Case Study: Face Recognition using PCA (Eigenfaces Method)

A university wants to build a face recognition system for secure access to a laboratory. The system uses Principal Component Analysis (PCA), also known as the Eigenfaces method, to recognize faces efficiently.
Each face image is a grayscale image of size 64 × 64 pixels.
In this session we will discuss this case study in details.
Session-4: Eigen values and Eigen vectors_
Introduction
In linear algebra, eigenvalues arise when we study how a linear transformation acts on vectors. Normally, when a matrix multiplies a vector, both the magnitude and direction of the vector change. However, for certain special vectors, the direction remains the same—only the magnitude is scaled.

These special vectors are called eigenvectors, and the corresponding scaling factors are called eigenvalues.
Let A be a **square matrix** of order $n \times n$.
A scalar $\lambda$is called an **eigenvalue** of matrix Aif there exists a **non-zero vector** $\mathbf{x} \neq \mathbf{0}$such that

$$A\mathbf{x} = \lambda\mathbf{x}$$

Here:

- $\mathbf{x}$ is called an **eigenvector** corresponding to the eigenvalue λ,
- λ represents the factor by which the eigenvector is scaled.

Rewriting the equation:

$$(A - \lambda I)\mathbf{x} = \mathbf{0}$$

For a non-trivial solution to exist,

$$det(A - \lambda I) = 0$$

Explanation: Characteristic equation
For a 2x2 matrix
Let A = $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ be a square matrix of order 2x2. We can form the matrix $A-\lambda I$, where λ is a scalar. The determinant of a square matrix equated to zero. i.e., $|A-\lambda I| = \begin{vmatrix} a_{11}-\lambda & a_{12} \\ a_{21} & a_{22}-\lambda \end{vmatrix}$ =0 is called the character equation of A.

On expanding the determinant, the characteristic equation taken the form

$$\boldsymbol{\lambda^2 - \lambda(trace\ of\ A) + determinant\ of\ A = 0}.$$

For a 3x3 matrix

Let A = $\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$ be a square matrix of order 3x3. We can form the matrix $A-\lambda I$, where λ is a scalar. The determinant of a square matrix equated to zero. i.e., $|A-\lambda I| = 0$. On expanding the determinant, the characteristic equation taken the form

$$\boldsymbol{\lambda^3 - \lambda^2(trace\ of\ A) + \lambda(sum\ of\ principal\ minors\ of\ A) - determinant\ of\ A = 0}.$$

The roots of the characteristic equation are called the characteristic roots or latent roots or Eigen values of matrix A.

Eigen vector: A non-zero vector X is called an **eigenvector** of a matrix A associated with an eigen value λ if $A\mathbf{X} = \lambda X$ holds. That is,

An eigenvector of a square matrix *A* is a non-zero vector *X* such that when *A* acts on it, the direction of *X* does not change—it only gets scaled.

Example: Consider the matrix

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

The characteristic equation is $\det(A - \lambda I) = 0$

$$\lambda^2 - 4\lambda + 3 = 0$$

Implies $\lambda = 1,3$.

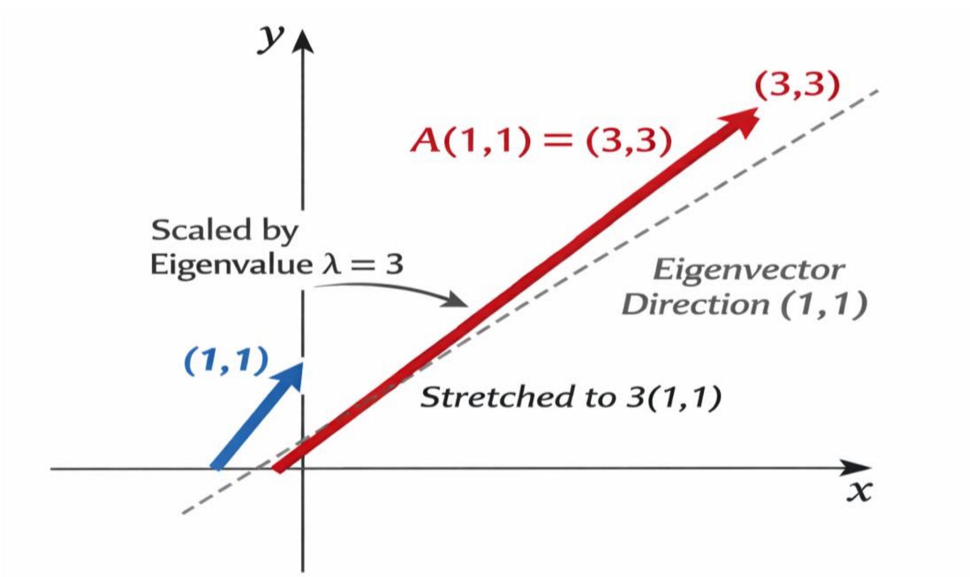
Step 3: Eigen Vector

For $\lambda = 3$:

$$(A - 3I)\mathbf{x} = 0 \Rightarrow \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}\mathbf{x} = 0$$

This gives

$$\mathbf{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$



Example-2: Find the eigenvalues and eigenvectors of the matrix

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}.$$

Properties of Eigen values Eigen Values

(1) The sum of the Eigen values of a matrix is the sum of the elements of the principal diagonal.
(2) If λ is an Eigen value of a matrix, A then $1/\lambda$ is the Eigen value of A^{-1}.
(3) If λ is an Eigen value of an orthogonal matrix, then $1/\lambda$ is also its Eigen value.
(4) If λ be an Eigen value for a non – singular matrix, A show that $\frac{|A|}{\lambda}$ is an Eigen value of the matrix (Adj A).
(5) The Eigen values of triangular matrix A are equal to the elements of the principal diagonal of A.
(6) If $\lambda_1, \lambda_2, ----, \lambda_n$ are the Eigen values of a matrix A then A^m has the Eigen values $\lambda_1^m, \lambda_2^m, --- \lambda_n^m$.
(7) Any square matrix A and its transpose A^T have same Eigen values.
(8) The product of the Eigen values of a matrix is equal to its determinant.

Definition of Principal Component Analysis (PCA):

Principal Component Analysis (PCA) is a statistical and linear algebra–based dimensionality reduction technique that transforms a set of possibly correlated variables into a new set of uncorrelated variables, called principal components, ordered so that the first few components retain most of the variation (information) present in the original data.

Mathematically, the principal components are the eigenvectors of the covariance matrix of the data, and the corresponding eigenvalues represent the amount of variance explained by each component.

Case Study: Eigenvalues and Eigenvectors in Face Recognition (PCA / Eigenfaces)
Introduction
Eigenvalues and eigenvectors play a crucial role in dimensionality reduction, data compression, and pattern recognition. One of the most popular real-world applications is Face Recognition using Principal Component Analysis (PCA), commonly known as the Eigenfaces method.
In this case study, we examine how eigenvalues and eigenvectors are used to extract meaningful features from facial images while reducing computational complexity.

Conceptual Diagram (Eigenfaces Method)

Face 1 (64×64) | Face 2 (64×64) | Face 3 (64×64)

↓

Vectorization (4096 × 1 each)

↓

Mean Face Subtraction

↓

Covariance Matrix
(via A^TA trick)

↓

Eigenvectors (Eigenfaces)

↓

Eigenface1 | Eigenface2 | Eigenface3

↓

Projection into k-D Eigenspace

↓

● ● ● (Training faces)

Test Face Image (64×64)

|

Mean Subtraction

|

Projection (●)

|

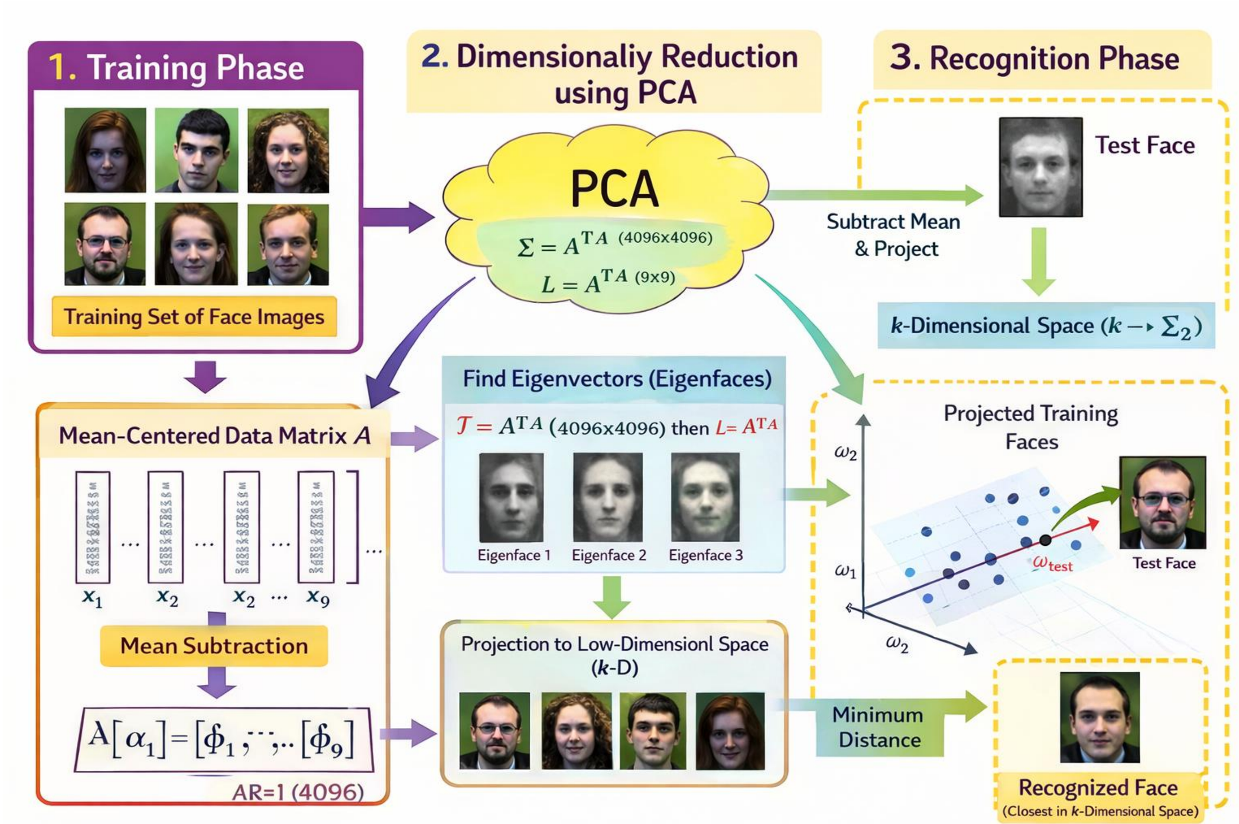
Distance Comparison in Eigenspace

|

Closest Match Found

|

Face Recognized



Challenges:

- High memory usage
- Slow computation
- Redundant information

Goal:

Reduce the dimensionality of face images while preserving the most important facial features.

Why Eigenvalues and Eigenvectors?

- Eigenvectors represent directions of maximum variance in data
- Eigenvalues indicate how important each direction is
- PCA uses eigenvectors of the covariance matrix

In face recognition, these eigenvectors are called Eigenfaces.

Methodology

Step 1: Data Collection

- Collect facial images of different people
- Convert images to grayscale
- Flatten each image into a vector

Example:

$$100 \times 100 \rightarrow 10{,}000 \text{ dimensional vector}$$

Step 2: Mean Centering

Compute the mean face and subtract it from each image vector:

$$X_{\text{centered}} = X - \mu$$

This ensures data is centered around the origin.
Step 3: Covariance Matrix

$$C = \frac{1}{n} X_{centered}^{T} X_{centered}$$

The covariance matrix captures relationships between pixels.
Step 4: Eigen value Decomposition
Solve:

$$CX = \lambda X$$

Where:

- X→ eigenvector (Eigenface)
- λ→ eigenvalue (importance)

Step 5: Feature Selection

- Sort eigenvalues in descending order
- Select top k eigenvectors
- These eigenvectors form a new feature space

Large eigenvalues = important facial patterns
Small eigenvalues = noise

Step 6: Projection
Project face images onto eigenfaces:

$$Y = X^{T} U$$

This gives a compact representation of faces.
Results

Aspect	Before PCA	After PCA
Dimensions	10,000	100–300
Storage	High	Low
Speed	Slow	Fast
Accuracy	Moderate	High

- Most facial information is captured using very few eigenvectors
- System becomes faster and more efficient

6. Interpretation of Eigenvalues and Eigenvectors

- Eigenvectors = Facial patterns (eyes, nose, mouth structures)
- Eigenvalues = Amount of variance captured by each pattern
- Larger eigenvalue → more important eigenface

Discussion about above mentioned cases study. For convenient purpose we are taking the case of 3 images.

Training Images (Grayscale)

Assume 3 training face images:

Image 1

$$I_1 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{bmatrix}$$

Image 2

$$I_2 = \begin{bmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \end{bmatrix}$$

Image 3

$$I_3 = \begin{bmatrix} 3 & 4 & 5 \\ 4 & 5 & 6 \\ 5 & 6 & 7 \end{bmatrix}$$

Step 1: Vectorization

Each 3×3image is converted into a 9×1column vector:

$$\mathrm{x}_1 = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 3 \\ 4 \\ 3 \\ 4 \\ 5 \end{pmatrix}, \mathrm{x}_2 = \begin{pmatrix} 2 \\ 3 \\ 4 \\ 3 \\ 4 \\ 5 \\ 4 \\ 5 \\ 6 \end{pmatrix}, \mathrm{x}_3 = \begin{pmatrix} 3 \\ 4 \\ 5 \\ 4 \\ 5 \\ 6 \\ 5 \\ 6 \\ 7 \end{pmatrix}$$

Step 2: Mean Face

$$\mu = \frac{1}{3}(\mathrm{x}_1 + \mathrm{x}_2 + \mathrm{x}_3) = \begin{pmatrix} 2 \\ 3 \\ 4 \\ 3 \\ 4 \\ 5 \\ 4 \\ 5 \\ 6 \end{pmatrix}$$

Step 3: Mean-Centered Images

$$\phi_i = \mathrm{x}_i - \mu$$

$$\phi_1 = \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{pmatrix}, \phi_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \; and \; \phi_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

Step 4: Data Matrix

$$A = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ & & \\ -1 & 0 & 1 \end{bmatrix}_{9\times 3}$$

Step 5: Compute $L = A^T A$

$$L = \begin{pmatrix} 9 & 0 & -9 \\ 0 & 0 & 0 \\ -9 & 0 & 9 \end{pmatrix}$$

Step 6: Eigenvalues and Eigenvectors

Eigenvalues:

$$\lambda_1 = 18, \lambda_2 = 0, \lambda_3 = 0$$

Corresponding eigenvector for λ_1: $\mathbf{v}_1 = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$

Step 7: Eigenface Construction $\mathbf{u}_1 = A\mathbf{v}_1 = \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix}$

Normalize:

$$\mathbf{u}_1 = \frac{1}{6}\begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{pmatrix}.$$

Step 8: Projection of Training Faces

$$\omega_i = \mathbf{u}_1^T \boldsymbol{\phi}_i$$

Face	**Projection**
Face 1	-3
Face 2	0
Face 3	3

Step 9: Test Face Recognition Test Image $I_{test} = \begin{bmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \end{bmatrix}$

Vectorized: $\mathrm{x}_{test} = \begin{pmatrix} 2 \\ 3 \\ 4 \\ 3 \\ 4 \\ 5 \\ 4 \\ 5 \\ 6 \end{pmatrix}$

Mean-centered:

$$\boldsymbol{\phi}_{test} = \mathbf{0}$$

Projection:

$$\boldsymbol{\omega_{test} = 0}$$

Recognition Decision

Distances:

$$|\ 0 - 0\ | = 0(\text{minimum})$$

Test face is recognized as Face 2.

Applications Beyond Face Recognition

- Image compression
- Noise reduction
- Recommendation systems
- Feature extraction in Machine Learning
- Deep Learning preprocessing (CNN inputs)

Problem: Face Recognition using PCA (Eigenfaces)

A database contains 5 grayscale face images, each of size 2×2pixels.

Each image is converted into a 4-dimensional column vector by stacking its pixels column-wise.

The face vectors are: $x_1 = \begin{pmatrix}1\\2\\3\\4\end{pmatrix}, x_2 = \begin{pmatrix}2\\3\\4\\5\end{pmatrix}, x_3 = \begin{pmatrix}3\\4\\5\\6\end{pmatrix}, x_4 = \begin{pmatrix}4\\5\\6\\7\end{pmatrix}, x_5 = \begin{pmatrix}5\\6\\7\\8\end{pmatrix}$

A test face image is given by: $x_3 = \begin{pmatrix}3\\4\\5\\6\end{pmatrix}$.

Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a mathematical technique widely used in machine learning for tasks like dimensionality reduction, noise reduction, and data compression. By breaking down a matrix into its fundamental components, SVD helps uncover patterns in data, making it easier to analyse and process large datasets.

Purpose of SVD in Machine Learning:

SVD enables the simplification of complex datasets by:

- Reducing dimensionality while retaining key information.
- Enhancing model performance by removing noise.
- Facilitating data compression for efficient storage.

Mathematics Behind SVD Algorithm

Singular Value Decomposition (SVD) is a mathematical process that decomposes a matrix into three distinct matrices: U, Σ, and V^T. This decomposition is the foundation of its applications in machine learning, allowing for efficient data transformation and analysis.

Definition and Components:

For a matrix A with dimensions m×n, SVD is represented as: $A = U\Sigma V^T$

where:

U (Left Singular Vectors): An m×m orthogonal matrix representing the row space of A.

Σ (Singular Values): A diagonal $m \times n$ matrix containing non-negative singular values, which represent the importance or weight of corresponding dimensions.

V^T (Right Singular Vectors): An $n \times n$ orthogonal matrix representing the column space of A.

Geometric Interpretation:

SVD geometrically transforms a dataset by:

1. **Rotation:** Aligning the data along its principal directions (defined by U and V).
2. **Scaling:** Adjusting the data based on the singular values in Σ.

This transformation helps identify the most significant features or patterns in the data, making it easier to process.

Relation to Eigenvalues and Eigenvectors:

SVD is closely related to eigen decomposition, a technique used to diagonalize square matrices:

- For $A^T A$, the eigenvectors form V, and the square roots of eigenvalues are the singular values.
- For AA^T , the eigenvectors form U, and the singular values remain the same.

Key Difference:

- Eigen decomposition works only for square matrices, while SVD applies to rectangular matrices, making it more versatile for real-world applications.

Singular Value Decomposition Example

The process of finding the singular value decomposition for 3×3 matrix and 2×2 matrix is the same. Let's have a look at the example of 2×2 matrix decomposition.

Singular Value Decomposition

2×2 Matrix Example

Example 1: Obtain the singular value decomposition of a matrix $A = \begin{bmatrix} -4 & -7 \\ 1 & 4 \end{bmatrix}$

Solution: Give that the matrix $A = \begin{bmatrix} -4 & -7 \\ 1 & 4 \end{bmatrix}$.

Step 1: Compute $A^T A$

$$A^T A = \begin{bmatrix} -4 & 1 \\ -7 & 4 \end{bmatrix}\begin{bmatrix} -4 & -7 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} 17 & 32 \\ 32 & 65 \end{bmatrix}.$$

Step 2: Eigenvalues of $A^T A$

Solve $\det(A^T A - \lambda I) = 0$:

$$\lambda^2 - 82\lambda + 81 = 0$$
$$\Rightarrow \lambda_1 = 81, \lambda_2 = 1.$$

Step 3: Singular values

$$\sigma_1 = \sqrt{81} = 9, \sigma_2 = \sqrt{1} = 1.$$

So,

$$\Sigma = \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix}.$$

Step 4: Right singular vectors (V)

Eigenvectors of $A^T A$:

- For $\lambda_1 = 81$: eigenvector $(1, 2)$
- For $\lambda_2 = 1$: eigenvector $(-2,1)$

Normalize:

$$v_1 = \frac{1}{\sqrt{5}}\begin{bmatrix} 1 \\ 2 \end{bmatrix}, v_2 = \frac{1}{\sqrt{5}}\begin{bmatrix} -2 \\ 1 \end{bmatrix}.$$
$$V = \frac{1}{\sqrt{5}}\begin{bmatrix} 1 & -2 \\ 2 & 1 \end{bmatrix}.$$

Step 5: Left singular vectors (U)

$$u_i = \frac{1}{\sigma_i} A v_i$$
$$u_1 = \frac{1}{\sqrt{5}}\begin{bmatrix} -2 \\ 1 \end{bmatrix}, u_2 = \frac{1}{\sqrt{5}}\begin{bmatrix} -1 \\ -2 \end{bmatrix}.$$
$$U = \frac{1}{\sqrt{5}}\begin{bmatrix} -2 & -1 \\ 1 & -2 \end{bmatrix}.$$

Final SVD

$$\boxed{A = U\Sigma V^T}$$

where

$$U = \frac{1}{\sqrt{5}}\begin{bmatrix} -2 & -1 \\ 1 & -2 \end{bmatrix}, \Sigma = \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix}, V^T = \frac{1}{\sqrt{5}}\begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}.$$

Numerical verification of SVD

From the previous result:

$$U = \frac{1}{\sqrt{5}}\begin{bmatrix} -2 & -1 \\ 1 & -2 \end{bmatrix}, \Sigma = \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix}, V^T = \frac{1}{\sqrt{5}}\begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}$$

Now compute $U\Sigma$:

$$U\Sigma = \frac{1}{\sqrt{5}}\begin{bmatrix} -18 & -1 \\ 9 & -2 \end{bmatrix}$$

Then

$$A = U\Sigma V^T = \frac{1}{5}\begin{bmatrix} -18 & -1 \\ 9 & -2 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}$$

Multiplying:

$$= \frac{1}{5}\begin{bmatrix} -16 & -35 \\ 5 & 20 \end{bmatrix} = \begin{bmatrix} -4 & -7 \\ 1 & 4 \end{bmatrix}$$

Verified numerically — the SVD is correct.
Rank-1 approximation (dominant singular value)
The rank-1 approximation is:

$$A_1 = \sigma_1 u_1 v_1^T$$

where

$$\sigma_1 = 9, u_1 = \frac{1}{\sqrt{5}}\begin{bmatrix} -2 \\ 1 \end{bmatrix}, v_1 = \frac{1}{\sqrt{5}}\begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Step 1: Compute $u_1 v_1^T$

$$u_1 v_1^T = \frac{1}{5}\begin{bmatrix} -2 & -4 \\ 1 & 2 \end{bmatrix}$$

Step 2: Multiply by $\sigma_1 = 9$

$$A_1 = \frac{9}{5}\begin{bmatrix} -2 & -4 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} -3.6 & -7.2 \\ 1.8 & 3.6 \end{bmatrix}$$

Final Results

Exact matrix

$$A = \begin{bmatrix} -4 & -7 \\ 1 & 4 \end{bmatrix}$$

Rank-1 approximation

$$\boxed{A_1 = \begin{bmatrix} -3.6 & -7.2 \\ 1.8 & 3.6 \end{bmatrix}}$$

Exmple-2: Obtain SVD of the matrix

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{3} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

This is a simple 3D example (diagonal matrix), but it perfectly illustrates how SVD works.
Step 1: Definition of SVD
For any matrix A,

$$A = U\Sigma V^T$$

where

- U= left singular vectors (orthonormal)
- Σ= singular values (diagonal, non-negative)
- V= right singular vectors (orthonormal)

Step 2: Compute $A^T A$

$$A^T A = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Step 3: Singular Values
Eigenvalues of $A^T A$are:

$$\lambda_1 = 9, \lambda_2 = 4, \lambda_3 = 1$$

Singular values:

$$\sigma_1 = 3, \sigma_2 = 2, \sigma_3 = 1$$

So,

$$\Sigma = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Step 4: Right Singular Vectors (V)

Since $A^T A$is diagonal, eigenvectors are the standard basis vectors:

$$V = I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Step 5: Left Singular Vectors (U)

$$U = AV\Sigma^{-1} = I$$

Final SVD

$$A = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{V^T}$$

Geometric Interpretation (3D)

- V^T: rotates the 3D coordinate system
- Σ: stretches space by **3, 2, and 1** along three orthogonal directions
- U: rotates the result again

This shows how SVD decomposes a 3D transformation into **rotation** → **scaling** → **rotation**.

Rank-1 Approximation (Optional)

Keeping only the largest singular value:

$$A_1 = 3 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Used in **3D data compression** and **noise reduction**.

Example 3: Given 3D Matrix

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

This kind of matrix appears in **3D data / image slices / correlated features**.

Solution:

Step 1: Compute $A^T A$

$$A^T A = \begin{bmatrix} 2 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Step 2: Eigenvalues of $A^T A$

Solve $\det(A^T A - \lambda I) = 0$

Eigenvalues:

$$\lambda_1 = 4, \lambda_2 = 0, \lambda_3 = 0$$

Step 3: Singular Values

$$\sigma_1 = \sqrt{4} = 2, \sigma_2 = 0, \sigma_3 = 0$$

So,

$$\Sigma = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Rank of A = 1

Step 4: Right Singular Vectors *V*

Eigenvector for $\lambda_1 = 4$:

$$v_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

Complete orthonormal basis:

$$V = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Step 5: Left Singular Vectors *U*

$$u_1 = \frac{Av_1}{\sigma_1} = \frac{1}{2} \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

So,

$$U = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Final SVD Decomposition

$$A = U\Sigma V^T$$

$$A = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}^T$$

Rank-1 Approximation (3D Compression)

Since only **one singular value is non-zero**:

$$A \approx \sigma_1 u_1 v_1^T$$

$$A = 2\left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}\right)\left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}\right)$$

Exact reconstruction using **one direction only**

Geometric Meaning in 3D

- All data lies along **one direction**
- Other two dimensions contain **no new information**
- Perfect example of **dimensionality reduction from 3D → 1D**

Example 4: Given 3D Rectangular Matrix (3×2)

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

This represents **3D data with 2 features** (very common in ML and image processing).

Step 1: Dimensions

- A is $\mathbf{3 \times 2}$
- $U \rightarrow \mathbf{3 \times 3}$
- $\Sigma \rightarrow \mathbf{3 \times 2}$
- $V^T \rightarrow \mathbf{2 \times 2}$

Step 2: Compute $A^T A$

$$A^T A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

Step 3: Eigenvalues of $A^T A$

$$\det(A^T A - \lambda I) = 0$$
$$(2 - \lambda)^2 - 1 = 0$$
$$\lambda_1 = 3, \lambda_2 = 1$$

Step 4: Singular Values

$$\sigma_1 = \sqrt{3}, \sigma_2 = 1$$
$$\Sigma = \begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Step 5: Right Singular Vectors V

Eigenvectors of $A^T A$:

For $\lambda_1 = 3$:

$$v_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

For $\lambda_2 = 1$:

$$v_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$
$$V = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Step 6: Left Singular Vectors U

$$u_i = \frac{A v_i}{\sigma_i}$$
$$u_1 = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}, u_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

Complete with a third orthogonal vector:

$$u_3 = \frac{1}{\sqrt{3}} \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}$$

Final SVD

$$A = U \Sigma V^T$$

Where:

$$U = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{3}} \\ \frac{2}{\sqrt{6}} & 0 & \frac{1}{\sqrt{3}} \end{bmatrix}$$

Rank of the Matrix

$$\text{Rank}(A) = 2$$

(Number of non-zero singular values)

Rank-1 Approximation (Dimensionality Reduction)

$$A_1 = \sigma_1 u_1 v_1^T$$

$$A_1 = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \\ 1 & 1 \end{bmatrix}$$

Reduced from **3D → 1D dominant direction**

Geometric Interpretation (3D)

- Data lives mainly on a **plane**
- SVD finds **two orthogonal directions**
- Third direction contains **no information**

Example-5: Obtain SVD of $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 2 & 2 \end{bmatrix}$

Solution: **Step 1: Compute $A^T A$**

$$A^T A = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 2 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} 9 & 8 \\ 8 & 9 \end{bmatrix}$$

Step 2: Eigenvalues of $A^T A$

$$\det(A^T A - \lambda I) = 0$$

$$\begin{vmatrix} 9-\lambda & 8 \\ 8 & 9-\lambda \end{vmatrix} = (9-\lambda)^2 - 64 = 0$$

$$\lambda^2 - 18\lambda + 17 = 0$$

$$\lambda_1 = 17, \lambda_2 = 1$$

Step 3: Singular Values

$$\sigma_1 = \sqrt{17}, \sigma_2 = 1$$

$$\Sigma = \begin{bmatrix} \sqrt{17} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Step 4: Right Singular Vectors V

For $\lambda_1 = 17$

$$(A^T A - 17I)v = 0 \Rightarrow \begin{bmatrix} -8 & 8 \\ 8 & -8 \end{bmatrix} v = 0$$

$$v_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

For $\lambda_2 = 1$

$$v_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$V = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Step 5: Left Singular Vectors *U*

$$u_i = \frac{Av_i}{\sigma_i}$$

First left singular vector u_1

$Av_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 3 \\ 3 \\ 4 \end{bmatrix}, u_1 = \frac{1}{\sqrt{34}}\begin{bmatrix} 3 \\ 3 \\ 4 \end{bmatrix}$

Second left singular vector u_2

$Av_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, u_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, Av_3 =$

Third vector u_3(orthogonal completion)

$$u_3 = \frac{1}{\sqrt{17}}\begin{bmatrix} -2 \\ -2 \\ 3 \end{bmatrix}$$

Final SVD

$$A = U\Sigma V^T$$

where

$$U = \begin{bmatrix} \frac{3}{\sqrt{34}} & \frac{1}{\sqrt{2}} & -\frac{2}{\sqrt{17}} \\ \frac{3}{\sqrt{34}} & -\frac{1}{\sqrt{2}} & -\frac{2}{\sqrt{17}} \\ \frac{4}{\sqrt{34}} & 0 & \frac{3}{\sqrt{17}} \end{bmatrix}$$

Rank of the Matrix

Rank$(A) = 2$ (two non-zero singular values)

Rank-1 Approximation (Compression)

$$A_1 = \sigma_1 u_1 v_1^T = \frac{\sqrt{17}}{2}\begin{bmatrix} 3 & 3 \\ 3 & 3 \\ 4 & 4 \end{bmatrix}$$

Practice Problems on Singular Value Decomposition

1. Obtain the SVD of the matrix A = [[2, 0], [0, 1]]. Also determine its rank.
2. Given A = [[1, 0], [0, 0]]:
   (a) Find the singular values
   (b) Construct the rank-1 approximation
   (c) Interpret the result.
3. Consider A = [[1, 1], [1, 1]]:
   (a) Compute the SVD
   (b) Find the rank
   (c) Explain why one singular value is zero.

4. Compute the SVD of the rectangular matrix A = [[1, 0], [0, 1], [0, 0]]:
   (a) Find U, Σ, V^T
   (b) Determine the rank.
5. Let A = [[1, 1], [0, 1], [1, 0]]:
   (a) Compute A^T A
   (b) Find singular values
   (c) Explain the geometric meaning.
6. Given A = [[2, 1], [1, 2], [2, 2]]:
   (a) Find the singular values
   (b) Determine the rank
   (c) Construct the rank-1 approximation.
7. A grayscale image is represented by A = [[3, 2, 1], [2, 1, 0], [1, 0, 0]]:
   (a) Compute the SVD
   (b) Determine the rank
   (c) Explain image compression using SVD.

Numerical Example: PCA using Eigenvalues & Eigenvectors
Given Data Matrix
Assume we have **3 face images**, each represented using **2 features** (e.g., simplified pixel intensities).

$$X = \begin{bmatrix} 2 & 4 \\ 4 & 6 \\ 6 & 8 \end{bmatrix}$$

Rows → Images, Columns → Features
Step 1: Compute the Mean Vector
Mean of each column:

$$\mu_1 = \frac{2+4+6}{3} = 4 \mu_2 = \frac{4+6+8}{3} = 6$$
$$\mu = \begin{bmatrix} 4 & 6 \end{bmatrix}$$

Step 2: Mean Center the Data

$$X_{centered} = X - \mu$$
$$= \begin{bmatrix} 2-4 & 4-6 \\ 4-4 & 6-6 \\ 6-4 & 8-6 \end{bmatrix} = \begin{bmatrix} -2 & -2 \\ 0 & 0 \\ 2 & 2 \end{bmatrix}$$

Step 3: Compute the Covariance Matrix

$$C = \frac{1}{n-1} X_{centered}^T X_{centered}$$
$$C = \frac{1}{2} \begin{bmatrix} -2 & 0 & 2 \\ -2 & 0 & 2 \end{bmatrix} \begin{bmatrix} -2 & -2 \\ 0 & 0 \\ 2 & 2 \end{bmatrix}$$
$$C = \frac{1}{2} \begin{bmatrix} 8 & 8 \\ 8 & 8 \end{bmatrix} = \begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}$$

Step 4: Find Eigenvalues
Solve:

$$| C - \lambda I | = 0$$

$$\begin{vmatrix} 4-\lambda & 4 \\ 4 & 4-\lambda \end{vmatrix} = 0$$

$$(4-\lambda)^2 - 16 = 0$$

$$\lambda^2 - 8\lambda = 0 \Rightarrow \lambda(\lambda - 8) = 0$$

$$\boxed{\lambda_1 = 8, \lambda_2 = 0}$$

Step 5: Find Eigenvectors

For $\boldsymbol{\lambda_1 = 8}$

$$(C - 8I)v = 0 \Rightarrow \begin{bmatrix} -4 & 4 \\ 4 & -4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0$$

$$x = y \Rightarrow v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Normalize:

$$v_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

For $\boldsymbol{\lambda_2 = 0}$

$$x = -y \Rightarrow v_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Step 6: Principal Component (PCA Direction)

- Largest eigenvalue: **8**
- Corresponding eigenvector:

$$\boxed{\text{Principal Component} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}}$$

This is the **Eigenface direction** in simplified form.

Step 7: Project Data onto Principal Component

$$Z = X_{centered} \cdot v_1$$

$$Z = \begin{bmatrix} -2 & -2 \\ 0 & 0 \\ 2 & 2 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -2\sqrt{2} \\ 0 \\ 2\sqrt{2} \end{bmatrix}$$

Data reduced from **2D → 1D**

Step 8: Interpretation (Face Recognition Context)

- Eigenvalue **8** → major facial variation
- Eigenvalue **0** → redundant feature
- PCA removes redundancy and noise
- Projection coefficients are used for **face matching**

Final Exam-Ready Conclusion

Using PCA, the original 2-dimensional data is reduced to 1 dimension by selecting the eigenvector corresponding to the largest eigenvalue. This principal component captures maximum variance and forms the basis of the Eigenfaces method in face recognition.

Rank-1 Reconstruction (Using PCA)

From the previous result:

Principal eigenvector: $v_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	Projected values: $Z = \begin{bmatrix} -2\sqrt{2} \\ 0 \\ 2\sqrt{2} \end{bmatrix}$
Rank-1 Approximation Formula $X_{rank1} = Zv_1^T$	**Add Mean Back** Mean vector:

$= \begin{bmatrix} -2\sqrt{2} \\ 0 \\ 2\sqrt{2} \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} -2 & -2 \\ 0 & 0 \\ 2 & 2 \end{bmatrix}$	$X_{reconstructed} = \begin{bmatrix} 2 & 4 \\ 4 & 6 \\ 6 & 8 \end{bmatrix}$
Perfect reconstruction here because data is already rank-1.	
Interpretation • Only **one eigenvalue is non-zero** • Data lies in **one principal direction** • PCA achieves **maximum compression without loss**	
Eigenfaces using $A^T A$ Trick (Numerical)	Image size ≫ number of images
Step 1: Mean-centered matrix $A = \begin{bmatrix} -2 & -2 \\ 0 & 0 \\ 2 & 2 \end{bmatrix}$ **Step 3: Eigenvalues** $A^T A$ *are* $\lambda_1 = 16, \lambda_2 = 0$	**Step 2: Compute $A^T A$** $A^T A = \begin{bmatrix} -2 & 0 & 2 \\ -2 & 0 & 2 \end{bmatrix} \begin{bmatrix} -2 & -2 \\ 0 & 0 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} 8 & 8 \\ 8 & 8 \end{bmatrix}$ $\lvert A^T A - \lambda I \rvert = \left\lvert \begin{bmatrix} 8 & 8 \\ 8 & 8 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right\rvert = 0$
Step 4: Eigenvector $v_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	**Step 5: Compute Eigenface** $u_1 = Av_1 = \begin{bmatrix} -2 & -2 \\ 0 & 0 \\ 2 & 2 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -2\sqrt{2} \\ 0 \\ 2\sqrt{2} \end{bmatrix}$
This vector reshaped → **Eigenface image**	
Distance-Based Face Recognition (Numerical) **Training weights:** $\omega_1 = -2\sqrt{2}, \omega_2 = 0, \omega_3 = 2\sqrt{2}$	
Test face: $\Gamma_{test} = \begin{bmatrix} 5 & 7 \end{bmatrix}$ and Mean-center: $\Phi_{test} = \begin{bmatrix} 1 & 1 \end{bmatrix}$	
Projection: $\omega_{test} = \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \sqrt{2}$	
Euclidean Distance $d_1 = \lvert \sqrt{2} + 2\sqrt{2} \rvert = 3\sqrt{2}$ $d_2 = \lvert \sqrt{2} - 0 \rvert = \sqrt{2}$ $d_3 = \lvert \sqrt{2} - 2\sqrt{2} \rvert = \sqrt{2}$	

📌 **Closest match → Face 2 or Face 3**

Face Images
↓
Vectorization
↓
Mean Face Computation
↓
Mean Subtraction
↓
Covariance Matrix ($A^T A$)
↓
Eigenvalues & Eigenvectors
↓
Eigenfaces Selection
↓
Projection (Weights)
↓

Distance Comparison
↓
Face Recognition