Addis Ababa Science and Technology University

Collage of Electrical and Mechanical Engineering

Department of Electrical and Computer Engineering

Computer Engineering Stream

Computer Vision

Assignment 2: Face Recognition Algorithms

By

1. **Gebeyaw Tigabu** **GSR 210/12**
2. **Sileshi Nibret** **GSR 217/12**

Submitted to: Dr. Beakal Gizachew

February 25, 2021

Introduction

Face recognition is an easy task for humans. Computers recognize faces by extracting meaningful features from an image, putting them into a useful representation and performing some kind of classification on the. Face recognition based on the geometric features of a face is probably the most intuitive approach to face recognition. Face recognition is good for: -

- ✓ Using our face as our password
- ✓ Searching for criminals
- ✓ Adapting advertisements personally suitable
- ✓ Family album organization (Picasa)

Part 1: Eigenfaces

This algorithm follows the concept that all the parts of face are not equally important or useful for face recognition. When we look at a face, we look at the places of maximum variation so that we can recognize that person. For example, from nose to eyes there is a huge variation in everyone's face. Eigenfaces algorithm works at the same principle. It takes all training faces of all people at once and looks at them as a whole an then it keeps the most important components and disards the rest. These important components are known as principle components. PCA method is less optimal in the separation between classes.

Due to the useful features of faces this algorithm uses it's known as eigen faces. But this algorithm also considers illumination as an important factor. So as it picks up light illuminations it considers it as a feature representing a face which is not true.This algorithm doesn't pay attention to the features that differentiate one individual from another. It just concentrates on the features that represent all the faces of all the people.So, to overcome problems of eigenfaces, Fisherfaces was introduced which iss an improved version of eigenfaces algorithm.

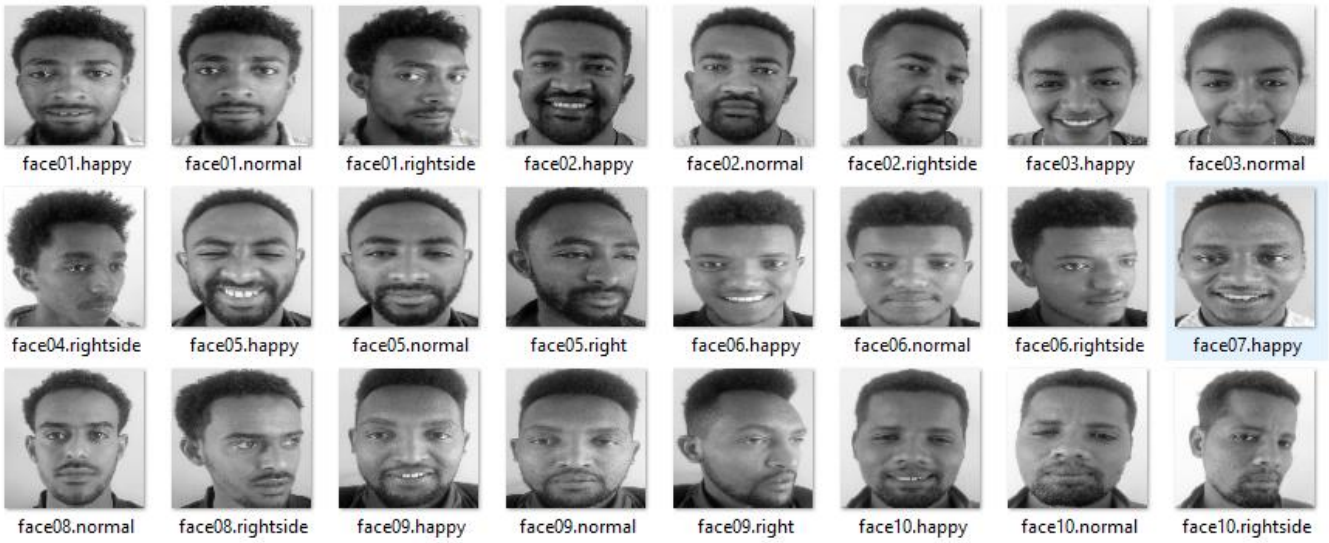
Methodology followed

- ✓ Collection of data is done in form of face images. Collections are done using photographs saved from a camera taken in the way that face is fully visible and facing forward.
- ✓ Our dataset has 10 persons each with 10 different facial expressions and a total of 100 images.
- ✓ The facial expressions are ***center light, normal, glass, left side, right side, sad, surprise, wink, happy, sleepy.***
- ✓ After that we divide our dataset into 2 groups such as training data set and testing data set with

the ratio of 70% for training and 30% for testing.

- ✓ Then we converted the face images into gray scale.
- ✓ Apply Image resampling to reduce the size of the images. In decimating or down sampling process Aliasing effect of signal processing is the main factor for image quality reduction.
- ✓ To overcome Aliasing effect, we were smoothing an image using ***Gaussian filtering*** algorithm.
- ✓ Eigenface and Fisherface method are applied to generate feature vector of facial image data and then to match vector of traits of training image with vector characteristic of test image using Euclidean distance.
- ✓ After the training is done, the next stage is image recognition process. The goal is to successfully recognize the test image.

- ✓ The following are a few test images that we preprocessed and downsampled to 100x100 resolution after Gaussian smothing .



Face Recognition methods used: Eigenfaces and Fisherfaces

Language used*:* Python 3.8 with Jupyter Notebook (Anaconda3)

Eigenfaces Algorithm

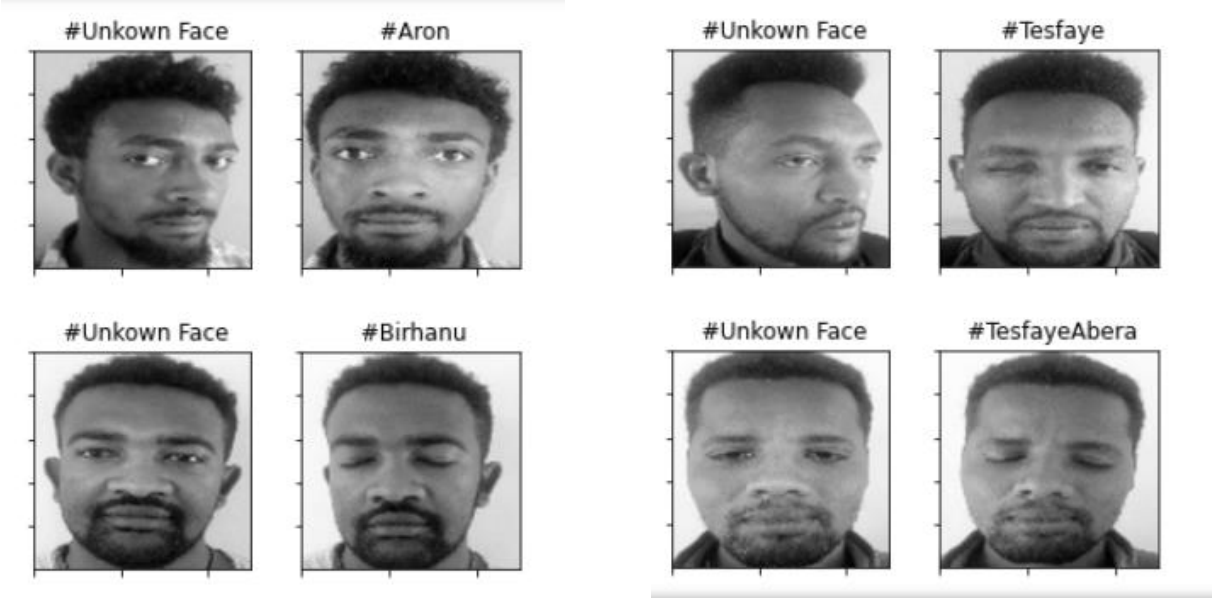
Eigenface follows the following algorithm to train and recognize faces

1. **Load training faces**
2. **Convert face images into column vectors** ←Put face image pixels to a column of matrix.

3. **Calculate the mean** ← Find mean value of column matrix to find "Mean Face".
4. **Subtract the mean from each image** ←subtract each image pixel value by mean pixel.
5. **Calculate Covariance matrix** ← $\boldsymbol{C = A.A^T}$ where A is a matrix from step4
6. **Calculate the eigenvectors and eigenvalues of the covariance matrix**
   - ✓ *Eigenvectors* of C is linear combination of image space with the eigenvectors of L
   - ✓ *Eigenvectors* represent the variation in the faces
7. **Select the principal components**
   - ✓ Higher the eigenvalue, characterize features of a face.
   - ✓ Lower eigenvalues can be ignored, as they explain only a small part of characteristic features.
8. **Visualize Eigenfaces**
   - ✓ Create the image, we need to scale it properly.
   - ✓ The smallest value in the vector should be converted to 0.
   - ✓ Largest value should be converted to 255.
   - ✓ Any value in between should be scaled within the min-max range.
9. **Reconstructing Faces.**
   - ✓ First we compute weight W[i] of the new input face for each EigenVector[i] (dot product of new face column and eigen-vector column) ←scalar W[i] per eigen-vector[i].
   - ✓ Because eigen-vectors are normalized, we simply multiply the W[i] to EigenVector[i] and added to mean face.
   - ✓ Note that we are adding mean face because we subtracted mean face at step4. In order to get the actual face, we need to add mean face back.
10. **Recognition** ← Euclidean distance where we used for our cases to calculate the distance between the face and its reconstruction.

Result from the experiment

While recognizing the face image using Eigenfaces, the training set contains 70 images of 10 persons and the testing set contains 30 images of 10 persons. We try to recognize all the test images and results looks like as follows.



Among 30 test images that we used in our experiment we get 28 correctly matching and 2 incorrectly matching.

Using different facial expressions, different head pose by applying gaussian filtering for resampling purpose we get the overall accuracy of **93.33%.** But when we tried to recognize without facial expressions and without different head pose, we get **100%**. We also check the effect of gaussian filtering for resampling. Without using gaussian filters, we get around **83%** accuracy.

From this experiment and results we conclude that eigenface method has limitations on different illumination, different head pose and different facial expressions and also gaussian filtering is better to get better accuracy.

Part 2: Fisherfaces

We know that eigenfaces considers illumination an important feature of a face but it actually isn't. Considering the illuminations as an important feature it may discard other people's features considering them less useful. We can fix this by tuning eigenfaces such that it extracts features of all individuals separately instead of looking at them as a whole. So, now even if one person's face data has high illumination changes, it will not affect other people's features.

Fisherfaces algorithm **extracts principle components that separates one individual from another**. So ,now an individual's features can't dominate another person's features. Image recognition using this

algorithm is based on reduction of face space domentions using PCA method and then applying LDA method also known as Fisher Linear Discriminant (FDL) method to obtain characteristic features of image.

LDA is used to find a linear combination of features that separates two or more classes or objects. It can be used for dimension reduction before further classification. It attempts to model the difference between classes of data .

This method doesn't capture illumination variations as obviously as Eigenfaces method.

- ✓ Data is assumed to be uniformly distributed in each class.
- ✓ Aim is to maximize the ratio of between-class scatter matrix and the within-class scatter matrix.
- ✓ It can produce good results even in varying illumination.

Algorithm of Fisher Faces

Face data used

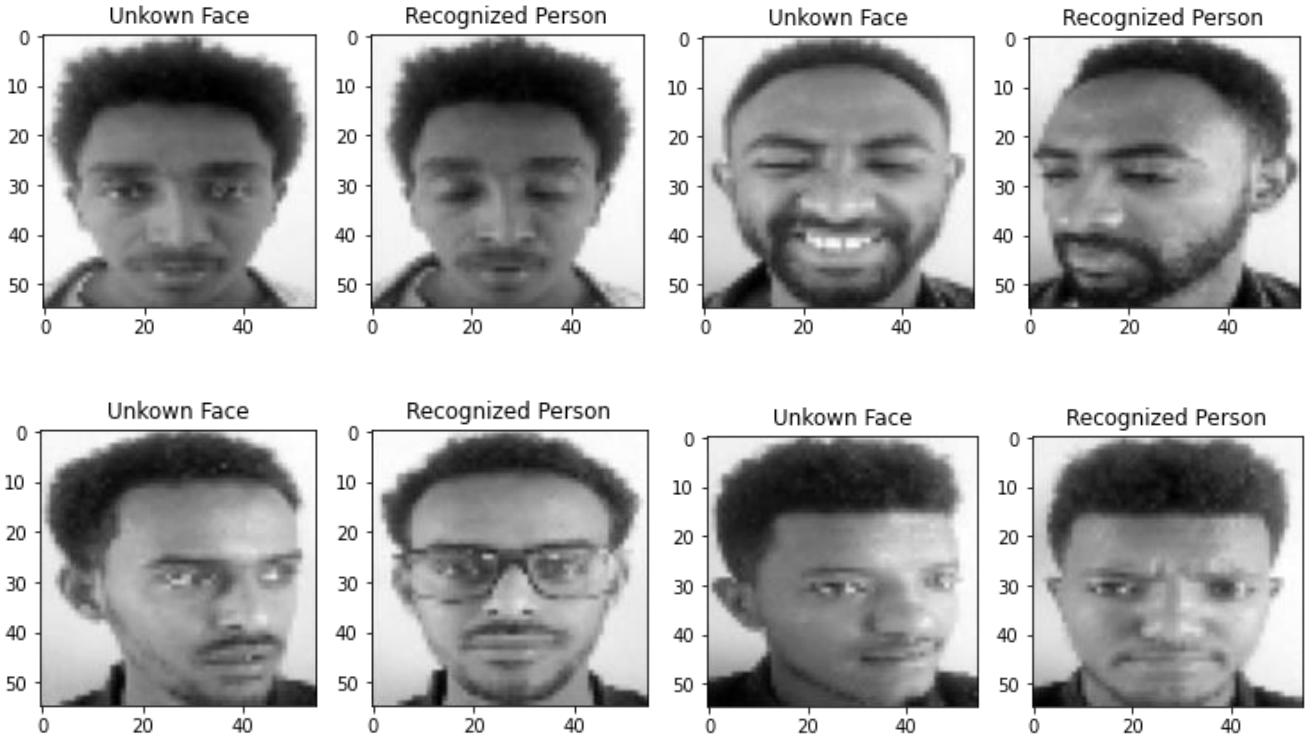
- ✓ Square images with width = height = N = 100 (in our case)
- ✓ 100 images in the database
- ✓ 10 persons in the database

1. Computing the average of all faces
2. Calculate mean vector of given training data of k-dimensions and also calculate the class-wise mean vector for the given training data.
3. Calculate SB and SW matrices needed to maximize the difference between means of given classes and minimize the variance of given classes.
4. After that calculate the matrix M, where M = (inv (SW)). (SB)
5. Calculate eigen values of M and get eigen vector pairs for first n (needed) dimensions.
   This eigen vectors give us the value of ‘w’ used to transform points to needed number of dimensions. If SW is singular (M<N^2)
   - ✓ Apply PCA first
   - ✓ Apply LDA
6. Project faces onto the LDA-space
7. To classify the face
   - ✓ Project on to LDA space

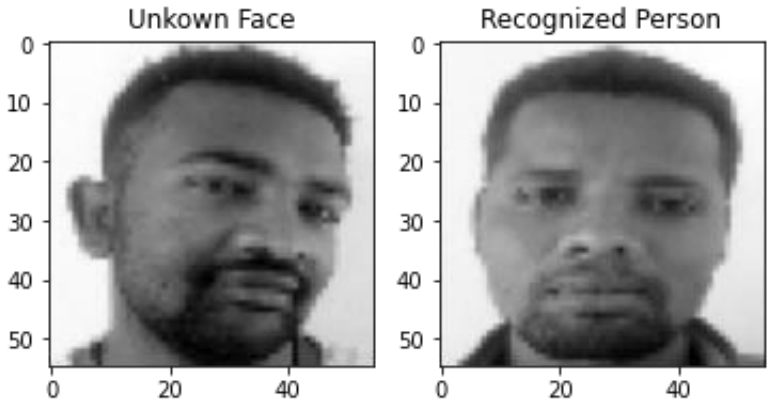
Results

In this method also we have used 70 images for training and 30 images for testing. And we also apply gaussian filtering for image resampling.

We tried to recognize all test images and we get 29 correctly matching and 1 incorrectly matching. And we get an *accuracy* of **96.67%.** But when we tried to test without gaussian filter, we get **85.53%** of recognition accuracy.



Incorrectly matching images



Conclusion

In doing face recognition experiment using eigenfaces and fisher faces we have seen that each algorithm has its own limitation and good side. For example, eigen faces have small accuracy when illuminated images are included in the training set. In computational time Eigenfaces take **8.27 seconds** for recognition of all test faces. However, Fisherfaces take **52.6 seconds** to complete the recognition process of test images. Since fisher faces gain better recognition accuracy using LDA the computational cost is much higher than that of the eigenfaces.

Generally, from this experiment we found that the Fisherfaces method is much better for facial recognition than the Eigenfaces method since the accuracy of Eigenfaces is **93.3 %** and **96.67 %** for Fisherfaces using same dataset. In addition to that the Fisherface method generally performs better even in the change in facial expression and even if the person wears glasses and sleepy. Applying Gaussian filter for image resampling improves accuracy of recognition even in eigenface method. In the case when we have a greater number of principal components, we can get better results but the computation cost is high.