Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection

Khursheed ali (16305009),Ayush Goyal (16305R011)

November 24, 2017

Chapter 1

Methods

1.1 Correlation

attr database Accuracy: 0.14
Yale database Accuracy: 0.7
Observation: Does't handle pose and illumination changes

1.2 EigenFace

1.2.1 Experiments

X-axis: Number of eigen Vectors(k)
Y-axis: Recognition rate

1. Attr Database
   Accuracy: 0.95 for k=29

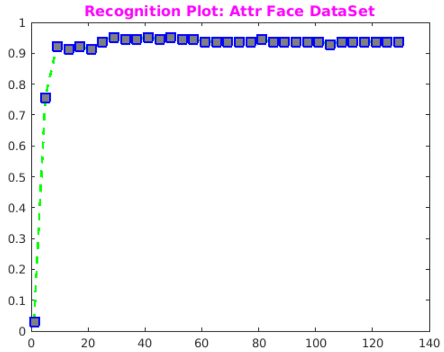

Figure 1.1: attr database

2. Yale Database
   Accuracy: 0.39 for k=289

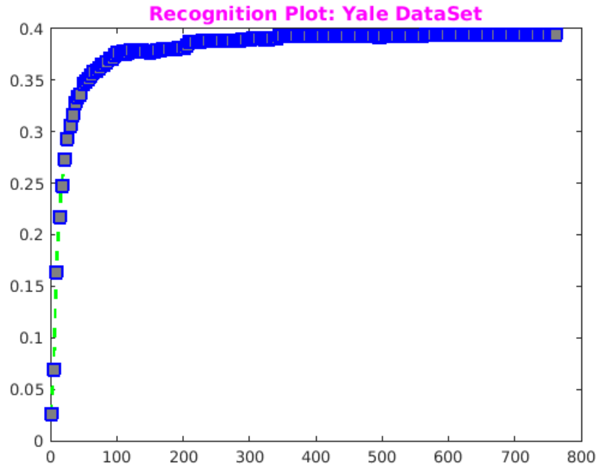

Figure 1.2: Yale database

3. Yale Database handling Illumination changes
   Accuracy: 0.68 for k=477

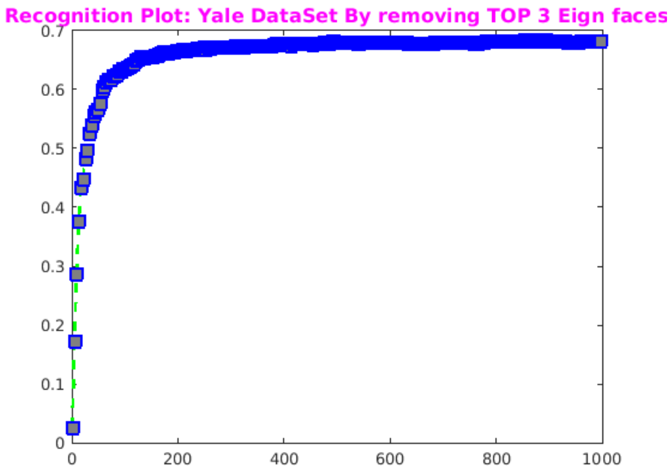

Figure 1.3: Yale database handing illumination changes

Yale Database handling Illumination changes with subsample images **Accuracy: 0.68**

1.3 Fisher Face

Function for Fisher Face :

$$W_{opt} = argmax_W \mid W_T S_B W \mid / \mid W_T S_W W \mid$$

where W_{opt} is optimal projection
S_B is between class scatter
S_W is within class scater

Problem: To obtain the Fisherfaces, we need to compute the inverse of S_w , i.e., S_w^{-1} . If the sample feature vectors are defined in a p-dimensional space and p is larger than the total number of samples n , then S_w is singular.
Solutions: There are three typically used solutions to this problem.

1. In the first solution, we project the sample vectors onto the PCA space of r dimensions, with r<=rank(S_w) and compute the Fisherfaces in this PCA space.
2. The second solution is to project the between- and within-class scatter matrices onto the PCA space of r dimensions, with r<=rank(S_w) and compute the Fisherfaces in this PCA space.
3. The third solution is to add a regularising term to S_w. That is, $S_w + \epsilon I$, where I is the identity matrix and ϵ is a small constant.

We have tested for both Second and third Method. Both methods give almost same results.

1.3.1 Optimization used for second method

Problem: Size of image vector was i.e d=H*W(32256 × 1 for Yale database). So Size of S_B and S_W(scatter matix) was d × d(i.e 32256 × 32256), which is equal to 7.8 gb each. Finding S_W and S_B was a computation and memory overhead.
Optimization: To reduce computation and memory overhead we have used following method.

1. For finding projected value of S_B :

   Mean vector for a particular class of Image=μ_i

   Mean Image vector all class images= $\overline{\mu}$

   $\overline{\mu_i}$=$\mu_i - \overline{\mu}$

   A=[$\overline{\mu_1} * \sqrt{n_1}$ $\overline{\mu_2} * \sqrt{n_2}$ ...]

   $S_B = AA^T$

$W_{PCA}^T S_B W_{PCA} = W_{PCA}^T.A.A^T.W_{PCA}$

$W_{PCA}^T S_B W_{PCA} = (W_{PCA}^T.A).(A^T.W_{PCA})$

Dimension of $W_{PCA}^T.A = (N-c) \times (N)$,where N in number of images and c in number of class

2. For finding projected value of S_W

   Image vector of a class=X_i

   Mean Image vector of a particular class = $\overline{\mu_{ci}}$

   $\overline{X_i}$=$X_i - \overline{\mu_{ci}}$

   B=[$\overline{X_1}$ $\overline{X_2}$ ...]

   $S_W = BB^T$

   $W_{PCA}^T S_W W_{PCA} = W_{PCA}^T.B.B^T.W_{PCA}$

   $W_{PCA}^T S_W W_{PCA} = (W_{PCA}^T.B).(B^T.W_{PCA})$

   Dimension of $W_{PCA}^T.B = (N-c) \times (N)$,where N in number of images and c in number of class

1.3.2 Experiments

1. Attr Database
   Accuracy: 0.906250

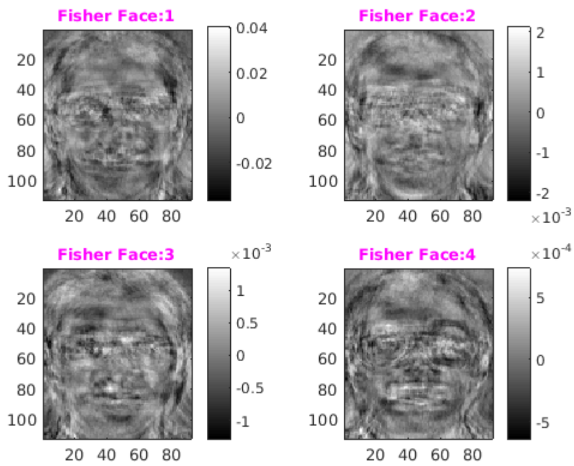

Figure 1.4: Fisher Faces for Attr Databse

2. Yale Database
   Scaling: 1
   Taking (N-c) vectors from W_{PCA} for Dimensionality reduction of image space
   Accuracy: 0.739474

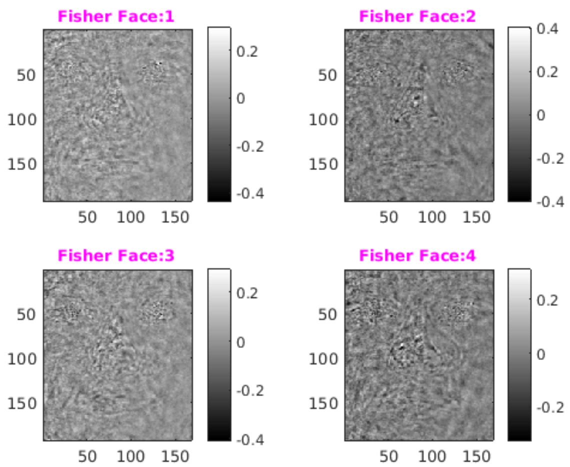

Figure 1.5: Fisher Faces for Yale Databse

3. Yale Database Removing first three vectors of PCA in calculation of Fisher faces then taking (N-c) vectors for Dimensionality reduction
   Scaling: 1
   Accuracy: 0.752
4. Yale Database
   Scaling: 0.5
   Accuracy: 0.781579

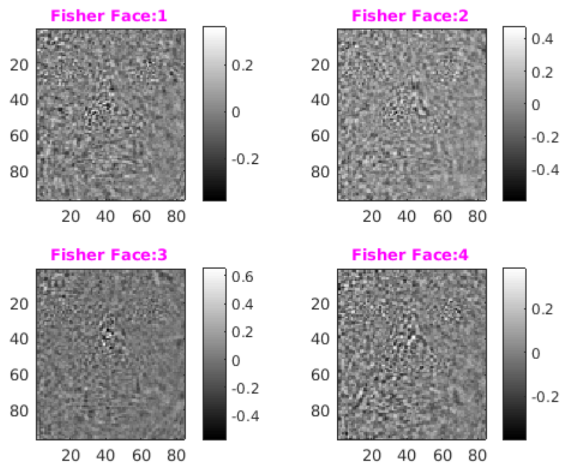

Figure 1.6: Fisher Faces for Yale Databse

5. Extended Yale Database
   Accuracy: 0.93

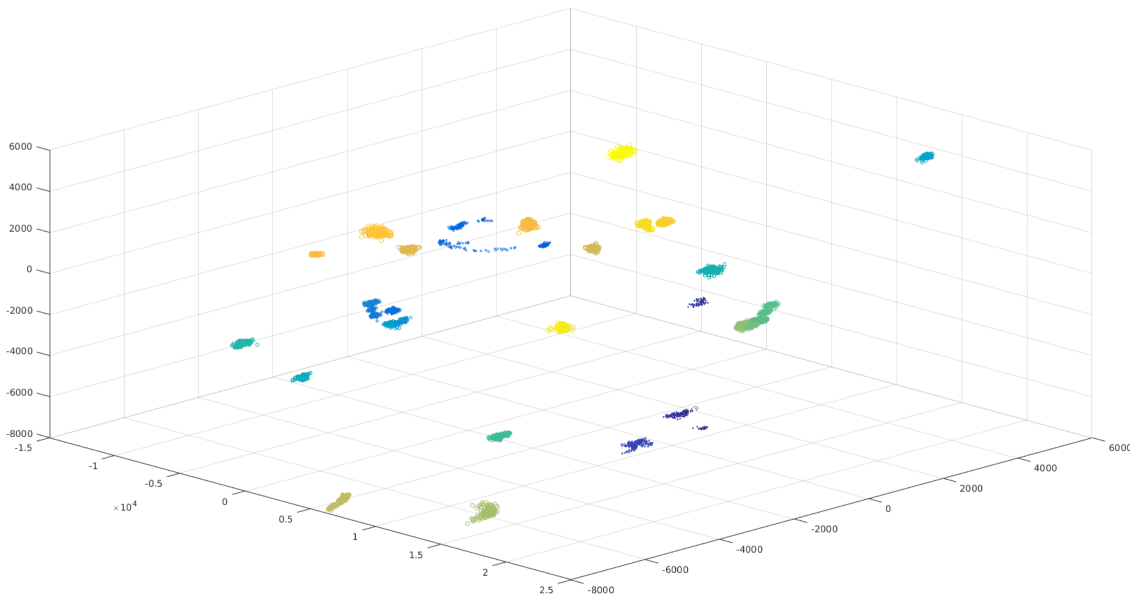

Figure 1.7: projection of test images on first 3 Fisher Face Vectors