Eigenface Recognition with Deep PCA

Yulai Zhao, Xiaxin Shen

Motivation

- Principal Component Analysis (PCA) is a mathematical technique used for reducing the dimensionality of high-dimensional data.
- Eigenfaces are the eigenvectors of the covariance matrix that are computed using PCA on a set of face images.
- As we know, Deep PCA is useful for abstracting features of CNN which is a better choice than MLPs in the field of computer vision.
- Use Deep PCA in finding eigenfaces

Eigenface

- Focused on the methodology of Turk and Pentland's paper, Face recognition using eigenfaces [1].
- Implement the workflow suing basic algebra function of Numpy and PyTorch, including images preprocessing, eigenfaces construction, eigenspace representation of images, face recognition based on K-nn (K near neighbors) algorithm, performance evaluation
- Worked AT&T face dataset for performance evaluation
    - Formerly 'The ORL Database of Faces'
- Apply Deep PCA to each image and test the performance of the recognition.
- Save an image with 4 times information lost, we can still achieve comparable recognition correctness using the same algorithm (eigenfaces + Knn)

Reconstruction using DPCA

- *Reconstruction of non-diversified space* $\mathfrak{X}$: In this case, we treat DPCA as an end product. This corresponds to the scenario when the augmenting hidden layer (cf. Figure 6) happens to be the final layer and no more convolutions are anticipated. This calls for a typical LSE formulation to derive the best matrix $\mathbf{P} \in \Re^{C_{in} \times \kappa m}$ such that

$$\min_{\mathbf{P}} || \mathfrak{X} - \mathbf{P} [\mathbf{D}_k \mathfrak{Y}] ||_F. \tag{38}$$

Dataset

- AT&T "The Database of Faces" (ORL Database of Faces)
  - Contains 40 distinct subjects (10 different images of each)
  - Varying the lighting
  - Varying facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses)
  - Taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement)
  - The size of each image is 92x112 pixels, with 256 grey levels per pixel
- Experiments
  - Original Images
  - 4 times' compressed images
  - Recovered Images via DPCA after compressing

Preprocessing

- For every image, we compress 2x directly
- Reduce image by 4 times
- Save a weight based on Deep PCA
- Minimize $||Y - FUX||_F^2$ where U is a combination of eigenvectors
- Recover image: $W = FU$

Methods

- Load images and convert every of them into a matrix.
- Compute the mean face by averaging over all images.
- Compute the normalized images by subtracting mean face.
- Compute the Covariance Matrix S, which is different from the covariance matrix, in order to avoid huge matrix for doing eigen decomposition.
- Compute the eigenvalue and eigenvector. Then we have completed the initialization process of eigenfaces.

Results

Table 1: Probability of Correctness

	probability of correctness
original images	96.25%
4 times compressed images	88.75%
images compressed then recovered by deep PCA	89.375%

Results

results > att_restore_results.txt

```
464   result: wrong, got 22
465
466   image: ./datasets/att_faces_restore/s39/10.pgm
467   result: correct
468
469   image: ./datasets/att_faces_restore/s40/2.pgm
470   result: correct
471
472   image: ./datasets/att_faces_restore/s40/3.pgm
473   result: correct
474
475   image: ./datasets/att_faces_restore/s40/6.pgm
476   result: correct
477
478   image: ./datasets/att_faces_restore/s40/10.pgm
479   result: correct
480
481   Correct: 89.38
482
```

```
Image: 1_1
psnr_ori_comp: (27.433550410291694+0j)
psnr_ori_rest: (27.433500155362623+0j)
psnr_rest_comp: (67.61628221656727+0j)

Image: 1_2
psnr_ori_comp: (27.44599922276737+0j)
psnr_ori_rest: (27.450670886353702+0j)
psnr_rest_comp: (67.65388370530034+0j)

Image: 1_3
psnr_ori_comp: (27.51373603889842+0j)
psnr_ori_rest: (27.51117355236549+0j)
psnr_rest_comp: (67.96702433198436+0j)

Image: 1_4
psnr_ori_comp: (27.85496527988096+0j)
psnr_ori_rest: (27.857829961891618+0j)
psnr_rest_comp: (67.1214285857681+0j)
```

References

1] Turk, Matthew A., and Alex P. Pentland. "Face recognition using eigenfaces." Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on. IEEE, 1991.

[2] Belhumeur, Peter N., Joao Pedo Hespanha, and David J. Kriegman. "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection." IEEE Transactions on pattern analysis and machine intelligence 19.7 (1997): 711-720.

[3] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[4] https://github.com/zwChan/Face-recognition-using-eigenfaces