Ukrainian Catholic University

Faculty of Applied Sciences

Data Science Master Programme

Face recognition using PCA

Linear Algebra final project report

Authors:
Vitalii Duma
Serhii Tiutiunnyk

24 January 2019

Abstract

This report covers development face recognition system based on Principal Component Analysis (PCA) method. PCA is a linear algebra method based on an orthogonal transformation which converts correlated variables into uncorrelated variables where first k principal components explain the biggest variance. This allows reducing the number of variables with the smallest loss of data. Images from the training set are represented as vectors, which are used for building covariance matrix. Eigenvectors of covariance matrix are called eigenfaces. The recognition process consists of projecting a test image onto the eigenfaces space and evaluation minimum Euclidean distance between original and projected images.

1 Introduction

Face recognition problem became very popular among computer vision problems. The face recognition problem appears in the security sphere, unlocks systems, natural sciences, social networks, etc. The base goal is to build the system which detects whether the face on the photo is related to the stored faces. Then this system can be expanded to the frontal view recognition, perhaps for recognition the gender of a person, etc. This project aims to develop a face recognition system with appropriate efficiency using principal component analysis approach and carry out tests for hyperparameters optimization.

2 Motivation

There is a set of possible solutions for face recognition problem, but there are some restrictions for this system. The solution requires integrity of adding changes to the base of faces, appropriate efficiency, high speed, insensitivity to small changes of image. The most significant advantage of the PCA approach is its simplicity and respectively low computation power requirements for changing face base.

3 Problem settings

The input data to this problem are the base of images (training set) and the test image which needs to be determined if it belongs to some class of images from the training set. If it is not recognized otherwise the test image is identified as unknown. This method measures image relation to some class by the distance from the test image to the training set of images. So, it requires to determine a threshold level of recognition test image as face related to the base of faces. The entire face recognition process consists of the training and recognition part.

The training part includes the next subproblems:

1) Find or build an appropriate set of face images for training.

2) Transform the image matrix to the grayscale image vector and form a training matrix of images.

3) Calculate the eigenvalues and eigenvectors from the training set covariance matrix.

4) Select an appropriate number of principal components.

5) Project training faces images onto the eigenface space.

After performing the training part, the next part can be fulfilled:

1) Calculate a projection of a test image onto the eigenfaces space.

2) Find an appropriate threshold and detect if the image is a face.

3) If the input image classified as a face, then find the class to which input face belongs.

An auxiliary task for the face recognition problem is applying criteria for finding an optimal number of principal components. This criterion should penalize a number of principal components and promote the fraction of explained variance.

4 Approach to solution

4.1 Image representation

m images of size $n \times n$ are represented as m vectors $\{p_1 \dots p_m\}$ of size $n^2 \times 1$. First of all, training images are transformed into image vector p_i. Then we build a matrix of images P.

$$P = [p_1, p_2, \dots, p_m] \quad (1)$$

So, the size of matrix P is $n^2 \times m$.

4.2 Mean centered image

The algorithm requires to centralize data. It will make finding a covariance matrix easier.

$$\mu = \frac{1}{m} \sum_{i=1}^{m} p_i \quad (2)$$

After that, subtract the average image vector μ from each training set face image.

$$p'_i = p_i - \mu, \quad i = 1, \dots, m \quad (3)$$

And build a data set matrix A. The size of matrix A is the same as P.

$$A = \{p'_1, p'_2, \dots, p'_m\} \quad (4)$$

4.3 Covariance matrix

The covariance matrix is calculated by formula:

$$C = AA^\top \quad (5)$$

The size of matrix is $n^2 \times n^2$. Then calculate eigenvalues and eigenvectors of the covariance matrix C. Calculation of eigenvalues and eigenvectors of matrix C directly is not efficient numerically as the number of nonzero eigenvalues of C is not greater than N. Then find eigenvalues and eigenvectors of matrix $A^\top A$ [3].

$$A^\top A v_i = \lambda_i v_i$$

Then multiply by A right sides, considering that λ_i is corresponding eigenvalue.

$$AA^\top A v_i = \lambda_i A v_i$$
$$C(Av_i) = \lambda_i (Av_i)$$

$$u_i = Av_i \quad (6)$$

Thus eigenvalues of C are equal to λ_i, and corresponding eigenvectors are $u_i = Av_i$, $i = 1, \dots, m$.

4.4 PCA

K eigenvectors (corresponding to K largest eigenvalues) are choosed to construct the eigenfaces space. There are different approaches how to choose the optimal number of principal components. An Akaike's information criterion (AIC) is applied to find the optimal number of principal components. The general AIC function of number components k is:

$$AIC(k) = -2n(m-k)log(\rho(k)) + 2k(2m-k)$$

In our case the number of observation is n^2

$$AIC(k) = -2n^2(m-k)log(\rho(k)) + 2k(2m-k),$$

where $\rho(k)$ is a fraction of explained variance for the first k components. It is crucial to select an appropriate k because otherwise, the eigenfaces space can reconstruct any image with pretty high quality. For example, on the figure 1 the dog's image is reconstructed by faces principal components

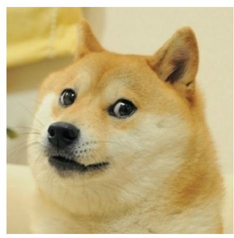

Figure 1: Original dog's image

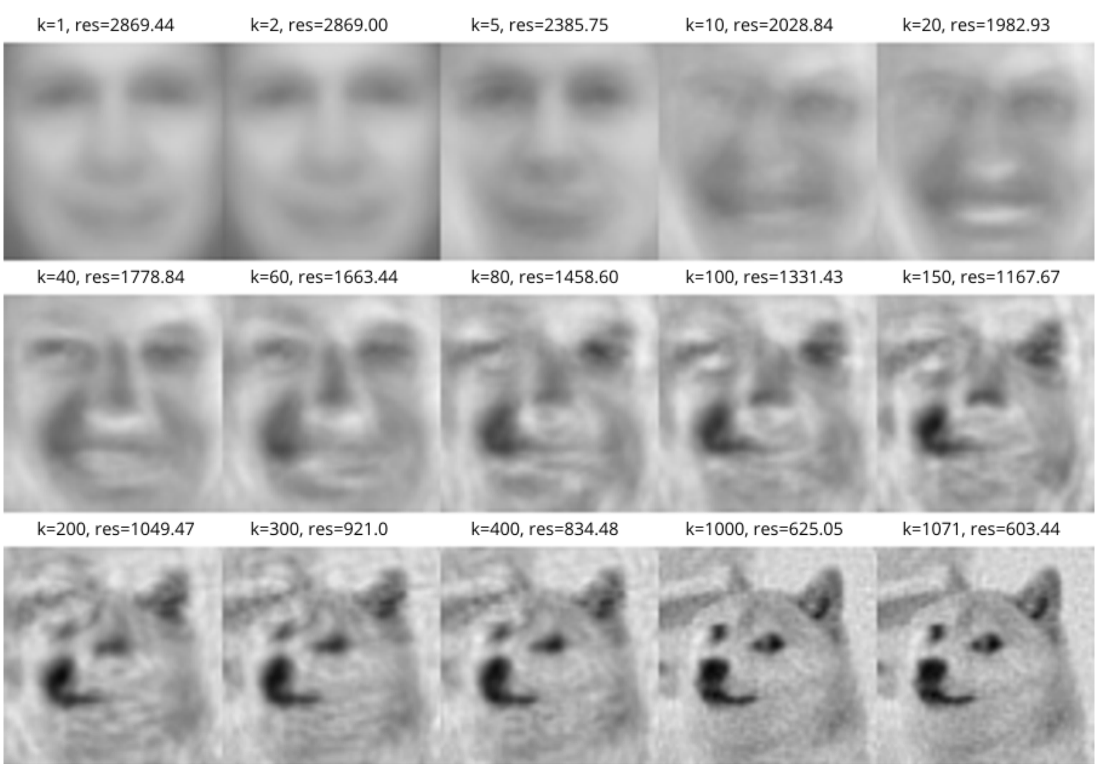

Figure 2: Reconstructed dog's image

After $k > 150$ the dog can be recognized. Having selected optimal K, training set is projected onto the eigenface space.

$$U = \{u_1, u_2, \ldots, u_K\}$$
$$\omega_i = u_i^\top (p_i - \mu), \qquad i = 1, \ldots, K$$
$$\Omega = U_K^\top (P_K - \mu) \qquad (7)$$

4.5 Test image recognition

Read test image file and convert image matrix into a vector and project it onto the eigenfaces space.

$$\omega_t = U^\top (p_t - \mu)$$

Then find the minimum Euclidean distance to the training images, projected onto the eigenfaces space.

$$d_{min} = \min_{i=1,\ldots,K} \sqrt{(w_t - w_i)^\top (w_t - w_i)}$$

Also find c which corresponds to the minimum distance to the classes of person's images. On this step the threshold needs to be found to detect whether the test image belongs to some class and whether the test image is the face image in the first place. Checking if the test image is a face can be provided by comparison initial test image vector p_t and reconstructed image vector p_t^* from the eigenfaces space U. The reconstructed image is found by the next formula:

$$p_t^* = UU^\top (p_t - \mu) + \mu \quad (8)$$

If the test image is not a face, Euclidian distance between initial image and reconstructed image $\|p_t^* - p_t\|$ will be very small and the threshold will refer it to the faces.

5 Solution

5.1 Faces database

For this problem LFWcrop [1] database was used. This database contain 1071 cropped face images. A few of them are shown below.

Figure 3: LFWcrop database

Face classifier is implemented in python and is available at public Github repository [7].
Mean face among training faces:

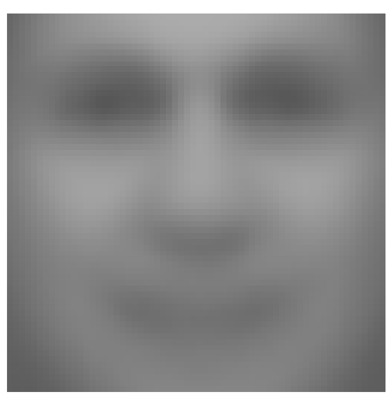

Figure 4: Mean face

Then mean face was subtracted from each image to have zero-mean data.

5.2 Principal components

The next figure shows the explained variance fraction depending on the number of components.

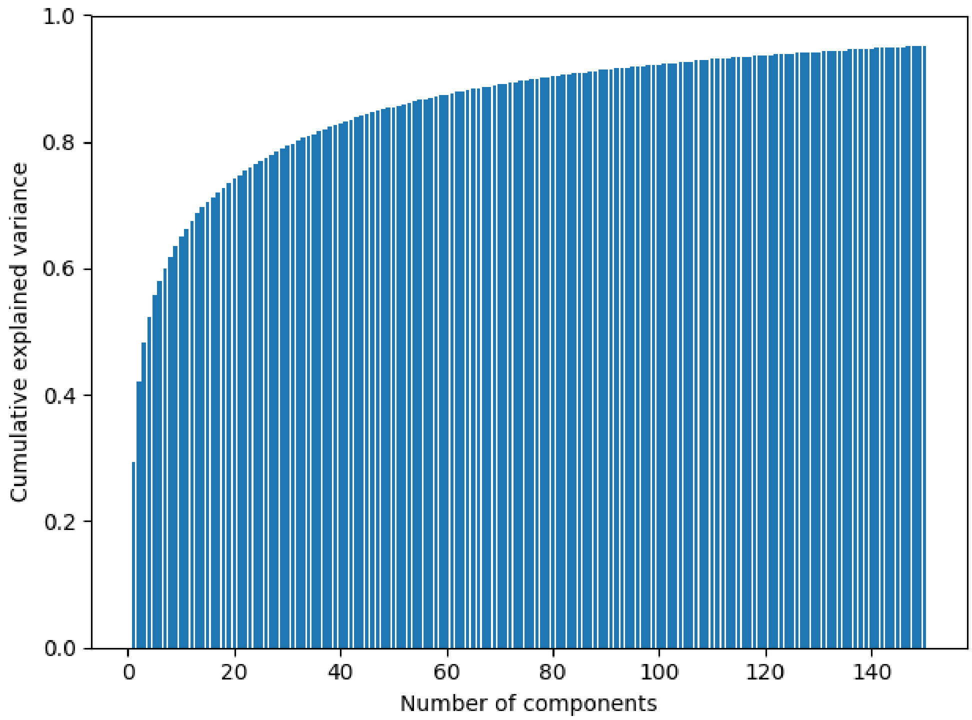

Figure 5: Variance explained by top k principal components

Figure 5 shows the top 100 eigenfaces i.e. top 100 principal components of faces. It is easy to see on the picture that top k components show global differences like shape of the face, shadows tints of large shapes of the picture. And the last components show more details of faces such as nose, eyes, mouth etc.

Figure 6: Top eigenfaces (first 100 PCA components)

5.3 Reconstructed image

The image of Donald Trump's face is reconstructed from eigenfaces space by formula (8). The original image is shown below:

Figure 7: Original image

Reconstructed images using different number of top principal components show that with increasing the number of components image becomes similar to the original image and the mean residual value (distance between original and reconstructed images) decreases.

Figure 8: Reconstructed image by projecting on the truncated eigenfaces space

The graph on the figure 8 shows the changes of mean residuals depending on the number of components.

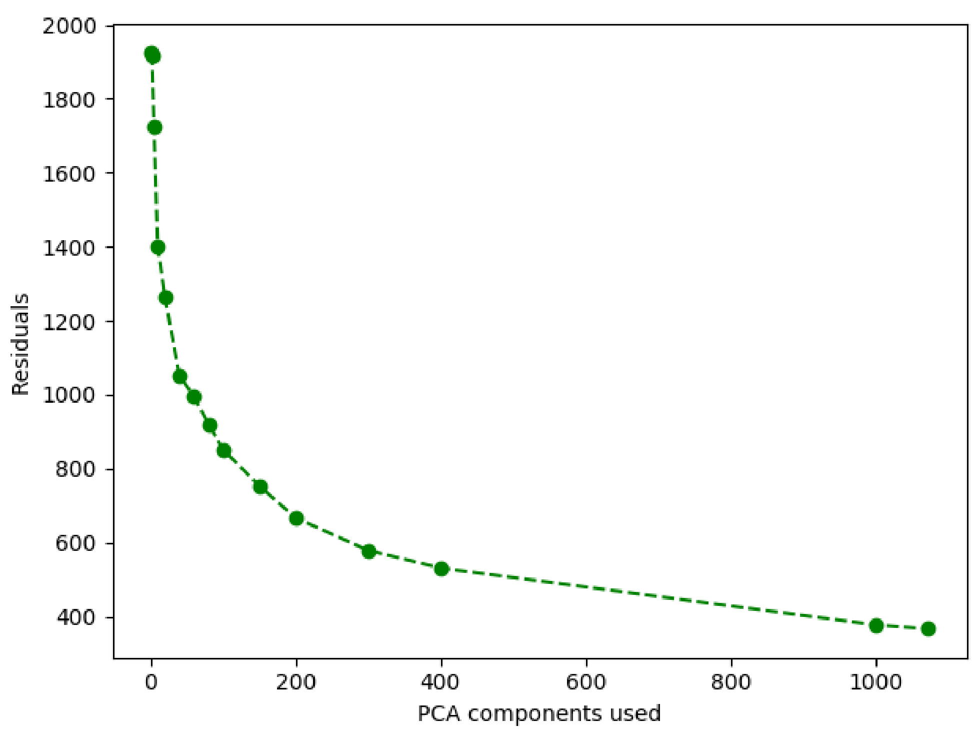

Figure 9: Error when first k principal components were used

The next step is to find appropriate value of threshold to recognize face image from all other images. The initial set of images are divided into 2 parts of 500 items and added 500 not faces images. The initial threshold is calculated by formula [4]:

$$\theta = \frac{1}{2} \max_{i,j=1\ldots k} \|w_i - w_k\| \quad (9)$$

where w_i, $i = 1 \ldots k$ are eigenfaces. So, as a base threshold we take the half maximal distance between 2 eigenfaces.

Figure 10 shows face and not face images mean residuals and the graph of the threshold value θ depending on the number of components. The mean residuals of not face images are larger than mean residuals of face images, and the difference is higher at the lower numbers of principal components. Of course, the optimal threshold can be found for the different number of principal components, but the accuracy is higher when the difference between residuals is more significant.

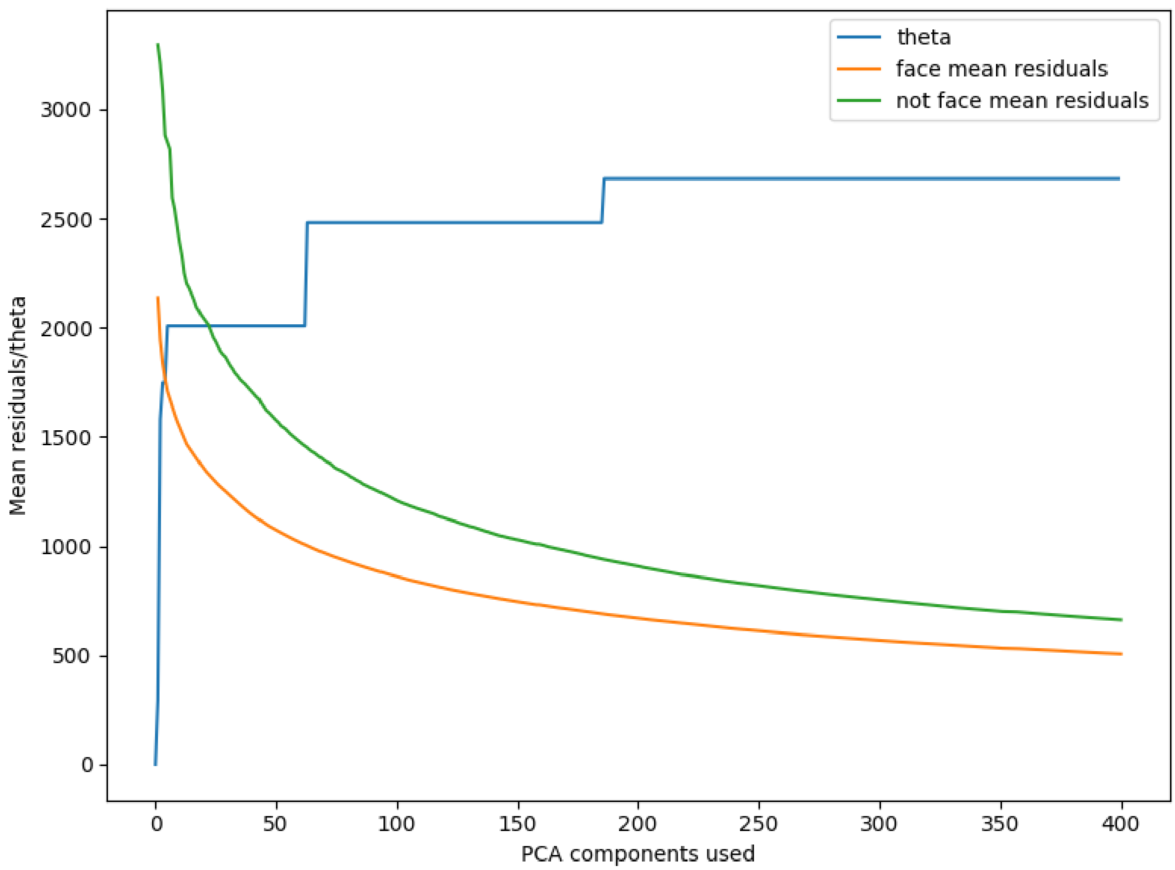

Figure 10:

6 Evaluation

There were used two sets of pictures to perform the experiment mentioned above. The first one is LFWcrop [1] database and the second one is CIFAR-10 [2]. The first half of LFWcrop face images is used to build the eigenfaces space and retrieve the principal components of the face images. The second half and CIFAR-10 database of not face images are used to perform validation in order to find the optimal number of principal components and the value of the threshold.

6.1 Plan of the experiment

The experiment consists of the next steps:

1) Build a training set of face images and transform it to the image vectors.

2) Calculate correlation matrix and find its eigenvalues and eigenvectors.

3) Project test set images onto the eigenfaces space and then reconstruct it.

4) Repeat the step 3 for different numbers of principal components and different thresholds.

5) Build the recognition accuracy graph if the image is a face depending on the number of principal components for different threshold.

6) Choose the optimal number of principal component and the threshold value.

The figure 11 shows graphs of the accuracy depending on the number of principal components for different thresholds.

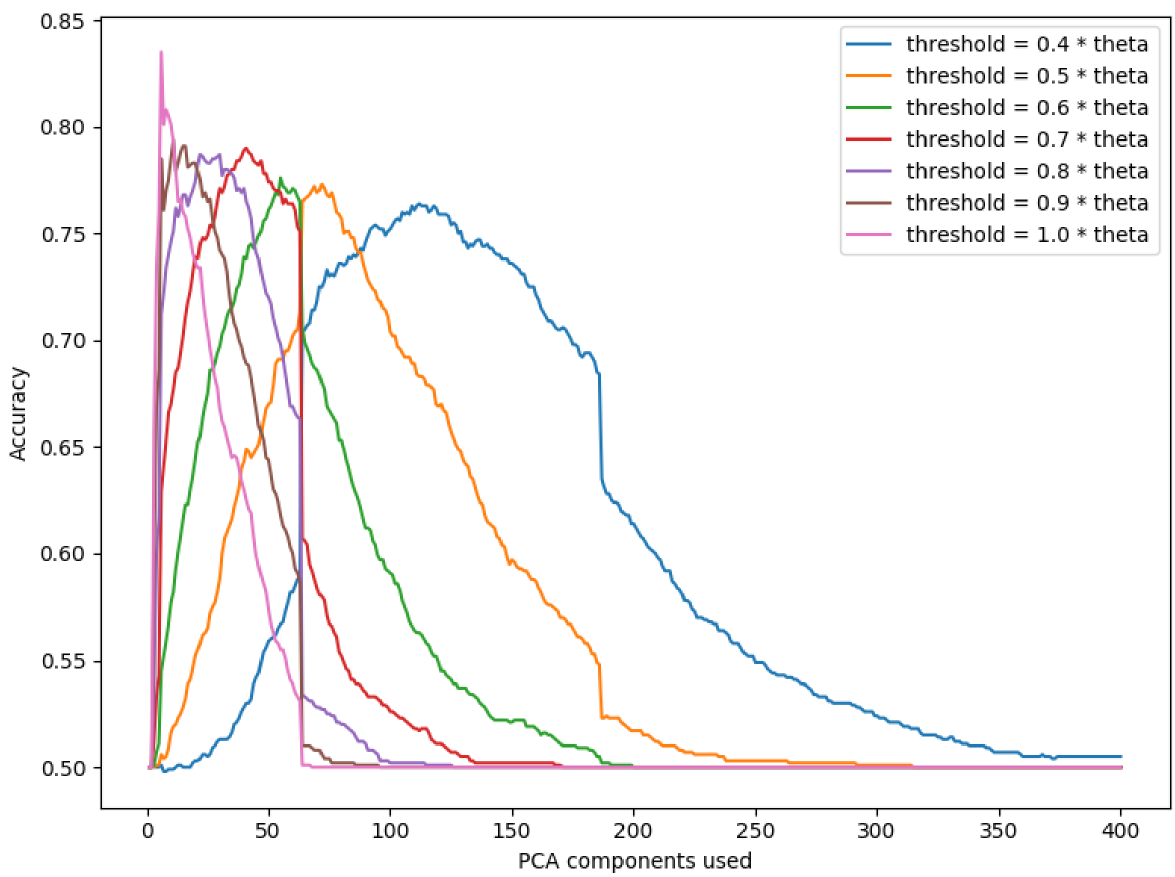

Figure 11: Accuracy depending on different numbers of components and thresholds

According tho the figure 11, the optimal threshold value is equal to θ and the optimal number of principal components for the task of distinguishing the face on the image from not face is equal to 6. Experiment shows that having 6 components is enough to distinguish images. Moreover, it's the optimal number in terms of accuracy. The accuracy equals to 0.84.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN},$$

where TP is a number of true positive answers, TN - true negative answers, FP and FN are false positive and false negative answers respectively.

6.2 Method complexity

There are a few numerical complicated tasks. The first one is finding eigenvalues and eigenvectors. The complexity of this problem is $O(n^3)$, where n is a number of face images which form eigenfaces space. It is equal to 1071 for the LFWcrop data set. The next numerical complicated task is matrix multiplication. The complexity of this task is also $O(n^3)$, where n is the size of the image vector. The size of the image vector is 4096. But the matrix multiplication operation is much easier as it takes a small number of principal components for recognition task. One more advantage of this method is unnecessary to change principal components for the new classification set of images. It requires only to project new image dataset and then compares it with the test images.

7 Conclusions

The main advantages of using the PCA method for the face recognition task are its simplicity and low computational power requirement. Also applying this method for classification task does not require face principal components recalculation. This project investigates choosing the number of principal components and threshold improvement over the PCA method. It turned out that it needs very few components to recognize a face and not face image. The optimal threshold is equal to the half of the maximal distance between eigenfaces.

7.1 Future Work

This method can be applied for the face classification problem, extended on the recognition frontal face view, etc. Along with it, the PCA approach for face recognition can be improved by Normalized Principal Component Analysis (N-PCA) to remove the lightning differences and background effects.

References

[1] Labeled Faces in the Wild Home, The University of Queensland, https://www.itee.uq.edu.au/conrad/lfwcrop/

[2] The CIFAR-10 dataset, University of Toronto, http://www.cs.toronto.edu/kriz/cifar.html

[3] arXiv:1705.02782 [cs.CV]

[4] Dey, Sandipan. "EigenFaces and A Simple Face Detector with PCA/SVD in Python." Sandipanweb, 8 Jan. 2018, sandipanweb.wordpress.com/2018/01/06/eigenfaces-and-a-simple-face-detector-with-pca-svd-in-python/.

[5] Pissarenko, Dimitri. "Eigenface-based facial recognition." December 1st (2002).

[6] M. Turk, A. Pentland: Face Recognition using Eigenfaces, Conference on Computer Vision and Pattern Recognition, 3 – 6 June 1991, Maui, HI , USA, pp. 586 – 591.

[7] Vitalii Duma, Serhii Tiutiunnyk, PCA face recognition, (2018), GitHub repository, https://github.com/dumavit/pca-face-recognition