

Face Recognition using Eigenfaces

Libraries Used :

opencv
pandas
numpy
fnmatch
os
sys

This is an implementation of the paper on face recognition using PCA algorithm by M. Turk and A. Pentland.

Face Recognition using AT&T dataset :

Dataset Used : AT&T face dataset

The Face Recognition model has two parts : Train and Test

Training :

I use the images in AT&T dataset as training images in folder AT&Ttrain. I remove some random images from the dataset and put them in another folder AT&Ttest. These images will be used while testing the model.

Working :

1. Each image I_i is a 2D vector of the form $N^2 \times N^2$
2. Flatten the images to form vectors Γ_i of the form $N^2 \times 1$
3. Find the mean vector Ψ of all image vectors Γ_i
4. Then we find the equivalent vector by subtracting the mean vector from image vectors

$$\Phi = \Gamma_i - \Psi$$

5. Then find the Covariance Matrix :

$$\mathbf{C} = \mathbf{A}\mathbf{A}^T$$

\mathbf{C} is a $N^2 \times N^2$ matrix

where $\mathbf{A} = [\Phi_1 \Phi_2 \Phi_3 \dots \Phi_M]$

\mathbf{A} is a $N^2 \times M$ matrix

6. Now we find the eigenvectors and eigenvalues of the covariance matrix

$\mathbf{C} = \mathbf{A}\mathbf{A}^T$, which is very large.

Let corresponding eigenvectors be \mathbf{u}_i

So we instead find eigenvalues and eigenvectors for $\mathbf{C} = \mathbf{A}^T\mathbf{A}$.

Let corresponding eigenvectors be \mathbf{v}_i

7. Eigenvectors \mathbf{v}_i and eigenvalues μ_i

$$\mathbf{A}^T\mathbf{A}\mathbf{v}_i = \mu_i \mathbf{v}_i$$

And $\mathbf{u}_i = \mathbf{A} \mathbf{v}_i$

Also normalize \mathbf{u}_i such that $\|\mathbf{u}_i\| = 1$

8. Now we arrange the eigenvalues in descending order and take the first k eigenvalues.

9. Finally we calculate weights corresponding to all the k eigenvectors

$$\mathbf{w}_j = \mathbf{u}_j^T \Phi_j$$

Such that

$$\Phi_{new} - mean = \sum \mathbf{w}_j \mathbf{u}_j \quad \text{for } j = 1 \dots k$$

10. This is the basic PCA algorithm.

Testing :

I use the test images that are present in the folder AT&Ttest. These images are randomly picked from the AT&T face dataset and are not present in training images.

Working :

1. Flatten the test image and subtract the mean found during training

$$\Phi = \Gamma - \Psi$$

2. Next we project this image on the eigenspace formed by the PCA algorithm

$$\Phi_{new} = \sum w_j u_j \text{ where } w_j = u_j^T \Phi_j$$

3. We find the difference between the weights of the training images and the test image such that

$$D = \|\Phi - \Phi_{new}\|$$

4. If the value of D is greater than a threshold value T then D is the minimum the face at minimum distance from the test face.
5. We can find the class of this face from the training set which is at the least distance from the test face. Hence, we recognise the test face.

Results :

The results is an output that shows in which class each test image belongs.

```
pranjali@pranjali-Inspiron-5567: ~/Desktop/jupyter notebooks/PRP/16110119_Pranjali_Jain_Assignment1
File Edit View Search Terminal Help
pranjali@pranjali-Inspiron-5567:~/Desktop/jupyter notebooks/PRP/16110119_Pranjali_Jain_Assignment1$ p
ython Face_Recognition_AT&T_dataset.py
FACE RECOGNITION

Options :
    1. Train and test AT&T dataset

Please enter the option (1)
1
Training existing AT&T dataset
size of images 10304
Recognising test images

Test face at path /home/pranjali/Desktop/jupyter notebooks/PRP/16110119/AT&Ttest/s356.pgm
This test face belongs to class s35

Test face at path /home/pranjali/Desktop/jupyter notebooks/PRP/16110119/AT&Ttest/s107.pgm
This test face belongs to class s10

Test face at path /home/pranjali/Desktop/jupyter notebooks/PRP/16110119/AT&Ttest/s47.pgm
This test face belongs to class s4

Test face at path /home/pranjali/Desktop/jupyter notebooks/PRP/16110119/AT&Ttest/s27.pgm
This test face belongs to class s2

Test face at path /home/pranjali/Desktop/jupyter notebooks/PRP/16110119/AT&Ttest/s57.pgm
This test face belongs to class s5

Test face at path /home/pranjali/Desktop/jupyter notebooks/PRP/16110119/AT&Ttest/s37.pgm
This test face belongs to class s3

Test face at path /home/pranjali/Desktop/jupyter notebooks/PRP/16110119/AT&Ttest/s97.pgm
This test face belongs to class s9

Test face at path /home/pranjali/Desktop/jupyter notebooks/PRP/16110119/AT&Ttest/s87.pgm
This test face belongs to class s8

Test face at path /home/pranjali/Desktop/jupyter notebooks/PRP/16110119/AT&Ttest/s77.pgm
This test face belongs to class s7

Test face at path /home/pranjali/Desktop/jupyter notebooks/PRP/16110119/AT&Ttest/s67.pgm
This test face belongs to class s6

Test face at path /home/pranjali/Desktop/jupyter notebooks/PRP/16110119/AT&Ttest/s17.pgm
This test face belongs to class s1

Test face at path /home/pranjali/Desktop/jupyter notebooks/PRP/16110119/AT&Ttest/s268.pgm
This test face belongs to class s26
Done
pranjali@pranjali-Inspiron-5567:~/Desktop/jupyter notebooks/PRP/16110119_Pranjali_Jain_Assignment1$
```

Real Time Face Recognition using webcam :

Real time face recognition has involves two steps :

1. Face Detection
2. Face Recognition

Training :

Working :

Creating the dataset :

1. First I use the webcam to create a dataset of images. The user can click their images and store them.
2. To click an image press enter and press esc to close the camera.

Face Detection :

I used haarcascades classifier to detect faces in these images.

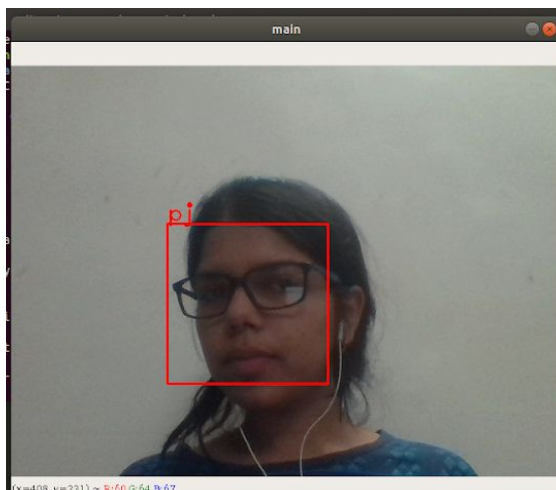
Testing :

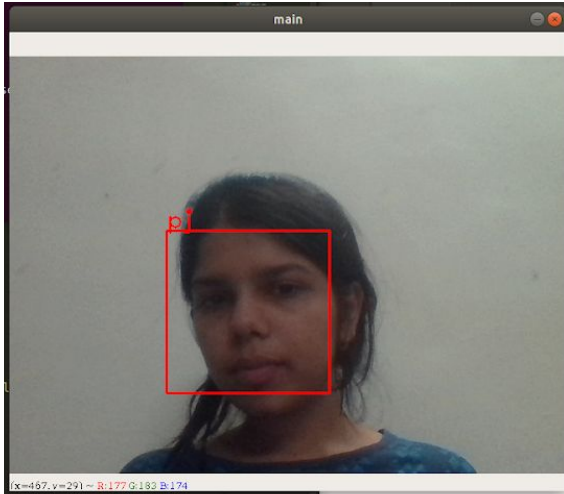
Face Recognition :

Finally, I use the above mentioned PCA algorithm to recognise faces.

We capture video from the camera and the PCA algorithm recognises the face class and displays it on the screen.

Results :





I am able to recognise faces in realtime using my webcam.

References :

1. Turk, Matthew A., and Alex P. Pentland. "Face recognition using eigenfaces." In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pp. 586-591. IEEE, 1991.
2. http://www.vision.jhu.edu/teaching/vision08/Handouts/case_study_pca1.pdf
3. <https://github.com/zwChan/Face-recognition-using-eigenfaces/blob/master/eigenFace.py>
4. <https://www.instructables.com/id/Real-time-Face-Recognition-an-End-to-end-Project/>
5. <https://github.com/povilasb/face-recognition>