The Application of PCA: Facial Recognition

Dai Ruihao
1024041106
Nanjing University of Posts and Telecommunications
School of Computer Science
Nanjing, China

***Abstract*—Facial recognition technology, as a pivotal component of biometric identification, has gained significant attention in computer vision and artificial intelligence. This paper presents a comprehensive overview of a facial recognition algorithm based on Principal Component Analysis (PCA). The algorithm efficiently classifies and recognizes facial images by reducing data dimensionality, extracting key features, and constructing facial space. The document explores the steps and key concepts of the PCA algorithm, including matrix representation, data vectorization, and the construction of facial space. The application of the algorithm in image classification, encompassing image projection, distance calculation, and classification decision-making, is detailed.**

***Index Terms*—Facial Recognition, Principal Component Analysis, Image Classification, Image Processing**

I. Introduction

The rapid development of digital image processing and pattern recognition has positioned facial recognition technology [1] as a prominent biometric identification method. Leveraging the distinct features of human faces, this technology finds widespread applications in security, identity verification, and intelligent systems [2]. This paper delves into a facial recognition algorithm based on Principal Component Analysis (PCA), which achieves efficient classification and recognition of facial images by employing matrix representation, vectorization, and facial space construction.

As informatization and digitization progress, the demand for precise and efficient facial recognition solutions is on the rise [3]. Conventional pixel-matching approaches face significant challenges in handling complex scenarios and large datasets. To address these limitations, the PCA-based [4] facial recognition algorithm offers a compelling alternative, balancing computational efficiency and recognition accuracy by reducing data dimensionality, extracting essential features, and constructing a representative facial space.

The structure of this paper is as follows: Section II details the core principles and steps of the PCA algorithm, including image matrix representation, data vectorization, and facial space construction. Section III explores the application of PCA in image classification, focusing on image projection, distance metrics, and classification strategies. Section IV examines the influence of parameter settings on algorithm performance and provides experimental validation. Finally, Section V and **??** concludes with a summary of key findings and suggests future directions for PCA-based facial recognition research.

II. Related work

S. Tyagi and K. Kundu [5] implemented a combination of k-means clustering with PCA for clustering similar images. Their approach achieved a recognition rate of 90%, which is slightly lower than the 93.33% recognition rate achieved by conventional PCA. Z. Fan et al. [6] introduced a modified PCA algorithm by integrating multiple similarity subspace models. This enhanced method demonstrated superior classification accuracy and clustering performance compared to conventional PCA. Furthermore, the modified PCA proved applicable to image reconstruction tasks. A. Alorf [7] conducted a performance evaluation of standard PCA and an improved PCA variant based on Shannon information theory. Their study revealed that the improved PCA exhibited faster processing speeds in image compression but fell short of the standard PCA in detection and recognition efficiency. S. Agarwal et al. [8] presented a case study utilizing PCA with OpenCV for concurrent multi-face detection. Manual face localization using PCA achieved a 100% recognition rate on the ORL database, while Illumination Adaptive LDA (IALDA) yielded a 98.9% recognition rate on the CMU PIE database. A. K. Bansal and P. Chawla [9] compared PCA with normalized PCA (NPCA) across the ORL, Indian Face Database (IFD), and Georgia Tech Faces (GT-Faces) datasets. Their findings indicated that NPCA consistently outperformed PCA. They also observed a positive correlation between the number of training images and recognition efficiency.

III. PCA Steps and Key Concepts

The Principal Component Analysis (PCA) algorithm is fundamental for effectively analyzing and representing facial images. It involves key processes such as image matrix representation, data vectorization, and the creation of a facial space.

A. Matrix Representation of Images

At the outset, facial images are represented in matrix form to facilitate various transformations and analyses. Grayscale images are encoded as matrices where each element corresponds to the pixel intensity at a specific position. The general matrix representation is expressed as:

$$I = \begin{bmatrix} I_{1,1} & \dots & I_{1,m} \\ \vdots & \ddots & \vdots \\ I_{n,1} & \dots & I_{n,m} \end{bmatrix} \quad (1)$$

In this representation, $I_{i,j}$ refers to the intensity of the pixel located at row i and column j in the image matrix I. This structured format serves as a foundation for subsequent image processing operations.

B. Data Vectorization

To facilitate efficient computation and extraction of features, the algorithm transforms the image matrices into column vectors. This process involves reshaping each image matrix into a single vector using the following representation:

$$\eta = \begin{pmatrix} \phi_{1,1} \\ \vdots \\ \phi_{n,m} \end{pmatrix} \in \mathbb{R}^{mn \times 1} = \mathbb{R}^p \tag{2}$$

Here, $\phi_{i,j}$ represents the pixel intensity at row i and column j of the image, and p is the dimensionality of the resulting vector space. This transformation ensures that all training images are uniformly represented in vector form.

C. Construction of Facial Space

Next, the algorithm centers the training images by subtracting the mean pixel intensity, creating a dataset matrix $\mathbf{\Psi}$. Each centered image vector $\boldsymbol{\phi}_c$ is derived by removing the mean intensity value, $\psi_{i,j}$, from the original pixel intensities. The centered dataset matrix is expressed as:

$$\mathbf{\Psi} = (\boldsymbol{\phi}_{c1}, \ldots, \boldsymbol{\phi}_{ck}) : \boldsymbol{\phi}_{ci} \in \mathbb{R}^p \tag{3}$$

This matrix organizes the centered image vectors column-wise, serving as a foundational element for constructing the facial space.

In conclusion, this section elaborates on the critical steps of the PCA algorithm, including image matrix representation, data vectorization, and the construction of the facial space. These foundational processes are integral to the algorithm's capacity for effective facial image classification.

IV. Algorithm Application

This section focuses on the practical implementation of the PCA-based facial recognition algorithm in the image classification process, covering key components such as dataset preparation, image projection, distance calculation, and classification decision-making.

A. Dataset Preparation

To utilize the training image set effectively, images must be centered around the mean and arranged into a matrix format, as illustrated in Figures 1 and 2.

Fig. 1. Example of an image centered around the mean

Fig. 2. Example of the mean value of the images

Principal Component Analysis (PCA) is a widely employed technique for dimensionality reduction and feature extraction in image processing. The process of centering a dataset involves the following steps: 1. Compute the mean image vector ($\bar{x}$) by averaging all image vectors in the dataset. 2. Subtract $\bar{x}$ from each image vector to create a set of centered image vectors (x'_i). 3. Organize the centered vectors into a data matrix (X') where each column represents a centered image vector. 4. Compute the covariance matrix (C) of X' and perform eigenvalue decomposition to obtain the eigenvectors (Q) and eigenvalues (Λ). 5. Select the top k eigenvectors from Q to form the principal components, resulting in a reduced-dimensional representation that captures essential variations in the dataset.

This preparation is fundamental for subsequent applications of PCA, such as facial recognition or image compression.

B. Distance Calculation

Once the images are projected onto the facial space, the next step involves calculating the distance between the projected image vector and the mean vector of each class. This step is critical for determining the similarity between the projected image and the class mean.

The Euclidean distance (ϵ) is used to quantify this similarity and is calculated as:

$$\epsilon_i = \|\Omega - \Omega_i\| \tag{4}$$

Where ϵ_i represents the Euclidean distance for class i, Ω is the projected image vector, and Ω_i is the mean vector of class i.

In this implementation, a function named `calculate-Euclidean-Distance` encapsulates the computation of ϵ_i, facilitating the measurement of similarity between the projected image and the mean vector of each class.

Algorithm 1 calculate-Euclidean-Distance

Require: $projectedImage, classAverage$
Ensure: $euclideanDistance$
$euclideanDistance \leftarrow norm(projectedImage - classAverage)$

Following the distance calculation, the image is classified by comparing its Euclidean distance to the mean vectors of each class. This classification is implemented using the `classify-Image` function, which iterates over all class averages and identifies the class with the minimum distance as the predicted result. If the minimum distance is below a predefined threshold, the corresponding class is returned. Otherwise, the image is labeled as "unknown."

C. Classification Decision-Making

The final stage in the classification process involves making informed decisions based on the computed distances. These decisions are influenced by predefined thresholds and rely on a systematic evaluation of the distances. The following variables are used:

- ϵ_i: The Euclidean distance between the projected image and the mean vector of class i.
- θ: The overall distance between the image and its projection onto the face space.

The classification decision is governed by the following rules:

- **Case 1:** If $\epsilon_i < \tau_1$ and $\theta < \tau_2$, the image is classified as belonging to class i.
- **Case 2:** If $\epsilon_i > \tau_1$ and $\theta < \tau_2$, the classification is considered undetermined. Additional analysis or refinement is required for confident classification.
- **Case 3:** If $\epsilon_i < \tau_1$ and $\theta > \tau_2$, the classification is also considered undetermined, necessitating further information or refinement.
- **Case 4:** If $\epsilon_i > \tau_1$ and $\theta > \tau_2$, the image is determined not to contain a face. The distances exceed predefined thresholds, indicating that the image does not belong to any trained face class.

Here, τ_1 and τ_2 represent thresholds for distance and projection, respectively. These thresholds play a critical role in fine-tuning the classification process, balancing sensitivity and specificity.

This framework ensures meaningful and accurate classification by considering both the specific distance (ϵ_i) to each class and the overall distance (θ) in the face space.

In subsequent sections, we explore the influence of τ_1 and τ_2 on the algorithm's performance. Experimental results are presented to validate the choice of these threshold values, demonstrating their effectiveness in achieving reliable facial recognition outcomes.

This comprehensive process ensures effective and accurate classification of facial images utilizing the PCA-based facial recognition algorithm. The combination of image projection, distance calculation, and classification decision-making contributes to the robustness of the algorithm in various application scenarios.

Fig. 3. Example of an image projected onto the facial space

D. Image Classification

In the image classification pipeline, the first pivotal step involves projecting new facial images onto the previously constructed facial space. This projection is achieved by transforming the centered image into the principal components space using the Singular Value Decomposition (SVD) method. Mathematically, the projection is represented as:

$$\Omega = U_0 \cdot (\eta - \psi) \tag{5}$$

Where:

- Ω represents the projected image in the facial space,
- U_0 denotes the matrix of principal components,
- η is the centered image, and
- ψ represents the mean image.

This mathematical representation encapsulates the essence of mapping facial images onto a space characterized by the primary modes of variation within the dataset. The principal components, encapsulated in the matrix U_0, act as the basis functions defining the most significant features in the facial dataset. The centered image η is transformed by subtracting the mean image ψ, ensuring that the projection captures deviations from the average facial structure, as illustrated in Figure 3.

In practical terms, this process enables the model to discern the unique characteristics of each facial image by expressing it as a combination of the principal components. The resulting projected image, Ω, is a representation in the facial space that emphasizes the distinctive features contributing to facial identity.

This step lays the foundation for subsequent stages in the facial recognition pipeline, enabling the classification algorithm to effectively discriminate and identify individuals based on their facial features.

V. Experimental Results

In this section, we investigate the crucial aspect of how parameter adjustments influence the performance of the PCA-based facial recognition algorithm. The selected parameters, namely the number of principal components (k), projection threshold, and classification threshold, play a pivotal role in determining the algorithm's success rate. Through systematic experimentation, we aim to provide insights into the optimal parameter configurations for achieving enhanced algorithmic efficacy.

A. Application Usage

Necessary Utilities and Modules:

- Python 3.8
- Matplotlib 3.4.1
- Numpy 1.20.2

To run the application, modify the values of the variables `train-path` and `test-path` in the `main.py` module. The `train-path` variable should point to the absolute path of the training samples file, while the `test-path` variable should point to the test samples file. To execute the application, use the following command in a terminal opened in the directory containing the `main.py` file:

python ./main.py

B. Experimental Setup

The dataset was divided into two subsets:

- **Training Set:** Used to train the PCA model.
- **Testing Set:** Reserved for evaluating the model's performance.

This partitioning ensures a clear evaluation of the algorithm's ability to generalize to unseen data. Experiments were conducted using a diverse dataset of facial images with consistent formats, enabling rigorous testing of the algorithm's robustness and adaptability.

C. Impact of Principal Components (k)

Recognition accuracy is a fundamental measure of the algorithm's effectiveness. It is computed as the percentage of correctly identified individuals from the test dataset. The following results summarize the impact of varying k on recognition accuracy, computation time, and dimensionality reduction:

- $k = 10$**:** Projection threshold = 20000, Class threshold = 20000, Achieved success rate: **93.33%**
- $k = 1$**:** Projection threshold = 20000, Class threshold = 20000, Achieved success rate: **40.00%**
- $k = 10$**:** Projection threshold = 10000, Class threshold = 20000, Achieved success rate: **86.67%**
- $k = 4$**:** Projection threshold = 10000, Class threshold = 20000, Achieved success rate: **33.33%**

D. Conclusions

The experiments demonstrate that the number of principal components (k) significantly impacts the PCA-based facial recognition algorithm. Higher values of k generally result in better recognition accuracy, although with a longer computation time. Conversely, smaller values of k lead to faster computation but may compromise accuracy.

Through systematic experimentation, we gain valuable insight into the trade-offs between recognition accuracy, computational efficiency, and dimensionality reduction. This understanding is instrumental in fine-tuning the algorithm for practical applications, ensuring both robustness and adaptability in real-world scenarios.

E. Application Output

For the dataset included in the archive and the parameters: $k = 10$, projection threshold = 20000, and class threshold = 20,000, the output of the application is summarized in Table I.

Explanation of Table Notation:

- **Pred: X** represents the predicted class to which the algorithm believes the current image belongs (X denotes the class name; e.g., 12, 10, 09, 14, etc.).
- **Actual: X** represents the actual class to which the image belongs.

If the value of `Pred:` matches the value of `Actual:`, the algorithm has correctly classified the image.

TABLE I
APPLICATION OUTPUT

Pred:	01	Actual:	01	Pred:	02	Actual:	02
Pred:	03	Actual:	03	Pred:	04	Actual:	04
Pred:	05	Actual:	05	Pred:	06	Actual:	06
Pred:	07	Actual:	07	Pred:	08	Actual:	08
Pred:	09	Actual:	09	Pred:	10	Actual:	10
Pred:	11	Actual:	11	Pred:	12	Actual:	12
Pred:	13	Actual:	13	Pred:	14	Actual:	14
Pred:	07	Actual:	15				

VI. SUMMARY

Although PCA-based facial recognition is relatively simple, it delivers satisfactory results when clear facial images are used following a specific format. However, certain limitations exist, particularly under challenging conditions, such as obstructions or unclear images.

A. Key Observations

- **Parameter Optimization:** The selection of optimal parameters, such as the number of principal components (k) and thresholds, requires iterative experimentation and analysis of success rates.
- **Numerical Stability:** Due to operations involving small values, the algorithm can exhibit numerical instability, which affects the reliability of the results.

In conclusion, while the PCA-based algorithm serves as a robust starting point for facial recognition, future iterations could focus on addressing these challenges to enhance performance in real-world scenarios.

REFERENCES

[1] D. Georgescu, "A Real-Time Face Recognition System using Eigenfaces," Journal of Mobile, Embedded and Distributed Systems, vol. 3, no. 4, pp. 193–204, 2011.

[2] M. U. Rahman, "A Comparative Study on Face Recognition Techniques and Neural Network," arXiv preprint arXiv:1210.1916, 2012.

[3] D. N. Parmar and B. B. Mehta, “Face Recognition Methods & Applications,” arXiv preprint arXiv:1403.0485, 2014.

[4] C. S. Chang, T. L. Liao, P. Y. Hsu, and K. K. Chen, “Human Face Recognition System using Modified PCA Algorithm and ARM Platform,” in 2010 International Symposium on Computer, Communication, Control and Automation (3CA), vol. 2, May 2010, pp. 294–297.

[5] S. Tyagi and K. Kundu, “Face Recognition using PCA and RBF Neural Network,” International Journal of Computer Science and Information Technology & Security (IJCSITS), vol. 6, 2016.

[6] Z. Fan, Y. Xu, W. Zuo, J. Yang, J. Tang, Z. Lai, and D. Zhang, “Modified Principal Component Analysis: An Integration of Multiple Similarity Subspace Models,” IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 8, pp. 1538–1552, Aug 2014.

[7] A. A. Alorf, “Performance Evaluation of the PCA versus improved PCA (IPCA) in Image Compression, and in Face Detection and Recognition,” in 2016 Future Technologies Conference (FTC), Dec 2016, pp. 537–546.

[8] S. Agarwal, P. Ranjan, and A. Ujlayan, “Comparative Analysis of Dimensionality Reduction Algorithms, Case study: PCA,” in 2017 11th International Conference on Intelligent Systems and Control (ISCO), Jan 2017, pp. 255–259.

[9] A. K. Bansal and P. Chawla, “Performance Evaluation of Face Recognition using PCA and N-PCA,” International Journal of Computer Applications, vol. 76, no. 8, 2013