Graphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Student Name

***Sanatkumar Rajmogali Ippalpalli***

Title of Project Report

***Guided Project 4 – Hashing Querying in Face Datasets***

Table of Contents

[Introduction 4](#_Toc82273686)

[Eckovation theme & Question 5](#_Toc82273687)

[Prerequisites before starting coding 6](#_Toc82273688)

[program DEVELOPMENT steps 8](#_Toc82273689)

[Dataset requirements 8](#_Toc82273690)

[Technique – PCA 8](#_Toc82273691)

[PROGRAM / CODE DEVELOPMENT 9](#_Toc82273692)

[Analysis 16](#_Toc82273693)

[CONCLUSION 18](#_Toc82273694)

[Figure 1 Import libraries and datasets/modules 9](#_Toc82272189)

[Figure 2 Load Datasets and converting Numpy array 9](#_Toc82272190)

[Figure 3 converting Numpy array 10](#_Toc82272191)

[Figure 4 Code for plotting images 10](#_Toc82272192)

[Figure 5 results for plotting images 11](#_Toc82272193)

[Figure 6 Splitting Dataset in Test and Train Data 11](#_Toc82272194)

[Figure 7 Dimensionality Reduction using PCA 12](#_Toc82272195)

[Figure 8 Using PCA reducing to number of components variable 12](#_Toc82272196)

[Figure 9 Verification of number of components variable in Test & Train dataset 13](#_Toc82272197)

[Figure 10 Training with MLPClassifier 13](#_Toc82272198)

[Figure 11 Precision estimation 14](#_Toc82272199)

[Figure 12 Confusion Matrix 14](#_Toc82272200)

[Figure 11 Code for plotting images as prediction 14](#_Toc82272201)

[Figure 14 Prediction vs True Matrix on Plotting of Images 15](#_Toc82272202)

[Figure 15 EigenFaces Plotting 16](#_Toc82272203)

[Figure 16 Accuracy of Model Estimate 16](#_Toc82272204)

***EXECUTIVE SUMMARY***

Science and technology improved many technologies and has guided numerous innovative features which advanced the smart phones and laptops and smart watches that we use daily. Security features designs are meant to protect personal information and accurate biometric recognition. The biometric recognition innovative technology included user defined (4/6/8 digit pins/passwords/patterns), fingerprint sensors and facial recognition. *One of the latest feature is facial recognition is secure, safe and one of the enhanced, best biometric identification process to identify, verify, and authenticate the person using facial features from any photo or video or live to unlock the device.*

Hash method in Python is a module that is used to return the hash value of an object. In programming, the hash method is used to return integer values that are used to compare dictionary keys using a dictionary look up feature.

As part of guided project, one of the features is the hashing coded and implemented to capture the details as mentioned in the question.

# Introduction

Science and technology improved many technologies and has guided numerous innovative features which advanced the smart phones and laptops and smart watches that we use daily. Security features designs are meant to protect personal information and accurate biometric recognition. The biometric recognition innovative technology included user defined (4/6/8 digit pins/passwords/patterns), fingerprint sensors and facial recognition.

In programming, the hash method is used to return integer values that are used to compare dictionary keys using a dictionary look up feature. Python offers **hash()** method to encode the data into unrecognisable value. Hash method in Python is a module that is used to return the hash value of an object. When used, it calls for the **hash()**  of an object which is set by default during the creation of the object by the user.

Advantages of Hashing

* Hashing algorithms today have become essential, especially in cryptography, cloud service, textual tools, data-intensive research, etc.
* Hashing gives a more secure and adjustable method of retrieving data compared to any other data structure. It is quicker than searching for lists and arrays.
* Hashed passwords cannot be modified, stolen, or jeopardized. No well-recognized and efficient key or encryption scheme exists that can be misused.
* Two files can be compared for equality easily through hashing.

Hashing algorithms today have become essential, especially in cryptography, cloud service, textual tools, data-intensive research, etc. Hence Eckovation includes this guided project in the courseware for students to understand, implementation / execute the code themselves.

This report includes the 5W1H about the theme of development of code and running the code with database available over the internet. At the end of the report, the conclusions share the features extracted and useful for next course of activities to gain advantages in the facial recognition activities development.

# Eckovation theme & Question

**Theme : Hashing Querying in Face Datasets**

**Hashing** is most commonly used to implement hash tables and for data encryption. A hash table stores key/value pairs in the form of a list, while the hash function maps the elements of the dataset via a key and generates hash values. A real-world recognition system has to cope with several unseen individuals and determine whether a given face image is registered or not, and thus, certain elements in databases can be found **much faster-using hashing** functions and **classification methods.**

**Question:**

Implement a basic hashing model from scratch that hashes the images. You can use any dataset of few images and can implement a-hash or any other hashing algorithm of your choice. For a-hash, given any images, first resize the image to a suitable size, followed by grayscale conversion of the image. Then mean normalize the image to obtain a binary image, whose sum can be used as a hash value. Using the hash model, encode all the images present inside your directory and then search for images similar to the query image.

# Prerequisites before starting coding

1. Who - Software needed?
2. What - Version / Release of software?
3. Any Prerequisites
4. How - to install the software
5. Which -libraries are needed to execute the problem statement
6. Where – dataset requirements, path location to include in the code
7. When – to use the above feature extraction
8. Who – Software neeed?

Python

1. What- Version / Release of software?

Python version 3.6 (latest version of python)

1. Any Prerequisites

RAM space availability & hard disk space availability

Admin rights to install the software

1. How - to install the software
2. The following url <https://www.python.org/downloads/>can be referred to download python.
3. 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/>
4. Which -libraries are needed to execute the problem statement
5. Sklearn (scikit-learn) ( pip install -U scikit-learn)
6. Numpy (pip install numpy)
7. Matplotlib (pip install matplotlib)
8. openCV
9. Where – dataset requirements, path location to include in the code
10. 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-external-](https://www.pythoncentral.io/add-python-to-path-python-is-not-recognized-as-an-internal-or-external-command/) [command/](https://www.pythoncentral.io/add-python-to-path-python-is-not-recognized-as-an-internal-or-external-command/).
11. Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic.
12. When – to use the above feature extraction
13. When – to use the above feature extraction

Hashing algorithms today have become essential, especially in cryptography, cloud service, textual tools, data-intensive research, etc

# program DEVELOPMENT steps

* Dataset requirement
* Technique selections
* Program / code development
* Analysis

### Dataset requirements

The data source used for this project has been generated using sklearn library dataset titled ***fetch\_lfw\_people***.

It’s the famous dataset which consist of famous personality images. The lfw dataset consists of a database of face photographs designed for studying the problem of unconstrained face recognition. The data set contains more than 13,000 images of faces collected from the web.

### Technique – PCA

**Principal Component Analysis (PCA)**is a machine learning algorithm that is widely used in exploratory data analysis and for making predictive models. It is commonly used for**dimensionality reduction** by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data’s variation as possible.

Faces are high dimensionality data consisting of a number of pixels. Data in high dimensionality is difficult to process and cannot be visualized using simple techniques like scatterplots for 2-dimensional data.What we will do is to use PCA to reduce the high dimensionality of data.

Let us hop to the inscribing carving!

### PROGRAM / CODE DEVELOPMENT

As explained step by step during the lecture by mentor, we would approach steps and understand the basics with brief explanation as needed.

#### Step 1: Import the relevant libraries and applicable datasets/modules

Graphical user interface, text, application

Description automatically generated

Figure Import libraries and datasets/modules

#### Step 2: Load dataset and convert Numpy array

Download the sklearn dataset using **fetch\_lfw\_people** and copy into the disk.

The lfw dataset consists of a database of face photographs designed for studying the problem of unconstrained face recognition. The data set contains more than 13,000 images of faces collected from the web. Each face has been labeled with the name of the person pictured. 1680 of the people pictured have two or more distinct photos in the data set. The images are in grayscale (pixel values = 0 — 255).

We will use the data arrays directly store it in X. We will use this data in our further processing.

Graphical user interface, text, application, email

Description automatically generated

Figure Load Datasets and converting Numpy array

Convert it Numpy arrays

Find the image into the Numpy arrays to learn the shape of the pictures. Using the NumPy shape attribute that returns a tuple with each index having the number of corresponding elements. Define the labels to identify the id of the person to whom the picture belongs as we proceed to target with names/label.

Here, y represents the target which is the label of each picture. The label is further defined by the target\_names variable which consists of the 7 names of the people to be identified.

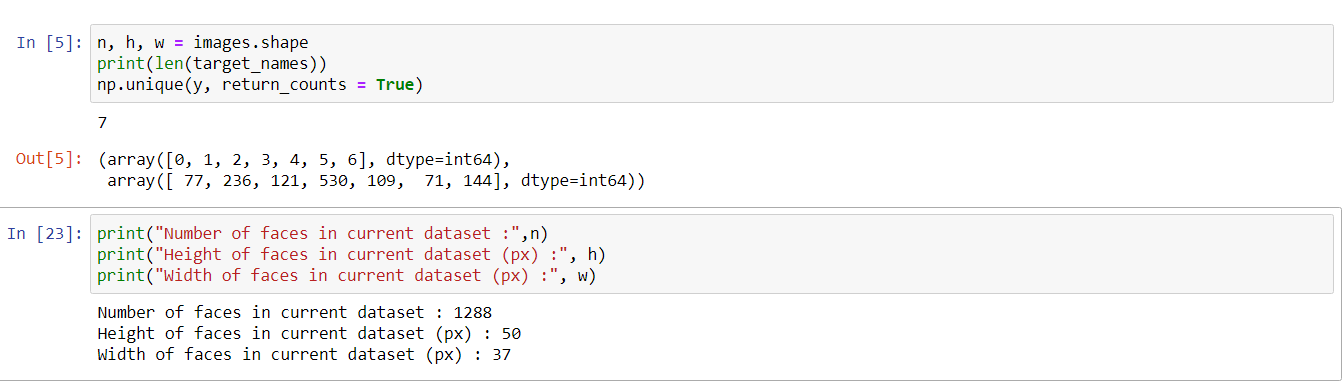


Figure converting Numpy array

As can be seen from the variable explorer, we have 1288 samples (pictures) with a height of 50 px and a width of 37 px (50 x 37 = 1850 features)

#### Step 3: Plot the images

Loop over images to plot the images in a function plot\_img as shown below code.

Graphical user interface, text, application

Description automatically generated

Figure Code for plotting images

The identified celebrity names are:

Hugo Chavez

George W Bush

Gerhard Schroeder

Colin Powell

Ariel Sharon

Tony Blair

A collage of a person's face

Description automatically generated

Figure results for plotting images

#### Step 4: Splitting Dataset in Test/Train

Apply the train\_test\_split module from sklearn.model\_selection and split the data (X-features and y-labels)into training data and testing data, with 10% of the data used for testing and the remaining 90% to train the model.

Graphical user interface, text, application, email

Description automatically generated

Figure Splitting Dataset in Test and Train Data

#### Step 5: Dimensionality Reduction using PCA

Apply PCA from sklearn.decomposition to select the top components to train the model. We have already imported PCA in the first block of code.

Graphical user interface, text, application, email

Description automatically generated

Figure Dimensionality Reduction using PCA

In our case, we have a total of 1159 features in the training set X\_train , and we will reduce them to 250 using PCA (dimensionality reduction).

Text

Description automatically generated

Figure Using PCA reducing to number of components variable

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

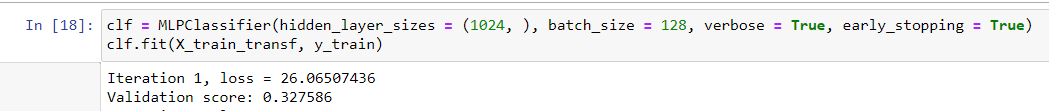
Figure Verification of number of components variable in Test & Train dataset

We will use PCA’s transform on both X\_train and X\_test to reduce the dimensionality.

As can be seen in the screenshot above, the dimensionality of both X\_train and X\_test were reduced through the PCA algorithm.

#### Step 6: Training with Classifier

After completion of dimensionality reduction, now proceed forward for classification.



A picture containing graphical user interface

Description automatically generated

Figure Training with MLPClassifier

#### Step 7: Classification Report and Confusion Matrix

Print the Classification Report which will display the precision, recall, F1-score, and support scores for the model. This gives a deeper intuition of the classifier’s behaviour.

Text

Description automatically generated with medium confidence

Figure Precision estimation

The Confusion Matrix prints the values of True Positives, False Positives, and False Negatives and gives an overview of how the classifier is.

Graphical user interface, text, application, email

Description automatically generated

Figure Confusion Matrix

#### Step 8: Plotting the images

Apply the prediction method to plot images.

Text

Description automatically generated

Figure Code for plotting images as prediction

Let us now plot the results of the prediction on a portion of the test set:

Text

Description automatically generated

A collage of a person's face

Description automatically generated

Figure Prediction vs True Matrix on Plotting of Images

#### Step 9: Plotting the EigenFaces

Apply the prediction method to plot EigenFaces.

Text

Description automatically generated with low confidence

A collage of a person's face

Description automatically generated with medium confidence

Figure EigenFaces Plotting

#### Step 10: Accuracy of the Model Generated

The final estimation of accuracy of the model shall give confirmation on the total train dataset.

Graphical user interface, text, application

Description automatically generated

Figure Accuracy of Model Estimate

### Analysis

Though we didn’t great score (~0.767) in the first attempt but improvement in the code while changing number of components as illustrated /commented in the program shall result in better accuracy score.

As we conclude, number assigned to min\_faces\_per\_person variable modified to (200) and number of components variable amended to 200 resulting accuracy level, f1-score 0.922.

Hence, the improvements in the code with time with multiple attempts may be checked and justified for the accuracy score.

This entire program runs within 5 seconds.

# CONCLUSION

In this guided project, we built a facial feature extraction / recognition model using dimensionality reduction technique PCA and classification technique MLCP Classifier. The Principal Component Analysis algorithm was used to reduce the dimensions of pixel values of the images. This is followed by using MLCP Classifier for classification by finding the best estimator by number of components tuning. We were able to classify the portraits and got an accuracy score of 0.767.

On further analysis for tuning on multiple variables as dynamics, number assigned to min\_faces\_per\_person variable modified to (200) and number of components variable amended to 200 resulting accuracy level, f1-score 0.922.

Hence, the improvements in the code with time with multiple attempts may be checked and justified for the accuracy score.

This entire program runs within few seconds.

references:

1. <https://www.geeksforgeeks.org/python-hash-method/>
2. <https://www.edureka.co/blog/hash-in-python/>
3. <https://www.geeksforgeeks.org/importance-of-hashing/>