

A novel steganography method for binary and color halftone images

Efe Çiftci¹ and Emre Sümer²

¹ Department of Computer Engineering, Çankaya University, Ankara, Turkey

² Department of Computer Engineering, Başkent University, Ankara, Turkey

ABSTRACT

Digital steganography is the science of establishing hidden communication on electronics; the aim is to transmit a secret message to a particular recipient using unsuspecting carriers such as digital images, documents, and audio files with the help of specific hiding methods. This article proposes a novel steganography method that can hide plaintext payloads on digital halftone images. The proposed method distributes the secret message over multiple output copies and scatters parts of the message randomly within each output copy for increased security. A payload extraction algorithm, where plain carrier is not required, is implemented and presented as well. Results gained from conducted objective and subjective tests prove that the proposed steganography method is secure and can hide large payloads.

Subjects Algorithms and Analysis of Algorithms, Cryptography

Keywords Steganography, Halftone image, Image processing, Plaintext payload, Secret sharing

INTRODUCTION

Steganography is the practice of hidden communication using unsuspecting media as carriers (*Cheddad et al., 2010*). The basis of this communication is that the presence of this communication should be known only by the participating parties while keeping everybody else unaware of this communication. In order to establish such hidden communication, steganographic methods require both a payload and a carrier cover media. Depending on the type of cover media, the payload is hidden onto the cover media by suitable algorithms. The result of this process is stego media, which is expected to look and feel exactly like a regular media and raise no suspicions.

While secret communication has been available since the oldest days of known human history, the advancements in digital communication have opened many new ways for steganography in the digital domain. In the digital domain, file types such as images, sounds, and videos are the most commonly used carriers for various steganography methods. Steganography methods for these carriers can be enhanced using additional techniques such as payload encryption or payload compression to obtain better results in terms of payload security, increased payload capacity, and improved stego media quality (*Sharma & Batra, 2021; Sari et al., 2019; Sharma et al., 2019*).

Digital image carriers can be mainly classified into (a) color images, (b) grayscale images, and (c) binary images. Color images are most commonly represented with 24 bits per pixel (red, green, and blue channels, 8-bits per channel), though 8-bit (indexed) and 32-bit (RGB + 8-bit alpha channel) color images are also available (*Shreiner & Group,*

Submitted 28 January 2022

Accepted 18 July 2022

Published 16 August 2022

Corresponding author

Efe Çiftci, efeciftci@cankaya.edu.tr

Academic editor

Sedat Akleylek

Additional Information and
Declarations can be found on
page 21

DOI [10.7717/peerj-cs.1062](https://doi.org/10.7717/peerj-cs.1062)

© Copyright

2022 Çiftci and Sümer

Distributed under

Creative Commons CC-BY 4.0

OPEN ACCESS









Channel 1 (Red):	0	0	0	0	1	1	1	1
Channel 2 (Green):	0	0	1	1	0	0	1	1
Channel 3 (Blue):	0	1	0	1	0	1	0	1
Result:								

Figure 1 Colors in a three-channel color halftone image.

Full-size  DOI: 10.7717/peerj-cs.1062/fig-1

2009). Pixels in grayscale images are represented with eight bits that store intensity information.

Binary images have only black and white tones; therefore, only one bit is enough to represent a pixel in binary images (0 – black, 1 – white). Binary images can be created from grayscale source images by either (a) thresholding (*Al-Amri, Kalyankar & Khamitkar, 2010; Huang & Chau, 2008; Wang, Chung & Xiong, 2008*) or (b) halftoning (*Chan & Chen, 1998; Shiau & Fan, 1994; Knuth, 1987; Floyd, 1976*). Halftone images are crafted from grayscale images through specific methods to achieve a look similar to continuous-tone grayscale images (*Lau & Arce, 2001; Ulichney, 1987*). They are most commonly preferred for printing in newspapers and books, where they help save ink by printing a black and white only image instead of more expensive grayscale images. Halftone images also occupy less space on storage devices, and they compress better as each pixel in the halftone image needs only one bit to be represented, as opposed to at least eight bits for grayscale or color images.

Inspired by color images with three channels, binary halftone images can also be improved to contain three separate binary channels to represent red, green, and blue color information in them. These types of images are called color halftone images and are produced by performing the regular halftoning procedure individually on all channels of a source color image. Because these images use three separate binary channels for each pixel, these images can represent colors with eight different values (*Fig. 1*). When compared to binary halftone images, the primary advantages presented by color halftone images are the increased capacity for payloads and lesser payload visibility due to the human visual system's poor perception of similar colors (*e.g., yellow and white*).

Due to the facts, such as there is a wide range of previously published steganography methods for color and grayscale images (as will be discussed in the *Related Work* section), and also few discovered examples that hide plaintext payloads rather than other types of payloads (such as images, as proposed by *John Blesswin et al., 2020; Yan et al., 2015, etc*); the steganography method that is proposed in this article focuses on binary and color halftone images as carriers and hides plaintext payloads in them. The payload is hidden into the carrier during conversion of the carrier from grayscale/color into halftone format.

The method is implemented to work on halftone images that are generated *via* either predefined patterns or error diffusion.

Although it would simplify the steps required for both embedding and extraction, hiding the payload in a single image may allow attackers to easily gain access to the whole payload with proper attack methods (Luo *et al.*, 2021; Quach, 2014; Jiang *et al.*, 2005). In order to improve the security of the payload for such cases, the proposed method takes inspiration from the secret sharing method (Naor & Shamir, 1995) and scatters portions of the payload into multiple slightly different output images. This way, it is ensured that the payload can be successfully extracted if and only if all output copies are collected back and processed together again. The results gained through conducted objective and subjective tests have shown that the proposed method can hide large plaintext payloads successfully without causing many disturbances on the carrier media. The length of the maximum payload is directly proportional to the spatial resolution of the cover media.

In addition to the hiding algorithm, a suitable payload extraction algorithm has also been implemented and presented in this article. In contrast to some existing methods, our extraction algorithm can successfully recover the payload without the need for the plain version of the carrier. This ensures a potential weakness is avoided where an attacker may attempt to extract the payload by comparing the plain and stego media.

The remainder of this article is organized as follows. In *Related Work*, we share some of the existing related and remarkable works in the steganography domain. In *Materials & Methods*, we explain both the proposed hiding algorithm (with its variations for mentioned carrier types and halftoning methods) and the payload extraction algorithm. In *Experiments & Results*, we present the conducted objective and subjective experimental tests and their results. In *Discussions*, we discuss the results of the experiments and highlight significant outcomes of these experiments. Finally, in the *Conclusions*, we summarize the presented study and state possible future studies related to the proposed method.

RELATED WORK

Digital images with higher bits per pixel (bpp) ratios, such as color and grayscale images, provide foundations for a wide variety of steganographic methods. The most common steganography method for color and grayscale images is the LSB method (Cheddad *et al.*, 2010), where the payload is hidden into the least significant bit (or more than one bit in some cases) of each byte. The changes caused by this method are usually insignificant to the human visual system but can easily be detected by computers. Therefore, variations of the LSB method that aim to evade being detected by software by utilizing additional methods such as encryption have since been published (Kordov & Zhelezov, 2021; Zhou *et al.*, 2016; Juneja & Sandhu, 2014, 2013; Hsiao, Chan & Chang, 2009; Sutaone & Khandare, 2008).

Steganography on color images is not limited to methods that operate on the least significant bit. For example, Nilizadeh *et al.* (2022) and Nilizadeh *et al.* (2017) propose methods that can hide any type of payload into the blue channel using matrix patterns generated from the green channel of an image. Mowafi *et al.* (2019) proposes a method that

can hide plaintext payloads into an image's Cb and Cr components using matrix patterns generated from the Y component. Color and grayscale images offer many possibilities, but the full review of color and grayscale images is out of this paper's scope.

Since binary images do not offer as many features as grayscale or color images, methods that aim to hide in halftone images differ from the methods that use color or grayscale carriers; this has resulted in the proposal of new different methods that especially exploit the distinct structure of binary images. [Cruz et al. \(2018\)](#) proposes a method in which letters, digits, and punctuation marks are represented with unique 3×3 patterns, and the pattern form of a plaintext payload is distributed in cover media appropriately. Since the embedding process on halftone images causes more visible distortions than grayscale or color images, several methods such as those proposed by [Yu et al. \(2021\)](#), [Lu et al. \(2019\)](#), [Xue et al. \(2019\)](#) have been developed to minimize these distortions. Some methods, such as those proposed by [Fu & Au \(2003, 2001a\)](#), [Pei & Guo \(2003\)](#), require a plain (*i.e.*, does not carry payload) halftone cover media during payload extraction. However, on the other hand, there are other methods, such as those proposed by [Rosen & Javidi \(2001\)](#), that do not require the plain cover media for payload extraction.

Several works published so far focus on improving the output quality of visual cryptography, which is a method for hiding visual payloads. [Naor & Shamir \(1995\)](#) propose the secret sharing method, where the pixels in the payload are scattered over multiple copies of the output (*i.e.*, shares). Instead of keeping a secret message in one place, secret sharing aims to make it difficult for attackers to access the whole secret message directly and to ensure the security of the secret message by splitting it into subparts that will not make sense on their own and sharing them among more than one person. In the mentioned method, all shares must be stacked over to extract the payload. [Wang, Arce & Di Crescenzo \(2006\)](#) propose a method where the pixels of a binary payload image are hidden and distributed in an amount of generated halftone images using visual cryptography. Their method aims to improve the output quality by encoding the pixels using direct binary search method ([Analoui & Allebach, 1992](#)) to decrease the noise caused by visual cryptography. [Fu & Au \(2001b\)](#) propose two methods, named intensity selection and connection selection, that aim to improve the visual quality of carrier outputs generated by error diffusion algorithms by choosing the best locations for hiding the payload. Of these methods, intensity selection offers better visual quality, while connection selection has lower computational complexity than intensity selection. Some methods that focus on visual cryptography suffer from image expansion, in which a 1×1 white or black pixel in the payload gets to be represented by a larger block of pixels (*e.g.*, 2×2) in the output images. Several methods, such as those proposed by [Askari, Heys & Moloney \(2013\)](#) and [Chen et al. \(2007\)](#), overcome this problem and enable both the cover and the payload images to be the same size.

MATERIALS AND METHODS

The proposed algorithm ([Algorithm 1](#)) requires a source image I , a plaintext payload P , and the number of desired output images $NSHARES$ as inputs. The algorithm adopts the previously explained secret sharing methodology; it produces a number of (*i.e.*, $NSHARES$)

Algorithm 1 Payload hiding.**Input:** source image (I), plaintext payload (P), number of shares (NSHARES)**Output:** a set of stego images (I_{ht})

```

1: procedure PAYLOADHIDING(I, P, NSHARES)
2:    $dLen \leftarrow I.capacity/P_{binary}.length$ 
3:   if  $dLen < 1$  then                                     ▶ terminate if payload is larger than
4:     return                                               ▶ the capacity
5:   end if
6:   let  $I_{ht}$  be a set of NSHARES-long empty images
7:   for each  $dLen$ -long blocks in I as block; do           ▶ choose a proper position for each
8:      $R \leftarrow$  randomly chosen output image           ▶ payload bit among all output images
9:      $RR \leftarrow$  randomly chosen (x, y) position in block ▶ and hide it using one of the four
10:     $B \leftarrow P_{binary}.next$                          ▶ provided methods depending on the
11:     $I_{ht}(R,RR) \leftarrow embed(R, block, RR, B)$        ▶ carrier type and halftoning method
12:  end for
13: end procedure

```

output halftone images I_{ht} and distributes each bit of the payload (*i.e.*, B) in calculated positions over a randomly chosen output image (*i.e.*, R) among all output images. This procedure produces multiple output images that have slight differences but still share an identical look. By distributing the payload this way, it is ensured that attackers will not be able to successfully extract the payload if they are missing even just a single output image.

Aside from the secret sharing mechanism explained above, the hiding procedure also ensures that all bits in a given output image are spread across the image and are not concentrated in a specific region. This is achieved by calculating a distance length $dLen$; which is the ratio of the number of usable pixels in the cover image to the length of the payload. The image is then divided into groups of $dLen$ sequential pixels (*i.e.*, *block*), and each bit is embedded into a different group. Furthermore, the positions of these bits are randomized within each group (*i.e.*, RR) in order to prevent detection from statistical attacks that target every N^{th} pixel. As a result of this randomization, the algorithm does not produce deterministic outputs; a different set of output images will be produced on each execution, although the inputs did not change.

The *embed()* function in Algorithm 1 denotes the four methods that have been implemented specifically for carrier types (*i.e.*, gray or color) and halftoning method (*i.e.*, halftoning *via* patterns or error diffusion). In order to hide a payload bit; the chosen output carrier and position (x, y) where the payload bit will be embedded must be determined earlier. Details of each embedding method are explained in the following sections.

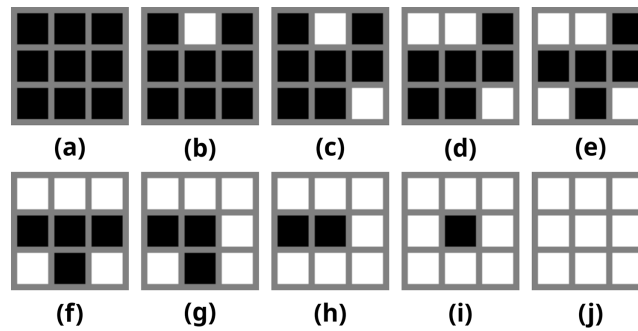


Figure 2 (A–J) Binary patterns used for constructing halftone images.

Full-size DOI: 10.7717/peerj-cs.1062/fig-2

Hiding on halftone images using patterns

This method requires a number of binary patterns for creating the halftone image. For this purpose, 10 3×3 binary patterns p_{0-9} have been defined as demonstrated by [Zhang \(2017\)](#) (Fig. 2).

Pixels in a regular 8-bit grayscale image contain 256 different intensity levels. In order to determine which pattern will be used for which intensity level in the source image I , these levels must be divided into 10 groups, and a simpler version of the source image (*i.e.*, $I_{grouped}$) must be generated according to Eq. (1).

$$I_{grouped}(i, j) = \left\lfloor \frac{I(i, j)}{26} \right\rfloor \quad (1)$$

The left-hand side of Eq. (1) is a set of values in the range of 0 and 9, and the values of pixels in the mentioned simpler version consist of these values. Then, the final halftone image will be created by matching the value of each pixel with the matching pattern. The stages of this process are demonstrated in Fig. 3. It should be noted that since every pixel in the original image is now represented with a 3×3 pattern, the spatial resolution of output images is larger (*e.g.*, 256×256 grayscale images are converted into 768×768 halftone images).

In order to hide the payload, we propose to alter this conversion process such that when a “0” or “1” bit is to be hidden in a pixel, the pattern prior or next to the actually determined pattern is used in a randomly chosen output image, while the determined pattern is still used in the same position in all other output images (Eq. (2)). Since neighboring patterns are involved in this process, our algorithm avoids hiding bits in pure black or white regions in the image as there is no pattern prior to p_0 or next to p_9 , respectively. This precaution also prevents the generation of visible noise in smooth black or white regions of the produced stego media.

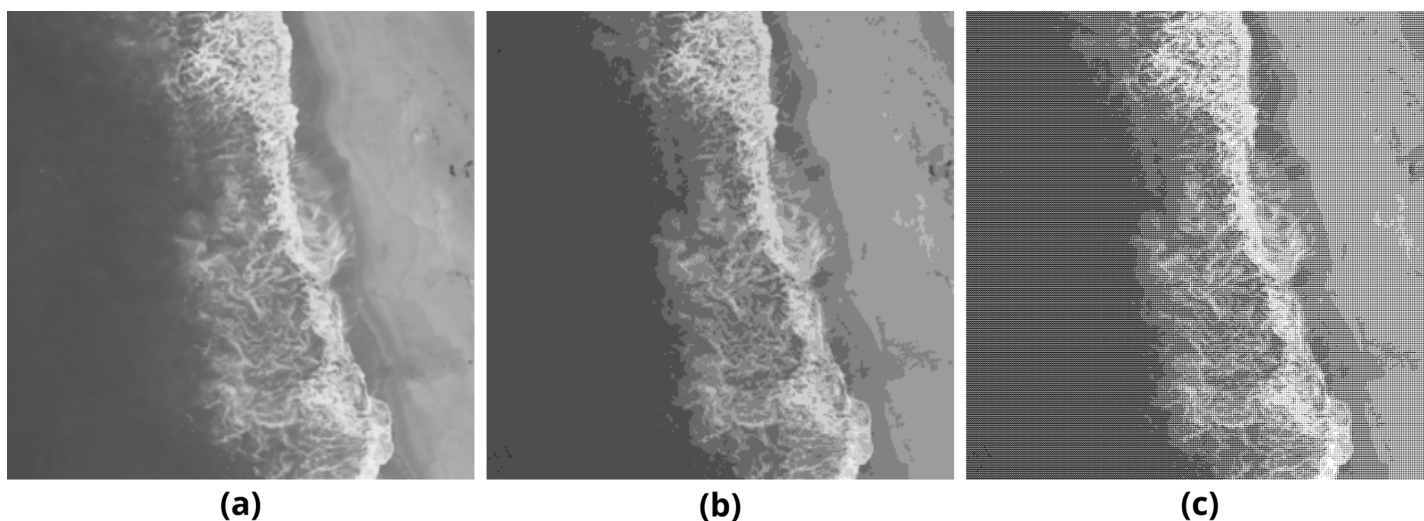


Figure 3 Stages of converting a grayscale image into pattern-based halftone image. (A) Source image, (B) Simpler image, (C) Final image.

Full-size  DOI: [10.7717/peerj-cs.1062/fig-3](https://doi.org/10.7717/peerj-cs.1062/fig-3)

$$I_{ht}(R, i, j) = \begin{cases} p(I_{grouped}(i, j) - 1), & \text{if } bit = 0 \\ p(I_{grouped}(i, j) + 1), & \text{if } bit = 1 \\ p(I_{grouped}(i, j)), & \text{if } I_{grouped}(i, j) = 0 \\ & \text{or } I_{grouped}(i, j) = 9 \end{cases} \quad (2)$$

Sample pairs of binary and color output images generated with the mentioned method are presented in Fig. 4. The length of the embedded payload is 2,048 bytes.

Hiding on halftone images using error diffusion method

Error diffusion is a popular halftoning method in which the residual error of each processed pixel is distributed to its neighboring pixels, creating a smoother appearance and a closer appearance to the original image in the process. In order to distribute the error, a coefficient filter that tells which neighboring pixel will receive how much of the error must be used. There have been numerous methods proposed with different filters; Floyd-Steinberg (Floyd, 1976), Shiao-Fan (Shiao & Fan, 1994), and Jarvis-Judice-Ninke (Jarvis, Judice & Ninke, 1976) are a few among the most popular. For example, the Floyd-Steinberg filter is demonstrated in Fig. 5.

We have previously explained that in order to hide a 0 or 1 bit, the proposed method chooses one random output share among others, and encodes a slightly different pattern than the ones used in the rest of the other share images. In order to adapt this method to error diffusion, we have simplified this process since we no longer have a set of 10 patterns to choose from. Instead, when a 0 or 1 is to be hidden, the value of the chosen pixel in one random output image is set to the desired value, while the pixels in the same coordinate in all the other output images are set to the opposite (Eq. (3)).

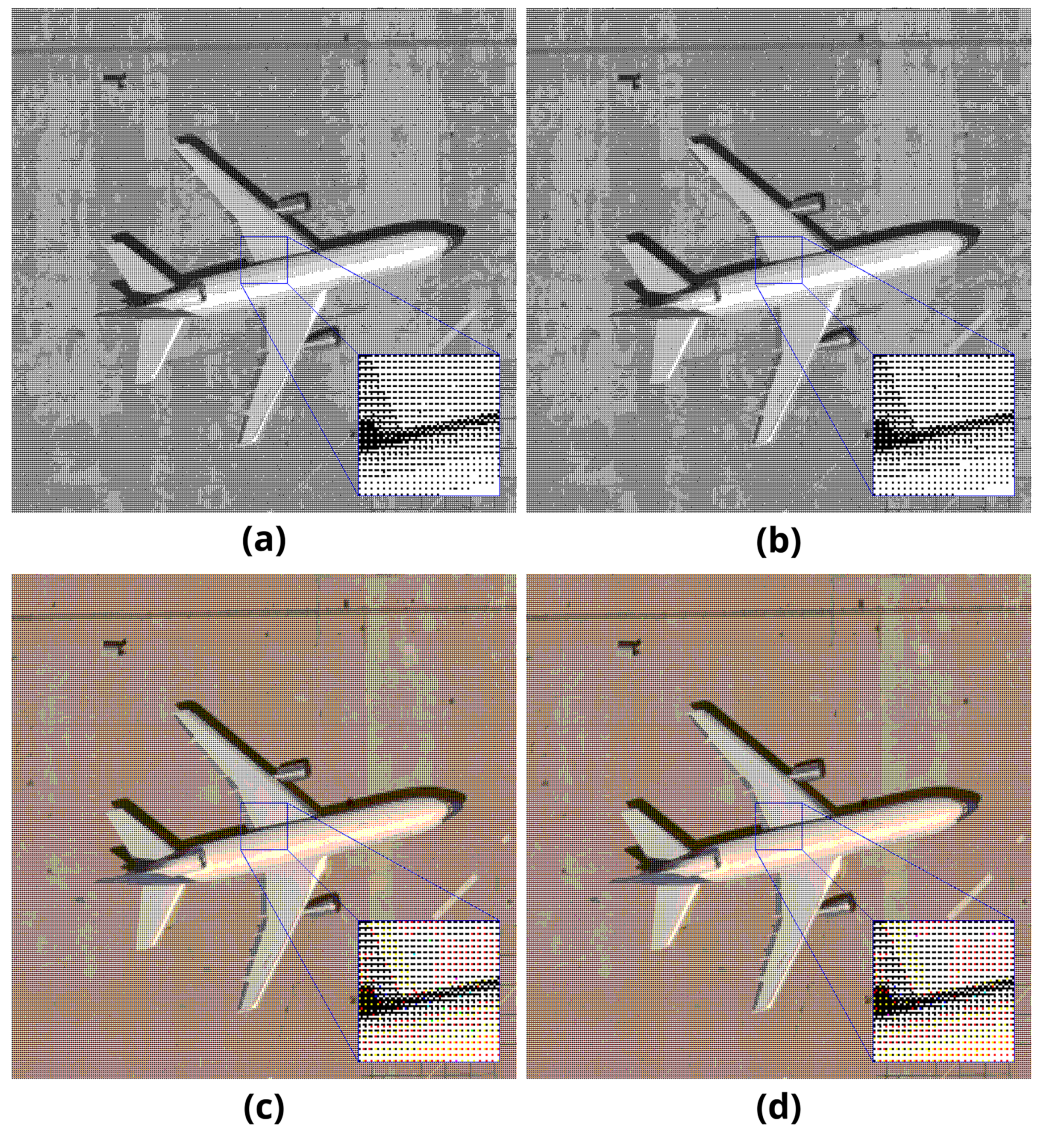


Figure 4 Pattern method results on (A-B) binary and (C-D) color images.

Full-size DOI: [10.7717/peerj-cs.1062/fig-4](https://doi.org/10.7717/peerj-cs.1062/fig-4)

	●	7/16
3/16	5/16	1/16

Figure 5 Floyd-Steinberg coefficient filter.

Full-size DOI: [10.7717/peerj-cs.1062/fig-5](https://doi.org/10.7717/peerj-cs.1062/fig-5)

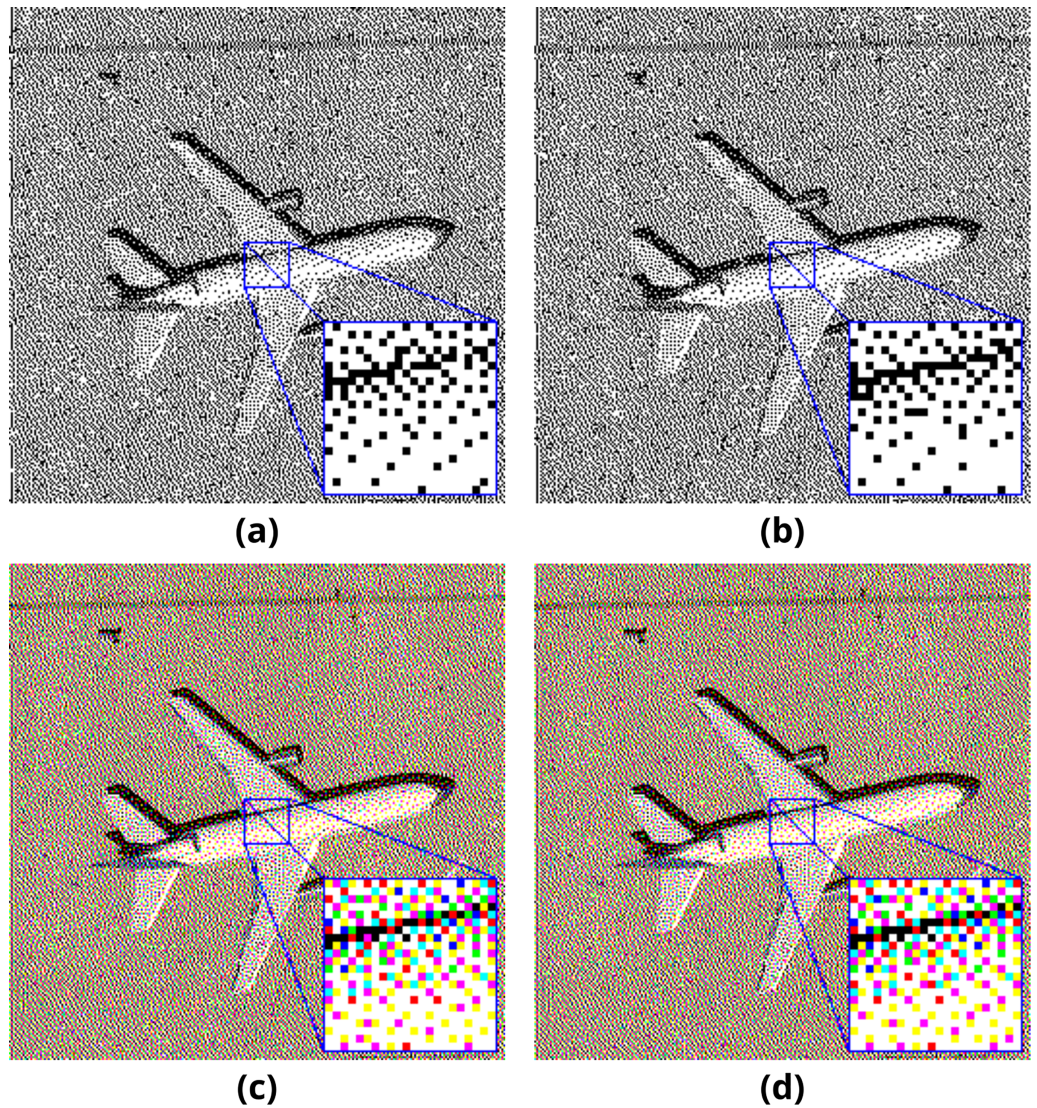


Figure 6 Error diffusion method results on (A–B) binary and (C–D) color images.

Full-size  DOI: [10.7717/peerj-cs.1062/fig-6](https://doi.org/10.7717/peerj-cs.1062/fig-6)

$$I_{ht}(R, i, j) = \begin{cases} 0, & \text{if } bit = 0 \\ 1, & \text{if } bit = 1 \end{cases} \quad (3)$$

Sample pairs of binary and color output images generated with the mentioned method are presented in Fig. 6. The length of the embedded payload is 2,048 bytes.

PAYLOAD EXTRACTION

The hiding methods explained earlier scatter each payload bit to a randomly chosen output image in a fashion similar to the secret sharing methodology, so that the whole payload is rendered inaccessible without access to all output images. These bits are embedded in such a way that the pattern or color at the determined position for each bit is represented

Algorithm 2 Payload extraction on pattern based carriers.**Input:** a set of stego media (SHARES)**Output:** extracted plaintext payload

```

1: procedure PAYLOADEXTRACTION(SHARES)
2:   let bitString be empty string
3:   for i = 1 → SHARESany.height do
4:     for j = 1 → SHARESany.width do
5:       for k = 1 → SHARESany.channels do
6:         patterns ← ∅
7:         for l = 1 → SHARES.length do
8:           patternsl,k ← SHARESl,i,j,k
9:         end for
10:        if count(unique(patterns)) = 2 then
11:          if count(prev(patterns)) > count(next(patterns)) then
12:            bitString += "1"
13:          else
14:            bitString += "0"
15:          end if
16:        end if
17:      end for
18:    end for
19:  end for
20:  return ASCII(bitString)
21: end procedure

```

- ▶ collect the patterns at the same
- ▶ (*i*, *j*) location from all images
- ▶ extract a 1 or 0 bit
- ▶ if a different pattern
- ▶ is detected in the
- ▶ set of previously
- ▶ collected patterns

differently in the chosen output image than the rest of the images; so whenever a bit is hidden, an irregularity among the outputs appears. The extraction algorithms (Algorithms 2 and 3) operate by seeking these irregularities among all provided output images (SHARES). If all the patterns (for pattern-based carriers) or colors (for error diffusion carriers) at the same coordinates have been found to be the same, it is assumed that no bits were hidden at that position, and that position is skipped without performing any further operations. However, if an irregularity is detected, a single 0 or 1 bit is extracted from that position depending on the visual relationship of the outlier and regular media, and the extracted bit is appended to a bit string. When this operation is over, the obtained bit string is converted into ASCII characters to reveal the payload.

As explained previously, all produced outputs for a given payload must be available for successful extraction. Otherwise, missing outputs cause a cascaded shift in extracted bits, resulting in illegible outputs (Fig. 7).

Algorithm 3 Payload extraction on error diffusion based carriers.**Input:** a set of stego media (SHARES)**Output:** extracted plaintext payload

```
1: procedure PAYLOADEXTRACTION(SHARES)
2:   let bitString be empty string
3:   for i = 1 → SHARESany.height do
4:     for j = 1 → SHARESany.width do
5:       for k = 1 → SHARESany.channels do
6:         pixels ← ∅
7:         for l = 1 → SHARES.length do           ► collect the pixels at the same
8:           pixelsl,k ← SHARESl,i,j,k           ► (i, j) location from all images
9:         end for
10:        if count(unique(pixels)) = 2 then       ► extract a 1 or 0 bit
11:          if count(black(pixels)) > count(white(pixels)) then ► if a different pixel
12:            bitString += "1"                    ► is detected in the
13:          else                                     ► set of previously
14:            bitString += "0"                    ► collected pixels
15:          end if
16:        end if
17:      end for
18:    end for
19:  end for
20:  return ASCII(bitString)
21: end procedure
```

```
N=1, Output=' '
N=2, Output='yyyyyyyyyyyyyyyyyyyyyyyyyyyyyy... '
N=3, Output='·TĐgš'Ùè}AXQtNé`Yh³Dê,*'Ù... '
N=4, Output='[êÈèS'£}úÆdÃx½='¹=... '
N=5, Output='Lb¹}9?N¼è}4±¼,¶f... '
N=6, Output='Lβ$@Cò=Oòzt*³höib8ß'... '
N=7, Output='LoÈ¹AÈ{Hd]¥ü@c¥Ð-~Ý[Z... '
N=8, Output='Lorem ipsum dolor sit amet... '
```

Figure 7 Outputs of payload extraction attempts with different numbers of inputs.Full-size  DOI: 10.7717/peerj-cs.1062/fig-7

RESULTS

In order to obtain results from the methods mentioned above, several tests that consist of embedding payloads of different lengths into different cover media have been conducted. The chosen cover media are airplane80, beach09, and forest22 images (Fig. 8) from UC Merced Land Use Dataset (Yang & Newsam, 2010).

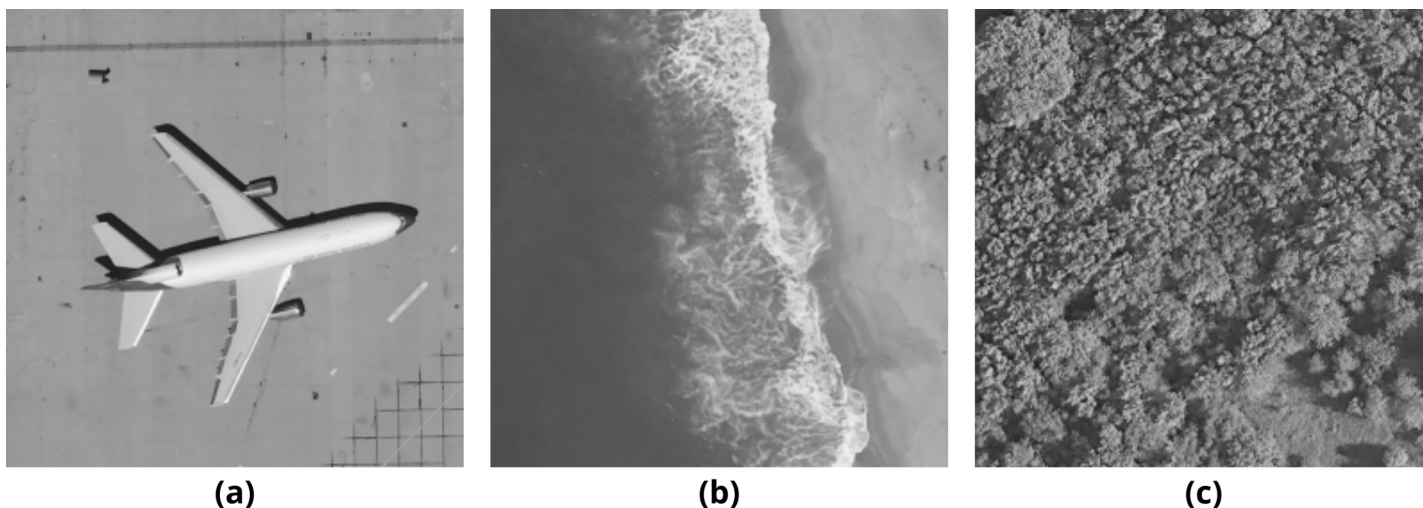


Figure 8 Three test images from UC Merced Land Use Dataset. (A) airplane80, (B) beach09, (C) forest22.

Full-size DOI: 10.7717/peerj-cs.1062/fig-8

According to the previously explained payload hiding methods, the maximum payload capacity of these images is calculated at about 8,000 bytes for grayscale carriers and 24,000 bytes for color carriers. Multiple payloads have been generated using the Lipsum generator¹, which are large enough to fill 25% and 50% of the binary outputs of chosen images. In order to observe the effects of the payloads on different images, the same payloads are used for both binary and color halftone images. These tests have been repeated three times to produce separate sets of 4, 8, and 12 output images, respectively. The output images created during these tests are shared online² for detailed inspection. In order to obtain a better quality assessment of the proposed method, the outputs obtained from these tests have been evaluated both objectively and subjectively.

Objective testing

For objective testing, SNR (Eq. (4)), PSNR (Eq. (5)) (Salomon, Motta & Bryant, 2007) and structural similarity (SSIM) (Eq. (6)) (Wang et al., 2004) values of each individual output for a selected test image and payload have been calculated. SNR and PSNR focus on the effects of added noise on the quality of the signal between two images (regular and stego media in our case), while SSIM focuses on perceptual differences between these images according to three key factors: luminance l , contrast c , and structure s . These metrics are used to compare modified digital images with their original counterparts to measure the differences between them and to evaluate the overall quality of the modified image.

$$SNR = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right) \quad (4)$$

¹ Lorem Ipsum—All the facts—Lipsum generator; <https://www.lipsum.com/>

² Halftone Steganography Results—Efe ÇİFTÇİ; <http://academic.cankaya.edu.tr/~efeciftci/halftone-stego/>

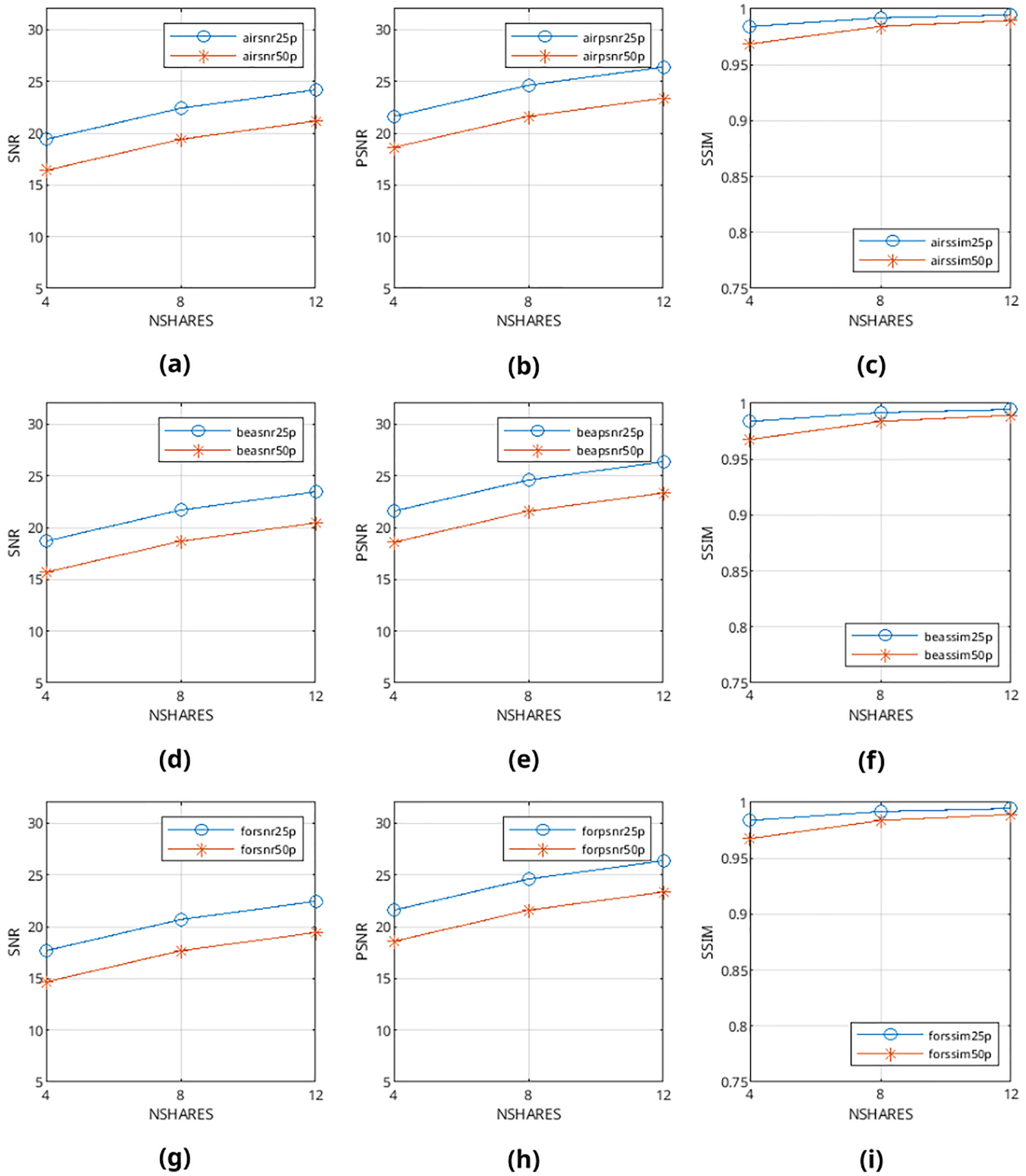


Figure 9 (A–I) SNR, PSNR and SSIM values of pattern-generated binary halftone images.

Full-size DOI: 10.7717/peerj-cs.1062/fig-9

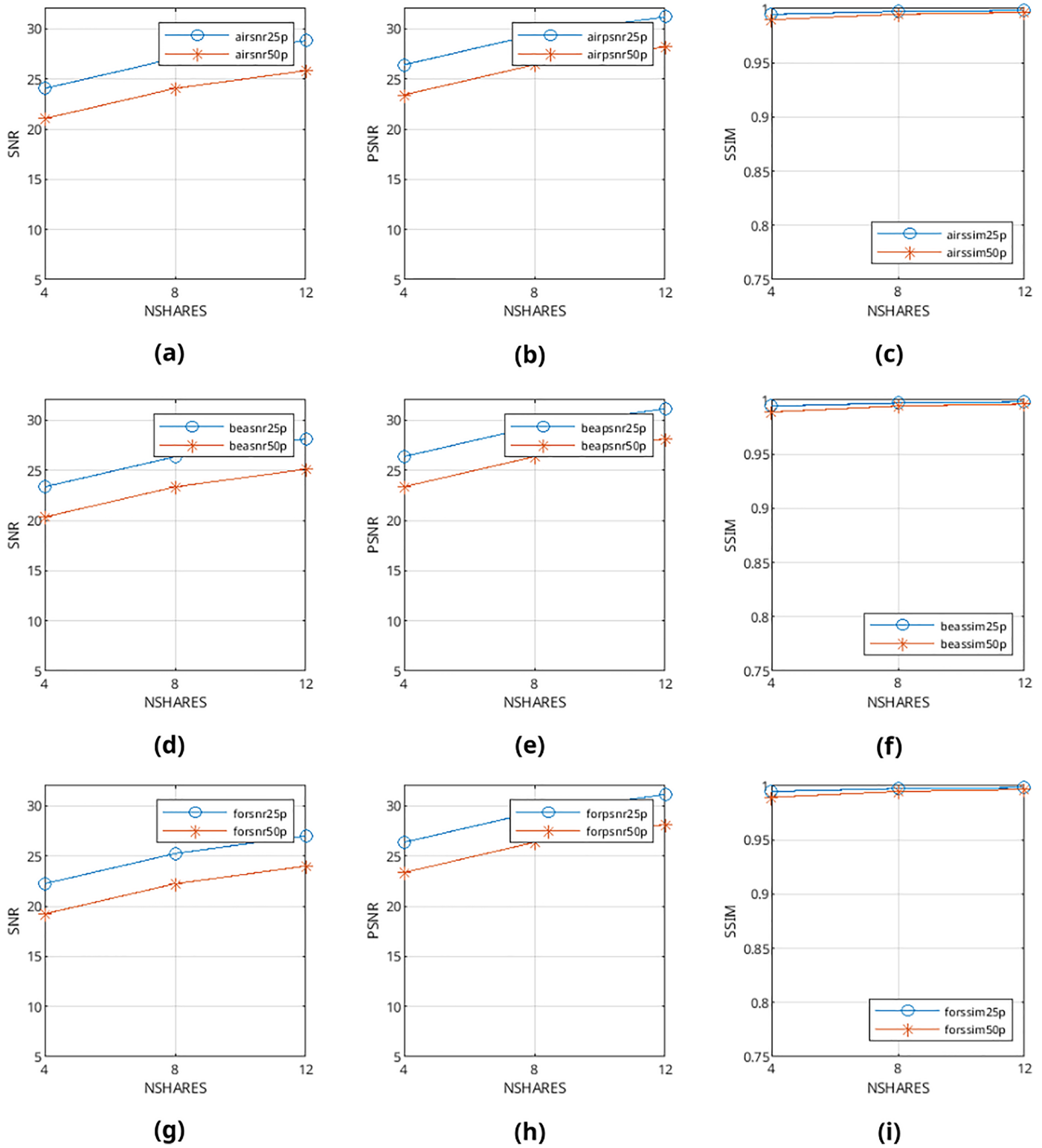


Figure 10 (A–I) SNR, PSNR and SSIM values of pattern-generated color halftone images.

Full-size DOI: 10.7717/peerj-cs.1062/fig-10

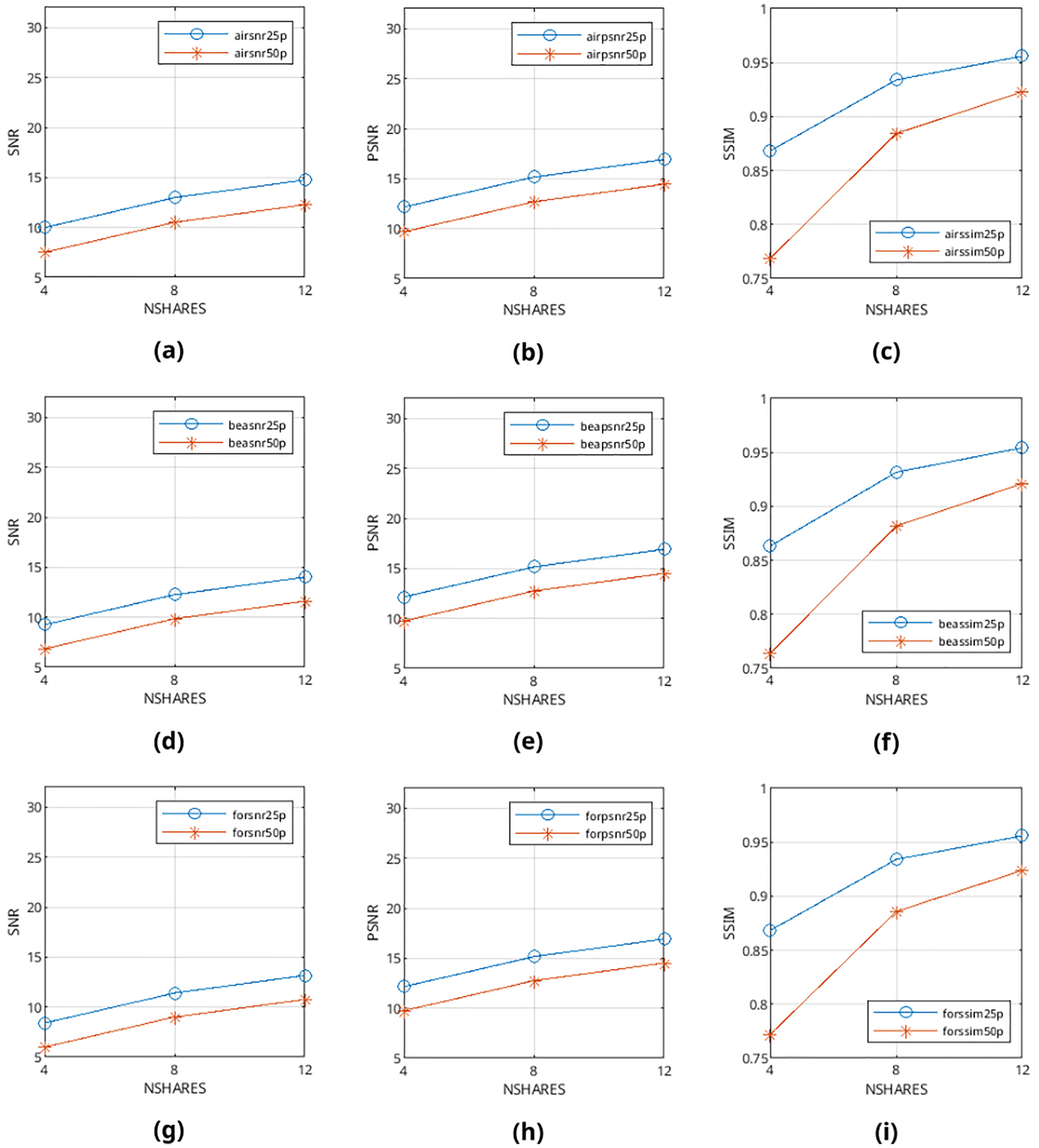


Figure 11 (A-I) SNR, PSNR and SSIM values of error diffusion-generated binary halftone images.

Full-size  DOI: 10.7717/peerj-cs.1062/fig-11

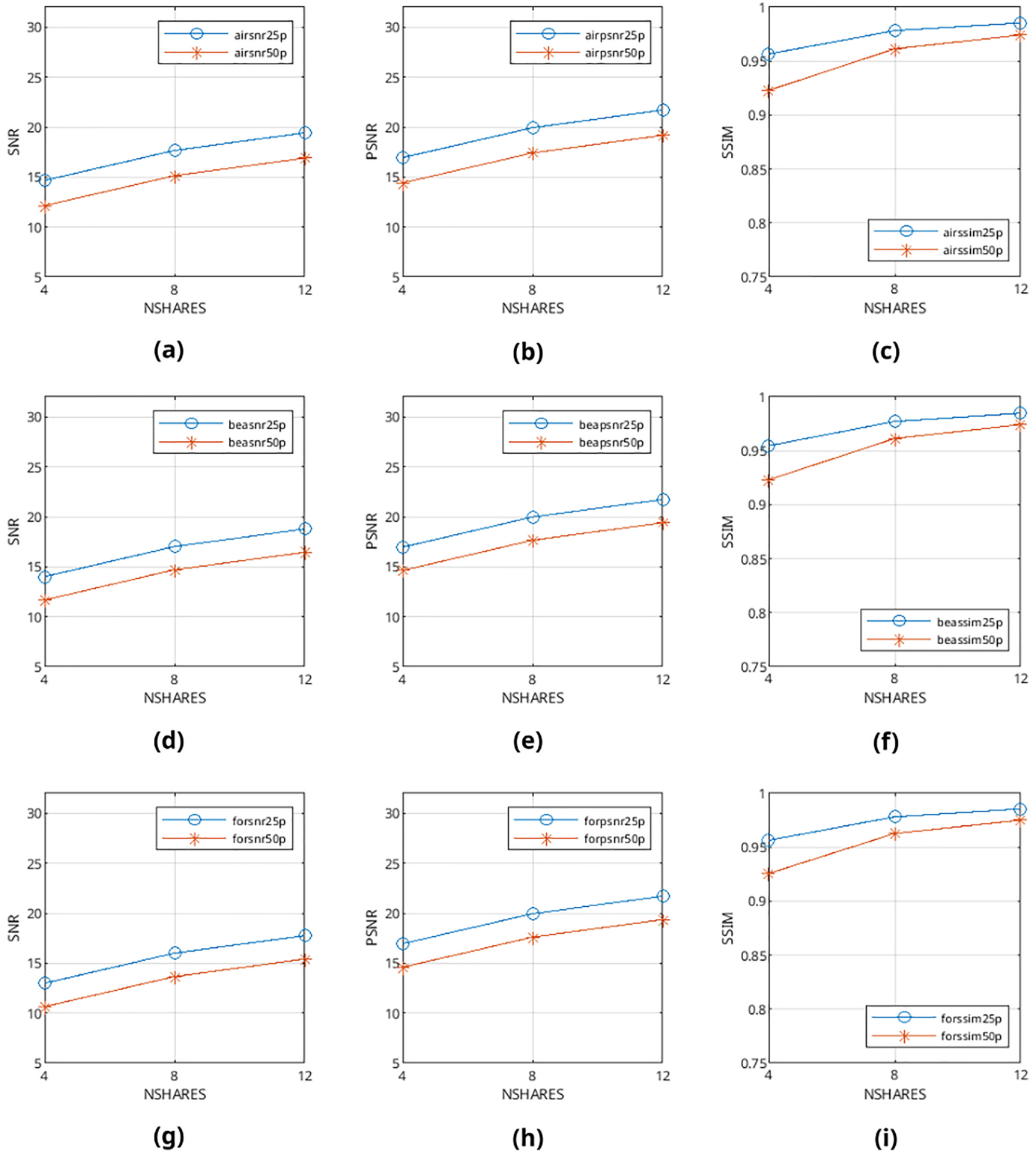


Figure 12 (A-I) SNR, PSNR and SSIM values of error diffusion-generated color halftone images.

Full-size  DOI: 10.7717/peerj-cs.1062/fig-12

Table 1 SNR, PSNR and SSIM values of pattern-generated binary halftone images.

NSHARE	%	airplane80			beach09			forest22		
		SNR	PSNR	SSIM	SNR	PSNR	SSIM	SNR	PSNR	SSIM
4	25	19.40	21.62	0.98	18.68	21.58	0.98	17.67	21.58	0.98
	50	16.39	18.61	0.96	15.67	18.57	0.96	14.66	18.57	0.96
8	25	22.41	24.63	0.99	21.69	24.59	0.99	20.68	24.59	0.99
	50	19.41	21.62	0.98	18.68	21.58	0.98	17.67	21.58	0.98
12	25	24.17	26.39	0.99	23.46	26.35	0.99	22.44	26.35	0.99
	50	21.17	23.39	0.98	20.44	23.34	0.98	19.43	23.34	0.98

Table 2 SNR, PSNR and SSIM values of pattern-generated color halftone images.

NSHARE	%	airplane80			beach09			forest22		
		SNR	PSNR	SSIM	SNR	PSNR	SSIM	SNR	PSNR	SSIM
4	25	24.04	26.40	0.99	23.34	26.35	0.99	22.23	26.35	0.99
	50	21.04	23.40	0.98	20.33	23.34	0.98	19.22	23.34	0.98
8	25	27.06	29.41	0.99	26.35	29.36	0.99	25.24	29.36	0.99
	50	24.06	26.41	0.99	23.34	26.35	0.99	22.23	26.35	0.99
12	25	28.82	31.17	0.99	28.11	31.13	0.99	27.00	31.12	0.99
	50	25.82	28.17	0.99	25.10	28.12	0.99	23.99	28.11	0.99

Table 3 SNR, PSNR and SSIM values of error diffusion-generated binary halftone images.

NSHARE	%	airplane80			beach09			forest22		
		SNR	PSNR	SSIM	SNR	PSNR	SSIM	SNR	PSNR	SSIM
4	25	9.97	12.14	0.86	9.24	12.13	0.86	8.39	12.14	0.86
	50	7.49	9.66	0.76	6.83	9.72	0.76	5.98	9.74	0.77
8	25	12.98	15.15	0.93	12.25	15.14	0.93	11.40	15.15	0.93
	50	10.50	12.67	0.88	9.84	12.73	0.88	9.00	12.75	0.88
12	25	14.74	16.91	0.95	14.01	16.90	0.95	13.16	16.91	0.95
	50	12.27	14.43	0.92	11.60	14.49	0.92	10.76	14.51	0.92

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right) \quad (5)$$

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (6)$$

Mean values are calculated from the values obtained from metrics mentioned above and presented in Figs. 9–12 and in Tables 1–4. The legends in these figures contain abbreviations of the carrier (*i.e.*, “air” for airplane80, “bea” for beach09, “for” for forest22), the metric (*i.e.*, snr, psnr, ssim), and the percentage of cover media the payloads are filling

Table 4 SNR, PSNR and SSIM values of error diffusion-generated color halftone images.

NSHARE	%	airplane80			beach09			forest22		
		SNR	PSNR	SSIM	SNR	PSNR	SSIM	SNR	PSNR	SSIM
4	25	14.66	16.94	0.95	14.00	16.95	0.95	12.99	16.93	0.95
	50	12.13	14.41	0.92	11.66	14.62	0.92	10.64	14.58	0.92
8	25	17.66	19.94	0.97	17.02	19.97	0.97	16.00	19.95	0.97
	50	15.13	17.41	0.96	14.69	17.65	0.96	13.65	17.60	0.96
12	25	19.42	21.70	0.98	18.77	21.72	0.98	17.75	21.70	0.98
	50	16.90	19.17	0.97	16.44	19.40	0.97	15.41	19.36	0.97

Table 5 Results of subjective testing.

	Pattern-based binary			Error diffusion binary		
	airplane80	beach09	forest22	airplane80	beach09	forest22
No differences	26%	25%	41%	12%	18%	26%
Few differences	54%	61%	47%	45%	35%	49%
Many differences	20%	14%	12%	43%	47%	25%
	Pattern-based color			Error diffusion color		
	airplane80	beach09	forest22	airplane80	beach09	forest22
No differences	49%	52%	56%	45%	40%	48%
Few differences	42%	39%	39%	41%	45%	45%
Many differences	8%	9%	5%	14%	15%	7%

(i.e., 25p, 50p). The x and y axes represent the number of shares generated (i.e., 4, 8, 12) and calculated SNR, PSNR, and SSIM values, respectively.

Subjective testing

In order to gain subjective results alongside the results discussed in the previous section, a survey was conducted on a group of 95 people. In this survey, attendees were presented with 12 pairs of images that consisted of regular and stego versions of the test images and were asked to answer how much difference they could detect between both versions of all pairs at first glance. Table 5 shows the results gained from the mentioned survey.

Comparison with existing methods and safety tests

Since the proposed method uses halftone images as carriers, none of the existing popular LSB or similar steganalysis methods (e.g., offered by tools such as StegExpose³) can produce reliable estimations about the payload even when the outputs are converted back to pseudo-grayscale *via* a gaussian filter. Due to the lack of steganalysis methods in the known literature that aim for halftone plaintext carriers, an alternative attack method has been implemented instead. In this method, the produced outputs are tested for resistance against extraction attempts with missing shares.

³ StegExpose; <https://github.com/b3dk7/StegExpose>



Figure 13 Extracted bytes from forest22 image using (A) three of four shares, (B) seven of eight shares, (C) 11 of 12 shares. Full-size DOI: 10.7717/peerj-cs.1062/fig-13

Since the algorithm hides the payload in bits, we have previously explained that a cascade extraction error is expected to occur even when a single output image is missing; unintelligible characters will be expected in these cases instead. In order to prove this is true for all attempts, multiple extraction attempts have been performed on the output images created during objective tests. Each extraction attempt started with one share, and the number of provided shares was increased until all shares were present (e.g., 1 of 8, until 8 of 8). As a result of these tests, it has been observed that unless all shares were present during tests, the maximum length of coincidentally revealed and intelligible pieces of the payload is always less than 2% of the total length of the payload. As an example, extracted bytes of the same payload from the forest22 image with various shares have been presented in Fig. 13.

DISCUSSION

Conduction of objective and subjective tests has made the evaluation of the quality of the method in different aspects possible. One common finding obtained from both evaluations is that the overall quality of color stego images is higher than their binary counterparts. On the other hand, objective and subjective evaluations produced several different results in different aspects.

From Figs. 9–12, it can be seen that the quality of produced stego output images increases when the length of payload decreases, the number of produced output images increases, or color images are used instead of binary images. Table 5 shows that the visibility and detection risk of the payload is lower in color images and images generated with patterns.

The values calculated from quality assessment methods in objective evaluation are completely consistent with each other; there are no outlier cases such as shorter payloads causing low scores even once. Also, it has been found that heterogeneous images generally scored better than homogenous images. This finding is supported by subjective

evaluation results as well: answers from the participants clearly show that the percentage of detected differences is always lesser in heterogeneous images. From this objective evaluation result, it can be deduced that cover images with large regions of heterogeneous textures (such as forest22) prove to be better cover images for our method.

As an interesting finding, most of the participants scored images generated with patterns higher than the ones generated with the error diffusion method. This finding is also present in objective results: metric scores of pattern-generated carriers are higher than their error diffusion-based counterparts. We believe the reason behind this is because alterations happening in patterns affect only one pixel in a 3×3 group of nine pixels, but they may affect every single pixel directly in images generated with error diffusion methods; the objective dissimilarity and subjective visibility of payload are higher in these images. From this result, it can be deduced that halftone images generated from patterns prove to be better cover images than images generated from error diffusion methods for our method.

When tested for payload extraction, it has been observed that when a large number of shares are present (*e.g.*, 11 of 12), short sequences of letters from the payload may appear in random positions of the extracted text. However, they are never long enough to reveal meaningful information. Since the proposed methods are nondeterministic and produce new different sets of output images on each execution, the tests have been repeated multiple times to verify that the exposed information is never long enough to reveal a meaningful payload.

CONCLUSIONS

In this article, a novel steganography method that operates on halftone cover images is proposed and demonstrated. In general, halftone images offer a cheaper alternative to grayscale images in aspects of being resource effective both in printed and digital media. They are also immune to numerous LSB steganalysis methods that target grayscale and color images.

The method hides given plaintext payloads and distributes them on multiple outputs. The secret sharing approach has proven to be an effective aid for both decreasing the detectability risk of hidden payloads and also preventing attackers from successful payload extraction.

The method has been tested with different test images and with plaintext payloads of different lengths. The experimental results show that our method provides high embedding capacity for any given cover image. Furthermore, results obtained from both objective and subjective measures show that our method can produce outputs mostly indistinguishable from their unmodified counterparts and perform better on pattern-generated cover images. The results obtained from objective, subjective, and payload extraction tests indicate that the proposed method is suitable for real-life use.

This study mainly focuses on the presentation of the proposed method; the robustness of the method against steganalysis attacks is out of scope and has not been thoroughly tested. Also, a comparative analysis of the proposed algorithm with other algorithms could

not be included as the implementation, inputs, or outputs of discovered methods are different from the proposed methods.

In order to further reduce detectability on halftone cover media with less spatial heterogeneous features, it is planned to implement a mechanism that proposes the maximum safest payload length according to spatial features of chosen cover media. Furthermore, as the proposed extraction algorithm currently operates only on digital carriers, it is also planned to improve the method for successful extraction from printed carriers as well.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

The authors received no funding for this work.

Competing Interests

The authors declare that they have no competing interests.

Author Contributions

- Efe Çiftci conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Emre Sümer analyzed the data, authored or reviewed drafts of the article, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The codes for the proposed message hiding and extraction methods were developed and tested in MATLAB 2021a and are available in the [Supplemental Files](#).

Sample images in the [Supplemental Files](#) are chosen from UC Merced Land Use Dataset: <http://weegee.vision.ucmerced.edu/datasets/landuse.html>.

Supplemental Information

Supplemental information for this article can be found online at <http://dx.doi.org/10.7717/peerj-cs.1062#supplemental-information>.

REFERENCES

- Al-Amri SS, Kalyankar NV, Khamitkar SD. 2010. Image segmentation by using threshold techniques. *ArXiv preprint*. DOI 10.48550/arXiv.1005.4020.
- Analoui M, Allebach JP. 1992. Model-based halftoning using direct binary search. In: *Human Vision, Visual Processing, and Digital Display III*. Vol. 1666. International Society for Optics and Photonics, 96–108.
- Askari N, Heys HM, Moloney C. 2013. An extended visual cryptography scheme without pixel expansion for halftone images. In: *2013 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. Piscataway: IEEE, 1–6.

- Chan TC, Chen KS. 1998.** Method and system for halftoning by tiling subsets of a threshold array over portions of an image. US Patent 5,761,347. Available at <https://patents.google.com/patent/US5761347>.
- Cheddad A, Condell J, Curran K, Mc Kevitt P. 2010.** Digital image steganography: survey and analysis of current methods. *Signal Processing* **90(3)**:727–752 DOI [10.1016/j.sigpro.2009.08.010](https://doi.org/10.1016/j.sigpro.2009.08.010).
- Chen Y-F, Chan Y-K, Huang C-C, Tsai M-H, Chu Y-P. 2007.** A multiple-level visual secret-sharing scheme without image size expansion. *Information Sciences* **177(21)**:4696–4710 DOI [10.1016/j.ins.2007.05.011](https://doi.org/10.1016/j.ins.2007.05.011).
- Cruz L, Patrão B, Gonçalves N, Diamanti O, Vaxman A. 2018.** Halftone pattern: A new steganographic approach. In: *Eurographics (Short Papers)*. 21–24.
- Floyd RW. 1976.** An adaptive algorithm for spatial gray-scale. *Proceedings of the Society of Information Display* **17**:75–77.
- Fu MS, Au OC. 2001a.** Data hiding in halftone images by stochastic error diffusion. In: *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*. Vol. 3. Piscataway: IEEE, 1965–1968.
- Fu MS, Au OC. 2001b.** Halftone image data hiding with intensity selection and connection selection. *Signal Processing: Image Communication* **16(10)**:909–930 DOI [10.1016/S0923-5965\(00\)00052-7](https://doi.org/10.1016/S0923-5965(00)00052-7).
- Fu MS, Au OC. 2003.** Steganography in halftone images: conjugate error diffusion. *Signal Processing* **83(10)**:2171–2178 DOI [10.1016/S0165-1684\(03\)00173-7](https://doi.org/10.1016/S0165-1684(03)00173-7).
- Hsiao J-Y, Chan K-F, Chang JM. 2009.** Block-based reversible data embedding. *Signal Processing* **89(4)**:556–569 DOI [10.1016/j.sigpro.2008.10.018](https://doi.org/10.1016/j.sigpro.2008.10.018).
- Huang Z-K, Chau K-W. 2008.** A new image thresholding method based on Gaussian mixture model. *Applied Mathematics and Computation* **205(2)**:899–907 DOI [10.1016/j.amc.2008.05.130](https://doi.org/10.1016/j.amc.2008.05.130).
- Jarvis JF, Judice CN, Ninke W. 1976.** A survey of techniques for the display of continuous tone pictures on bilevel displays. *Computer Graphics and Image Processing* **5(1)**:13–40 DOI [10.1016/S0146-664X\(76\)80003-2](https://doi.org/10.1016/S0146-664X(76)80003-2).
- Jiang M, Wong EK, Memon N, Wu X. 2005.** Steganalysis of halftone images. In: *Proceedings. (ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005*. Vol. 2. IEEE, ii/793–ii/796.
- John Blesswin A, Raj C, Sukumaran R, Selva Mary G. 2020.** Enhanced semantic visual secret sharing scheme for the secure image communication. *Multimedia Tools and Applications* **79(23–24)**:17057–17079 DOI [10.1007/s11042-019-7535-2](https://doi.org/10.1007/s11042-019-7535-2).
- Juneja M, Sandhu PS. 2013.** An improved LSB based steganography technique for RGB color images. *International Journal of Computer and Communication Engineering* **2(4)**:513 DOI [10.7763/IJCCE.2013.V2.238](https://doi.org/10.7763/IJCCE.2013.V2.238).
- Juneja M, Sandhu PS. 2014.** Improved LSB based steganography techniques for color images in spatial domain. *International Journal of Network Security* **16(6)**:452–462 DOI [10.7763/IJCCE.2013.V2.238](https://doi.org/10.7763/IJCCE.2013.V2.238).
- Knuth DE. 1987.** Digital halftones by dot diffusion. *ACM Transactions on Graphics (TOG)* **6(4)**:245–273 DOI [10.1145/35039.35040](https://doi.org/10.1145/35039.35040).
- Kordov K, Zhelezov S. 2021.** Steganography in color images with random order of pixel selection and encrypted text message embedding. *PeerJ Computer Science* **7**:e380 DOI [10.7717/peerj-cs.380](https://doi.org/10.7717/peerj-cs.380).
- Lau DL, Arce GR. 2001.** *Modern digital halftoning*. Boca Raton: CRC Press.

- Lu W, Xue Y, Yeung Y, Liu H, Huang J, Shi Y-Q. 2019.** Secure halftone image steganography based on pixel density transition. *IEEE Transactions on Dependable and Secure Computing* **18**(3):1137–1149 DOI [10.1109/TDSC.2019.2933621](https://doi.org/10.1109/TDSC.2019.2933621).
- Luo J, Lin C, Zeng L, Liang J, Lu W. 2021.** Halftone image steganalysis by reconstructing grayscale image. In: *International Conference on Artificial Intelligence and Security*. Springer, 412–423.
- Mowafi M, Oudat O, Taqieddin E, Banimelhem O. 2019.** Image steganography using YCbCr color space and matrix pattern. In: *2019 2nd International Conference on Signal Processing and Information Security (ICSPIS)*. Piscataway: IEEE, 1–4.
- Naor M, Shamir A. 1995.** Visual cryptography. In: De Santis A, ed. *Advances in Cryptology—EUROCRYPT’94*. Vol. 950. Berlin, Heidelberg: Springer. Lecture Notes in Computer Science.
- Nilizadeh A, Mazurczyk W, Zou C, Leavens GT. 2017.** Information hiding in RGB images using an improved matrix pattern approach. In: *CVPR Workshops*. 1407–1415.
- Nilizadeh A, Nilizadeh S, Mazurczyk W, Zou C, Leavens GT. 2022.** Adaptive matrix pattern steganography on RGB images. *Journal of Cyber Security and Mobility* **11**:1–28 DOI [10.13052/jcsm2245-1439.1111](https://doi.org/10.13052/jcsm2245-1439.1111).
- Pei S-C, Guo J-M. 2003.** Data hiding in halftone images with noise-balanced error diffusion. *IEEE Signal Processing Letters* **10**(12):349–351 DOI [10.1109/LSP.2003.817856](https://doi.org/10.1109/LSP.2003.817856).
- Quach T-T. 2014.** Extracting hidden messages in steganographic images. *Digital Investigation* **11**: S40–S45 DOI [10.1016/j.diin.2014.05.003](https://doi.org/10.1016/j.diin.2014.05.003).
- Rosen J, Javidi B. 2001.** Hidden images in halftone pictures. *Applied Optics* **40**(20):3346–3353 DOI [10.1364/AO.40.003346](https://doi.org/10.1364/AO.40.003346).
- Salomon D, Motta G, Bryant D. 2007.** *Data compression: the complete reference*. London: Springer. Molecular Biology Intelligence Unit.
- Sari CA, Ardiansyah G, Rachmawanto EH, Setiadi DRIM. 2019.** An improved security and message capacity using AES and Huffman coding on image steganography. *TELKOMNIKA Indonesian Journal of Electrical Engineering* **17**(5):2400–2409 DOI [10.12928/TELKOMNIKA.v17i5.9570](https://doi.org/10.12928/TELKOMNIKA.v17i5.9570).
- Sharma K, Aggarwal A, Singhanian T, Gupta D, Khanna A. 2019.** Hiding data in images using cryptography and deep neural network. *ArXiv preprint*. DOI [10.48550/arXiv.1912.10413](https://doi.org/10.48550/arXiv.1912.10413).
- Sharma N, Batra U. 2021.** An enhanced Huffman-PSO based image optimization algorithm for image steganography. *Genetic Programming and Evolvable Machines* **22**(2):189–205 DOI [10.1007/s10710-020-09396-z](https://doi.org/10.1007/s10710-020-09396-z).
- Shiau J-N, Fan Z. 1994.** Method for quantization gray level pixel data with extended distribution set. US Patent 5,353,127. Available at <https://patents.google.com/patent/US5353127A/en>.
- Shreiner D, Group B. 2009.** *OpenGL programming guide: the official guide to learning OpenGL, versions 3.0 and 3.1*. London: Pearson Education.
- Sutaone M, Khandare M. 2008.** Image based steganography using LSB insertion technique. In: *2008 IET International Conference on Wireless, Mobile and Multimedia Networks*. London: IET, 146–151.
- Ulichney R. 1987.** *Digital halftoning*. Cambridge: MIT Press.
- Wang Z, Arce GR, Di Crescenzo G. 2006.** Halftone visual cryptography via direct binary search. In: *2006 14th European Signal Processing Conference*. Piscataway: IEEE, 1–5.
- Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. 2004.** Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* **13**(4):600–612 DOI [10.1109/TIP.2003.819861](https://doi.org/10.1109/TIP.2003.819861).

- Wang S, Chung F-I, Xiong F. 2008.** A novel image thresholding method based on Parzen window estimate. *Pattern Recognition* **41(1)**:117–129 DOI [10.1016/j.patcog.2007.03.029](https://doi.org/10.1016/j.patcog.2007.03.029).
- Xue Y, Liu W, Lu W, Yeung Y, Liu X, Liu H. 2019.** Efficient halftone image steganography based on dispersion degree optimization. *Journal of Real-Time Image Processing* **16(3)**:601–609 DOI [10.1007/s11554-018-0822-8](https://doi.org/10.1007/s11554-018-0822-8).
- Yan X, Wang S, Niu X, Yang C-N. 2015.** Halftone visual cryptography with minimum auxiliary black pixels and uniform image quality. *Digital Signal Processing* **38(10)**:53–65 DOI [10.1016/j.dsp.2014.12.002](https://doi.org/10.1016/j.dsp.2014.12.002).
- Yang Y, Newsam S. 2010.** Bag-of-visual-words and spatial extensions for land-use classification. In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 270–279.
- Yu M, Yin X, Liu W, Lu W. 2021.** Secure halftone image steganography based on density preserving and distortion fusion. *Signal Processing* **188(3)**:108227 DOI [10.1016/j.sigpro.2021.108227](https://doi.org/10.1016/j.sigpro.2021.108227).
- Zhang Y. 2017.** *Image processing*. Berlin: Walter de Gruyter GmbH & Co KG.
- Zhou X, Gong W, Fu W, Jin L. 2016.** An improved method for LSB based color image steganography combined with cryptography. In: *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. Piscataway: IEEE, 1–4.