Face Recognition - Eigenfaces and Fisherfaces algorithms using Python

CSE 555 – Introduction to Pattern Recognition

By

Satya Lakshmipathi Raju Sampathirao

UB Person # 50097219

UBIT Name: satyalak

Project Description:

Face recognition and detection is done in the project using two algorithms - Eigenfaces and Fisherfaces. The data set used is the Yale Face database, normalized and cropped manually (Credits: Prof. Deb Roy, MIT Media Lab). This dataset has 165 images of 15 subjects (subject01, subject02, etc.). There are 11 images per subject, one for each of the following facial expressions or configurations: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink.

N images are randomly selected from each subject (N < 11). For each experiment, 15N images are chosen from the database as training samples, and the rest as testing samples.

Three values of N are chosen N = {3, 5, 7} and 10 experiments for each N are conducted using both the Eigenfaces and Fisherfaces algorithms. The experimental results are collected to analyze the performance of the algorithms. The average misclassification rate is plotted against the number of dimensions chosen for the algorithm.

Description of the algorithms:

The main problem with the representation of images is their dimension. A m x n array of pixels will be equal to mn dimensions which is very high. So to deal with images, first we should reduce their dimensions and work with them. We discuss the implementation details of Eigenfaces and Fisherfaces algorithms and their uses. Later we will see the experimental results and their analysis.

- **Eigenfaces algorithm:**

  The approach of using Eigenfaces for recognition was developed by Sirovich and Kirby (1987) and used by Matthew Turk and Alex Pentland in face classification. The Eigenfaces algorithm we implemented is based on the Principal Component Analysis (PCA) proposed by Karl Pearson and Harold Hotelling individually which turns a set of correlated variables into a smaller set of uncorrelated variables. The Principal Component method finds the directions with the greatest variance called principal components.

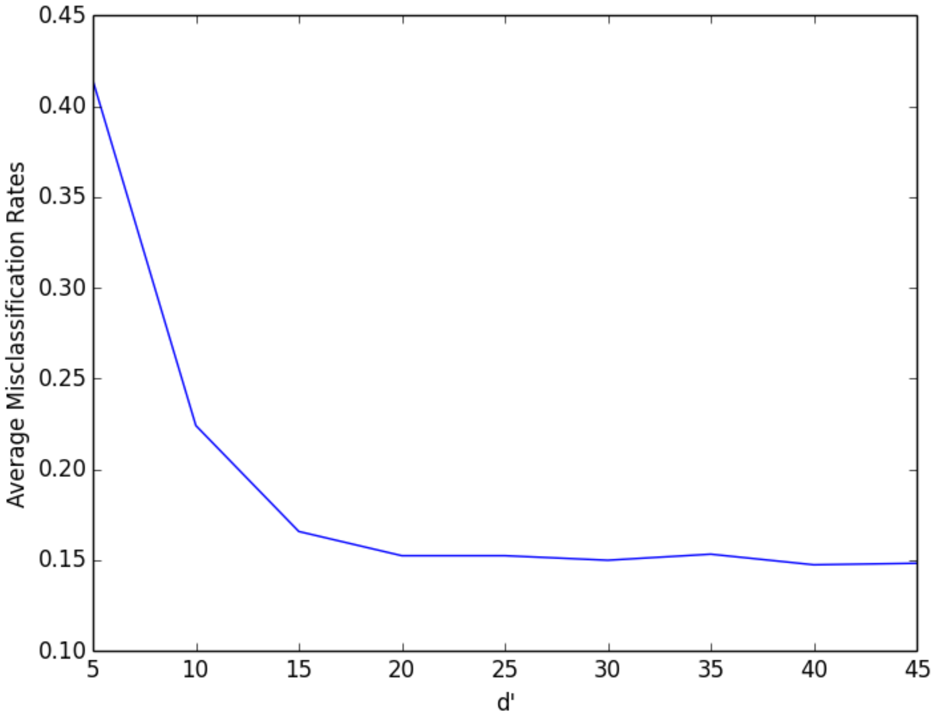
- **Fisherfaces algorithm:**

  The Fisherfaces algorithm is based on Sir R. A Fisher's Linear Discriminant Analysis (LDA) to find a linear combination of features which characterizes or separates the faces into different classes. Fisherfaces algorithm also used PCA to first reduce the dimensions and then finds a linear combination of features.

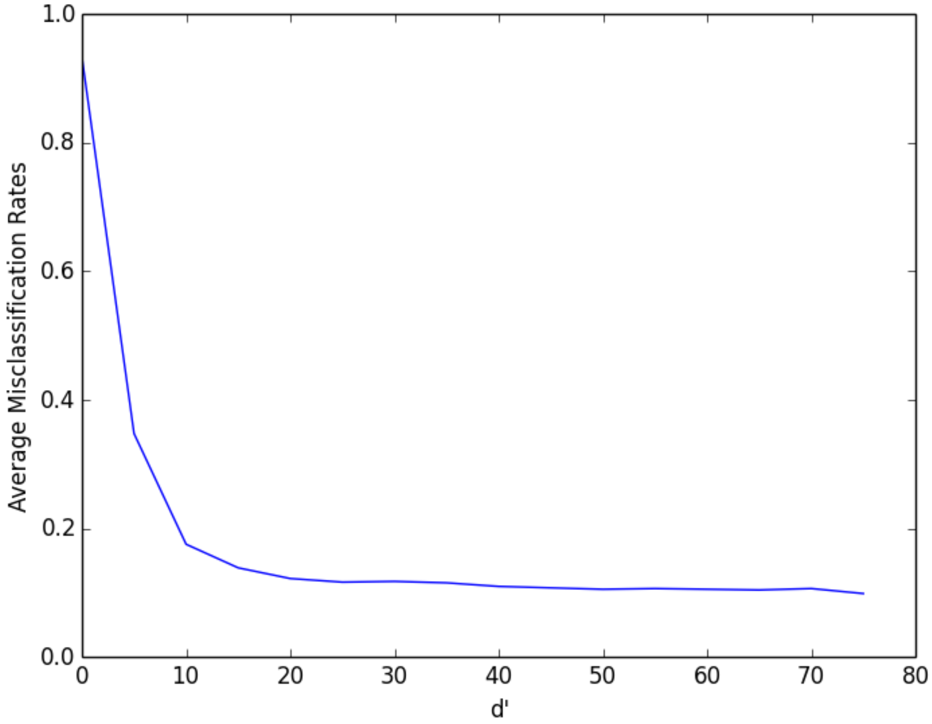
Experimental Results:

The following results show the graphs plotted for both Eigenfaces and Fisherfaces algorithms for 3 different values of N = {3,5,7}

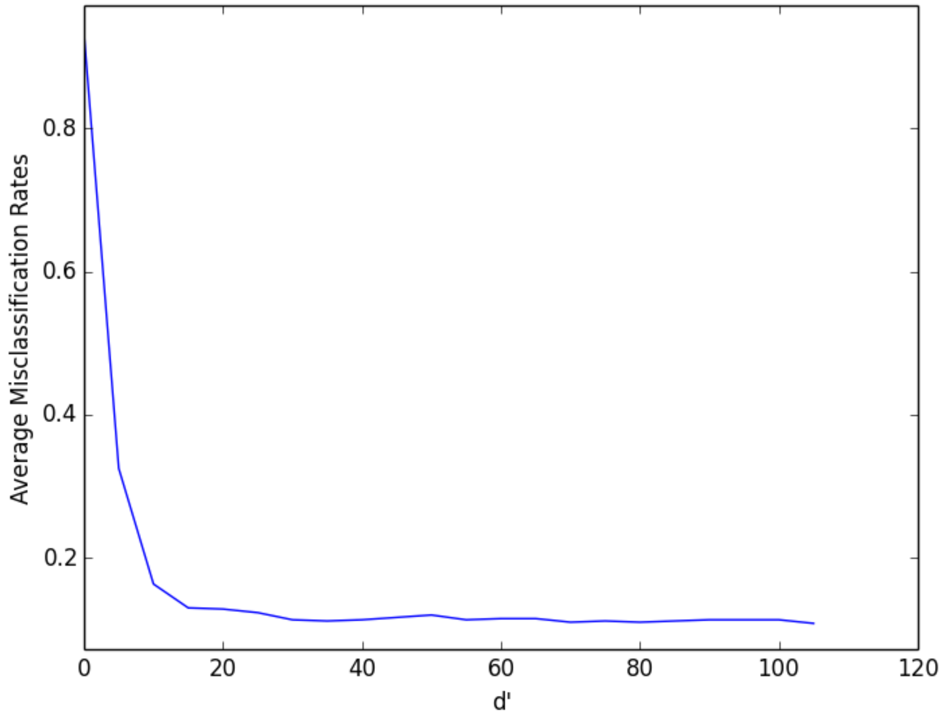
Eigen Faces with N=3



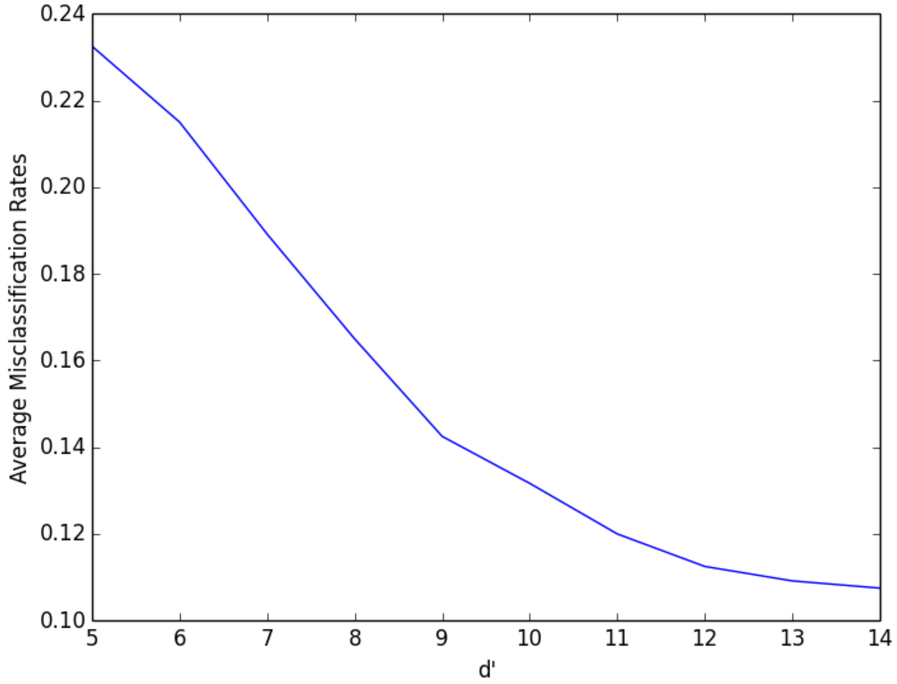
Eigen Faces with N=5



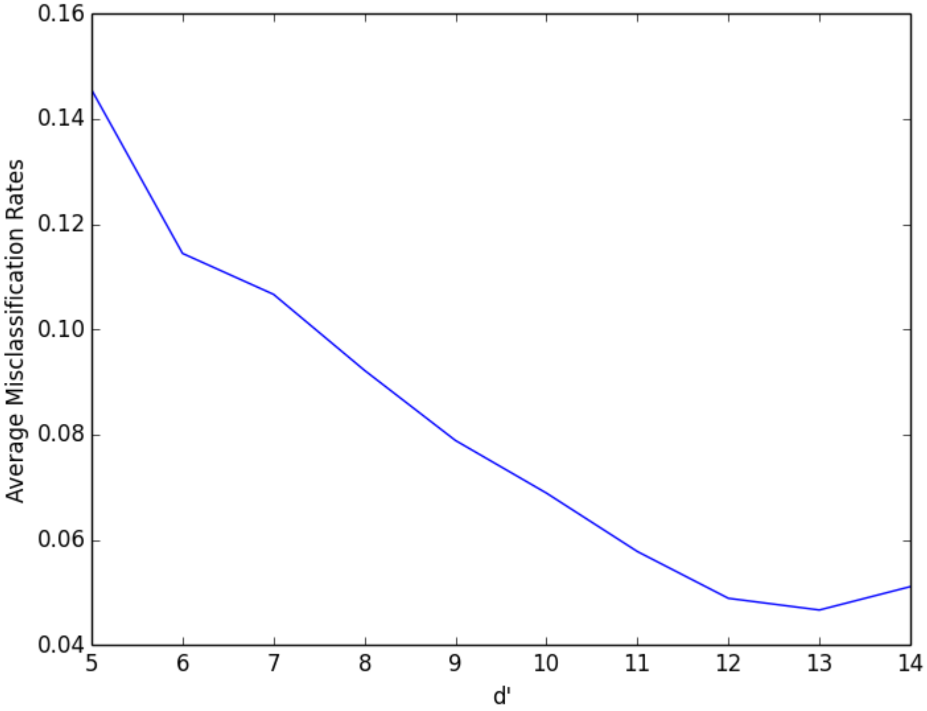
Eigen Faces N=7



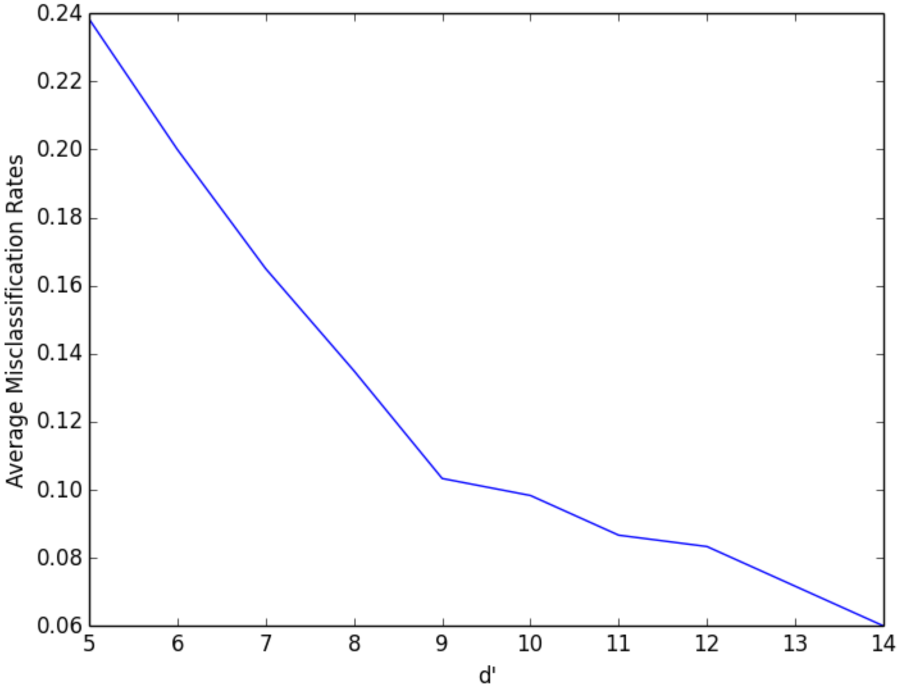
Fisher faces N=3



Fisher face N=5



Fisher faces N=7



From the experimental results, it can be concluded that the misclassification rate is higher for Eigenfaces algorithm compared to the Fisherfaces algorithm. Also, from the results, as the number of testing images increases the misclassification rate curve moves towards lesser values which means the misclassification rate is decreasing with increase in the N over 3,5,7.

The following files are included in the project submission along with this documentation:

- README.txt
- Project1.pdf (This file)
- 3 folders (data, libraries and main)
    - Data – Contains 165 training and test images in .pgm format (The Yale Face Database)
    - Libraries – Contains algorithms.py, utility.py and __init.py__ (Do not delete! Need for importing these files in the main python code)
    - Main – Contains eigenfaces.py and fisherfaces.py

IMPORTANT: Refer the README.txt for details about running the code.