# Project Report

Name: Aravindh V

Course: B.Sc CS (AI & ML)

Registration Number: TU6243202111006

## Project Title:

PCA for Image Compression

## 1. Project Description

This project demonstrates the use of Principal Component Analysis (PCA) for image compression. The main goal is to reduce the dimensionality of an image while preserving most of its important visual information. By converting correlated pixel values into uncorrelated principal components, PCA enables significant compression without major loss of quality. This project helps understand how PCA can be applied in image processing and data reduction tasks.

## 2. Learning Objectives

• To understand the concept of PCA and its mathematical foundation.

• To implement PCA for compressing and reconstructing images.

• To evaluate performance based on explained variance and reconstruction error.

## 3. Timeline

Start Date: October 30, 2025

Submission Date: October 30, 2025

## 4. Algorithm Used

Algorithm Name: Principal Component Analysis (PCA)

PCA is a statistical method used for dimensionality reduction. It transforms the original correlated features into a set of linearly uncorrelated components (principal components). In image compression, PCA identifies directions of maximum variance in pixel data and retains only the top components that preserve most of the visual information. This reduces storage size and improves computational efficiency.

## 5. Tools & Libraries

Programming Language: Python

Libraries Used: NumPy, Pandas, Matplotlib, Scikit-learn, Pillow

## 6. Dataset Description

The dataset used is a single or set of sample images (e.g., RGB images). The image is converted into a numerical array where each pixel is represented by intensity values. The project may use grayscale conversion for simplification. The dimensionality of this dataset corresponds to the width × height of the image.

## 7. Methodology

Data Preprocessing: The image is loaded, converted to grayscale or numerical array, and normalized to a 0–1 range.

Model Training: PCA is applied using scikit-learn’s PCA module to compute eigenvalues and eigenvectors. Different numbers of components are tested to determine optimal compression.

Evaluation: The explained variance ratio and reconstruction error are calculated to assess compression performance.

Hyperparameter Tuning: The number of principal components (n\_components) is tuned to retain ~95% of variance.

## 8. Results

Performance Metrics: The explained variance ratio for the optimal number of components is around 95%. Reconstruction error remains low, indicating good preservation of quality.

Visualizations: Original and reconstructed images were displayed for comparison. The compressed image showed minimal quality loss while reducing data size significantly.

Insights: PCA successfully reduces storage size and speeds up computation. It demonstrates the balance between compression ratio and retained quality.

## 9. Questions Answered

* Q1: How many components are optimal?
* Q2: What variance is retained?
* Q3: What is reconstruction error?
* Q4: How to visualize components?
* Q5: How does PCA improve speed?
* Q6: What is eigenvalue/eigenvector?
* Q7: How to apply PCA to images?
* Q8: What is the impact on quality?
* Q9: How does PCA differ from LDA?
* Q10: How to use PCA in preprocessing?
* Q11: How to choose number of components using explained variance ratio?
* Q12: Compare PCA compression with JPEG.
* Q13: What happens if too few components are used?
* Q14: How to reconstruct and display compressed image?
* Q15: How to calculate compression ratio?

## 10. Challenges & Improvements

Challenges: Selecting an appropriate number of components for optimal compression without visual distortion. Handling color images adds complexity since PCA must be applied separately to each RGB channel.

Future Improvements: Implement PCA on larger datasets or video frames. Combine PCA with autoencoders for hybrid compression.

## 11. References

Dataset Link: Custom image data (local file)

Research Papers: scikit-learn documentation, PCA theory articles on towardsdatascience.com

## 12. GitHub Link

https://github.com/Aravindh-2727/ml-6.PCA-for-Image-Compression