Face Recognition Using Eigen Faces

Nisarg Ujjainkar
16110102

24^{th} October 2019

1 About the Project

This is a facial recognition system that uses unsupervised learning to learn and recognise human faces. It does so by projecting the training data-set into a much lower dimension eigen vector space called *Eigen Faces*.

This method of facial recognition was introduced in the paper "Face Recognition Using Eigenfaces"[1] from the MIT Media Labs.

1.1 Training

Main Idea behind eigenfaces is to project the images into a much smaller diamension eigen vector space. To do so we read all the images. The images are in the form of a NxN matrix, we flatten them into N^2x1 vector and combine all such vectors from M images to make a n^2xM matrix.

We now calculate the mean of all the M images and subtract that mean from all the images. Γ_i represent the vector for i^{th} image, the mean vector Ψ is given by

$$\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i$$

$$\Phi_i = \Gamma_i - \Psi$$

We now calculate the covariance matrix of the above matrix and calculate its eigen vectors and eigen values. Let A=$[\Phi_1\Phi_2...\Phi_M]$, covariance matrix of A is given by AA^T. Size of A is N^2xM hence the size of covariance matrix would be N^2xN^2. Computing the eigen vectors and values would e computationally very heavy. We therefore calculate eigen vectors and values of A^TA, whos size is MxM, this is computationally less heavy since M<<N^2. Eigen vector matrix of A^TA is V then eigen vector matrix of AA^T, U is given by AV.

$$u_i = Av_i,\ (||u_i|| = 1)$$

The M eigen values(along with corresponding eigen vectors) are the M largest eigen values of AA^T[2]. We now select K best eigen vectors corresponding to K largest eigen values and project shifted image data Φ to thus new eigen vector space.

$$\hat{\Phi}_i = \sum_{j=1}^{K} w_j u_j,\ (w_j = u_j\Phi_i)$$

Here u_j are the eigen faces. Each training face image Φ_i is represented in the basis vector by

$$\Omega_i = \begin{bmatrix} w_1^i \\ w_2^i \\ \vdots \\ w_K^i \end{bmatrix}\ i = 1, 2, 3, \cdots, M$$

1.2 Testing

Read the test image and convert it to the vector Γ_t. Now calculate $\Phi_t = \Gamma_t - \Psi$. Represent Φ as

$$\Omega_t = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_K \end{bmatrix}\ (w_i = u_i^T\Phi)$$

Calculate index such that $||\Omega_{index} - \Omega_t||$ is minimum and return that index.

2 About the Dataset

The dataset used is "ORL Face Database"[3]. It has the following properties

- Number of classifications = 40
- Training images per classification = 9
- Test images per classification = 1
- Size of each image 92x112

3 Results

After testing for all the test images, results are as follows

Source	Predicted
S1	S1
S2	S2
S3	S3
S4	S4
S5	S5
S6	S6
S7	S7
S8	S8
S9	S9
S10	S8
S11	S11
S12	S12
S13	S13
S14	S14
S15	S15
S16	S16
S17	S17
S18	S18
S19	S19
S20	S20
S21	S21
S22	S22
S23	S23
S24	S24
S25	S25
S26	S26
S27	S27
S28	S28
S29	S29
S30	S30
S31	S31
S32	S32
S33	S33
S34	S34
S35	S40
S36	S36
S37	S37
S38	S38
S39	S39
S40	S40

Accuracy is 95% by keeping threshold value very large and selecting 20 best eigen vectors.

4 About the Code

The code is written in python. to run the code use the following command
`$python3 faceRecognition.py`
This would train the classifier and run the test on all the test images and print a list of predictions. To test for a particluar test image source eg. S6 use the command.
`$python3 faceRecognition.py 6`
This will take test image from S6 try to classify it.

Please note that code file must be placed in dataset folder and the current working directory of terminal should be that folder.

4.1 Dependencies

numpy and openCV

References

[1] Matthew A. Turk and Alex P. Pentland *Face Recognition Using Eigen Faces*. Vision and Modeling Group, The Media Lan, MIT, 1991.

[2] M. Turk and A. Pentland, "Eigenfaces for Recognition". *Eigenfaces for Face Detection/Recognition*. Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991.

[3] ATT "Database of Faces", `https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`. (Link Dead).