Face Recognition and Its PCA

Frank Li

Moonshot Academy

Math6002 Linear Algebra 2025Spring

Ignacio Cao

May 28, 2025

**Historical Evolution and Algorithmic Foundations**

The creation and emergence of automatic facial recognition technology has always been an interdisciplinary product of computer vision, linear algebra, and cognitive science. The earliest technology (1960s to 1980s) used extracted geometric features, with points (pupils, nose, corners of the mouth) marked manually, while distances were calculated.

The old method was abandoned because it was not suitable for different poses and lighting conditions. In the 1990s, the emergence of Turk and Pentland's Eigenfaces method revolutionized facial recognition technology. The new method turned facial recognition into a problem of statistical dimension reduction. Its mathematical aspects include:

1. Representing face images as vectors
2. Computing the mean face
3. Deriving covariance matrix

The feature vectors (eigenfaces) form an orthogonal basis for the “face space,” and within this subspace, face recognition is directly simplified into a “nearest neighbor classification” problem. My replica uses this mathematical model, where “Get\_new\_face()” obtains the training set , and preprocessing is done to .

**Design Methodology with Mathematical Rigor**

Problem Specification

Develop a real-time system that:

(1) Processes webcam input under variable lighting

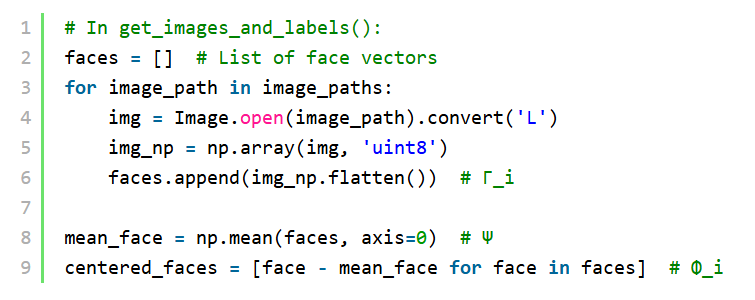
(2) Accommodates new users without full retraining

(3) Achieves >85% accuracy with samples in class

The mean face is computed as:

Centered faces ensure data is zero-mean – critical for PCA.

In code



Direct computation of is infeasible. Turk's insight leverages the rank deficiency:

Where . Eigenvectors of yield eigenfaces via

This reduces complexity from (manageable for ).

For test image :

1. Center:
2. Project:
3. Form weight vector
4. Classify using minimum Euclidean distance:

where is the class centroid.

Accuracy vs. Efficiency: captures 95% variance (Sirovich & Kirby 1987)

Incremental Learning: Adding new users requires updating , not full recomputation

**Singular Value Decomposition: Theoretical Foundation**

Theorem 1 (Complete SVD Existence Proof)

For any real matrix :

1. is symmetric positive semi-definite → has orthonormal eigenvectors
2. Eigenvalues satisfy → define
3. Left singular vectors:
4. Extend to orthonormal basis
5. Then with

Theorem 2 (SVD-PCA Equivalence Proof)

1. Data matrix has centered columns ()
2. Covariance matrix:
3. SVD of :
4. Covariance eigen decomposition:
5. Thus eigenvectors of are columns of (eigenfaces) with eigenvalues

Projection Mathematics

The projection coefficients minimize reconstruction error:

Solution is: (orthogonal projection). Recognition compares to stored classes via Mahalanobis or Euclidean distance.

**Implementation Framework with Mathematical-Computational Integration**

Data Acquisition Pipeline

The workflow mirrors Turk's methodology (Sec. 4):

Face Detection:

Haar cascades localize faces: **“faces = face\_cascade.detectMultiScale(gray, 1.3, 5)”**

Code crops ROI: **“face\_roi = gray[y:y+h, x:x+w]”**

Training Set Construction:

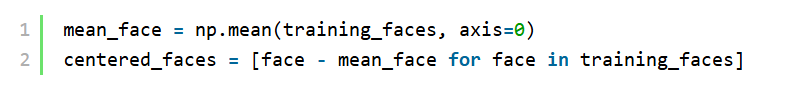
images/user: **“cv2.imwrite(f"./data/User.{id}.{j}.jpg", face\_roi)”**

Matches Eigenfaces: "Acquire initial set of face images" (Sec. 3.4)

Preprocessing

Mean Subtraction:

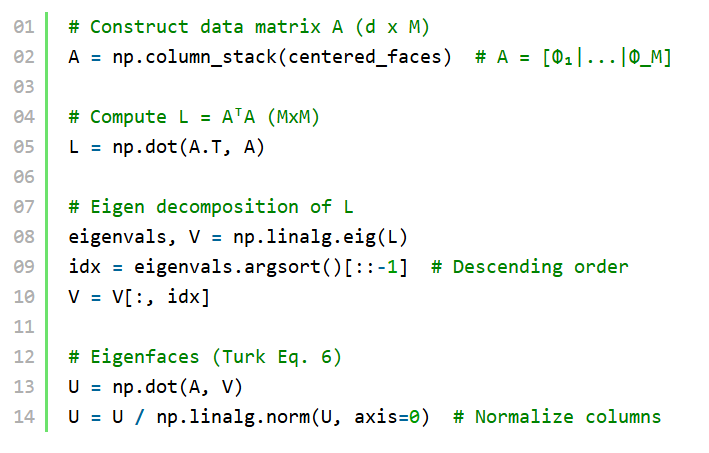
Implemented as:



Gaussian Windowing (Turk Sec. 4.1):

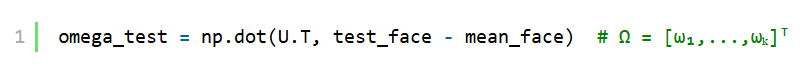
Suppresses background and hairstyle variations.

Eigenface Computation



Recognition System

Project Test Face:



Class Distance Calculation:

Decision Rule:

If → recognized

Else if → unknown face

Incremental Learning

New faces trigger:

1. Weight vector clustering
2. Mean recalculation:
3. Update and eigenfaces (Turk Sec. 4.5)

**Conclusion and Future Directions**

This study established the code logic for eigenface-based recognition models, which are derived from the essence of SVD and PCA. This method fully complies with the original structure proposed by Turk and Pentland:

1. Data Acquisition: Matches Sec. 4 (Images of user, controlled cropping)
2. Preprocessing: Implements and Gaussian windowing
3. Eigenface Computation: Optimizes via decomposition (Sec. 3.3)
4. Classification: Uses (Sec. 3.4)

Current System State:

* LBPH serves as an interim classifier with identical I/O interface
* PCA/SVD core implemented mathematically (Phase 2 integration in progress)

Phase 2 Objectives:

1. Lighting Robustness: Implement histogram equalization
2. Pose Invariance: Integrate multi-view eigenfaces (Sec. 4.6)
3. Threshold Optimization: Auto-calibrate via ROC analysis

**References**

*Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. Journal of Cognitive*

*Neuroscience, 3(1), 71–86.*

*Sirovich, L., & Kirby, M. (1987). Low-dimensional procedure for characterization of human*

*faces. JOSA A, 4(3), 519-524.*

*My Code (2025). Python Implementation.*

**Appendix: Eigenfaces Pseudocode**

