IIT GANDHINAGAR

PROBABILITY AND RANDOM PROCESSES

ES 331

Face Recognition Using Eigenfaces

Submitted by:
S. Vinu Sankar
16110143

October 21, 2018

1 Introduction

The report is on an approach which was actually introduced in the paper Face Recognition Using Eigenfaces [1] by Turk et al. wherein detection and identification of human faces are performed in near real-time. 2D face images known to us are used for training the model to project face images onto a feature (face) space that encodes the variation among known faces. The algorithm employs principal component analysis to make the computations easier. The model is tested on a dataset [2] with total 400 images (280 training images + 120 test images) to get an accuracy of **95.83%**. The implementation in this report takes on average **18 milliseconds** to identify and recognize a single test image.

2 Methodology

Preprocessing:

1. Acquire image dataset [2].
2. Split into train (7 per class) plus test (3 per class) data items for each of 40 classes (identities).
3. Resize all images to same size N×N, where N = 128.
4. Center all images such that all features like eyes, noses, and, mouth are all aligned across all images. (optional)

Algorithm:

1. Let the flattened ith train image be represented by Γ_i.
2. Find average face image $\Psi = \dfrac{1}{M}\sum_{n=1}^{M}\Gamma_n$, where M is the total number of train images (280).
3. Find $\Phi_i = \Gamma_i - \Psi$.
4. Make an M×M matrix L, where $L_{mn} = \Phi_m^T\Phi_n$.
5. Calculate the eigenvectors v and eigenvalues λ for L matrix.
6. Calculate the eigenvectors u given by $u = v^T.\Phi$ and normalize it.

7. Select the first K highly weighed eigenvectors from the calculated u vector based on the λ values and set it to be u, where K = 150.

8. Let $\Omega = \Phi.u^T$ be the matrix with vectors Ω_k representing the kth face class.

9. To predict a face, get the test image, resize it to N×N and flatten it. Let the vector be Γ.

10. Let $\Omega_t = u.(\Gamma - \Psi)^T$.

11. If $||\Omega_t - \Omega_k||^2 < \epsilon_k$, the test face image belongs to the class k. ϵ_k denotes the distance of the test image face space vector from the class representative face vector space. The value of ϵ_k comes out to be 20,028,607 according to the implementation.

12. If $\epsilon = ||\Phi - \Phi_f||$ is less than a threshhold value , then the test image is a face image. Else it is not, where $\Phi_f = \Omega^T.u$

3 Results

The dataset contains a total of 40 classes with 10 images each. On using 7 images per class for training, the test accuracy obtained is **95.83%** and that on using 6 images per class is **95.00%**. The algorithm on an average takes **18 milliseconds** to predict a single test image class using the code written in Python 3.6. Whereas the work in [1] reports a run time of 4 milliseconds, as we know that Python is 5 times slower than C++. The threshhold values are found to be $\epsilon_k = 20028607$ for class detection and $\theta = 2858$ for face detection.

4 Conclusions

1. The model seems to improve on increasing K value.

2. On increasing training set accuracy improves.

3. Improvements happen only with the trade off run time.

4. PCA makes the algorithm near real-time without compromising much on accuracy.

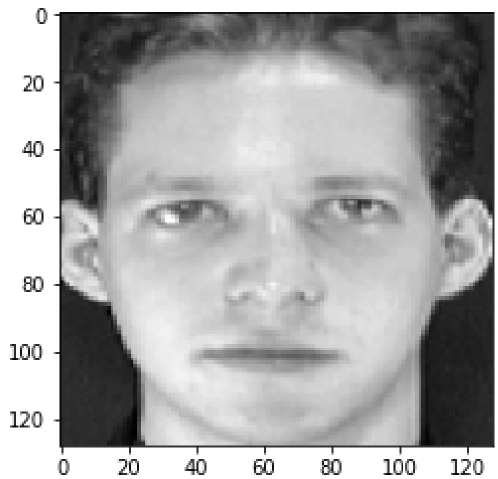

Figure 1: A sample train image face.

5 References

1. M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Maui, HI, USA, 1991, pp. 586-591.

2. https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

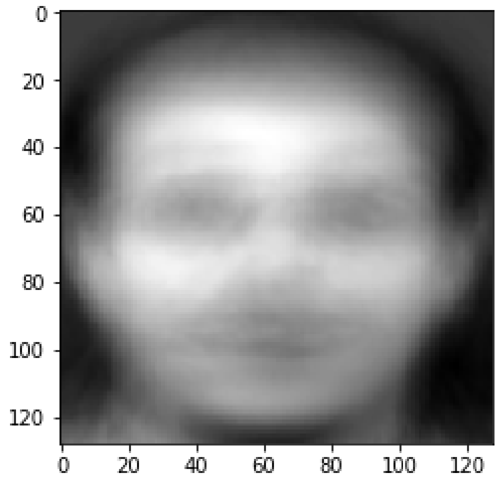

Figure 2: Average face image across all train images.

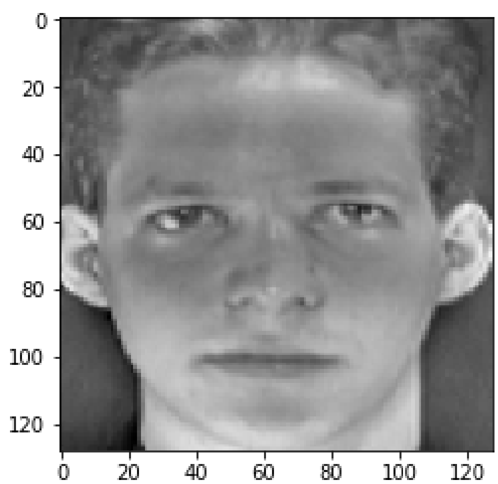

Figure 3: A sample Φ train image.

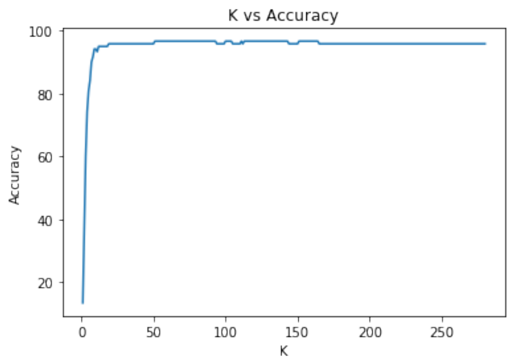

Figure 4: K vs Accuracy plot.

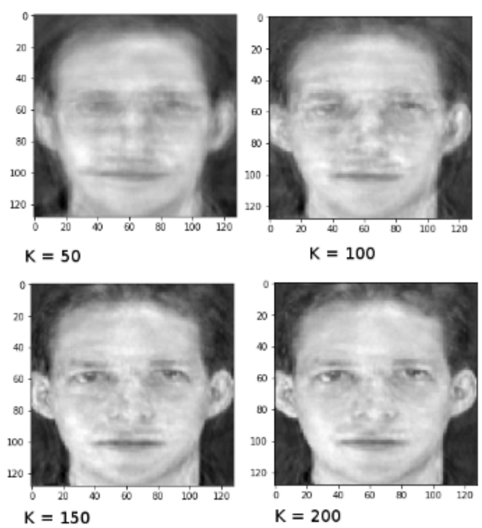

Figure 5: Image reconstruction with different K values.