Evaluating the effectiveness of steganography techniques based on pixel value difference

Denys Fokin[1,*,†], Maryna Yevdokymenko[1,†] and Oleksandr Fediushyn[1,†]

[1]*Kharkiv National University of Radio Electronics, Nauky Ave., 14, Kharkiv, 61166, Ukraine*

Abstract

The paper presents an assessment of the effectiveness of the main methods of steganography based on the pixel difference. The performance of these methods was evaluated using key metrics, signal-to-noise ratio, mean squared error, quality and structural similarity indices, and through pixel difference histogram analysis. To evaluate the effectiveness of the selected steganography methods under the same conditions, a demonstration software implementation for Windows was developed that works with images in shades of gray. The analysis of the performance parameters of the selected steganography methods allows for the formulation of recommendations for their use in the context of accuracy, reliability and efficiency of information embedding, as well as from the point of view of the quality of information embedding and resistance to stegoanalysis.

Keywords

steganography, pixel value difference, performance evaluation

1. Introduction

In today's world, where information is one of the most valuable resources, data protection is becoming increasingly important. Steganography, as the art of concealing information in media files, plays a key role in this area. One method of steganography is pixel value difference-based steganography, which is based on using the difference in pixel values to embed information into an image. This method, due to its ability to effectively mask large amounts of data and adapt the amount of embedded information depending on the characteristics of the image area, providing a better balance between invisibility and capacity, has gained considerable interest among researchers and practitioners [1, 2].

In addition, the growing focus on cybersecurity is driving the development of more sophisticated stegoanalysis techniques aimed at uncovering hidden information. Here, pixel difference-based methods differ due to their ability to better withstand these analysis techniques, as they allow for a more natural embedding of data into images. This makes PVD essential for the development of steganographic systems capable of resisting an expanded range of stegoanalytic attacks [3].

This work focuses on evaluating the effectiveness of such steganographic methods, which were among the most recent to be published. Five such methods were implemented and analyzed to test them under the same experimental conditions. The effectiveness of these methods was evaluated using key metrics, signal-to-noise ratio, mean square error, and quality and structural similarity indices, etc., as well as through widespread analysis of the pixel difference histogram [4], thus assessing not only the quality of information embedding, but also resistance to stegoanalysis.

CH&CMiGIN'25: Fourth International Conference on Cyber Hygiene & Conflict Management in Global Information Networks, June 20–22, 2025, Kyiv, Ukraine

*Corresponding author.

†These authors contributed equally.

denys.fokin@nure.ua (D. Fokin); maryna.yevdokymenko@ieee.org (M. Yevdokymenko); oleksandr.fediushyn@nure.ua (O. Fediushyn)

0009-0002-3282-842X (D. Fokin); 0000-0002-7391-3068 (M. Yevdokymenko); 0000-0002-3600-405X (O. Fediushyn)

2. Methods reviewed

2.1. Eight-directional PVD

The proposed by Gandharba Swain method enhances hiding capacity and PSNR by using eight-directional PVD and LSB substitution, effectively resisting RS and PDH analysis. It exploits edges in multiple directions and combines LSB substitution with PVD for robust security. The algorithm offers two variants: Type 1, with higher PSNR, and Type 2, with greater hiding capacity, allowing users to choose based on their needs. However, the complexity increases due to eight-directional PVD and careful selection of embedding locations, leading to higher computational resource requirements. The method processes 3×3 pixel blocks and calculates eight difference values for each block. Key characteristics include operation on non-overlapping 3×3 pixel blocks, with the central pixel undergoing modified LSB substitution. It uses a (2n+1)-ary notational system for embedding, enhancing the process. The approach is adaptive, embedding more bits in high-textured regions compared to low-textured ones, optimizing hiding capacity [5].

2.2. Adaptive PVD with 2×3-pixel blocks

The proposed by Anita Pradhan, K. Raja Sekhar, and Gandharba Swain adaptive PVD steganography algorithm offers enhanced hiding capacity and reduced detection risk by utilizing horizontal, vertical, and diagonal edges in six-pixel blocks. Its adaptive quantization ranges, based on pixel correlations, improve performance with higher hiding capacity and less distortion, as shown by improved PSNR values. The algorithm is robust against PDH and RS steganalysis, providing better security than traditional methods. However, the complexity of the adaptive approach may increase computational demands, potentially reducing efficiency. The algorithm uses 2 × 3 and 3 × 2 pixel blocks, known as variant 1 and variant 2, respectively, to avoid step effects and enhance capacity and imperceptibility. Overall, it is presented as an advancement by balancing high capacity, security, and image quality [6, 7].

2.3. Overlapped pixel value differencing with modulus function

The OPVDMF method proposed by Aditya Kumar Sahu and Gandharba Swain enhances embedding capacity and PSNR in image steganography by using pixel overlapping, which optimizes space within image blocks. It uses the difference between the first four pixels and the fifth pixel for data embedding, with pixel adjustments to minimize distortion, maintaining image quality. However, the method's complexity may pose challenges, requiring precise calculations during embedding and extraction. Key characteristics include operation on 1x5 pixel blocks and a modulus function to compute stego-pixels, ensuring image integrity. The method shows competitive performance in PSNR, embedding capacity, and execution time compared to existing techniques. Its security is verified using RS analysis, enhancing its reliability for secure data hiding [8].

2.4. Overlapped pixel value differencing

The OPVD method proposed by Aditya Kumar Sahu and Gandharba Swain in the same paper as OPVDMF is a steganography technique designed to enhance EC and PSNR by utilizing pixel overlapping. This method divides image blocks into four sub-blocks, using the first and fifth pixels, the fifth and second, the third and fifth, and the fifth and fourth pixels for data embedding. This approach allows for increased EC as one pixel is used in multiple sub-blocks, significantly improving performance compared to existing techniques. The OPVD method operates on 1x5 pixel blocks, ensuring efficient use of space and maintaining image quality through pixel adjustments to minimize distortion. A key advantage of OPVD is its competitive PSNR, with a reported value of 37.01 dB, indicating high image quality post-embedding. Additionally, the method demonstrates a slightly reduced execution time compared to other methods, enhancing its practicality. However, the complexity of the OPVD method may present challenges, requiring precise calculations during the embedding and extraction phases. Despite this, its

security is robust, successfully resisting RS analysis, which verifies its reliability for secure data hiding [8].

2.5. Quotient value differencing and pixel value correlation

The proposed by Reshma Sonat and Gandharba Swain steganography method combines quotient value differencing (QVD) with pixel value correlation (PVC) to enhance data hiding capabilities. This approach is designed to overcome the fall-off boundary problem (FOBP) commonly associated with PVD techniques, ensuring that the stego-images remain imperceptible. The method involves a two-stage data embedding process on 3x3 pixel blocks, where QVD and remainder substitution are applied to five central pixels, followed by PVC embedding on the four corner pixels. Additionally, the method demonstrates robustness against detection by RS and PDH tests, as the PDH curves do not exhibit a zigzag pattern, and the RS test is unable to detect the steganography technique. However, the time complexity of the method is linear, depending on the length of the secret data, which may impact performance in scenarios involving large data volumes. Overall, the proposed method offers a novel and effective solution for secure data hiding with improved performance metrics [9].

3. Parameters of the effectiveness of steganography methods

3.1. Mean squared error

MSE, or Mean Squared Error, in steganography, is a statistical measure of the difference between an original image and an image that has been altered using steganographic techniques. This is a quantitative indicator that allows you to evaluate the degree of changes made to the image when embedding a secret message. The formula for calculating MSE is:

$$\text{MSE} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} (p_{ij} - q_{ij})^2 ,$$

where p_{ij} is the pixel of the cover image, q_{ij} is the corresponding stego pixel, and m and n are the pixel dimensions of the image. The sum passes across all pixels of the image, and the square of the intensity difference of each pixel in the original and steganographed images is calculated.

The smaller the MSE, the smaller the difference between the original and the steganographed image, which is generally considered preferable in the context of steganography because it means that the changes made to embed the message are less noticeable [9, 10].

3.2. Peak Signal-to-Noise Ratio

PSNR, or Peak Signal-to-Noise Ratio, in steganography, is a measure used to evaluate image quality after applying steganographic techniques. It measures the ratio of the maximum possible signal strength (i.e., the original image) to the noise power made by changes made when embedding a secret message. The formula for calculating PSNR is as follows:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{M^2}{\text{MSE}} \right) , \tag{1}$$

where M is the maximum possible pixel value of an image (e.g., 255 for an 8-bit image) and MSE (Mean Squared Error) is the root mean square error between the original and steganographed images.

The PSNR is higher when the MSE error is smaller, which means that the steganographed image is closer to the original. A high PSNR often indicates better steganographic image quality, but it is important to consider that it may not always adequately assess the visibility of steganographic changes, especially in the context of human perception [10].

3.3. Structural Similarity Index Measure

The SSIM, or Structural Similarity Index Measure, in steganography, is a method of assessing the quality of images based on human perception. It is used to measure the similarity between two images, usually between the original and steganographed (altered) images. SSIM more accurately reflects visual changes in an image than MSE or PSNR because it takes into account changes in structure, brightness, and contrast. The formula for calculating SSIM is:

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

where x and y are the image windows in the original and steganographed images, respectively. μ_x, μ_y are the average pixel values in x and y. σ_x^2, σ_y^2 are the pixel variance in x and y. σ_{xy} is covariance between x and y; and c_1, c_2 are constant variables to avoid division by zero.

SSIM values close to 1 usually indicate high similarity between images, while values closer to 0 indicate low similarity [11].

3.4. Quality Index

QI, or Quality Index, also known as the Universal Quality Index (UQI), is a metric used to evaluate the quality of an image. This index is used to compare two images, usually the original and modified (e.g., steganographed). UQI evaluates the similarity between these images in three aspects: brightness, contrast, and structure.

$$Q = \frac{4\sigma_{xy}\,\overline{p}\,\overline{q}}{(\sigma_x^2 + \sigma_y^2)\,[(\overline{p})^2 + (\overline{q})^2]}, \tag{3}$$

where $\overline{p}$ denotes the average pixel value of the original image, $\overline{q}$ denotes the average pixel value of the stego image, σ_x^2 denotes the standard deviation of the pixel values of the original image, σ_y^2 denotes the standard deviation of the pixel values of the stego image, and σ_{xy} is covariance between the original and stego image.

The UQI can have a value between -1 and 1, where values closer to 1 indicate high similarity between images. The maximum value can be 1 if the original image and the same image are the same [12].

3.5. Bits Per Byte and Bits Per Pixel

BPB or Bits Per Byte measures the number of bits of a secret message that can be embedded in each byte of a medium. A high BPB means that more information can be embedded in the medium, but it can increase the risk of steganography detection. It is calculated using the formula:

$$\mathrm{BPB} = \frac{\text{hiding capacity}}{\text{image size in bytes}}. \tag{4}$$

BPP or Bits Per Pixel measures the number of bits of a secret message that can be embedded in each pixel in an image. A higher BPP allows you to embed more information into an image, but it can increase the chances of detection or affect the visual quality of the image. It is calculated using the formula [9, 7]:

$$\mathrm{BPP} = \frac{\text{hiding capacity}}{\text{image size in pixels}}. \tag{5}$$

4. Choice of development tools

To demonstrate the capabilities of the selected steganography methods and evaluate their effectiveness under the same conditions, a demonstration software implementation for Windows working with grayscale images was developed. In the desktop program, you can select an image, enter the text to embed, and get the result in the form of five stegoimages in which information is embedded according

Table 1
Embedding Options for 512X512, 100%

Name	Hiding T, ms	Extract. T, ms	BPB	BPP	HC, bits
8-Directional	143	65	2.1352	2.9766	1560544
QVD/PVC	412960	964677	2.0726	3.0619	1605296
Adaptive	324797	487971	1.3369	1.8685	979664
OPVD	371353	758063	1.7994	2.5401	1331792
OPVDMF	435504	1047044	2.1690	3.0618	1605280

to the appropriate algorithms. It is also possible to retrieve text that has been extracted using the appropriate extraction algorithms for each of the methods. After carrying out these operations, it is possible to compare the results of the performance evaluation parameters, which are calculated for each method.

The choice of C#, combined with Windows Forms and System.Drawing, creates a strong platform for developing and demonstrating effective pixel difference-based steganography techniques. This allows not only the efficient implementation of steganography algorithms, but also provides convenience in visualizing the results necessary for the analysis and comparison of different techniques. Additionally, C#'s ability to seamlessly integrate with other technologies and libraries provides flexibility in expanding and improving the project.

5. Results of the evaluation of perfomance parameters

For verification, we use standard gray images (8 bits per pixel) from the SIPI database with a size of 512x512 pixels and 256x256 pixels. Let's consider and comment on the results of several experiments: Author names can have some kinds of marks and notes:

- 100% embedding capacity in a 512x512 textured image (with a sufficient number of edge zones) with a size of 256 kB;
- 100% embed capacity in 256x256 mostly smooth image size 256 kB;
- 6.4 kilobytes of secret message in a 256x256 medium-textured image of 64 kB.

5.1. 100% capacity of embedding in the image 512x512

Let's start with an image with the following options:

- 512x512 pixels;
- 256 kilobytes.

First, we will embed the maximum amount of data in the image, to do this, we will generate a random message of 100,000 bytes to check the maximum embedding capacity for each method and get the following result.

As for the integrity of the message, all methods showed good results, we did not find any significant artifacts in the extracted data.

It can be seen (Table 1) that OPVDMF has the highest embed capacity, but at the same time it has the highest embed-in and extraction time. The best results here, in our opinion, have an eight-directional method that embeds and extracts messages very quickly, while having a capacity slightly lower than OPVDMF and QVD/PVC – 15600000 bits versus 1600000 for OPVDMF and QVD/PVC.

According to the performance indicators (Table 2), it can be seen that QVD/PVC shows the worst results, which coincides with the visual perception of the steg image. The adaptive method has the best results, and the eight-directional method, taking into account its results in capacity and speed.

If you look at the analysis using the pixel difference histogram (Figure 1), you can see that all the histograms of stego images are significantly different, but the most natural one can be noted in the eight-directional method.

Table 2
Perfomance Score for 512X512, 100%

Name	MSE	PSNR	QI	SSIM
8-Directional	11.418	37.554905	0.992221	0.999514
QVD/PVC	53.05919	30.883197	0.965344	0.997889
Adaptive	4.897251	41.231279	0.996743	0.999812
OPVD	13.32119	36.885374	0.991102	0.999295
OPVDMF	36.54034	32.503077	0.978046	0.998342

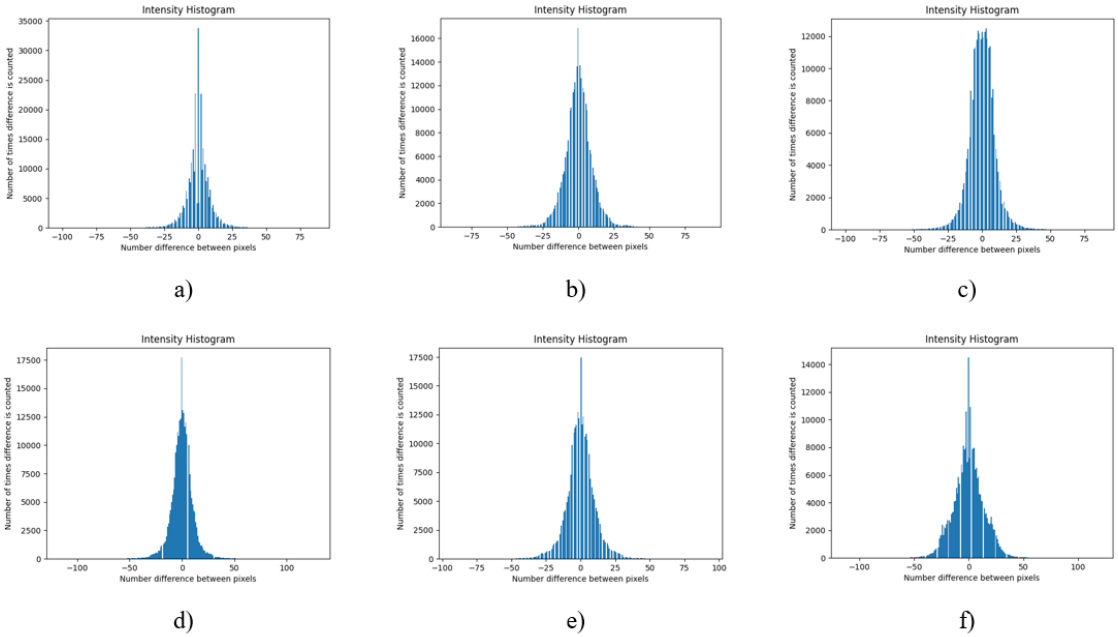

Figure 1: PDH analysis of stegoimages 512x512 at 100% capacity: a) original image; b) 8-Directonal (V-1); c) Adaptive PVD; d) OPVD; e) OPVDMF; f) QVD/PVC.

Table 3
Embedding Options for 256X256, 100%

Name	Hiding T, ms	Extract. T, ms	BPB	BPP	HC, bits
8-Directional	40	19	2.4246	2.9766	390096
QVD/PVC	165795	126730	2.5238	3.9160	513296
Adaptive	90581	44211	1.3602	1.6844	220800
OPVD	99896	57508	2.0023	2.4862	325872
OPVDMF	128892	90081	2.4071	3.0375	398144

5.2. 100% image embedding capacity 256x256

Next, let's run tests with an image with the following parameters:

- 256x256 pixels;
- 64 kilobytes;
- Large Smooth Area.

In this case, we will also embed the maximum number of bits.

In terms of embedding efficiency (Table 3), if we exclude the QVD/PVC and OPVDMF methods due to their poor performance in such conditions, the eight-directional method is the clear winner. Expectedly,

Table 4
Perfomance Score for 256X256, 100%

Name	MSE	PSNR	QI	SSIM
8-Directional	36.91248	32.459071	0.982895	0.996154
QVD/PVC	70.93866	29.621973	0.968691	0.996033
Adaptive	5.840347	40.466416	0.99737	0.999915
OPVD	18.63626	35.427215	0.991583	0.996787
OPVDMF	43.99612	31.696659	0.980903	0.997345

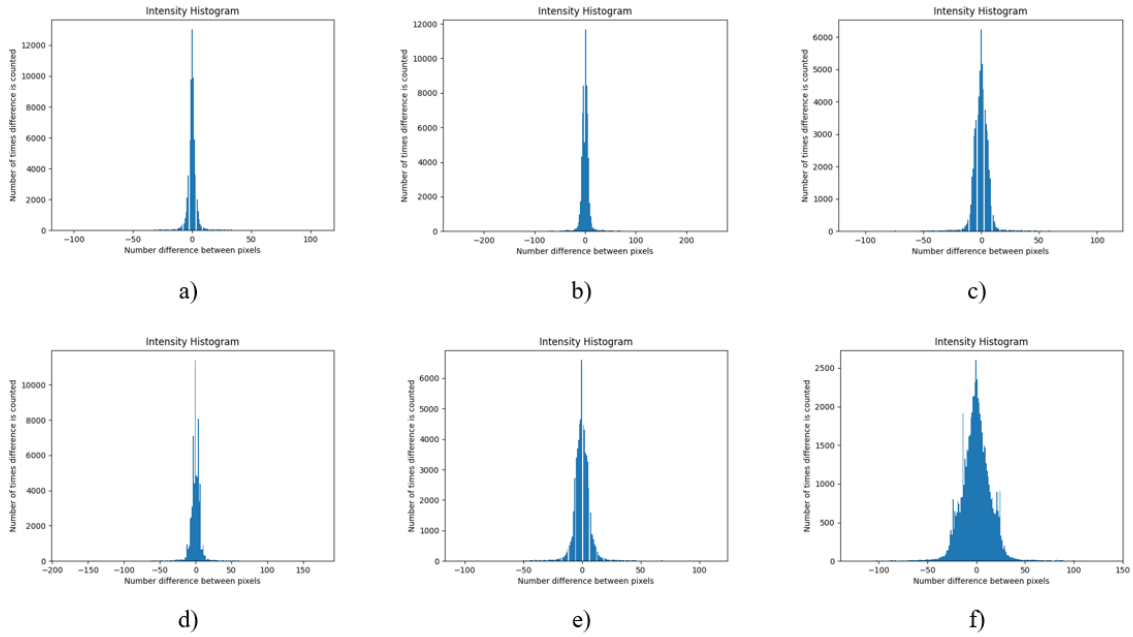

Figure 2: PDH analysis of stegoimages 256x256 at 100% capacity: a) original image; b) 8-Directonal (V-1); c) Adaptive PVD; d) OPVD; e) OPVDMF; f) QVD/PVC.

according to the results of the performance evaluation (Table 4), QVD/PVC and OPVDMF show the worst results, which coincides with the visual analysis of the image. The adaptive method has a clear advantage, being significantly ahead of the eight-directional and OPVD methods.

When analyzed using the pixel difference histogram (Figure 2), the histograms can be arranged by plausibility as follows: eight-directional method; Adaptive PVD; OPVD; OPVDMF; QVD/PVC.

There is a problem in extracting a message from such an image, the smoother the image, the more artifacts from embedding are visible on it, and also the worse the integrity of the message.

5.3. 10% of image size 256x256

Now let's check the ideal conditions for each method, as a container – an image of the surface of the echo with a sufficient number of edge zones and the presence of smooth ones. Let's embed 10% of the image size with each method – 6400 bytes.

As a result of the extraction, all methods did an equally good job of embedding and deleting the message.

Based on the embedding metrics (Table 5), since the number of embedded bits, and therefore the number of bits per byte and per pixel, is the same, we can only estimate the performance of embedding and extracting. In this case, the eight-directional method is again far ahead of the others. The slowest method is adaptive. The fastest are QVD/PVC, OPVD and OPVDMF are somewhat slower and at the

Table 5
Embedding Options for 256X256, 10%

Name	Hiding T, ms	Extract. T, ms	BPB	BPP	HC, bits
8-Directional	10	4	0.5303	0.7844	102720
QVD/PVC	4111	6321	0.5242	0.7849	102720
Adaptive	8030	11158	0.5280	0.7839	102720
OPVD	6821	9604	0.5273	0.7839	102720
OPVDMF	6238	8654	0.5287	0.7839	102720

Table 6
Perfomance Score for 256X256, 10%

Name	MSE	PSNR	QI	SSIM
8-Directional	3.75798	42.381258	0.997558	0.999727
QVD/PVC	13.35873	36.873150	0.991394	0.999549
Adaptive	1.925018	45.286455	0.998753	0.999892
OPVD	8.26825	38.956667	0.994652	0.999431
OPVDMF	13.39502	36.861370	0.991408	0.999396

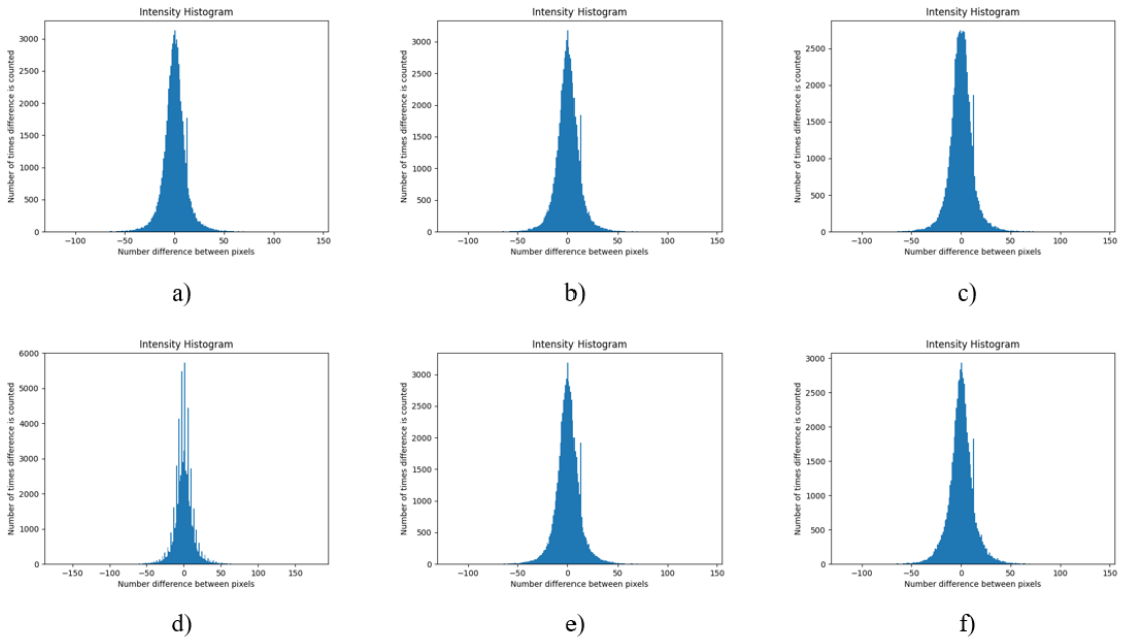

Figure 3: PDH analysis of stegoimages 256x256 at 10% capacity: a) original image; b) 8-Directonal (V-1); c) Adaptive PVD; d) OPVD; e) OPVDMF; f) QVD/PVC.

same level.

According to the results of the performance evaluation (Table 6), the QVD/PVC and OPVDMF methods, as it was noticeable visually in the stego images, have approximately the same and worse indicators. Among others, the adaptive method is the most effective.

Based on the pixel difference histograms (Figure 3) you can see that QVD/PVC is closest to the original.

6. Conclusions

In conclusion, it can be noted that steganography based on pixel difference is an important and promising direction in the field of information security. Considering the various steganography techniques presented in this paper, its significance in the context of modern data security and privacy requirements becomes apparent. Methods that have been analyzed and compared in detail, including the eight-directional method, the adaptive PVD method, the OPVD method, the OPVDMF method, and the QVD/PVC method demonstrate the depth and complexity of modern steganography. The analysis of the efficiency parameters of the selected steganography methods gives an idea of the importance of these methods in the context of accuracy, reliability, and efficiency.

Generally, PVD methods can be used in digital forensics to embed metadata within images. This metadata can include information about the origin, authenticity, and history of the digital content, which is crucial for forensic investigations. The adaptive PVD method, for instance, offers high capacity and security, making it suitable for embedding detailed forensic data without compromising image quality. Also, such methods can be adapted for network steganography, where data is hidden within network protocols. This can be used to transmit sensitive information across networks without detection. For example, the eight-directional method, which is robust against detection by RS and PDH tests, could be adapted to hide data within network packets, ensuring secure communication over potentially insecure channels.

Based on the results of the software implementation and the tests carried out, it can be concluded that the eight-directional (V-1) method has the highest speed of operation and fairly balanced other characteristics while having the problem of going beyond the values from 0 to 255 (FOBP). The adaptive method and the OPVD method showed the greatest ability to embed in mostly smooth images. The OPVDMF method, in my opinion, has shown the worst results, it has a FOBP problem, it does not integrate well into smooth areas of the image and is slow, but at the same time it is very easy to implement programmatically. The most difficult in terms of software implementation was the method based on the difference in coefficient values and the correlation of pixel values, it demonstrates better embedding capacity and has a fairly natural histogram of the pixel difference, this method really solves the FOBP problem, while being slow and poorly coping with smooth areas of the image. In general, we can conclude that the adaptive method, based on the results of a comprehensive analysis, is the most optimal: it has a simple software implementation, FOBP has not been noticed for it, has good performance indicators, and is quite invisible to the human eye. But we pay for such advantages with average performance, poor resistance to steganoanalysis, and the smallest embed capacity.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] N. Hamid, A. Yahya, R. B. Ahmad, O. Al-qershi, Image steganography techniques: An overview, International Journal of Computer Science and Security 6 (2012) 168–187.

[2] G. Swain, S. Lenka, Classification image steganography techniques in spatial domain: A study, International Journal of Computer Science Engineering and Technology 5 (2014) 219–232.

[3] A. Singh, M. Rawat, A. Shukla, A. Kumar, B. Singh, An overview of pixel value differencing based data hiding techniques, in: Proceedings of the International Conference on Contemporary Computing (IC3), 2018, pp. 1–3. doi:10.1109/IC3.2018.8530673.

[4] X. Zhang, S. Wang, Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security, Pattern Recognition Letters 25 (2004) 331–339. doi:10.1016/j.patrec.2003.10.014.

[5] G. Swain, Digital image steganography using eight-directional pvd against rs analysis and pdh analysis, Advances in Multimedia 2018 (2018). doi:10.1155/2018/4847098.

[6] A. Pradhan, R. S. Krovi, G. Swain, Adaptive pvd steganography using horizontal, vertical, and diagonal edges in six-pixel blocks, Security and Communication Networks 2017 (2017). doi:10.1155/2017/1924618.

[7] A. Pradhan, A. K. Sahu, G. Swain, K. R. Sekhar, Performance evaluation parameters of image steganography techniques, in: 2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS), Bangalore, India, 2016, pp. 1–8. doi:10.1109/RAINS.2016.7764399.

[8] A. K. Sahu, G. Swain, Pixel overlapping image steganography using pvd and modulus function, 3D Research 9 (2018). doi:10.1007/s13319-018-0188-5.

[9] R. Sonar, G. Swain, Steganography based on quotient value differencing and pixel value correlation, CAAI Transactions on Intelligence Technology 6 (2021) 504–519. doi:10.1049/cit2.12050.

[10] A. Kumar Rana, A review of comparison techniques of image steganography (2012).

[11] D. I. M. Setiadi, Psnr vs ssim: imperceptibility quality assessment for image steganography, Multimedia Tools and Applications 80 (2021) 8423–8444. doi:10.1007/s11042-020-10035-z.

[12] C.-C. Chang, W.-C. Wu, Reversible quantization-index modulation using neighboring correlation, in: Y. Shi, H. Kim, S. Katzenbeisser (Eds.), Digital Watermarking. IWDW 2007, volume 5041 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 211–225. doi:10.1007/978-3-540-92238-4_17.