Report

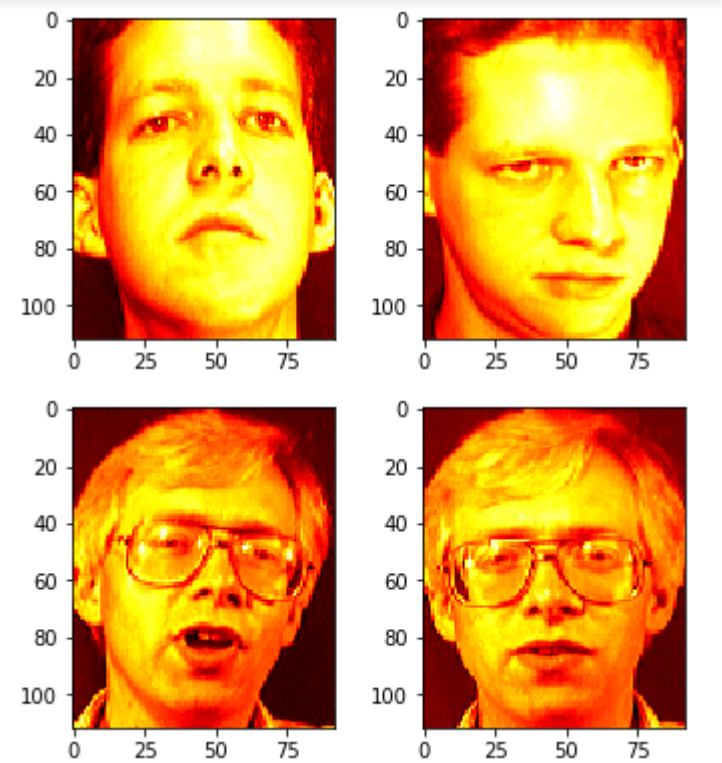
Face identification problem

Task overview

Find solution for image classification task for solving for face identification problem on ORL dataset. Dataset link here.

Given: 40 persons with 10 images of each. Total 400 images. Data specifications: format PGM, size 92 x 112 pixels, 256 colors.

For example



Algorithm concept

In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" straight-line distance between two points in Euclidean space. If we assume that the image is an n-dimensional vector in space, then the similarity of the two images can be compared with the Euclidean distance

Split input images on train/verification datasets. For example, train/verification percentage is 80/20 (320 train and 80 verification images). Then each image is vectorized. Vector dimension is 10 304 (92 x 112 x 1)

For each of the test face images [80], calculate the Euclid Distance for the images from the train set [320], resulting in a vector of 320 values. Then, find the index of the minimum value in this vector and by this index get the face-id from the vector Y (which will contain person indices) and compare it with the real id stored in the vector Y from 80 elements.

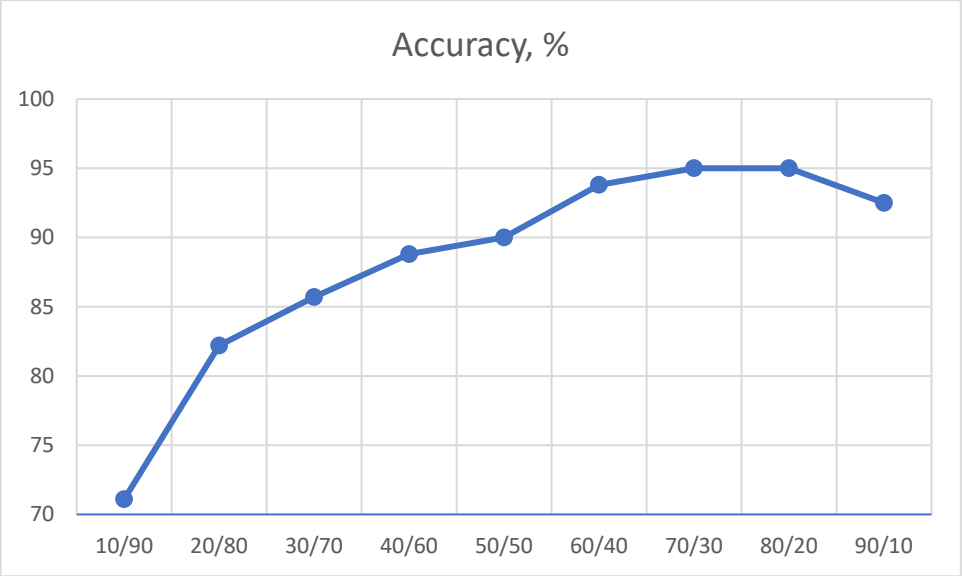
Algorithm steps

1. Define constants and model parameters (input data folder, train/verification percentage of sampling).
2. Load images in train/verification datasets.
3. Reshape image matrix to vector.
4. Calculate distances from one verification image vector to each train image vector.
5. Find minimal distance and select corresponding person id.
6. Repeat 4-5 steps for all image vector in verification dataset.
7. Compare predicted and actual persons id for each image in verification dataset.
8. Calculate model accuracy.

Results

Model accuracy for different variations train/verification dataset separation

Train/verification, %	10/90	20/80	30/70	40/60	50/50	60/40	70/30	80/20	90/10
Accuracy, %	71.1	82.2	85.7	88.8	90.0	93.8	95.0	95.0	92.5



Further calculations will be considered for optimally, 80/20%, train/verification dataset separation.

Optimization

As optimization was selected normalizations and algorithm PCA (Principal Component Analysis).

Normalization variation:

- Basic normalization (divide on max byte value, i.e. mathematical scaling data in range [0; 1])).
- Subtract mean (by feature or sample axis).
- Subtract mean divide std deviation (by feature or sample axis).
- Mathematical scaling data in range [-1; 1] (by feature or sample axis).

Model accuracy for different data normalization

Normalization	Accuracy, %
Without normalization	95.00
Basic normalization (scaling in range [0; 1])	95.00
Subtract mean (by features)	93.75
Subtract mean (by samples)	93.75
Subtract mean divide std (by features)	93.75
Subtract mean divide std (by samples)	93.75
Scaling in range [-1; 1] (by features)	**96.25**
Scaling in range [-1; 1] (by samples)	93.75

PCA (Principal Component Analysis).

The principal component analysis method (PCA) is one of the main ways to reduce data dimension by losing the least amount of information. The calculation of the principal components can be reduced to the calculation of the singular decomposition of the data matrix or to the calculation of the eigenvectors and eigenvalues of the covariance matrix of the initial data.

PCA method build for different values of pca energy parameter (0.95, 0.98, 0.99, and 1.00). After algorithm execution reduced dimensions of data:

```
(400, 10304) Original array size
(400, 190)   for pca energy 0.95
(400, 279)   for pca energy 0.98
(400, 325)   for pca energy 0.99
(400, 416)   for pca energy 1.00
Wall time: 7min 21s
```

Model accuracy for different data normalization and pca energy:

Accuracy, %				
Normalization	**PCA energy**			
	0.95	**0.98**	**0.99**	**1.00**
Subtract mean (by features)	92.50	93.75	95.00	93.75
Subtract mean (by samples)	92.50	93.75	95.00	93.75
Subtract mean divide std (by features)	93.75	93.75	93.75	93.75
Subtract mean divide std (by samples)	93.75	92.50	92.50	92.50
Scaling in range [-1; 1] (by features)	**95.00**	**95.00**	**96.25**	**96.25**
Scaling in range [-1; 1] (by samples)	93.75	93.75	93.75	93.75

The most time consuming operation is the decomposition of the covariance matrix. Time calculation of the eigenvectors and eigenvalues of the covariance matrix of the current dataset is more than 7 minutes. But these calculations are done once, and then using resulting PCA transition matrix for modification datasets before run face recognition model. Model execution time with PCA reduced dimensions decreases by about 10 times.

References

https://en.wikipedia.org/wiki/Standard_deviation
https://en.wikipedia.org/wiki/Euclidean_distance
https://en.wikipedia.org/wiki/Principal_component_analysis
http://seat.massey.ac.nz/personal/s.r.marsland/MLBook.html
http://seat.massey.ac.nz/personal/s.r.marsland/Code/Ch6/pca.py

2019-04-16
Ievgen Shumelchyk
Ievgen.shumelchyk@gmail.com