INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

Face Recognition using Eigen Faces

Team 9: Data crew

Sr. No.	Name	Roll No.
1.	Bhoomeendra Singh Sisodiya	2021701037
2.	Pavan Baswani	2021701035
3.	Prateek Jaiswal	2021701009
4.	Vrund Shah	2021701025

INDEX

GitHub repo : https://github.com/bhoomeendra/Face-Recognition-Using-Eigenfaces/tree/master

Problem Statement:

Face detection is used to find and identify human faces in digital images. Face detection technology can be applied to various fields including biometrics, law enforcements, entertainment and personal safety to provide surveillance and tracking of people in real time.

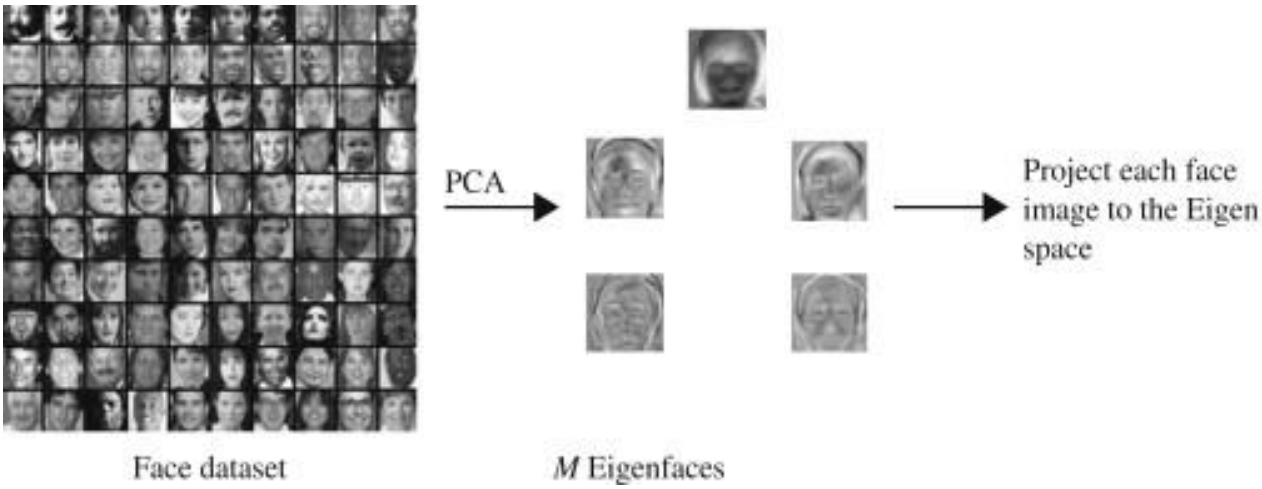
Face detection has progressed from rudimentary computer vision techniques to advances in machine learning to increasingly sophisticated artificial neural networks and related technologies the result has been continuous performance improvements. It now plays an important role as the first step in many key applications including face tracking, face analysis and facial recognition.

Figure: *How face detection of a face works by generating a face mask of specific points present in a human face such as eyes, ears, nose and fore-head etc.*

Goals and approach

This approach is about the detection and identification of human faces. It uses the fact that faces are normally upright and thus may be described by a small set of 2D characteristic views. Face images are projected onto a feature space that best encodes the variation among known face images.

The face space is defined by the “Eigen-Faces”, which are the eigenvectors of the set of faces; they do not necessarily correspond to isolated features such as eyes, ears and noses. The framework provides the ability to learn to recognize new faces in an unsupervised manner.



Here, is an example of how algorithms generally detect a face from a large dataset of images in such a way it can correctly predict the original image.

Eigenfaces for recognition

We want principal components of the distribution of faces or the eigen vectors of the covariance matrix of the set of face images. These eigenvectors can be thought of as a set of features which together characterize the variation between face images. Each image location contributes more or less to each eigenvector, so that we can display the eigenvector as a sort of ghostly face which we call an eigenface. The number of possible eigenfaces is equal to the number of face images in the training set. However the face can also be approximated using only the best eigenfaces those that have the largest eigenvalues and which therefore account for the most variance within the set of face images.

Implementation

The project divided into the following subtasks

1. **Exploring Faces Dataset :**

   The dataset we are using consist of 13000 images of 64x64 faces of people in pgm (Portable Grayscale Image) format. We will be using a subset of this images as we don't need that many eigen vectors to identify faces.

2. **PCA :**

   We will implement PCA from scratch to identify the eigen vectors of $A.A^T$ where $A_{m \times n}$ (M number of images ,N is the size of flatten image) in decreasing order of eigen values and then selecting top k eigen vectors to form face-space with those k eigen vectors as basis and project the known face to this new face space.

3. **Face Detection :**

   After projecting all the faces to the face space we will take the average of all the projection of the faces and will have an average face in face-space now when we encounter a new image we will calculate its Euclidean distance from the average face and if it is below a certain threshold then we will consider it as a face else it is not a face.

4. **Face Recognition :**

   When we detect a face then we will perform Nearest Neighbour Classification to the identity the person. If it is a match (Euclidean distance smaller then a threshold) then we will show the image of person it matched to else we will label is as a Unkown face.

5. **Automatic Clustering :**

   When we encounter Unkown face we will label it and store it in our know face list and when we see that face again we will be able to recognize it

6. **Multiple Face Detection in Image :**

   We can detect Multiple Faces by performing Face Detection on local sub images for every (x,y) in the given new image.

7. **Fischer faces implementation**

8. **Comparison of Eigenfaces and Fischer faces**

Report

1. Abstract
2. Introduction
3. Literature survey
4. Face Recognition
    a. Eigen faces (methodology)
    b. Fischer faces (methodology)
5. Experimental results
6. Comparisions (Eigenfaces vs Fischerfaces) and Error analysis
7. Exploration on proposed methodology
8. Future scope
9. References

Work Distribution

Sr. No	Name	Work
1.	Bhoomendra	Exploring the dataset and implementing PCA, face detection and recognition and multiple face detection. Write Future scope section of the report.
2.	Pavan	Do Literature survey and write Abstract, Introduction and Literature Survey sections of Report. Implement code for automatic clustering.
3.	Prateek	Understand the Fisherfaces approach to Face Recognition and write it in Report and also code it
4.	Vrund	Understand the Eigenfaces approach to Face Recognition and write it in Report. Exploration on proposed methodology. Write the comparison section of report

Note: Proof reading of report and analysis will be done by all members.

Milestones and Timeline

Timeline	Milestones
26 October 2021, 5:58 PM	Project allocation
07-November-2021 11:55PM	Project proposal submission
Week-1	1. Paper understanding and relevant reading 2. methodology implementation (recognition using eigenfaces, automatic clustering of new faces) 3. Writing Introduction, Literature review and methodology of problem statement in manuscript/report.
Week-2	1. Implementation of multiple face identification in image 2. Code alignment in GitHub with readme. 3. Tabulating Eigenfaces results and proof reading of report
16-20 November	Mid Evaluation
Week-3	1. Recognition using Fischer faces code implementation 2. Comparisons of Eigen faces vs Fischer faces results and tabulating in report. 3. Performing error analysis on the results obtained for both Eigen faces and Fischer faces.
Week-4	Finishing the report with proof reading and publishing the GitHub repo with readme file.
1-4 December	Final Evaluation

Deliverables

Mid Evaluation

1. Dataset description
2. GitHub repo of code implementation
    a. Implement PCA
    b. Perform Face Detection and Recognition
3. Report pdf
    a. Introduction and Literature review
    b. Methodologies used
    c. Objectives achieved
    d. Analysis/Findings from results
    e. Future improvements

Final Evaluation

1. GitHub repo of code implementation with dataset
    a. Multiple face detection
    b. Automatic Clustering
    c. Implement Fisher faces
2. Report pdf
    a. Problem statement (abstract)
    b. Introduction and Literature review
    c. Methodologies
    d. Results and discussions (Error analysis)
    e. Comparisions of different approaches
    f. Future scope
    g. References and Acknowledgements