/

# Computer Vision
# Assignment 2

Sami Sellami

April 11, 2019

## Face recognition algorithm:

**Introduction:** In this homework, we propose to implement a face recognition algorithm from scratch, the solution should be able to detect the proper id of a given photo using a pretrained model, for the detection of the faces, third party libraries were used (Viola Jones and Deep learning algorithm), and for recognition, Eigen faces implementation from scratch was used.

After colleting the dataset of photos of members of the group, we proceeded with labeling each photo according to table

| name | id |
|---|---|
| Aydar Ahmetzyanov | 0 |
| Lyailya Aminova | 1 |
| Oleg Balakhnov | 2 |
| Dmitriy Desyatkin | 3 |
| Victor Massague Respall | 4 |
| Ahmed Nawaz | 5 |
| Maksim Rassabin | 6 |
| Adelya Sabirova | 7 |
| Sami Sellami | 8 |
| Arslan Siddique | 9 |
| Valeriya Skvortsova | 10 |

Figure 1: ids corresponding to each person

Then, we used the following algorithm for the face recognition training phase

**Training phase:**

```
Input: trainig image
Output: the projection weights corresponding to the eigen faces
1.  read the images
2.  detect the faces in the training images
3.  if there is faces detected
4.      extract the faces and resize the resulting images with the same dimensions
5.      computing mean face and the eigen faces
6.      Project the image into the face space and compute the corresponding weights
7.  else discard the image and go to the next image
8.  return weights after the projection into face space
```

Now lets dive deeper into each step of the implementation:

**step1-2: face detection:**

After reading the images, we used two methods for the face detection, fisrt is *Viola Jones algorithm*, OpenCV provide an easy function for that purpose namely *cv2.CascadeClassifier*, however this algorithm was not able to give good results when it comes to multiple face detection; there is no way to control which of the faces present in the image has to be extracted, this is why we used another solution which is a deep learning based method *cv2.dnn.readNetFromTensorflow* which give better results and allow to control the confidence parameters for the size of the face detected , figure  shows a face detected in an input image using the two method aforementioned

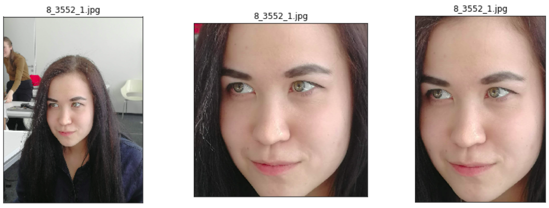

Figure 2: detection using Viola Jones(left) and deep learning(right)

We can see that Viola Jones gives squared faces whereas dnn gives rectangular faces closer to reality
Next figure (figure ) shows samples of the face detected in the dataset

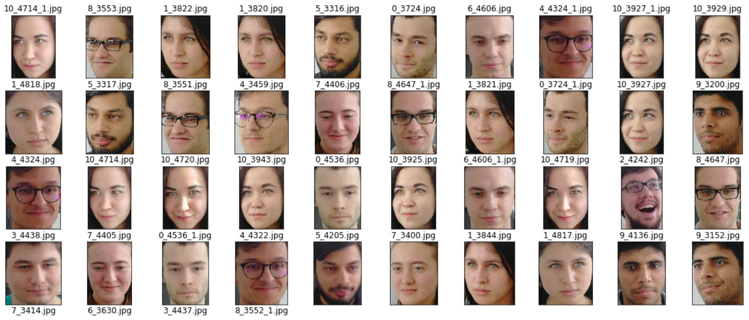

Figure 3: faces detection

**step 3-4: resizing face-images:**

In this step we performed resizing of the images with the same dimensions, this is a essential step in the eigen faces detection algorithm that we're gonna perform

**step 5-6: Projecion onto the face space :**

In those steps, we implemented the actual face recognition algorithm, first we represent each image as vector

$$\Gamma_i \in N^2 \quad \text{for image} \quad N \times N$$

then we calculate the covariance matrix $\quad C = AA^T \quad$ where $A = [\Phi_1\Phi_2....\Phi_M]$
The next step is to compute the best eigen vector of $AA^T$ : $u_i = Av_i$ and normalize them, figure shows representations of the mean face and some of the eigen faces

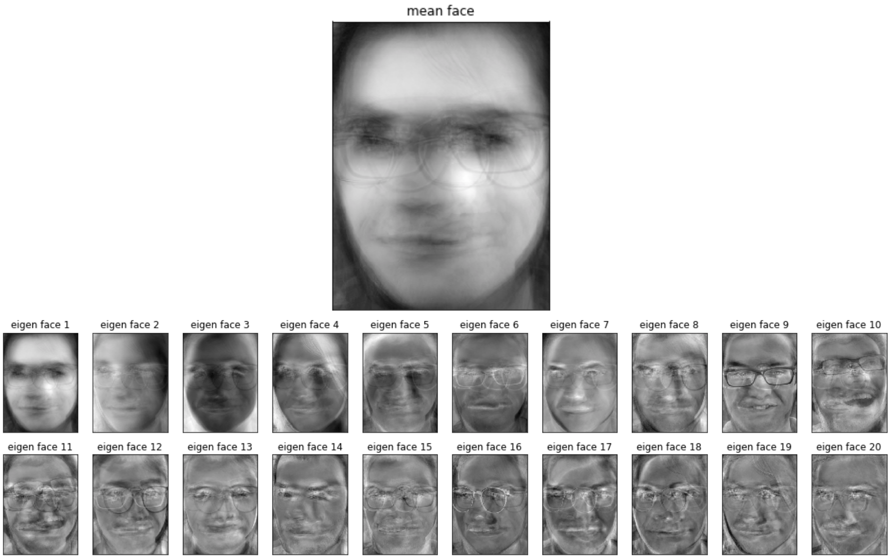

Figure 4: the mean face and the eigen faces

Figure and figure shows a face a its corresponding recontructed face using the eigen faces (projection)

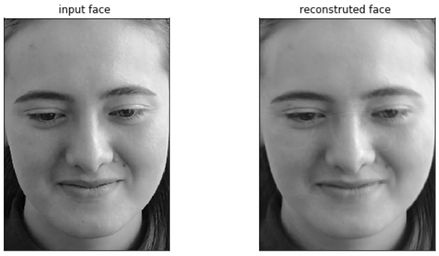

Figure 5: an original face image and its projection onto the face space

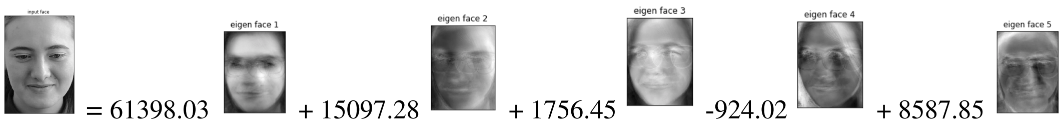

Figure 6: a recontructed face as the sum of its eigen faces

**step 7-8: returning the id:**

We discard the image if there is no precise face inside it, otherwise, we return the weights corresponding to the projection into face space

In this next part, we're going to detail the stages of the testing phase where we can properly test our face recognition algorithm:

**Testing phase:**

```
Input: test image
Output: the id of the person in the image
1.   read the images
2.   detect the face in image
3.   if there is face detected
4.       extract the face and resize it
5.       project the image into the face space and compute the corresponding weights
6.       compute the distances within face space
7.       label the input image as person id corresponding the face l for which the distance is munimum
8.   else discard the image
9.   return id of the person recognized
```

Now lets dive deeper into each step of the implementation:

**step 1-4: preprocessing:**

These steps are the same as in the training part, we read the image, detect the face and resize the image

**step 5-8: face recognition:**

After projecting the face into the face space, we compute the distance withiin the face space of the corresponding weights from each of the weights calculated in the training phase, a simple euclidean distance was used:

$$e_r = ||\Omega - \Omega^l||$$

where $\Gamma = [\omega_1 \quad \omega_2 ...... \quad \omega_k]^T$

After that, to know which face the weights correspond to, we take the k minimum distances and compute the occurences of each id corresponding to those distances, the final id of the input image is assigned to the most occurent id in the k distances, however k= 1 shows better results ie we take the minimum distance in the whole distances array Figure  shows an exemple of the recognition algorithm

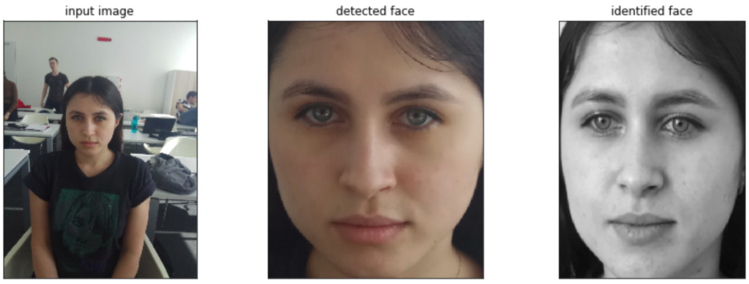

Figure 7: original image (left) detected face(middle) and the identified face (right)

**NB:** In the face rec function, one should find the following boolean variables:

- $using\_VJ$: equal 1 if we want use viola jones in the detection step
- $plot\_and\_print$: equal 1 if we want to plot the different stages of the recognition algorithm (detection and identification) and also print the distance within and from the space space

**step 9-11: returning the id:**

We discard the image where there is no face detected, label the face as unknown if the distance within the face space is too large, and return the person's id if everything went well

**Conclution:**

In this assignment, we were able to implement a face recognition algorithm based on the eigen faces representation proposed by *Matthew Terk and Alex Pentland* in 1991, we used Viola Jones and Deep learning based methods for the detecting the faces , and an algorithm implemented by scratch for the recognition, the latter showed reasonably good results in most of the cases, except for cases when the face present a relative orientation or when the conditions are drastically different from the training ones.