DIGITAL ASSIGNMENT -1

COURSE: DESIGN AND ANALYSIS OF ALGORITHMS

COURSE CODE:BCSE202L

TEAM E:

AVIRAL JAIN (21BCT0153)

CHANDRASHEKHAR KRISHNAN (21BCE3028)

ANURAG THAKUR (21BCE3944)

YASHVARDHAN SINGH SOLANKI (21BCE3196)

Project (image classification using convolutional neural network)

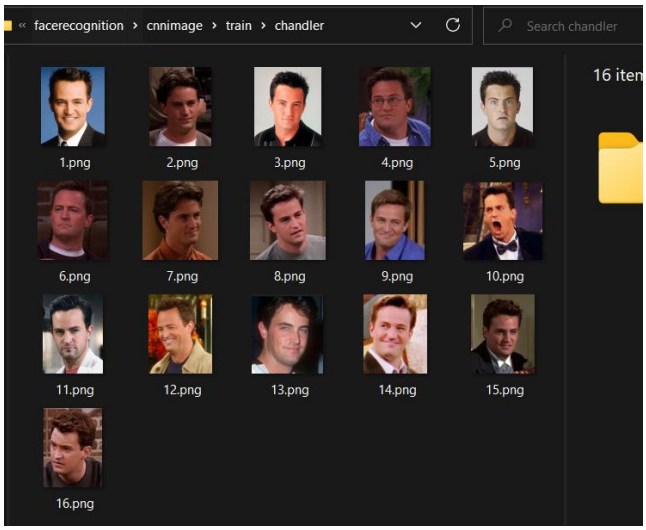
Code:

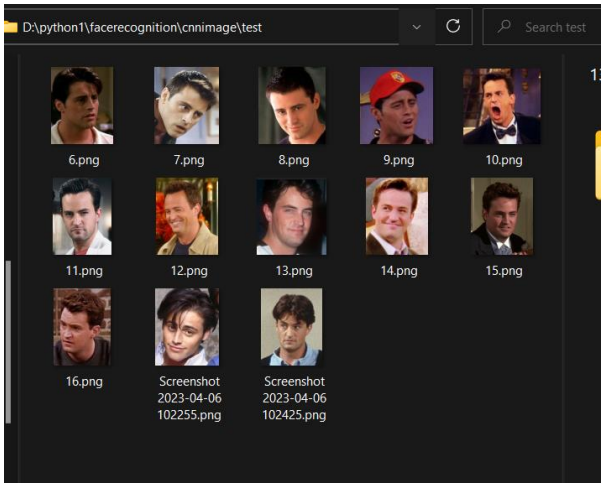
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D,MaxPool2D,Dense,Flatten
import matplotlib.pyplot as plt
import cv2
import os
import numpy as np
```

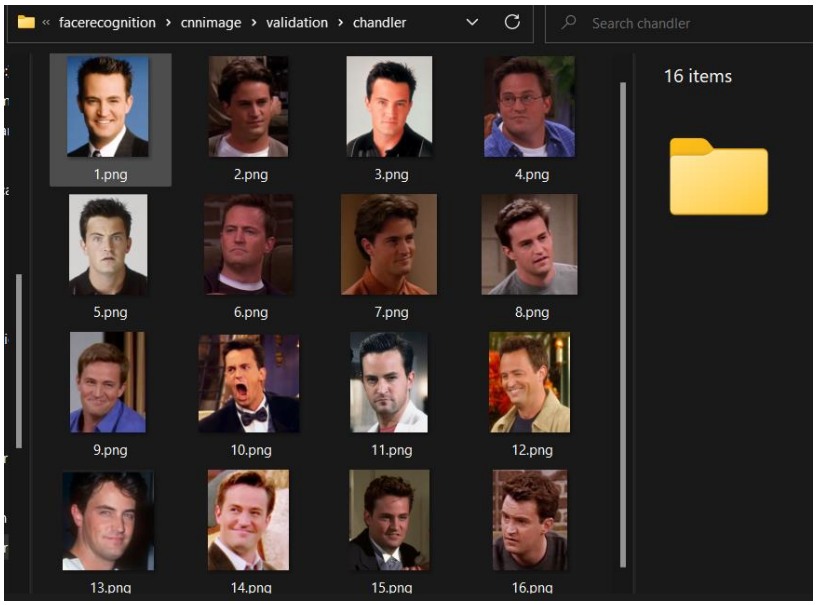
✓ 10.2s

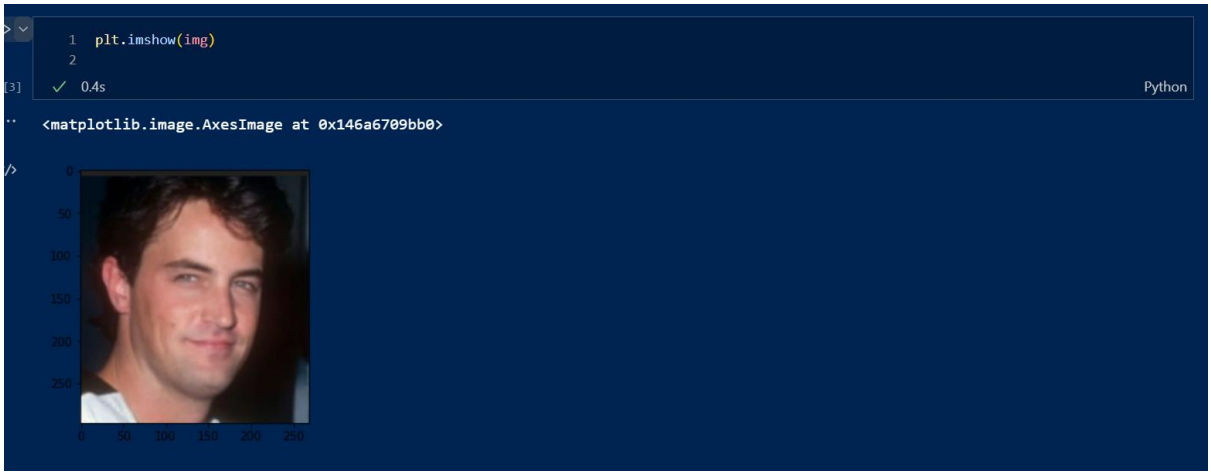
```
img=image.load_img("D:/python1/facerecognition/cnnimage/train/chandler/13.png")
```

✓ 0.3s Python




D:\python1\facerecognition\cnnimage\test
Search test
6.png
7.png
8.png
9.png
10.png
11.png
12.png
13.png
14.png
15.png
16.png
Screenshot 2023-04-06 102255.png
Screenshot 2023-04-06 102425.png


facerecognition › cnnimage › validation › chandler
Search chandler
16 items
1.png
2.png
3.png
4.png
5.png
6.png
7.png
8.png
9.png
10.png
11.png
12.png
13.png
14.png
15.png
16.png


1 plt.imshow(img)
2
[3] ✓ 0.4s
Python
<matplotlib.image.AxesImage at 0x146a6709bb0>
0
50
100
150
200
250
0
50
100
150
200
250

```
1  cv2.imread("D:/python1/facerecognition/cnnimage/train/chandler/13.png")
```

✓ 0.0s

Output exceeds the size limit. Open the full output data in a text editor

```
array([[[ 33,  33,  33],
        [ 33,  33,  33],
        [ 33,  33,  33],
        ...,
        [ 33,  33,  33],
        [ 33,  33,  33],
        [ 33,  33,  33]],

       [[ 33,  33,  33],
        [ 33,  33,  33],
        [ 33,  33,  33],
        ...,
        [ 33,  33,  33],
        [ 33,  33,  33],
        [ 33,  33,  33]],
```

```
1  cv2.imread("D:/python1/facerecognition/cnnimage/train/chandler/13.png").shape
```

✓ 0.0s Python

```
(297, 269, 3)
```

```
1  train=ImageDataGenerator(rescale=1/255)
2  validation=ImageDataGenerator(rescale=1/255)
```

✓ 0.0s Python

```
1  irectory(directory="D:/python1/facerecognition/cnnimage/train",target_size=(200,200),batch_size=3,class_mode='binary')
2
3  low_from_directory(directory="D:/python1/facerecognition/cnnimage/validation",target_size=(200,200),batch_size=3,class_mode='binary')
```

✓ 0.1s Python

```
Found 32 images belonging to 2 classes.
Found 32 images belonging to 2 classes.
```

```
1  train_dataset.class_indices
```

✓ 0.0s Python

```
{'chandler': 0, 'joey': 1}
```

```python
model=Sequential([Conv2D(16,(3,3),activation='relu',
    input_shape=(200,200,3)),MaxPool2D(2,2),
    Conv2D(32,(3,3),activation='relu'),
    MaxPool2D(2,2),Conv2D(64,(3,3),activation='relu'),MaxPool2D(2,2),
    Flatten(),Dense(512,activation='relu'),Dense(1,activation='sigmoid')])
```

✓ 1.4s Python

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

✓ 0.0s Python

+ Code + Markdown

```python
model_fit=model.fit(train_dataset,steps_per_epoch=3,epochs=10,validation_data=validation_dataset)
```

✓ 8.6s Python

```
Epoch 1/10
3/3 [==============================] - 2s 448ms/step - loss: 2.9196 - accuracy: 0.2222 - val_loss: 1.0437 - val_accuracy: 0.5000
Epoch 2/10
3/3 [==============================] - 1s 255ms/step - loss: 1.4127 - accuracy: 0.0000e+00 - val_loss: 0.7002 - val_accuracy: 0.5
Epoch 3/10
3/3 [==============================] - 1s 264ms/step - loss: 0.7042 - accuracy: 0.3750 - val_loss: 0.6328 - val_accuracy: 0.7500
Epoch 4/10
3/3 [==============================] - 1s 257ms/step - loss: 0.5829 - accuracy: 0.7778 - val_loss: 0.7916 - val_accuracy: 0.5000
Epoch 5/10
```

```
Epoch 5/10
3/3 [==============================] - 1s 266ms/step - loss: 0.8923 - accuracy: 0.3333 - val_loss: 0.6594 - val_accuracy: 0.6875
Epoch 6/10
3/3 [==============================] - 1s 257ms/step - loss: 0.7428 - accuracy: 0.4444 - val_loss: 0.6560 - val_accuracy: 0.5938
Epoch 7/10
3/3 [==============================] - 1s 255ms/step - loss: 0.6661 - accuracy: 0.6667 - val_loss: 0.6510 - val_accuracy: 0.9062
Epoch 8/10
3/3 [==============================] - 1s 245ms/step - loss: 0.6421 - accuracy: 1.0000 - val_loss: 0.6445 - val_accuracy: 0.7812
Epoch 9/10
3/3 [==============================] - 1s 266ms/step - loss: 0.6610 - accuracy: 0.6250 - val_loss: 0.6365 - val_accuracy: 0.5938
Epoch 10/10
3/3 [==============================] - 1s 260ms/step - loss: 0.5691 - accuracy: 0.6667 - val_loss: 0.6203 - val_accuracy: 0.5312
```

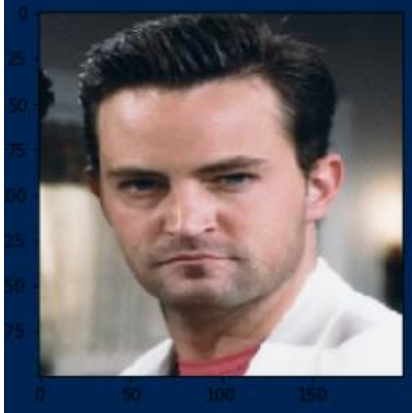
```python
dir_path='D:/python1/facerecognition/cnnimage/test'
for i in os.listdir(dir_path):
    img = image.load_img(dir_path +'//'+i,target_size=(200,200))
    plt.imshow(img)
    plt.show()

    x=image.img_to_array(img)
    x=np.expand_dims(x,axis=0)
    images=np.vstack([x])
    val=model.predict(images)
    if val==0:
        print("this guy is CHANDLER")
    if val==1:
        print("this guy is JOEY")
```

✓ 4.1s Python

Output:

his guy is CHANDLER



his guy is CHANDLER