```
# unzipping dataset
from google.colab import drive
drive.mount('/content/gdrive')
!unzip gdrive/MyDrive/FaceRecognition/archive2.zip
```

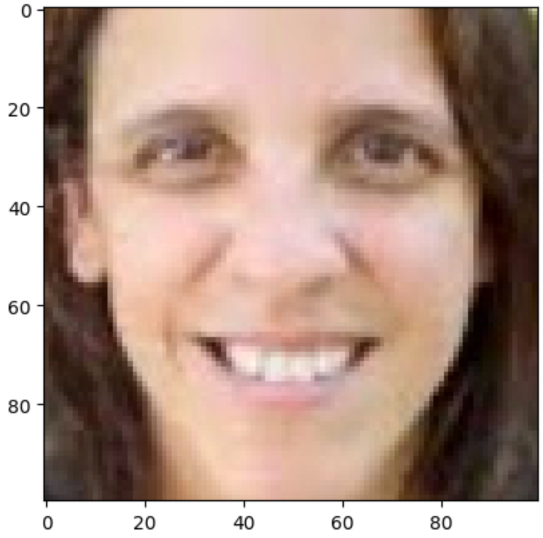
```
      inflating: archive2/UTKFace/9_0_0_20170110224437980.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._9_0_0_20170110224437980.jpg.chip.jpg
      inflating: archive2/UTKFace/26_0_0_20170113210319942.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._26_0_0_20170113210319942.jpg.chip.jpg
      inflating: archive2/UTKFace/1_1_2_20161219203318222.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._1_1_2_20161219203318222.jpg.chip.jpg
      inflating: archive2/UTKFace/26_1_2_20170116180824329.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._26_1_2_20170116180824329.jpg.chip.jpg
      inflating: archive2/UTKFace/19_1_3_20170104222642335.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._19_1_3_20170104222642335.jpg.chip.jpg
      inflating: archive2/UTKFace/34_0_1_20170116003716387.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._34_0_1_20170116003716387.jpg.chip.jpg
      inflating: archive2/UTKFace/28_1_2_20170116192959114.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._28_1_2_20170116192959114.jpg.chip.jpg
      inflating: archive2/UTKFace/40_0_0_20170116232746033.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._40_0_0_20170116232746033.jpg.chip.jpg
      inflating: archive2/UTKFace/36_0_1_20170116144540721.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._36_0_1_20170116144540721.jpg.chip.jpg
      inflating: archive2/UTKFace/62_0_0_20170105164057133.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._62_0_0_20170105164057133.jpg.chip.jpg
      inflating: archive2/UTKFace/60_0_3_20170119205730463.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._60_0_3_20170119205730463.jpg.chip.jpg
      inflating: archive2/UTKFace/22_1_4_20170103233803763.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._22_1_4_20170103233803763.jpg.chip.jpg
      inflating: archive2/UTKFace/1_1_0_20170109194452834.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._1_1_0_20170109194452834.jpg.chip.jpg
      inflating: archive2/UTKFace/16_1_0_20170109212413425.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._16_1_0_20170109212413425.jpg.chip.jpg
      inflating: archive2/UTKFace/47_0_0_20170116214228124.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._47_0_0_20170116214228124.jpg.chip.jpg
      inflating: archive2/UTKFace/39_0_1_20170113151528768.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._39_0_1_20170113151528768.jpg.chip.jpg
      inflating: archive2/UTKFace/1_1_3_20161219224454728.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._1_1_3_20161219224454728.jpg.chip.jpg
      inflating: archive2/UTKFace/17_1_1_20170112230710598.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._17_1_1_20170112230710598.jpg.chip.jpg
      inflating: archive2/UTKFace/37_0_0_20170117135950641.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._37_0_0_20170117135950641.jpg.chip.jpg
      inflating: archive2/UTKFace/68_1_1_20170113012709282.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._68_1_1_20170113012709282.jpg.chip.jpg
      inflating: archive2/UTKFace/71_0_0_20170104185305424.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._71_0_0_20170104185305424.jpg.chip.jpg
      inflating: archive2/UTKFace/49_0_3_20170117130537950.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._49_0_3_20170117130537950.jpg.chip.jpg
      inflating: archive2/UTKFace/54_0_0_20170109010040814.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._54_0_0_20170109010040814.jpg.chip.jpg
      inflating: archive2/UTKFace/36_0_0_20170104181520477.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._36_0_0_20170104181520477.jpg.chip.jpg
      inflating: archive2/UTKFace/35_0_0_20170117155112202.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._35_0_0_20170117155112202.jpg.chip.jpg
      inflating: archive2/UTKFace/52_0_3_20170119200211340.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._52_0_3_20170119200211340.jpg.chip.jpg
      inflating: archive2/UTKFace/26_0_1_20170116010114628.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._26_0_1_20170116010114628.jpg.chip.jpg
      inflating: archive2/UTKFace/28_0_1_20170117015458481.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._28_0_1_20170117015458481.jpg.chip.jpg
      inflating: archive2/UTKFace/50_1_1_20170120220813715.jpg.chip.jpg
      inflating: __MACOSX/archive2/UTKFace/._50_1_1_20170120220813715.jpg.chip.jpg
```

```
# import modules
import random
import keras
from tensorflow.keras import layers
from keras.models import Sequential
import tensorflow as tf
import tensorflow_datasets as tfds
from keras.layers.core import Dense, Activation, Dropout, Flatten
from keras.layers.convolutional import Convolution2D, MaxPooling2D
from keras.optimizers import SGD, RMSprop, adam
from keras.utils import np_utils
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
import matplotlib.pyplot as plt
import numpy as np
import os
import cv2
```

```
from PIL import Image
import matplotlib.pyplot as plt
from keras.utils import np_utils
from sklearn.model_selection import train_test_split
from keras.applications.vgg16 import VGG16
from keras.applications.vgg16 import VGG16
from keras.models import Model
from tensorflow.keras import layers
from tensorflow.keras.applications.vgg16 import preprocess_input
import tensorflow_datasets as tfds


path = "/content/archive2/UTKFace/"

training = []
amounts = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
for img in os.listdir(path):

    if img != ".DS_Store"and img != "20170116174525125.jpg.chip.jpg" and img != "20170109142408075.jpg.chip.jpg":
      classification = img.split("_")
      age = int(classification[0])

      if age < 5:
        category = 0

      elif age < 10 :
        category = 1

      elif age < 15 :
        category = 2

      elif age < 20  :
        category = 3

      elif age < 25 :
        category = 4

      elif age < 30 :
        category = 5

      elif age < 35:
        category = 6

      elif age < 40 :
        category = 7

      elif age < 45 :
        category = 8

      elif age < 50:
        category = 9

      elif age < 60 :
        category = 10

      elif age < 70 :
        category = 11

      elif age < 80 :
        category = 12

      elif age < 90 :
        category = 13

      elif age < 100:
        category = 14

      else:
        category = 15

      if amounts[category] < 900:
        amounts[category] +=1
        img_array = cv2.imread(os.path.join(path, img))
        img_array = cv2.cvtColor(img_array,cv2.COLOR_RGB2BGR)
        imga = cv2.resize(img_array,dsize = (100,100), interpolation = cv2.INTER_CUBIC)
        training.append([imga, category])
# randomizes data
random.shuffle(training)
```

```
plt.imshow(training[0][0])
print(amounts)
```

```
[900, 895, 589, 900, 900, 900, 900, 900, 900, 900, 900, 900, 699, 504, 137, 32]
```



```
X = []
y = []

for features, label in training:
    X.append(features)
    y.append(label)

Xarray = np.array(X).reshape(-1, 100, 100, 3)
Xarray.shape
Xarraydiv = Xarray/ 255

Xarraytype = Xarraydiv.astype('float32')
nexty = np_utils.to_categorical(y,16)
print(nexty.shape)
```

```
(11856, 16)
```

```
X_train, X_test, y_train, y_test = train_test_split(Xarraytype,nexty,test_size = .2, random_state = 16)
```

```
input_shape = (100,100,3)
model6 = keras.Sequential(
    [
        keras.Input(shape=input_shape),

        layers.Conv2D(32, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Dropout(0.2),
        layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.RandomRotation(0.1),
        layers.RandomFlip("horizontal"),
        layers.Conv2D(128,kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2,2)),
        layers.Dropout(0.2),
        layers.Conv2D(256,kernel_size=(3,3), activation = "relu"),
        layers.MaxPooling2D(pool_size=(2,2)),
        layers.Conv2D(512,kernel_size = (3,3), activation = "relu"),
        layers.MaxPooling2D(pool_size = (2,2)),
        layers.Flatten(),
        layers.Dropout(0.5),
        layers.Dense(150,activation="sigmoid"),
        layers.Dense(16, activation="sigmoid"),
    ]
)
model6.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 98, 98, 32)        896       
                                                                 
 max_pooling2d (MaxPooling2D  (None, 49, 49, 32)       0         
 )                                                               
                                                                 
 dropout (Dropout)           (None, 49, 49, 32)        0         
                                                                 
 conv2d_1 (Conv2D)           (None, 47, 47, 64)        18496     
                                                                 
 max_pooling2d_1 (MaxPooling  (None, 23, 23, 64)       0         
 2D)                                                             
                                                                 
 random_rotation (RandomRota  (None, 23, 23, 64)       0         
 tion)                                                           
                                                                 
 random_flip (RandomFlip)    (None, 23, 23, 64)        0         
                                                                 
 conv2d_2 (Conv2D)           (None, 21, 21, 128)       73856     
                                                                 
 max_pooling2d_2 (MaxPooling  (None, 10, 10, 128)      0         
 2D)                                                             
                                                                 
 dropout_1 (Dropout)         (None, 10, 10, 128)       0         
                                                                 
 conv2d_3 (Conv2D)           (None, 8, 8, 256)         295168    
                                                                 
 max_pooling2d_3 (MaxPooling  (None, 4, 4, 256)        0         
 2D)                                                             
                                                                 
 conv2d_4 (Conv2D)           (None, 2, 2, 512)         1180160   
                                                                 
 max_pooling2d_4 (MaxPooling  (None, 1, 1, 512)        0         
 2D)                                                             
                                                                 
 flatten (Flatten)           (None, 512)               0         
                                                                 
 dropout_2 (Dropout)         (None, 512)               0         
                                                                 
 dense (Dense)               (None, 150)               76950     
                                                                 
 dense_1 (Dense)             (None, 16)                2416      
                                                                 
=================================================================
Total params: 1,647,942
Trainable params: 1,647,942
Non-trainable params: 0
_________________________________________________________________
```

```
batch_size = 32
epochs = 20
model6.compile(loss="categorical_crossentropy", optimizer="adam", metrics=["accuracy"])

model6.fit(X_train, y_train, batch_size=batch_size, epochs=epochs, validation_data =(X_test,y_test))
```

```
Epoch 1/20
297/297 [==============================] - 24s 26ms/step - loss: 2.6850 - accuracy: 0.0799 - val_loss: 2.6759 - val_accuracy:
Epoch 2/20
297/297 [==============================] - 6s 20ms/step - loss: 2.6452 - accuracy: 0.1069 - val_loss: 2.5959 - val_accuracy:
Epoch 3/20
297/297 [==============================] - 7s 23ms/step - loss: 2.4504 - accuracy: 0.1587 - val_loss: 2.3653 - val_accuracy:
Epoch 4/20
297/297 [==============================] - 6s 20ms/step - loss: 2.2415 - accuracy: 0.2152 - val_loss: 2.2230 - val_accuracy:
Epoch 5/20
297/297 [==============================] - 6s 21ms/step - loss: 2.1470 - accuracy: 0.2293 - val_loss: 2.0646 - val_accuracy:
Epoch 6/20
297/297 [==============================] - 6s 20ms/step - loss: 2.0705 - accuracy: 0.2492 - val_loss: 1.9765 - val_accuracy:
Epoch 7/20
297/297 [==============================] - 6s 21ms/step - loss: 2.0312 - accuracy: 0.2588 - val_loss: 1.9788 - val_accuracy:
Epoch 8/20
297/297 [==============================] - 6s 21ms/step - loss: 1.9872 - accuracy: 0.2724 - val_loss: 1.8788 - val_accuracy:
Epoch 9/20
297/297 [==============================] - 6s 21ms/step - loss: 1.9527 - accuracy: 0.2767 - val_loss: 1.9180 - val_accuracy:
Epoch 10/20
297/297 [==============================] - 6s 21ms/step - loss: 1.9329 - accuracy: 0.2806 - val_loss: 1.8812 - val_accuracy:
Epoch 11/20
297/297 [==============================] - 6s 21ms/step - loss: 1.9067 - accuracy: 0.2880 - val_loss: 1.8566 - val_accuracy:
Epoch 12/20
297/297 [==============================] - 6s 22ms/step - loss: 1.8786 - accuracy: 0.3015 - val_loss: 1.8220 - val_accuracy:
Epoch 13/20
```

```
297/297 [==============================] - 6s 20ms/step - loss: 1.8812 - accuracy: 0.2973 - val_loss: 1.8251 - val_accuracy:
Epoch 14/20
297/297 [==============================] - 6s 22ms/step - loss: 1.8384 - accuracy: 0.3082 - val_loss: 1.7828 - val_accuracy:
Epoch 15/20
297/297 [==============================] - 6s 20ms/step - loss: 1.8214 - accuracy: 0.3096 - val_loss: 1.8393 - val_accuracy:
Epoch 16/20
297/297 [==============================] - 6s 21ms/step - loss: 1.8149 - accuracy: 0.3156 - val_loss: 1.8512 - val_accuracy:
Epoch 17/20
297/297 [==============================] - 6s 21ms/step - loss: 1.8067 - accuracy: 0.3131 - val_loss: 1.7558 - val_accuracy:
Epoch 18/20
297/297 [==============================] - 6s 21ms/step - loss: 1.7820 - accuracy: 0.3229 - val_loss: 1.7582 - val_accuracy:
Epoch 19/20
297/297 [==============================] - 6s 22ms/step - loss: 1.7680 - accuracy: 0.3240 - val_loss: 1.7771 - val_accuracy:
Epoch 20/20
297/297 [==============================] - 6s 21ms/step - loss: 1.7563 - accuracy: 0.3268 - val_loss: 1.7703 - val_accuracy:
<keras.callbacks.History at 0x7fc447d31c30>
```

```
path = "/content/gdrive/MyDrive/AgeRecognition"
model6.save(path)
```

```
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_conv
```

✓ 3s completed at 7:40 PM