Capstone Project Report

Name: Sumit Patel

Course: AI and ML (Batch – AUG 2020)

Duration: 10 months

Face Detection and PCA

Problem Statement:

Using PCA create a face recognition system that gives access to only certain people. To implement this, you can use LFW_peoples dataset provided in the scikit-learn library. Given this dataset, use only those classes that have a minimum (use min_faces_per_person = 70, resize = 0.4) 70 images (should give you only 11 classes). Given this subset of images, apply PA to obtain the corresponding eigen face for each class. You can additionally train a classifier for recognition purpose.

Prerequisites

What things you need to install the software and how to install them:

Python 3.6 This setup requires that your machine has latest version of python. The following url https://www.python.org/downloads/ can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: https://www.pythoncentral.io/add-python-to-path-python-is-not-recognized-as-an-internal-or-externalcommand/. Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic. Second and easier option is to download anaconda and use its anaconda prompt to run the commands.

To install anaconda check this url https://www.anaconda.com/download/ You will also need to download and install below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.6 then run below commands in command prompt/terminal to install these packages:

```
pip install -U scikit-learn

pip install numpy

pip install pandas

pip install matplotlib
```

If you have chosen to install anaconda then run below commands in anaconda prompt to install these packages:

```
conda install -c scikit-learn

conda install -c anaconda numpy

conda install -c anaconda pandas

conda install -c anaconda matplotlib
```

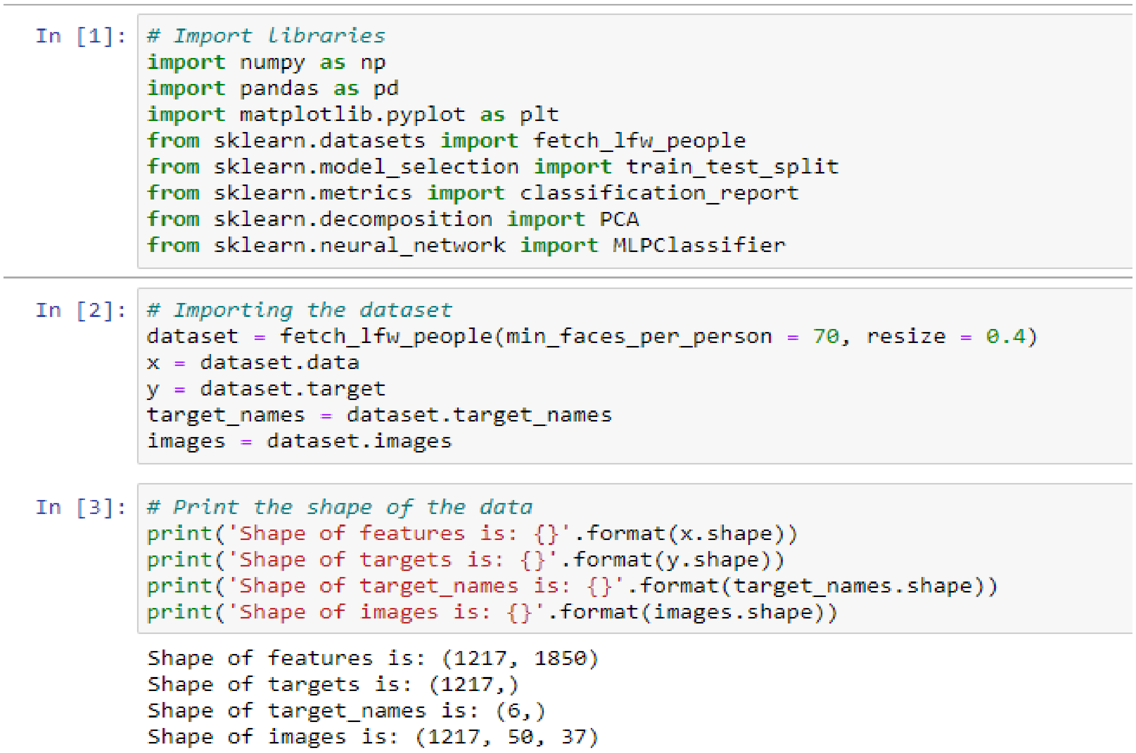
Dataset used:

The data source used for this project is fetch_lfw_people from sklearn.datasets.

Method used for detection

PCA and MLPClassifier

Importing the libraries:



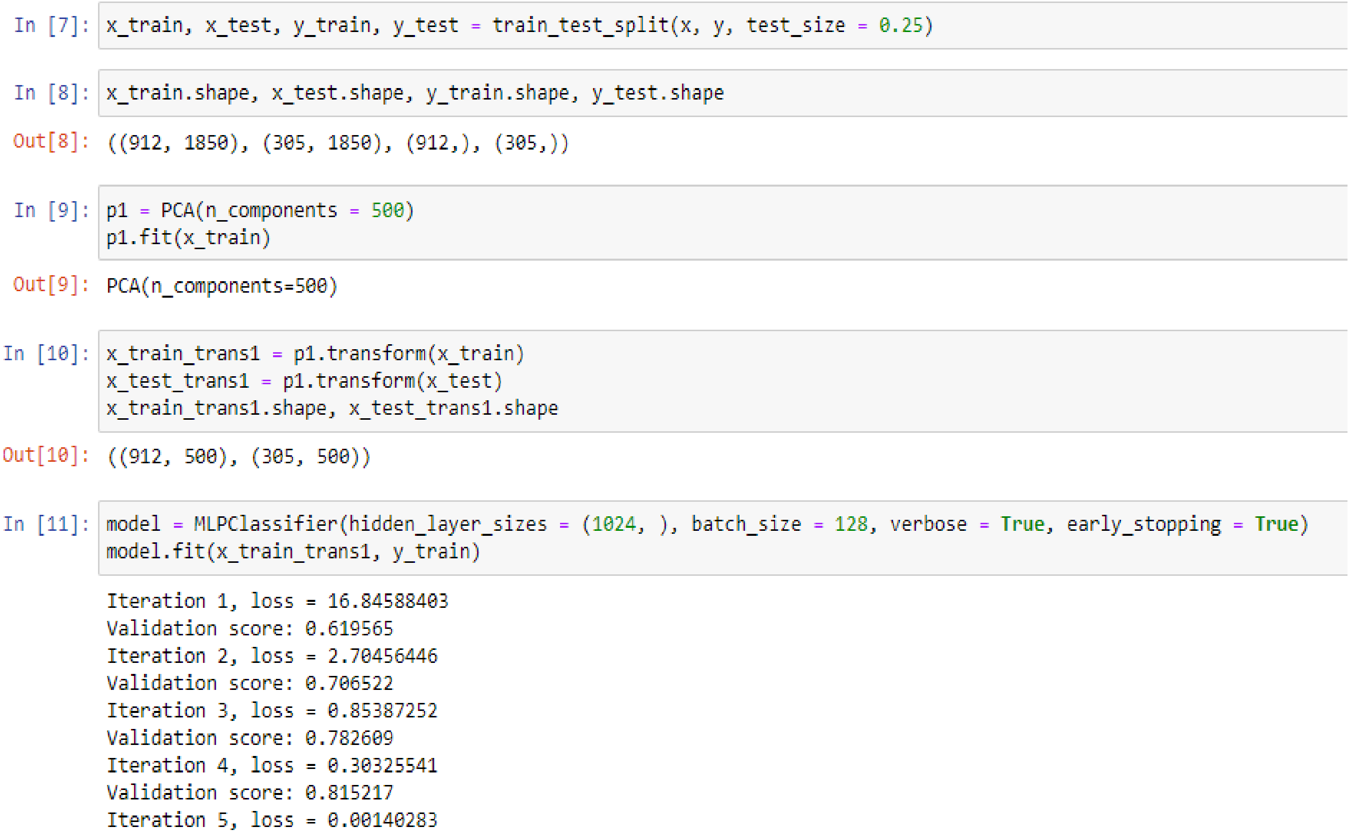
```
In [1]: # Import libraries
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        from sklearn.datasets import fetch_lfw_people
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import classification_report
        from sklearn.decomposition import PCA
        from sklearn.neural_network import MLPClassifier
```

```
In [2]: # Importing the dataset
        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
```

```
In [3]: # Print the shape of the data
        print('Shape of features is: {}'.format(x.shape))
        print('Shape of targets is: {}'.format(y.shape))
        print('Shape of target_names is: {}'.format(target_names.shape))
        print('Shape of images is: {}'.format(images.shape))

        Shape of features is: (1217, 1850)
        Shape of targets is: (1217,)
        Shape of target_names is: (6,)
        Shape of images is: (1217, 50, 37)
```

Training Data:



```
In [7]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25)
```

```
In [8]: x_train.shape, x_test.shape, y_train.shape, y_test.shape

Out[8]: ((912, 1850), (305, 1850), (912,), (305,))
```

```
In [9]: p1 = PCA(n_components = 500)
        p1.fit(x_train)

Out[9]: PCA(n_components=500)
```

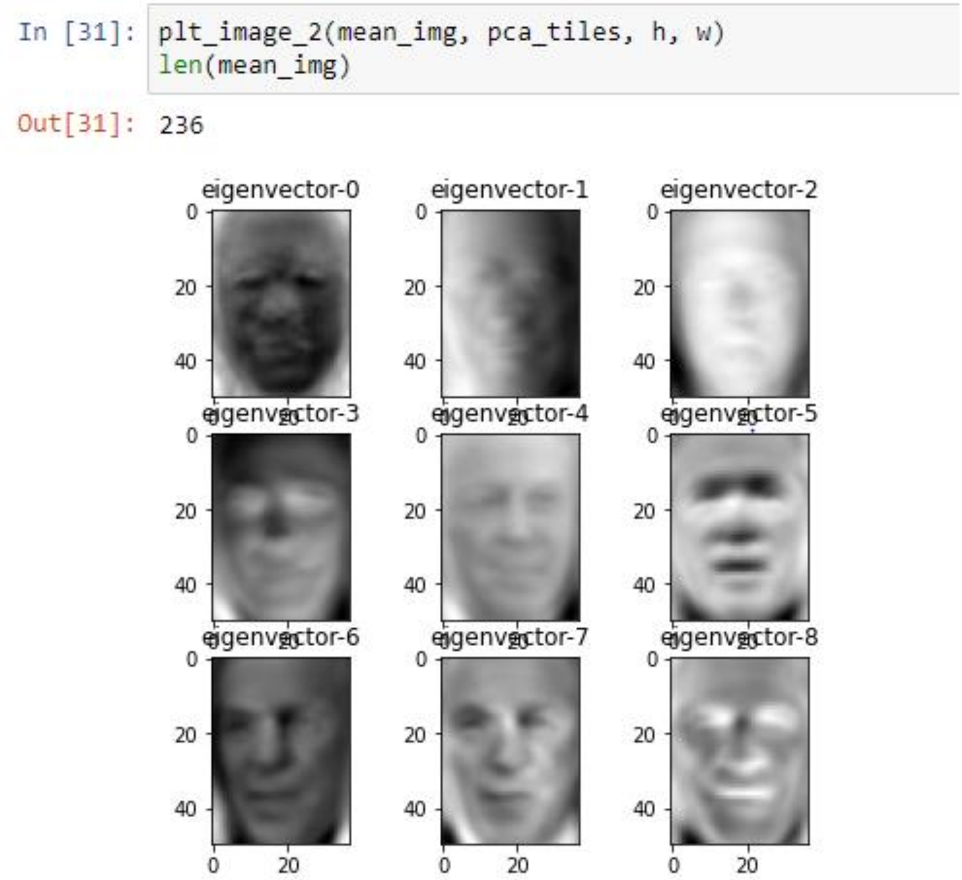
```
In [10]: x_train_trans1 = p1.transform(x_train)
         x_test_trans1 = p1.transform(x_test)
         x_train_trans1.shape, x_test_trans1.shape

Out[10]: ((912, 500), (305, 500))
```

```
In [11]: model = MLPClassifier(hidden_layer_sizes = (1024, ), batch_size = 128, verbose = True, early_stopping = True)
         model.fit(x_train_trans1, y_train)

         Iteration 1, loss = 16.84588403
         Validation score: 0.619565
         Iteration 2, loss = 2.70456446
         Validation score: 0.706522
         Iteration 3, loss = 0.85387252
         Validation score: 0.782609
         Iteration 4, loss = 0.30325541
         Validation score: 0.815217
         Iteration 5, loss = 0.00140283
```

Final Output after PCA



```
In [31]: plt_image_2(mean_img, pca_tiles, h, w)
         len(mean_img)
```

```
Out[31]: 236
```