Eigenface++: Face Recognition with Deep PCA

Yulai Zhao
Princeton University
yulaiz@princeton.edu

Xiaxin Shen
Princeton University
xiaxin.shen@princeton.edu

Abstract

Convolution serves as a critical part of modern deep learning. Though many convolutional models showed surprising effects in various tasks, the computation and storage consumption of these models remain a main drawback. In this work, we aim to employ convolution in the classical principal component analysis (PCA) efficiently. Specifically, we extend the pioneering work of eigenfaces [4] and propose Eigenface++ that introduces convolution to PCA without any back-propagations. Our Eigenface++ includes two phases: (1) deep-LSE (DLSE) and (2) deep-PCA (DPCA), where stage (1) serves as a preparation stage for the latter as a milestone in our project's deliverables. Thanks to the design of efficient convolutional PCA, we present competitive recognition accuracy on human faces even if all face data is 4 times compressed. The immense reduction in storage makes this work significant and may shed light on future studies of efficient machine learning. Implementation details can be found at: https://github.com/zhaoyl18/Deep-PCA/.

1 Introduction

Recently, there has been a proliferation of work in the field of deep learning that leverages backpropagation. With the emergence of large models containing a significant number of parameters, ensuring efficiency has become a pressing challenge. To address this challenge, our work seeks to explore whether we can still build on the power of convolution to build computationally and storage-efficient models, even without relying on backpropagation, which is commonly used in deep learning. Some promising solutions to address this question are proposed like deep-LSE (DLSE) and deep-PCA (DPCA) [3].

Eigenface [4] is introduced using traditional PCA and KNN which minimize the Euclid Distance for face recognition task, and eigenface is another name for the principle components of the initial training set of face images. While face recognition is a well-established problem, the existing solutions based on backpropagation lack both explainability and efficiency. To overcome these challenges, our project proposes Eigenface++, a framework that extends the original Eigenface [4] by employing deep-PCA [3] for the task of face recognition.

2 Methodology

In terms of choosing deep-PCA as our methodology, the convolution operations can bring diversity to the input, which may lead to better learning feature maps with this kind of diversity. For PCA, compared with neural networks using backpropagation, PCA stands out for its beautiful explainability and efficiency. The idea of decomposing the matrix to lower the rank is promising and can be a perfect fit for the work of heavy-linear algebra. The use cases of lowering rank by matrix decomposition deserve exploring with the goal of improving efficiency.

One point worth noting is that the deep PCA we implemented in this project is a customized DPCA, different from the original paper [3] where it was proposed.

For the original DPCA proposed in paper [3], for the encoder part, all the ground-truth images X are expanded ω^2 times from the original size. The expanded matrix $D_\omega(X)$ is the STEM matrix after the convolution operation, which is the input. The output of the encoder is the STEM matrix's components Y. For the decoder, the input is the STEM matrix $D_{\omega'}(Y)$ of those components which are expanded ω'^2 times, and the output is the ground-truth images X (original dataset).

For our project, instead of applying DPCA once for just one matrix comprising the whole dataset, we apply it one image by one image. With our dataset including 400 face images, for each ground truth image $x_{teacher}$, we first downsample $1/4$ times to form the compressed image $x_{compress}$. In order to restore this image, we need to first form the input. We upsample the compressed image 4 times back to the original size of the original image, which results in the upsampled image x. Then, we use the STEM matrix $D_3(x)$ of the upsampled image as the input, which is $\omega'^2 = 3^2 = 9$ times of the size of $x_{teacher}$. Then we extract $m = 1$ components to form U from this diversified STEM matrix $D_3(x')$. After knowing U, we can get F. Finally, $W = FU$, and w is $m \times m$'s matrix, namely 1×1 in our experiment. Here, we minimize

$$\|x_{teacher} - FUD_{\omega'}(x')\|_F^2$$

which is the customized decoder phase of the DPCA.

With W and compressed image $x_{compress}$, assuming the original image's size is $H \times W$. The image can be stored using $1/4 \times H \times W + 1 \times 1$ pixels. When the image needs to be used, it can be restored by WX, which is storage efficient.

The prior work deep-LSE before our customized deep-PCA come with similar motivation. In the scenario of transferring the image between the Earth and Mars. We would like to only transfer the compressed image and the W. It would be expected that the compressed image and the W can construct the image with sufficient quality. The difference between deep-LSE and our personalized deep-PCA is that we did not use F and U, but use W directly for deep-LSE. U is m components extracted via PCA, used in deep-PCA.

3 Experiments

In this section, we specify the experimental settings. In this work, we utilized the well-known AT&T human faces dataset [1] (formerly the ORL Database of Faces). There are 40 distinct person images in total. For each person, there are 10 images. These 10 images were taken considering light varying, and face expressions with a focus on open or closed eyes, and whether smiling. Wearing glasses also is considered a factor in facial details. The dataset was collected in a dark homogeneous background. People are in an upright and facing front but with side movement.

3.1 Preprocessing

For every image, we compress 4x directly. Then we save an up-sampling version to the `*dataset/att_faces_compress*` folder. It is true that directly reducing the image by 4 times would lead to a lot of information being lost. However, we save a weight based on Deep-PCA, which could reconstruct the image. During this process, we minimize

$$\|X - FUY\|_F^2$$

where U is a combination of eigenvectors, Y is a diversified compressed image, and X is the teacher image (the ground truth image from the dataset). We set kernel size $w = 3$ and the number of components $m = 1$. Finally, the weight used to recover the image is $W = FU$. For every compressed image, we apply convolution kernel W and save the restored image to `datasets.att_faces_restore`

After preparing three datasets, including original, compressed, and restored images, we implemented PCA on three datasets. The specific steps are as follows:

1. For each of the three datasets. Load images and convert each of them into a matrix.
2. Compute the mean face by averaging over all images.
3. Compute the normalized images by subtracting mean face.

4. Compute the covariance Matrix S, which is different from the covariance matrix, in order to avoid performing the eigendecomposition of a huge matrix.
5. Compute the eigenvalue and eigenvector. Then we have completed the initialization process of eigenfaces.

Note that the number of components here is calculated based on `energy`. We set `energy` as 85%. We estimated the number of components to be around 50 60.

3.2 Recognition

The application of eigenfaces implies that we consider each image as the linear combination of the eigenfaces (by projecting onto the eigenspace). The weights of the corresponding eigenfaces therefore represented the image itself. We only used the top n eigenfaces to represent an image, where the n was determined by how much variance this sub-eigenspace can represent. We investigated 85% percent of variance for each dataset. To recognize an unknown face, we used the KNN algorithm to find the close subject in the database. The steps include:

- For each image in a dataset, we considered it as a query image and the other images in the dataset as training data.
- We got the nearest K neighbors and let them vote to determine the label of the query image. Whenever there was a tie, we used the label with the least average distance.
- If the prediction label is the same as the ground label, it is a true positive; otherwise, it is a false positive. We calculated the precision as the performance of the recognition of the result.

4 Results

Here we introduce the experimental findings and make discussions. To reproduce the results in this section, we refer interested readers to *the submission version* [1] for codes and instructions.

4.1 Deep-LSE

Figure 4.1 shows the original image and the reconstruced images with different kernel size. we got from deep-LSE. We utilized bicubic as the interpolation for both downsampling and upsampling. When kernel $w = 2$, the psnr is 25.82. When kernel $w = 3$, the psnr is 19.61. $w = 2$ has higher psnr than $w = 3$.

Table 1: PSNR with different kernel size

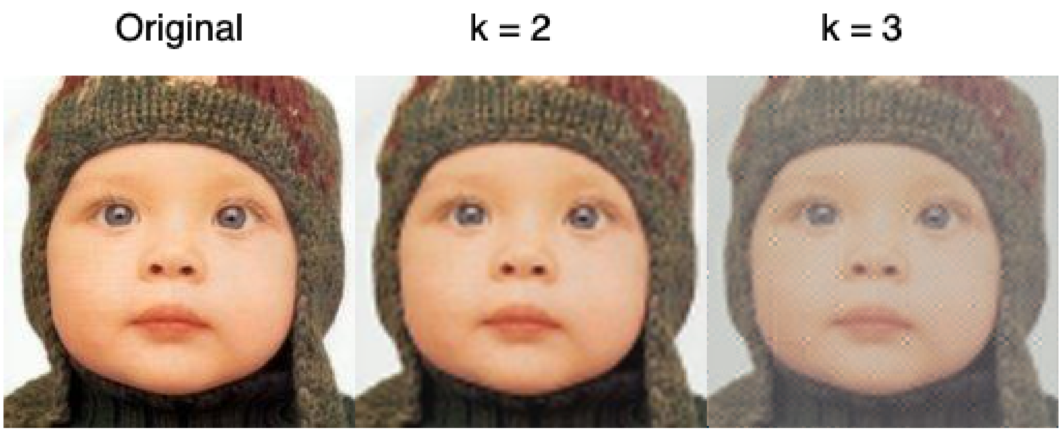
	psnr
kernel = 2	25.82
kernel = 3	19.61

Table 2: Recognition Accuracy

	accuracy
original images	96.25%
4 times compressed images	95.00%
images compressed then recovered by deep PCA	95.63%

[1] https://github.com/zhaoyl18/Deep-PCA/tree/571_submission

Figure 1: Comparison Between Reconstructed Images and Original Images



4.2 Deep-PCA

Figure 4.1 shows an example of the original (or raw) image, the compressed image, and the restored image via our personalized PCA.

Figure 2: Reconstructed Image with kernel = 3

Table 2 shows that we got 96.25% correctness for original images, 95% for 4 times compressed images, and 95.63% for the restored images via deep-PCA. From 95% to 95.63%, we achieved a direct improvement of 0.63 without heavy computation overhead or additional burden.

5 Discussion and Conclusion

The eigenfaces are one of the most popular approaches to representing an image, with the basic idea that the top k component eigenvectors (eigenfaces) represent as much variance as possible. This criterion need not be meaningful. It is also susceptible to illumination and background around the face. Fisherfaces [2] is widely considered a better representation than eigenfaces since it is more robust to illumination. But both of them do not contain semantic meanings for humans to understand a facial image. A possible further study is the deep neural network approach that produces state-of-the-art performance by now, in terms of further researching the face recognition field.

In conclusion, this project proposes Eigenface++ by revolutionizing PCA to deep-PCA. Our experiments demonstrate that Eigenface++ truly facilitate the storage and computation efficiency.

References

[1] AT&T database of faces. https://www.kaggle.com/datasets/kasikrit/att-database-of-faces. Accessed: 2023-05-22.

[2] Peter N. Belhumeur, Joao P Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

[3] S. Y. Kung. XNAS: A Regressive/Progressive NAS for Deep Learning. *ACM Transactions on Sensor Networks*, 18(4):1–32, 2022.

[4] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*, pages 586–587. IEEE Computer Society, 1991.