Addis Ababa University
Addis Ababa Institute of Technology
School of Information Technology and Engineering

Face recognition report

Group Members	**ID**
Bethelhem Yemane	UGR/1894/13
Gelila Tefera	UGR/9453/13
Leul Wujira	UGR/8834/13
Milka Fasika	UGR/7126/13
Naol Taye	UGR/1048/13
Yohannes Ahunm	UGR/4045/13

Introduction

In recent years, face recognition technology has significantly advanced, making real-life applications of face recognition systems a reality. This report aims to compare two techniques in face recognition: Eigenfaces and Fisherfaces. Through the implementation of these methods, we explore the respective benefits and limitations of each approach.

Dataset

We used the Extended Yale B image dataset, which had images of 39 subjects, with each subject having 64 photographs captured under varying lighting conditions. For our experiment, we specifically chose the initial 10 subjects, each with 10 images. Additionally, we set aside 20 images per subject for the test set. Finally, we included 63 non-facial images and 66 images from other subjects in the Yale dataset, intended for use as unknown faces. The exact dataset we used can be found here.

We resized the original images from the dataset, which were initially 168x192 pixels, to a dimension of 168x168 pixels during the loading process.

Eigenfaces Approach

Eigenfaces is a face recognition technique based on Principal Component Analysis (PCA) that represents face images as a set of eigenvectors. By constructing a covariance matrix from set of face images and decomposing it, eigenfaces capture the most significant facial features across the dataset. During recognition, a new face image is projected onto the eigenface subspace to obtain weights, which are used to reconstruct the face and find the closest match among the stored eigenfaces.

Steps

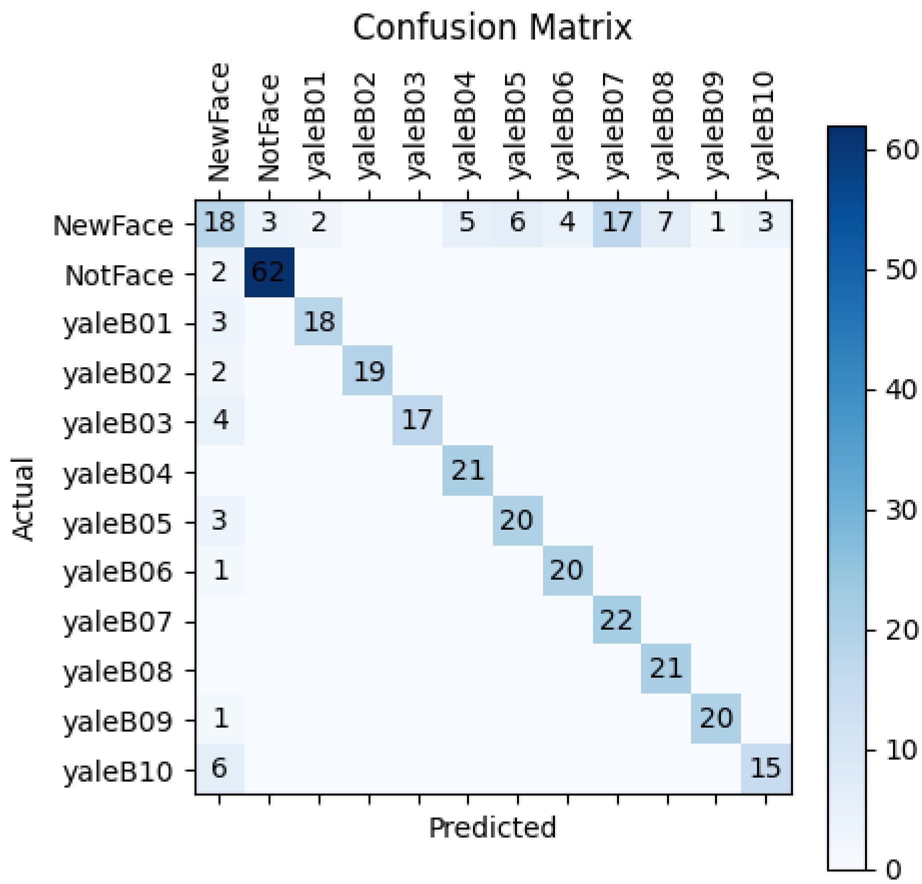
1. **Preprocessing** – all input images are grey-scaled and each image has been resized to 168x168 pixels, then flattened into a vector of dimension 28,224. The image vectors form a matrix (**A**) of face image data. The matrix at this stage was 28,224x100 since we had 100 images (10 subjects each having 10 images).
2. **Compute mean face** – computed the average of all flattened vectors
3. **Subtract mean face** – to remove common feature from all faces
4. **Compute covariance matrix** – computing the covariance matrix as it is was not computationally efficient (the resulting matrix would be 28,224x28,224 over 5.9 GB) thus, we computed $\mathbf{A}^T\mathbf{A}$**,** let's call it **L.**
5. **Compute Eigenvalues and Eigenvectors** – we applied PCA to the computed matrix, **L**, obtaining eigenvalues and eigenvectors. Since we had small dataset, we selected all from 10 to 100 top eigenvectors as eigenfaces. For larger dataset, we would have needed to limit the number of eigenfaces to the top eigenvectors with the most significant eigenvalues.
6. **Face representation** – we projected the original 100 images onto the eigenface space to obtain the weights representing contribution of each eigenfaces to the original face.

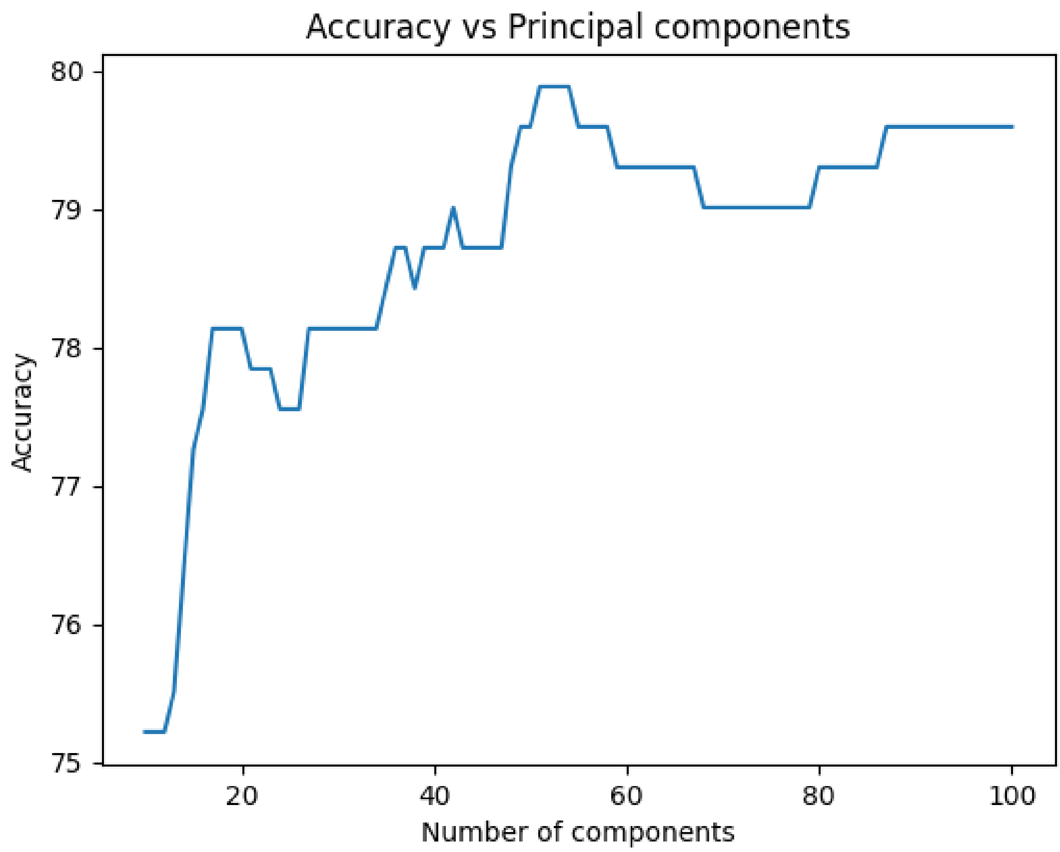
7. **Threshold** – we computed two thresholds, a **reconstruction threshold** which we later use to decide whether an image is face or not. We also computed **distance threshold** which is we used to decide, whether a face image belongs to one of the 10 subjects.
8. **Recognition** – subtract mean face from input image, followed by projecting it into the eigenface space gives us weights of input face. We compute the distance between this weight and all projected known faces in the dataset to determine closest match. We also reconstruct input image from its weight. If the distance between reconstructed and input image is greater than reconstruction threshold, we consider the image as **NOT FACE.** Otherwise, if minimum distance is less than distance threshold then it means we have identified the image. If not, it means the face is not in our known dataset.

Results

By following the above steps, we were able to achieve **79%** accuracy on the dataset. The eigenface method faced difficulty in identifying faces with varying light conditions. The following shows confusion matrix and accuracy.

The eigenface method achieved **90%** accuracy on test dataset that doesn't include new faces and other object images.





Peak accuracy of 79.8% was achieved when using around 51 components to represent the images.

Fisherfaces Approach

The Fisherfaces method, also known as Linear Discriminant Analysis (LDA), is an extension of the Eigenfaces approach that emphasizes the discriminative power of the extracted features. Unlike Eigenfaces, which focuses on maximizing the variance in the dataset, Fisherfaces aims to maximize the ratio of between-class scatter to within-class scatter. This means that Fisherfaces not only captures the variations within each class of faces but also enhances the separability between different classes, leading to improved discrimination between individuals.

Steps

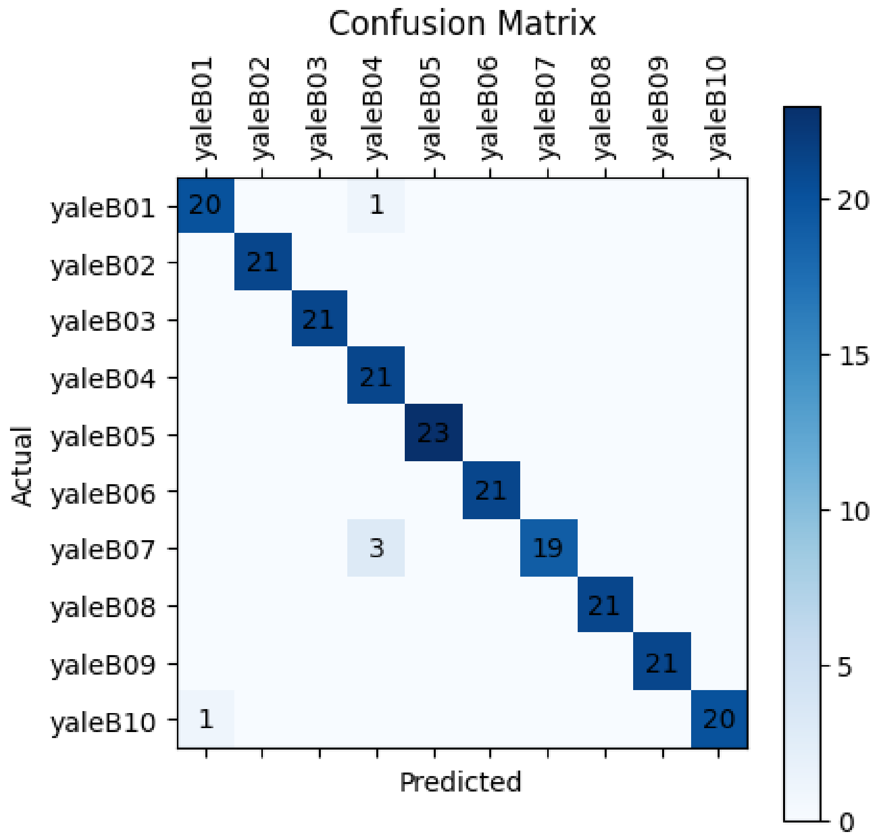
1. Preprocessing – all input images are grey-scaled and each image has been resized to 168x168 pixels, then flattened into a vector of dimension 28,224. The image vectors form a matrix (**A**) of face image data. The matrix at this stage was 28,224x100 since we had 100 images (10 subjects each having 10 images).
2. Compute mean face – compute the average of all flattened vectors
3. Compute mean face for each class
4. Compute the within-class scatter matrix and between-class scatter matrix
5. Compute the Fisherfaces by finding the optimal linear transformation that maximizes the ratio of between-class scatter to within-class scatter.

6. Project the face images onto the Fisherface subspace to obtain the discriminant features for classification or recognition.
7. We used KNN Classifier to decide the class of image.

Results

By following the above steps, we were able to achieve **97%** accuracy on the dataset. Compared to the eigenface method the fisherface can't identify images that are don't belong to training classes. Thus, the test doesn't include unknown faces and unknown objects.

The following show the confusion matrix.



Conclusion

In conclusion, this report has explored two popular techniques in face recognition: Eigenfaces and Fisherfaces. Through a detailed examination of these methods, we have gained insights into their underlying principles, strengths, and limitations.

Eigenfaces, based on principal component analysis (PCA), offer a powerful approach for face representation and recognition. By capturing the main variations in a face dataset, Eigenfaces can effectively reduce the dimensionality of face images and enable efficient recognition. However, Eigenfaces are sensitive to variations in lighting conditions and facial expressions, which can limit their performance in real-world scenarios.

On the other hand, Fisherfaces, aim to maximize the between-class scatter while minimizing the within-class scatter. This approach enhances the discriminative power of face recognition systems and can handle variations caused by lighting and expressions to some extent. Fisherfaces, however, require a sufficient number of training samples per class and may suffer from overfitting when faced with limited data.

References

- Belhumeur, Peter N., et al. "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection." IEEE Transactions on Pattern Analysis and Machine Intelligence (1997).
- Turk, Matthew, and Alex Pentland. "Face recognition using eigenfaces." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Vol. 3. 1991.