Steganography in RGB Images Using Adjacent Mean

Yun-Hsin Chuang 1, Bor-Shing Lin 2, Yan-Xiang Chen 3 and Hung-Jr Shiu 3,*

1Department of Electrical Engineering, National Taiwan University, Taipei 10617 Taiwan
2Department of Computer Science and Information Engineering, National Taipei University, Sanxia, New Taipei City 237303 Taiwan
3Department of Computer Science, Tunghai University, Xitun, Taichung 407224 Taiwan

Corresponding author: Hung-Jr Shiu (e-mail: hjshiu@thu.edu.tw).

This research was partly supported by the Ministry of Science and Technology in Taiwan, under grants MOST 109-2221-E-305-001-MY2 and MOST 110-2314-B-305-001. This research was also partly supported by the University System of Taipei Joint Research Program, under grant USTP-NTPU-TMU-109-03 and USTP-NTPU-NTOU-110-01, Faculty Group Research Funding Sponsorship by National Taipei University, under grant 2021-NTPU-ORDA-02, and the “Academic Top-Notch and Features Field Project” Funding Sponsorship of National Taipei University, Taiwan, under grant 109-NTPU_ORDA-F-005.

ABSTRACT Steganography is the practice of hiding information or data in a seemingly innocuous cover medium, such as message, file, image, audio, and video. In the past decades, many approaches of steganography in images were proposed for various applications. In social communication and the information highly exposed society, steganography requires high embedding capacity to transmit secret data efficiently. Generally, there is a trade-off between fidelity and embedding capacity. In this paper, we propose a novel and efficient data hiding algorithm in 24-bit color images with super high embedding capacity and acceptable peak signal-to-noise ratio (PSNR) using spatial-domain-adjacent mean. In the proposed algorithm, the embedding rate is about 7.4 bits per pixel (bpp) when the PSNR is nearly 30, and the embedding rate is about 8.88 bpp when the PSNR is nearly 25. The advantage of the proposed method is no need to transform data in another domain and without training data. Experiments also demonstrate the imperceptibility under some state-of-art steganalysis. The proposed steganography provides an efficient way to transmit sensitive information in the information highly exposed society.

INDEX TERMS Steganography, RGB images, peak signal-to-noise ratio, data hiding; bits per pixel

I. INTRODUCTION

Steganography is the technique of concealing secret data within a non-secret, ordinary, message or file in order to avoid detection. In an approach of data hiding in images, the sender hides the embedding data into a cover image to derive the stego image and sends the stego image to the receiver, and then the receiver extracts the embedding data from the stego image. The embedding data can be an encrypted file, message, image, audio, or video, which is encrypted by the sender and can be decrypted by the receiver. The fidelity of a data hiding scheme for images is usually measured by peak signal-to-noise ratio (PSNR) between the cover image and the stego image, and the embedding capability is usually measured by the payload and the embedding rate. Generally speaking, there is a trade-off between fidelity and embedding capability. In the past decades, various approaches of data hiding in images were proposed for various applications.

For general applications, the data hiding scheme requires the balance of fidelity and embedding capability. In 2013, Juneja and Sandhu [1] proposed an improved least-significant-bit (LSB) based steganography technique for RGB color images, and Hemalatha et al. [2] proposed an image steganography technique. In 2014, Tan et al. [3] proposed image steganography using multi-layer embedding, Lin [4] proposed a data hiding scheme based upon discrete cosine transform (DCT) coefficient modification, and Jung and Yoo [5] proposed a data hiding scheme using edge detector for scalable images. In 2015, Shen et al. [6] proposed a data hiding for color images based on pixel value difference and modulus function, and Hamad and Khalifa [7] proposed non-blind data hiding for RGB images using discrete cosine transform DCT-based fusion and H.264 compression concepts. In 2016, Xu et al. [8] proposed an improved LSB substitution method using the modulo three strategy, and Nilizadeh and Nilechi [9] proposed a steganography method based on matrix pattern and LSB algorithms in RGB images. In 2017, Muhammad et al. [10] proposed color image steganography using stego key-
directed adaptive LSB substitution method, and Setiadi et al. [11] proposed an image steganography algorithm based on DCT with one-time-pad (OTP) encryption. In 2018, Farhan and Alwan [12] proposed an improved method using a two exclusive-or to binary image in RGB color image steganography. In 2019, Tyagi [13] proposed steganography protected using Shamir’s threshold scheme and permutation framework.

For medial and military applications, the data hiding scheme requires high fidelity of the stego image and the reversible cover image, which can recover the original cover image without any distortion from the stego image after the hidden embedding data have been extracted. Many reversible data hiding algorithms for images were proposed [14-21]. However, most of the reversible data hiding schemes have extremely low embedding capability. In this case, a lot of reversible data hiding algorithms [22-33] were proposed for encrypted images to increase the embedding capability; meanwhile, keeps high fidelity. However, embedding secret data in a meaningless image deviate from the essence of steganography. Transmitting a non-ordinary image may attract the notice. A clear overview and classification of Steganography was proposed in [34] and two types of methodology were listed as spatial and transform domain. The authors proposed a hybrid Steganography using pixel value difference and modulus function [35]. Another research works [36] proposed addition and subtraction logics on LSB planes. Also, the authors used LSB matching and pixel difference [37]. In [38], the authors improved [35] to the optimal version of that kind of methodology. Related work in [39-41] are proposed to use transform domain to achieve hiding data with integer wavelet transform. Transform domain based methods need more time-consuming to processing data. Neural networking based methods are proposed recently [42-47], authors used long short-term memory in [42]; Deep learning is adopted in [43]; Generative Adversarial Nets (GAN) is used in [44, 46]; Author used the source in ImageNet database with deep learning [45]. Machine learning based methods need pretrained dataset to use and also might suffer the quality of dataset.

In some undemocratic sociality or organizations, the speech and expression of the members are monitored and dataset. Learning [45]. Machine learning based methods need more time-consuming to processing data. Neural networking based methods are proposed recently [42-47], authors used long short-term memory in [42]; Deep learning is adopted in [43]; Generative Adversarial Nets (GAN) is used in [44, 46]; Author used the source in ImageNet database with deep learning [45]. Machine learning based methods need pretrained dataset to use and also might suffer the quality of dataset.

In some undemocratic sociality or organizations, the speech and expression of the members are monitored and dataset. Learning [45]. Machine learning based methods need more time-consuming to processing data. Neural networking based methods are proposed recently [42-47], authors used long short-term memory in [42]; Deep learning is adopted in [43]; Generative Adversarial Nets (GAN) is used in [44, 46]; Author used the source in ImageNet database with deep learning [45]. Machine learning based methods need pretrained dataset to use and also might suffer the quality of dataset.

In some undemocratic sociality or organizations, the speech and expression of the members are monitored and dataset. Learning [45]. Machine learning based methods need more time-consuming to processing data. Neural networking based methods are proposed recently [42-47], authors used long short-term memory in [42]; Deep learning is adopted in [43]; Generative Adversarial Nets (GAN) is used in [44, 46]; Author used the source in ImageNet database with deep learning [45]. Machine learning based methods need pretrained dataset to use and also might suffer the quality of dataset.

In some undemocratic sociality or organizations, the speech and expression of the members are monitored and dataset. Learning [45]. Machine learning based methods need more time-consuming to processing data. Neural networking based methods are proposed recently [42-47], authors used long short-term memory in [42]; Deep learning is adopted in [43]; Generative Adversarial Nets (GAN) is used in [44, 46]; Author used the source in ImageNet database with deep learning [45]. Machine learning based methods need pretrained dataset to use and also might suffer the quality of dataset.
Step 4. Since the expected value of the embedding value approaches $2^{k-1}$, we pre-subtract $2^{k-1}$ before we add an embedding decimal integer to the embedding position. We add the embedding decimal integers to the embedding positions in sequence by the following algorithm.

Let $k=1$; 
for $(i=2, i<m, i++)$
for $(j=2, j<n, j++)$
if ($(i,j)$ is even and $D_k$ is not NULL) {
if ($D_k$ is NULL { let $D_{k+1}=-1;$
} if $D_{k+2}$ is NULL {
let $D_{k+2}=-1$;
} $rs_{ij}=rm_{ij}2^{k-1}+D_k$;
$gs_{ij}=gm_{ij}2^{k-1}+D_{k+1};$
$bs_{ij}=bm_{ij}2^{k-1}+D_{k+2};$ 
$s_{ij}=(rs_{ij}, gs_{ij}, bs_{ij});$
$k=k+3;$
} 

Figure 1 illustrates the embedding procedure of the proposed method. Suppose that $S$ is a stego image represented by the matrix $S_{n \times p}$, where each element $s_{ij}=(rs_{ij}, gs_{ij}, bs_{ij})$ denotes the RGB value of the pixel. Then we can extract the embedding data string $E$ from the stego image $S$ by the following steps. 

Step 1. Classify the embedding positions and non-embedding positions. If $(i,j)$ is even, $i \neq j$, and $j \neq q$, then $s_{ij}$ is a embedding position; otherwise, $s_{ij}$ is a non-embedding position. Compute the length of the embedding sub-string $N=rs_{1,1}-(rs_{1,2}+rs_{2,1})/2$ and $R=gs_{1,1}+((gs_{1,2}+gs_{2,1})/4)$. 

Step 2. For each embedding position $s_{ij}$, compute $m_{ij}=[s_{ij}+(s_{i+1,j}+s_{i+1,j}+s_{i,j}+s_{i,j})]/4 = \left(\left(\left(\left(\left(\left(s_{i+1,j}+s_{i+1,j}+s_{i,j}+s_{i,j}\right)\right)\right)\right)\right)\right)\right).$

Step 3. Extract the embedding data string $E$ by the following algorithm. 
Let $k=1; E=NULL;$
for $(i=2, i<m, i++)$
for $(j=2, j<n, j++)$
if ($(i,j)$ is even) { 
$D_k=rs_{ij}-rm_{ij}2^{k-1};$
$D_{k+1}=gs_{ij}-gm_{ij}2^{k-1};$
$D_{k+2}=bs_{ij}-bm_{ij}2^{k-1};$
if $D_k=1$
} break;
else 

TABLE 1. Cover image C.

| $c_{1,1}$ | $c_{1,2}$ | $c_{1,3}$ | $c_{1,4}$ | $c_{1,5}$ |
|----------|----------|----------|----------|----------|
| (25,128,42) | (30,110,106) | (57,85,131) | (61,128,150) | (255,255,255) |
| (80,37,200) | (63,142,98) | (50,169,62) | (72,196,110) | (94,240,42) |
| (76,37,200) | (85,193,74) | (43,137,74) | (83,255,200) | (150,37,200) |
| (105,37,200) | (90,37,200) | (86,111,95) | (102,237,156) | (200,37,200) |
| (150,37,200) | (100,37,200) | (150,200,255) | (142,198,114) | (255,255,255) |

TABLE 2. Stego image $S_{k+7}$.

| $s_{1,1}$ | $s_{1,2}$ | $s_{1,3}$ | $s_{1,4}$ | $s_{1,5}$ |
|----------|----------|----------|----------|----------|
| (25,128,42) | (30,110,106) | (57,85,131) | (61,128,150) | (255,255,255) |
| (80,37,200) | (61,128,109) | (50,169,62) | (72,196,110) | (94,240,42) |
| (76,37,200) | (85,193,74) | (75,185,108) | (83,255,200) | (150,37,200) |
| (105,37,200) | (43,97,139) | (86,111,95) | (123,145,147) | (200,37,200) |
| (150,37,200) | (100,37,200) | (150,200,255) | (142,198,114) | (255,255,255) |

TABLE 1. Cover image C.

| $c_{1,1}$ | $c_{1,2}$ | $c_{1,3}$ | $c_{1,4}$ | $c_{1,5}$ |
|----------|----------|----------|----------|----------|
| (25,128,42) | (30,110,106) | (57,85,131) | (61,128,150) | (255,255,255) |
| (80,37,200) | (63,142,98) | (50,169,62) | (72,196,110) | (94,240,42) |
| (76,37,200) | (85,193,74) | (43,137,74) | (83,255,200) | (150,37,200) |
| (105,37,200) | (90,37,200) | (86,111,95) | (102,237,156) | (200,37,200) |
| (150,37,200) | (100,37,200) | (150,200,255) | (142,198,114) | (255,255,255) |
convert $D_k$ to a $N$–bit binary string $E_k$;

$$E = E[|E_k|];$$

if $D_{k1} = 1$

break;

} else

convert $D_{k1}$ to a $N$–bit binary string $E_{k1}$;

$$E = E[|E_{k1}|];$$

if $D_{k2} = 1$

break;}

else

convert $D_{k2}$ to a $N$–bit binary string $E_{k2}$;

$$E = E[|E_{k2}|];$$

Step 4. Delete the last $r$ bits from the embedding data string $E$.

FIGURE 2. Extracting procedure.

Figure 2 illustrates the extracting procedure of the proposed method. Now an example is given as follows. Suppose that $E=100 101 011 100 010 001 011 111 101 000 111 001 0$ is a binary embedding data string, which length $E$ is 37 bits. Let $N=3$ be the length of the embedding sub-string in each embedding position. Compute $r = N - (L \text{ mod } N)\equiv 3-(37 \text{ mod } 3)=2$. Let $C$ be an 8-bit RGB color cover image, which height is 6 pixels and width is 7 pixels. Table 1 shows the matrix $C_{6 \times 7}$, which represents the cover image $C$. We embed $E$ to the cover image $C$ to produce the stego image $S$, which is listed in Table 2 by the following steps.

Step 1. Let $S_{6 \times 7}$ be a $6 \times 7$ matrix that represents the stego image $S$. The embedding positions are $s_{2,2}$, $s_{2,4}$, $s_{3,3}$, $s_{4,2}$, and $s_{4,4}$. For each element in $S_{6 \times 7}$, let $s_{ij} = c_{ij}$ if $s_{ij}$ is not a embedding position.}

Step 2. The mean values of the adjacent pixels of the embedding positions are

$$m_{2,2}=[(c_{2,1}+c_{2,1}+c_{2,2}+c_{2,2})/4]=(61,127,110),$$

$$m_{2,4}=[(c_{2,3}+c_{2,5}+c_{1,4}+c_{3,4})/4]=(72,198,113),$$

$$m_{3,3}=[(c_{3,2}+c_{3,2}+c_{3,3}+c_{3,3})/4]=(76,182,107),$$

$$m_{4,2}=[(c_{4,1}+c_{4,3}+c_{3,2}+c_{3,2})/4]=(47,94,142),$$

and

$$m_{4,4}=[(c_{3,3}+c_{3,3}+c_{3,4}+c_{3,4})/4]=(127,150,152).$$

Step 3. Let $r_{s1}=\gcd(1,2,2)/2\cdot N=58,$

$$g_{s1}=\gcd(1,2,2)/2\cdot r=75, b_{s1}=b_{c1}, i=42,$

and $s_{1,1}=(r_{s1}, g_{s1}, b_{s1})=(58, 74, 42).$ Since $N=3$, $E$ is decomposed to the following binary sub-strings with length 3 bits: $E_1=100$, $E_2=011$, $E_3=100$, $E_4=010$, $E_5=001$, $E_6=111$, $E_7=101$, $E_8=000$, $E_9=111$, $E_{10}=001$, $E_{11}=000$. Then $D_1=4$, $D_2=5,$

$D_3=3$, $D_4=2$, $D_5=1$, $D_6=7$, $D_7=5$, $D_{10}=0$, $D_{11}=7$, $D_{12}=1$, $D_{13}=0$.

Step 4. The expected value of each embedding value would be $2^{1-4}$. $s_{2,2}=m_{2,2}-(4,4,4)+(D_1,D_2,D_3)=

(61,127,110)-(4,4,4)+(4,5,3)=(61,128,109)$

$s_{2,4}=m_{2,4}-(4,4,4)+(D_1,D_2,D_3)=

(72,198,113)-(4,4,4)+(4,2,1)=(72,196,110)$

$s_{3,3}=m_{3,3}-(4,4,4)+(D_1,D_2,D_3)=

(76,182,107)-(4,4,4)+(3,7,5)=(75,185,108)$

$s_{4,2}=m_{4,2}-(4,4,4)+(D_1,D_2,D_3)=

(47,94,142)-(4,4,4)+(0,7,1)=(43,97,139)$

$s_{4,4}=m_{4,4}-(4,4,4)+(D_1,D_2,D_3)=

(127,150,152)-(4,4,4)+(0,1,1)=(123,145,147)$

Suppose that $S$ is a stego image represented by the matrix $S_{6 \times 7}$, where each element $s_{ij}=(r_{s1}, g_{s1}, b_{s1})$ denotes the RGB value of the pixel. Then we can extract the embedding data string $E$ from the stego image $S$ by the following steps.

Step 1. The embedding positions are $s_{2,2}$, $s_{2,4}$, $s_{3,3}$, $s_{4,2}$, and $s_{4,4}$. Compute the length of the embedding sub-string $N=r_{s1}=\gcd(1,2,2)/2=58-55=3$, and

$r=g_{s1}=\gcd(1,2,2)/2=75-73=2.$

Step 2. Compute $m_{2,2}=\gcd(s_{2,1}+s_{2,3}+s_{1,2}+s_{3,2})/4=(61, 127, 110)$, $m_{2,4}=\gcd(s_{2,3}+s_{2,5}+s_{1,4}+s_{3,4})/4=(72,198,113)$,

$m_{3,3}=\gcd(s_{3,2}+s_{3,4}+s_{3,2}+s_{3,4})/4=(76, 182, 107)$,

$m_{4,2}=\gcd(s_{4,1}+s_{4,3}+s_{3,2}+s_{3,2})/4=(47, 94, 142)$,

and $m_{4,4}=\gcd(s_{4,3}+s_{4,5}+s_{4,3}+s_{4,5})/4=(127, 150, 152)$.

Step 3. Extract the embedding data string $E$.

$$D_1=r_{s2,2}-(r_{s2,2}+r_{s2,2}/2-1)/2-0=61-61+4=4;$$

$$D_2=g_{s2,2}-(r_{s2,2}+r_{s2,2}/2-1)/2-0=128-127+4=5;$$

$$D_3=b_{s2,2}-(r_{s2,2}+r_{s2,2}/2-1)/2-0=109-110+4=3;$$

$$E_1=100;$$

$$E_2=101;$$

$$E_3=100;$$

$$E_4=111;$$

$$E_5=101;$$

$$E_6=101;$$

$$E_7=101;$$

$$E_8=010;$$

$$E_9=100;$$

$$E_{10}=011;$$

$$E_{11}=000;$$

$$E_{12}=011;$$

$$E_{13}=011;$$

$$E_{14}=010;$$

$$E_{15}=011;$$
III. EXPERIMENTS AND ANALYSES
In this section, we simulate the embedding procedure on nine cover images by using Python 3.7, where the embedding binary data is randomly generated. We evaluate the proposed algorithm by embedding rate (ER) and peak signal-to-noise ratio (PSNR), which are defined as follows.

Embedding capacity (EC) is usually defined by payload or Embedding rate (ER). Payload is defined by the total number of the embedding bits. Embedding rate (ER) is defined by

\[
ER = \frac{\text{Number of the embedding bits}}{\text{Number of the pixels of stego image} \times \text{bits per pixel (bpp)}}.
\]

Mean squared error (MSE) between cover image \(C\) and stego image \(S\) is defined by

\[
\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |c(i,j) - s(i,j)|^2,
\]

where \(|c(i,j) - s(i,j)|^2\) and \(c(i,j)\) and \(s(i,j)\) are pixel values of \(C\) and \(S\), respectively, at location \((i,j)\). For a RGB image, \(|c(i,j) - s(i,j)|^2 = \frac{1}{3} \left( (cR(i,j) - sR(i,j))^2 + (cG(i,j) - sG(i,j))^2 + (cB(i,j) - sB(i,j))^2 \right)\), where \(R\), \(G\), and \(B\), respectively, denotes the red, green, and blue value of the pixel at location \((i,j)\). Peak signal-to-noise ratio (PSNR) is defined by \(\text{PSNR} = 10 \times \log_{10} \frac{255^2}{\text{MSE}}\) (dB).

The nine tested images includes three typical images: Lena.jpg (Figure 3), airplane.png (Figure 4), peppers.png (Figure 5), and six ordinary photos in daily life (Table 3): flower.jpg, toy.jpg, toddler.bmp, sheeps.png, Burano.png, children.png.

![Figure 3](image-url) The experimental result of Lena.jpg.

Cover image
ER: 1.476, PSNR: 39.87

ER: 5.918, PSNR: 35.83

ER: 7.398, PSNR: 31.26

ER: 2.96, PSNR: 39.57

ER: 4.439, PSNR: 38.52

ER: 8.877, PSNR: 25.77

ER: 10.357, PSNR: 20.05

VOLUME XX, 2017

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3132424, IEEE Access
It is shown that though the proposed method only performs better than [3], it is no bottleneck to embed more bit while other proposed researches can embed no more than 6 bpp. The experimental results are shown in Table 4 and Figure 6. Generally, PSNR is said to be barely acceptable when it is greater than 20 dB, and is good when it is greater than 30 dB. In the proposed algorithm for the tested images, the ER achieves 7.4 bpp when the PSNR is nearly 30 dB, the ER achieves 8.8 bpp when the PSNR is nearly 25 dB, and the ER achieves 10.3 bpp when the PSNR is nearly 20 dB. This
section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation as well as the experimental conclusions that can be drawn.

IV. DISCUSSION

Table 5 lists the comparisons of the proposed and related work. The embedding rate is more than 7.4 bits per pixel (bpp) when the PSNR is nearly 30, and the embedding rate is more than 8.87 bpp when the PSNR is nearly 25. The proposed steganography is efficient because the operation is under the time series domain. The result also demonstrates the proposed steganography is practical under an acceptable distortion after the secrets embedding to generate a stego-image. Compare to some related work, some of other works lost information [1-3, 7-8] and some of works are based on transform domain [4, 7], which means un-efficiency. From other time domain steganography, though the proposed scheme obtains higher distortion, the capacity is larger than other works.

| Cover image | N=3 | N=5 |
|-------------|-----|-----|
| flower.jpg  | ER: 4.469, PSNR: 41.54 | ER: 8.939, PSNR: 26.45 |
| toy.jpg     | ER: 4.479, PSNR: 40.56 | ER: 8.957, PSNR: 25.94 |
| toddler.bmp | ER: 4.468, PSNR: 39.82 | ER: 8.937, PSNR: 25.89 |
| File name  | Dimension, Size | File size | Payload (bits) | ER (bpp) | PSNR (dB) | N  |
|-----------|----------------|-----------|----------------|----------|-----------|----|
| Lena.jpg  | 512*512 (14.7 KB) | 22.5 KB   | 387858         | 1.476    | 39.87     | 1  |
|           |                 | 22.6 KB   | 775716         | 2.959    | 39.57     | 2  |
|           |                 | 22.5 KB   | 1163474        | 4.439    | 38.52     | 3  |
|           |                 | 25.7 KB   | 1551432        | 5.918    | 35.83     | 4  |
|           |                 | 34.4 KB   | 1939290        | 7.398    | 31.26     | 5  |
|           |                 | 53.4 KB   | 2327148        | 8.877    | 25.77     | 6  |
|           |                 | 84.8 KB   | 2715006        | 10.357   | 20.05     | 7  |
| airplane.png | 512*512 (439 KB) | 412 KB   | 387858         | 1.476    | 36.74     | 1  |
|           |                 | 422 KB   | 775716         | 2.959    | 36.59     | 2  |
|           |                 | 440 KB   | 1163474        | 4.439    | 36.02     | 3  |
| File Name       | Size       | Height | Width  | Resolution | Dimensions | Size (KB) | Width (KB) | Height (KB) | Resolution (KB) | Dimensions (KB) |
|----------------|------------|--------|--------|------------|------------|-----------|------------|-------------|-----------------|-----------------|
| peppers.png    | 512*512    | 526 KB | 441 KB | 1.476      | 1          | 5.918     | 34.30      | 4           | 33.49           |                |
| flower.jpg     | 769*1025   | 135 KB | 75.3 KB| 1.490      | 1          | 45.46     | 44.56      | 1           | 43.77           |                |
| toy.jpg        | 1477*1108  | 108 KB | 115 KB | 1.493      | 1          | 40.56     | 43.02      | 1           | 42.40           |                |
| toddler.bmp    | 1024*952   | 2.78 MB| 2.78 MB| 1.489      | 1          | 39.82     | 41.79      | 1           | 41.32           |                |
| sheeps.png     | 2000*1500  | 3.95 MB| 4 MB   | 1.494      | 1          | 39.33     | 39.63      | 1           | 38.86           |                |
| Burano.png     | 854*864    | 711 KB | 754 KB | 1.486      | 1          | 38.69     | 38.86      | 1           | 38.12           |                |
| children.png   | 1900*2400  | 6.7 MB | 7.08 MB| 1.490      | 1          | 38.12     | 38.86      | 1           | 38.12           |                |
Table 6 is the structure similarity index measure (SSIM) of the results corresponding to Table 4 and it shows the proposed method is well-performed under this steganalysis (all the return values are near 1).

Table 7 and Table 8 are the result of Correlation and Intersection method, obviously, the proposed method obtains the same values under Intersection and still well-performed under Correlation which all values are near 1. Table 9 and Table 10 are the Chi-Square and Bhattacharyya, all the values grow according to the number of embedded secrets. Figure 7 to 10 is the visualization of the Correlation, Intersection, Chi-Square and Bhattacharyya steganalysis. The base is the cover image and result1 to result 7 is the stego-images from N=1 to 7. All the curves point out the statistical measure perform sharply bad since N=6. Table 11 illustrates the LSB enhancement of the stego-images, the results show the steganalysis does not work on the proposed method, duckling and Lena perform well when embedded secrets increase.

Table 12 is the image processing attacks for evaluations. Rotation, scaling, cutting pieces and cropping are adopted. Table 13 demonstrates that the proposed method is robust under rotation, scaling and cutting pieces but not resistant under cropping. More cropping percentage makes more data disappear.

V. CONCLUSION AND FUTURE WORKS
In this paper, a time-series steganography is proposed using adjacent mean to embed secrets in a cover image. The operation is efficient because it is under time domain computation and the distortion is acceptable while the bits per pixel is 7.4 and comes with the PSNR 31.26; the bit per pixel is 8.88 and comes with the PSNR 25.77. The proposed method is well-performed under SSIM, Correlation, Intersection and LSB enhancement. More different images will be used to test the performance in the future.

![Figure 6. PSNR (dB) versus Embedding rate (bpp) for the test images.](image)

### TABLE 5. The comparisons of the proposed and related work.

| Cover image | Embedding capacity | bpp | PSNR | Cover image | Embedding capacity | bpp | PSNR |
|-------------|-------------------|-----|------|-------------|-------------------|-----|------|
| Lena        | 775716 bits       | 2.96| 39.57| Peppers     | 775716 bits       | 2.96| 33.42|
| Peppers     | 1163474 bits      | 4.44| 38.52| Peppers     | 1163474 bits      | 4.44| 33.15|
| Proposed    | 512 x 512 (14.7 KB)| 5.92| 35.83| 512 x 512   | 1551432 bits      | 5.92| 32.20|
|             | 1939290 bits      | 7.40| 31.26| 1939290 bits| 7.40             | 7.40| 29.69|
|             | 2327148 bits      | 8.88| 25.77| 2327148 bits| 8.88             | 8.88| 25.33|
| [13]        | 512 x 512 (463 KB)| 1.48| 56.74| 512 x 512   | 393216 bits       | 1.48| 58.26|
|             | 786432 bits       | 2.89| 53.51| 512 x 512   | 786432 bits       | 2.89| 56.83|
|             | 1572864 bits      | 5.87| 50.11| 1572864 bits| 1572864 bits      | 5.87| 55.33|
| [12]        | 512 x 512 (463 KB)| 16.9 KB | <0.05| 69.71 | - | - |
|             | 512 x 512 (463 KB)| 16.9 KB | <0.02| 42.54 | - | - |
|             | 512 x 512 (32 x 32)| <0.01| 50.91| 512 x 512 (32 x 32)| <0.01| 51.05|

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
TABLE 6. Structure similarity index measure (SSIM) of the stego-images corresponding to Table 4.

| Images       | Cover image | N=1  | N=2  | N=3  | N=4  | N=5  | N=6  | N=7  |
|--------------|-------------|------|------|------|------|------|------|------|
| Lena.jpg     | 0.9999      | 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999|
| airplane.png | 0.9999      | 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999|
| peppers.png  | 0.9999      | 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999|
| flower.png   | 0.9840      | 0.9860| 0.9859| 0.9859| 0.9860| 0.9859| 0.9859| 0.9850|
| toy.png      | 0.9999      | 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999|
| todoc.bmp    | 0.9999      | 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999|
| sheeps.png   | 0.9999      | 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999|
| Burano.png   | 0.9999      | 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999|
| children.png | 0.9999      | 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999| 0.9999|

TABLE 7. Correlation of the stego-images corresponding to Table 4.

| Images       | Cover image | N=1  | N=2  | N=3  | N=4  | N=5  | N=6  | N=7  |
|--------------|-------------|------|------|------|------|------|------|------|
| Lena.jpg     | 0.9999      | 0.995 | 0.994 | 0.996 | 0.990 | 0.984 | 0.953 | 0.864 |
| airplane.png | 0.9999      | 0.998 | 0.996 | 0.976 | 0.957 | 0.962 | 0.975 | 0.986 |
| peppers.png  | 0.9999      | 0.996 | 0.993 | 0.993 | 0.992 | 0.978 | 0.967 | 0.927 |
| flower.png   | 0.975       | 0.971 | 0.964 | 0.936 | 0.902 | 0.881 | 0.753 | 0.518 |
| toy.png      | 0.997       | 0.998 | 0.998 | 0.998 | 0.998 | 0.995 | 0.958 | 0.859 |
| todoc.bmp    | 0.990       | 0.984 | 0.972 | 0.963 | 0.963 | 0.957 | 0.957 | 0.749 |
| sheeps.png   | 0.969       | 0.961 | 0.958 | 0.958 | 0.958 | 0.967 | 0.979 | 0.934 |
| Burano.png   | 0.998       | 0.994 | 0.981 | 0.963 | 0.963 | 0.957 | 0.979 | 0.977 |
| children.png | 0.982       | 0.979 | 0.974 | 0.969 | 0.968 | 0.960 | 0.871 | 0.871 |

TABLE 8. Intersection of the stego-images corresponding to Table 4.

| Images       | Cover image | N=1  | N=2  | N=3  | N=4  | N=5  | N=6  | N=7  |
|--------------|-------------|------|------|------|------|------|------|------|
| Lena.jpg     | 30.052      | 29.706| 29.603| 30.046| 30.043| 30.043| 30.046| 30.051|
| airplane.png | 19.699      | 18.570| 19.512| 19.698| 19.699| 19.699| 19.699| 19.699|
| peppers.png  | 18.549      | 18.324| 18.442| 18.544| 18.545| 18.547| 18.549| 18.549|
| flower.png   | 10.417      | 10.361| 10.369| 10.412| 10.412| 10.413| 10.414| 10.416|
| toy.png      | 36.105      | 33.904| 33.958| 36.097| 36.100| 36.102| 36.104| 36.105|
| todoc.bmp    | 87.634      | 87.601| 87.631| 87.634| 87.634| 87.634| 87.634| 87.634|
| sheeps.png   | 25.482      | 25.479| 25.481| 25.482| 25.482| 25.482| 25.482| 25.482|
| Burano.png   | 45.637      | 44.844| 44.804| 45.637| 45.637| 45.637| 45.637| 45.637|
| children.png | 41.704      | 41.447| 41.603| 41.703| 41.703| 41.704| 41.704| 41.704|
### TABLE 9. Chi-square of the stego-images corresponding to Table 4.

| images    | cover image | N=1 | N=2 | N=3 | N=4 | N=5 | N=6 | N=7 |
|-----------|-------------|-----|-----|-----|-----|-----|-----|-----|
| Lena.jpg  | 0.000       | 1.160 | 1.047 | 0.869 | 1.207 | 5.131 | 22.334 | 257.876 |
| airplane.png | 0.000   | 0.288 | 0.753 | 9.940 | 134.481 | 492.855 | 519.397 | 529.643 |
| peppers.png | 0.000     | 0.468 | 1.089 | 1.503 | 2.298 | 10.349 | 62.484 | 182.128 |
| flower.jpg | 0.000     | 5.349 | 6.190 | 7.653 | 16.727 | 39.951 | 141.793 | 941.424 |
| toy.jpg   | 0.000     | 1.085 | 1.172 | 1.225 | 1.500 | 2.376 | 30.542 | 461.164 |
| toddler.bmp | 0.000   | 12.572 | 38.450 | 156.938 | 543.545 | 1244.621 | 2750.055 | 2042.957 |
| sheeps.png | 0.000      | 9.970 | 25.378 | 175.174 | 462.535 | 760.360 | 1882.113 | 8613.994 |
| Burano.png | 0.000      | 0.436 | 3.341 | 21.323 | 102.341 | 439.281 | 922.341 | 1327.185 |
| children.png | 0.000 | 673.549 | 848.934 | 659.230 | 1242.368 | 2227.952 | 3769.040 |

### TABLE 10. Bhattacharyya of the stego-images corresponding to Table 4.

| images    | cover image | N=1 | N=2 | N=3 | N=4 | N=5 | N=6 | N=7 |
|-----------|-------------|-----|-----|-----|-----|-----|-----|-----|
| Lena.jpg  | 0.000       | 0.046 | 0.047 | 0.044 | 0.057 | 0.068 | 0.125 | 0.245 |
| airplane.png | 0.000     | 0.042 | 0.046 | 0.113 | 0.210 | 0.296 | 0.365 | 0.408 |
| peppers.png | 0.000     | 0.042 | 0.045 | 0.043 | 0.057 | 0.128 | 0.233 | 0.340 |
| flower.jpg | 0.000     | 0.107 | 0.109 | 0.115 | 0.140 | 0.175 | 0.252 | 0.373 |
| toy.jpg   | 0.000     | 0.058 | 0.058 | 0.058 | 0.061 | 0.076 | 0.147 | 0.277 |
| toddler.bmp | 0.000   | 0.046 | 0.077 | 0.146 | 0.220 | 0.291 | 0.366 | 0.432 |
| sheeps.png | 0.000      | 0.069 | 0.093 | 0.151 | 0.219 | 0.285 | 0.347 | 0.409 |
| Burano.png | 0.000      | 0.033 | 0.067 | 0.129 | 0.203 | 0.269 | 0.330 | 0.330 |
| children.png | 0.000 | 0.115 | 0.125 | 0.146 | 0.189 | 0.253 | 0.326 | 0.388 |

### FIGURE 7. Correlation of the test stego-images.

### FIGURE 8. Intersection of the test stego-images.

### FIGURE 9. Chi-Square of the test stego-images.

### FIGURE 10. Bhattacharyya of the test stego-images.
TABLE 11. LSB enhancement of all cover and stego-images.

| Cover image: Lena | N=1 | N=2 | N=3 | N=4 | N=5 | N=6 | N=7 |
|------------------|-----|-----|-----|-----|-----|-----|-----|
| ![Enhanced LSB]  | ![Enhanced LSB] | ![Enhanced LSB] | ![Enhanced LSB] | ![Enhanced LSB] | ![Enhanced LSB] | ![Enhanced LSB] | ![Enhanced LSB] |

| Cover image: airplane | N=1 | N=2 | N=3 | N=4 | N=5 | N=6 | N=7 |
|-----------------------|-----|-----|-----|-----|-----|-----|-----|
| ![Enhanced LSB]      | ![Enhanced LSB] | ![Enhanced LSB] | ![Enhanced LSB] | ![Enhanced LSB] | ![Enhanced LSB] | ![Enhanced LSB] | ![Enhanced LSB] |
### TABLE 12. Image process attacks.

| Attack Type        | Rotation | Scaling | Cutting Pieces | Cropping |
|--------------------|----------|---------|----------------|----------|
| stego-image rotated | 0        | 0       | 0              | 10       |
| stego-image scaled up (10%) | 0        | 10%     | 0              | 10%      |
| stego-image cutting pieces | 0        | 0       | 10%            | 10%      |
| stego-image cropping (10%)  | 0        | 0       | 0              | 10%      |

### TABLE 13. Robustness evaluation. (Damaged bits percentage after image processing)

| Image/attack | Rotation | Scaling | Cutting Pieces | Cropping |
|--------------|----------|---------|----------------|----------|
| Lena.jpg     | 0        | 0       | 0              | 10       |
| airplane.png | 0        | 0       | 0              | 10       |
| peppers.png  | 0        | 0       | 0              | 10       |
| flower.jpg   | 0        | 0       | 0              | 10       |
| toy.jpg      | 0        | 0       | 0              | 10       |
| toddler.bmp  | 0        | 0       | 0              | 10       |
| sheeps.png   | 0        | 0       | 0              | 10       |
| Burano.png   | 0        | 0       | 0              | 10       |
| children.png | 0        | 0       | 0              | 10       |

### REFERENCES

[1] M. Juneja and P. S. Sandhu, “An improved LSB based steganography technique for RGB color images,” *International Journal of Computer and Communication Engineering*, vol. 2, pp. 513-517, 2013.

[2] S. Hemalatha, U. D. Acharya, A. Renuka and P. R. Kamath, “Secure and high capacity image steganography technique,” *Signal & Image Processing : An International Journal*, vol. 4, pp. 83-89, 2013.

[3] M. Tang, J. Hu and W. Song, “A high capacity image steganography using multi-layer embedding,” *Optik*, vol. 125, pp. 3972-3976, 2014.

[4] Y. K. Lin, “A data hiding scheme based upon DCT coefficient modification,” *Comput. Stand. Interfaces*, vol. 36, pp. 855-862, 2014.

[5] K. H. Jung and K. Y. Yoo, “Data hiding using edge detector for scalable images,” *Multimed Tools Appl*, vol. 71, pp. 1455-1468, 2014.
[6] S. Shen, L. Huang and Q. Tian, “A novel data hiding for color images based on pixel value difference and modulus function,” Multimed Tools Appl, vol. 74, pp. 707-728, 2015.

[7] S. Hamad and A. Khalifa, “Non-blind Data hiding for RGB images using DCT-based fusion and H.264 compression concepts,” ACSII Advances in Computer Science: an International Journal, vol. 4, pp. 97-103, 2015.

[8] W. L. Xu, C. C. Chang, T. S. Chen and L. M. Wang, “An improved least-significant-bit substitution method using the modulo three strategy,” Displays, vo. 42, pp. 36-42, 2016.

[9] A. Nilizadeh and A. R. N. Nilchi, “A Novel Steganography Method Based on Matrix Pattern and LSB Algorithms in RGB Images,” in 1st Conference on Swarm Intelligence and Evolutionary Computation, Barn, Iran, 2016.

[10] K. Muhammad, J. Ahmad, N. U. Rehman, Z. Jan and M. Sajjad, “CISSKA-LSB: color image steganography using stego key-directed adaptive LSB substitution method,” Multimed Tools Appl, vol. 76, pp. 8597-8626, 2017.

[11] D. R. I. M. Setiadi and E. H. Rachmaswanto, “Secure Image Steganography Algorithm Based on DCT with OTP Encryption,” Journal of Applied Intelligent System, vol. 2, pp. 1-11, 2017.

[12] H. M. Farhan and Z. A. Alwan, “Improved method using a two Exclusive-OR to binary image in RGB color image steganography,” International Journal of Engineering and Technology, vol. 7, pp. 4295-4299, 2018.

[13] S. Tyagi, R. K. Dwivedi and A. K. Saxena, “High Capacity Steganography Protected using Shamir’s threshold scheme and Permutation Framework,” International Journal of Innovative Technology and Exploring Engineering, vol. 8, pp. 784-795, 2019.

[14] Y. Y. Tsai, D. S. Tsai and C. L. Liu, “Reversible data hiding scheme based on neighboring pixel differences,” Digital Signal Processing, vol. 23, pp. 919-927, 2013.

[15] J. Li, X. Li and B. Yang, “Reversible data hiding scheme for color image based on prediction-error expansion and cross-channel correlation,” Signal Processing, vol. 93, pp. 2748-2758, 2013.

[16] K. J. Besteena and J. Philumon, “Reversible Data Hiding in Selectively Encrypted RGB Images by Reserving Room in Advance,” in 2014 First International Conference on Computational Systems and Communications, Trivandrum, India, 2014.

[17] R. Thabit and B. Khoo, “A new robust lossless data hiding scheme and its application to color medical images,” Digital Signal Processing, vol. 38, pp. 77-94, 2015.

[18] H. T. Wu, J. L. Dugelay and Y. Q. Shi, “Reversible image data hiding with contrast enhancement,” IEEE Signal Processing Letters, vol. 22, pp. 81-85, 2015.

[19] B. Ou, X. Li, Y. Zhao and R. Ni, “Efficient color image reversible data hiding based on channel-dependent payload partition and adaptive embedding,” Signal Processing, vol. 108, pp. 642-657, 2015.

[20] H. T. Wu, S. Tang, J. Huang and Y. Q. Shi, “A novel reversible data hiding method with image contrast enhancement,” Signal Processing: Image Communication, vol. 62, pp. 64-73, 2018.

[21] P. Singh and B. Raman, “Reversible data hiding based on Shamir’s secret sharing for color images over cloud,” Information Sciences, vol. 422, pp. 77-97, 2018.

[22] X. Liao and C. Shu, “Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels,” J. Vis. Commun. Image R., vol. 28, pp. 21-27, 2015.

[23] N. Kittawi and A. Al-Haj, “Reversible Data Hiding in Encrypted Images,” in 8th International Conference on Information Technology, Amman, Jordan, 2017.

[24] Z. Tang, Q. Lu, H. Lao, C. Yu and X. Zhang, “Error-free reversible data hiding with high capacity in encrypted image,” Optik, vol. 157, pp. 750-760, 2018.

[25] C. Qin, Z. He, X. Luo and J. Dong, “Reversible data hiding with differential compression in encrypted image,” Information Sciences, vol. 465, pp. 285-304, 2018.

[26] L. Xiong and Y. Shiu, “On the Privacy-Preserving Outsourcing Scheme of Reversible Data Hiding over Encrypted Image Data in Cloud Computing,” Computers, Materials & Continua, vol. 55, pp. 523-539, 2018.

[27] Z. L. Liu and C. M. Pun, “Reversible image reconstruction for reversible data hiding in encrypted images,” Signal Processing, vol. 161, pp. 50-62, 2019.

[28] Y. C. Chen, T. H. Hung, S. H. Hsieh and C. W. Shiu, “A New Reversible Data Hiding in Encrypted Image Based on Multi-Secret Sharing and Lightweight Cryptographic Algorithms,” IEEE Transactions on Information Forensics and Security, vol. 14, pp. 3332-3343, 2019.

[29] C. Qin, X. Qian, W. Hong and X. Zhang, “An efficient coding scheme for reversible data hiding in encrypted image with redundancy transfer,” Information Sciences, vol. 487, 2019.

[30] Y. Fu, P. Kong, H. Yao, Z. Tang and C. Qin, “Effective reversible data hiding in encrypted image with adaptive encoding strategy,” Information Sciences, vol. 494, pp. 21-36, 2019.

[31] D. Xiao, F. Li, M. Wang and H. Zheng, “A Novel High-Capacity Data Hiding in Encrypted Images Based on Compressive Sensing Progressive Recovery,” IEEE Signal Processing Letters, vol. 27, pp. 296-300, 2020.

[32] D. Huang and J. Wang, “High-capacity reversible data hiding in encrypted image based on specific encryption process,” Signal Processing: Image Communication, vol. 80, pp. 115632, 2020.

[33] Y. Qiu, Q. Ying, X. Lin, Y. Zhang and Z. Qian, “Reversible Data Hiding in Encrypted Images With Dual Data Embedding,” IEEE Access, vol. 8, pp. 23209-23220, 2020.

[34] A. K. Sahu and M. Sahu, “Digital Image Steganography and Steganalysis: A Journey of the Past Three Decades,” Open Computer Science, vol. 10, pp. 296-342, 2020.

[35] A. K. Sahu, G. Swain, M. Sahu and J. Hemalatha, “Multi-Directional Block Based PVD and Modulus Function Image...
Steganography to Avoid FOBP and IEP,” *Journal of Information Security and Applications*, vol. 58, 2021.

[36] M. Sahu, N. Padhy, S. S. Gantayat and A. K. Sahu, “Shadow Image Based Reversible Data Hiding Using Addition and Substraction Logic on the LSB Planes,” *Sensing and Imaging*, vol. 22, 2021.

[37] A. K. Sahu and G. Swain, “High Fidelity Based Reversible Data Hiding Using Modified LSB Matching and Pixel Difference,” *Journal of King Saud University-Computer and Information Sciences*, 2019.

[38] A. K. Sahu and G. Swain, “An Optimal Information Hiding Approach Based on Pixel Value Differencing and Modulus Function,” *Wireless Personal Communications*, vol. 108, pp. 159-174, 2019.

[39] M. Y. V alandar, P. Ayubi and M. J. Barani, “A New Transform Domain Steganography Based on Modified Logistic Chaotic Map for Color Images,” *Journal of Information Security and Applications*, vol. 34, pp. 142-151, 2017.

[40] M. Y. V alandar, M. J. Barani, P. Ayubi and M. Aghazadeh, “An Integer Wavelet Transform Image Steganography Method Based on 3D Sine Chaotic Map,” *Multimedia Tools and Applications*, vol. 78, pp. 9971-9989, 2019.

[41] P. Ayubi, M. J. Barani, M. Y. V alandar, B. Y. Irani and R. S. M. Sadigh, “A New Chaotic Complex Map for Robust Video Watermarking,” *Artificial Intelligence Review*, vol. 54, pp. 1237-1280, 2021.

[42] C. C. Chang, “Neural Reversible Steganography with Long Short-Term Memory,” *Security and Communication Networks*, vol. 2021, 2021.

[43] M. Asif, L. Kumar, G. Swami and A. Arora, “High-Capacity Reversible Data Hiding Using Deep Learning,” *ASIACON*, PUNE, India, 2021.

[44] J. Zhu, R. Kaplan, J. Johnson and F. F. Li, “HiDDeN: Hiding Data With Deep Networks,” *ECCV*, 2018.

[45] S. Baluja, “Hiding Images in Plain Sight: Deep Steganography,” *31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017.

[46] K. R. Prasad, “The Design and Development of Data Hiding Using Deep Learning,” *Journal of Advances and Scholarly Researches in Allied Education*, vol. 16, pp. 970-974, 2019.

[47] Y. Shang, S. Jiang, D. Ye and J. Huang, “Enhancing the Security of Deep Learning Steganography via Adversarial Examples,” *Mathematics*, vol. 8, pp. 1446-1455, 2020.