[0])[:-11:-1]  #Get class names for top 8 categories
```




```
#Print classes and corresponding probabilities
for i in range(8):
    print("{}".format(classes[sorted_categories[i]])+" ({:.3})".format(proba[0][sorted_categories[i]]))
```

```
 partial_faces (1.0)
 is_female (0.997)
 youth (0.953)
 senior (0.0)
 middleage (0.0)
 teenager (0.0)
 child (0.0)
 baby (0.0)
```

```
#Validation on an image

img = image.load_img('/content/1742605_726331307388933_664355036_a.jpg', target_size=(SIZE,SIZE,3))

img = image.img_to_array(img)
img = img/255.
plt.imshow(img)
img = np.expand_dims(img, axis=0)

classes = np.array(df.columns[2:]) #Get array of all classes
proba = model.predict(img)  #Get probabilities for each class
sorted_categories = np.argsort(proba[0])[:-11:-1]  #Get class names for top 8 categories
```


```
#Print classes and corresponding probabilities
for i in range(8):
    print("{}".format(classes[sorted_categories[i]])+" ({:.3})".format(proba[0][sorted_categories[i]]))
```

```
 partial_faces (1.0)
 youth (0.0279)
 is_female (3.57e-05)
 senior (0.0)
 middleage (0.0)
 teenager (0.0)
 child (0.0)
 baby (0.0)
```

```
df.columns[2:]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-8b9d275a737c> in <module>()
----> 1 df.columns[2:]

NameError: name 'df' is not defined
```

SEARCH STACK OVERFLOW