**Kathmandu University**

**Dhulikhel, Kavrepalanchok**



**A Project Report**

**on**

**“Principal Component Analysis in Image Processing”**

**[Course Code: MCSC 202]**

**Submitted by**

**Bibash Acharya**

**Registration No: 022409-17**

**Group: Computer Engineering**

**Batch: 2019**

**Submitted to:**

**Dr. Gokul KC**

**Department of Mathematics**

**Submission Date: 1st September, 2023**

# Abstract

The proposal has been drafted in order to meet the requirements of course MCSC 202 (Numerical Methods) offered by the Department of Mathematics, Kathmandu University. In this project, I intend to work in Principal Component Analysis in Image Processing. Principal Component Analysis (PCA) in image processing is a technique used for dimensionality reduction, feature extraction, and data compression. It transforms high-dimensional image data into a lower-dimensional form while retaining most of the significant information. PCA involves flattening images into vectors, calculating the covariance matrix, and finding its eigenvalues and eigenvectors. In facial recognition, eigenfaces are the principal components (eigenvectors) derived from a set of facial images. They represent the "basis images" that capture the most important features distinguishing one face from another.

**Keywords:** Principal Component Analysis (PCA), Eigen Faces

# Table of Contents

Abstract i

List of figures iv

List of Abbreviations/Acronyms v

Chapter 1: Introduction 1

1.1 Background 1

1.2 Objectives 2

1.2 Motive and Significance 2

Chapter 2: Related Works 3

2.1 Face Recognition using PCA 3

2.2 Principal Component Analysis testing on Image data 4

Chapter 3: Implementation 5

3.1. Tools Used 5

3.1.1 Anaconda 5

3.2 Packages and Libraries Used 5

3.3 Code Logic 6

3.4 Demonstration 9

Chapter 4: Discussion on the Achievements 12

4.1 Discussion 12

4.2 Features 12

4.2.1. Successful Dimensionality Reduction: 12

4.2.2 Data Compression: 12

4.2.3. Enhanced Computational Efficiency: 12

Chapter 5. Conclusion and Recommendation 13

5.1 Limitations 13

5.2 Future Enhancements 13

Chapter 6: Source Code 14

Chapter 7: Video Demonstration 14

References 15

# List of figures

Figure 3.3(a): Importing PCA Algorithm in main file 6

Figure 3.3(b): PCA Algorithm Class in main file 6

Figure 3.3(c): Logic for Recorded Files 7

Figure 3.3(d): Logic for Video Files 7

Figure 3.3(e): Logic for Image Files 8

Figure 3.3(f): Ending Code for main file 8

Figure 3.4(a): Running Face\_Recognition.py file 9

Figure 3.4(b): Running Command in Code Editor Terminal 9

Figure 3.4(c): Original Image 10

Figure 3.4(d): Eigen Face 0 10

Figure 3.4(e): After PCA Image 11

# 

# List of Abbreviations/Acronyms

PCA = Principal Component Analysis

ML = Machine Learning

# Chapter 1: Introduction

## Background

Image processing involves analyzing and manipulating visual data, often represented as high-dimensional matrices of pixel values. With the increasing size and complexity of image datasets, efficiently processing and analyzing these large volumes of data becomes a significant challenge. Traditional methods may struggle with the computational demands and the "curse of dimensionality," where the performance of algorithms deteriorates as the number of dimensions increases.

Principal Component Analysis (PCA) emerges as a powerful solution to this problem by reducing the dimensionality of image data while preserving its most critical features.

In image processing, PCA is particularly valuable due to the high dimensionality inherent in images. For instance, a typical grayscale image of 100×100100 \times 100100×100 pixels has 10,000 dimensions. When applied to a collection of such images, PCA identifies the directions of maximum variance, represented by eigenvectors (also known as principal components), and reduces the data's dimensionality by projecting it onto these principal components. The corresponding eigenvalues indicate the amount of variance captured by each principal component, guiding the selection of the most informative components.

One notable application of PCA in image processing is in facial recognition, where it is used to derive "eigenfaces." These eigenfaces are the principal components that capture the most distinctive features of human faces, enabling efficient and accurate recognition. By reducing the dimensionality of facial images while retaining the most significant variations, PCA allows for the creation of a compact and informative representation of faces, facilitating tasks such as identity verification and face detection.

Overall, PCA has become a fundamental tool in image processing, offering a robust method for dimensionality reduction, noise reduction, and feature extraction. It enhances the efficiency and effectiveness of image analysis techniques, making it an essential component in modern image processing workflows.

For this project, I have used Anaconda ( A Python Data Science Distributions ) along with other Python Data Science Libraries like numpy, scipy, and opencv.

## 1.2 Objectives

When the project is completed, I aim to achieve the following objectives:

1. To apply PCA to reduce the dimensionality of image data while retaining the most significant features, enhancing computational efficiency in image analysis tasks.
2. To extract key features from images, such as eigenfaces in facial recognition, to improve the accuracy and effectiveness of classification and recognition algorithms.

## Motive and Significance

The primary motive behind employing Principal Component Analysis (PCA) in image processing is to address the challenges associated with high-dimensional data. Images, inherently high-dimensional, pose significant computational and analytical difficulties, particularly when dealing with large datasets. By reducing the dimensionality of image data, PCA simplifies the complexity, making it more manageable for various tasks such as classification, recognition, and compression.

The significance of PCA in image processing lies in its ability to extract the most important features from images while discarding redundant or less informative data. This leads to several key benefits: Enhanced Computational Efficiency, Improved Accuracy,and Data Compression.

Therefore, PCA serves as a foundational technique in image processing, enabling more efficient and effective handling of complex visual data, and contributing to advancements in various fields such as computer vision, medical imaging, and multimedia applications.

# Chapter 2: Related Works

## 2.1 Face Recognition using PCA

This is a similar project done by an Indian Programmer which applies Principal Component Analysis to solve problem of Face Recognition (Code Heroku, 2019).



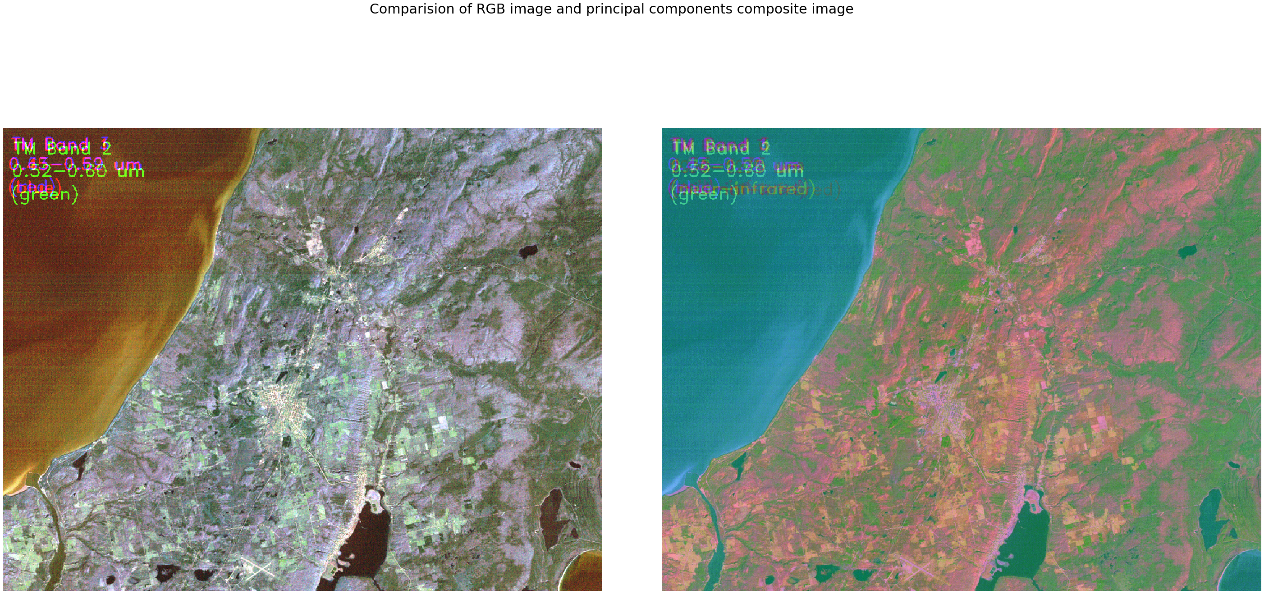
*fig 2.1: Face Recognition using PCA (Code Heroku Project)*

Mint is a free personal finance management tool developed by Intuit that allows users to manage their finances in one place. Mint allows you to create a budget and track your spending in real-time. You can set spending limits for different categories such as groceries, dining out, and entertainment, and Mint will track your spending in those categories. Mint allows you to track your investments and monitor your portfolio performance. You can see your asset allocation and compare your returns to market benchmarks (Mint, n.d.).

You can set financial goals such as saving for a down payment on a house or paying off debt. Mint will help you track your progress towards those goals.

## 2.2 Principal Component Analysis testing on Image data

This project involves application of PCA technique on image data and assessing its performance in terms of information retention and compressibility (Kumar, 2020)



*Fig 2.2: Original vs PCA Image (Principal Component Analysis testing on Image data project)*

# Chapter 3: Implementation

## 3.1. Tools Used

### 3.1.1 Anaconda

The Anaconda Distribution is a popular open-source distribution of the Python and R programming languages. It's widely used in data science,machine learning, and scientific computing due to its ease of use and comprehensive package management (Anaconda.org, 2024).

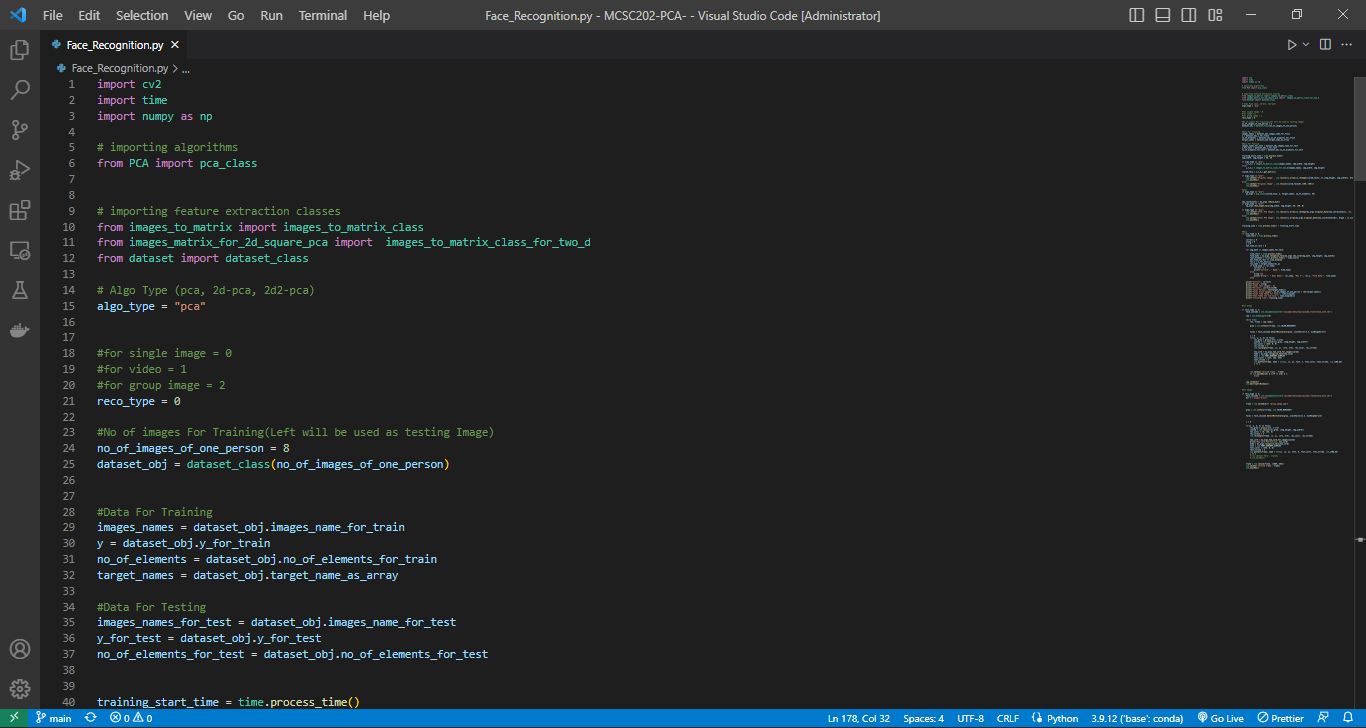
## 3.2 Packages and Libraries Used

* numpy
* opencv
* scipy

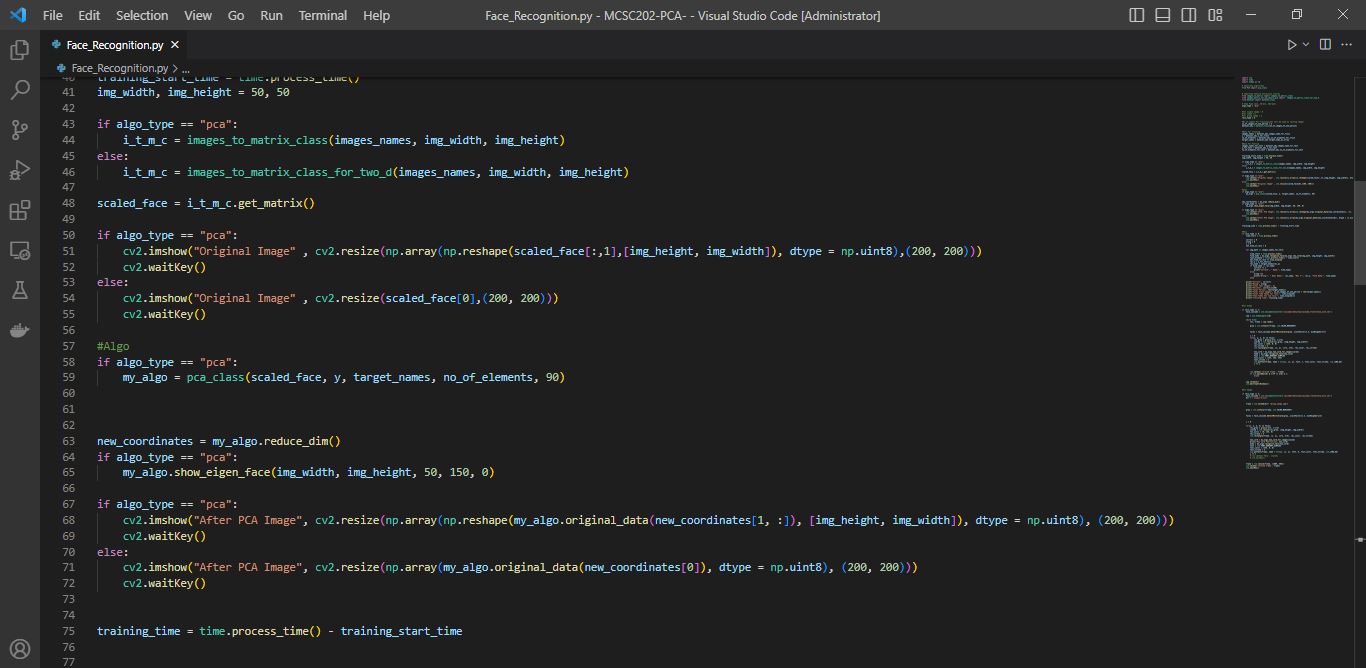
## 3.3 Code Logic

The project uses ORL dataset. The main code to run is in Face\_Recognition.py file which imports PCA Algorithms code modules from PCA.py and other necessary file.

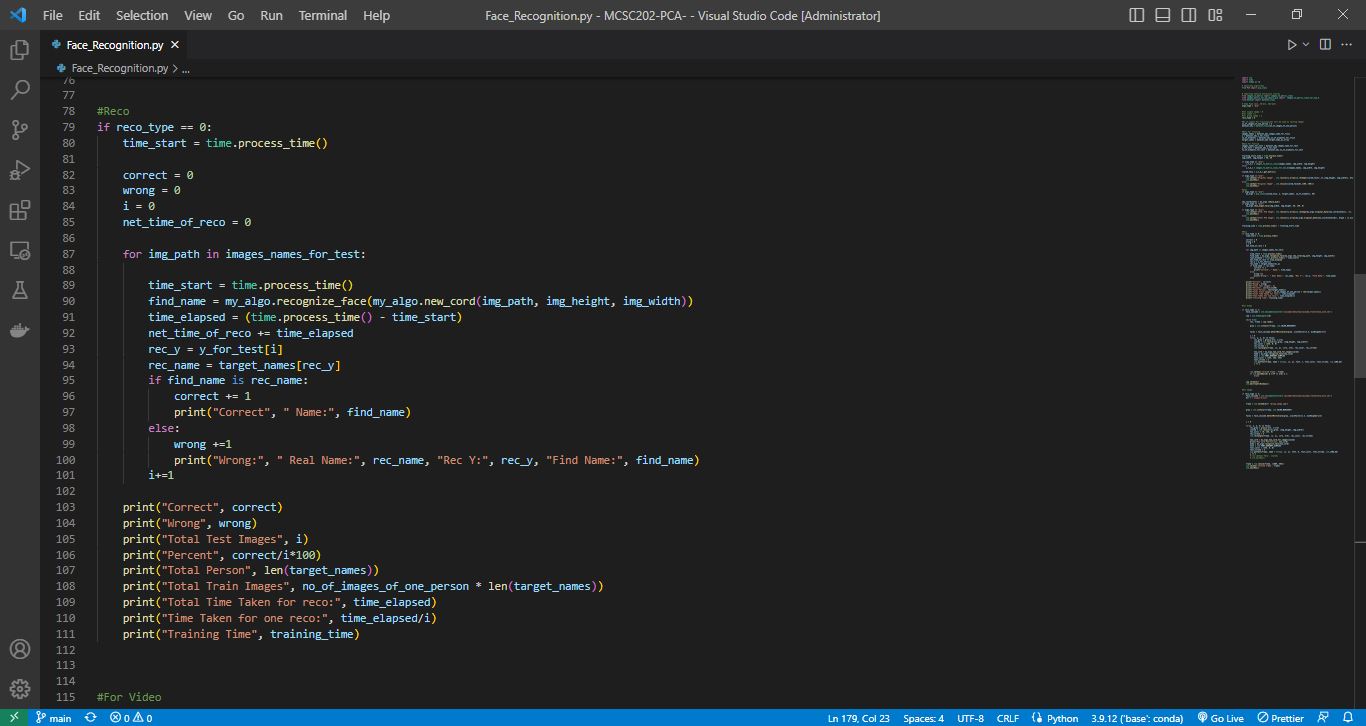
Here are some screenshots from our project available:

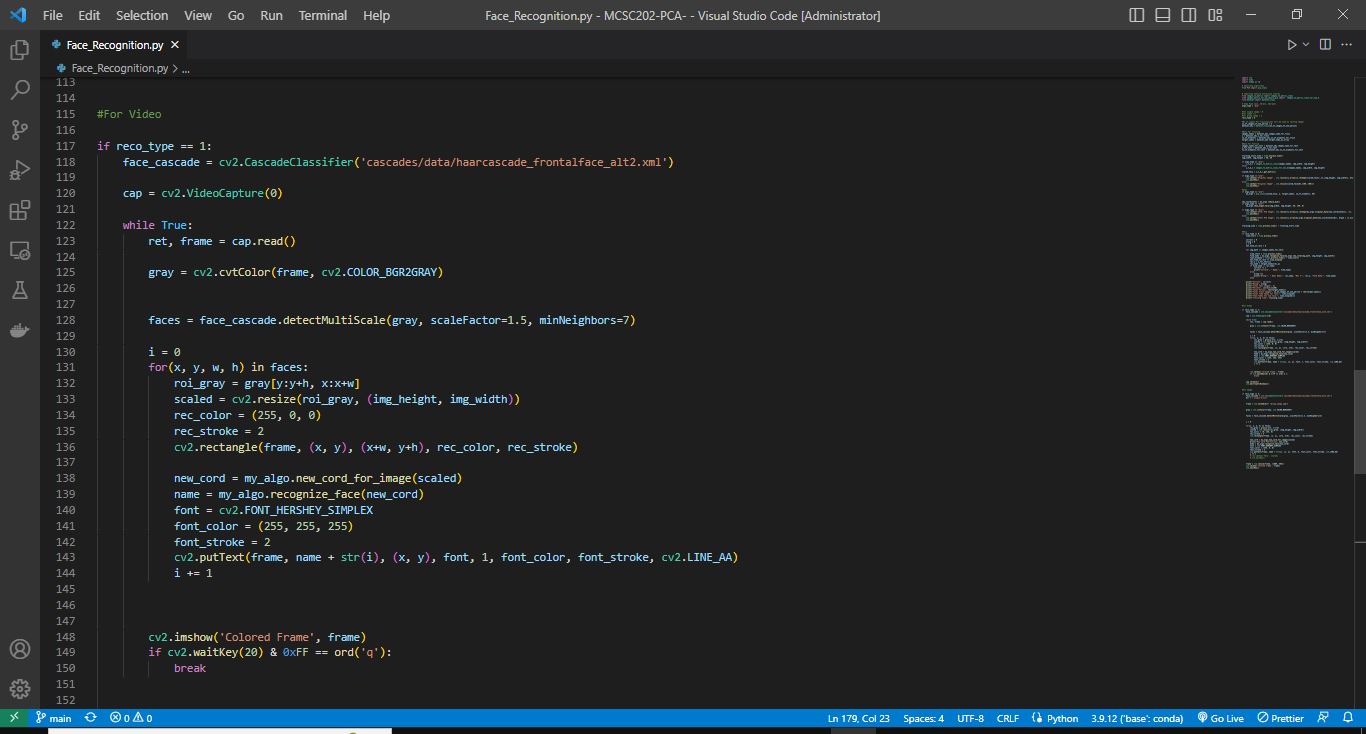


*fig 3.3 (a): Importing PCA Algorithm in main file*

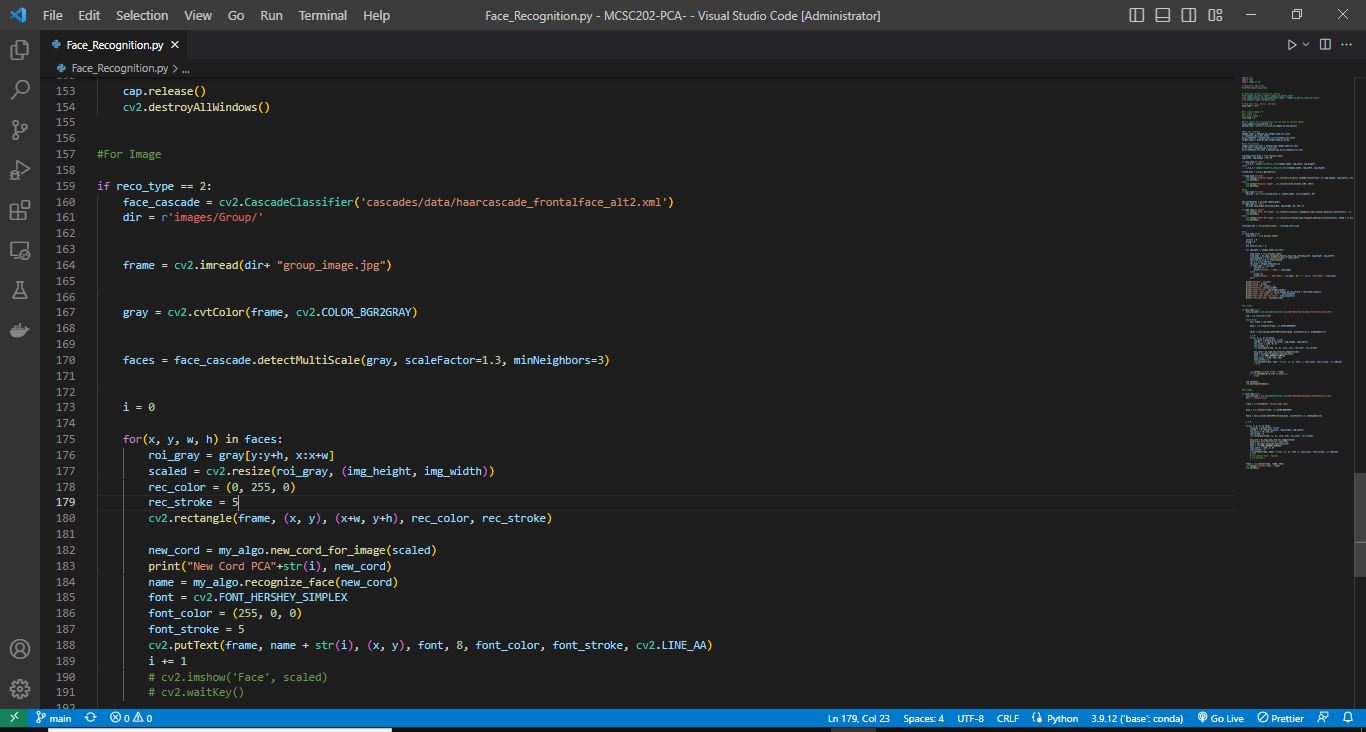


*fig 3.3 (b): PCA Algorithm Class in main file*

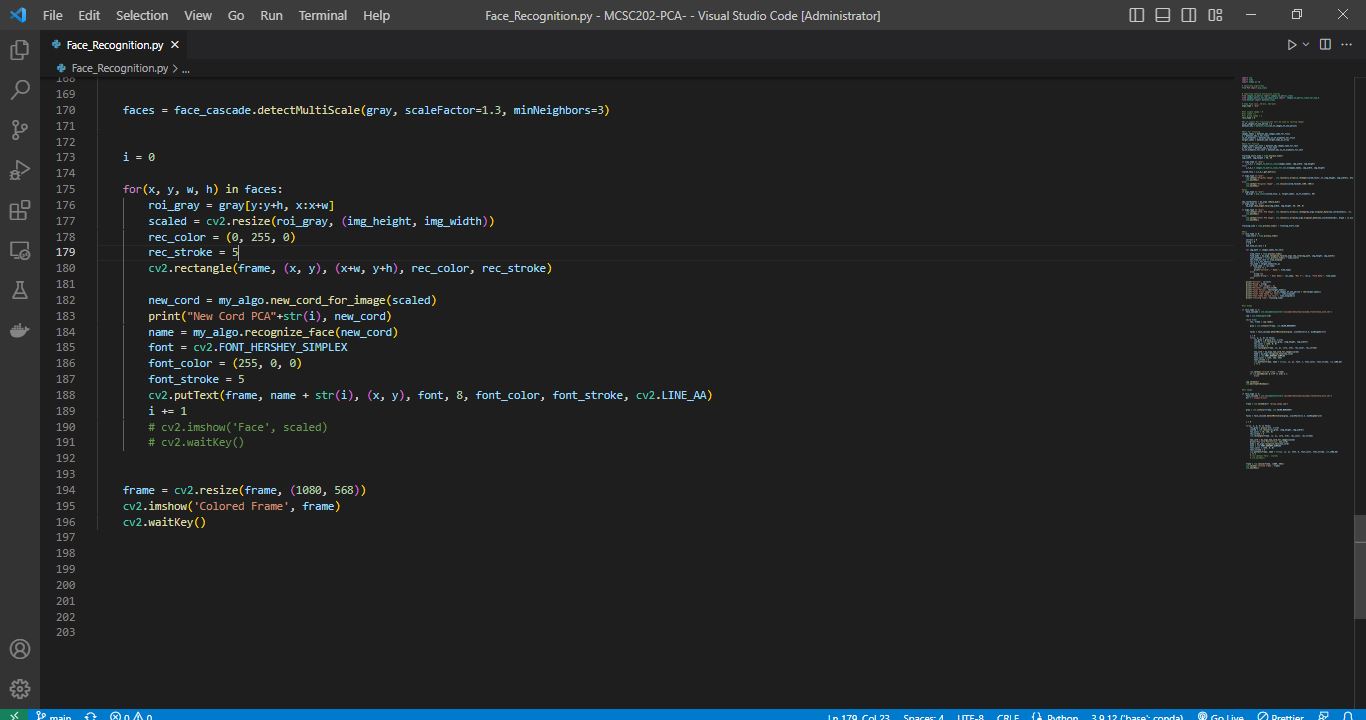
* fig 3.3 (c): Logic for Recorded Files*

**

*fig 3.3 (d): Logic for Video Files*

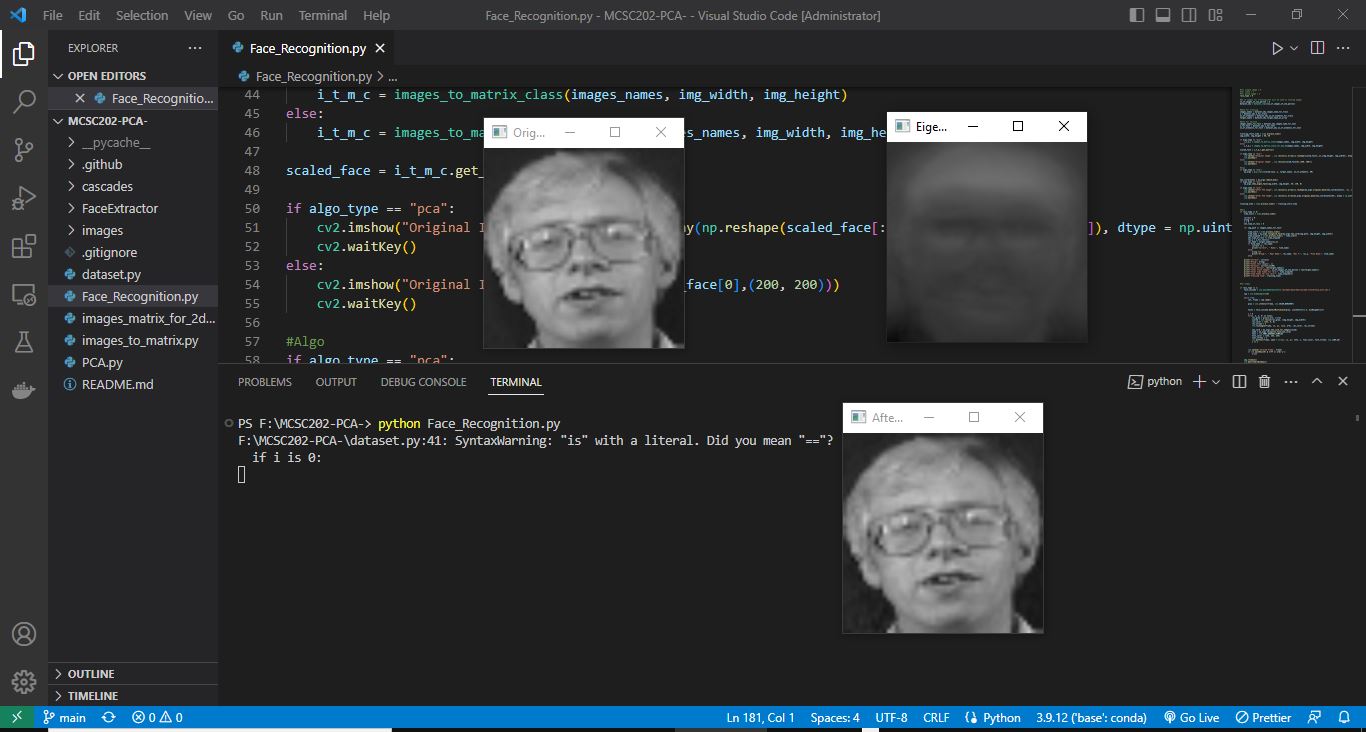
**

*fig 3.3 (e): Logic for Image Files*

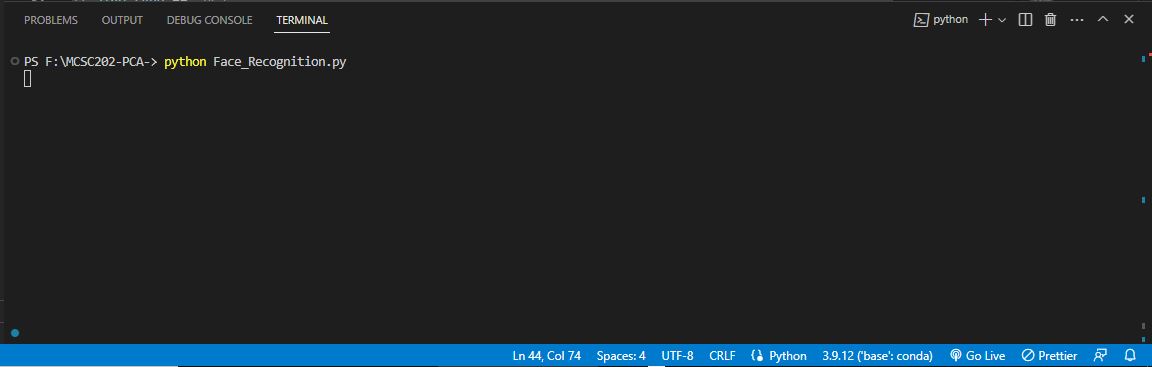
**

*fig 3.3 (f): Ending Code for main file*

## 3.4 Demonstration



*fig 3.4(a): Running Face\_Recognition.py file*

**

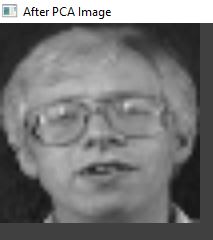
*fig 3.4(b): Running Command in Code Editor Terminal*

**

*fig 3.4(c): Original Image*

**

*fig 3.4(d): Eigen Face 0*

**

*fig 3.4(e): After PCA Image*

# Chapter 4: Discussion on the Achievements

Although many difficulties were faced in building this project, the project ran successfully, therefore, the objective was achieved,. So, basically in this section achievements of this project has been discussed.

## 4.1 Discussion

The application of Principal Component Analysis (PCA) in this image processing project has led to several notable achievements, demonstrating its effectiveness in addressing the challenges associated with high-dimensional image data.

## 4.2 Features

The features available in our project are listed below.

### 4.2.1. Successful Dimensionality Reduction

PCA effectively reduced the dimensionality of the image data, transforming large datasets into more manageable forms without significant loss of critical information.

### 4.2.2 Data Compression

PCA enabled effective data compression, significantly reducing the storage requirements for large image datasets.

### 4.2.3. Enhanced Computational Efficiency

The reduction in dimensionality and the elimination of noise led to enhanced computational efficiency across various image processing tasks.

# Chapter 5. Conclusion and Recommendation

After all the hard work of several weeks, the project turned out as we expected. Objectives were successfully met for implementing learning and coding algorithms for Principal Components Analysis in Image Processing There were shortcomings in different sections but altogether the project demonstration in satisfactory.

## 5.1 Limitations

Some of the shortcomings of this Principal Component Analysis Projects are:

* It is limited to only Images File Data only.
* It is accurate for small dataset.

## 5.2 Future Enhancements

If I plan to continue the development of our project in the future, some of the enhancements could be:

* To expand this project to work on video files .
* To work on larger latasets

# Chapter 6: Source Code

<https://github.com/acharyabibash/MCSC202-PCA->

# Chapter 7: Video Demonstration

<https://www.dropbox.com/scl/fi/hn7s8mgskge20vtpadnum/PCA_Demo.mp4?rlkey=hbkkrzfbbtwjtek2d86j6opq5&st=3f9wxlw9&dl=0>

# References

Anaconda.org. (2024). *About-us*. Retrieved from Anaconda.com: https://www.anaconda.com/about-us

Code Heroku. (2019). *Intro to Machine Learning*. Retrieved from github.com: https://github.com/codeheroku/Introduction-to-Machine-Learning/tree/master/Face%20Recognition%20Using%20PCA

Kumar, S. (2020). *Principal-Component-Analysis-testing-on-Image-data*. Retrieved from github.com: https://github.com/Skumarr53/Principal-Component-Analysis-testing-on-Image-data