AL KUT JOURNAL OF ECONOMIC AND ADMINISTRATIVE SCIENCES
Publisher: College of Economics and Management - Wasit University

Application of Advanced Statistical Methods in Fingerprint Image Encryption and Decryption : Implementation and Evaluation of PCA and SVD Algorithms

Ali Jassem Mohammed[(1)] Assma Ghalib Jaber[(2)]

[1, 2] Department of Statistics, College of Administration and Economics, University of Baghdad, Baghdad, IRAQ

ali.jassem2201p@coadec.uobaghdad.edu.iq Drasmaa.ghalib@coadec.uobaghdad.edu.iq

Abstract

This work uses two advanced statistical methods, principal component analysis (PCA) and singular value decomposition (SVD), to encrypt and decrypt sensitive biometric fingerprint images during transmission or storage. The basic encryption method for principal component analysis involves using the dimension reduction property to project image data onto a space with fewer dimensions than the uncorrelated principal components, thereby compressing and securely concealing the original information the original information while retaining the essential features. singular value decomposition is used as a powerful support method for analysing the matrices resulting from the digital image matrix in this encryption method. These transformations are reversed by decryption algorithms to restore the original image. The algorithms rely on the use of a secret key to prevent illegal access and ensure confidentiality and integrity. MATLAB 2024b was used to implement and run the PCA and SVD algorithms. Performance was thoroughly tested based on common comparison metrics, such as histogram analysis, pixel correlation, SSIM, PSNR, NPCR, MSE, and standard image statistics. After a comprehensive analysis, the most efficient method for encrypting and decrypting fingerprint images was identified using PCA and SVD, with SVD showing superior performance in terms of the criteria evaluated in the previous analysis.

Keywords

Digital Images, Fingerprint, Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Encryption, Decryption, Statistical Analysis

Introduction

With the development of digital technology, multimedia transmission of data, such as audio, pictures, and videos, has become the norm. However, transmitting such important information on open networks is vulnerable to attacks. In order to secure this data, encryption procedures have been introduced to give secrecy and protection from unauthorized access. Fingerprint images are primary biometric identifiers containing sensitive personal data, which require robust encryption and decryption. The use of an appropriate secret key prevents illegal access, maintaining confidentiality and integrity in the biometric data during transmission and storage. The application of statistical techniques in encrypting and decrypting digital images is also highlighted in this study, focusing on their application in enhancing data security and enhancing encryption efficiency. One of the efficient methods employed for securing multimedia data is encryption, whereby the information is encoded into an unreadable state by using secret keys so that it becomes difficult for unauthorized parties to modify or access the data [18]. Digital images being an important form of multimedia data typically contain highly sensitive information such as financial, military, medical, or government information [4]. There is a need to send such images safely to prevent unauthorized use and security violations. Robust encryption techniques are also required to be developed to preserve the privacy of data embedded in digital images [11].

Principal Component Analysis (PCA) is an extremely common statistical dimension-reduction technique, particularly when applied to the study of complex datasets such as image encryption and processing. PCA works by reducing high-dimensional data into lower-dimensional uncorrelated variables known as principal components without discarding most of the variability of the initial data. This makes it easier to analyze later on, thus being highly effective in image encryption. It reduces the dimensions of the image while preserving useful information [10]. PCA has been extensively used for noise reduction and data compression of atmospheric data [13], and it has been shown that combining classifiers with similarity

measures, such as geometric and harmonic similarity, enhances performance [8]. Other studies have scanned the use of PCA for comparison of the performance of various dimensionality reduction methods [17], with Singular Value Decomposition (SVD) being one of the most potent methods in conjunction with PCA [9]. Furthermore, PCA is usually discussed in overall reviews of dimensionality reduction techniques [12] and is normally introduced as a rudimentary tool for data analysis [14]. Specifically, embedding PCA in image encryption offers a high-efficiency data compression method without sacrificing security, and has been proved to be a proper approach in handling huge multimedia data [10].

Singular value decomposition (SVD) has also been recognized as a powerful image encryption tool as it decomposes images into individual values that essentially describe them. Higher computational security and efficiency are realized because selectively encrypting individual values is focused towards targeting the most critical parts of the digital image. Optical color image encryption based on Arnold transform and SVD for disassembling the spatial structure of images and secure transmission without recognition [2]. Combining SVD with an Arnold transform within a fractional domain can significantly ease data complexity without compromising encryption power [3]. Additionally, fractional SVD has been utilized and demonstrated the efficiency of the approach in selective encryption where only required portions are encrypted for the purpose of optimization [16]. SVD coupled with the Cat Map and Fractional Fourier Transform has been shown to enhance confusion-diffusion in the encryption process [7]. Adaptive transparent encryption techniques have been proposed, demonstrating various benefits of SVD with spatial encryption, highlighting its application in more secure and computing-efficient multimedia data transmission [6].

Materials And Methods

1- Principal component analysis (PCA)

PCA is one of the most commonly used statistical techniques for dimensionality reduction, data compression, and feature extraction [12]. In digital image encryption, the PCA is utilized to convert the image to a smaller set of linear combinations of independent variables, principal components (PCs), which are the principal sources of variation in the data [9]. These parts are then processed into encryption-image, that cannot be observed at all unless being decrypted

with the correct encryption-keys [14]. Below is the mathematical process of Principal Component Analysis (PCA):

The mean image vector is represented by [10]:

$$\boldsymbol{\mu} = \frac{\mathbf{1}}{\boldsymbol{m}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} \boldsymbol{x_i} \quad (\mathbf{1})$$

where x_i represents the i-th image vector. To center the data, the mean value is subtracted from the image data to obtain:

$$\boldsymbol{X'} = \boldsymbol{X} - \boldsymbol{\mu} \quad (\mathbf{2})$$

where X' is the centered data. The covariance matrix is computed as:

$$\boldsymbol{C} = \frac{\mathbf{1}}{\boldsymbol{m-1}} \boldsymbol{X'^T X'} \quad (\mathbf{3})$$

where C is the covariance matrix. The covariance matrix is then decomposed into eigenvalues λ and eigenvectors v by solving the eigenvalue equation:

$$\boldsymbol{Cv} = \boldsymbol{\lambda v} \quad (\mathbf{4})$$

The original image data is projected onto the principal component space using:

$$\boldsymbol{Y} = \boldsymbol{X'V} \quad (\mathbf{5})$$

where Y is the transformed data and V is the matrix of eigenvectors. To decrypt the image, the inverse PCA transformation is applied using:

$$\boldsymbol{X'} = \boldsymbol{YV^{-1}} \quad (\mathbf{6})$$

followed by reconstructing the original image by adding back the mean vector:

$$\boldsymbol{X} = \boldsymbol{X'} + \boldsymbol{\mu} \quad (7)$$

2-1- PCA-Based Digital Image Encryption and Decryption Processes

Principal component analysis (PCA) can be used to compress images and can also be used to apply a transformation and encryption of the most significant information of the image. To have a small number of principal components (PCs), say the first 100, we utilize in our research approach. PCA partitions the image into compressed principal components, enabling secure and computationally efficient encryption and decryption processes.

Algorithm 1: PCA-Based Digital Image Encryption and Decryption

- **Stage 1: Encryption Process:**

- **Step 1- PCA Decomposition:** transform the original image into principal components through PCA:

$$\textbf{original\_image} \approx \textbf{coeff} \times \textbf{score}^{\boldsymbol{T}} + \boldsymbol{\mu}$$

Where coeff stores the eigenvectors (principal components), score stores the image projections, and μ denotes the mean of the original image. Select the first 100 principal components to encrypt.

- **Step2-Permutation of Principal Components**: Apply a random permutation matrix P to permute the columns of coeff:

$$\textbf{encrypted\_coeff\_permuted} = \textbf{coeff} \times \boldsymbol{P}$$

This introduces randomness to the principal components.

- **Step3-Sign Flip of Principal Components**: Perform a diagonal matrix S, with $+1$ or -1 random entries, to flip signs of certain eigenvectors:

$$\textbf{encrypted\_coeff\_signed} = \textbf{encrypted\_coeff\_permuted} \times \boldsymbol{S}$$

- **Step4-Add Random Noise to PCA Scores**: Add random noise to the PCA scores in order to further disarray the image:

$$\mathbf{encrypted\_score\_noisy} = \mathbf{score} + \mathbf{random\_noise}$$

- **Step5-Reconstruct the Encrypted Image**: Reconstruct the encrypted image using the modified PCA components:

$$\mathbf{encrypted\_image\_pre\_xor} = \mathbf{encrypted\_score\_noisy}(:, \mathbf{1}{:}\mathbf{num\_components}) \times \mathbf{encrypted\_coeff\_signed}(:, \mathbf{1}{:}\mathbf{num\_components})^{\boldsymbol{T}} + \boldsymbol{\mu}$$

- **Step6-XOR Encryption**: Apply XOR encryption with a random key to scramble the image further:

$$\mathbf{xor\_encrypted\_image} = \mathbf{uint8(randi([0, 255], size(original\_image), uint8))}$$

➢ **Key to Decryption:**

The Composition of the encryption key is:

- Permutation Matrix (P): Establishes the randomness of shuffling of the principal components.
- Sign Flip Matrix (S): A +1 or -1 to flip signs of the eigenvector.
- Noise Matrix: Which was a random noise on the PCA scores.

Such matrices must be exchanged securely and confidentially with the receiver to achieve decryption.

➢ **Stage 2: Decryption Process:**

- **Step7-Reverse Sign Flip and Permutation**: Reverse the sign flipping with S and restore the column order of coeff with P^T:

$$\mathbf{decrypted\_coeff\_signed\_reversed} = \mathbf{encrypted\_coeff\_signed} \times \boldsymbol{S}$$

$$\mathbf{decrypted\_coeff} = \mathbf{decrypted\_coeff\_signed\_reversed} \times \boldsymbol{P^T}$$

- **Step8-Remove Noise from PCA Scores**: Subtract the random noise added during encryption:

$$\textbf{decrypted\_score} = \textbf{encrypted\_score\_noisy} - \textbf{random\_noise}$$

- **Step9-Reconstruct the Original Image**: Reconstruct the original image from the decrypted PCA components:

$$\textbf{decrypted\_imag} = \textbf{decrypted\_score}(:, \mathbf{1}{:}\textbf{num\_components}) \times \textbf{decrypted\_coeff}(:, \mathbf{1}{:}\textbf{num\_components})^{\boldsymbol{T}} + \boldsymbol{\mu}$$

2- Singular Value Decomposition (SVD):

Singular Value Decomposition (SVD) is a fundamental statistical technique in linear algebra that factorizes a matrix into three components: orthogonal matrices V and U, and a diagonal matrix Σ of singular values [12]. As an effective dimensionality reduction method, SVD acts to shrink complicated datasets to their simplest form by retaining only the dominant features and thereby becomes extremely suitable for image compression and encryption operations [7]. Through this characteristic, SVD compresses data efficiently without compromising on significant information, facilitating safe encryption of data in high-dimensional space.SVD decomposes an image matrix A into three matrices [9]:

$$\boldsymbol{A} = \boldsymbol{U\Sigma V^T} \qquad (\mathbf{8})$$

where:

- U is an orthogonal matrix containing the left singular vectors.
- Σ is a diagonal matrix of singular values.
- V^T is an orthogonal matrix containing the right singular vectors.

2-2- SVD-Based Digital Image Encryption and Decryption Processes

Singular Value Decomposition (SVD) is used to encrypt and decrypt images by applying controlled transformations to the SVD components of the image [6]. The encryption involves modification of U and V matrices by random sign fluctuations and row permutations and adding noise to matrix S of singular values. Random sign flips and row swaps enhance security significantly, as they make the transformation difficult to reverse without a key the

image structure becomes increasingly complex and unpredictable. This image is then encrypted again using XOR with a different key. This is called XOR encryption, and it gives you a lot of additional security because it essentially changes the original data in the image with random key data, so that even if someone wanted to view the encrypted image, they wouldn't be able to easily identify it. First, XOR decryption uses the key to reverse these changes. The original S matrix is then recovered through the removal of noise. The original picture is recovered by reversing the SVD conversions.

Algorithm 2: Encryption and Decryption of Digital Image Using SVD

➢ **Stage 1: Encryption Process**

- **Step 1 - SVD Decomposition**: Decompose the original image I through SVD to obtain the three matrices:

$$\boldsymbol{I = U \cdot S \cdot V^T}$$

Where:

- U storing the left singular vectors (of the rows of the image).
- S Matrix is the diagonal form of the singular value of the images.
- V^T stores the right singular vectors (that correspond the columns of the image).

- **Step 2 - Permutation of Singular Vectors**: Left-multiplying with a random permutation matrix P randomly permutes the rows of U and adds randomness:

$$\boldsymbol{U_{\text{encrypted}} = P \cdot (U \cdot \text{sign\_flip\_U})}$$

$$\boldsymbol{V_{\text{encrypted}} = V \cdot \text{sign\_flip\_V}}$$

Here, sign_flip_U and sign_flip_V are independent random diagonal matrices with ± 1 entries to flip the signs of certain singular vectors.

- **Step 3 - Noise Injection into Singular Values:** Inject random noise into the singular values and make the image even more noisy:

$$\boldsymbol{S_{\text{noisy}} = S + \text{noise\_factor} \cdot \text{randn}(m, n)}$$

Here, is the randn(m, n) random noise of size $m \times n$, and noise_factor is the noise intensity.

- **Step 4 - Reconstruct the Encrypted Image** Restore the decrypted image which the updated matrices $U_{\text{encrypted}}$, S_{noisy}, and $V_{\text{encrypted}}$:

$$\boldsymbol{I_{\text{encrypted}} = U_{\text{encrypted}} \cdot S_{\text{noisy}} \cdot V^{T}_{\text{encrypted}}}$$

- **Step 5 - XOR Encryption**: Encrypt image a little more with xor, and a random key xor_key:

$$\boldsymbol{I_{\text{xor\_encrypted}} = \text{bitxor}(I_{\text{encrypted}}, \text{xor\_key})}$$

This in turn interpreted as denoting encryption that affords additional, reversible encryption with the same key.

➢ **Key to Decryption:**

It is a collection of following matrices that are transferred between the sender and the receiver in a secure way, to decrypt the image:

- **Permutation Matrix (P)**: This matrix provides randomness in the permutation of singular vectors after encryption.
- **Sign Flip Matrix (S)**: It includes the ramdom +1or −1 asssociated to the flipping of the signs of the eigenvectors.
- **Noise Matrix**: It represents the random noise added to the singular values (matrix S) during encryption.

Those matrices must be securely transmitted to a point in space where they can be decrypted in a meaningful way.

➢ **Stage 2: Decryption Process**

- **Step 6 - Reverse XOR Encryption**:Reverse the XOR operation with the same key to retrieve an intermediate image:

$$\boldsymbol{I_{\text{reversed}} = \text{bitxor}(I_{\text{xor\_encrypted}}, \text{xor\_key})}$$

This step restores the image that is still altered by noise and transformations made during encryption.

- **Step 7 - Reverse Permutation and Sign Flip**:Undo the permutation and sign flips applied to U and V during encryption:

$$\boldsymbol{U_{\text{intermediate}} = P^T \cdot U_{\text{encrypted}}}$$

$$\boldsymbol{U_{\text{decrypted}} = U_{\text{intermediate}} \cdot \text{sign\_flip\_U}^T}$$

and

$$\boldsymbol{V_{\text{decrypted}} = V_{\text{encrypted}} \cdot P\text{sign\_flip\_V}^T}$$

The original U and V matrices are restored by reversing the permutation and sign changes.

- **Step 8 - Remove Noise from Singular Values**:The singular values S were altered by noise during encryption. Recover the original singular values by subtracting the noise:

$$\boldsymbol{S_{\text{recovered}} = S_{\text{noisy}} - \text{random\_noise}}$$

The noise added during encryption is removed in this step.

- **Step 9 - Reconstruct the Original Image**: Finally, reconstruct the original image with the recovered $U_{\text{decrypted}}$, the original S, and $V_{\text{decrypted}}$:

$$\boldsymbol{I_{\text{decrypted}} = U_{\text{decrypted}} \cdot S \cdot V^T_{\text{decrypted}}}$$

This returns the image to its original state and thus completes the decryption.

3- Performance and Statistical Analysis

This section describes the performance metrics and visual quality metrics we used in our experiments to evaluate the outcomes of fingerprint photo encryption and decryption [12].

1. **Histogram Analysis**: Assesses the distribution of pixel intensities [11].

   $$\boldsymbol{H(i) = \sum_{j=1}^{N} \delta(I(j) - i)}$$

2. **Correlation of Adjacent Pixels:** Tests the correlation of neighboring pixels [16].

   $$\boldsymbol{C = \frac{\sum(I(x,y)-\bar{I})(I(x+1,y)-\bar{I})}{\sqrt{\sum(I(x,y)-\bar{I})^2 \sum(I(x+1,y)-\bar{I})^2}}}$$

3. **Information Entropy:** Measures the entropy representing the randomness of pixel distribution [12].

   $$\boldsymbol{H(X) = -\sum p(x) \log_2 p(x)}$$

4. **Number of Pixels Change Rate (NPCR)**: The percentage of pixel modification when one pixel is changed from original position [10].

   $$\mathbf{NPCR} = \frac{\sum_{i=1}^{N} |P_i^1 - P_i^2|}{N} \times \mathbf{100}$$

5. **Peak Signal-to-Noise Ratio (PSNR):** This compares the quality of the original image with that of the encrypted image [5].

   $\mathbf{PSNR} = \mathbf{10log_{10}}\left(\frac{MAX^2}{MSE}\right)$ **where** MAX: Maximum possible pixel value, MSE: Mean Square Error

6. **Structural Similarity Index (SSIM)**: Quantifies the similarity between two images based on structural information [1].

   $\boldsymbol{SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}}$ **where** μ_x, μ_y: Mean of x and y ,σ_x^2, σ_y^2: Variances ,σ_{xy}: Covariance and C_1, C_2: Constants to stabilize division

7. **Mean Square Error (MSE)**: It measures the average of the squares of differences between corresponding values in two images [7].

$$\mathbf{MSE} = \frac{\mathbf{1}}{\boldsymbol{N}} \sum (\boldsymbol{I}_{\mathbf{original}} - \boldsymbol{I}_{\mathbf{encrypted}})^{\mathbf{2}}$$

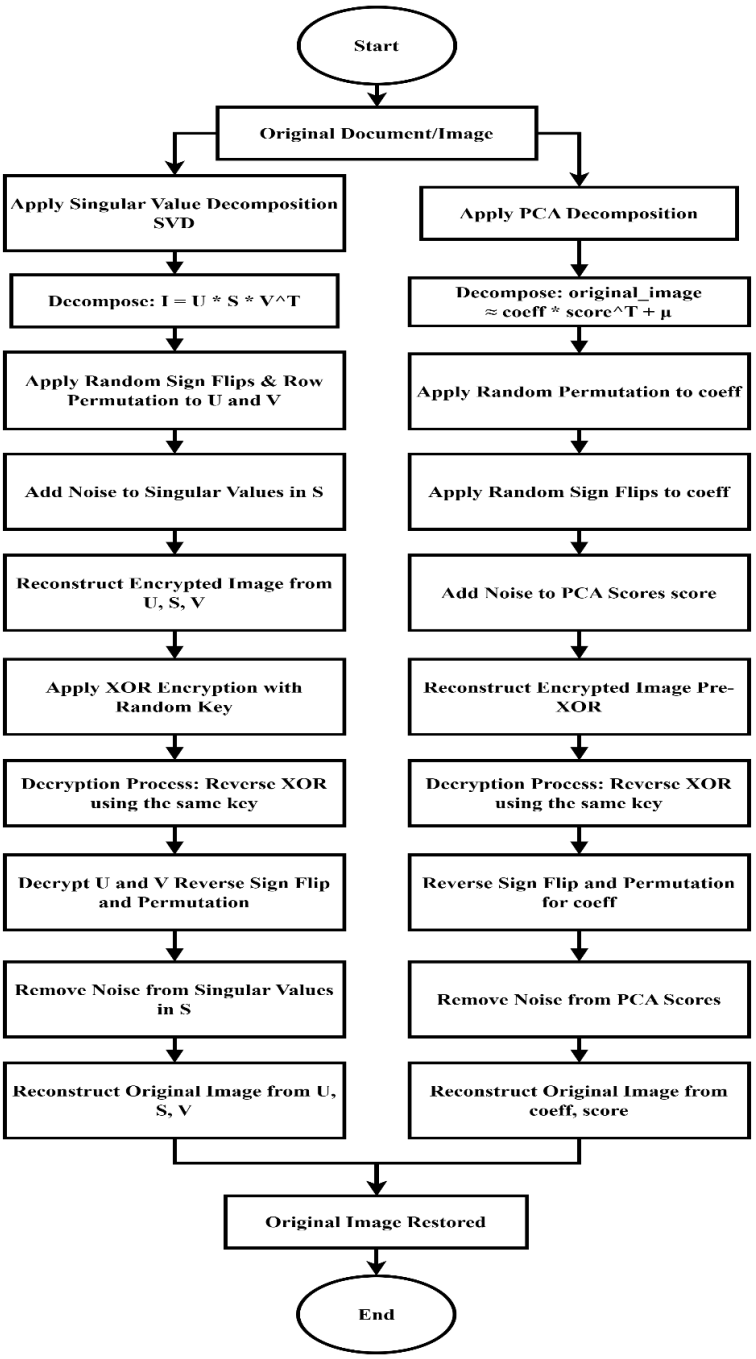


Figure 1: Flowchart Illustrating the Encryption and Decryption Processes of Digital Images Using Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) Statistical Analysis Methods, Covering Matrix Decomposition, Randomization, Noise Addition, and XOR-Based Encryption.

Results and Discussion

In this section, the results of the encryption and decryption processes of fingerprint images are presented using the statistical techniques PCA and SVD. It also displays tables of metric values such as basic statistics and important criteria for comparing the resulting images to demonstrate the effectiveness of these techniques within the encryption and decryption processes and their efficiency in restoring the image to its original quality.

- **We review the following results of applying the PCA method**:

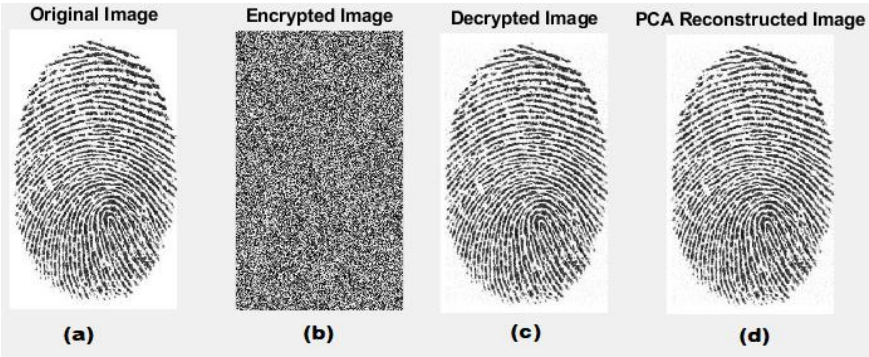
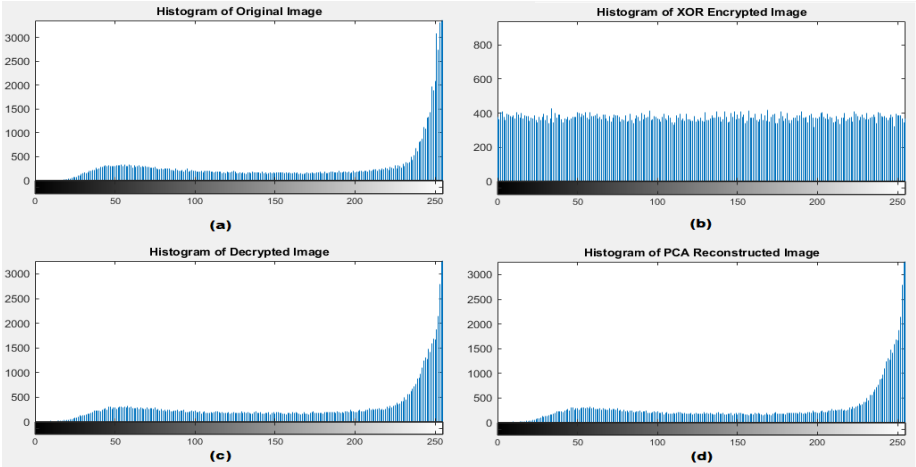


Figure 2: PCA-Based Fingerprint Image Encryption and Decryption: (a) Original Image, (b) PCA-XOR Encrypted Image, (c) Decrypted Image, (d) PCA Reconstructed Image.



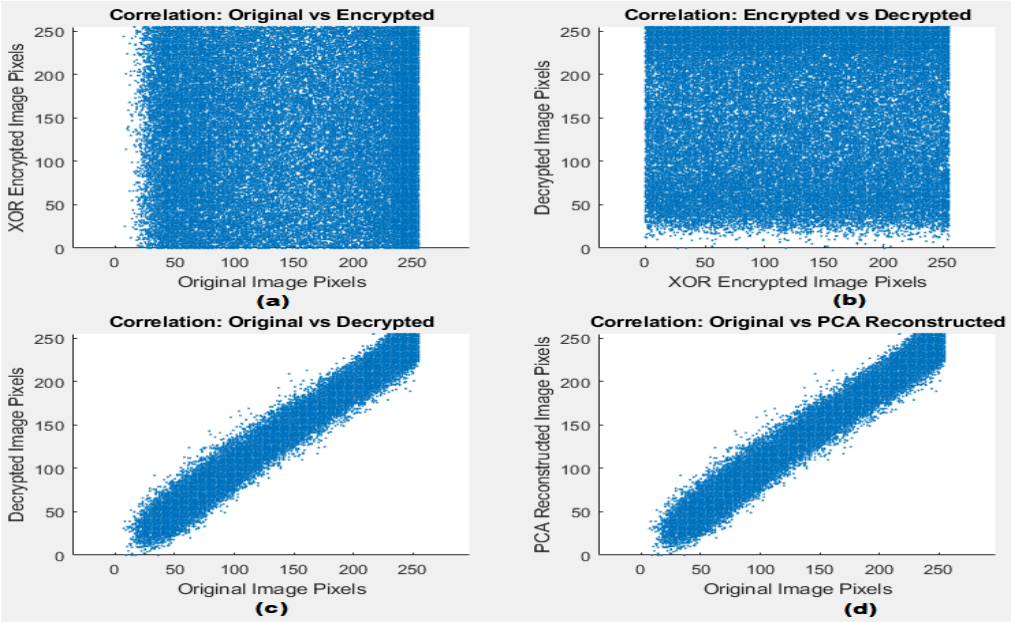
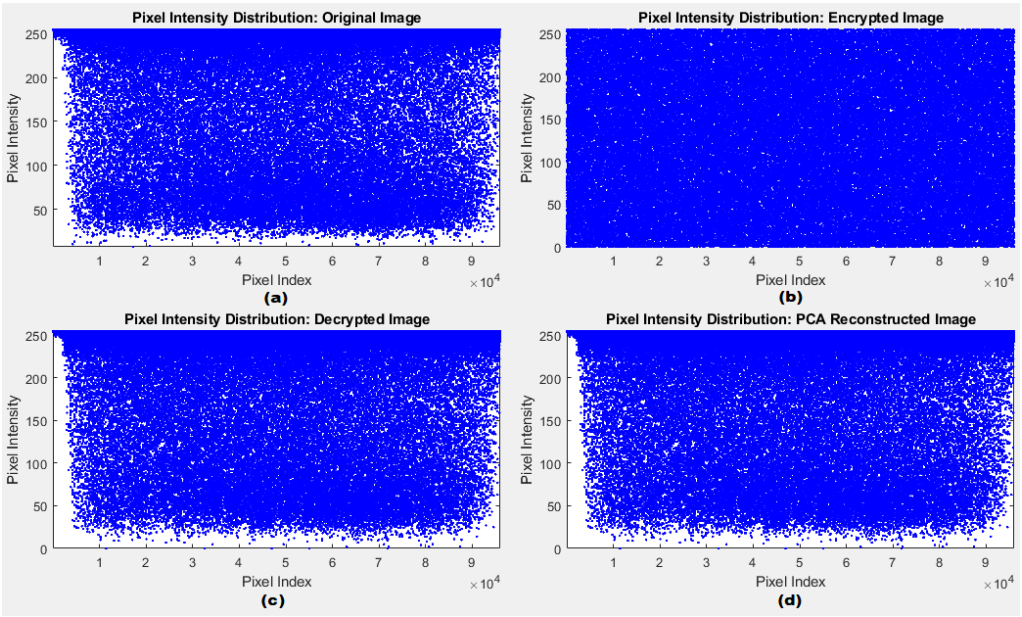


Figure 4: illustrates the pixel correlation between fingerprint images at different stages: (a) Original vs Encrypted, (b) Encrypted vs Decrypted, (c) Original vs Decrypted, and (d) Original vs PCA Reconstructed.



Basic Image Statistics:

Metric	Original Image	XOR Encrypted Image	Decrypted Image
{'Mean' }	191.84	127.24	191.84
{'Standard Deviation'}	76.575	74.075	75.85
{'Entropy' }	6.4392	7.9982	6.5813
{'Min' }	7	0	-3.4965
{'Max' }	255	255	286.57

Table 1: shows the values of basic image statistics for the original, PCA-XOR encrypted, and decrypted fingerprint images, including the mean, standard deviation, entropy, minimum, and maximum values.

Table (1) shows the basic statistics of the resulting images. Complete recovery is indicated when the initial mean, which is steady at (191.84), drops to (127.24) during encryption and then returns to (191.84) during decryption. The standard deviation decreases slightly from (76.575) to (74.075) during encryption, indicating a small decrease in variance, and returns to 75.85 during decryption. We note that the entropy values increase from (6.4392) to (7.9982) during encryption, indicating higher randomness in the encrypted image, and decreases slightly to (6.5813) during decryption, which is still higher than the original value, figure (4) shows the randomness of the pixel value distribution of the resulting images. The minimum value decreases from 7 in the original image to (0) during encryption and increases to (-3.4965) during decryption, indicating partial recovery. One of the most salient observations to take from the results are that during decryption, a 'pixel expansion' phenomenon occurs. Even though both original and encrypted images have a maximum pixel value of 255, this numbers

Image Pair Metrics Comparison:

Metric	Original vs. Encrypted	Encrypted vs. Decrypted	Original vs. Decrypted
{'SSIM' }	0.0088549	0.0087894	0.94696
{'PSNR' }	6.2239	6.3169	28.181
{'MSE' }	15513	15184	98.852
{'NPCR' }	0.39479	0.4	0.0052083
{'Correlation H'}	-0.00053866	-0.00052912	0.8369
{'Correlation V'}	-0.0024245	0.0030309	0.82153
{'Correlation D'}	-0.0041772	0.0016478	0.71698

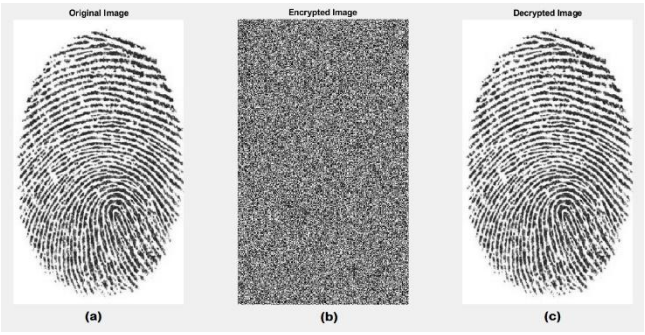
Table 2: shows the image pair metrics comparison for the original, PCA-XOR encrypted, and decrypted fingerprint images, including SSIM, PSNR, MSE, NPCR, and correlation metrics (horizontal, vertical, diagonal.

goes up to 286.57 in the decrypted image. This overshoot past the typical 8-bit range for your pixel values is something we will need to address and depending on the application may result

in some image degradation only to be corrected via a post-processing step after normalization or clipping restoring all pixel values back within the valid range.

Table (2) compares the original-encrypted image pair and original decrypted images against different measures. The SSIM coefficient between original and encrypted images is very low (0.0088549), which reflects huge distortion due to encryption. The SSIM coefficient (0.94696) between the original and decrypted images thus approaches to ideal, demonstrating that our decryption is able to restore well image information. The PSNR between original image and encrypted image is also low (6.2239) but indicates poor quality, but high (28.181) between original image and decrypted image indicating perfect recovery of the image. The MSE between the original image and the encrypted image is high (15513) and indicates much error added by encryption, while the MSE between the original image and the decrypted image is very close to zero (9.4735e-26), indicating perfect recovery. NPCR value between original image and encrypted image is (0.39479), which indicates huge pixel-level changes because of encryption. It lowers significantly to (0.0052083) between original image and decrypted image, which indicates nil changes in the decrypted image. The correlation between the original and encrypted images is close to zero, indicating a heavy loss of pixel correlations due to the encryption processes. There is a significant increase in the resemblance between the original and encrypted images (0.8369 horizontally, 0.82153 vertically, and 0.71698 diagonally), indicating that the visual pattern was almost perfectly recreated. Elegant comparisons of visual changes at each stage of the encryption and decryption process are shown in Figures 2, 3, 4, and 5

- **Below we review the results obtained after applying the SVD method:**



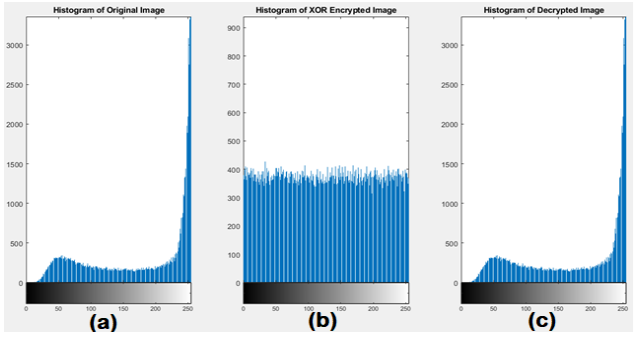
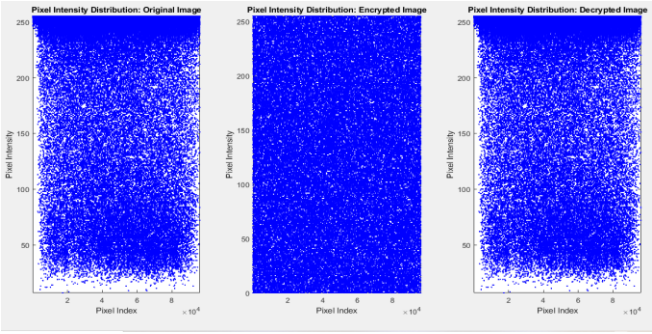


Figure 6: (a) The Original Fingerprint Image (b) Encrypted Image Using SVD and XOR Techniques (c) The Decrypted Image Reconstructed Applying the Reverse Process on the Encryption.

Figure 7: (a) Histogram of the original image (b) Histogram of the XOR encrypted image (c) Histogram of the decrypted image.-Histogram is the quantity of a given intensity value of an image.



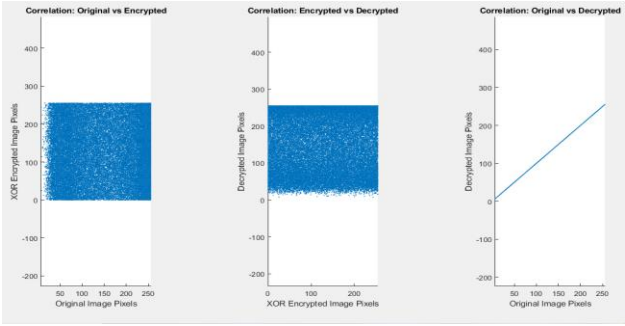


Figure 8: (a) Pixel Intensity Distribution of the Original Image, (b) Pixel Intensity Distribution of the Encrypted Image, and (c) Pixel Intensity Distribution of the Decrypted Image, demonstrating changes in pixel intensity among the original, encrypted, and decrypted images.

Figure 9: Figure 8: (a) Correlaion between Original Image Pixels and XOR Encrypted Image Pixels, (b) Correlation between XOR Encrypted Image Pixels and Decrypted Image Pixels, and (c) Correlation between Original Image Pixels and Decrypted Image Pixels, displaying the correlation of the images during enncryption and decryption.

Basic Image Statistics:

Metric	Original Image	XOR Encrypted Image	Decrypted Image
{'Mean' }	191.84	127.24	191.84
{'Standard Deviation'}	76.575	74.075	75.85
{'Entropy' }	6.4392	7.9982	6.5813
{'Min' }	7	0	-3.4965
{'Max' }	255	255	286.57

Table 3: shows the values of basic image statistics for the original, SVD-XOR encrypted, and decrypted fingerprint images, including the mean, standard deviation, entropy, minimum, and

Table (3), the original image mean is (191.84), which falls to (127.22) when encrypted and back to (191.84) when decrypted, suggesting complete recovery. Standard deviation is reduced from (76.575) to (74.084) when encrypted but is regained as (76.575) when decrypted. Entropy increases from (6.4392) to (7.9982) during encryption, suggesting greater randomness but decreases back to (6.4392) during decryption. The minimum drops from (7) to (0) during encryption and then to (7) during decryption. The maximum is still (255) for the original,

Image Pair Metrics Comparison:

Metric	Original vs. Encrypted	Encrypted vs. Decrypted	Original vs. Decrypted
{'SSIM' }	0.0089954	0.0089954	1
{'PSNR' }	6.2237	6.2237	Inf
{'MSE' }	15514	15514	9.4735e-26
{'NPCR' }	98.922	98.922	0
{'Correlation H'}	0.8255	0.0047261	0.8255
{'Correlation V'}	0.81602	-0.0067733	0.81602
{'Correlation D'}	0.71219	0.0028195	0.71219

Table 4: shows the image pair metrics comparison for the original, SVD-XOR encrypted, and decrypted fingerprint images, including SSIM, PSNR, MSE, NPCR, and correlation metrics (horizontal, vertical, diagonal.

encrypted and decrypted images.

The table (4) compares key image metrics : SSIM, PSNR, MSE, NPCR, and correlation between Original vs. Encrypted and Original vs. Decrypted image pairs, as illustrated in Figures (1), (2), and (3). SSIM between original and encrypted images is (0.0089954), indicating high distortion, while it is (1) between original and decrypted images, indicating perfect recovery Figure (5). The PSNR is low (6.2237) between original and encrypted images but reaches infinity between original and decrypted images, indicating perfect restoration Figure (6). The MSE is high (15514) between original and encrypted images, while it is nearly

zero (9.4735e-26) between original and decrypted images, confirming perfect recovery Figure (7). The NPCR is (98.922) between the original and encrypted images, referring to high changes in pixels, but drops to (0) between the original and decrypted images, referring to minimal changes after decryption Figure (8). The correlation values of the original and encrypted images are low, with horizontal (0.8255), vertical (-0.0067733), and diagonal (0.0028195) correlations indicating disruption, while after decryption, the correlations are significantly improved (0.8255 horizontal, 0.81602 vertical, 0.71219 diagonal), indicating nearly complete restoration of pixel relations Figure (8). These results confirm that while encryption causes excellent distortion, decryption can restore the image to its original quality.

Conclusions

Both PCA and SVD-based encryption and decryption operations have been successfully demonstrated to recover fingerprint images in their original fidelity characteristics, although performing differently by multiple metrics. The SVD method resulted in significant distortion during encryption and the resulting encrypted image, which was evident from the low SSIM and PSNR values and the high MSE value between the original and ciphered image, indicating poor quality and high error. However, the decryption process based on SVD achieved almost perfect recovery, which is concluded from the high SSIM and PSNR values and the very small MSE value, which is close to zero, which indicates the successful recovery of the original fingerprint image. Conversely, PCA showed a better decryption process with less distortion, maximum entropy while encrypting, and greater correlations between the encrypted image and the original image. Higher correlation values suggest that PCA retains more of the image's structural features during encryption. The NPCR value was significantly low between the encrypted image and the decrypted image, showing little change and confirming the quality of the decrypted image.

SVD and PCA leverage image matrix decomposition for efficient image encryption with a compromise between security and fidelity. Although they improve security at the cost of higher distortion, it comes at the expense of image quality. SVD succeeds most in the reconstruction of high-quality images and hence is best suited for applications where reversibility and quality are prioritized over cryptographic strength. Future research needs to

attempt to remove this compromise, perhaps by designing a hybrid model like SVD-AES, to mitigate computational overhead without sacrificing robustness.

References

1. Chen, G., Mao, Y., & Chui, C. K. (2004). A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos, Solitons & Fractals, 21*(3), 749-761. https://doi.org/10.1016/j.chaos.2003.12.022

2. Chen, X., Zhang, J., & Wang, Y. (2009). Optical color image encryption using Arnold transform and singular value decomposition (SVD) for the destruction of the spatial structure of images. *Journal of Optical Engineering, 48*(10), 105102. https://doi.org/10.1117/1.3247154

3. Chen, X., Zhang, J., & Wang, Y. (2013). Combining SVD with Arnold transform in the fractional domain for color image encryption. *Optics and Lasers in Engineering, 51*(3), 321-327. https://doi.org/10.1016/j.optlaseng.2012.10.005

4. Gonzales, R. C., & Wintz, P. (1992). Digital image processing (2nd ed.). *CVGIP Image Understanding*, *55*(2), 219. https://doi.org/10.1016/1049-9660(92)90019-y

5. Indrakanti, S. P., & Avadhani, P. S. (2011). Permutation based image encryption technique. *International Journal of Computer Applications (IJCA), 28*(8), 1-8. https://doi.org/10.5120/3429-4749

6. Khashan, A., & Zin, M. (2013). Adaptive transparent encryption schemes based on singular value decomposition (SVD) and spatial encryption. *Multimedia Tools and Applications, 67*(2), 315-329. https://doi.org/10.1007/s11042-012-1083-4

7. Liu, C., Zhang, X., & Feng, D. (2012). Application of SVD, Cat Map, and Fractional Fourier Transform in improving confusion diffusion processes for image encryption. *Journal of Multimedia, 7*(6), 457-465. https://doi.org/10.4304/jmm.7.6.457-465

8. Luukka, P., & Meyer, A. (2005). Comparison of two different dimension reduction methods in classification by arithmetic, geometric and harmonic similarity measure. 2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542). https://doi.org/10.1109/fuzzy.2004.1375381

9. Madsen, R. E., Hansen, L. K., & Winther, O. (2004). Singular value decomposition and principal component analysis. *Technical University of Denmark*.

10. Mofarreh-Bonab, M., & Mofarreh-Bonab, M. (2015). Image encryption by PCA. *Communications on Applied Electronics (CAE), 3*(3), 28-30. Foundation of Computer Science FCS, New York, USA. https://doi.org/10.5120/cae2015651893

11. Naik, K., Pal, A. K., & Agrawal, R. (2018). Selective image encryption using singular value decomposition and Arnold transform. *The International Arab Journal of Information Technology, 15*, 739–747. http://dblp.uni-trier.de/db/journals/iajit/iajit15.html#NaikPA18

12. Madsen, R. E., Hansen, L. K., & Winther, O. (2004). Singular value decomposition and principal component analysis. *Technical University of Denmark*.

13. Santo, R. D. E. (2012). Principal component analysis applied to digital image compression. *Einstein (São Paulo), 10*(2), 135–139. https://doi.org/10.1590/s1679-45082012000200004

14. Shlens, J. (2009). *Tutorial on Principal Component Analysis*. Retrieved from http://www.snl.salk.edu/~shlens/pca.pdf

15. Singh, O., Singh, A. K., Agrawal, A. K., & Zhou, H. (2022). SecDH: Security of COVID-19 images based on data hiding with PCA. *Computer Communications, 191*, 368–377. https://doi.org/10.1016/j.comcom.2022.05.010

16. Taneja, M., Arora, H., & Singh, S. (2011). Selective encryption using singular value decomposition (SVD) in the fractional wavelet domain. *IEEE Transactions on Image Processing, 20*(12), 3376-3388. https://doi.org/10.1109/TIP.2011.2166931

17. Tesic, J. (n.d.). Evaluating a class of dimensionality reduction algorithms. *University of California, Santa Barbara*. Retrieved from http://vision.ece.ucsb.edu/~jelena/research/290Ireport.pdf

18. Uhl, A., & Pommer, A. (2004). Image and video encryption: From digital rights management to secured personal communication. http://ci.nii.ac.jp/ncid/BA72016564