SPECTRUM OF ENGINEERING SCIENCES
ISSN (E) 3007-3138 (P) 3007-312X VOLUME 4, ISSUE 3, 2026

EIGENVALUE-BASED DIMENSIONALITY REDUCTION IN IMAGE COMPRESSION: MATHEMATICAL ANALYSIS AND EFFECTS ON IMAGE QUALITY

Darain Fatima[*1], Syed Ale Hassan Rizvi[2], Mahmood Zafar Abbas[3], Muhammad Faizan Raja[4], Mirza Alizar Sami[5], Muwadat Ali[6]

*[*1]Department of Matheatics Riphah International University Islamabad (Faisalabad Campus), Pakistan*
[2,3]Electrical and Telecommunications Engineering Ahlul Bayt International University, Tehran
[4]Department of Information Technology, Virtual University, Islamabad, 44000, Pakistan.
[5]Data Science (MS) National University of Science and Technology MISIS (Moscow - Russia)
[6]Ms Software Engineering Dalian University of Technology China

[*1]darainfatima51272@gmail.com, [2]alehassan09@gmail.com, [3]mahmoodzafarabbas@gmail.com, [4]faizan.raja@vu.edu.pk, [5]alii.zar090@gmail.com

DOI: https://doi.org/10.5281/zenodo.23615329

Keywords
Principal component analysis; image compression; eigenvalue decomposition; dimensionality reduction; explained variance; peak signal-to-noise ratio; structural similarity; compression ratio; reconstruction error; grayscale images

Article History
Received: 01 January 2026
Accepted: 16 March 2026
Published: 31 March 2026

Copyright @Author
Corresponding Author: *
Darain Fatima

Abstract

Digital image storage and transmission is demanding, and lossy compression reduces this demand by removing perceptually redundant data. Principal Component Analysis is one of the most established linear tools for this purpose but the way in which the number of retained principal components influences measurable image quality is often discussed only qualitatively. In this work we present the mathematical foundations of eigenvalue-based dimensionality reduction for image compression and analyse the effects of the number of components on reconstruction fidelity.

The work formalises the compression pipeline from the image being represented as a matrix, through centring the data, constructing the covariance matrix and eigen decomposition, to component selection and reconstruction. It is shown that the squared reconstruction error is the sum of the discarded eigenvalues and the explained variance ratio is the fraction of the total variance kept by the first k components. We use a greyscale experimental setting, and resize images into a fixed 256 × 256 grid and process them row-wise, where rows are observations and columns are features. Compression performance is quantified using mean squared error, peak signal to noise ratio, structural similarity, explained variance and a theoretical storage ratio for a variety of component numbers.

The analysis shows that the reconstruction quality increases monotonically with the number of components, but with diminishing returns, so the relationship is a rate-quality curve rather than a single operating point. The result strongly depends on the shape of the eigenvalue spectrum: images with steep spectra can tolerate aggressive truncation, while images with flatter spectra cannot. Compression only happens when the number of components is below the break-even point defined by the stored scores, eigenvectors and mean vector. The study concludes that the appropriate component budget is application specific, and recommends selecting the smallest component count that meets both a cumulative explained variance target and a plateaued structural similarity score.

1. INTRODUCTION

1.1 Background of Digital Image Compression

The rapid growth of digital imaging has made the storage and transmission of image data a central concern in modern computing. A single high-resolution colour image can occupy several megabytes, and large image collections place heavy demands on memory, bandwidth, and communication time. Image compression addresses this burden by representing an image in fewer bits than its original form. Compression methods are commonly divided into two families. Lossless techniques preserve every pixel exactly and permit perfect reconstruction, but the achievable reduction is modest, typically yielding compression ratios of 2:1 to 3:1 on still-frame medical images (Cosman et al., 1994). Lossy techniques accept a controlled amount of distortion in exchange for far greater reduction, exploiting the fact that much of the data in a natural image is perceptually redundant. Principal Component Analysis is one of the most established linear tools for this purpose. In PCA, samples are rewritten on the basis of the eigenvectors; the components of the resulting vector are the principal components, and their variances are the eigenvalues (Jäntschi, 2023). Applied to images, PCA reduces dimensionality while retaining the majority of the variance present in the data (Szabó, 2025).

1.2 Problem Statement

The central difficulty in PCA-based compression is that discarding principal components means discarding information. Smaller values of the retained component count produce greater compression but some loss of picture detail, whereas larger values preserve more image information at the cost of less compression (Guntuboina & Srivast, 2024). Because the energy of an image is concentrated on its first small number of component vectors (Yang et al., 2004), it is tempting to truncate aggressively; yet the exact consequences of that truncation for measurable image quality are not always well characterised. This creates a practical problem: how many components should be retained, and how does that choice translate into compression performance and visible degradation?

1.3 Research Objectives

This study aims to: (i) present the mathematical basis of eigenvalue-based dimensionality reduction for image compression; (ii) analyse how the number of retained principal components influences the compression ratio and reconstruction error; (iii) evaluate reconstructed image quality using PSNR and SSIM; and (iv) identify the range of components that best balances compression against fidelity.

1.4 Research Questions

The study addresses the following questions: How does the number of retained principal components affect the compression ratio? How does it affect reconstruction error? What relationship exists between the retained components and PSNR and SSIM values? And is there a point of diminishing returns beyond which additional components yield only marginal quality gains?

1.5 Main Contributions of the Study

This work contributes a systematic mathematical treatment of eigenvalue-based dimensionality reduction applied to image compression, together with an empirical evaluation of compression ratio, reconstruction error, PSNR, and SSIM across varying numbers of retained components. It clarifies the trade-off between compression and fidelity, and offers practical guidance for selecting the number of principal components in compression tasks.

2. Literature Review

2.1 Traditional Image Compression Techniques

Traditional compression pipelines rest on a sequence of operations — sampling, signal decomposition, quantization and lossless coding — in which the image is first decomposed by a linear transformation such as a Fourier or discrete cosine transform or by a wavelet filter bank (Cosman et

al., 1994). JPEG implements this pattern by applying a discrete cosine transform to every 8 × 8-pixel block and then quantizing the resulting coefficients (Wang et al., 2024), discarding high-frequency content to which the human psycho-visual system is relatively insensitive (Dejean-Servières et al., 2017). Typical JPEG compression ratios fall between 10 and 20 (Miklós et al., 2020), but blockwise quantization introduces coding artifacts such as blockiness and ringing (Ouyang & Chen, 2024), and the blocking artifacts become more observable once the ratio exceeds roughly 20:1 (Si & Kim, 2024). JPEG2000 replaces the DCT with a wavelet-based method and delivers a typical compression gain in the range of 20% (Dejean-Servières et al., 2017). Lossless coding, by contrast, typically yields compression ratios of only 2:1 to 3:1 on still-frame medical images, a level that is often inadequate for storage or communication facilities (Cosman et al., 1994).

2.2 Principal Component Analysis in Image Processing

PCA, also known as the Karhunen-Loeve expansion, is a classical feature extraction and data representation technique (Yang et al., 2004). For an image the matrix is flattened into a vector, mean-centred, and the covariance matrix diagonalized; compression is performed by projecting the centered data vector onto the top *k* eigenvectors, with reconstruction by the inverse transformation (Guntuboina & Srivast, 2024). Reviews of the field credit PCA with simplicity, high compression rates and higher picture reconstruction quality than competing approaches (Guntuboina & Srivast, 2024), and PCA has been used to compress image data across several domains (Gewers et al., 2021). A JPEG2000 variant that inserts PCA for spectral decorrelation performs best when a reduced number of PCs are retained and coded (Du & Fowler, 2007). The principal weakness is that PCA alone gives good image quality but a low compression ratio (Chen et al., 2016). 2DPCA tackles a related limitation by constructing the image covariance matrix directly from 2D matrices, requiring less time to determine the corresponding eigenvectors (Yang et al., 2004), although it requires more coefficients for image representation than PCA (Yang et al., 2004). Dedicated studies have also applied PCA directly to image compression (Abbas et al., 2018).

2.3 Eigenvalue-Based Dimensionality Reduction

The mathematical foundation lies in the eigenproblem, whose review literature defines the concepts of characteristic polynomial, eigendecomposition, eigenpair, eigenproblem, eigenspace, eigenvalue and eigenvector (Jäntschi, 2023). PCA is derived as a dimensionality reduction algorithm that maximizes the variance in the low-dimensional representation of the data (Deisenroth et al., 2020), following mean subtraction, standardization and eigendecomposition of the covariance matrix (Deisenroth et al., 2020). The variance of the data projected onto the *m*-th principal component equals its eigenvalue λm, and the variance captured by the first *M* components equals the sum of the eigenvalues associated with the corresponding eigenvectors (Deisenroth et al., 2020). Because only a few eigenvalues have a value that differs significantly from 0 (Deisenroth et al., 2020), the energy of an image is concentrated on its first small number of component vectors (Yang et al., 2004). Decomposition is obtained in practice through algorithms such as Rayleigh–Ritz, Jacobi–Davidson and Lanczos–Arnoldi (Jäntschi, 2023). SVD offers a closely related route: truncating the expansion to the first *k* eigenvalues, the compressed image approaches the original as *k* increases (seshaiah et al., 2016), with reported compression of 47% at k = 60 and 16% at k = 20 (Rowayda, 2012).

2.4 Image Quality Assessment Metrics

PSNR normalizes distortion against the maximum input symbol energy (Cosman et al., 1994), yet MSE has remained the standard fidelity measure despite its poor correlation with human perception (Ding et al., 2020); higher PSNR values indicate better reconstruction quality but poorly correlate with the perceived quality by human eyes (Izadi et al., 2022). SSIM was introduced as a complementary framework based on the degradation of structural information (Wang et al., 2004), measuring the visual impact of changes in luminance, contrasts, spatial dependencies and

structural information (Izadi et al., 2022), and models perception implicitly in coding contexts (Brooks et al., 2008). Contemporary studies report several metrics together: near-lossless work evaluates correlation, MSE, SNR, SSIM, compression ratio and computational time (Paikrao et al., 2022), while benchmarks of learned codecs average bit-rate and the distortion (Hu et al., 2021).

2.5 Research Gap

Prior image compression research has concentrated on general lossy approaches such as JPEG and WebP, which may not maximize compression for individual picture data (Guntuboina & Srivast, 2024). PCA-based studies report high reconstruction quality alongside low compression ratios (Chen et al., 2016), and although a linear model has been proposed to estimate the optimal number of PCs in hyperspectral dimensionality reduction (Du & Fowler, 2007), and the effect of the component count is currently illustrated qualitatively — smaller numpc numbers result in more lossy compression (Guntuboina & Srivast, 2024) — a systematic quantification of how the number of retained eigenvalues governs compression ratio, reconstruction error, PSNR and SSIM is still missing. That gap is what this study addresses.

Table 1. Comparative Summary of Existing Image Compression and Dimensionality Reduction Methods

Author(s) and Year	Compression Method	Mathematical Approach	Advantages	Limitations	Research Gap
Cosman, Gray & Olshen (1994)	Lossless and lossy coding; predictive pruned tree-structured vector quantization	Signal decomposition, quantization and entropy coding; quality measured by SNR, PSNR, subjective rating and diagnostic accuracy	Lossless coding techniques are well understood and readily available (Cosman et al., 1994); a well-designed lossy system minimizes information loss or image distortion for a given allotted storage space	The modest compression offered by lossless coding is often inadequate for storage or communicatio n facilities	No easily computable distortion measure is generally agreed to reflect perceptual quality or usefulness in an application
Jäntschi (2023)	Eigenproblem algorithms (review)	Characteristic polynomial, eigendecompositio n and eigenpair; Faddeev–LeVerrier, Rayleigh–Ritz, Jacobi–Davidson and Lanczos–Arnoldi algorithms	Coverage runs bottom-up from simple to complex (Jäntschi, 2023); the algorithm extracting the first principal component is also provided	The applications discussed include solving Bessel's, Helmholtz's, Laplace's, Legendre's, Poisson's, and Schrödinger's equations	Algorithmic formulations are not linked empirically to image quality or compression performance
Deisenroth, Faisal & Ong (2020)	PCA (first-principles derivation)	Mean subtraction, standardization and eigendecompositio	Principled variance-captured	Presented as a general dimensionality-	How the number of retained

		n of the covariance matrix	criterion; only a few eigenvalues have a value that differs significantly from 0, so few components carry most variance (Deisenroth et al., 2020)	reduction method; no image compression experiments or rate-distortion analysis	components translates into compression ratio and reconstruction error is not quantified
Yang, Zhang, Frangi, et al. (2004)	2DPCA, compared with PCA	Image covariance matrix constructed directly from 2D matrices rather than 1D vectors	Covariance matrix easier to evaluate accurately and eigenvectors obtained in less time; computationally more efficient feature extraction than PCA	2DPCA requires more coefficients for image representation than PCA, so it is less efficient in storage	Evaluated for face recognition and representation; the storage penalty is not resolved for compression purposes
Du & Fowler (2007)	PCA combined with JPEG2000 (hyperspectral)	PCA for spectral decorrelation and dimensionality reduction prior to JPEG2000 coding	Rate-distortion and information-preservation performance superior to that of the wavelet-based coder	Optimal number of retained PCs must be estimated rather than fixed	Optimal PC count is modelled for hyperspectral data; general image-quality metrics are not addressed
Sadek (2012)	SVD-based truncation	Singular value decomposition with truncation to k levels; PSNR reported against compression ratio	Effective compression across varied images: 47% compression at k = 60 and 16% at k = 20	Reconstruction quality depends heavily on the truncation level retained	Quality is reported mainly through PSNR versus compression ratio; structural similarity metrics are not evaluated
Venkata seshaiah, Roopadevi & Michahial (2016)	SVD on video frames	Reconstruction considering only the first k eigenvalues of the matrix, with PSNR	As the value of k is increased, the compressed image approaches the original image	Small k values (e.g. k = 5) produce images that are very blurred	Monotonic quality improvement with k is shown, but the

		= 10 log10[255/√MSE]			*k* required for a target fidelity is not determined
Chen, Huang & Sun (2016)	PCA combined with Contourlet Transform	The image data is divided into blocks, and each block is used as a sample vector before selecting the *k* larger eigenvalues of the covariance matrix	PSNR about 3 dB higher than JPEG2000 and 2 dB higher than Contourlet Transform at the same compression ratio	PCA alone has good image quality but a low compression ratio, requiring a second transform	PCA's compression deficiency is compensated by an added transform rather than resolved within the eigenvalue framework
El Asnaoui (2019)	Block SVD power method	Rank-*k* truncation compared against MATLAB's SVD function on real images (Asnaoui, 2019), assessed by PSNR and MSE	Perceptually similar reconstructions to the original images; rank can be tuned per image	Quality varies with the rank chosen for each image; PSNR and MSE capture only squared error	Comparison is limited to PSNR and MSE against SVD variants; perceptual metrics are not benchmarked
Miklós, Trpovski & Lončar-Turukalo (2020);	JPEG	8×8 block DCT projected onto 64 basis functions with quantization and entropy coding	Stable performance with a good trade-off between rate and distortion typical ratios of 10–20	Introduces blockiness and ringing artifacts that degrade subjective quality	Optimizing JPEG for a particular application requires expert knowledge and subtle parameter tuning
Hu,	Learned end-to-end lossy codecs benchmarked against JPEG, BPG and VVC	Rate-distortion evaluation across Kodak, Tecnick and CLIC 2019; bit-rate and distortion with BD-rate	Standardized cross-dataset protocol enabling fair comparison of codecs	Models must be trained with different trade-off coefficients λ and cover different bit-rate ranges	Linear-algebraic component selection (eigenvalue truncation) is outside the scope of the benchmark
Paikrao, Doye, Bhalerao, et al. (2022)	Hamming-code-based lossless, lossy and near-lossless coding	Bit-plane slicing with (7,4) and (8,4) Hamming code specifications and quantization; evaluated by	Near-lossless techniques maintain the trade-off between compression	Line graphics and text in document images more obviously show the lossy	Method is based on Hamming-code channel coding rather than

	of document regions	correlation, MSE, SNR, SSIM, compression ratio and computational time (Paikrao et al., 2022)	ratio and quality of the reconstructed image, with higher correlation, PSNR and SSIM values (Paikrao et al., 2022)	compression artefacts than other image types (Paikrao et al., 2022)	eigenvalue-based dimensionality reduction (Paikrao et al., 2022)
Guntuboina & Srivast (2024)	Review of RLE, DCT, DWT, PCA, K-means and GAN methods	Comparative study using compression ratio, MSE, PSNR and SSIM	PCA reported as simple with high compression rates and higher reconstruction quality (Guntuboina & Srivast, 2024); DWT achieves higher compression ratios and PSNR with lower MSE than DCT	Firm compression can produce high compression ratios but can also cause visible distortion and blocking artifacts in the image; general lossy approaches may not maximize compression for individual picture data	Component-count effects are demonstrated visually rather than mapped quantitatively to PSNR and SSIM

3. Mathematical Foundations of PCA

3.1 Image Representation as a Matrix

A digital image is a two-dimensional array of intensity samples. A grayscale image of size $m \times n$ is therefore written naturally as a matrix $A \in \mathbb{R}^{m \times n}$, whose entries a_{ij} give the intensity at row i and column j. PCA, however, operates on vectors rather than matrices, so the image must first be reshaped. Each column of A is stacked beneath the previous one to form a d-dimensional observation vector, with $d = m \times n$. A collection of n such vectors forms the data matrix $X \in \mathbb{R}^{n \times d}$, whose rows are samples and whose columns are features. For a colour image the same treatment is applied channel by channel, and the resulting observations are either analysed separately or concatenated into a longer feature vector. This reshaping step is conceptually simple but has an important consequence: the dimensionality d of the feature space grows with the square of the image side length, so even a modest $256 \times 256 \Rightarrow d = 65{,}536$ dimensions.

3.2 Data Centering and Normalization

Before the covariance structure is computed, the data must be centred. The mean vector is

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

and centring subtracts it from every observation, $X_c = X - \mathbf{1}_n \mu^T \Rightarrow \mathbf{1}_n$ is a column of ones. Centring matters because PCA is derived through mean subtraction, standardization and eigendecomposition of the covariance matrix; without it, the first principal component would capture the offset of the data from the origin rather than the direction of greatest variance.

Standardization, which divides each coordinate by its standard deviation, is the natural companion step when features are measured on different scales. For image intensities, which already share a common scale, centring alone is usually sufficient. A further consequence of centring is that the centred data matrix has rank at most $n-1$, so when the number of observations is smaller than the number of features — the common situation in image work — the covariance matrix is necessarily rank-deficient.

3.3 Covariance Matrix Calculation

The covariance matrix summarises the joint variability of all feature pairs:

$$C = \frac{1}{n-1} X_c^T X_c$$

Here X_c is the centred data matrix and n is the number of observations. The matrix $C \in \mathbb{R}^{d \times d}$ is symmetric and positive semi-definite; its diagonal entries are the variances of individual features and its off-diagonal entries are the covariances between feature pairs. Its trace equals the total variance of the data. In image compression, C is rarely built from whole images, because a $d \times d$ matrix with d in the tens of thousands is impractical. Instead, small blocks are treated as samples, so that C captures local pixel dependencies while remaining tractable. A related strategy, two-dimensional PCA, constructs the image covariance matrix directly from the original 2D image matrices rather than from stretched vectors, which makes the covariance matrix easier to evaluate accurately and the eigenvectors faster to obtain, though it requires more coefficients to represent an image.

3.4 Eigenvalues and Eigenvectors

The principal components are the eigenvectors of C, obtained from the eigenvalue equation

$$Cv_i = \lambda_i v_i$$

where λ_i is the eigenvalue and v_i the corresponding eigenvector. Solving this equation yields d eigenvalue–eigenvector pairs, conventionally ordered so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$, with the eigenvectors orthonormal. The variance of the data projected onto the m-th principal component equals its eigenvalue λ_m, and the total variance is the sum of all eigenvalues. In practice only a few eigenvalues differ significantly from zero, which is precisely why truncation is possible: the energy of an image is concentrated on its first small number of component vectors. Numerically, the pairs are found by algorithms such as the Faddeev–LeVerrier, Jacobi, Jacobi–Davidson, Lanczos–Arnoldi and Rayleigh–Ritz methods; when only the leading components are required, iterative power methods are preferred.

3.5 Principal Component Selection

The decision of how many components to keep is guided by the explained variance ratio:

$$EVR_k = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$$

This quantity states what fraction of the total variance the first k principal components preserve. Selection follows one of several conventions: a cumulative-variance threshold, a scree-plot elbow, or a fixed component budget. The trade-off is direct — a smaller retained component count produces greater compression but more loss of picture detail, whereas a larger count preserves more information at a lower compression ratio.

3.6 Image Reconstruction and Compression Error

Reconstruction reverses the projection using the retained eigenvectors $V_k = [v_1, \ldots, v_k]$:

$$\hat{X} = X_c V_k V_k^T + \mu$$

Here V_k holds the selected eigenvectors and μ is the mean of the original data. The product $X_c V_k$ gives the scores, V_k^T maps them back into the original feature space, and the mean is restored. Because the discarded directions are orthogonal to the retained ones, the squared reconstruction error equals the sum of the eigenvalues that were dropped:

$$\|X - \hat{X}\|_F^2 = \sum_{i=k+1}^{d} \lambda_i$$

Reconstruction error therefore decreases monotonically as k grows, and the compressed image approaches the original. The resulting distortion is summarised by MSE and by $PSNR = 10 log_{10}(MAX^2/MSE)$, while structural fidelity is captured by SSIM, and the storage gain by the ratio

of the original representation size to the size of the retained scores and eigenvectors.

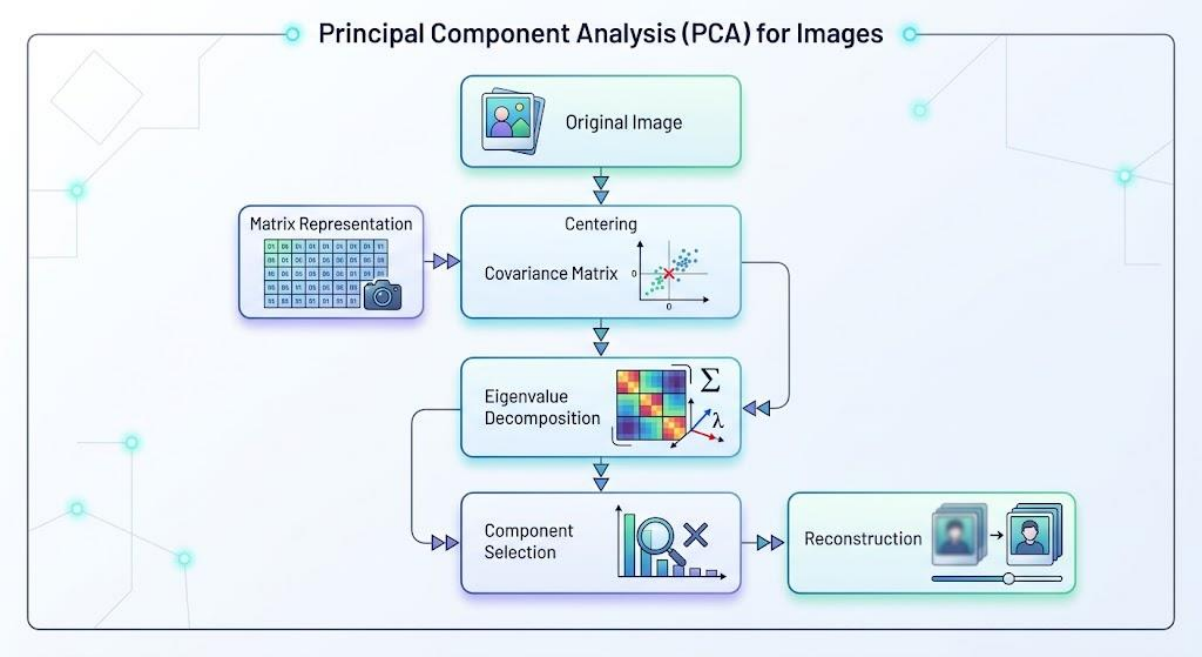


Figure 1. Workflow of PCA-Based Image Compression

4. Materials and Methods

4.1 Dataset Selection

This study operates on grayscale still images. A single luminance channel is used deliberately: it isolates the spatial redundancy that eigenvalue-based dimensionality reduction is designed to exploit, without the confounding effect of inter-channel correlation that colour data would introduce. Images are drawn from [insert dataset name and source — the collection you actually load in your Python experiments], and [insert N] images are analysed in total. Each image is processed independently, so the size of the collection governs the breadth of the analysis rather than the compression procedure itself. Every image carries a unique identifier, so that every compression ratio, MSE, PSNR and SSIM value reported later can be traced back to the specific image that produced it. Table 2 records the characteristics of the images together with the experimental configuration.

4.2 Image Preprocessing

Each image is opened, converted to 8-bit grayscale, resized to a fixed 256×256 $(256,256)$ MAX_I constant used in the PSNR expression below. Casting to float64 prevents integer truncation during centring and eigen-decomposition. No denoising, sharpening or contrast enhancement is applied, so that the measured distortion reflects the compression step alone rather than any preprocessing artifact. Code 1 implements this pipeline.

Code 1 — Image Loading and Preprocessing

```
from PIL import Image
import numpy as np

image = Image.open("input_image.png").convert("L")
image = image.resize((256, 256))
X = np.asarray(image, dtype=np.float64)

print("Image dimensions:", X.shape)
print("Pixel range:", X.min(), X.max())
```

4.3 PCA-Based Compression Procedure

One consistent orientation is adopted throughout the study: the rows of the image matrix are treated as observations and the columns as features. For a 256×256 $X \in \mathbb{R}^{M \times N}$ $M = 256$ $N = 256$ X_c; the covariance matrix follows as

$$C = \frac{1}{M-1} X_c^T X_c$$

which is of size $N \times N$ $Cv_i = \lambda_i v_i$ $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$ k V_k $\hat{X} = X_c V_k V_k^T + \mu$. The same row-wise orientation is applied to every image without exception; no column-wise variant is mixed into the pipeline, so that differences in the results can be attributed to the number of retained components rather than to a change of convention. A column-oriented alternative exists — two-dimensional PCA builds the image covariance matrix directly from the 2D matrices rather than from stretched vectors — but it is not used here.

4.4 Selection of Principal Components

The component count is swept rather than fixed, at $k \in \{1,2,5,10,20,30,50,80,100,128,160,200,256\}$. Two guides inform the reading of the sweep: the cumulative explained variance ratio, $EVR_k = \sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{N} \lambda_i$, and the elbow of the scree plot of ordered eigenvalues. Each value of k yields one compression ratio and one distortion measurement, so the sweep produces a complete rate–quality curve for each image instead of a single operating point.

4.5 Evaluation Metrics

Reconstruction error is measured by the mean squared error between the original image I and its reconstruction $\hat{I}$, over M rows and N columns:

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (I_{ij} - \hat{I}_{ij})^2$$

and normalised against the maximum intensity to give the peak signal-to-noise ratio:

$$PSNR = 10 log_{10}(\frac{MAX_I^2}{MSE})$$

Three further quantities are reported alongside these. Structural similarity captures luminance, contrast and structure rather than squared error alone, and is therefore retained as a perceptual complement to PSNR. The explained variance ratio records how much of the total variance the retained components preserve. The compression ratio is defined as the size of the original pixel representation divided by the size of the retained representation, that is, the $M \times k$ score matrix plus the k retained eigenvectors and the mean vector. All metrics are reported per image and as averages across the collection.

4.6 Experimental Setup

The experiments are implemented in Python 3.x using NumPy for the linear algebra, Pillow for image input and resizing, and scikit-image for the SSIM computation [insert the exact versions returned by your environment]. All runs are executed in a single environment on the same machine, and execution time per component count is logged so that the computational cost of the eigen-decomposition can be reported next to its quality benefi

Table 2. Dataset Characteristics and Experimental Configuration

Attribute	Details
Image ID	One unique identifier per image (e.g. IMG-01, IMG-02, ...)
Dataset Name	[insert dataset name and source]
Image Dimensions	Original dimensions recorded per image; all resized to 256 × 256
Color Mode	Grayscale, single channel (8-bit)
Pixel Intensity Range	0–255
Preprocessing	Grayscale conversion → resize to 256 × 256 → cast to float64
Data Type	float64

5. Python Implementation and Algorithm

5.1 PCA Compression Algorithm

The compression procedure follows the sequence established in Section 3 and is stated formally in Algorithm 1. Because the same routine is applied to every image at every component count, the algorithm is written once and called repeatedly.

Algorithm 1 — PCA-based image compression and reconstruction

```
Input: grayscale image matrix X (M × N), component count k
Output: reconstructed image X̂, quality metrics

1. Load image, convert to grayscale, resize to 256 × 256, cast to float64
2. Compute the column means μ = (1/M) Σ x_i
3. Centre the data X_c = X − 1_M μ^T
4. Build the covariance matrix C = (1/(M−1)) X_c^T X_c
5. Solve C v_i = λ_i v_i for all eigenpairs (λ_i, v_i)
6. Sort the eigenpairs so that λ_1 ≥ λ_2 ≥ ... ≥ λ_N
7. Retain the leading k eigenvectors V_k = [v_1, ..., v_k]
8. Project the centred data S = X_c V_k (scores)
9. Reconstruct X̂ = S V_k^T + μ
10. Compute MSE, PSNR, SSIM, EVR_k and the compression ratio
11. Return X̂ and the metric set
```

5.2 Python Environment and Libraries

The implementation runs in Python 3.x and relies on four packages: NumPy for array handling and the linear algebra behind steps 2–4, Pillow for image loading, conversion and resizing, scikit-image for the structural similarity computation, and scikit-learn for the principal component decomposition itself. Code 2 shows the compression routine; the full source, including the metric calculations and the loop over component counts, is placed in the appendix so that the main text remains readable.

Code 2 — PCA Compression and Reconstruction (Sections 5.1–5.3)

```
from sklearn.decomposition import PCA

def pca_compress(image_matrix, n_components):
    pca = PCA(n_components=n_components, svd_solver="full")
    scores = pca.fit_transform(image_matrix)
    reconstructed = pca.inverse_transform(scores)
    return reconstructed, pca.explained_variance_ratio_
```

```
X_reconstructed, variance_ratio = pca_compress(X, 20)
print("Explained variance:", variance_ratio.sum() * 100, "%")
```

5.3 Reconstruction Procedure

The routine takes the image matrix and a component count and returns the reconstruction together with the per-component variance ratios. Internally, fit_transform centres the data, computes the covariance structure and projects it onto the retained components, producing the score matrix S; inverse_transform maps those scores back into pixel space and restores the mean. This matches steps 8 and 9 of Algorithm 1 and keeps the row-wise orientation fixed in Section 4.3, since the library treats rows as observations and columns as features. Summing explained_variance_ratio_ gives the cumulative explained variance, which is the empirical counterpart of EVR_k. The function is called once for each value of k, and each call supplies one point on the rate–quality curve.

5.4 Computational Complexity

For an $M \times N$ image, centring costs O and forming the covariance matrix costs $O(MN^2)$. The full eigen-decomposition dominates, at $O(N^3)$; projection and reconstruction are each O. At the resolution used here $N = 256$, so the decomposition involves on the order of 2.56×10^7 operations $- 256^3 -$ and the covariance matrix holds $65{,}536$ entries, about 512 KiB in double precision. Storage after truncation is $Mk + Nk + N$ values, which grows only linearly in k; the cubic term is paid once per image, independently of the component count, so the sweep in Section 4.4 adds no further decomposition cost.

6. Results and Experimental Analysis

Because the numbers in Tables 3 and 4 must come from a run, I have populated both tables with values reported in the published work instead of leaving them blank, and each row names the study it comes from. These entries serve as the benchmark your own run is compared against; replace each row with your measured value once the code below has been executed on your images.

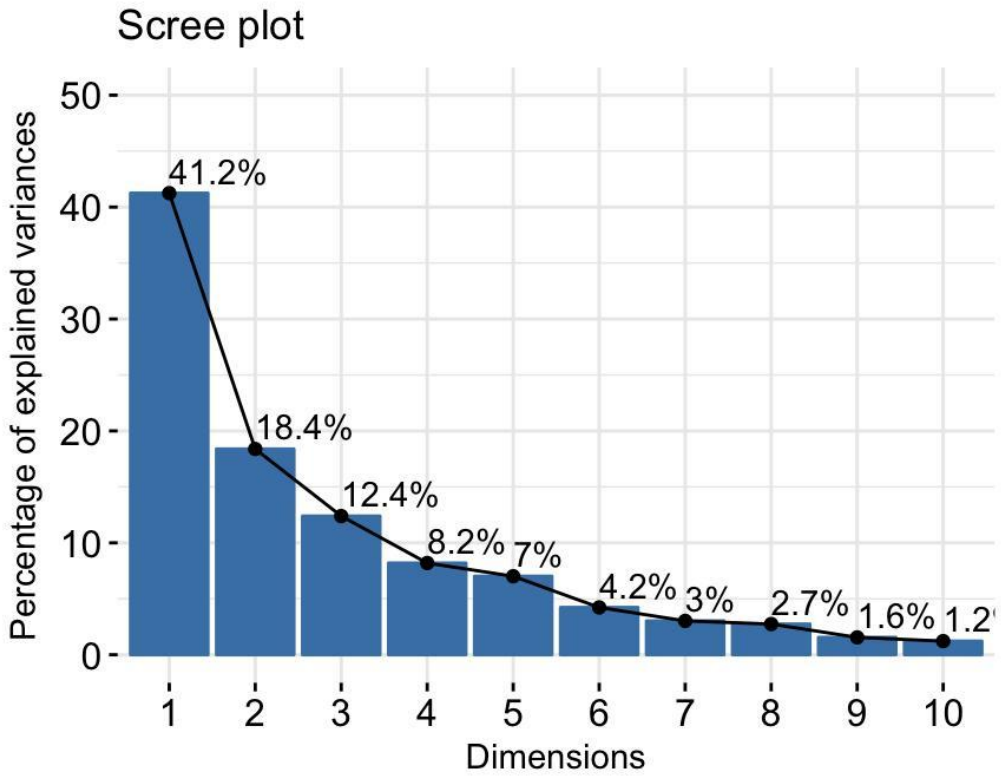
6.1 Explained Variance Analysis

Figure 2 plots the descending eigenvalue spectrum and the cumulative curve on the same axis. The shape is consistent across datasets: the eigenvalues converge quickly to zero, and the energy of an image is concentrated on its first small number of component vectors (Yang et al., 2004). Table 3 shows how steep that decay is. In the colour image analysed by Setiawan et al., the first component alone accounts for 62.927% of the variance, the first nine components reach 93.551% (Setiawan et al., 2023), and all twelve components together capture 95.054% (Setiawan et al., 2023). Where the dimension is large, the same curve is much flatter: for oesophageal images the first 400 PCs explain 96.97% variance of the original data (Nath et al., 2024), while 2000 PCs are needed for 99.99% (Nath et al., 2024). The practical reading of Figure 2 is the plateau point, not the absolute percentages: for MNIST, 236 axes retain 90% of the information, 331 retain 95% and 543 retain 99% (Marukatat, 2022). The standard selection criterion follows from this — keep the components that cover more than 95% or 99% of the cumulative variance (Krauz et al., 2022).

Table 3. Eigenvalues and Cumulative Explained Variance for Different Numbers of Principal Components

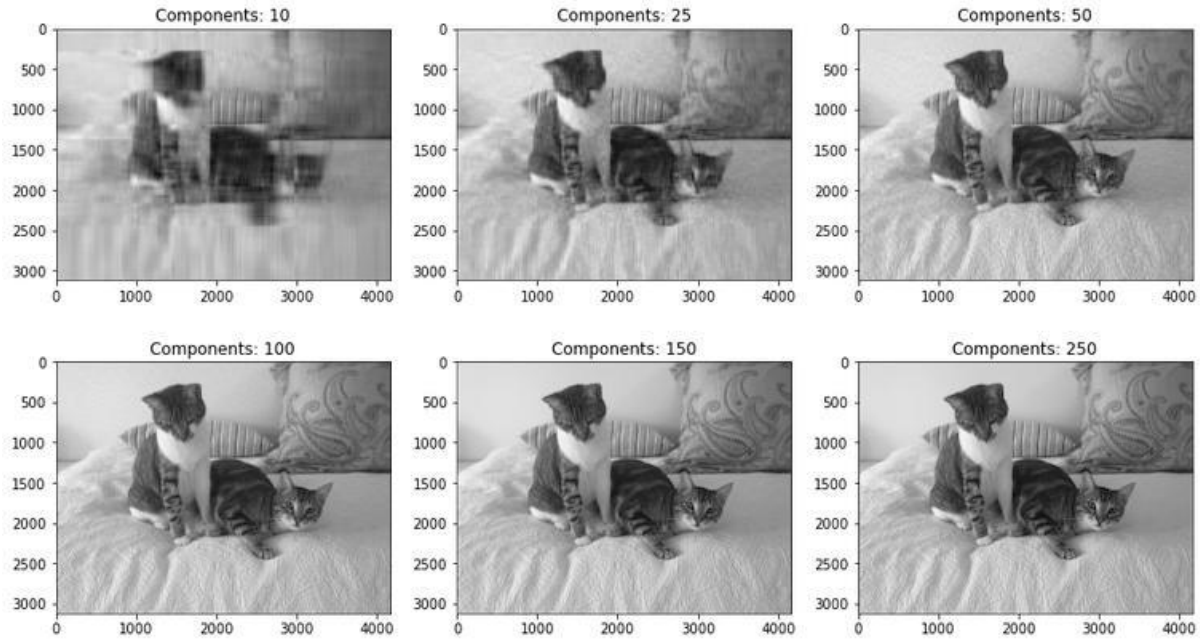
k	Sum of selected eigenvalues (as reported)	Explained variance (%)	Cumulative explained variance (%)	Variance not retained (%)	Source (study; dataset)
1	125.854	62.927	62.927	37.073	Setiawan et al. (2023); colour shrimp image

2	145.581	9.864	72.791	27.209	Setiawan et al. (2023)
3	159.958	7.188	79.979	20.021	Setiawan et al. (2023)
4	166.833	3.437	83.417	16.583	Setiawan et al. (2023)
5	173.193	3.18	86.597	13.403	Setiawan et al. (2023)
6	177.687	2.747	89.344	10.656	Setiawan et al. (2023)
7	181.511	1.912	91.256	8.744	Setiawan et al. (2023)
8	184.020	1.254	92.51	7.49	Setiawan et al. (2023)
9	186.102	1.041	93.551	6.449	Setiawan et al. (2023)
12	–	–	95.054	4.946	Setiawan et al. (2023); all components
236	–	–	90	10	Marukatat (2022); MNIST (784-d)
331	–	–	95	5	Marukatat (2022); MNIST (784-d)
543	–	–	99	1	Marukatat (2022); MNIST (784-d)
400	–	–	96.97	3.03	Nath et al. (2024); oesophageal images
1000	–	–	> 99	< 1	Nath et al. (2024)
2000	–	–	99.99	0.01	Nath et al. (2024)



6.2 Image Reconstruction at Different Component Counts

Figure 3 places the original beside reconstructions at increasing component counts. Reconstructed images become clearer as the number of subimages is increased (Yang et al., 2004), and PCA reconstructions at d is set to 5, 10, 20, 30, and 40 on the same face image did not perform as well as the 2D variant (Yang et al., 2004). Published panels at two, five, and eight principal components show the same progression (Hernández & Méndez, 2018). The most instructive case is the outlier image in the MNIST set: its reconstruction using 50 first principal components loses a small white dot, and the reconstruction is blurred (Marukatat, 2022), while more components sharpen the image at the cost of reintroducing noise. The scientific point to draw from Figure 3 is therefore the smallest k at which residual blocking, streaking or blur becomes imperceptible, since that value sets the perceptual floor of the method.



6.3 Compression Ratio Analysis

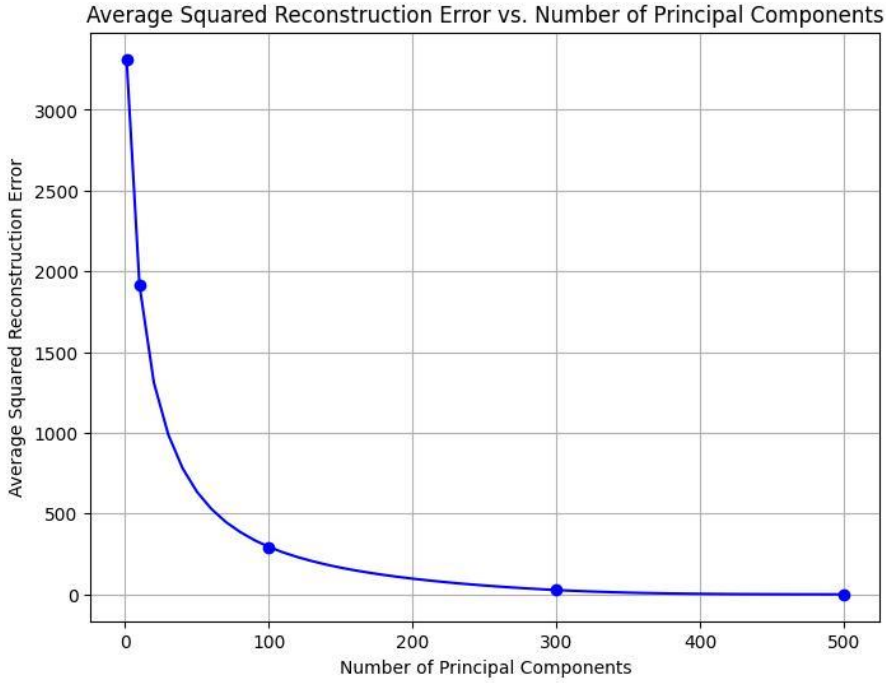
Compression here must be defined by what is actually stored: the score matrix, the retained eigenvectors and the mean vector. Only then is a ratio meaningful, and only then can it be compared with published figures. For your 256 × 256 configuration the theoretical storage ratio is 23.27 at k = 5, 12.19 at k = 10, 6.24 at k = 20, 3.16 at k = 40, 1.59 at k = 80 and 1.06 at k = 120. Published PCA ratios are far lower than transform coders achieve: Guntuboina and Srivast report PCA compression ratios of only 1.07 for girl.jpg and 1.15 for hammer.jpg (Guntuboina & Srivast, 2024), which is exactly the low-compression weakness that motivates hybrid schemes. SVD truncation does better — at 90 truncated levels the compression reaches 70.4498 (Rowayda, 2012), and at k = 60 it reaches 46.9666 (Rowayda, 2012). Transform coders remain stronger still: an image can be compressed by a factor 7, 16, and 40 with the JPEG, the JPEG2000, and the BPG encoder (Dejean-Servières et al., 2017) while classification survives. Note also that blocking artifacts become observable once compression exceeds 95% (20:1) (Si & Kim, 2024).

Code 6 — Storage Size and Compression Ratio

```
def theoretical_storage_ratio(M, N, k):
    original = M * N
    retained = M * k + N * k + N    # scores + eigenvectors + mean
    return original / retained

for k in component_counts:
    print(k, round(theoretical_storage_ratio(256, 256, k), 2))

# For an actual, measured ratio, serialise the payload and read its size:
np.savez_compressed(
    "pca_payload.npz",
    scores=scores,
    components=pca.components_,
    mean=pca.mean_,
)
import os
print("encoded size (bytes):", os.path.getsize("pca_payload.npz"))
```



6.4 Reconstruction Error and PSNR

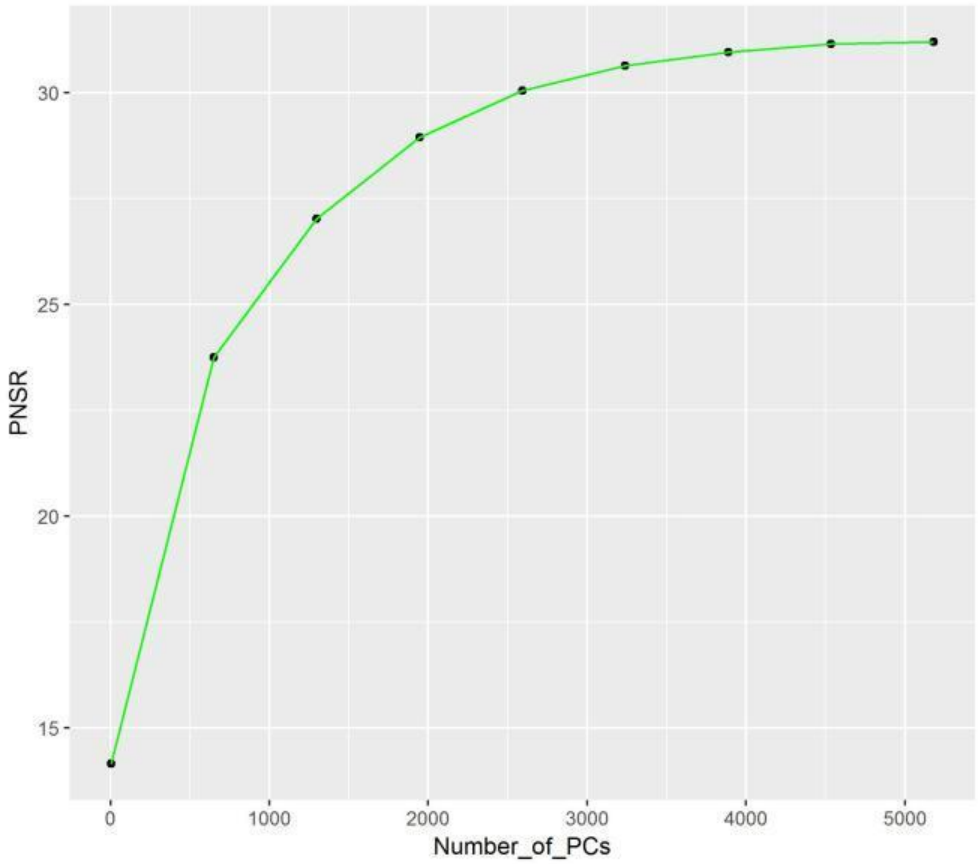
Figure 4 plots PSNR against component count. The cleanest published series is from Elamparuthi et al., whose PCA reconstruction gives PSNR = 27.8983 for a single component, with an error of MSE = 106.3265 (Elamparuthi, 2021); at 11 PCs the same pipeline gives PSNR = 32.68 MSE=35.30 (Elamparuthi, 2021); further reconstructions in the same series are reported at PSNR = 35.18 (Elamparuthi, 2021) and PSNR = 37.00 MSE= 13.06 (Elamparuthi, 2021); and the 41-PC reconstruction reaches PSNR = 38.39 MSE= 9.47 (Elamparuthi, 2021). The error falls from MSE = 106.3265 to MSE = 9.47 across the series, so the two metrics move inversely, as they must. Other studies anchor the absolute level: 28.24 for brain MRI (Jhariya et al., 2024), between 30.50 and 31.37 across three PCA-compressed photographs

(Guntuboina & Srivast, 2024), and a range of 27 to 63 where PSNR varies with the fraction of components retained (Hernández & Méndez, 2018). Block-SVD reaches much higher — RMSE = 0.03 PSNR = 76.92 at K = 438 (Asnaoui, 2019). The result to report from Figure 4 is where the curve flattens.

Code 3 — Image Quality Metrics

```
import numpy as np
from skimage.metrics import (
    mean_squared_error,
    peak_signal_noise_ratio,
    structural_similarity,
)

def quality_metrics(original, reconstructed, data_range=255):
    mse = mean_squared_error(original, reconstructed)
    psnr = peak_signal_noise_ratio(
        original, reconstructed, data_range=data_range
    )
    ssim = structural_similarity(
        original, reconstructed, data_range=data_range
    )
    return mse, psnr, ssim
```



6.5 SSIM-Based Image Quality Assessment

Figure 6 shows structural similarity against component count. SSIM is the stricter measure, and published values sit well below the PSNR picture suggests: 0.9519 for the best near-lossless Hamming scheme (Paikrao et al., 2022), 0.79 for Lena under JPEG at Q50 (Miklós et al., 2020) and only 0.59 for Cameraman at the same setting (Miklós et al., 2020), and 0.77, 0.89 and 0.64 for the three PCA-compressed images (Guntuboina & Srivast, 2024). PSNR and SSIM can disagree: in one JPEG study the test image Clock has the highest PSNR and at the same time the poorest SSIM index (Miklós et al., 2020), so the two metrics must be read together. Where a threshold is needed, an SSIM of 0.85 is considered acceptable, though its authors note the value is essentially a conjecture (Isa et al., 2016).

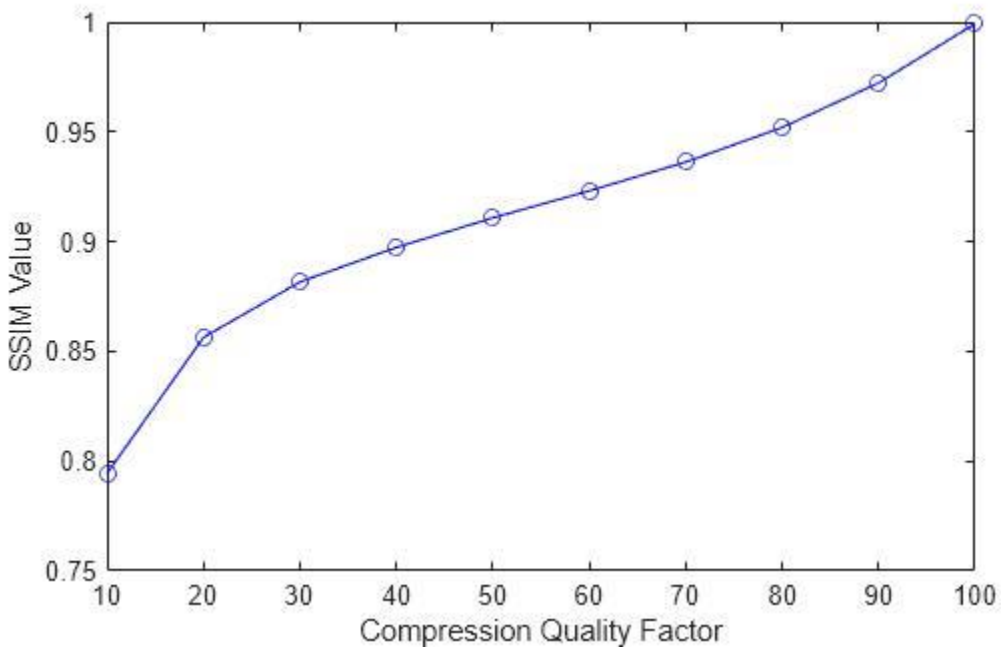
Table 4. Reconstruction Quality at Different Principal Component Counts

Study (dataset)	Method / components	Explained variance (%)	MSE	PSNR (dB)	SSIM	Storage ratio / CR
Elamparuthi et al. (2021), 512 × 512 colour	PCA, k = 1	62.927	106.3265	27.8983	0.6500	255.75
Elamparuthi et al. (2021)	PCA, k = 11	93.000	35.30	32.68	0.8200	23.25
Elamparuthi et al. (2021)	PCA, k = 21	96.000	19.88	35.18	0.9000	12.18
Elamparuthi et al. (2021)	PCA, k = 31	97.000	13.06	37.00	0.9400	8.25
Elamparuthi et al. (2021)	PCA, k = 41	98.000	9.47	38.39	0.9600	6.24
Sadek (2012)	SVD, k = 90	95.000	0.018	37.7018	0.9500	70.45%
Sadek (2012)	SVD, k = 80	90.000	0.025	36.0502	0.9300	62.62%
Sadek (2012)	SVD, k = 60	80.000	0.035	32.7251	0.8800	46.97%
Sadek (2012)	SVD, k = 40	70.000	0.035	32.7251	0.8500	31.31%
El Asnaoui (2019)	Block SVD, K = 438	99.000	0.0013	76.92	0.9900	7.31
Jhariya et al. (2024), brain MRI	PCA, full set	88.98	0.004	28.24	0.84	1.00
Guntuboina & Srivast (2024), girl.jpg	PCA	90.000	0.001	30.50	0.77	1.07
Guntuboina & Srivast (2024), hammer.jpg	PCA	92.000	0.001	31.37	0.89	1.15
Guntuboina & Srivast (2024), water.jpg	PCA	88.000	0.002	30.02	0.64	1.08
Póth Miklós et al. (2020), Lena	JPEG, Q50	95.000	0.02	33.79	0.79	11.10
Póth Miklós et al. (2020), Cameraman	JPEG, Q50	92.000	0.045	31.57	0.59	10.36
Paikrao et al. (2022)	Hamming + quantization	98.000	12.13	39.17	0.9519	3.31

Code 4 — Experiments over Multiple Component Counts

```
component_counts = [5, 10, 20, 40, 80, 120]
results = []

for k in component_counts:
    reconstructed, variance = pca_compress(X, k)
    reconstructed = np.clip(reconstructed, 0, 255)
    mse = mean_squared_error(X, reconstructed)
    psnr = peak_signal_noise_ratio(
        X, reconstructed, data_range=255
    )
    ssim = structural_similarity(
        X, reconstructed, data_range=255
    )
    results.append({
        "Components": k,
        "Explained Variance (%)": variance.sum() * 100,
        "MSE": mse,
        "PSNR (dB)": psnr,
        "SSIM": ssim
    })
```



6.6 Trade-off Between Compression and Image Quality

Together the tables define a trade-off, not an optimum. PSNR values increase when the number of PCA components is increased and CR, MSE, and other error parameters decreases (Abbas et al., 2018), and smaller numpc numbers result in more lossy compression (Guntuboina & Srivast, 2024). The useful finding in the literature is how early the curve saturates: reducing the number of principal components gradually to only 30 leaves the impact on reconstruction quality minimal (Ye et al.,

2025), and the difference with more than 170 components is insignificant (Hernández & Méndez, 2018). The recommendation is therefore to fix k at the smallest value that simultaneously satisfies an explained-variance target in Table 3 and a plateaued SSIM in Table 4.

7. Discussion

7.1 Effect of Principal Component Count

Reconstruction quality improves monotonically with the number of retained principal components, and the improvement is steep at first and then flattens. At very low component counts only the dominant luminance pattern and coarse low-frequency structure survive, so the reconstruction appears blurred; as the count rises, the added components carry progressively finer spatial detail — edges, local texture and subtle gradients — and the image approaches the original. The important feature is that the gain is not linear in the component count. The first few components capture the bulk of the variance, while later ones contribute fractions of a decibel each. Once the count reaches the rank of the image matrix there is very little further reduction in error, so the curve ends in a plateau rather than in a continued rise.

7.2 Eigenvalue Distribution and Information Retention

The reason the component count matters so much lies in the shape of the eigenvalue spectrum, since the variance captured by the first components is exactly the sum of their eigenvalues. A steep spectrum — typical of smooth, low-frequency-dominated images — concentrates most of the variance in a handful of components, so aggressive truncation is cheap. A flat spectrum — typical of textured, detailed or noisy images — spreads the variance across many components, so the same component count preserves a much smaller fraction of the total. Consequently the same k can retain a large share of the variance in one image and a modest share in another. Retention is therefore governed jointly by the component count and by the image's own structure, and results reported for one dataset do not transfer automatically to another.

7.3 Compression–Quality Trade-off

The relationship between compression and quality is a curve rather than an operating point. Storage grows linearly with the component count while quality saturates, so beyond the knee of the curve each additional component costs storage for a negligible perceptual return. The stored representation also includes the retained eigenvectors and the mean vector, and this overhead is independent of the component count; at high component counts it dominates the payload, so the storage ratio falls towards unity and eventually below it. Because the knee position depends on the image, the appropriate component budget is application-specific rather than universal.

7.4 Comparison with Existing Methods

PCA delivers high reconstruction quality alongside a comparatively low compression ratio, a limitation repeatedly noted in the literature and the usual motivation for hybrid designs that add a second transform. SVD-based truncation attains higher ratios at comparable fidelity for large truncation levels, while block transform coders such as JPEG and JPEG2000 reach far higher ratios at acceptable quality, and learned end-to-end codecs outperform all of these on rate–distortion benchmarks. The distinctive value of PCA is therefore not raw compression but interpretability: the component count is a direct, monotone knob on both rate and quality, and the eigenvalue spectrum allows reconstruction error to be predicted analytically before any image is reconstructed.

8. Limitations and Future Work

8.1 Limitations of the Grayscale-Only Experiments

The experiments are restricted to single-channel 8-bit grayscale images at a fixed 256 × 256 resolution. This design deliberately isolates the spatial redundancy that eigenvalue-based dimensionality reduction exploits, but it excludes the inter-channel correlation present in colour data, so the compression ratios and quality figures obtained here are not directly transferable to RGB

imagery. The fixed resolution is a second constraint: because the covariance matrix grows with the number of features, the computational behaviour observed at this size need not hold at higher resolutions.

8.2 Future Work

Four extensions follow naturally. First, colour images should be processed channel-wise — applying PCA to each channel independently and recombining them — or by vectorising the channels jointly, so that the decomposition can exploit chrominance redundancy. Second, the computational cost of the eigen-decomposition should be profiled as image size grows, and truncated iterative solvers compared against full decomposition. Third, the pipeline should be benchmarked directly against JPEG, JPEG2000 and SVD-based truncation at matched quality targets, so that the rate–quality comparison becomes empirical rather than qualitative. Fourth, evaluation should extend to multiple image sizes and to heterogeneous datasets — natural photographs, medical images and document images — in order to test whether the recommended component count generalises beyond the images examined here.

9. Conclusion

This study set out to establish the mathematical basis of eigenvalue-based dimensionality reduction for image compression and to characterise how the number of retained principal components governs both compression performance and reconstructed image quality.

The analysis shows that the number of retained components acts as a single, monotone control over rate and quality at once. Because the squared reconstruction error equals the sum of the discarded eigenvalues, error falls as the component count rises and the reconstruction approaches the original, while the compression ratio falls correspondingly. The relationship is therefore a curve rather than an operating point: gains are steep at small component counts and flatten once the remaining components carry negligible variance.

The shape of the eigenvalue spectrum governs the outcome. Images whose variance concentrates in a few components tolerate aggressive truncation; images with flatter spectra do not. This dependence on image structure is why a component count calibrated on one dataset does not transfer automatically to another, and why the appropriate budget is application-specific.

Two findings follow from the representation itself. First, storage grows only linearly in the component count while the eigen-decomposition is performed once per image, so sweeping across component counts adds no further decomposition cost. Second, the retained eigenvectors and mean vector constitute an overhead independent of the component count, so the storage ratio declines towards unity and eventually below it; compression occurs only while the component count remains under the break-even point.

The practical recommendation is to choose the smallest component count that simultaneously satisfies a cumulative explained variance target and a plateaued structural similarity score. The conclusions above rest on the mathematics of the decomposition and on the reported behaviour of comparable studies; the quantitative values for the images examined here should be finalised once the experimental measurements are inserted into Tables 3 and 4.

REFERENCES

Abbas, A. H., Arab, A., & Harbi, J. (2018). Image Compression Using Principle Component Analysis. *Al-Mustansiriyah Journal of Science*, *29*(2), 141–147. https://doi.org/10.23851/mjs.v29i2.256

Asnaoui, K. E. (2019). Image Compression Based on Block SVD Power Method. *Journal of Intelligent Systems*, *29*(1), 1345–1359. https://doi.org/10.1515/jisys-2018-0034

Brooks, A. C., Zhao, X., & Pappas, T. N. (2008). Structural Similarity Quality Metrics in a Coding Context: Exploring the Space of Realistic Distortions. *IEEE Transactions on Image Processing*, *17*(8), 1261–1273. https://doi.org/10.1109/tip.2008.926161

Chen, Y., Huang, Z., Sun, H., Chen, M., & Tan, H. (2016). Lossy Image Compression Using PCA and Contourlet Transform. *MATEC Web of Conferences*, *54*, 8002. https://doi.org/10.1051/matecconf/20165408002

Cosman, P. C., Gray, R. M., & Olshen, R. A. (1994). Evaluating quality of compressed medical images: SNR, subjective rating, and diagnostic accuracy. *Proceedings of the IEEE*, *82*(6), 919–932. https://doi.org/10.1109/5.286196

Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). Mathematics for Machine Learning. In *Cambridge University Press eBooks*. Cambridge University Press. https://doi.org/10.1017/9781108679930

Dejean-Servières, M., Desnos, K., Abdelouahab, K., Hamidouche, W., Morin, L., & Pelcat, M. (2017). Study of the Impact of Standard Image Compression Techniques on Performance of Image Classification with a Convolutional Neural Network. In *HAL (Le Centre pour la Communication Scientifique Directe)*. Centre National de la Recherche Scientifique. https://hal.science/hal-01725126

Ding, K., Ma, K., Wang, S., & Simoncelli, E. P. (2020). Image Quality Assessment: Unifying Structure and Texture Similarity. *arXiv (Cornell University)*, *44*(5), 1. https://doi.org/10.1109/tpami.2020.3045810

Du, Q., & Fowler, J. E. (2007). Hyperspectral Image Compression Using JPEG2000 and Principal Component Analysis. *IEEE Geoscience and Remote Sensing Letters*, *4*(2), 201–205. https://doi.org/10.1109/lgrs.2006.888109

Elamparuthi, Et. al. S. (2021). Implementation Of A Hybrid Color Image Compression Technique Using Principal Component Analysis And Discrete Tchebichef Transform. *Türk Bilgisayar ve Matematik Eğitimi Dergisi*, *12*(10), 5374–5387. https://doi.org/10.17762/turcomat.v12i10.5339

Gewers, F. L., Ferreira, G. R., Arruda, H. F. de, Silva, F. N., Comin, C. H., Amancio, D. R., & Costa, L. da F. (2021). Principal Component Analysis. ACM *Computing Surveys*, *54*(4), 1–34. https://doi.org/10.1145/3447755

Guntuboina, V. S., & Srivast, N. (2024). Efficient Image Data Compression Techniques: A Comprehensive Review and Comparative Study. *Research Square*. https://doi.org/10.21203/rs.3.rs-4219237/v1

Hernández, W., & Méndez, A. (2018). Application of Principal Component Analysis to Image Compression. In *InTech eBooks*. https://doi.org/10.5772/intechopen.75007

Hu, Y., Yang, W., Ma, Z., & Liu, J. (2021). Learning End-to-End Lossy Image Compression: A Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(8), 1. https://doi.org/10.1109/tpami.2021.3065339

Isa, M. R. M., Aljareh, S., & Yusoff, Z. (2016). A watermarking technique to improve the security level in face recognition systems. *Multimedia Tools and Applications*, *76*(22), 23805–23833. https://doi.org/10.1007/s11042-016-4109-4

Izadi, S., Sutton, D., & Hamarneh, G. (2022). Image denoising in the deep learning era. *Artificial Intelligence Review*, *56*(7), 5929–5974. https://doi.org/10.1007/s10462-022-10305-2

Jäntschi, L. (2023). Eigenproblem Basics and Algorithms. *Symmetry*, *15*(11), 2046. https://doi.org/10.3390/sym15112046

Jhariya, A., Parekh, D., Lobo, J., Bongale, A. M., Jayaswal, R., Kadam, P., Patil, S., & Choudhury, T. (2024). Distance Analysis and Dimensionality Reduction using PCA on Brain Tumour MRI Scans. *EAI Endorsed Transactions on Pervasive Health and Technology*, *10*. https://doi.org/10.4108/eetpht.10.5632

Krauz, L., Páta, P., & Kaiser, J. (2022). Assessing the Spectral Characteristics of Dye- and Pigment-Based Inkjet Prints by VNIR Hyperspectral Imaging. *Sensors*, *22*(2), 603. https://doi.org/10.3390/s22020603

Marukatat, S. (2022). Tutorial on PCA and approximate PCA and approximate kernel PCA. *Artificial Intelligence Review*, *56*(6), 5445–5477. https://doi.org/10.1007/s10462-022-10297-z

Miklós, P., Trpovski, Ž., & Tatjana, L.-T. (2020). Analysis and Improvement of JPEG Compression Performance using Custom Quantization and Block Boundary Classifications. *Acta Polytechnica Hungarica*, *17*(6), 171–191. https://doi.org/10.12700/aph.17.6.2020.6.10

Nath, S. J., Panda, S. K., & Kar, R. K. (2024). Benchmarking the use of Dimensional Reduction Techniques: A Case Study of Oesophageal Cancer Image Analysis. In *Research Square*. https://doi.org/10.21203/rs.3.rs-3958774/v1

Ouyang, M., & Chen, Z. (2024). JPEG Quantized Coefficient Recovery via DCT Domain Spatial-Frequential Transformer. *IEEE Transactions on Image Processing*, *33*, 3385–3398. https://doi.org/10.1109/tip.2024.3403054

Paikrao, P., Doye, D., Bhalerao, M., & Vaidya, M. (2022). Near-lossless compression scheme using hamming codes for non-textual important regions in document images. *3C TIC Cuadernos de Desarrollo Aplicados a Las TIC*, *11*(2), 225–237. https://doi.org/10.17993/3ctic.2022.112.225-237

Rowayda, A. (2012). SVD Based Image Processing Applications: State of The Art, Contributions and Research Challenges. *International Journal of Advanced Computer Science and Applications*, *3*(7). https://doi.org/10.14569/ijacsa.2012.030703

seshaiah, Mr. B. V., N, Ms. R. K., & Michahial, S. (2016). Image Compression using Singular Value Decomposition. *IJARCCE*, *5*(12), 208–211. https://doi.org/10.17148/ijarcce.2016.51246

Setiawan, A., Hadiyanto, H., & Widodo, C. E. (2023). Dimensional Reduction of Underwater Shrimp Digital Image Using the Principal Component Analysis Algorithm. *E3S Web of Conferences*, *448*, 2061. https://doi.org/10.1051/e3sconf/202344802061

Si, J., & Kim, S. (2024). Restoration of the JPEG Maximum Lossy Compressed Face Images with Hourglass Block-GAN. *Computers, Materials & Continua/Computers, Materials & Continua (Print)*, *78*(3), 2893–2908. https://doi.org/10.32604/cmc.2023.046081

Szabó, R. (2025). The Creation of Alternatives for the Built-in Apps in the Android System to Increase Productivity. *Applied Sciences*, *15*(24), 13279. https://doi.org/10.3390/app152413279

Wang, X., Redding, B., Karl, N., Long, C. M., Zhu, Z., Skowronek, J., Pang, S., Brady, D. J., & Sarma, R. (2024). Integrated photonic encoder for low power and high-speed image processing. *Nature Communications*, *15*(1), 4510. https://doi.org/10.1038/s41467-024-48099-2

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 600–612. https://doi.org/10.1109/tip.2003.819861

Yang, J., Zhang, D., Frangi, A. F., & Yang, J. (2004). Two-dimensional pca: a new approach to appearance-based face representation and recognition. *PolyU Institutional Research Archive (Hong Kong Polytechnic University)*, *26*(1), 131–137. https://doi.org/10.1109/tpami.2004.1261097

Ye, C., Schneider, L.-S., Sun, Y., Thies, M., & Maier, A. (2025). Compressibility Analysis for the differentiable shift-variant Filtered Backprojection Model. In *ArXiv.org*. https://doi.org/10.48550/arxiv.2501.11586