Govardhan Seshasayee

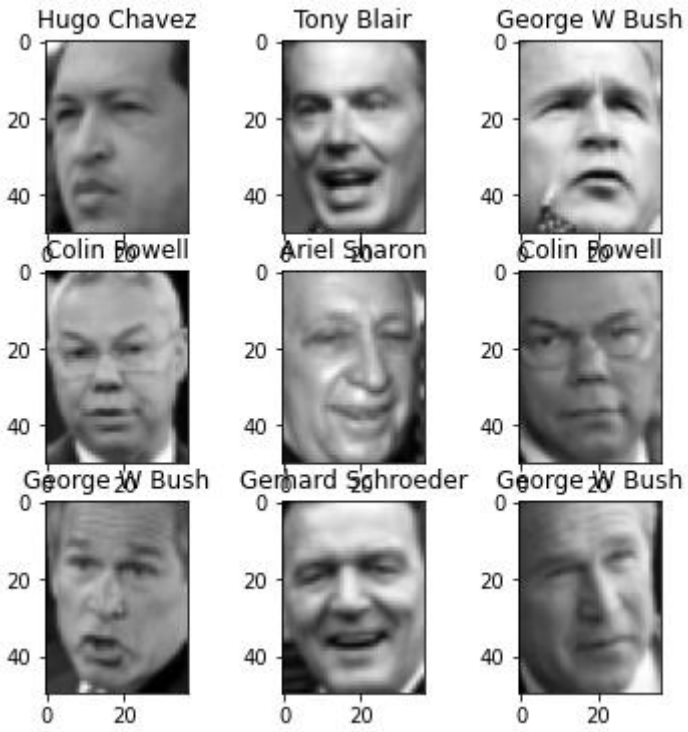
Face feature extraction Project Report

Facial feature extraction is **the process of extracting face component features like eyes, nose, mouth, etc from human face image**. Facial feature extraction is very much important for the initialization of processing techniques like face tracking, facial expression recognition or face recognition.

Principal component analysis (PCA) is the process of computing the principal components and using them to perform a change of basis on the data, sometimes using only the first few principal components and ignoring the rest.

In this project principal component analysis is used to sort the face features as it is generally used for reducing the dimensionality of such datasets that have interpretability but at same time minimizing loss. It is also used in case of denoising and data compression.

The various inbuilt datasets are first imported. An inbuilt dataset lfw_people which contains the basic data of some of the renowned personality is used to do this project. Minimum faces per person is taken as 70 and the resize value of 0.4 is given as an input to the data set after which the target and the data are separated with which the train and test sets are created. The inbuilt sklearn which consists of PCA is used to further solve the dataset and the train data is trained, with also the MLP classifier function for better fit, where the hidden layers are further specified. The validation score obtained by training for several epochs was 74.13%. With this the prediction is made and the results are obtained.



```
y_pred= clf.predict(X_test_trans_1)
print (classification_report(y_test, y_pred, target_names=target_names))
```

```
                   precision    recall  f1-score   support

     Ariel Sharon       0.40      0.50      0.44         4
     Colin Powell       0.77      0.77      0.77        22
  Donald Rumsfeld       0.67      0.75      0.71        16
    George W Bush       0.89      0.88      0.88        56
Gerhard Schroeder       0.50      0.50      0.50        10
      Hugo Chavez       0.67      0.67      0.67         6
       Tony Blair       0.69      0.60      0.64        15

         accuracy                           0.76       129
        macro avg       0.66      0.67      0.66       129
     weighted avg       0.76      0.76      0.76       129
```

Code for the Results obtained:

```
from sklearn.datasets import fetch_lfw_people
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.neural_network import MLPClassifier
```

```
dataset= fetch_lfw_people(min_faces_per_person=70, resize=0.4)
X=dataset.data
y=dataset.target
target_names=dataset.target_names
images=dataset.images
```

```
n, h, w=images.shape
print(n)
print(h)
print(w)
```

```
X.shape
```

```
(1288, 1850)
```

```
len(target_names)
```

```
7
```

```
np.unique(y, return_counts=True)
```

```
(array([0, 1, 2, 3, 4, 5, 6], dtype=int64),
 array([ 77, 236, 121, 530, 109,  71, 144], dtype=int64))
```

```
def plot_grid(images, titles, h,w, rows=3, cols=3):
    plt.figure(figsize=(2*cols,2*rows))
    for i in range(rows*cols):
        plt.subplot(rows,cols, i+1)
        plt.imshow(images[i].reshape(h,w),cmap="gray")
        plt.title(target_names[titles[i]])
plot_grid(X,y,h,w)
```

```
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.1)
```

```
X_train.shape
```

```
(1159, 1850)
```

```
pl=PCA(n_components=500)
pl.fit(X_train)
```

```
PCA(n_components=500)
```

```
X_train_trans_1=pl.transform(X_train)
X_test_trans_1=pl.transform(X_test)
```

```
X_train_trans_1.shape
```

```
(1159, 500)
```

```
clf=MLPClassifier(hidden_layer_sizes=(256,),batch_size=128, verbose=True, early_stopping=True)
clf.fit(X_train_trans_1, y_train)
```

```
y_pred= clf.predict(X_test_trans_1)
print (classification_report(y_test, y_pred, target_names=target_names))
```

```
                   precision    recall  f1-score   support

     Ariel Sharon       0.40      0.50      0.44         4
     Colin Powell       0.77      0.77      0.77        22
  Donald Rumsfeld       0.67      0.75      0.71        16
    George W Bush       0.89      0.88      0.88        56
Gerhard Schroeder       0.50      0.50      0.50        10
      Hugo Chavez       0.67      0.67      0.67         6
       Tony Blair       0.69      0.60      0.64        15

         accuracy                           0.76       129
        macro avg       0.66      0.67      0.66       129
     weighted avg       0.76      0.76      0.76       129
```

```
p=PCA()
p.fit(X_train)
```

```
PCA()
```

```
p.transform(X_train).shape
```

```
(1159, 1159)
```