Project 5: Face Recognition Using Principal Component Analysis and Eigenfaces in MATLAB

Matteo De Angelis
College of Engineering
University of Miami
Miami, USA
mld163@miami.edu

Carter Falkenberg
College of Arts and Sciences
University of Miami
Miami, USA
ckf25@miami.edu

Kelly Rojas
College of Engineering
University of Miami
Miami, USA
knr20@miami.edu

***Abstract* — This project implements a face recognition system using the eigenfaces method based on Principal Component Analysis (PCA). The system is trained to detect and recognize facial images by projecting them into a reduced eigenspace and comparing them against known individuals. Two testing sets are used to evaluate the system: one containing known faces and another with unknown or non-face images. Key performance metrics include recognition accuracy, false positives, and the effect of varying the number of eigenfaces. The results demonstrate the effectiveness of PCA in dimensionality reduction and the trade-offs between recognition performance and computational efficiency. The project is implemented in MATLAB using grayscale face images, emphasizing theoretical understanding and practical application of eigenface-based recognition.**

Keywords* — *Face Recognition, Principal Component Analysis (PCA), Eigenfaces, MATLAB, Image Classification, Dimensionality Reduction

I. Introduction

Face recognition is critical in computer vision, with widespread applications in human-computer interaction, biometrics, and even security. Among the various techniques developed for face recognition, the eigenfaces method is highlighted due to its simplicity and effect8iveness in dimensionality reduction and feature extraction[1]. This project focuses on implementing the eigenfaces approach using PCA to develop a recognition system in MATLAB.

The system is trained on a dataset of facial images to compute the eigenfaces from PCA, and then tested on two sets of images: T1 (known faces), and T2 (unknown or non-faces). The primary objectives are to classify whether a test image is a face, and if it is a face, to identify it as a known individual from the training set. Only a subset of the most significant eigenfaces (those with the highest eigenvalues) is used, allowing analysis of recognition performance versus the number of retained components. Furthermore, the project investigates the system's accuracy, failure cases, and sensitivity to the number of principal components. A simplified explanation of this approach is commonly found in educational resources that explain PCA's role in facial classification[3].

II. Theory

The eigenfaces methos is based on PCA, a statistical technique that transforms high-dimensional data into a lower-dimensional spaces while preserving the most important variance[2]. In the context of face recognition, each grayscale image is considered a high-dimensional vector, and PCA helps find a new set of orthogonal basis vectors (called eigenfaces) that best describe the dataset[1],[4].

First, the training images are mean-centered and then arranged into a matrix where each column represents a face vector. PCA is applied to this matrix to compute eigenvectors (eigenfaces) and their corresponding eigenvalues. The top eigenfaces (those with the largest eigenvalues) are retained, capturing the key variations among facial features[4].

A test image is projected into the reduced eigenspace to obtain a feature vector (its PCA coefficients). Recognition is then performed by comparing this vector to the feature vectors of training images using a distance metric like Euclidean distance. If the minimum distance is below a pre-defined threshold, the test image is classified as a known face. Otherwise, it is rejected as either an unknown or a non-face image. Additionally, PCA reduces noise and redundancy in facial data, and its comparison with other dimensionality reduction techniques further demonstrate its usefulness in features extraction[2],[3].

III. Methodology

A. Dataset Preparation

The dataset used for training and testing comprised of:

- 32 face images used for training
- 44 face images used for testing
- 7 non-face images used for testing

From each collection of face and non-face images that can be used for testing, only half of them respectively are chosen at random and used for testing. This entails that 22 face images and 4 non-face images are present in the testing dataset.

B. Vectorization

All images are sized 640x480 pixels and are all vectorized to a 1D array. To try and achieve the best results the images used before vectorization are taken both in RGB color format and converted to a greyscale format. This leads with the following vector sizes depending if greyscale or color images are used:

- Color images: $640 \times 480 \times 3 = 921600$
- Greyscale images: $640 \times 480 = 307200$

This is done consistently in two different processes.

C. Mean Faces

To center the data before applying PCA, the mean face normalization vector is computed by averaging all training image vectors:

$$m = \frac{1}{N}\sum_{i=1}^{N} x_i$$

The training and testing datasets are then normalized by subtracting the mean:

$$X_{train,centered} = X_{train} - m$$

$$X_{test,centered} = X_{test} - m$$

D. Principal Component Analysis

PCA is applied to center the data and to obtain the eigenvectors which are used to form the basis of the eigenface space. In order to retrieve only the most important eigenfaces to be used for testing, an explained cumulative variance of at least 95% is retained. This is done by choosing the first n values of K that satisfy this formula:

$$\frac{\sum_{i=1}^{n} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \geq 0.95$$

where, n is the minimum number of K necessary to retain at least 95% of the information, and N is the total number of K found using PCA.

The final eigenfaces found to represent the eigenface space are represented by:

$$U = [u_1, u_2, \dots, u_k]$$

E. Face Recognition

Both the training and testing images are projected onto the eigenface space to obtain lower-dimensional feature representations.

$$A_{train} = U^T X_{train,centered}$$

$$A_{test} = U^T X_{test,centered}$$

To determine whether an image is a face, the reconstruction error is calculated by projecting the image back to the original space:

$$X_{new} = U A_{test} + m$$

The error is defined as:

$$error = \|X_{test} - X_{new}\|_2$$

If the error exceeds a predefined threshold of 15000, the image is classified as "NON-FACE" image, if it is lower than that it is classified as "FACE" image. 12000 was chosen to balance classifications with misclassifications for the "NON-FACE" and "FACE" classes.

For recognized "FACE" images, the Euclidean distance between the projected test vector and all projected training vectors is calculated:

$$distance = \|A_{test} - A_{train}\|_2$$

The closest match is determined by finding the minimum distance, and the corresponding training image is considered the best match.

IV. RESULTS

A. K=21

When utilizing 95% of cumulative variance, K=21 is chosen. Results for all test images on K=21 can be found in the appendix, each with their errors and distances to predicted matches, if applicable. The overall results for classification are shown below, in figure 1:

Test Set

OUTPUT \ TARGET	FACE	NOT FACE	SUM
FACE	18 69.23%	0 0.00%	18 100.00% 0.00%
NOT FACE	4 15.38%	4 15.38%	8 50.00% 50.00%
SUM	22 81.82% 18.18%	4 100.00% 0.00%	22 / 26 84.62% 15.38%

Figure 1: Confusion matrix for K=21 on test set.

Overall accuracy on the test set was 84.62%, with a misclassification rate of 15.38%. Precision was very strong, being 1.00, with a recall of 0.82, and overall F1-score of 0.90. These results indicate that K=21 yields strong classification performance. Specifically, when the model predicts that an image is a face, it is extremely accurate in its prediction. However, the model is much less accurate when predicting that an image is not a face, doing no better than chance. This is hard to correct for, since the images that are truly faces but predicted to not have large errors, some even larger than true non face images.

Depending on the use case for facial recognition, different threshold values can be chosen. Certain higher risk tasks, like security cameras for burglary detection, may prefer for a prediction of a face to be wrong than for a prediction of a non-face to be wrong. Other less vital tasks may prefer to only be notified of a face if the model is extremely sure it is actually a face.

Matching was more of a difficult task for the model. Of the images that were predicted to be a face, the match was correct 8 times out of 18. Distance values for correct matches ranged from 11,256 to 22,025. Incorrect matches distance values ranged from 12,107 to 22,556. There was no notable difference in the distance metric between true matches and false matches.

Figure 2 shows an example of why classification of non faces a difficult task is. The image on the left, a mountain, has an error of 12,151. Meanwhile, the image on the right, clearly a face that is taken in a similar fashion to those in the training set, has a much larger error of 20,926.

Figure 2: Difference in errors of a non-face image and a face image.

B. K=10

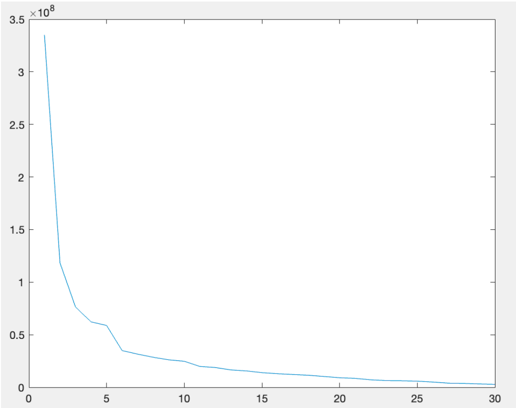


Figure 3: Plotted Eigenvalue size vs index

K=10 was chosen from the plot of the eigenvalues, as it is the K for which the gradient begins to approach 0. The results for classification are shown below, in Figure 4:

Test Set			
TARGET / OUTPUT	FACE	NOT FACE	SUM
FACE	10 38.46%	0 0.00%	10 100.00% 0.00%
NOT FACE	12 46.15%	4 15.38%	16 25.00% 75.00%
SUM	22 45.45% 54.55%	4 100.00% 0.00%	14 / 26 53.85% 46.15%

Figure 4: Confusion matrix for K=10 on test set.

Overall accuracy on the test set was 53.85%, with a misclassification rate of 46.15%. Precision remained very strong, being 1.00, with a recall of 0.45, and overall F1-score of 0.63. Overall accuracy dropped considerably compared to K=21, though precision remained the same. Recall and F1 scores also dropped by a large margin compared to K=21. This is likely because not enough of the variance of the dataset is represented by K=21, thus the model is underfitting and unable to detect faces well. Of the 10 faces predicted, 6 were correctly matched. This yields a slightly higher accuracy compared to K=21, but this is simply due to higher misclassification of the other faces. There are still fewer accurate matches than with K=21.

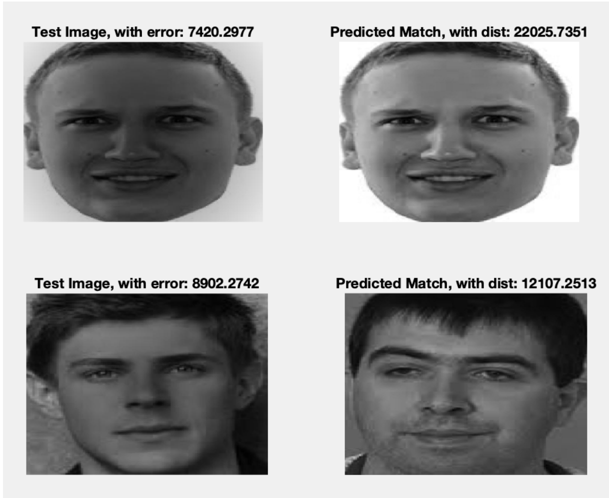


Figure 3: High distance metric for a true match compared to a low distance metric for a false match.

C. K=30

K=30 shows the results of not choosing a strict subset of eigenvalues but instead choosing all of them. Figure 5 shows the confusion matrix for K=30.

Test Set			
TARGET / OUTPUT	FACE	NOT FACE	SUM
FACE	19 73.08%	1 3.85%	20 95.00% 5.00%
NOT FACE	3 11.54%	3 11.54%	6 50.00% 50.00%
SUM	22 86.36% 13.64%	4 75.00% 25.00%	22 / 26 84.62% 15.38%

Figure 5: Confusion matrix for K=30.

Overall accuracy on the test set was 84.62%, with a misclassification rate of 15.38%. These two metrics are identical to K=21. Precision was 0.95, with a recall of 0.86, and overall F1-score of 0.90. These results indicate that K=30 still yield strong classification performance, even though they do not select a subset of eigenvalues. Similarly to K=21, when the model predicts that an image is a face, it is extremely accurate in its prediction. In this case, there is one example of an image that the model predicted was a face that was not truly a face. This difference compared to K=21 could be a result of overfitting the training set. The extra 9 eigenvectors explain considerably less variance than the initial 21, which may cause the model to learn certain noise in the training set that does not relate to true facial features.

V. CONCLUSION

The experimental results demonstrate that the eigenfaces method, when implemented using PCA, is effective for face recognition, reducing the dimensionality of facial image data while retaining critical identifying features. The model achieved the highest classification performance when using K = 21 principal components, corresponding to approximately 95% of the explained cumulative variance. This configuration yielded an overall test accuracy of 84.62%, a recall of 0.82, and an F1-score of 0.90, confirming that an optimal balance of information retention and noise exclusion can be attained by selecting an appropriate number of eigenfaces.

Furthermore, an analysis of different K values reveals critical trade-offs between underfitting and overfitting. When K = 10, the model retained insufficient variance from the training data, resulting in underfitting. This condition is reflected in a significant drop in recall (0.45) and overall accuracy (53.85%), showing the model's inability to correctly classify face images that deviate slightly from the training distribution. Conversely, when K = 30, all eigenfaces were retained, including those associated with low variance. Although this configuration produced classification metrics identical to those for K = 21, it introduced increased sensitivity to noise, potentially leading to overfitting and misclassification of some non-face images.

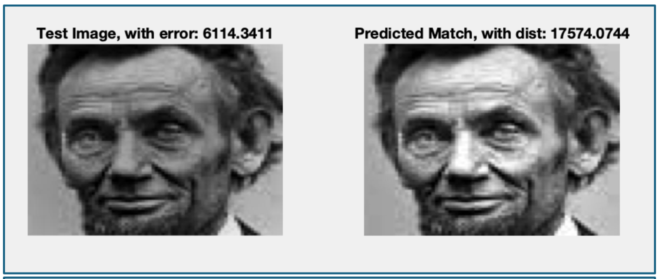
Misclassifications were often observed in the non-face category, such as portraits or realistic synthetic faces, which were mistakenly classified as faces due to low reconstruction error. Additionally, a number of true face images were incorrectly rejected due to high reconstruction errors. This indicates that while the reconstruction error thresholding method can be simple and effective, it can also lack robustness in edge cases where facial structure is ambiguous or atypical. The selection of thresholds is therefore critical, and should ideally be adapted based on application-specific tolerances for false positives and false negatives.

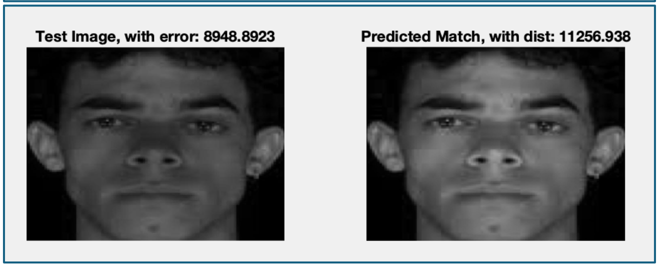
Recognition of individual identities was a more challenging tasks. Among the 18 test images classified as faces using K = 21, only 8 were matched to their corresponding identities from the training set using the Euclidean distance metric. There was considerable overlap in the distance values between correct and incorrect matches, meaning Euclidean distance may not be an ideal metric for identity verification in the reduced PCA space. Alternative classification approaches, such as nearest-neighbor classifiers with more discriminative distance metrics or supervised learning methods, may provide improved performance for identity recognition.

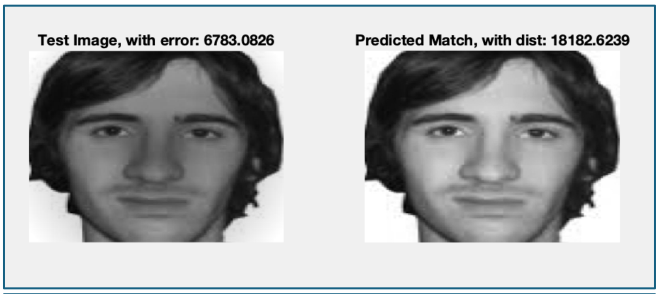
In conclusion, the eigenfaces method provides a computationally efficient and interpretable framework for face detection and recognition. Its performance is dependent on the number of principal components retained, and the chosen classification thresholds. While PCA offers strong generalization for face or non-face classification, improvements are needed for robust identity recognition, specifically in real-world conditions involving different facial structures, noise, and even lighting variations.

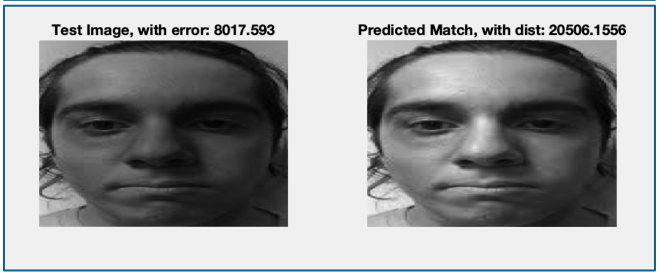
VI. APPENDIX

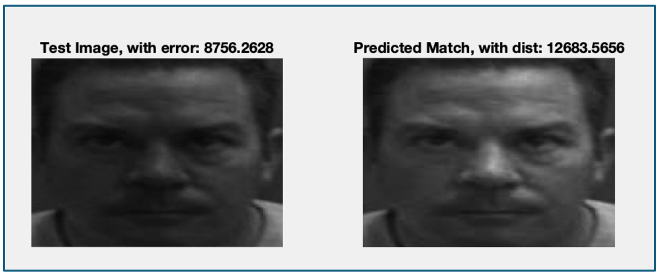
A. K = 21

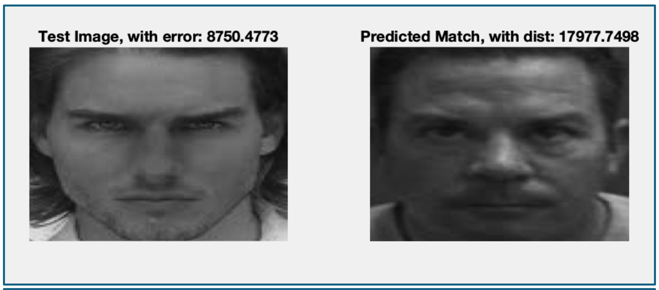




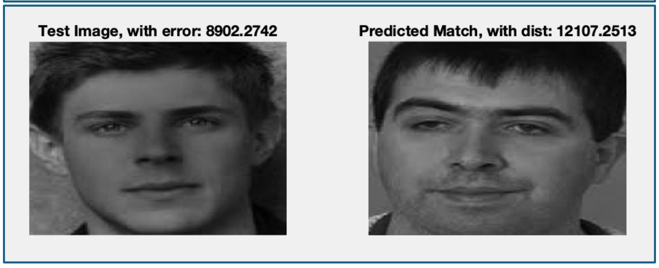


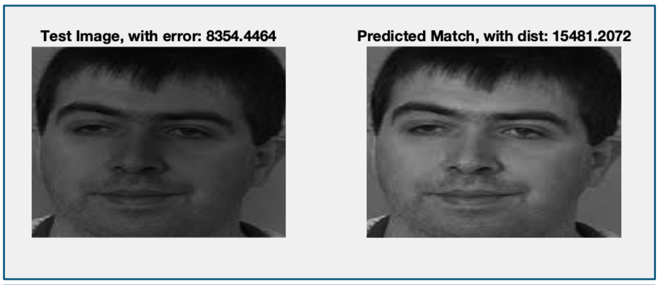


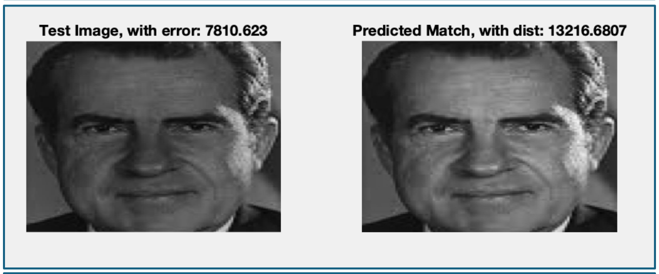


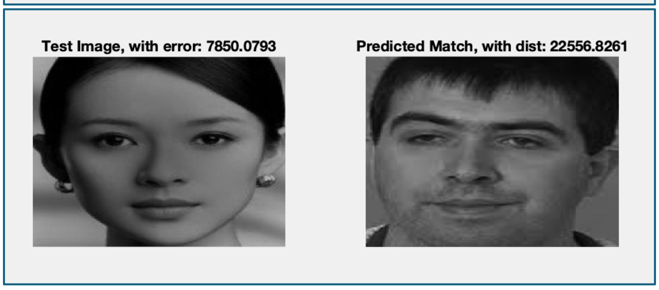

Test Image, with error: 8750.4773
Predicted Match, with dist: 17977.7498

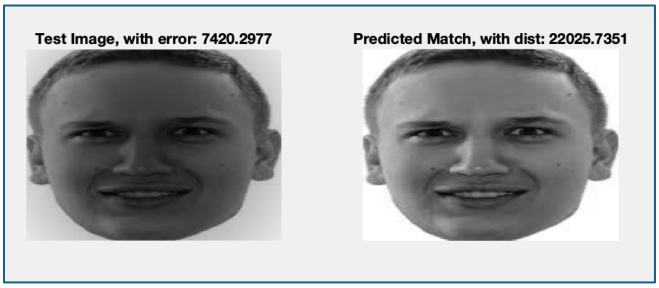

Test Image, with error: 6971.1166
Predicted Match, with dist: 20496.1486

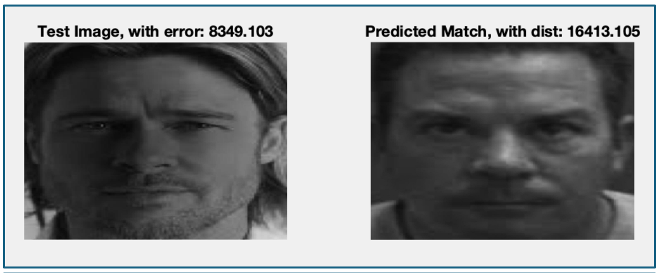

Test Image, with error: 8902.2742
Predicted Match, with dist: 12107.2513

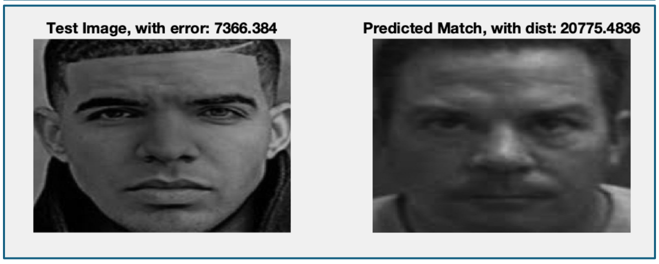

Test Image, with error: 8354.4464
Predicted Match, with dist: 15481.2072

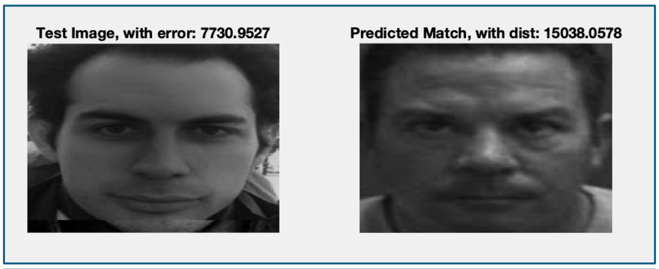

Test Image, with error: 7810.623
Predicted Match, with dist: 13216.6807

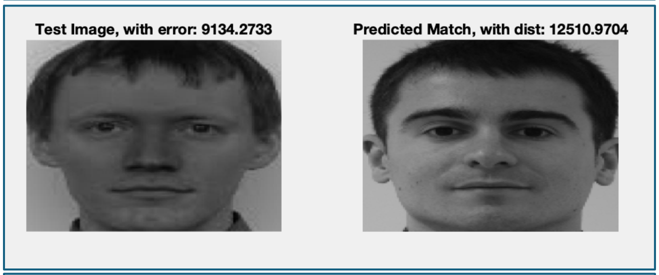

Test Image, with error: 7850.0793
Predicted Match, with dist: 22556.8261

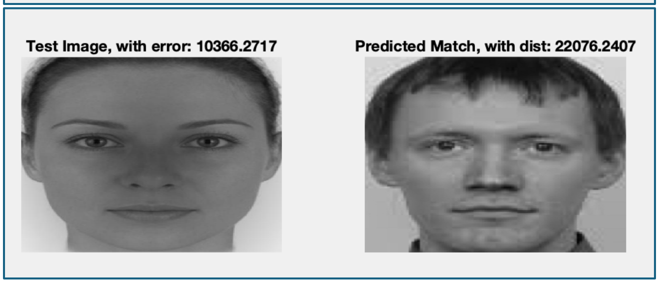

Test Image, with error: 7420.2977
Predicted Match, with dist: 22025.7351


Test Image, with error: 8349.103
Predicted Match, with dist: 16413.105


Test Image, with error: 7366.384
Predicted Match, with dist: 20775.4836


Test Image, with error: 7730.9527
Predicted Match, with dist: 15038.0578


Test Image, with error: 9134.2733
Predicted Match, with dist: 12510.9704


Test Image, with error: 10366.2717
Predicted Match, with dist: 22076.2407

Not face, error: 12250.1811

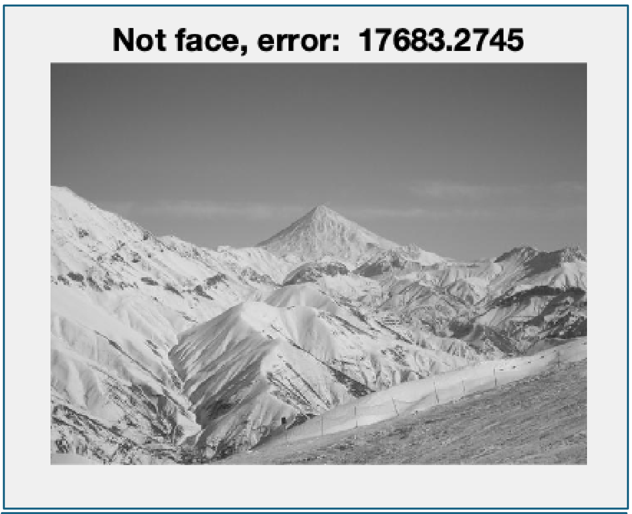
Not face, error: 17187.7444

Not face, error: 20926.0304

Not face, error: 19912.8255

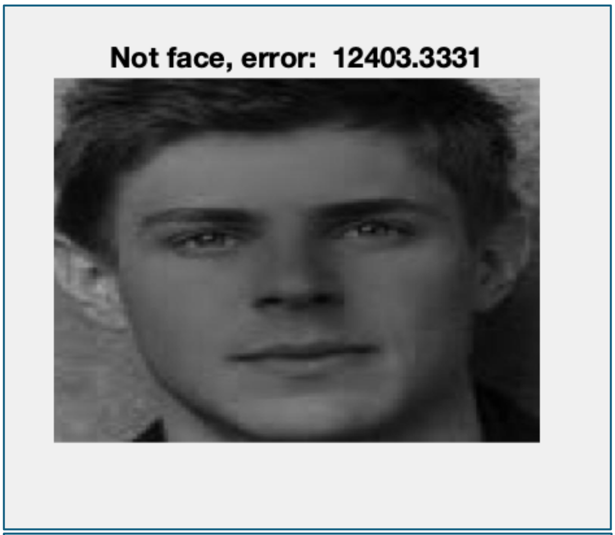
Not face, error: 12151.6811

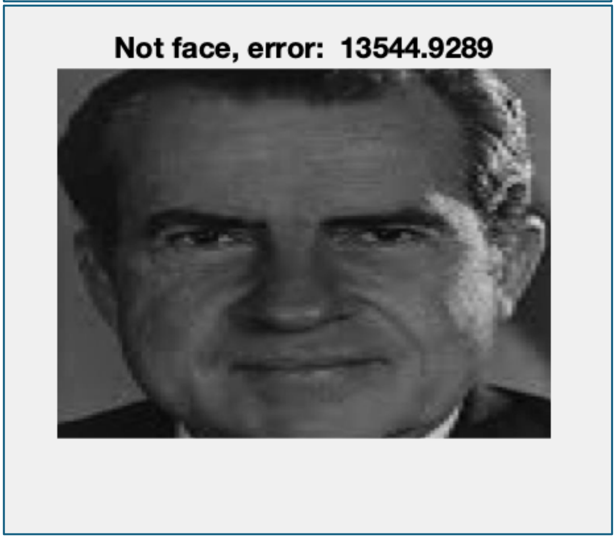
Not face, error: 19327.0093



B. K = 10

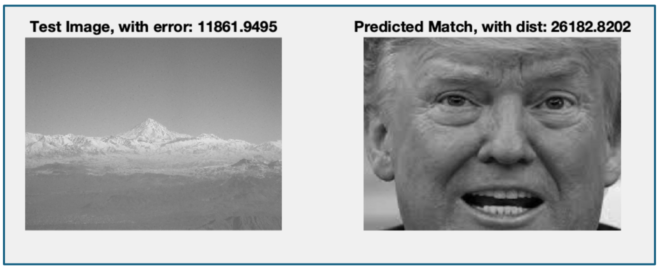
Shown below are **unique** results for K=10, compared to K=21. Results that were the same as K=21 are omitted.



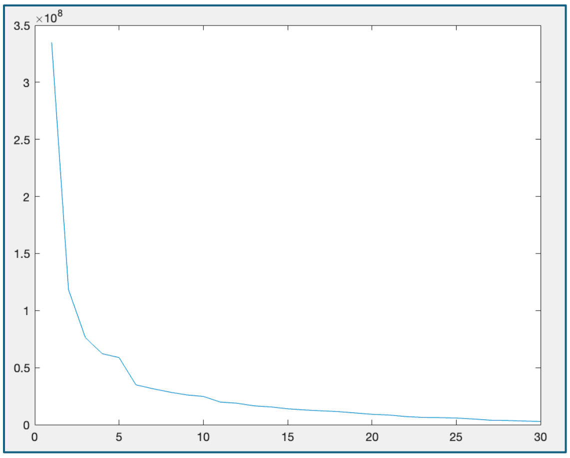


C. K=30

Shown below are **unique** results for K=30, compared to K=21. Results that were the same as K=21 are omitted.



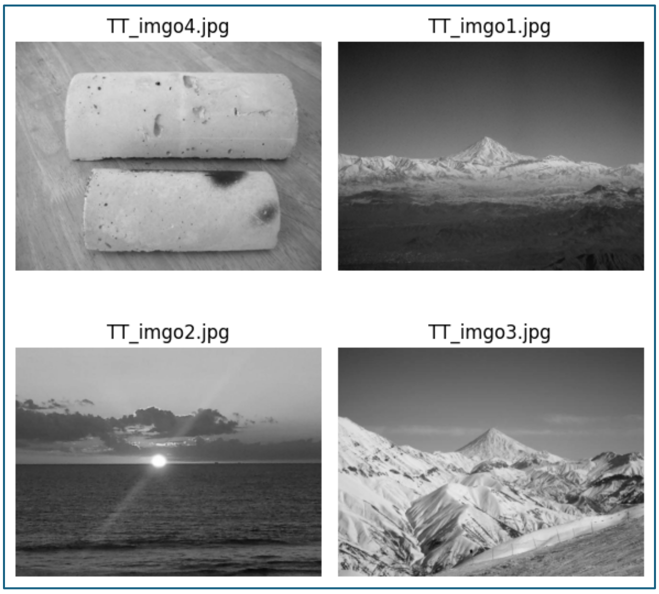
D. Other



Plotted Eigenvalue size vs. index

All Training Images Used

All T1 Images Used



All T2 Images Used

REFERENCES

[1] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991. [Online]. Available: https://www.face-rec.org/algorithms/PCA/jcn.pdf

[2] S. Raschka, "PCA vs LDA," *Python Machine Learning Book - GitHub FAQ*, 2016. [Online]. Available: https://github.com/rasbt/python-machine-learning-book/blob/master/faq/lda-vs-pca.md

[3] A. Mishra, "Face Recognition using PCA (Eigenfaces)," *YouTube*, Sep. 18, 2021. [Online]. Available: https://www.youtube.com/watch?v=g4Urfno4aTc

[4] Wikipedia contributors, "Eigenface," *Wikipedia, The Free Encyclopedia*, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Eigenface