Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/

Increasing The Payload of Reversible Data Hiding Scheme Using Modified Pixel Value Ordering (PVO)

Puteri Awaliatush Shofro[1]*, Ari Moesriami Barmawi[2]
[1]Universitas Negeri Yogyakarta
[2]Universitas Telkom
Email: puteriawaliatush@uny.ac.id[1]

Abstract
Reversible Data Hiding (RDH) is a kind of data hiding technique that allows the embedded data can be retrieved as needed and can restore the stego image exactly as the original image after the extraction of embedded data. Recently, RDH has been applied for medical record management, where any distortion of digital images is not allowed. Many new RDH techniques have been developed such as Pixel Value Ordering (PVO). This method is very popular because it has the Management of Medical Records: Facts andFigures for Surgeons advantage of reducing the number of pixels shifted, such that it can improve the stego image quality. However, the existing method has a low embedding capacity (31%). To overcome this problem, this paper proposed a modified Pixel Value Ordering where the difference value is expanded after sorting the pixel values resulting from the embedding process based on different values of 0 and 1. In this case, the difference values of -2, 0, 1, and 3 are used to embed the data, instead of only using the difference values 0 and 1. The proposed method also uses the frequency of bit 0 and 1 in the message to determine the pixels mapping that can be used to embed the message. The experiments results show that the proposed method achieved an average embedding capacity (37%) and obtained a higher PSNR value of 61 dB.

Keywords: Reversible Data Hiding (RDH); Medical record; Medical images; Pixel Value Ordering (PVO).

Introduction

Medical images are a very important part of a patient's records and information. It is important for doctors and medical institutions to properly maintain patient records for 2 important reasons. The first is to assist in proper patient evaluation and to plan treatment protocols. The second is that the legal system relies primarily on documentary evidence in cases of medical negligence (Bali et al., 2011). Therefore, medical images must be maintained properly to serve the interest of the doctor as well as his patient.

Recently, RDH has been applied for medical record management such as the method proposed by Raju et al. (Upendra Raju & Amutha Prabha, 2018), where any distortion of digital images is not allowed.

There are two spatial-based data hiding techniques, lossy and lossless data hiding. One of them is Reversible Data Hiding (RDH), a lossless data hiding (Shi et al., 2016). The stego image can be recovered after the embedded data is extracted as the original image.

The main research directions of the current RDH techniques mainly focused on how to enhance the hiding capacity and to enhance the stego image quality (Lu et al., 2016). These methods use the prediction of the distribution of error values, by finding the difference between a pair of adjacent pixel values then finding the value that frequently occurs and considering

them as the peak value points, while the values that have not yet appeared considers as a point value of zero.

Several RDH methods have been proposed to improve hiding capacity (Hong et al., 2009; Thodi & Rodríguez, 2007; Tian, 2003). Confidential information is embedded based on the prediction error generated by the original pixel value and the predicted value. The more accurate the prediction value, the smaller the error, and the greater the hiding capacity. In 2013, Li et al. (Li et al., 2013) proposed a new RDH prediction method called Pixel Value Ordering (PVO) to further improve the prediction accuracy. The advantage of this method is reducing the number of pixels shifted, such that it can improve the stego image quality. RDH schemes based on the PVO method generally realize the prediction procedure by employing the sorted pixels in fixed block size. The confidential information is embedded based on the prediction error of the sorted pixels from the highest to the lowest. The pixel value difference between the first and second pixel of a block, and between the third and fourth-pixel values are then calculated. To improve the performance of Li's method, Peng et al. (Peng et al., 2014) introduce the Improved PVO method (IPVO), which assigns an index in the block and sorts the pixel values to obtain the prediction error. Furthermore, Wang et al. (Wang et al., 2015) modify Peng's method(Peng et al., 2014), by introducing a dynamic block division algorithm, which can adaptively determine the block size according to the complexity of the region. Thus, subtle areas can be fully used to embed secret messages. Wang's method uses the complexity of a set of pixels to improve the secret message hiding capacity.

Lu et.al. (Lu et al., 2018) proposed a method to improve both Peng's (Peng et al., 2014) and Wang's method (Wang et al., 2015) by analyzing the variance of block in the image to obtain the block complexity and applying a quantification strategy to quantify pixels for ensuring pixels that are invertible. Wang's method adopts an adaptive threshold generation mechanism to find the appropriate threshold. This mechanism is applied for different images to achieve higher image quality and hiding capacity. This method also uses the difference in values of 0 and 1 to embed a message. Since the embedding capacity is used to embed the secret message and other additional information, then the number of characters of the secret messages that can be embedded in less than the maximum embedding capacity. For increasing the payload, the proposed method modifies the PVO scheme where the prediction error is expanded after sorting the pixel values resulting from the embedding process based on different values of 0 and 1. In this case, the difference values of -2, 0, 1, and 3 are used to embed the data, instead of only using the difference values 0 and 1. The proposed method also uses the frequency of bit 0 and 1 in the message to determine the pixels mapping that can be used to embed the message. In this case, the pixel value which is frequently occurred in the messages is embedded into the pixel difference that is frequently occurred in the cover image. Based on the experiment result, it is proven that the embedding capacity of the proposed method is higher than the previous method. However, the PSNR of the proposed method is higher than the previous method because there are fewer pixel changes in the previous method (Lu et al., 2018) compared with the proposed one.

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/

Method

This section discusses the modified Pixel Value Ordering method, which consists of two main processes, embedding, and extraction process. The embedding process is used by the sender to embed the secret message into the cover image, while the extraction process is used by the receiver to read the secret message.

a. Embedding Process

The embedding process in the proposed method consists of three processes, pre-processing of the cover images and secret messages, and the PVO embedding. The overview of the embedding process is shown in Figure 1.

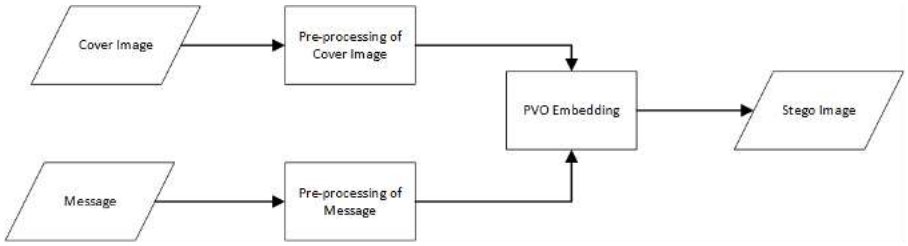


Figure 1. Embedding process

The proposed method is a modification of the PVO embedding method. In modified PVO, the pixel difference values that are used to embed secret messages are -2, 0, 1, and 3. To reduce the number of pixels that are shifted during the hiding process, a limit on the pixels that can be shifted is used. The pixel limit that can be shifted depends on the location of the pixel that has a difference with the next pixel more than 2 (for embedding (0,1) - mode) and more than 4 (for embedding (-2,3) - mode).

b. Pre-processing of Cover Image

In the pre-processing, the cover image has to be divided into non-overlapping blocks of size 2 x 2. Let X = $\{x_1, x_2, x_3, x_4\}$ be the block and total number of blocks is N. The pixel value in each block is sorted out in ascending order to obtain $Xsorted = \{x_{\Delta(1)}, x_{\Delta(2)}, x_{\Delta(3)}, x_{\Delta(4)}\}$, where Δ is the index represented the pixel position after sorting process. Pixel values is sorted in a sequence from the smallest pixel value to the largest one, ($x_{\Delta(1)} \leq x_{\Delta(2)} \leq x_{\Delta(3)} \leq x_{\Delta(4)}$). Then, the pixel location map $LM = \{LM_1, LM_2, \dots, LM_N\}$ is recorded. LM is the overflow (or underflow) indicator, where the value of the indicator is determined as follows..

$$LM = \begin{cases} 1 & \text{if } (x_{\Delta(1)} = 0) \text{ or } (x_{\Delta(4)} = 255) \\ 2 & \text{if } (x_{\Delta(1)} = 1 \text{ or } 2) \text{ or } (x_{\Delta(4)} = 253 \text{ or } 254) \\ 0 & otherwise \end{cases} \quad (1)$$

If LM=1 then the block cannot be used to embed data based on the difference of (0,1), while if LM=2, the block cannot be used to embed data based on the difference of (-2,3). If LM=0 the

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/

block can be used to embed the secret message, either based on the difference of (0,1) or (-2,3). All LM are recorded and further compressed using arithmetic coding (Coding, 2008). The compression results are stored in the metadata along with other additional information used to extract the secret message.

c. Pre-processing of the Secret Message

Before the message is embedded into the cover image, the message length is calculated as well as the number of bits 1 and 0 contained in the message. Character in the message is converted to ASCII, and further converted it to binary numbers, where each character is represented by 8 bits. After obtaining the binary number of all characters used in the message, the message length is calculated. Furthermore, the total number of bits 0 and 1 in the message is calculated. If the number of bit 0 is greater than the number of bit 1, then k=0, which means that the pixel to be embedded with message bit 0 will not change its value. Otherwise, if k = 1, the pixel will not change its value when it is embedded by message 1.

d. Modified PVO Embedding

Modified Pixel Value Ordering (PVO) increased the embedding capacity by extending the embedding limit with (-2,3). From the result of the pre-processing step, we get LM information for each block, the message length in bits, and the numbers of bit 0 and 1. The proposed method proposed the modification of maximum difference (*dmax*) and minimum difference (*dmin*) that can be embedded by information to improve the hiding performance. Dmin and dmax are calculated using equations similar to a modification of the previous method.

In the previous method, each pixel in the block that has *dmin* value outside the range of 0 and 1 needs to be shifted. The impact of the shifting is the decreasing of PSNR. To solve this problem, the proposed method does not shift all pixels in the blocks that have *dmin* value outside the ranges of 0 and 1 for embedding (0,1) – mode and the pixel that have *dmin* value outside the range of -2 and 3 for embedding (-2,3) – mode, as well as the *dmax*. To reduce the number of pixels that are shifted, the range of *dmin* and *dmax* value for the shifted pixel must be observed. The following are the steps to determine the limit of the shifted pixel.

1. Sorting all *dmin* and *dmax* values in an image from the smallest to the largest value.
2. For embedding (0,1) – mode, separate *dmin* values < 2 and *dmin* values > -1
3. The shifted pixel is the smallest pixel of the block which has $2 \leq dmin$ and $dmin \leq -1$
4. If the $dmin_i$ is outside the range between -1 and 2, it is necessary to check whether the difference between the $dmin_i$ and the next *dmin* is $dmin_{i+1}$ (outside the range between –1 and 2) is greater than or equal to 2. If the difference between this two *dmin* is greater than or equal to 2, then the smallest pixels on the block that have the $dmin_{i+1}$ and all *dmin* after the $dmin_{i+1}$ do not need to be shifted. On the other hand, if the difference between this two *dmin* is smaller than 2, then the $dmin_{i+1}$ is shifted. Note: index i+1 for $dmin \leq -1$, is the index of *dmin* which is smaller than $dmin_i$. For $dmin \geq 2$, index i+1 indicates the index of *dmin* is greater than $dmin_i$.
5. For embedding (-2,3) – mode separate *dmin* values > 4 and *dmin* values < -3
6. The shifted pixel is the largest pixel of the block which has $4 \leq dmin$ and $dmin \leq -3$
7. If the $dmin_i$ is outside the range between -3 and 4, then it is necessary to check whether the difference between the $dmin_i$ and the next *dmin* is $dmin_{i+1}$ (outside the range between

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/

-3 and 4) is greater than or equal to 4. If the difference between this two *dmin* is greater than or equal to 4 then the smallest pixels on the block that have the $dmin_{i+1}$ and all *dmin* after $dmin_{i+1}$ do not need to be shifted. Conversely, if the difference between this two *dmin* is less than 4, then the $dmin_{i+1}$ is shifted. Note: index i+1 for *dmin* ≤ -3, is the index of *dmin* which is smaller than $dmin_i$. For *dmin* ≥ 4, index i+1 indicates the index of *dmin* is greater than $dmin_i$.

8. For embedding (0,1) – mode separate the values of *dmax* values < 2 and *dmax* values > -1
9. The shifted pixel is the largest pixel of the block which has 2 ≤ *dmax* and *dmax* ≤ -1
10. If the $dmax_i$ is outside the range between -1 and 2, then it is necessary to check whether the difference between the $dmax_i$ and the next *dmax*, is $dmax_{i+1}$ (outside the range between -1 and 2) is greater than or equal to 2. If the difference between this two *dmax* is greater than or equal to 2, then the largest pixel on the block that has $dmax_{i+1}$ and all *dmax* after $dmax_{i+1}$ do not need to be shifted. Conversely, if the difference between this two *dmax* is smaller than 2, then the $dmax_{i+1}$ is shifted. Note: index i+1 for *dmax* ≤ -1, is the index of *dmax* which is smaller than $dmax_i$. For *dmax* ≥ 2, index i+1 indicates the index of *dmax* which is greater than $dmax_i$.
11. For embedding (-2,3) – mode separate *dmax* values > 4 and *dmax* values < -3
12. The shifted pixel is the largest pixel of the block which has 4 ≤ *dmax* and *dmax* ≤ -3.
13. If the $dmax_i$ is outside the range between -3 and 4, then it is necessary to check whether the difference between the $dmax_i$ and the next *dmax*, is $dmax_{i+1}$ (outside the range between -3 and 4) is greater than or equal to 4. If the difference between this two *dmax* is greater than or equal to 4, then the largest pixel on the block that has $dmax_{i+1}$ and all *dmax* after $dmax_{i+1}$ do not need to be shifted. On the other hand, if the difference between this two *dmax* is smaller than 4, then the $dmax_{i+1}$ is shifted. Note: index i+1 for *dmax* ≤ -3, is the index of *dmax* which is smaller than $dmax_i$. For *dmax* ≥ 4, index i+1 indicates the index of *dmax* which is greater than $dmax_i$.

For embedding a secret message using the modified PVO embedding method, two embedding processes are introduced. The first one is embedding process based on *dmin* = 0 or *dmin* = 1 as well as *dmax* = 0 or *dmax* = 1. The second embedding process is based on *dmin* = -2 or *dmin* = 3 as well as *dmax* = -2 or *dmax* = 3. The second embedding process is only used if the length of the message is greater than the embedding capacity when using the embedding process based on *dmin* = 0 or *dmin* = 1. The embedding process based on *dmin* = 0 or *dmin* = 1 is similar to Lu's method (Lu et al., 2018). The embedding process based on *dmin*= -2 or *dmin* = 3 is conducted based on equation (2) – (5).

In the embedding process based on *dmin* = 0 or *dmin* = 1, the message is embedded in the pixel in the block that has the smallest value. In this case, if *dmin* equals 0 or 1, as well as for *dmax* if it equals 0 or 1, then the message is embedded in the pixel which has the greatest value. For the embedding process based on *dmin* = -2 or *dmin* = 3, if *dmin* equals -2 or 3 then the message is embedded in the pixel in the block that has the smallest value, as well as for *dmax* if it equals -2 or 3 then the message is embedded in the pixel in the block that has the largest value.

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/

The detail of the embedding process based on *dmin* = 0 or *dmin* = 1 is shown in equation (3.11) – (3.14). In this case, if the number of bit 0 in the secret message is greater than the number of bit 1, then equations (2) and (3) are used, while if the number of bit 1 in the secret message is greater than the number of bit 0, then equation (4) and (5) are used.

- Embedding (0,1) – mode is an embedding method that embeds a secret message (in bits) into the smallest or largest pixel that has a difference (*dmin* or *dmax*) of 0 or 1 with the adjacent pixel in a block.

  If k = 0, then the embedding result is as follows:

$$x'_{\Delta(1)} = \begin{cases} x_{\Delta(1)}, & if\ (dmin = 0\ or\ 1)\ and\ message = 0 \\ x_{\Delta(1)} + 1, & if\ (dmin = 0\ or\ 1)\ and\ message = 1 \end{cases} \quad (2)$$

$$x'_{\Delta(4)} = \begin{cases} x_{\Delta(4)}, & if\ (dmax = 0\ or\ 1)\ and\ message = 0 \\ x_{\Delta(4)} + 1, & if\ (dmax = 0\ or\ 1)\ and\ message = 1 \end{cases} \quad (3)$$

  If k = 1, then then the embedding result is as follow:

$$x'_{\Delta(1)} = \begin{cases} x_{\Delta(1)} + 1, & if\ (dmin = 0\ or\ 1)\ and\ message = 0 \\ x_{\Delta(1)}, & if\ (dmin = 0\ or\ 1)\ and\ message = 1 \end{cases} \quad (4)$$

$$x'_{\Delta(4)} = \begin{cases} x_{\Delta(4)} + 1, & if\ (dmax = 0\ or\ 1)\ and\ message = 0 \\ x_{\Delta(4)}, & if\ (dmax = 0\ or\ 1)\ and\ message = 1 \end{cases} \quad (5)$$

The detail of embedding process based on *dmin* = -2 or *dmin* = 3 is shown in equation (6) - (9). In this case, if the number of bit 0 in the secret message is greater than the number of bit 1, then equations (6) and (7) are used, while if the number of bit 1 in the secret message is greater than the number of bit 0, then equation (8) and (9) are used.

- Embedding (-2,3) - mode is an embedding method that embeds a secret message (in bits) into the smallest or largest pixel that has a difference (*dmin* or *dmax*) of -2 or 3 with the adjacent pixel in a block

  If k = 0, then the embedding result is as follows:

$$x'_{\Delta(1)} = \begin{cases} x_{\Delta(1)}, & if\ (dmin = -2\ or\ 3)\ and\ message = 0 \\ x_{\Delta(1)} + 1, & if\ (dmin = -2\ or\ 3)\ and\ message = 1 \end{cases} \quad (6)$$

$$x'_{\Delta(4)} = \begin{cases} x_{\Delta(4)}, & if\ (dmax = -2\ or\ 3)\ and\ message = 0 \\ x_{\Delta(4)} + 1, & if\ (dmax = -2\ or\ 3)\ and\ message = 1 \end{cases} \quad (7)$$

  If k = 1, then then the embedding result is as follow:

$$x'_{\Delta(1)} = \begin{cases} x_{\Delta(1)} + 1, & if\ (dmin = -2\ or\ 3)\ and\ message = 0 \\ x_{\Delta(1)}, & if\ (dmin = -2\ or\ 3)\ and\ message = 1 \end{cases} \quad (8)$$

$$x'_{\Delta(4)} = \begin{cases} x_{\Delta(4)} + 1, & if\ (dmax = -2\ or\ 3)\ and\ message = 0 \\ x_{\Delta(4)}, & if\ (dmax = -2\ or\ 3)\ and\ message = 1 \end{cases} \quad (9)$$

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/

After all messages are embedded, each pixel in the block that implements the embedding process based on *dmin* = 0 or *dmin* = 1 that have the *dmin* value is outside the range of 0 and 1, and the difference between the $dmin_i$ and $dmin_{i+1}$ is greater than or equal to 2, then the $dmin_i$ is the limit of the shifted pixel. The shifted pixel is the smallest pixel of a block that is shifted 1 bit to the left. For the embedding process based on *dmax* = 0 and *dmax* = 1 that have the *dmax* outside the range between 0 and 1 and the difference between the $dmax_i$ and $dmax_{i+1}$ is greater than or equal to 2, then the $dmax_i$ is the limit of the shifted pixel. The shifted pixel is the largest pixel of a block shifted that is 1 bit to the right. The detail of this process is shown in equations (10) – (11).

For the embedding process, based on *dmin* = -2 or *dmin* = 3 that have the *dmin* value is outside the range between -3 and 4 and the difference between the $dmin_i$ and $dmin_{i+1}$ is greater than or equal to 4, then the $dmin_i$ is the limit of the shifted pixel. The shifted pixel is the smallest pixel of a block that is shifted 3 bits to the left. For the embedding process based on *dmax* = -2 and *dmax* = 3 that have the *dmax* outside the range between -3 and 4 and the difference between the $dmax_i$ and $dmax_{i+1}$ is greater than or equal to 4, then the $dmax_i$ is the limit of the shifted pixel. The shifted pixel is the largest pixel of a block that is shifted 3 bits to the right. The detail of this process is shown in equations (16) – (17).

- Pixel shifting for embedding (0,1) – mode

$$x'_{\Delta(1)} = \begin{cases} x_{\Delta(1)}, & if\ (dmin_i > 2\ \ or\ dmin_i < -1)\ and\ (|dmin_i - dmin_{i+1}| \geq 2) \\ x_{\Delta(1)} - 1, & if\ (dmin_i \geq 2\ \ or\ dmin_i \leq -1)\ and\ (|dmin_i - dmin_{i+1}| < 2) \end{cases} \quad (10)$$

$$x'_{\Delta(4)} = \begin{cases} x_{\Delta(4)}, & if\ (dmax_i > 2\ \ or\ dmax_i < -1)\ and\ (|dmax_i - dmax_{i+1}| \geq 2) \\ x_{\Delta(4)} + 1, & if\ (dmax_i \geq 2\ \ or\ dmax_i \leq -1)\ and\ (|dmax_i - dmax_{i+1}| < 2) \end{cases} \quad (11)$$

The limit is the pixel limit that is not shifted which can be seen from the difference value. To obtained the limit, using equations (12) – (15) and (18) – (21).

- Shifting limit of *dmin* (0,1)
  - For $dmin_i \leq -1\ and\ \Delta \geq 2$

$$limit_{(dmin_i \leq -1;\ \Delta \geq 2)} = dmin_{i+1} \quad ; \quad where\ dmin_{i+1} < -1\ and\ \Delta \geq 2 \quad (12)$$

  - For $dmin_i \geq 2\ and\ \Delta \geq 2$

$$limit_{(dmin_i \geq 2;\Delta \geq 2)} = dmin_{i+1} \quad ; \quad where\ dmin_{i+1} > 2\ and\ \Delta \geq 2 \quad (13)$$

- Shifting limit of *dmax* (0,1)
  - For $dmax_i \leq -1\ and\ \Delta \geq 2$

$$limit_{(dmax_i \leq -1;\Delta \geq 2)} = dmax_{i+1} \ ; \ where\ dmax_{i+1} < -1\ and\ \Delta \geq 2 \quad (14)$$

  - For $dmax_i \geq 2\ and\ \Delta \geq 2$

$$limit_{(dmax_i \geq 2;\Delta \geq 2)} = dmax_{i+1} \quad ; \quad where\ dmax_{i+1} > 2\ and\ \Delta \geq 2 \quad (15)$$

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/

- Pixel shifting for embedding (-2,3) – mode

$$x'_{\Delta(1)} = \begin{cases} x_{\Delta(1)}, & if\ (dmin_i > 4\ \ or\ dmin_i < -3)\ and\ (|dmin_i - dmin_{i-1}| \geq 4) \\ x_{\Delta(1)} - 1, & if\ (dmin_i \geq 4\ \ or\ dmin_i \leq -3)\ and\ (|dmin_i - dmin_{i+1}| < 4) \end{cases} \quad (16)$$

$$x'_{\Delta(4)} = \begin{cases} x_{\Delta(4)}, & if\ (dmax_i > 4\ \ or\ dmax_i < -3)\ and\ (|dmax_i - dmax_{i+1}| \geq 4) \\ x_{\Delta(4)} + 1, & if\ (dmax_i \geq 4\ \ or\ dmax_i \leq -3)\ and\ (|dmax_i - dmax_{i+1}| < 4) \end{cases} \quad (17)$$

- Shifting limit of *dmin* (-2,3)
  - For $dmin_i \leq -3\ and\ \Delta \geq 4$

$$limit_{(dmin_i \leq -3;\Delta \geq 4)} = dmin_{i+1} \qquad where\ dmin_{i+1} < -3\ and\ \Delta \geq 4 \quad (18)$$

  - For $dmin_i \geq 4\ and\ \Delta \geq 4$

$$limit_{(dmin_i \geq 4;\Delta \geq 4)} = dmin_{i+1} \qquad where\ dmin_{i+1} > 4\ and\ \Delta \geq 4 \quad (19)$$

- Shifting limit of *dmax* (-2,3)
  - For $dmax_i \leq -3\ and\ \Delta \geq 4$

$$limit_{(dmax_i \leq -3;\Delta \geq 4)} = dmax_{i+1} \qquad where\ dmax_{i+1} < -3\ and\ \Delta \geq 4 \quad (20)$$

  - For $dmax_i \geq 4\ and\ \Delta \geq 4$

$$limit_{(dmax_i \geq 4;\Delta \geq 4)} = dmax_{i+1} \qquad where\ dmax_{i+1} > 4\ and\ \Delta \geq 4 \quad (21)$$

After all messages are embedded and generate a stego image, retrieve the original LSB pixels from the pixels that will be used for embedding compressed additional messages and location map (location map is an indicator that indicates whether the block has an overflow (or underflow) problem or not (see section Pre-processing of Cover Image)). This is process has to be conducted to produce a sequence of original LSB pixel values in binary after the extraction process is carried out. The sequence of original LSB pixel values is stored in the metadata. Then, the embedding process of compressed additional messages and location map into the LSB pixels of the image is conducted and started at the first of the pixel the image. The information that will be embedded using the LSB method includes the following parameters:

- Compressed Location Map (LMc)
- Information of arithmetic coding (symbols, counts, length of LM)
- $Limit_i$ of the shifted pixel, which can be calculated by equation ((12) – (15) and (18) – (21)). Note: i is the index value of 1 to 8
- Position of the last data-carrying block (kend)
- Mark of the most message bit (k)

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/

Example of modified PVO embedding is shown in Figure 2.

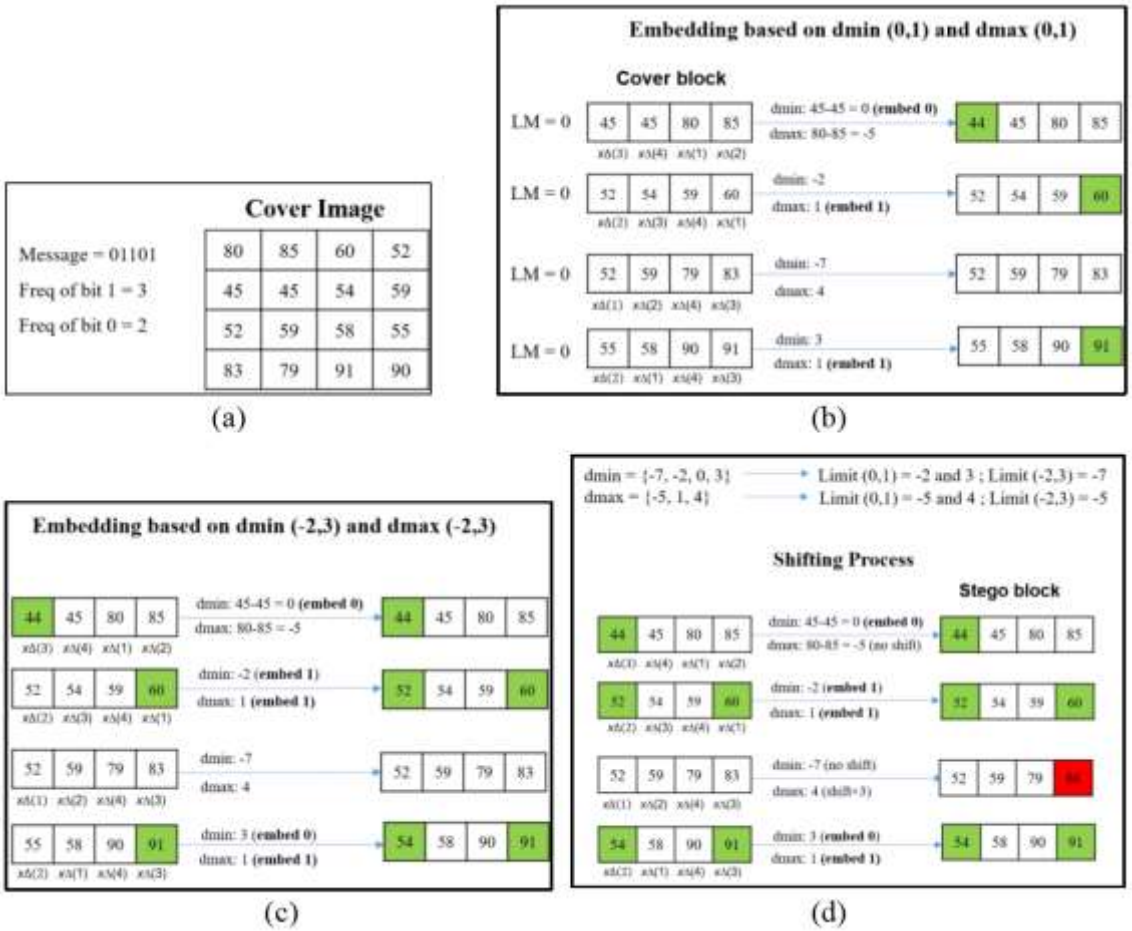


Figure 2. (a) Cover image and secret message; (b) Embedding process based on dmin = 0 or 1 and dmax = 0 or 1; (c) Embedding process based on dmin = -2 and 3, and dmax = -2 and 3; (d) Shifting process

e. Extraction and Recovery

To obtain the secret message, the receiver should perform the extraction process. This process began by extracting the stego image. After the secret message is obtained, the next step is to recover the original image process. The overview of the extraction and recovery process in the proposed method is shown in Figure 3.

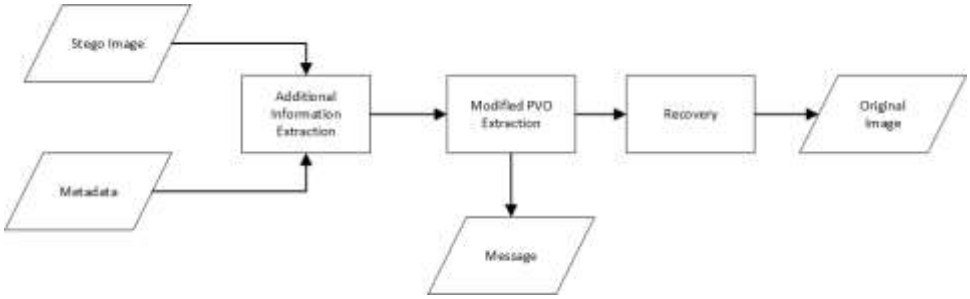


Figure 3. Extraction and Recovery Process

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/

f. Additional Information and Location Map Extraction

To obtain the compressed additional information (symbols, counts, length of LM, limit, k, kend) and location map (LM), we have to extract this information from the image using an LSB-based extraction process. For recovering the original pixels in the image after LSB- based extraction the sequence of original pixel values that have been embedded by the LSB method is extracted from the metadata. The original pixel is used to change all LSB values that have already been embedded by additional information and location map (LM).

g. Modified PVO Extraction

Similar to the embedding process, the extraction process is divided into two processes, the extraction process is based on *dmin* = 0 or *dmin* = 1, and the extraction process is based on *dmin* = -2 or *dmin* = 3.

In the extraction process, the stego image is divided into non-overlapping blocks of size 2x2. Then, the pixels in each block is sorted in ascending order. Furthermore, the LM value of the block has to be evaluated. If LM = 0 then the block can be embedded by a secret message, otherwise the block cannot be embedded. If the block can be embedded then calculate the *dmin* and *dmax* values. Then check the value of k. If k = 1 then the extraction process can be carried out using equation (22), otherwise the extraction process can be carried out using equation (23).

- Extraction of embedding (0, 1) - mode, if LM ≠ 1.

If k = 1

$$message = \begin{cases} 1 & if\ dmin'\ (dmax') = 0\ or\ 1 \\ 0 & if\ dmin'\ (dmax') = -1\ or\ 2 \\ no\ message & otherwise \end{cases} \quad (22)$$

If k = 0

$$message = \begin{cases} 0 & if\ dmin\ (dmax) = 0\ or\ 1 \\ 1 & if\ dmin\ (dmax) = -1\ or\ 2 \\ no\ message & otherwise \end{cases} \quad (23)$$

In the extraction process based on *dmin* = -2 or *dmin* = 3, all the pixels in each block is sorted in ascending order. Then, the LM value of each block is evaluated. If LM = 0 or 1 then the block can be embedded by the secret message, while if LM = 2 then the block cannot be embedded. If the block can be embedded then the *dmin* and *dmax* values have to be calculated. Then evaluate the value of k. If k = 1 then the extraction process can be carried out using equation (24), while if k = 0 then the extraction process can be carried out using equation (25).

- Extraction of embedding (-2, 3) - mode, if LM ≠ 2

If k = 1

$$message = \begin{cases} 1 & if\ dmin'(dmax') = -2\ or\ 3 \\ 0 & if\ dmin'\ (dmax') = -2\ or\ 3 \\ no\ message & otherwise \end{cases} \quad (24)$$

If k = 0

$$message = \begin{cases} 0 & if\ dmin'\ (dmax') = -2\ or\ 3 \\ 1 & if\ dmin'\ (dmax') = -2\ or\ 3 \\ no\ message & otherwise \end{cases} \quad (25)$$

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/

h. Recovery

Similar to the extraction process, the recovery process is divided into two processes, the recovery process based on *dmin* = 0 or *dmin* = 1, and the recovery process based on *dmin* = -2 or *dmin* = 3.

In the recovery process based on *dmin* = 0 or *dmin* = 1, the pixel that has *dmin* outside the range between -1 and 2, and *dmin* is smaller than the $limit_i$ (for *dmin* > 2) or *dmin* is greater than the $limit_{i+1}$ (for *dmin* < -1), it is shifted 1 bit to the right. If *dmin* = -1 or *dmin* = 2 and k=0, the pixel value is shifted 1 bit to the right, and if *dmin* = 0 or *dmin* = 1 and k = 1, it is shifted 1 bit to right, otherwise the pixels are not shifted. The detail of this process is shown in equation (26).

In the recovery process based on *dmax* = 0 or *dmax* = 1, the pixel that has *dmax* outside the range between -1 and 2, and *dmax* is smaller than the $limit_{i+2}$ (for *dmax* > 2) or *dmax* is greater than the $limit_{i+3}$ (for *dmax* < -1), it is shifted 1 bit to the left. If *dmax* = -1 or *dmax* = 2 and k=0, pixel value is shifted 1 bit to the left, and if *dmax* = 0 or *dmax* = 1 and k =1, it is shifted 1 bit to left, otherwise the pixels are not shifted. The detail of this process is shown in equation (27).

- Recovering for (*dmin* = 0 or *dmin* = 1) and (*dmax* = 0 or *dmax* = 1)

$$x_{\Delta(1)} = \begin{cases} x'_{\Delta(1)} + 1, & if\ (dmin < -1\ and\ dmin > limit_i)\ or\ (dmin > 2\ and\ dmin < limit_{i+1}) \\ x'_{\Delta(1)} + 1, & if\ (dmin = -1\ or\ 2)\ and\ k = 0 \\ x'_{\Delta(1)} + 1, & if\ (dmin = 0\ or\ 1)\ and\ k = 1 \\ x'_{\Delta(1)}, & otherwise \end{cases} \quad (26)$$

$$x_{\Delta(4)} = \begin{cases} x'_{\Delta(4)} - 1, if\ (dmax < -1\ and\ dmax > limit_{i+2})\ or\ (dmax < -1\ and\ dmax > limit_{i+3}) \\ x'_{\Delta(4)} - 1, & if\ (dmax = -1\ or\ 2) and\ k = 0 \\ x'_{\Delta(4)} - 1, & if\ (dmax = 0\ or\ 1) and\ k = 1 \\ x'_{\Delta(4)}, & otherwise \end{cases} \quad (27)$$

In the recovery process based on *dmin* = -2 or *dmin* = 3, the pixel that have *dmin* outside the range between -3 and 4 and (*dmin* < $limit_{i+4}$ and *dmin* > 4) or (*dmin* > $limit_{i+5}$ and *dmin* < -3), is shifted 3 bit to the right. If (*dmin* = -3 or *dmin* = 4 and k = 0), the pixel with smallest value is shifted 1 bit to the right, and if (*dmin* = -2 or *dmin* = 3 and k = 1), the pixel with smallest value is shifted 1 bit to right, otherwise the pixels are not shifted. The detail of this process is shown in equation (28).

For the recovery process based on *dmax* = -2 or *dmax* = 3, the pixel that have *dmax* outside the range between -3 and 4 and (*dmax* < $limit_{i+6}$ and *dmax* > 4) or (*dmax* > $limit_{i+7}$ and *dmax* < -3), is shifted 3 bit to the left. If (*dmax* = -3 or *dmax* = 4 and k = 0), the pixel with largest value is shifted 1 bit to the left, and if *dmin* = 2 or *dmin* = 3 and k = 1 it is shifted 1 bit to left, otherwise the pixels are not shifted. The detail of this process is shown in equation (29).

- Recovering for (*dmin* = -2 or *dmin* = 3) and (*dmax* = -2 or *dmax* = 3)

$$x_{\Delta(1)} = \begin{cases} x'_{\Delta(1)} + 3, if\ (dmin < -3\ and\ dmin > limit_{i+4})\ or\ (dmin > 4\ and\ dmin < limit_{i+5}) \\ x'_{\Delta(1)} + 1, & if\ (dmin = -3\ or\ 4)\ and\ k = 0 \\ x'_{\Delta(1)} + 1, & if\ (dmin = -2\ or\ 3)\ and\ k = 1 \\ x'_{\Delta(1)}, & otherwise \end{cases} \quad (28)$$

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/

$$x_{\Delta(4)} = \begin{cases} x'_{\Delta(4)} - 3, if\ (dmax < 3\ and\ dmax > limit_{i+6})\ or\ (dmax > 4\ and\ dmax > limit_{i+7}) \\ x'_{\Delta(4)} - 1, \quad if\ (dmax = -3\ or\ 4)\ and\ k = 0 \\ x'_{\Delta(4)} - 1, \quad if\ (dmax = -2\ or\ 3)\ and\ k = 1 \\ x'_{\Delta(4)}, \quad otherwise \end{cases} \tag{29}$$

An example of modified PVO extraction and recovery is shown in Figure 4.

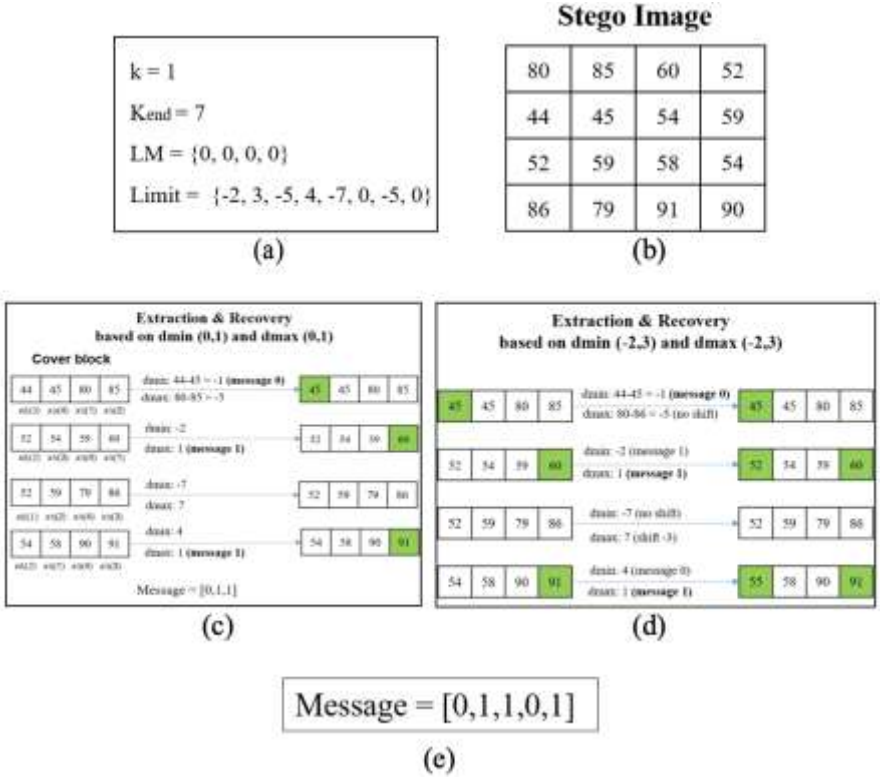


Figure 4. (a) Additional information, and LM; (b) Stego image; (c) Extraction and recovery process based on dmin = 0 and 1, and dmax = 0 and 1; (d) Extraction and recovery processbased on dmin = -2 and 3, and dmax = -2 and 3; (e) The secret message

Results and Discussion

This section discusses the analysis of experimental results that have been conducted for evaluating the performance of the proposed method compared with Lu's method (Lu et al., 2018). The performance evaluation is done based on the capacity, imperceptibility, robustness, and steganalysis.

a. Analysis of Maximum Embedding Capacity

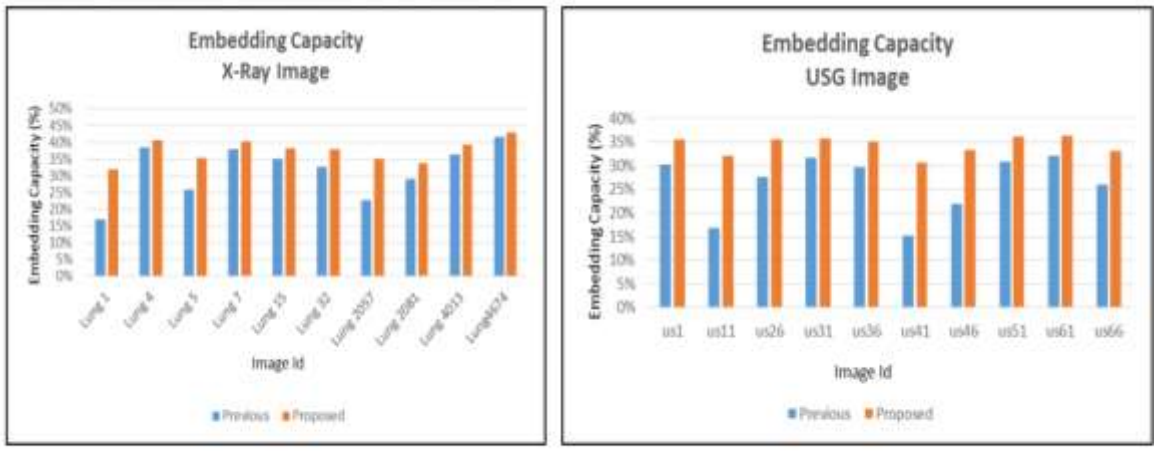
For evaluating the embedding capacity, the experiment uses secret messages which have a maximum size of the secret message embedded into the cover image of size 512 x 512 using both the previous and the proposed methods.

According to the experiment results, as shown in (Figure 5 (a) – 5 (b)), the embedding capacity of each cover image using the proposed method is greater than using Lu's method (Lu et al., 2018). This condition has occurred because for embedding a secret message, the proposed method uses the expansion of the difference value. The difference values of -2, 0, 1, 3 are used

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/

in the proposed method, while the previous method only uses the difference value of 0, 1 to embed a message. It is clear that the expansion of the difference values increases the embedding capacity, such that the embedding capacity of the cover image using the proposed method is higher than using Lu's method (Lu et al., 2018).



(a) (b)

Figure 5. (a) Embedding capacity of x-ray image; (b) Embedding capacity of usg image

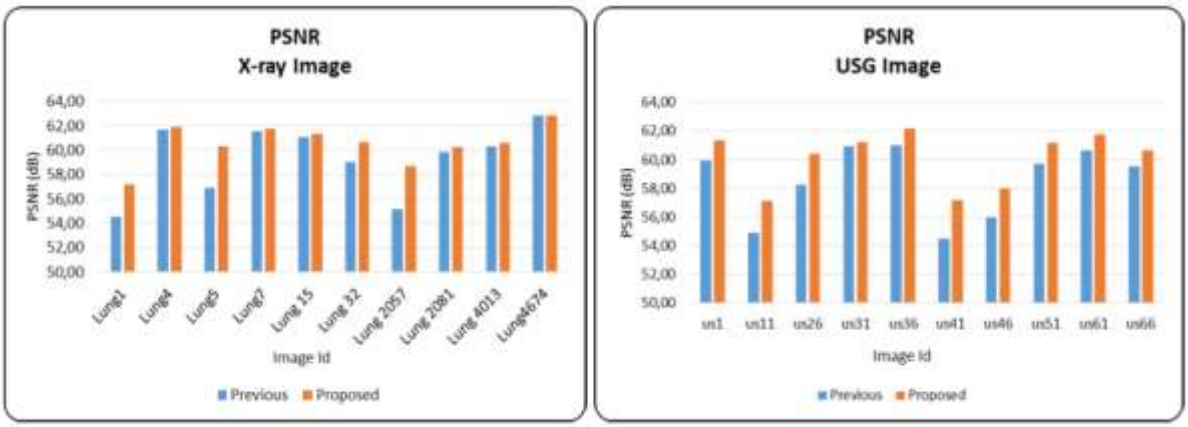
The increasing of the embedding capacity has also occurred because the proposed method only embeds a secret message into the pixels, while in Lu's method instead of a secret message, other additional information is also embedded into the cover image.
Based on Figure 5 (a), the maximum embedding capacity of the proposed method is 43% for lung4674 (x-ray image), while Lu's method has a maximum embedding capacity is 42% for lung4674 (x-ray image). Based on Figure 5 (b), the maximum embedding capacity of the proposed method is 36% for us61 (usg image), while Lu's method has a maximum embedding capacity is 32% for us61 (usg image). It can be shown that the proposed method can increase the embedding capacity.

b. Analysis of Imperceptibility

Imperceptibility testing is carried out to see the level of changes in the cover image that occurs before and after the message is embedded. To calculate the imperceptibility, MSE and PSNR are used. PSNR measures the quality of the stego-image by comparing the mean squared error (MSE) between the cover and the stego image. The higher the PSNR, the better the image quality. To maintain image quality when using the proposed method the bit frequency of 0 and 1 in the message has to be observed. Bits with a larger frequency of occurrence is kept unchanged during the embedding, and vice versa.

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/



(a) (b)

Figure 6. (a) PSNR comparison for x-ray image; (b) PSNR comparison for usg image

According to the experiment results (Figure 6 (a) – 6 (b)), it can be concluded that for each cover image with the same message size, the PSNR of the stego image is resulted using the proposed method achieves ≥ 57 dB while using Lu's method ≥ 54 dB. In general, the PSNR value for the x-ray image using the proposed method is 1.3 dB greater than using Lu's method, while the PSNR using the proposed method for the usg image is greater by 1.6 dB.

c. Analysis of Robustness

Robustness evaluation was carried out to evaluate the ability to extract secret messages even though the stego-image was corrupted by noise, cropping, and scratching.

- **Analysis of Robustness against Noise Attacks**

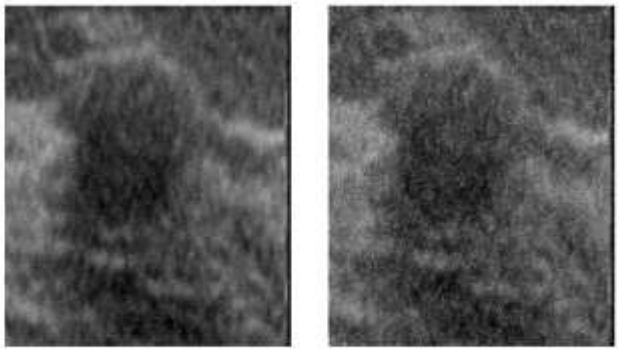
  For noise attacks, salt and pepper are used. In the experiment results of robustness against noise attacks salt and pepper, the evaluation is done based on noise rates of 5% and 15%. The result of the robustness against salt and pepper attack is shown in (Figure 7 (b) - 7(c)).

  Based on (Figure 7 (b) – 7 (c)), the proposed method outperforms Lu's method and successfully recovers the secret image from stego images under Salt and Pepper noise rate of 5%. This occurred because the smaller the noise attacked the images, the greater the message that can be recovered.

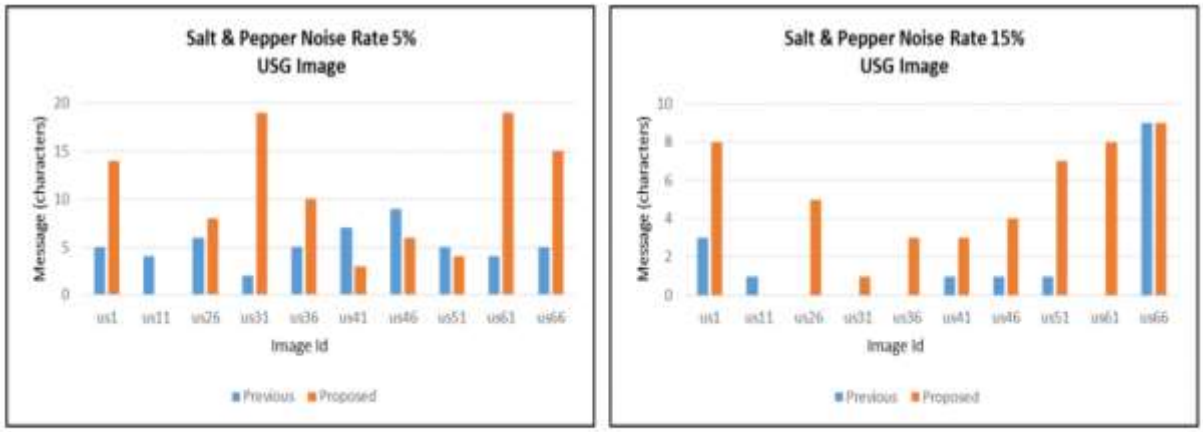
  Figure 7 (c) shows that using noise rate of 15%, the proposed method recovered the secret messages more than the previous method besides in image us11. This condition occurs because in us11 image produce a lot of pixels that have value of 0 and 255, which means that some blocks cannot be embedded, even though in actual conditions these blocks can be embedded. Thus, there will be a loss of some messages because some blocks are considered not embedded which are then not analyzed.

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/



(a)



(b) (c)

Figure 7. (a) Example of attacked images by noise rate of 5% and 15%; (b) Noise rate 5%; (c) Noise rate 15%.

- **Analysis of Robustness against Cropping Attacks**

In the experiment for evaluating the robustness against cropping, 10 different crop positions are used. Cropping was performed in specific regions of each image with size 144 x 144 pixels. The result robustness against cropping attack is shown in (Figure 8 (b) – 8 (c)).

Based on (Figure 8 (b) – 8 (c)), both the proposed and Lu's methods can successfully recover all secret messages perfectly in area 4 – area 10. Especially in areas 1, 2, and 3, both methods can only receive less messages than the other areas, because additional information needed to extract messages is embedded in this area. However, the results of the proposed method outperform Lu's method.

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/

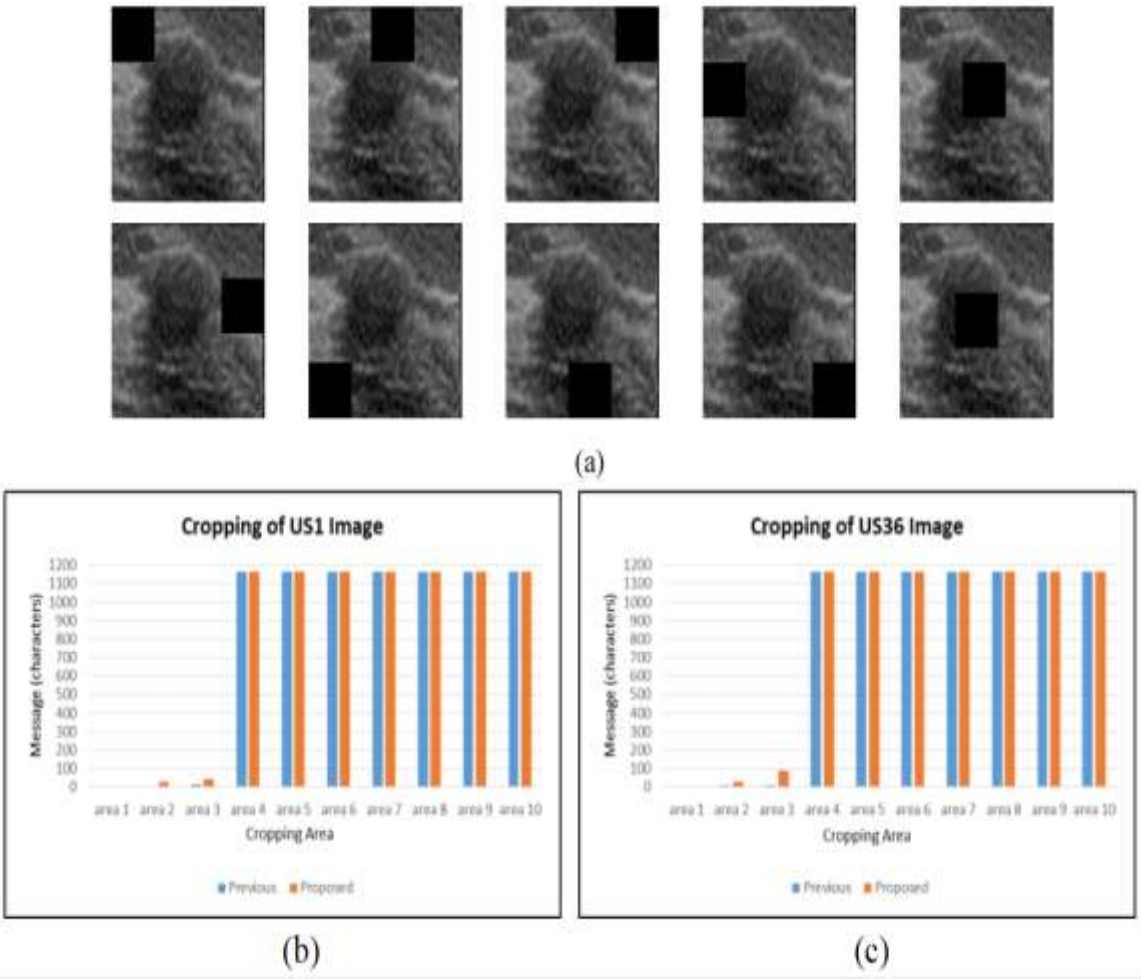


Figure 8.(a) Example of images that are attacked by cropping; (b) Cropping of usg-1 image (c) Cropping of usg-36 image

- **Analysis of Robustness against Scratching Attacks**

In the experiment for evaluating the robustness against scratching attacks, three types of scratches are used, single line, multiple lines, and zigzag. The result robustness against scratching attack is shown in (Figure 9 (b) – 9 (d)).

Based on Figure 9 (b), the stego image that uses the proposed methods can successfully recover all secret messages perfectly after scratch attacks are applied to the stego image, while when using the previous method some secret messages cannot be recovered such as occurred in images us11, us41, and us46. This condition occurs because the scratch changes the pixels in the stego image which is embedded by additional information needed to extract messages such that the information cannot be extracted properly.

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/

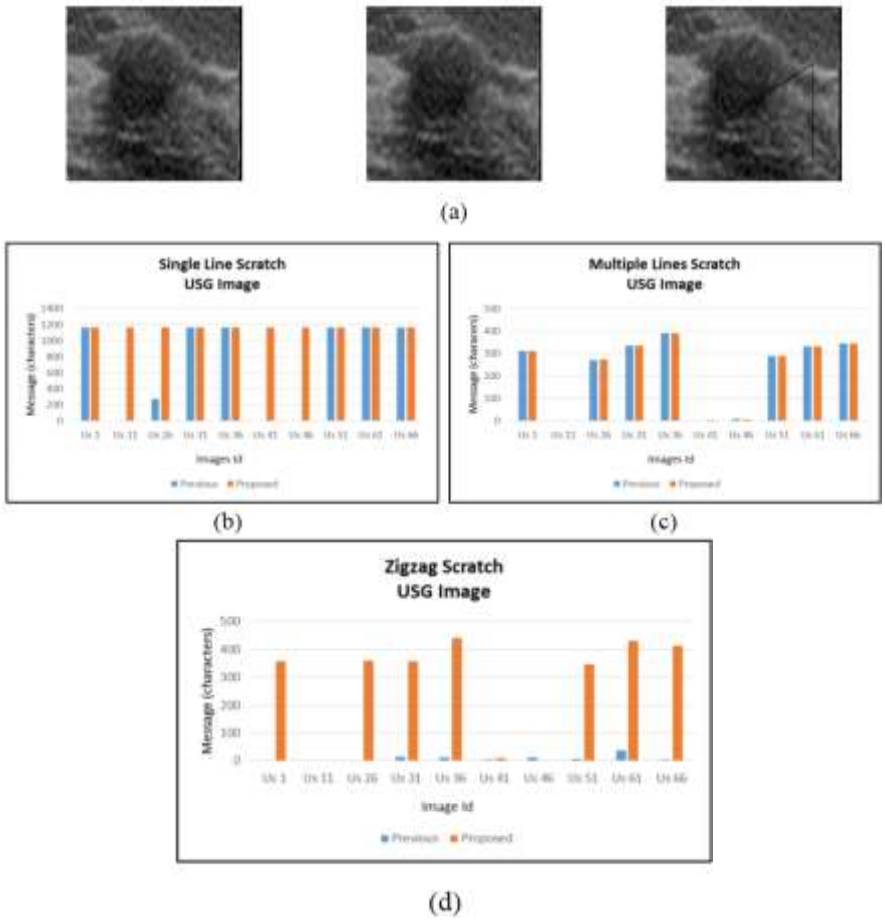


Figure 9. (a) Example of images that are attacked by scratch; (b) Single line scratch (c) Multiple lines scratch (d) Zigzag scratch

Figure 9 (c) – 9 (d), shows that the scratch attacks affect both the proposed and Lu's method, especially in images us11, us41, and us46. This condition occurs because the images produce a lot of pixels that have a value of 0 and 255, which means that some blocks cannot be embedded, even though in actual conditions these blocks can be embedded. Thus, there will be a loss of some messages because some blocks are considered not embedded which are then not analyzed.

d. Steganalysis

Triples steganalysis has been implemented to both the proposed and Lu's method. To evaluate the performance of Lu's method and the proposed method, the True p-value (Tp) and the estimated p-value (Ep) of each cover image is used and further the probability of successful prediction of the embedding message (P) is calculated using equation (30).

$$P = \frac{E_p}{T_p} \tag{30}$$

A larger P-value means that the estimated p-value is close to the true p value, which means that the attacker's probability of predicting the true p-value is high. In this case, Triples

Review: Journal of Multidisciplinary in Social Sciences

Volume 01 No 05 April 2024
E ISSN : 3031-6375
https://lenteranusa.id/

estimates the p-value is almost accurate. The smaller P-value the larger difference between the estimated p-value and the true p-value, such that an attacker cannot easily predict the embedded secret message (Ker, 2006).

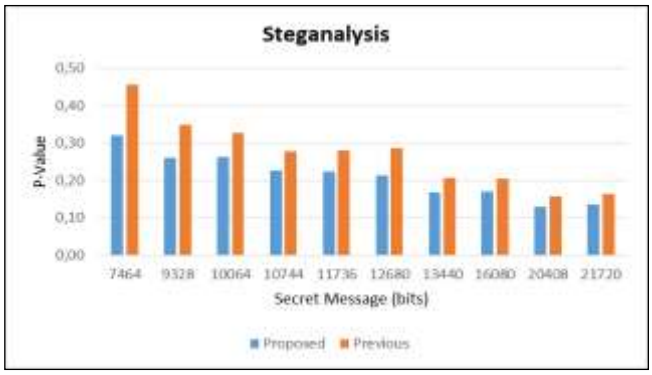


Figure 10. The result of Triples Steganalysis

Based on Figure 10, the p-value of the proposed method is less than Lu's method, which means that the probability of an attacker to predict the presence of the secret message when using Lu's method is higher than when using the proposed method. The larger the message size, the lower the p-value, because the triples method is very sensitive to the small size of the secret message than the large ones. Hence, the proposed method performs better than Lu's method in hiding the secret message.

Conclusion

To overcome the problem of embedding capacity in Lu's method (Lu et al., 2018), this study proposes a modified PVO method. The difference value is expanded in the embedding process. In this case, the difference values of -2, 0, 1, and 3 can be used to embed the secret message. The proposed method also uses bit frequencies of 0 and 1 in the message to determine which pixel values will change during the embedding process, and proposes the Limit of the pixels. to shift. This is done to reduce the number of pixels that will shift such that the image quality is maintained.

Based on the experimental results, the maximum embedding capacity of the proposed method is higher than Lu's method with an average increment of 6% on x-ray images and 8% on usg images. In addition, the average PSNR value of the proposed method for the same message and the same image size is 60.57 dB on x-ray image and 60.10 dB on usg image, while the average PSNR value of Lu's method is 59.29 dB on x-ray image and 58.53 dB on usg image. The higher the PSNR, the higher the imperceptibility of the stego-image.

Based on the analysis results, it is shown that the proposed method can increase the embedding capacity while maintaining image quality. In addition, the proposed method has a better performance against salt and pepper attacks, cropping, scratching, and triples analysis.

References

Bali, A., Bali, D., Iyer, N., & Iyer, M. (2011). Management of Medical Records: Facts and Figures for Surgeons. *Journal of Maxillofacial and Oral Surgery*, *10*(3), 199–202. https://doi.org/10.1007/s12663-011-0219-8

Coding, A. (2008). *Arithmetic Coding*. 123–142. https://doi.org/10.1007/978-1-84800-072-8_4

Hong, W., Chen, T. S., & Shiu, C. W. (2009). Reversible data hiding for high quality images using modification of prediction errors. *Journal of Systems and Software*, *82*(11), 1833–1842. https://doi.org/10.1016/j.jss.2009.05.051

Ker, A. D. (2006). A general framework for structural steganalysis of LSB replacement. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3727 LNCS*, 296–311. https://doi.org/10.1007/11558859_22

Li, X., Li, J., Li, B., & Yang, B. (2013). High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. *Signal Processing*, *93*(1), 198–205. https://doi.org/10.1016/j.sigpro.2012.07.025

Lu, T. C., Tseng, C. Y., Huang, S. W., & Vo., T. N. (2018). Pixel-value-ordering based reversible information hiding scheme with self-adaptive threshold strategy. *Symmetry*, *10*(12). https://doi.org/10.3390/sym10120764

Lu, T. C., Tseng, C. Y., & Wu, J. H. (2016). Asymmetric-histogram based reversible information hiding scheme using edge sensitivity detection. *Journal of Systems and Software*, *116*, 2–21. https://doi.org/10.1016/j.jss.2015.04.085

Peng, F., Li, X., & Yang, B. (2014). Improved PVO-based reversible data hiding. *Digital Signal Processing: A Review Journal*, *25*(1), 255–265. https://doi.org/10.1016/j.dsp.2013.11.002

Shi, Y. Q., Li, X., Zhang, X., Wu, H. T., & Ma, B. (2016). Reversible data hiding: Advances in the past two decades. *IEEE Access*, *4*, 3210–3237. https://doi.org/10.1109/ACCESS.2016.2573308

Thodi, D. M., & Rodríguez, J. J. (2007). Expansion embedding techniques for reversible watermarking. *IEEE Transactions on Image Processing*, *16*(3), 721–730. https://doi.org/10.1109/TIP.2006.891046

Tian, J. (2003). Reversible Data Embedding Using a Difference Expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, *13*(8), 890–896. https://doi.org/10.1109/TCSVT.2003.815962

Upendra Raju, K., & Amutha Prabha, N. (2018). A review of Reversible Data Hiding technique based on steganography. *ARPN Journal of Engineering and Applied Sciences*, *13*(3), 1105–1114.

Wang, X., Ding, J., & Pei, Q. (2015). A novel reversible image data hiding scheme based on pixel value ordering and dynamic pixel block partition. *Information Sciences*, *310*, 16–35. https://doi.org/10.1016/j.ins.2015.03.022