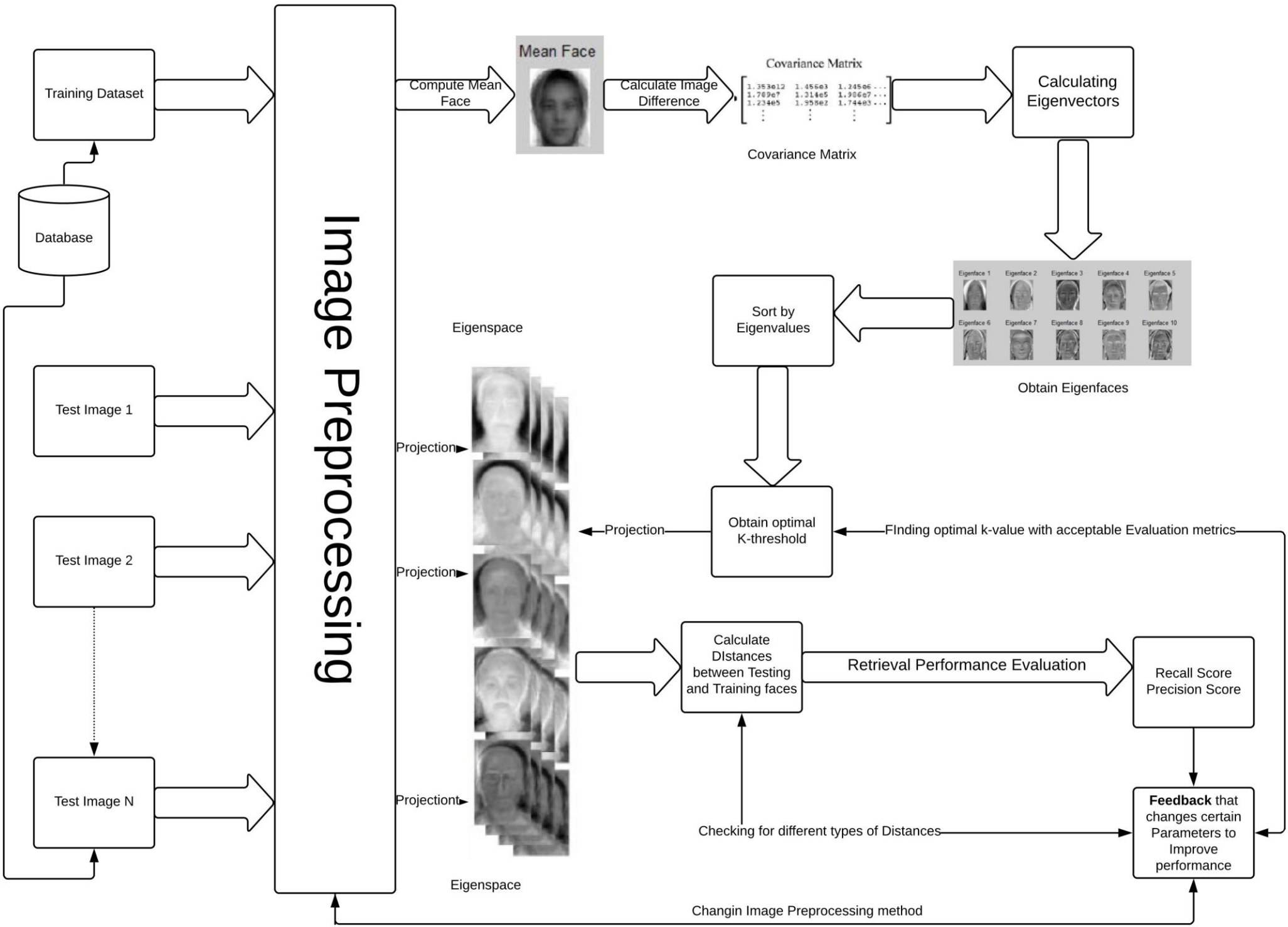

Training Dataset
Database
Test Image 1
Test Image 2
Test Image N
Image Preprocessing
Compute Mean Face
Mean Face
Calculate Image Difference
Covariance Matrix
Covariance Matrix
Calculating Eigenvectors
Eigenface 1
Eigenface 2
Eigenface 3
Eigenface 4
Eigenface 5
Eigenface 6
Eigenface 7
Eigenface 8
Eigenface 9
Eigenface 10
Obtain Eigenfaces
Sort by Eigenvalues
Eigenspace
Projection
Projection
Projection
Projection
Obtain optimal K-threshold
FInding optimal k-value with acceptable Evaluation metrics
Calculate DIstances between Testing and Training faces
Retrieval Performance Evaluation
Recall Score
Precision Score
Checking for different types of Distances
Feedback that changes certain Parameters to Improve performance
Eigenspace
Changin Image Preprocessing method

Description of Design draft for CSE5IR assignment

Database

Preparing a database which stores images(Training and Testing datasets) in a matrix format with their corresponding names in text file.

Training Dataset

Set of Images already provided in the project with additional set of images made based on methodology in Appendix 2.

27 images from Yale dataset with addition 3 images.

Total Number: 30 images

Test Dataset

Set of Images that will be used for testing and performance evaluation of the model

18 images from Yale dataset with addition 2 images.

Total Number: 20 images

Image Pre-processing(Subject to change based on Feedback)

Resizing: All input images should have the same shape and size. Also to avoid too much computation time we can lower the size of the image. However, this is actually a trade-off between quality of Images obtained with time

Cropping: In face recognition applications, we don't need the background of the face image. We are only interested in the face image only. Therefore, we should crop the input images to the head face only

Conversion to Grayscale :To avoid high computational cost of color images we will be converting all the images to grey

Feature extraction Method:

Extract features of face by using PCA on image after converting them into 1D array

Process:-

Compute Mean Face

It computes the mean value when each image is represented as a single vector

Calculate Image Difference, Covariance Matrix Eigenvectors and Eigenvalues

Using mean Image value to calculate image difference and making the covariance matrix for calculating Eigenvectors and Eigenvalues and then sorting the Eigenvalues in descending order

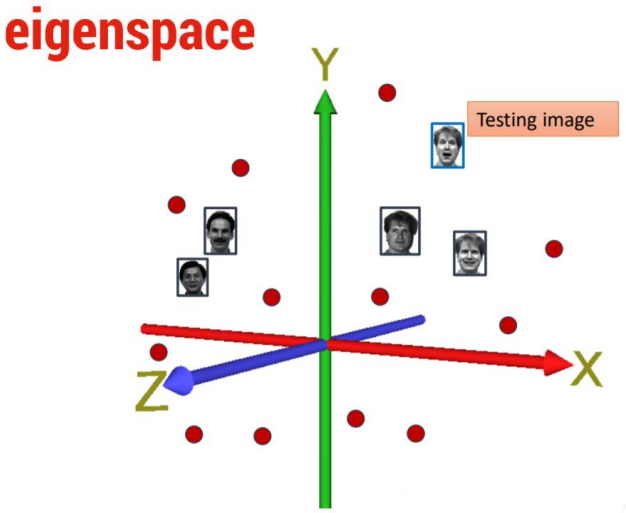
Obtain optimal K-threshold

Determine K1 and K2 according to the following formulas:

$$T_{K1} = \frac{\sum_{p=1}^{K1} \lambda_p}{\sum_{p=1}^{N} \lambda_p} \geq 0.85 \quad \text{and} \quad T_{K2} = \frac{\sum_{p=1}^{K2} \lambda_p}{\sum_{p=1}^{N} \lambda_p} \geq 0.95$$

Then Investigating its affects on evaluation metrics. Also trying find an acceptable trade-off between optimal K-value with the corresponding Recall, Accuracy and Precision scores.

Eigenspace and Distance calculations



Projecting training and testing faces into the eigenspace and calculating the distance between each testing face with respect to the training face to find the most closest faces with testing face.

Using appropriate distance is subject to change based on the performance in the results(Accuracy, Precision and Recall scores).A number of distance calculation methodology will be tried.(eg Euclidean distance, Mahanbolis distance etc..)

Evaluation Metrics and Feedback:

Evaluating the performance of the system based on

- **Precision** quantifies the number of positive class predictions that actually belong to the positive class.
- **Recall** quantifies the number of positive class predictions made out of all positive examples in the dataset.
- **F-Measure** provides a single score that balances both the concerns of precision and recall in one number.

Feedback is loop which connects to choosing a suitable distance metric, Obtaining an optimal K-threshold value(K-set of eigenvectors) and adding/removing a new image pre-processing technique that could improve the above mentioned scores