FMDB *Transactions on Sustainable Computer Letters*

Enhanced PVD Steganography with Dynamic Programming and Scrambling for Robust Security

Lekshmi Kalinathan[1,*], K. Devi[2], S. Benila[3], Vasudev Arun Ganesh[4], R. Santhosh Krishna[5], Anudeep Lam[6]

[1,2,3,4,5,6]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India.
lekshmi.k@vit.ac.in[1], devi.k@vit.ac.in[2], benila.s@vit.ac.in[3], vasudevarun.ganesh2021@vitstudent.ac.in[4], santhoshkrishna.r2021@vitstudent.ac.in[5], lam.anudeep2021@vitstudent.ac.in[6]

Abstract: Pixel Value Differencing (PVD) is one of the most widely used steganographic techniques, which relies on the differences in pixel intensities between adjacent pixels to conceal secret data. By altering these differences, PVD enables efficient data hiding, minimising significant image-quality degradation from slight, often unnoticeable alterations in pixel values to the human eye. However, classical PVD struggles to achieve an optimal balance between capacity and visual invisibility. This work introduces an enhanced PVD-based steganography system that incorporates a dynamic programming approach to optimise the embedding process. By dynamically selecting pairs of pixels and allocating data optimally, it improves the trade-off between image distortion and embedding capacity. Additionally, an extra layer of security is implemented using a Rubik's Cube scrambling algorithm that rearranges the pixels in the stego image. This hybridisation of PVD optimisation with scrambling ensures that even when a stego image is successfully obtained, extracting the hidden data becomes significantly more challenging. Our approach demonstrates superior image quality and robustness against steganalysis, providing a secure and efficient method for covert communication.

Keywords: Pixel Value Differencing (PVD); Image Steganography; Visual Invisibility; Data Embedding Optimization; Steganalysis Resistance; PVD Optimization; Steganography System.

Received on: 10/09/2024, **Revised on:** 13/11/2024, **Accepted on:** 16/12/2024, **Published on:** 03/06/2025

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSCL

DOI: https://doi.org/10.69888/FTSCL.2025.000422

Cite as: L. Kalinathan, K. Devi, S. Benila, V. A. Ganesh, R. S. Krishna, and A. Lam, "Enhanced PVD Steganography with Dynamic Programming and Scrambling for Robust Security," *FMDB Transactions on Sustainable Computer Letters*, vol. 3, no. 2, pp. 60–75, 2025.

1. Introduction

The field of secure data transmission has seen a significant need for robust, undetectable information embedding within digital media. Image-based steganography, which conceals data within image pixels, is a popular approach due to its high capacity and relatively low detectability [25]. PVD is a widely recognised image steganography technique that uses subtle variations in pixel intensities to embed information, particularly in grayscale or colour images. This paper focuses on an advanced, dynamic programming (DP)-enhanced PVD method for secure, high-capacity image steganography, ensuring the integrity of both

*Corresponding author.

embedded data and image quality. Using dynamic programming, this approach enhances data retrieval efficiency, optimizing encoding and decoding processes to maximize embedding capacity while minimizing visual distortions.

The Pixel Value Differencing technique is advantageous for its adaptability and the ability to embed varying numbers of bits based on pixel intensity differences. In traditional PVD, data are embedded based on the absolute difference between pairs of neighbouring pixels. Small pixel differences correspond to regions with little to no visual variation (such as smooth areas), allowing only a few bits to be embedded. In contrast, larger differences, typically found in high-contrast or edge areas, enable greater embedding capacities without creating noticeable artefacts. This variability in embedding capacity enhances security by allowing more data to be embedded in locations that are less detectable to the human eye. However, traditional PVD methods often struggle to achieve optimal embedding efficiency because they may require recalculations for each pixel pair to determine the most efficient embedding path. The DP-based approach of this method improves PVD by optimising the embedding process using dynamic programming, enabling the retrieval of embedded data with minimal computational overhead.

2. Literature Survey

In recent years, image steganography has undergone significant development, transforming the fields of information security and digital communication. The progressive development of complex techniques in the spatial and frequency domains has improved the ability to protect hidden and secret data during transmission while preserving image fidelity [7]. The method adopted, Pixel Value Differencing with a good subtlety range, generates subtle, almost imperceptible changes to convey secret information by exploiting variations in the natural variation of each pixel and its adjacent pixel pair. Although revolutionary in itself, the traditional PVD approach suffers from various technical issues that researchers have worked diligently to overcome. Its anomaly of PDH —the ability to be constructed as recognisable patterns in images, disguised or otherwise —and the distortion of boundaries, potentially arranged to degrade image quality, have thus become a specific object. These limitations led to the emergence of multidirectional PVD steganography. This sophisticated technique embeds data in multiple pixel orientations—horizontal, vertical, and diagonal—resulting in a much more intricate and robust embedding pattern, which significantly increases both data capacity and stealthiness [20]. In modern steganography, the substitution of LSBs remains a starting point due to its simplicity and effectiveness in basic data-hiding applications. The elegance of this method lies in altering the least impactful bits of the pixel values, resulting in very low visible distortion of the carrier image [6]; [19].

However, due to the inherent limitations of LSB as a detection method, hybrid approaches have been developed that combine LSB with other techniques. The most significant contribution is the combination of PVD with the modulus function, which yields a much more robust system that solves First-Order Border Problems (FOBP) more robustly. In fact, combining modulus and PVD functions forms an extremely effective method for controlling refined, complex details in image regions; therefore, it significantly improves resistance to statistical steganalysis methods [13]. This method was further improved by introducing edge detection algorithms, which enabled the differential imposition of data in high-texture regions with natural pixel variations, thereby making the embedded information almost imperceptible to the image's normal properties [23]. The application of computational intelligence in steganography has led to remarkable improvements in embedding efficiency. Genetic Algorithms (GA) have revolutionised the pixel pair selection process, introducing sophisticated optimisation techniques that carefully balance the competing demands of imperceptibility and payload capacity [3]. These algorithms implement complex fitness functions that simultaneously evaluate multiple criteria, ensuring optimal embedding locations are identified while maintaining image quality. Another giant step in this field is the development of adaptive PVD techniques. These high-tech systems dynamically adjust the embedding strategy based on local image characteristics, using variable-sized pixel blocks and multidirectional embedding patterns within six-pixel clusters.

This flexible method has significantly increased the payload while maintaining outstanding image quality [16]. This hybrid combination of PVD and LSB resulted in a highly robust system that leverages multilayer protections against both visual and statistical attack methods [18]. The implementation selection techniques for quantisation added greater sophistication to the steganographic system, as they are adaptive to different static image characteristics and adjust the embedding process by carefully balancing multiple parameters [1]. The adaptive approach would achieve the highest data capacity with this key visual imperceptibility requirement. Image preprocessing using the scrambling technique has now become the strongest feature for enhancing security. The algorithms based on Rubik's Cube scrambling provide the most advanced technique for scrambling pixel positions, making it much harder to extract unauthorised data [21]; [17]. This mechanism is further enhanced by utilizing the superchaotic map, which imposes highly nonlinear transformations to produce pixel configurations that are nearly unbreakable [14]; [11]. Thorough reviews of these scrambling methods were presented, suggesting their use in various applications to produce strong steganographic schemes resistant to attack [12]. Thus, advanced hybrid techniques emerged, including transform-domain methods, further advancing sophisticated encryption combined with steganography. The use of the discrete cosine transform domain, combined with advanced encryption standards such as AES or RSA, ensures the provision of robust systems that can withstand various attacks, including JPEG compression attacks, as demonstrated in references [5], [10], and [22].

The methods ensure the integrity of the data under challenging conditions, thereby providing reliable and secure communication channels. This has enabled new opportunities in data embedding and steganalysis using deep learning technologies, especially Convolutional Neural Networks (CNNs). Such advanced systems can learn complex feature representations that traditional techniques often miss, allowing them to automatically discover appropriate embedding patterns and thus provide robust protection against advanced detection methods [4]; [8]. The applications of steganographic techniques have expanded exponentially, especially in sensitive areas such as the Industrial Internet of Things (IIoT). Due to such applications, security requirements need to be carefully balanced with operational efficiency. Thus, such optimised techniques are developed for specific use cases [2]. Further refinement of such applications involved implementing adaptive payload distribution systems that intelligently dispersed hidden data across multiple carrier images using sophisticated texture analysis [24]. Highly accurate steganalysis systems are designed to contribute to the evolution of both offensive and defensive systems, as cited in [15]. This includes more sophisticated techniques, such as optimal embedding strategies using dynamic programming, as well as innovative transformation techniques, such as Rubik's Cube-based transformations, which reflect the shape the field is taking as it develops sophisticated and secure systems. Hence, it was more than an incremental improvement; it was a fundamental change in how we pose the problem of secret data embedding. It combined old, time-tested techniques with new, fast computational methods, thereby developing robust systems to meet the growing demand for secure communication in an increasingly interconnected world.

3. Methodology

This paper proposes a revised approach to data security by combining two established techniques: modified Pixel Value Decomposition (PVD) steganography, utilizing Dynamic Programming, and Rubik's Cube image scrambling.

3.1. PVD Steganography for Secure Data Embedding

PVD steganography offers a robust method for concealing secret data within a cover image. The process involves fragmenting the cover image into multiple shares, each containing a portion of the secret message. These shares are then distributed or stored independently. Reconstruction of the original message requires the collection and combination of a predefined threshold number of shares. This inherent redundancy enhances security by ensuring data remains inaccessible without enough shares. The methodology leverages established visual cryptography schemes (VCS), such as (2,2)-PVD or (3,3)-PVD, for secure data embedding. A custom-designed modified PVD algorithm will be developed to partition the cover image into shares and seamlessly integrate secret data within them. The design will prioritise maintaining the visual fidelity of the cover image while ensuring the embedded data remains imperceptible to the human eye. It works by subtly altering the pixel values of the cover image, making the changes invisible to human observers. Here are some formulas and a rudimentary description of how PVD functions:

- Our PVD algorithm modifies the pixel values of the cover image based on the secret data to be embedded. It involves calculating the difference between adjacent pixel values. This difference is calculated, and the most optimal pairs of pixels are chosen using a custom Dynamic Programming algorithm.
- The calculated differences are then used to embed the secret data in the cover image. For example, if the difference between two adjacent pixels needs to represent a binary value (0 or 1) of the secret data, then the difference is modified accordingly to encode this information.
- The modified cover image, now containing embedded secret data, is referred to as the stego image.
- To retrieve the secret data from the stego image, the extraction process reverses the embedding steps.

$\Delta I'(x, y) = f(\Delta I(x, y), \text{secret data})$

- **f:** A function that determines how the differences are modified based on the secret data.
- **I (x, y):** The intensity (pixel value) of the cover image at the location (x, y).
- **I′ (x, y):** The intensity (pixel value) of the stego image at location (x, y).
- **ΔI (x, y):** The difference between the pixel values of adjacent pixels in the cover image.
- **ΔI′ (x, y):** The difference between the pixel values of adjacent pixels in the stego image.

Algorithm 1: Steganography via Optimal Pixel Pair Embedding

Require: Cover image I, secret message M

Ensure: Stego image I' with embedded data Convert secret message M to binary stream *Mbits* Initialize embedding capacity and counters.

for each row i in I **do** *{*Step 2: Optimal Pixel Pair Selection*}*
Select pixel pairs (p_1, p_2) using dynamic programming
Compute pixel difference $\Delta = |p_1 - p_2|$ for embedding suitability
end for

for each selected pixel pair (p_1, p_2) **do** *{*Step 3: Embedding Capacity Calculation*}*
Calculate embedding capacity: bits per pair derived from difference Δ
Store pixel pairs offering maximum embedding capacity with minimal distortion
end for

for each selected pixel pair (p_1, p_2) **do** *{*Step 4: Data Embedding*}*
Embed bits from *Mbits* by adjusting pixel difference Δ within allowed range
Apply overflow/underflow control to keep pixel values in [0, 255] range
Modify pixel pairs accordingly to encode secret bits

end for

Save the modified image I' in uncompressed format
Record and output total embedding capacity achieved

3.1.1. Purpose and Role of Dynamic Programming Arrays

The two DP arrays are designed to manage the embedding capacity at each pair of pixels in a way that avoids significant perceptual distortion while maximising the amount of embedded data. First DP Array (Capacity Array): This array, denoted dp1, stores the maximum achievable embedding capacity (in bits) at each pixel-pair step. It represents the optimal solution up to the current pair of pixels, capturing cumulative embedding capacity at each point. Second DP Array (Modification Array): The second array, dp2, tracks the specific modifications applied to the pixel values to achieve the embedding capacity recorded in dp1. This modification array enables efficient backtracking to reconstruct optimal embedding decisions without recalculating or reevaluating the embedding choices of each pixel pair.

3.1.2. Step-By-Step Approach for Filling the DP Arrays

The dynamic programming (DP) approach systematically fills both arrays by iteratively processing each pixel pair and calculating the potential embedding capacity based on the difference between the pixels' values.

- **Evaluating Pixel Differences:** For each pixel pair, calculate the difference between the two values of the pixels. This difference guides the selection of an embedding interval that determines the bit capacity for that pair of pixels.
- **Determining Bit Embedding Capacity:** The interval in which Δ falls corresponds to a specific embedding capacity. Each range represents a tolerance level for visual changes. The algorithm identifies the interval and calculates the maximum number of bits that can be embedded without perceptual impact. The dp1 array is updated to store the highest cumulative embedding capacity:
    - Compare the current pixel pair's potential bit capacity with the maximums previously stored in dp1.
    - If the new capacity is greater, update dp1 and store the corresponding pixel adjustments in dp2.
- **Incremental Summation in dp1:** dp1 maintains a running total of embedding capacities. The embedding potential of each pixel pair is added to the cumulative total from the previous step. This ensures that each step builds upon the optimal decisions made in the previous steps.

3.1.3. Backtracking for Optimal Embedding Decision

Once both dp1 and dp2 are filled, the embedding process proceeds with a backtracking step. This stage identifies the specific embedding choices that maximise embedding capacity while preserving image quality. Executing Backtracking with the Modification Array (dp2): Starting from the final position in dp1, backtracking is used to determine which pixel justifications stored in dp2 contributed to the cumulative embedding capacity. This ensures that only the most efficient modifications, which optimise data embedding while preserving visual quality, are applied to the image. Following the adjustments in dp2, the algorithm constructs an optimal path to embed, thereby securely and effectively embedding the bits without having to recalculate each decision.

3.1.4. Necessity and Benefits of the DP Approach

Utilizing a DP approach in the PVD embedding process provides significant advantages, particularly in enhancing embedding capacity while maintaining the security and imperceptibility of the embedded data. Maximised Embedding with Controlled Distortion: Through a systematic evaluation of all possible embedding paths, the DP arrays identify the figuration that maximizes bit capacity for each pixel pair while minimizing perceptual changes. Improved Security and Robustness: By optimising embedding choices, the DP approach ensures only minimal, essential pixel modifications are made, reducing the likelihood of detection by steganalysis. Computational Efficiency: The DP approach eliminates redundant calculations by storing intermediate values in arrays, enabling efficient backtracking and leading to faster computations without sacrificing optimality.

3.2. Rubik's Cube Image Scrambling for Enhanced Obfuscation

This approach incorporates Rubik's Cube image scrambling to further obfuscate the hidden data [9]. The scrambling process treats each colour pixel within the image as a distinct cubelet. Subsequently, operations analogous to those used in Rubik's Cube manipulations are applied to achieve a controlled rearrangement of these cubelets, effectively scrambling the image data. A critical aspect of this process is reversibility, ensuring that the original image can be perfectly reconstructed from its scrambled counterpart.

3.2.1. Developing the Scrambling Algorithm

The scrambling algorithm 2 emulates Rubik's Cube-like moves on the image's pixel grid, accounting for image dimensions, the number of operations applied, and a strong randomisation mechanism to ensure unpredictability. Designed for efficiency and reversibility, it integrates pixel shifts with XOR-based encryption. The XOR operation adds a vital security layer by altering pixel values using row and column keys, achieving dual objectives: (1) Enhanced security through varied transformations that resist reverse-engineering, and (2) complementary obfuscation, where positional scrambling and intensity masking together ensure the original image remains concealed even if shifts are reversed.

Algorithm 2: Image Scrambling Algorithm

Require: Original image I with dimensions (W, H)
Ensure: Scrambled image I′
Load input image I; Extract width W, height H, alpha channel (if any)
{Step 1: Initialisation}
Initialize 2D array A[W][H] for grayscale pixel values {Prepare storage for processing}
for each pixel (x, y) in I do
    Assign grayscale (red channel or brightness) to A[x][y] {Step 2: Pixel assignment}
end for
Generate random row shifts KR[0..H − 1] {Step 3: Generate row keys}
Generate random column shifts KC[0..W − 1] {Step 3: Generate column keys}
for each row i in A do
    Compute α = sum modulo 2 of row i {Step 4: Determine row shift direction}
    if α = 0 then
    Left shift row i by KR[i]
    else
    Right shift row i by KR[i]
    end if
end for
for each column j in A do
    Compute β = sum modulo 2 of column j {Step 5: Determine column shift direction}
    if β = 0 then
    Upward shift column j by KC[j]
    else
    Downward shift column j by KC[j]
    end if
end for
for each pixel (i, j) in A do
    if i odd then
    XOR pixel (i, j) with KC[j] {Step 6: XOR operation part 1}

```
        else
        XOR pixel (i, j) with KC[W − 1 − j]
        end if
end for
for each pixel (i, j) in A do
        if j odd then
        XOR pixel (i, j) with KR[i] {Step 6: XOR operation part 2}
        else
        XOR pixel (i, j) with KR[H 1−0 1 − i]
        end if
end for
Convert A to uint8 image format {Step 7: Create output image}
Generate a scrambled image I′
Save I′
```

4. Result

4.1. Output

Figure 3 illustrates the steps and transformations involved in embedding, scrambling, and detecting data differences, demonstrating both the effectiveness of data hiding and the minimal impact on image quality.

- **Original Image 3(a):** This is the original, unaltered image before any steganographic or scrambling operations have been applied. It serves as a baseline for comparison.
- **Image after steganography 3(b):** The image is also steganographic. Essentially, it embeds a hidden message or data within the image. The differences in representation when data are embedded are minimal compared to the original, so that fidelity in the visual presentation remains intact and the message is not easily discernible to the human eye.
- **Scrambled image 3(c):** It is an encrypted version of the original image. Here, the pixels are shuffled or encoded, likely using a scrambling encryption algorithm. The outcome would be an image that appears noisy and is not visually interpretable. That is why it enhances the security of the hidden data.

The strength of this approach lies in its ability to alter pixel values minimally, ensuring that hidden data remains undetectable through casual visual inspection. This high level of imperceptibility preserves the image's appearance, thereby improving security while retaining overall image quality and making embedded information very difficult to detect or extract without the appropriate key or method.

4.2. Comparative Analysis of Steganography Algorithms

In the following section, we report the quantitative and comparative performance of three steganography techniques: LSB, PVD, and the enhanced PVD based on DP. A relevant publication provides the standard image dataset for testing: The Oxford 102 Flower Dataset of Nilsback and Zisserman [25], which contains images of several species of flowers, along with possible ground conditions at different colours and textures, to test each of the developed methods. Compared to other methods, LSB is the simplest and most high-fidelity because it makes the fewest modifications to pixel values during the encoding process. However, LSB appears less secure than other methods, since data is often embedded in predictable patterns, making it prone to steganalysis. In contrast, PVD enhances security because the data is encoded as differences between pairs of pixels, which are imperceptible to both visual and statistical detection. This scheme introduces some distortion to achieve robustness, aiming to strike a balance between fidelity and security. Our DP-enhanced PVD extends mainstream PVD by dynamically optimising with dynamic programming to achieve the highest level of embedding capacity and robustness. The proposed method aims to achieve higher levels of security and resistance to detection by strategically modulating pixel values to balance image quality during the attempt. The performance of the PVD, LSB, and modified DP-based PVD methods can be evaluated using metrics such as SSIM, MSE, and PSNR. Each image quality metric reveals something distinct about the robustness, security, and fidelity of each algorithm. Let us briefly look at each one. SSIM - SSIM is one of the most popular criteria in digital image processing and computer vision. It is designed to measure the similarity between two images by assessing changes in pixel values and the structural information that the human eye perceives. SSIM compares three aspects of images:

- **Luminance Comparison:** Measures the similarity of the luminance (brightness) between the two images.
- **Comparison of contrasts:** It evaluates the similarity of the contrast (brightness differences) between the two images.
- **Structure comparison:** Examines the similarity of the structures (patterns, textures) between the two images.

These three aspects are combined to produce an overall SSIM score that typically ranges from -1 to 1. A score of 1 indicates perfect similarity, while a score of -1 indicates perfect dissimilarity.

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \quad (1)$$

Where:

μx is the average of x
μy is the average of y
σ2 is the variance of x
σ2 is the variance of y
σxy is the covariance of x and y
c1 = (K1L)2 and c2 = (K2L)2
L is the dynamic range of pixel values
K1 = 0.01 and K2 = 0.03

Table 1: Experimental results and comparison between steganography algorithms

Image	Metric	PVD	LSB	PVD+DP
Image_03647	PSNR	66.60673	70.06131	63.08677
	SSIM	0.999817	0.999948	0.99975
	MSE	0.014204	0.006411	0.031264
Image_03653	PSNR	67.82532	70.01888	63.76177
	SSIM	0.999916	0.999963	0.999832
	MSE	0.010729	0.006474	0.027347
Image_03975	PSNR	59.24816	69.52499	57.97884
	SSIM	0.99989	0.999957	0.999841
	MSE	0.067689	0.007254	0.084309
Image_04027	PSNR	67.72639	70.25611	64.03818
	SSIM	0.999918	0.999968	0.999853
	MSE	0.010976	0.00613	0.02566
Image_04182	PSNR	66.22015	69.57434	62.71363
	SSIM	0.999827	0.999939	0.999712
	MSE	0.015526	0.007172	0.034811
Image_04222	PSNR	64.71904	69.62066	57.08622
	SSIM	0.999884	0.999951	0.999821
	MSE	0.01866	0.007096	0.085596
Image_04324	PSNR	60.50929	68.84502	58.41773
	SSIM	0.999779	0.999914	0.999566
	MSE	0.045511	0.008484	0.067466
Image_04383	PSNR	59.87644	69.82165	59.80732
	SSIM	0.998375	0.999781	0.998132
	MSE	0.046082	0.006775	0.047927
Image_04501	PSNR	63.41306	69.14329	60.29616
	SSIM	0.999902	0.99995	0.999843
	MSE	0.029075	0.00792	0.055161
Image_04577	PSNR	66.16358	70.01684	62.98905
	SSIM	0.999833	0.99995	0.999755
	MSE	0.01573	0.006477	0.032672
Image_04706	PSNR	62.28906	70.03576	61.45663
	SSIM	0.999942	0.999978	0.999896
	MSE	0.033841	0.006449	0.043769
Image_04770	PSNR	63.20092	68.745	60.19468
	SSIM	0.99981	0.999957	0.999656
	MSE	0.031116	0.008681	0.056279
Image_04840	PSNR	63.89547	68.87416	59.87218

	SSIM	0.999779	0.99992	0.999695
	MSE	0.026225	0.008427	0.05645
Image_04993	PSNR	66.91679	69.98335	63.59223
	SSIM	0.999923	0.999969	0.999822
	MSE	0.013225	0.006527	0.028435
Image_05026	PSNR	66.39305	70.03795	62.48777
	SSIM	0.999912	0.999966	0.999844
	MSE	0.01492	0.006446	0.034854
Image_05153	PSNR	65.04365	68.45081	61.38196
	SSIM	0.999739	0.999906	0.999632
	MSE	0.020357	0.00929	0.047303
Image_05231	PSNR	66.42533	69.40654	62.52286
	SSIM	0.999936	0.999967	0.99989
	MSE	0.01481	0.007455	0.036374
Image_05291	PSNR	60.24613	59.56206	57.72252
	SSIM	0.999798	0.999933	0.999652
	MSE	0.037902	0.007192	0.069941

MSE -It is a measure of the average square difference between the original and reconstructed images (Figure 1).

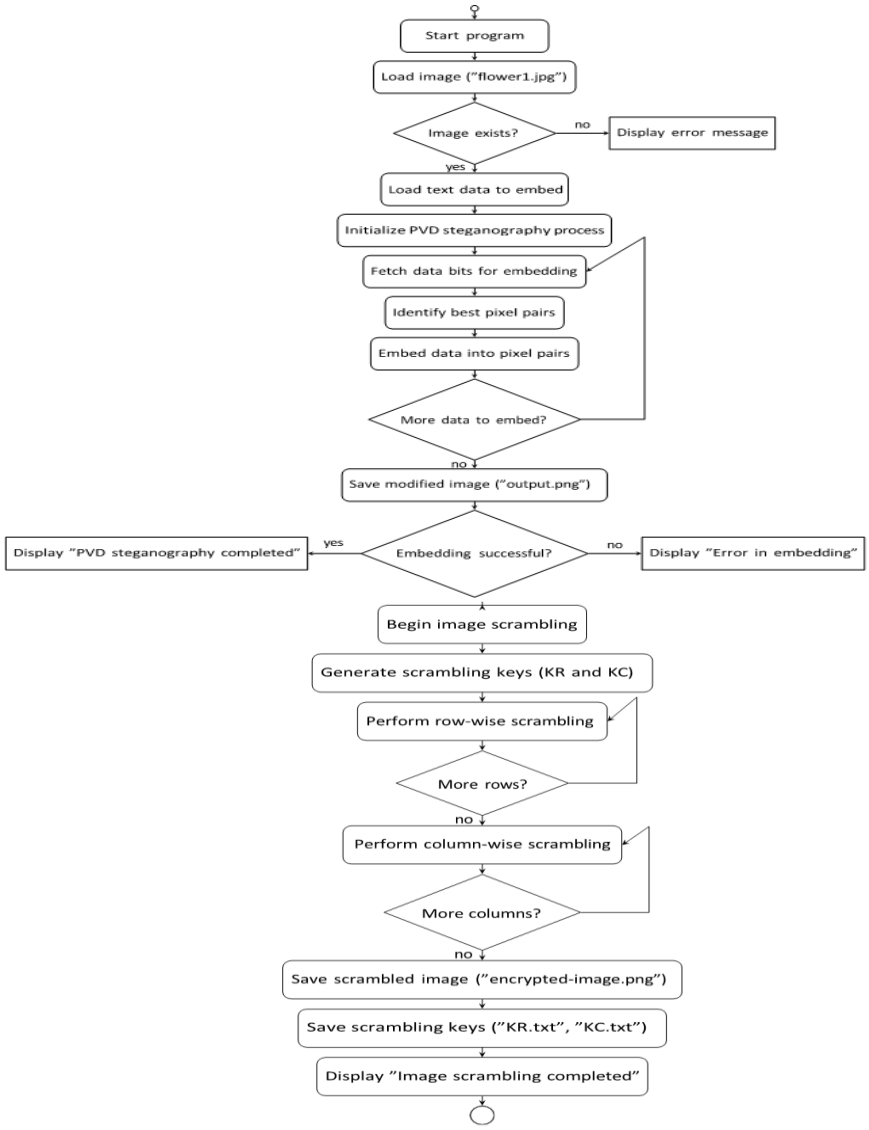


Figure 1: State diagram of system 1

Low MSE values indicate greater similarity between the two images, suggesting fewer distortions or errors in the construction process (Figure 2).

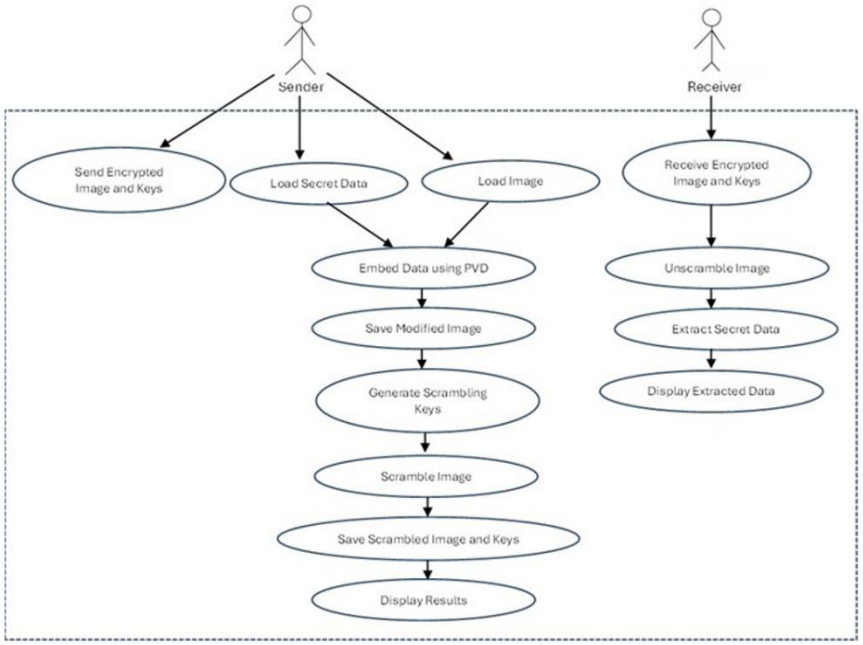


Figure 2: Class diagram of the system

The MSE is generally used in loss functions for various image processing tasks, such as denoising, compression, or restoration, where the difference between the original and processed images must be minimised. Here is how it is calculated:

- For each corresponding pixel in the original and reconstructed images, subtract the pixel value of the original image from the pixel value of the reconstructed image.
- Square the result of each subtraction.
- Calculate the average of all squared differences.

$$\mathrm{MSE} = \frac{\sum_{M,N}[I_1(m,n) - I_2(m,n)]^2}{M \times N} \qquad (2)$$

PSNR - The PSNR is another widely used metric in digital image processing and compression. The result is the quality of a compressed or reconstructed image relative to its original. The value of PSNR is usually computed using the MSE between the original image I and the reconstructed image I‘.The higher the PSNR, the better the quality, indicating a stronger signal relative to noise. It is used for gauging the performance of image and video compression algorithms (Figure 3).

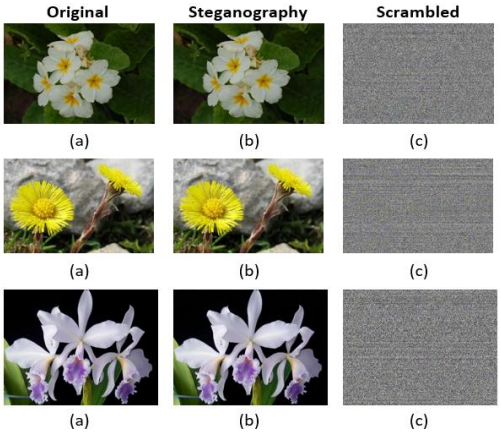


Figure 3: (a) Original image, (b) Image after steganography, and (c) Image after scrambling, for three different examples

$$\text{PSNR} = 10 \times \log\left(\frac{R^2}{\text{MSE}}\right) \quad (3)$$

Table 1 compares the three steganographic methods — PVD, LSB, and PVD+DP (Pixel Value Differencing with Dynamic Programming) — across evaluation metrics for different images in the dataset, using a constant input text for each method. PSNR measures the similarity between the source and stego images. Higher PSNR values indicate less distortion and, consequently, imply better image quality. LSB yields the highest PSNR across all the images tested (around 68-70 dB), indicating nearly negligible distortion. This is supposed to be because LSB alters only the lowest-order bits, giving almost no visual effect. However, LSB is usually susceptible to various types of attack, as the scheme is predictable. PVD achieves a good PSNR, but it is always less than the LSB values (59-67 dB). PVD introduces minimal distortion because it increases the number of pixels when their differences from the original are noticeable, making it more noticeable than LSB but optically minimal. A lower PSNR is traded for higher capacity and slightly improved security, making PVD preferable to LSB. The PSNR values of the DP-enhanced PVD are generally lower than those of both traditional PVD and LSB, ranging from 57 to 63 dB. This decrease in PSNR indicates greater pixel modification, thereby reflecting the effect of DP enhancement on embedding capacity and complexity at the cost of increased visible distortion. This technique focuses on security and capacity, with image hiding achieved through a minor, imperceptible distortion, particularly in applications where additional data needs to be concealed.

The SSIM measures the structural similarity between the original and the steganographically encoded images. Values approaching one and close to each other indicate a nearly negligible difference in perceptibility. LSB provides SSIM values that approach 1 (e.g., 0.999948), indicating high preservation of structural integrity. The low impact on perceptual quality aligns with the high PSNR values provided by LSB. PVD has SSIM values of 0.9997–0.9999, which are slightly lower than those of LSB. The slight drop indicates minor structural distortions due to differential embedding, which is tolerable given PVD's higher capacity and security. For DP-enhanced PVD, the SSIM values remain high, in the range of 0.9996-0.9998; however, they are generally lower than for LSB and traditional PVD. Although this method maintains high structural similarity, the lower SSIM compared to that achieved with LSB and PVD reveals the trade-off between security and capacity that the embedding strategy entails. The MSE evaluates the average squared difference between the pixel values. Lower values reflect fewer deviations from the original image. LSB yields the smallest MSE values, ranging from 0.006 to 0.009 across images. This is expected because LSB involves minimal alteration of pixel values, resulting in the lowest average error among the methods. PVD generally yields higher MSE values than LSB, typically by about 0.01-0.07, since its approach to data embedding exploits differences in pixel values. There is a trade-off: increased error per pixel for greater data capacity, and slightly improved security compared to the LSB method. DP-enhanced PVD reliably generates the highest MSE values, ranging from 0.03 to 0.08. In other words, this method introduces the greatest pixel deviation. It sacrifices some image fidelity to increase robustness and security against steganalysis, thanks to its improved embedding strategy.

4.3. Summary and Comparison

- **LSB:** Offers the highest fidelity with minimal impact on visual and structural quality, as seen in its high PSNR, near-perfect SSIM, and very low MSE. However, it is the least robust to steganalysis and attacks due to its predictable embedding patterns, limiting its security applications.
- **PVD:** Balances fidelity and security, with higher data capacity and robustness than LSB at the cost of slightly lower PSNR, SSIM, and higher MSE. PVD's embedding strategy is more secure because it utilizes pixel differences, making it more difficult to detect with simple steganalysis techniques.
- **DP-Enhanced PVD:** Prioritises embedding capacity and robustness, showing a more substantial deviation in PSNR, SSIM, and MSE. This method is suitable where a slight decrease in image quality is acceptable in exchange for higher security and resistance to steganalysis. DP optimisation increases the embedding rate and complexity, providing enhanced security against statistical attacks and better adaptability in more challenging steganography applications.

In summary, the choice between LSB, PVD, and DP-enhanced PVD should be guided by the specific requirements of the steganographic application.

- LSB for high-fidelity, low-capacity, and low-security applications.
- PVD for balanced security and fidelity with moderate capacity.
- DP-enhanced PVD for high-security applications where capacity and robustness are prioritised over image fidelity.

This comparative analysis demonstrates that DP-based PVD outperforms in terms of embedding capacity and security, though at the cost of some quality trade-offs, making it suitable for scenarios that demand enhanced security. Histogram and RS (Regular and Singular) analysis are important in estimating the effectiveness and security of image steganography methods. His togram analysis provides insight into how data embedding affects an image's structure by examining its pixel intensity

distribution. A smooth histogram that closely matches the original image indicates that the steganography method preserves the integrity of the visual structure, making it more challenging for detection techniques to identify anomalies. RS analysis partitions pixel blocks in an image into regular and singular groups, quantifying these categories using the Regular Block Ratio (RBR) for regular blocks and the Singular Block Ratio (SBR) for singular blocks. This method would detect very subtle variations in the relationships between pixels induced by data embedding, providing another layer of analysis beyond the simple distribution of pixel intensities. Identifies and measures the patterns in pixel groups with a classification declaration of being either" regular" (R) or" singular" (S) relative to a mask m. This method tends to expose minute statistical anomalies introduced by hidden data, even though these changes are very subtle.

- **Mask Application:** A mask m is applied to a group of pixels in the cover image. This mask consists of values -1, 0, and 1, which either flip or preserve pixel values. For example, applying -1 could decrease the pixel value, applying one could increase it, and applying zero would leave it unchanged.

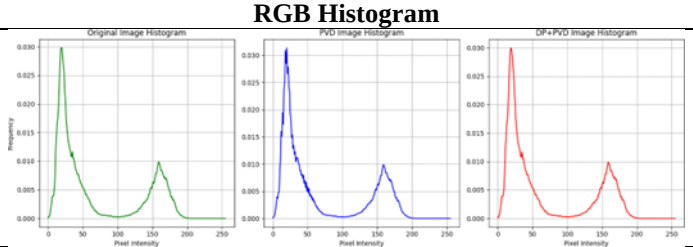
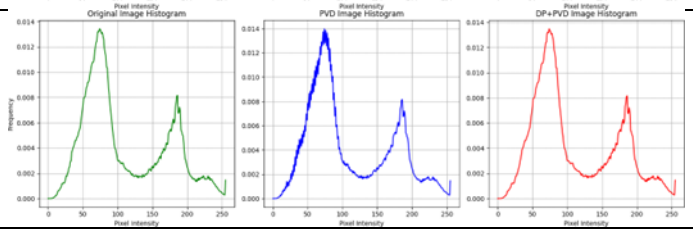
- **Classification of Pixel Groups:**

    - The groups are classified into regular or singular based on their smoothness or randomness after applying the mask.
    - A regular group Rm shows no significant change when the mask is applied, indicating that the pixel values follow a predictable pattern.
    - A singular group, Sm, on the other hand, exhibits irregular changes, indicating that the pattern of pixel values deviates after applying the mask.

- **RS Analysis with and without Mask Inversion:**

    - RS analysis evaluates two sets of groups: one with the mask m and the other with its inverse, −m. This creates four main groups:
    - Rm: Regular group with mask m.
    - R−m: Regular group with inverse mask −m.
    - Sm: Singular group with mask m.
    - S−m: Singular group with inverse mask −m.

In image steganography, the goal is to minimise the difference between these groups (Rm, R-m, Sm, and S-m) so that the presence of hidden data cannot be detected. As embedding rates increase, a widening difference between these groups can indicate the presence of hidden data, making the embedding detectable. The comparison of histograms in our analysis reveals significant differences between PVD-based and DP-based PVD steganographic methods, as shown in Table 2. The histogram of the image modified by PVD remains full of large fluctuations, or" jitters," indicating that the pixel tensities have been changed in a manner that breaks the structure of the original image (Figure 4).

Images	RGB Histogram
	
	

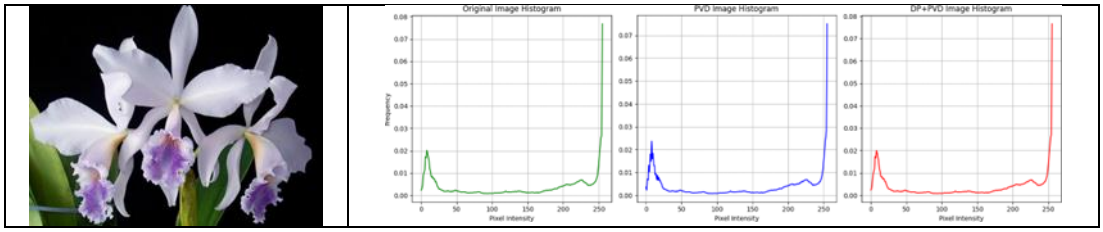


Figure 4: Histogram analysis of sample images: each row displays an original image (left) and its corresponding RGB histogram (right) for three different samples

This suggests that the PVD method, as effective as it is at data embedding, does so in a highly evident manner, thereby making it more susceptible to histogram-based steganalysis. In contrast, the DP-based PVD histogram exhibits a smoother dispersion and is more closely aligned with the original histogram. This reduced variability implies that the DP-based PVD better preserves the natural intensity distribution of pixels, most likely because it uses dynamic programming to optimise the locations of embedding, thereby increasing visual imperceptibility (Table 3).

Table 2: Regular and singular block ratios (rbr and sbr) for each RGB channel under original, pvd, and pvd+dp schemes for three sample images

Image	Channel	Metric	Original	PVD	PVD+DP
	Red	RBR	0.3508	0.3490	0.3380
		SBR	0.6492	0.6510	0.6620
	Green	RBR	0.3625	0.3609	0.3625
		SBR	0.6375	0.6391	0.6375
	Blue	RBR	0.3366	0.3343	0.3366
		SBR	0.6634	0.6657	0.6634
	Red	RBR	0.1783	0.1776	0.1737
		SBR	0.8217	0.8224	0.8263
	Green	RBR	0.2119	0.2113	0.2119
		SBR	0.7881	0.7887	0.7881
	Blue	RBR	0.2097	0.2091	0.2097
		SBR	0.7903	0.7909	0.7903
	Red	RBR	0.2935	0.2904	0.2765
		SBR	0.7065	0.7096	0.7235
	Green	RBR	0.2859	0.2834	0.2859
		SBR	0.7141	0.7166	0.7141
	Blue	RBR	0.2940	0.2905	0.2940
		SBR	0.7060	0.7095	0.7060

Additional support in RS analysis. The block ratios of the original image, regarding the red channel, are approximately 0.3508 for regular and 0.6492 for singular. Changes to all channels have been implemented for the PVD method, with the block ratios for the red channel (regular and singular) set to 0.3490 and 0.6510, and correspondingly for the green and blue channels. Unlike the DP-based PVD approach, which has nearly retained the original ratios in both the green and blue channels, only the red channel differs slightly, with regular values at 0.3380 and singular values at 0.6620. This implies that DP-based PVD introduces fewer detectable distortions to block structures, especially in the red channel, but PVD is more significant in all channels.

The slight red-channel deviation in the DP-based PVD indicates that it is controlled by embedding, reducing detectable changes. In summary, these analyses reflect that the DP-based PVD approach boasts visual imperceptibility and robustness against statistical detection that are stronger than those of conventional PVD. The smoother histogram and smaller RS devices indicate that DP-based PVD obtrusively preserves the statistical properties of the image much better, making it harder to detect via steganalysis. At the same time, although the traditional PVD can embed data across all colour channels, it induces detectable variations in pixel intensity and block patterns, thereby revealing a trade-off between capacity and invisibility. In general, DP-based PVD is more secure for applications that use this method to detect evasion.

4.4. Effectiveness of OS Scrambling

In this section, we test the integrity and security of a scrambling-based PVD steganography approach by analyzing key image properties and evaluating its resistance to various detection methods. A series of tests was conducted to measure the randomness, sensitivity, and structural preservation of the scrambled and stego images, examining their performance in terms of Shannon entropy, Number of Pixel Change Rate (NPCR), Unified Average Changing Intensity (UACI), and pixel correlation. These tests provide insight into the effectiveness of the scrambling method and the stego image's ability to retain the characteristics of the cover image.

4.4.1. Shannon Entropy

Entropy measures randomness or information density in an image. High entropy in scrambled images indicates that the scrambling effectively disrupts the original image structure, resembling noise, while similar entropy in the stego and original images supports the imperceptibility of data embedding.

4.4.2. NPCR (Number of Pixels Change Rate)

It is a metric that quantifies the rate at which pixels in an image change when a transformation, such as encryption or data embedding, is applied. Specifically, NPCR measures the percentage of pixels in an image that differ from their corresponding pixels in a modified version of that image, like an encrypted or steganographically embedded version. This metric is crucial for evaluating an algorithm's ability to generate unpredictable, extensive pixel changes, thereby providing a robust layer of security against detection. If the NPCR value is very high for a particular steganographic or encryption technique, then the algorithm is said to have achieved widespread diffusion. In other words, the embedded or encrypted data has spread in such a manner that it alters a significant portion of the image. Such is useful for applications requiring strong data concealment, as it makes pattern detection through statistical or visual inspection very unlikely. A higher value of the NPCR indicates that even minor data changes or secret messages diffuse throughout the image, thereby generating a more complex and unpredictable pixel structure that better hides the information embedded within. In practice, NPCR is very important for image security, as it establishes whether the embedded or encrypted content diffuses sufficiently throughout the image so that any unauthorised viewer cannot easily view it. For instance, if the encryption algorithm is strong and yields high NPCR values, many image pixels would be changed, making it quite challenging to pinpoint areas that might contain hidden or encrypted information. Applications involving digital watermarking and steganography require a high NPCR value, as this indicates that the transformation process heavily scrambles the original pixel arrangement. Lower detection chances of watermarks or secret messages, and therefore better confidentiality and security, result from such heavy scrambling of the original pixel arrangement.

4.4.3. UACI (Unified Average Changing Intensity)

This focuses on the intensity of pixel changes rather than just their number. UACI calculates the average change in intensity (brightness) for each pixel in the image after the transformation is applied. This metric provides a measure of the perceptual impact of the modifications, assessing whether the transformation is visually or statistically significant across the image. A high UACI value indicates substantial differences in pixel intensity, which can obscure underlying structures in the image, such as patterns or features that might hint at hidden data. In image encryption applications, UACI is critical because it quantifies the extent to which the encryption distorts the image’s overall appearance. A higher UACI value suggests that the transformation process has scrambled pixel intensities, masking the image's visual appearance and preventing an attacker from deducing patterns or extracting meaningful information.

In steganography, UACI measures the degree of visual disturbance, helping ensure that the embedded information is not visible or easily detectable based on intensity analysis. This metric is crucial when the objective is to avoid detection through statistical analysis, as higher UACI values imply that the intensity shifts are pronounced enough to effectively conceal any trace of embedded data. UACI is also valuable in the context of differential attacks, where an attacker might compare two images to identify the presence of hidden data. High UACI values imply that the changes in intensity per pixel are sufficiently varied to prevent reverse engineering. Even if an attacker has access to both the original and the modified image, high UACI suggests that the differences are distributed in a way that does not reveal any specific pattern or clue to the embedded information. Thus, UACI supports the robustness of image encryption or steganography by ensuring the embedded data remains hidden, even under scrutiny from statistical methods.

4.4.4. Pixel Correlation Analysis

Natural images exhibit a strong correlation between adjacent pixels. Scrambling should ideally reduce this correlation, thereby preventing the detection of structured patterns. Low correlation values in the scrambled image indicate that the scrambling is

effective. In contrast, similar correlation values between the original and stego images suggest that embedding preserves the natural image structure, thereby enhancing stealth (Table 3).

Table 3: Results for analysis of the scrambling algorithm

Metric	Channel	Image 1	Image 2	Image 3
Shannon Entropy	Red	6.0472	6.4298	6.0646
(Original)	Green	7.0785	7.0951	7.8227
	Blue	6.7328	7.4884	7.2143
Shannon Entropy	Red	6.0472	6.4298	6.0646
(Stego)	Green	7.0727	7.0951	7.787
	Blue	6.5335	7.3134	7.0627
Shannon Entropy	Red	7.9995	7.9996	7.9993
(Scrambled)	Green	7.9995	7.9995	7.9992
	Blue	7.9994	7.9995	7.9994
NPCR	Red	99.61	99.61	99.59
(%)	Green	99.6	99.61	99.59
	Blue	99.6	99.6	99.6
UACI	Red	50.06	50.04	50.03
(%)	Green	50.01	50.03	49.95
	Blue	50.01	50.05	50.03
Correlation	Red	0.9932	0.9706	0.9793
(Original)	Green	0.9932	0.9839	0.9913
	Blue	0.9942	0.9919	0.9958
Correlation	Red	0.9932	0.9706	0.9793
(Stego)	Green	0.9932	0.9839	0.9908
	Blue	0.9929	0.9901	0.9953
Correlation	Red	0.0017	0.0002	0.0031
(Scrambled)	Green	0.0018	0.0004	-0.00003
	Blue	0.003	0.0018	-0.00003

These tests can collectively confirm the effectiveness of the scrambling approach in enhancing security and the robustness of the steganography method in preserving the integrity of the cover image, making hidden data less detectable. The results of our analysis, presented in Table 3, demonstrate the effectiveness of scrambling and steganographic techniques in maintaining the security and integrity of an image. Shannon entropies for images embedded using our scrambling techniques are uniformly close to 8, as shown by the following values: Red: 7.9995, Green: 7.9995, and Blue: 7.9994 for Image 1. These values indicate a high level of randomness, indicating that the scrambling process transforms the image into a noise-like pattern that effectively conceals any embedded data. The entropy values for the stego images are not much different from the original image and fall in the range of: Red: 6.0472, Green: 7.0727, Blue: 6.5335 for Image 2, meaning that embedding does not alter the statistical properties of images too much, making the stego image appear to be the cover image and making the detection process hard. More solid proof of the scrambling process's strength is provided by the number of pixel changes, which in all images exceeds 99.5%. As a simple example, consider Image 03647: the NPCR values for Red, Green, and Blue are 99.61%, 99.60%, and 99.60%, respectively. Such high NPCR rates are symptomatic of scrambling that produces almost every pixel change in the image, completely shattering any existing patterns. For example, UACI values are very close to 50%; for example, Red is 50.06%, Green is 50.01%, and Blue is 50.01% in Image 03647. This indicates that the intensity variations between the original and scrambled images are quite large, ensuring that the scrambled image is not a likeness of the original.

Pixel correlation analysis indicates that the scrambling technique disrupts spatial continuity. High correlation values are evident in the original images; for example, Red: 0.9932, Green: 0.9932, and Blue: 0.9942 for the sample image, which is typical of natural images. After scrambling, the correlations drop dramatically, in the order of Red: 0.0017, Green: 0.0018, and Blue: 0.0030 for Image 03647, indicating that scrambling is complete rather than fragmenting the spatial relationships between neighbouring pixels. The correlation values of the stego images are similar to those of the source images: Red: 0.9932, Green: 0.9932, Blue: 0.9929 for the sample image. Therefore, the embedding does not distort the original pixel structure. We are confident that our scrambling and steganographic techniques have indeed achieved adequate security and acceptable imperceptibility in our images. With high Shannon entropy values for scrambled images, the scrambling is confident to

appropriately randomise pixel data in an image. In contrast, with minimal influence on entropy in stego images, the hidden information remains nondetectable. Moreover, high NPCR and UACI values confirm that the scrambling algorithm is robust when modifying pixel values and intensities to hide information and the embedded data. The low pixel correlation of scrambled images and the preserved correlation in stego images will support the steganographic method's ability to preserve image integrity while hiding data. These results confirm that integrating image scrambling with PVD-based steganography is secure and efficient, enabling data embedding without any traceable marks.

5. Conclusion

This paper proposes a secure, non-perceptible image steganography system that combines DP and PVD to enhance security and integrates a scrambling technique. In doing so, extensive analysis and comparison studies have been conducted using traditional methods such as the least significant bit and standard PVD, as well as the proposed method, which offers distinct advantages in terms of robustness, high data embedding capacity, and resistance to steganalysis. Although LSB steganography maintained the highest visual fidelity, it demonstrated the lowest security because of its relatively predictable embedding pattern, as evidenced by a high PSNR, nearly perfect SSIM, and very low MSE. Traditional PVD had a suitable balance between fidelity and curity. PVD had lower PSNR and SSIM but greater robustness than the considered attacks. The DP-based PVD technique further enhanced these benefits by increasing the embedding capacity and further complexity, making it more resistant to statistical detection while tolerating a moderate decrease in image quality. This approach enabled greater data hiding with minimal impact on visual quality, as evidenced by the controlled deviations in PSNR, SSIM, and MSE.

The scrambling enhancement added a layer of security by randomly distributing pixel values and intensities. The high Shannon entropy values in the scrambled images confirmed the randomness, with high NPCR and UACI scores showing robustness to changes in pixel intensities, which are needed to avoid attacks based on cryptographic and ganalysis principles. Low pixel correlation in scrambled images indicated successful scrambling, and maintaining correlation in stego images generally preserved image integrity while enhancing imperceptibility. In conclusion, DP-based PVD steganography, as implemented in this paper, provides an advanced, secure, and adaptive approach to covertly embedding data. The method strikes a fine balance between embedding capacity and robustness, albeit at the expense of image quality, for sensitive applications where security is a primary concern. By incorporating optimisation via DP and scrambling, this approach further advances image steganography by providing greater security against visual and statistical detection while maintaining acceptable image fidelity.

Acknowledgment: The authors wish to acknowledge Vellore Institute of Technology for its infrastructure and guidance that facilitated this research. They also thank all contributors whose suggestions and collaboration enhanced the quality of the work.

Data Availability Statement: A standard image dataset was used for testing, and we employed the Oxford 102 Flower Dataset of Nilsback and Zisserman [25], which comprises a collection of images of various flower species with diverse colors and textures.

Funding Statement: This research work and manuscript were conducted without any external financial assistance or funding support from any organization or agency.

Conflicts of Interest Statement: The authors collectively declare that there are no conflicts of interest or competing financial or personal relationships that could have influenced the outcomes of this study.

Ethics and Consent Statement: The authors confirm that this research complies with all established ethical standards, and informed consent was obtained from all participants involved in the study.

References

1. P. N. Andono and D. R. I. M. Setiadi, “Quantization selection based on characteristic of cover image for PVD steganography to optimize imperceptibility and capacity,” *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 3561–3580, 2023.
2. B. G. Banik and S. K. Bandyopadhyay, “Implementation of image steganography algorithm using scrambled image and quantization coefficient modification in DCT,” *in Proc. IEEE Int. Conf. Research in Computational Intelligence and Communication Networks (ICRCICN)*, Kolkata, India, 2015.
3. A. Fahim and Y. Raslan, “Optimized steganography techniques based on PVDs and genetic algorithm,” *Alexandria Engineering Journal*, vol. 85, no. 12, pp. 245–260, 2023.
4. M. Hassaballah, M. A. Hameed, A. I. Awad, and K. Muhammad, “A novel image steganography method for industrial internet of things security,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7743–7751, 2021.

5. Y. Huang, Z. Liu, Q. Wu, and X. Liu, “Robust image steganography against JPEG compression based on DCT residual modulation,” *Signal Processing*, vol. 219, no. 6, p. 109431, 2024.
6. S. A. Jebur, A. K. Nawar, L. E. Kadhim, and M. M. Jahefer, “Hiding information in digital images using LSB steganography technique,” *International Journal of Interactive Mobile Technologies*, vol. 17, no. 7, pp. 167-178, 2023.
7. S. Kaur, S. Singh, M. Kaur, and H.-N. Lee, “A systematic review of computational image steganography approaches,” *Archives of Computational Methods in Engineering*, vol. 29, no. 7, pp. 4775–4797, 2022.
8. X. Liao, J. Yin, M. Chen, and Z. Qin, “Adaptive payload distribution in multiple images steganography based on image texture features,” *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 897–911, 2020.
9. K. Loukhaoukha, J.-Y. Chouinard, and A. Berdai, “A secure image encryption algorithm based on Rubik’s cube principle,” *Journal of Electrical and Computer Engineering*, vol. 2012, no. 1, p. 173931, 2012.
10. L. Meng, X. Jiang, and T. Sun, “A review of coverless steganography,” *Neurocomputing*, vol. 566, no. 1, p. 126945, 2024.
11. P. M. Modak and V. Pawar, “A comprehensive survey on image scrambling techniques,” *International Journal of Science and Research (IJSR)*, vol. 4, no. 12, pp. 814–818, 2015.
12. A. Nair, D. Dalal, and R. Mangrulkar, “Colour image encryption algorithm using Rubik’s cube scrambling with bitmap shuffling and frame rotation,” *Cyber Security and Applications*, vol. 2, no. 8, p. 100030, 2024.
13. F. Pan, J. Li, and X. Yang, “Image steganography method based on PVD and modulus function,” *in Proc. Int. Conf. Electronics, Communications and Control (ICECC)*, Ningbo, China, 2011.
14. R. Patanwadia and R. Mangrulkar, “Divide and scramble — A recursive image scrambling algorithm utilizing Rubik’s cube,” *in Proc. Int. Conf. Recent Trends on Electronics, Information, Communication, and Technology (RTEICT)*, Bangalore, India, 2021.
15. M. Płachta, M. Krzemiń, K. Szczypiorski, and A. Janicki, “Detection of image steganography using deep learning and ensemble classifiers,” *Electronics*, vol. 11, no. 10, pp. 1–14, 2022.
16. A. Pradhan, K. R. Sekhar, and G. Swain, “Adaptive PVD steganography using horizontal, vertical, and diagonal edges in six-pixel blocks,” *Security and Communication Networks*, vol. 2017, no. 1, p. 1924618, 2017.
17. H. Qiu, X. Xu, Z. Jiang, K. Sun, and C. Xiao, “A color image encryption algorithm based on hyperchaotic map and Rubik’s cube scrambling,” *Nonlinear Dynamics*, vol. 110, no. 3, pp. 2869–2887, 2022.
18. A. K. Sahu and G. Swain, “A review on LSB substitution and PVD based image steganography techniques,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 2, no. 3, pp. 712–719, 2016.
19. A. K. Sahu, G. Swain, M. Sahu, and J. Hemalatha, “Multidirectional block based PVD and modulus function image steganography to avoid FOBP and IEP,” *Journal of Information Security and Applications*, vol. 58, no. 5, p. 102808, 2021.
20. M. Sahu, N. Padhy, and S. S. Gantayat, “Multidirectional PVD steganography avoiding PDH and boundary issue,” *Journal of King Saud University – Computer and Information Sciences*, vol. 34, no. 10, pp. 8838–8851, 2022.
21. V. K. Sharma, J. Khandelwal, and S. Singhal, “Identification of best image scrambling and descrambling method for image steganography,” *Multimedia Tools and Applications*, vol. 82, no. 29, pp. 45659–45678, 2023.
22. N. Subramanian, O. Elharrouss, S. Al-Maadeed, and A. Bouridane, “Image steganography: A review of the recent advances,” *IEEE Access*, vol. 9, no. 1, pp. 23409–23423, 2021.
23. B. Vishnu, L. V. Namboothiri, and S. R. Sajeesh, “Enhanced image steganography with PVD and edge detection,” *in Proc. 4th Int. Conf. Computing Methodologies and Communication (ICCMC)*, Erode, Tamil Nadu, India, 2020.
24. M. A. Wani and B. Sultan, “Deep learning based image steganography: A review,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 13, no. 3, p. e1481, 2023.
25. M. E. Nilsback and A. Zisserman, “Automated flower classification over a large number of classes,” *in Proc. 6th Indian Conf. Computer Vision, Graphics and Image Processing*, Bhubaneswar, Odisha, India, 2008.