BAŞKENT UNIVERSITY
INSTITUTE OF SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING
DOCTOR OF PHILOSOPHY IN COMPUTER ENGINEERING

ENCRYPTION AND MULTI-SHARE-BASED STEGANOGRAPHY METHODS ON IMAGES WITH LOW SPECTRAL RESOLUTION

BY

EFE ÇİFTCİ

DOCTOR OF PHILOSOPHY THESIS

ANKARA – 2023

BAŞKENT UNIVERSITY
INSTITUTE OF SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING
DOCTOR OF PHILOSOPHY IN COMPUTER ENGINEERING

ENCRYPTION AND MULTI-SHARE-BASED STEGANOGRAPHY METHODS ON IMAGES WITH LOW SPECTRAL RESOLUTION

BY

EFE ÇİFTCİ

DOCTOR OF PHILOSOPHY THESIS

ADVISOR

ASSOC. PROF. DR. EMRE SÜMER

ANKARA – 2023

BAŞKENT UNIVERSITY

INSTITUTE OF SCIENCE AND ENGINEERING

This study, which was prepared by Efe ÇİFTCİ, has been approved in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY in the Computer Engineering Department by the following committee.

Date of Thesis Defense: 03 / 01 / 2023

Thesis Title: Encryption and Multi-Share-Based Steganography Methods On Images With Low Spectral Resolution

Examining Committee Members	**Signature**
Assoc. Prof. Dr. Emre SÜMER, Başkent University	………………
Assoc. Prof. Dr. İhsan Tolga MEDENİ, Ankara Yıldırım Beyazıt University	………………
Assist. Prof. Dr. Hakan TORA, Atılım University	………………
Assist. Prof. Dr. Mehmet DİKMEN, Başkent University	………………
Assist. Prof. Dr. Didem ÖLÇER, Başkent University	………………

APPROVAL
Prof. Dr. Faruk ELALDI
Director, Institute of Science and Engineering
Date: …. / …. / 2023

BAŞKENT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

DOKTORA TEZ ÇALIŞMASI ORİJİNALLİK RAPORU

Tarih: 05 / 01 / 2023

Öğrencinin Adı, Soyadı: Efe ÇİFTCİ

Öğrencinin Numarası: 21310019

Anabilim Dalı: Bilgisayar Mühendisliği Anabilim Dalı

Programı: Bilgisayar Mühendisliği Doktora Programı

Danışmanın Unvanı/Adı, Soyadı: Doç. Dr. Emre SÜMER

Tez Başlığı: Düşük Spektral Çözünürlüklü Görüntülerde Şifreleme ve Çok Paylılık Tabanlı Steganografi Yöntemleri

Yukarıda başlığı belirtilen Doktora tez çalışmamın; Giriş, Ana Bölümler ve Sonuç Bölümünden oluşan, toplam 46 sayfalık kısmına ilişkin, 05/01/2023 tarihinde tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı %11'dir. Uygulanan filtrelemeler:

1. Kaynakça hariç

2. Alıntılar hariç

3. Beş (5) kelimeden daha az örtüşme içeren metin kısımları hariç

"Başkent Üniversitesi Enstitüleri Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Usul ve Esaslarını" inceledim ve bu uygulama esaslarında belirtilen azami benzerlik oranlarına tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Öğrenci İmzası:

ONAY
Tarih: 05 / 01 / 2023
Doç. Dr. Emre SÜMER

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my thesis advisor Assoc. Prof. Dr. Emre SÜMER for his continuous support since the first days of this study.

I would also like to thank my thesis committee members, Assist. Prof. Dr. Hakan TORA and Assist. Prof. Dr. Mehmet DİKMEN, who supported me and shared their valuable opinions and suggestions.

I want to express my deepest gratitude to my family. I am especially grateful to my mother, Ayseli ÇİFTCİ, and my father, Mukatder ÇİFTCİ, for supporting me for my entire life and introducing me to the computer engineering profession.

Finally, I thank all my colleagues and friends who have shared their views and ideas with me throughout this thesis study.

ABSTRACT

Efe ÇİFTCİ

ENCRYPTION AND MULTI-SHARE-BASED STEGANOGRAPHY METHODS ON IMAGES WITH LOW SPECTRAL RESOLUTION

Başkent University Institute of Science and Engineering

Department of Computer Engineering

2023

Steganography is the name given to secret communication methods that third parties cannot detect. This secret communication is performed by hiding the secret information to be transmitted on a carrier medium so that the carrier does not raise any suspicions. Steganography science, of which many examples can be presented from the past to the present, has gained new application areas with the development of digital technologies. This thesis aims to develop new steganography methods that hide secret messages in plain text format on binary images, which have a lower spectral resolution when compared to color or grayscale images, used in digital devices as carriers. It has been observed that all implemented methods can successfully hide considerable lengths of plaintext payloads on binary images generated by both thresholding and halftoning methods, and this finding has been reinforced with conducted objective and subjective evaluations.

KEYWORDS: Steganography, Image Processing, Binary Image, Halftone Image

ÖZET

Efe ÇİFTCİ

DÜŞÜK SPEKTRAL ÇÖZÜNÜRLÜKLÜ GÖRÜNTÜLERDE ŞİFRELEME VE ÇOK PAYLILIK TABANLI STEGANOGRAFİ YÖNTEMLERİ

Başkent Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

2023

Steganografi, üçüncü şahıslar tarafından tespit edilmeyecek şekilde gizli iletişim kurma yöntemlerine verilen isimdir. Bu gizli iletişim, iletilmek istenen gizli bilginin şüphe uyandırmayacak bir şekilde bir taşıyıcı ortamın üzerine gizlenmesiyle gerçekleşir. Geçmişten günümüze bir çok örneği sunulabilen steganografi bilimi, dijital teknolojilerin gelişmesiyle yeni uygulama alanları kazanmıştır. Bu tezin amacı, düz metin biçimindeki gizli mesajları, taşıyıcı olarak dijital cihazlarda kullanılan renkli veya gri tonlu görüntülere kıyasla daha düşük spektral çözünürlüğe sahip ikili görüntüler üzerine gizleyecek olan yeni steganografi yöntemlerinin geliştirilmesidir. Geliştirilen tüm yöntemlerin hem eşikleme, hem de yarıtonlama yöntemleriyle üretilen ikili görüntülere büyük uzunluklarda düz metin türünde veriyi başarıyla gizleyebildikleri görülmüş ve yapılan objektif ile subjektif değerlendirmelerle de bu bulgu pekiştirilmiştir.

ANAHTAR KELİMELER: Steganografi, Görüntü İşleme, İkili Görüntü, Yarıtonlu Görüntü

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

RGB	Red-Green-Blue
bpp	bits-per-pixel
HVS	Human Visual System
LSB	Least Significant Bit
KB	Kilobyte
XOR	Exclusive OR
ASCII	American Standard Code for Information Interchange
SNR	Signal-to-noise ratio
PSNR	Peak signal-to-noise ratio
SSIM	Structural similarity index

1. INTRODUCTION

Since the first day human beings started written communication, secure communication has always been a demand for protecting critical messages from unwanted people. Rooting from this demand, multiple ways to ensure the security of such messages have been born over millennia. Encryption, for example, has been a widely chosen method for secure communication since its earliest recorded examples from ancient civilizations [1]. Encryption provides a good level of security by rendering the message unreadable without the correct encryption key. However, it is usually not concerned with the secrecy of the message itself, and encrypted texts have often been relayed visible to the human eye.

Steganography is the solution to secret communication. As the combination of two Greek words ("steganos" for "covered" and "graphei" for "writing" [2]), steganography is considered both an art and a science. Steganography ensures the secrecy of critical messages by hiding them in unsuspicious-looking carriers. For example, a photograph of a father with his newborn baby hanging on the wall in the father's office may appear innocent and even pretty at first glance; however, it may secretly contain passwords of servers their father frequently connects to at work, written with invisible ink. If the father tells nobody about this secret, then most probably none of his colleagues at work would suspect the photograph has any secret writings.

With the technological advancements in digital computing, steganography has gained a new and vast domain. Nowadays, digital devices such as desktop computers and smartphones are everywhere. The user data, such as images, videos, and audio, are generated and stored in specific file formats on these devices. The structure of these files can be exploited to hide secret messages.

This thesis aims to provide insight into binary images, their structural features, and implementations of novel steganography methods that use them as carriers. Binary images are a specific type of image that can only display images with exactly two colors: black and white. Since binary images offer fewer features than grayscale or color images, the implementation of steganography methods using them as carriers is not studied as much as steganography methods using other image types as carriers.

1.1 Contribution of the Thesis

This thesis aims to provide novel digital steganography methods for hiding plaintext payloads over binary images. The developed methods can hide provided payloads into thresholded images containing graphical texts and halftone images that have been generated via patterns and error diffusion. The developed methods aim to increase the security of the payloads with additional measures such as encryption and distributing parts of them into multiple carriers.

1.2 Structure of the Thesis

In the Introduction chapter, the purpose of the thesis is explained. The second chapter explains background information regarding steganography and various digital image formats. The third chapter explains the implemented steganography and respective payload extraction methods for binary image carriers. In the fourth chapter, the conducted tests for each proposed steganography method, along with their results, are presented. In the fifth chapter, the results obtained from the tests have been commented on. Finally, in the Conclusions chapter, the purpose of this thesis and the knowledge obtained from it are summarized.

2. BACKGROUND

2.1. Steganography

The art and science of establishing secret communication are named steganography. Steganography aims to relay messages between multiple parties while keeping them safe from unwanted eyes; this is achieved by hiding the messages so that they are appropriately camouflaged within unsuspicious carriers.

A well-known theoretical example of secret communication is The Prisoners' Problem. Defined by Simmons [3], The Prisoners' Problem involves three people: two prisoners locked in separate cells in prison and a warden. The prisoners are permitted to communicate using letters, and the warden lets them exchange letters only if the information contained in them is safe. On the other hand, the prisoners must devise an escape plan; therefore, they must trick the warden by secretly embedding their escape plan in the letters (Figure 2.1).

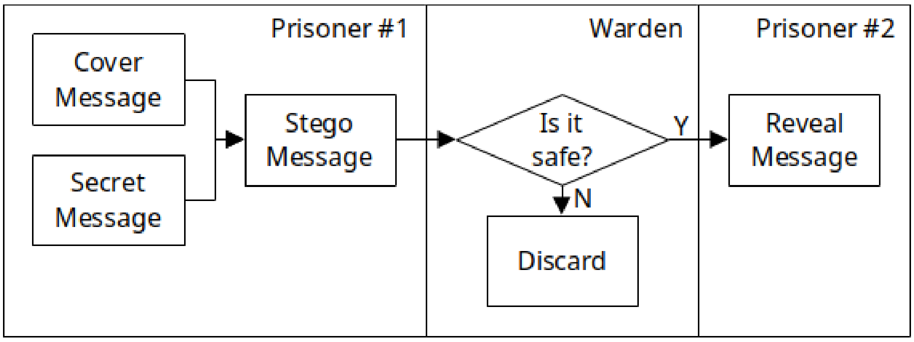


Figure 2.1 The Prisoners' Problem

Steganography belongs to a family of methods involved with information security. Cryptography ensures the payload's security by scrambling it with a key, where the key itself is shared among only the relevant people. Watermarking ensures the uniqueness of a material by embedding copyright information on it. Although all these methods are related to the security of relayed information and their purposes look similar, the requirements and approaches of these three topics are different [4] (Table 2.1).

Table 2.1 Comparison of cryptography, steganography, and watermarking

	Cryptography	**Steganography**	**Watermarking**
Purpose	Data protection	Secret communication	Copyright
Message	Plaintext	Payload	Watermark
Fails when	Deciphered	Payload is detected	Watermark is removed

One of the fundamental requirements of steganography is that the hiding method must be reversible, and hidden messages must be extractable when required so that the message's sender and the recipient can communicate. In order to successfully extract the secret message, the extraction method must be implemented closely related to the carrier media type and how the message was initially hidden.

The carriers and methods can vary greatly depending on how the communication takes place and the available technology. For example, in ancient Greece, one of the earliest known steganography methods was conceived through wax tablets. A wax tablet is a writing medium with two layers: an underlying hard layer of a wooden surface and a softer layer of opaque wax that covers the underlying surface. Usually, the messages were written on the softer layer. When the information written on the tablet is no more helpful, the same tablet could be reused with different messages simply by renewing the wax surface. Instead of writing the message on the wax surface, writing the message directly on the underlying layer and then covering it with smooth wax simply made the message invisible; only the people involved in the communication knew the existence of the message carved over the underlying surface, while everybody else saw only a regular wax tablet [5].

Again in ancient Greece, Histaeus used a slave as a messenger to relay a message to his son-in-law, but instead of sending the message visibly, the slave's hair was shaved, and the message was tattooed on his shaved scalp instead. This slave was sent to the destination after his hair grew back, and his scalp was shaved again to reveal the message after he reached the destination city [6].

An another method of steganography is achieved through the use of invisible ink through centuries. Vinegar, milk, and similar liquids, which leave no residues visible to the human eye, are used to write secret messages on papers or other surfaces. The messages written with these liquids cannot be seen with the eye when dried, but they create brownish stains when the surface they are written on is heated [7].

When moved onwards to the modern age, steganography became widely popular, especially in the 20th century. The Germans conceived microdot technology in World War II, which is the method of shrinking large amounts of information into the size of a comma, and imprinting this shrunken information using printers on paper [8].

Another method that became famous thanks to printing machines is the word/line shifting method [9]. Altering the distance between words or lines unsuspiciously to the human eye can be used for hiding secret messages across the printed document. For example, in Figure 2.2, the words on the top row have been printed with default spacing, while the highlighted letters on the bottom row have been printed with extra padding to their left to hide the secret word "lost."

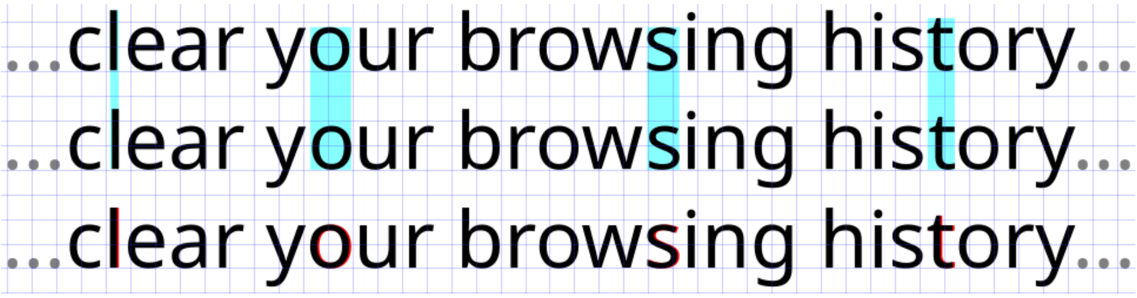

...clear your browsing history...
...clear your browsing history...
...clear your browsing history...

Figure 2.2 Word shifting method

In 1965, the Vietnamese army captured an American military officer named Jeremiah Denton as a prisoner of war. In a video recording made public by Vietnamese authorities, Jeremiah Denton was seemingly answering the questions directed at him. However, he was also blinking his eyes in Morse code to spell the word "torture" to acknowledge his situation to the public [10].

Advancements in digital computing have enabled numerous new ways of steganography in this domain. File formats such as digital images, text, audio, and video are the most preferred carriers for digital steganography. Each file type stores the data it carries in different formats, so the implementation of steganography methods varies from carrier to carrier.

2.2. Digital Images

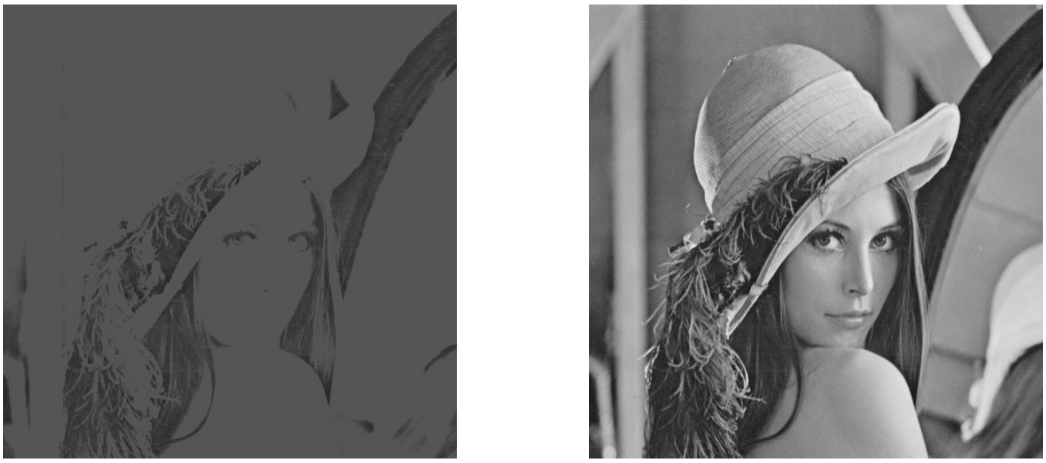
Digital images are files that are used for storing visual and graphical information. A digital image consists of two-dimensional color information stored in units named a pixel. Every pixel in an image represents a different position in the image. The resolution of an image represents how many rows and columns of pixels an image contains; for example, a 1024x768 image has a total of 786,432 pixels stored in 1024 columns and 768 rows. Bit depth, however, represents how many unique levels of color information each pixel can represent. Depending on the bit depth, digital images can be classified as color, grayscale, and binary.

In modern-day computing, a typical color image usually consists of 3 separate 8-bit channels representing red, green, and blue (RGB) colors, using 24 bits per pixel (bpp). Such images can represent a total of 2563 = 16,777,216 different colors. With an extra 8-bit alpha channel, a 32bpp color image with transparency information can be constructed. Rather than photography, this type of translucent image is preferred in application development and design. With modern-day advancements, a newer color image type that uses three channels of 10 bpp is gaining popularity. In contrast to standard RGB images, these images represent 1,073,741,824 different colors (64 times larger than 24 bpp images). This technology is commonly used by digital photography today but also gaining popularity in other areas.

Grayscale images are obtained when all color information from a color image is removed and replaced with a single gray level intensity information instead. Grayscale images can be acquired from optical sensors (e.g., infrared cameras) or generated by converting from color images. A simple color-to-gray conversion approach is to calculate the average of the sum of red, green, and blue channels (Equation 2.1); but since blue is conceived darker than green by the human visual system (HVS), outputs obtained by this method will appear darker, thus will not be visually appealing. Instead of simple averaging, more accurate and visually appealing results can be obtained via Equation 2.2 [11], where the coefficients are altered with more emphasis on the green channel and less on the blue channel. Figure 2.3 displays results obtained from both equations for comparison.

$$x=\frac{R+G+B}{3} \tag{2.1}$$

$$x=0.299\,R+0.587\,G+0.114\,B \tag{2.2}$$

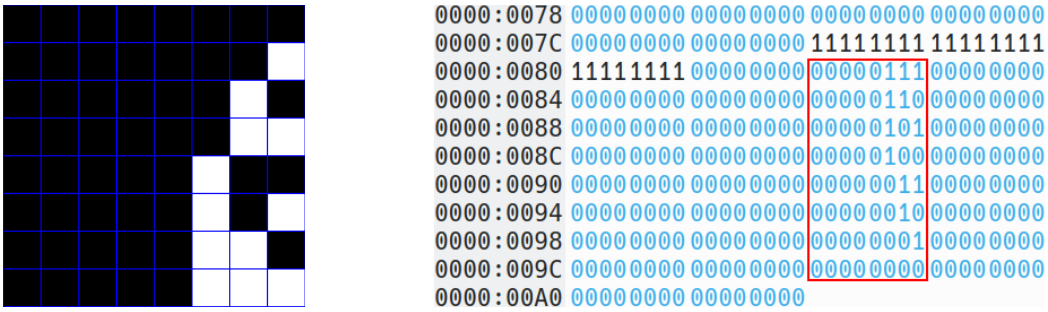


Balanced

Weighted

Figure 2.3 Grayscale images generated with different RGB coefficients

A typical grayscale image uses a single 8-bpp channel, in which each pixel can represent 256 unique intensity values. These grayscale images can further be shrunken down to create a binary image. Binary images use only one bit per pixel; thus, pixels in binary images can have only one of two values: black pixels are represented with a 0 bit, and white pixels are represented with a 1 bit. If saved in an uncompressed format (e.g., bitmap), it can be observed that each byte in such images represents eight sequential pixels in the image (Figure 2.4).



```
0000:0078 00000000 00000000 00000000 00000000
0000:007C 00000000 00000000 11111111 11111111
0000:0080 11111111 00000000 00000111 00000000
0000:0084 00000000 00000000 00000110 00000000
0000:0088 00000000 00000000 00000101 00000000
0000:008C 00000000 00000000 00000100 00000000
0000:0090 00000000 00000000 00000011 00000000
0000:0094 00000000 00000000 00000010 00000000
0000:0098 00000000 00000000 00000001 00000000
0000:009C 00000000 00000000 00000000 00000000
0000:00A0 00000000 00000000
```

Image

File

Figure 2.4 A binary image and its bitmap representation on disk

Binary images can be created by converting grayscale images into them. All gray pixels in the source image except pure black and pure white during this conversion must be mapped to either of these colors. What color each pixel will be represented as in the resulting binary image depends on how the grayscale image was converted to binary. Available conversion methods are displayed in Figure 2.5.

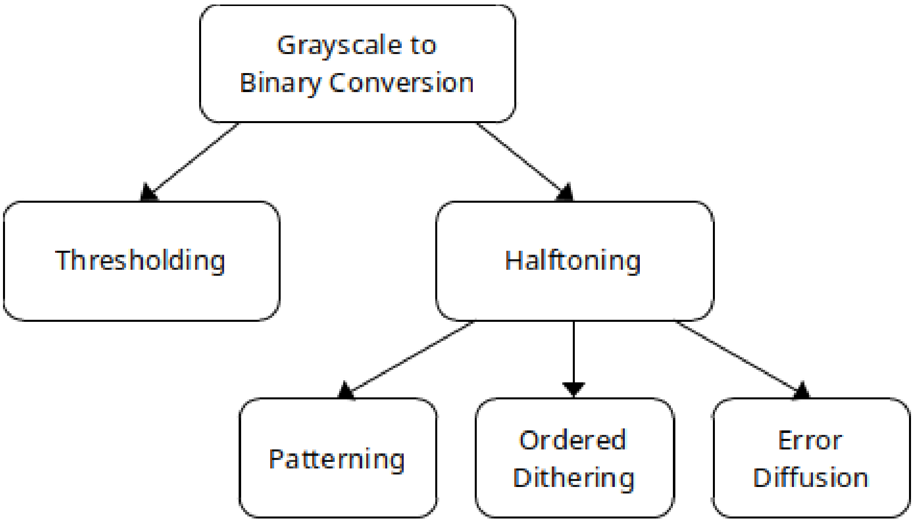
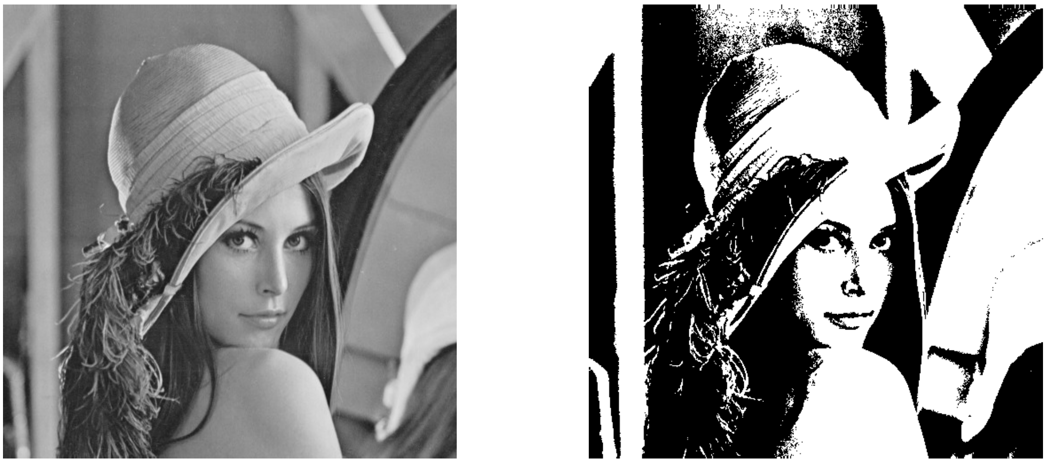


Figure 2.5 Conversion methods for grayscale to binary

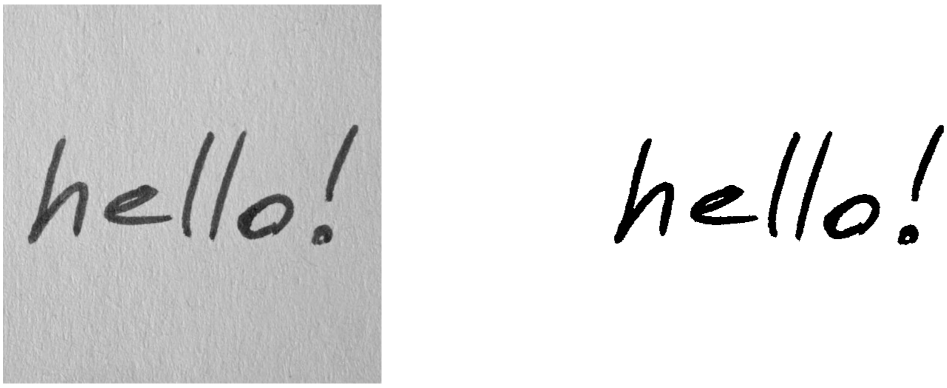
Thresholding is the simplest grayscale-binary conversion method (Appendix 1). In this method, a threshold value T must be determined in the range of 0 to 255. Then all pixels in the grayscale image which have intensity levels below T must be replaced with 0 (black), and all the other pixels (whose intensity levels are above T) must be replaced with 255 (white) in the new binary image. A grayscale image and its thresholded binary version (with T = 128) can be observed in Figure 2.6.



Grayscale Binary

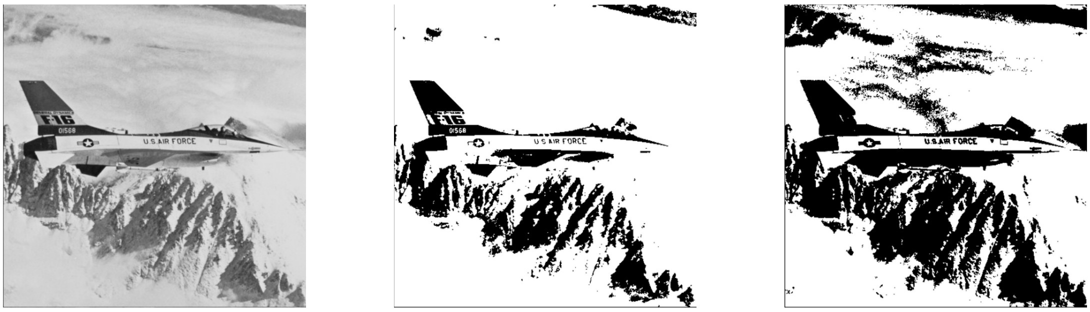
Figure 2.6 Photographic image in grayscale and binary formats

Thresholding is a straightforward and fast method and can create satisfying results for cases such as converting a scanned document to black and white (Figure 2.7). However, it presents problems when applied to more complex images. One such problem encountered in the thresholding method is the difficulty in choosing proper threshold values. Since every digital image has its unique gray level distribution, a threshold value chosen specifically for a grayscale image to better reflect features in binary may not perform well with another image; thus, proper threshold values must be determined for each image before conversion. For example, converting the grayscale image in Figure 2.7 into binary with T = 128 results in a good output, but the same threshold value produces a poor output when applied to a different photographic image (Figure 2.8), where a better output can be produced with a higher threshold value.



Grayscale Binary

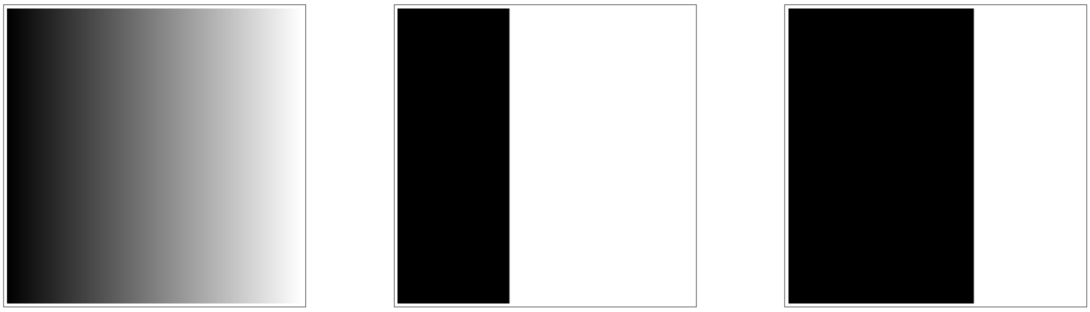
Figure 2.7 Image of handwriting in grayscale and binary formats



Original Grayscale Binary (T = 128) Binary (T = 192)

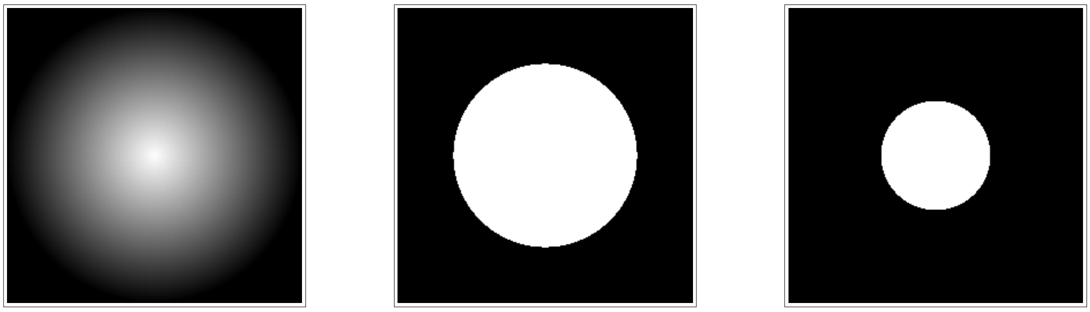
Figure 2.8 Results of thresholding with different threshold values

Another problem with the thresholding method is that since all intensity values below and above the T value are replaced with either a black or white pixel, spatial features in the source image, such as gradient textures and details, will not be preserved well and will be lost. This problem can be observed in Figure 2.9 and Figure 2.10; the smooth gradients will always be lost no matter what threshold value is used.



Grayscale Binary (T = 96) Binary (T = 160)

Figure 2.9 Grayscale linear gradient and its binary versions



Grayscale Binary (T = 96) Binary (T = 160)

Figure 2.10 Grayscale radial gradient and its binary versions

Therefore, although the thresholding method can be a fast and practical solution for converting images with simple features into binary format, it cannot be considered a desirable conversion method for all grayscale images. For grayscale images with complex features (such as images with photographic features), halftoning is a better alternative than thresholding. The purpose of halftoning algorithms is to let the HVS perceive the generated images as identical to the original image; this is achieved by imitating the color depth of the original image by creating an illusion using a limited color palette [12, 13].

Halftone images have been used in the publishing industry since its invention in the second half of the 19th century for printing grayscale photographs using mechanical printers that use only black ink – printing such photographs with printers required converting the photographs into black and white using dots of various sizes (Figure 2.11).

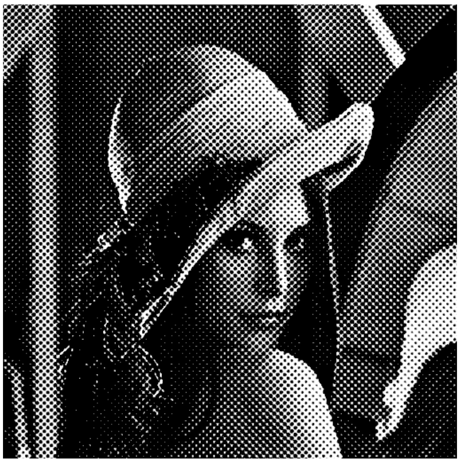


Figure 2.11 Halftone image

With the advancements in computer technologies in the last decades, halftoning has been adapted to the digital world as digital halftoning. Digital halftoning replaces the dots used in classical halftoning with pixels (Figure 2.12). In its early days, digital halftoning was widely used to display detailed graphics in display devices with limited color depths. In addition to displaying graphics, digital halftone images have also been used with digital printing devices.

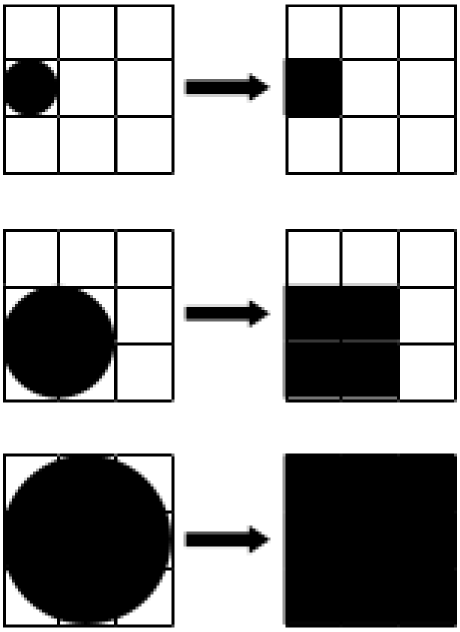
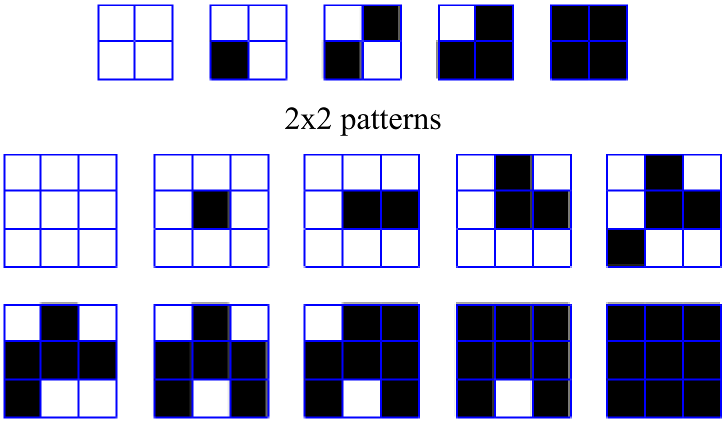


Figure 2.12 Classical and digital halftoning

As a result of the visual imitation of continuous tones, halftone images almost always offer better visual quality than thresholded images. There are different halftoning methods; the most popular halftoning algorithms can be studied under three topics: patterning, dithering, and error diffusion.

In the patterning method, binary images are generated using predefined patterns representing different intensity values stored in the source images. The patterning method can be considered an enhanced version of the thresholding method. In the thresholding method, pixels in the source image are divided into two intensity ranges, and the output is made of either 1x1 black or 1x1 white blocks. In the patterning method, the source image gets divided into N^2+1 intensity ranges, and the output is made of various NxN blocks, where N is the width or height of the patterns used [14] (Figure 2.13). Then, the output image is constructed by choosing the proper pattern for each pixel (Equation 2.3).

$$r=\left\lfloor\frac{P}{N^2+1}\right\rfloor \tag{2.3}$$

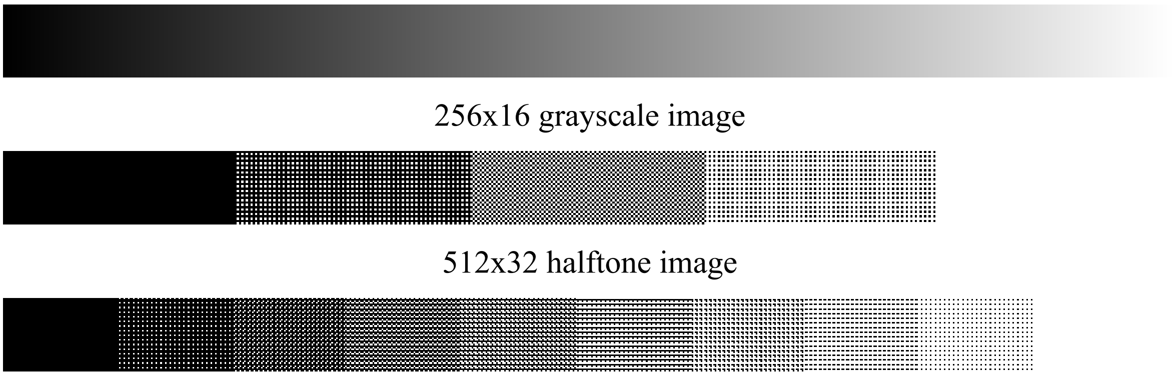


2x2 patterns

3x3 patterns

Figure 2.13 Sample patterns used for halftoning

It should be noted that when this halftoning method is chosen, each pixel in the source image will be represented by larger patterns in the output image; therefore, the output images will always have a higher spatial resolution than their source versions (Figure 2.14).



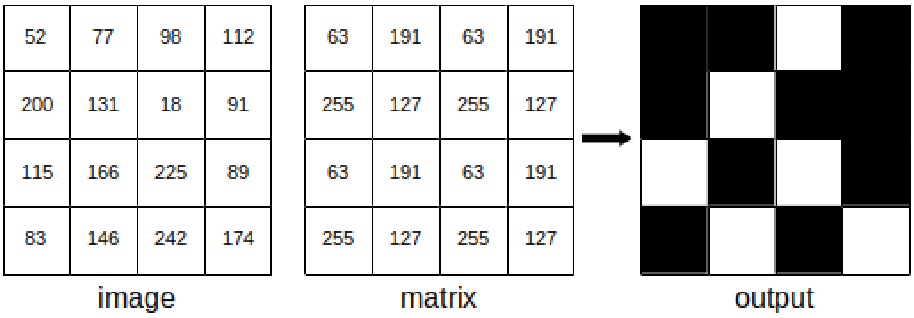
256x16 grayscale image

512x32 halftone image

768x48 halftone image

Figure 2.14 Source grayscale image and its halftone versions obtained with patterning

Another popular halftoning method is dithering. In dithering, a matrix of threshold values (also known as a threshold map) is used to generate binary images. This threshold map is tiled over the source image, and every pixel in the source image is thresholded with a different value from the map to generate the output image (Figure 2.15). Unlike patterning, the dimensions of the outputs generated with this method are the same as their



52	77	98	112
200	131	18	91
115	166	225	89
83	146	242	174

image

63	191	63	191
255	127	255	127
63	191	63	191
255	127	255	127

matrix

Figure 2.15 Ordered Dithering

original versions because dithering is basically a thresholding method applied per-pixel basis.

The threshold maps for dithering can be generated with different methods. Clustered dot dithering, for example, keeps threshold values closer to each other, while dispersed dot dithering distributes thresholds far away from each other. A well-known dithering method is Bayer Dithering [15]. Bayer dithering uses an N-by-N threshold map, where N is a value chosen from powers of 2. The smallest Bayer matrix is given in Figure 2.16. The numbers in the Bayer matrix are distributed such that the distance between neighboring numbers is as large as possible. The matrix keeps this property even after rotation and reflection.

0	2
3	1

matrix values

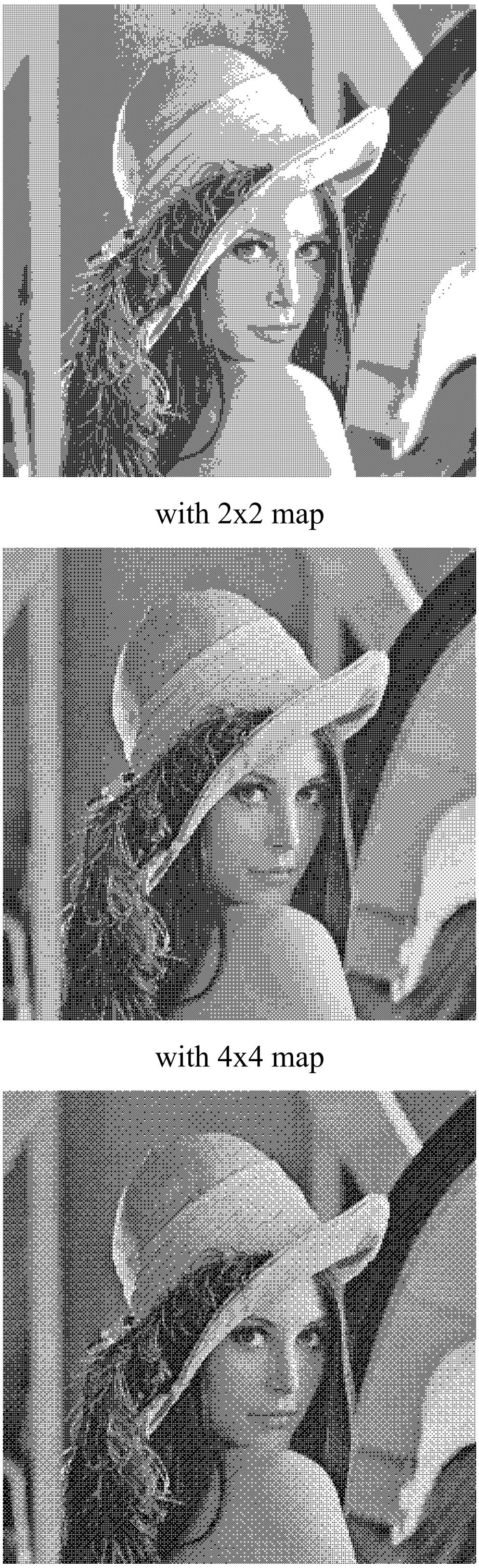
63	191
255	127

thresholds in 0 - 255 range

Figure 2.16 2x2 Bayer Matrix

It is possible to define four thresholds using the 2x2 threshold map in Figure 2.16. For more precise outputs, the map's dimensions can be increased with the help of a recursive operation (Equation 2.4). Presented in Figure 2.17 is the comparison of outputs produced using threshold maps with different dimensions.

$$M(n)=4*\begin{bmatrix} M(n-1)+0 & M(n-1)+2 \\ M(n-1)+3 & M(n-1)+1 \end{bmatrix} \tag{2.4}$$



with 2x2 map

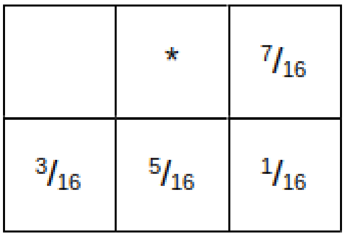
with 4x4 map

with 8x8 map

Figure 2.17 Bayer Dithering results obtained from using maps of different sizes

Error diffusion is an another popular method for binary image generation. In error diffusion, the value of each pixel in the binary image is again determined by a threshold value T. But unlike thresholding or dithering methods, where the job on each pixel is complete after the thresholding operation, the difference between the original and new values (called the quantization error) of each pixel is additionally distributed among unprocessed neighboring pixels using error distribution kernels until all pixels in the source image is processed. Similar to the previously explained dithering methods, images generated with error diffusion have the same spatial resolution as their original grayscale counterparts.

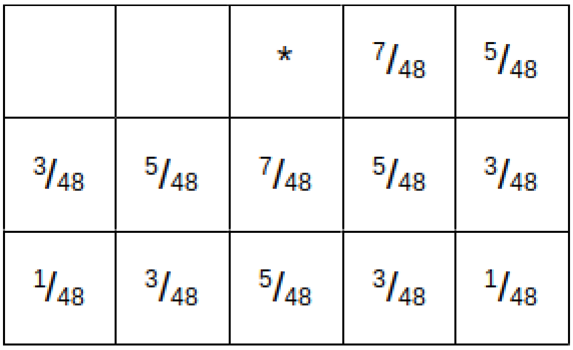
The original kernel (Figure 2.18) proposed by Floyd and Steinberg in 1976 is a popular and widely studied one [16]. This kernel produces fast results as it uses only four coefficients, but generated outputs are also prone to containing worm artifacts (i.e., a set of pixels that are aligned in one direction due to pushing the quantization error) and incorrectly generated smooth regions where there is none in the original image.



	*	${}^{7}/_{16}$
${}^{3}/_{16}$	${}^{5}/_{16}$	${}^{1}/_{16}$

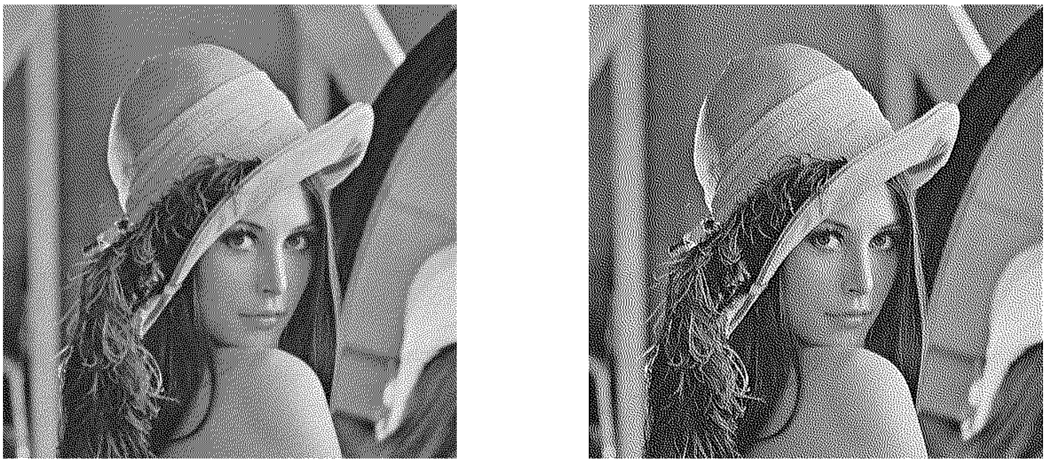
Figure 2.18 The Floyd-Steinberg Kernel

An another halftoning method named "Minimized Average Error" was proposed by Jarvis, Judice, and Ninke [17] also in 1976. Their kernel (Figure 2.19) uses more coefficients than the Floyd-Steinberg kernel; therefore, their method is more complicated than Floyd and Steinberg's. However, it distributes the quantization error better and generates a smoother output. A comparison of outputs generated using Floyd-Steinberg’s and Jarvis-Judice-Ninke’s methods is presented in Figure 2.20. Both images are created with T = 128.



		*	$^{7}/_{48}$	$^{5}/_{48}$
$^{3}/_{48}$	$^{5}/_{48}$	$^{7}/_{48}$	$^{5}/_{48}$	$^{3}/_{48}$
$^{1}/_{48}$	$^{3}/_{48}$	$^{5}/_{48}$	$^{3}/_{48}$	$^{1}/_{48}$

Figure 2.19 The Jarvis-Judice-Ninke Kernel



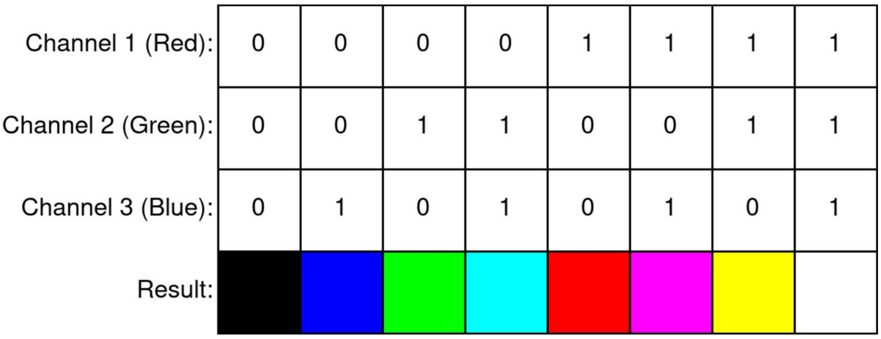
Floyd-Steinberg

Jarvis-Judice-Ninke

Figure 2.20 Output comparison of Floyd-Steinberg and Jarvis-Judice-Ninke methods

In addition to Floyd-Steinberg and Jarvis-Judice-Ninke methods, there are other derivatives of these methods, such as Shiau-Fan [18], Stucki [19], and Atkinson[20], that basically follow the same procedures with modified coefficients.

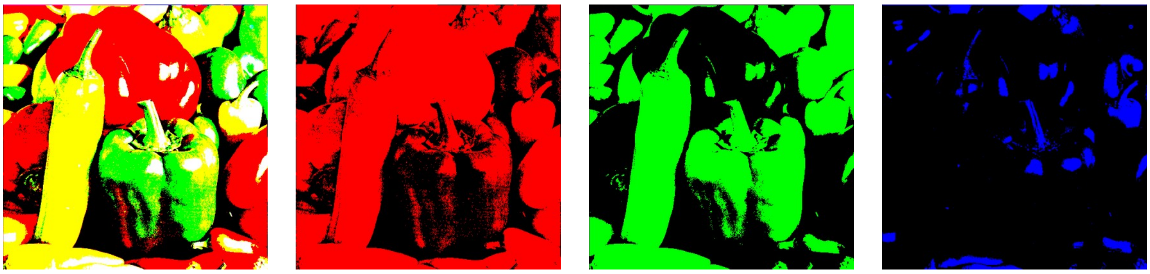
As explained previously, grayscale images commonly use a single 8-bpp channel, and color images most commonly use three separate 8-bpp channels. If this logic is applied to binary images by increasing the number of channels from one to three, binary color images can be generated. Such images can represent a total of $2^3 = 8$ different colors (Figure 2.21).



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:	⬛	🟦	🟩	🩵	🟥	🟪	🟨	⬜

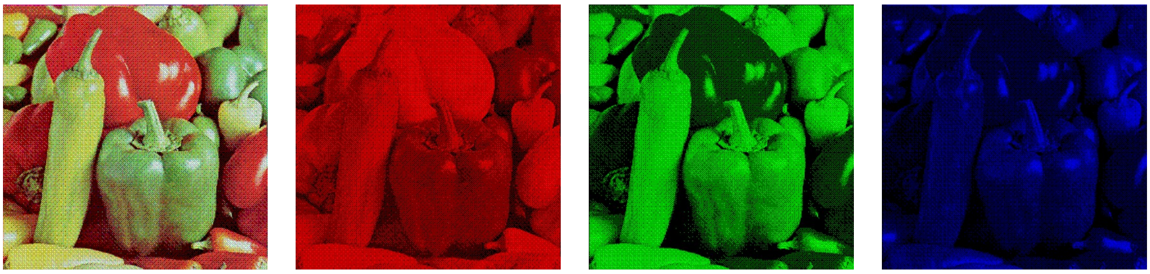
Figure 2.21 Colors in a 3-bit binary image

Since binary color images are composed of three separate 1-bpp channels, they can easily be generated from color images using the previously explained conversion methods by applying them separately on all three channels (Figure 2.22, Figure 2.23). Unfortunately, it is impossible to generate this type of images suitable for people suffering from various types of color blindness due to the limited amount of possible color options.



Composite Image Red Channel Green Channel Blue Channel

Figure 2.22 A thresholded binary image and its channels



Composite Image Red Channel Green Channel Blue Channel

Figure 2.23 A Bayer Dithered binary image and its channels

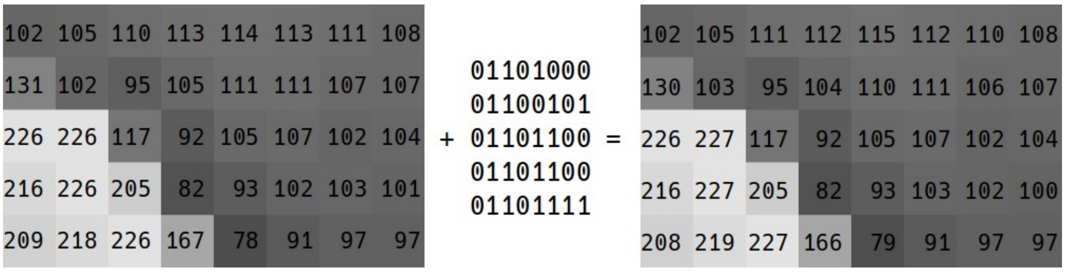
2.3. Ecology-Friendly Printing

When a document is to be printed, the printer uses different substances depending on the type of printer. Laser printers use toner cartridges for printing; toner is a fine-grained solid substance that is transferred and heated to stick to the surface during printing. On the other hand, inkjet printers use wet ink cartridges that spray tiny drops of ink over the surface during printing. When ink is sprayed over the paper, it spreads over the surface, which means that the same quality output can be produced using less ink.

There are printing methods that aim to conserve ink by exploiting this behavior [21, 22]; these approaches add tiny empty spaces inside the printed texts. The neighboring sprayed ink will fill these spaces; thus, the same quality of printed material can be obtained with less consumed ink. It has been estimated that materials printed with this technology consume almost 50% less ink than traditional media [22].

2.4 Related Works

The most common steganography approach encountered for digital image carriers is hiding the payload in the least significant bit (LSB) of pixels in the carrier image or variations of this approach [23, 24, 25, 26, 27]. In this approach, the payload is converted to an array of bits, and LSBs of pixels in the carrier image are appropriately replaced with these payload bits. The values stored in these bits are so insignificant to the pixel itself that the HVS cannot detect the difference when the LSB of one of the two neighboring pixels with the same value is toggled. A demonstration is presented in Figure 2.3, where the bits of the secret word "hello" are embedded in an image. As evident in this figure, the changes caused by the LSB steganography method are usually undetectable by the HVS.



102	105	110	113	114	113	111	108
131	102	95	105	111	111	107	107
226	226	117	92	105	107	102	104
216	226	205	82	93	102	103	101
209	218	226	167	78	91	97	97

+

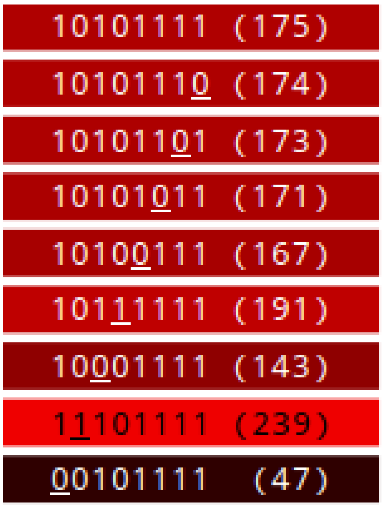
01101000
01100101
01101100
01101100
01101111

=

102	105	111	112	115	112	110	108
130	103	95	104	110	111	106	107
226	227	117	92	105	107	102	104
216	227	205	82	93	103	102	100
208	219	227	166	79	91	97	97

Figure 2.24 LSB Steganography

Since it is possible to use all pixels' LSBs in the image, the maximum payload capacity is usually limited to the number of pixels in the carrier image. If the payload is larger than the total amount of pixels, this limit can be raised by using more bits per pixel at the expanse of increased risk of exposure to the human eye because the change in the pixel's color becomes more apparent with each additionally used bit (Figure 2.4).



10101111 (175)
10101110 (174)
10101101 (173)
10101011 (171)
10100111 (167)
10111111 (191)
10001111 (143)
11101111 (239)
00101111 (47)

Figure 2.25 Effects of flipping each bit in a pixel

Although the changes caused by LSB steganography will often avoid detection by the HVS, they can still be detected with the aid of computers. In order to avoid detection and extraction by such precise systems, the security of payloads within their carriers can always be enhanced by using additional security measures such as encryption and encoding. For example, [28] encrypts and scatters the payload using a pseudo-random number generator. The results they have obtained are of high visual quality and show that their method can embed payloads as long as 2000 bytes in 256x256 color images. [29] encrypts the payload with Vigenere Cipher before embedding it in the LSBs. [30] applies AES encryption not on the payload, but rather on the stego image obtained after hiding the payload in the LSBs of the carrier image. When the payload is longer, it can be compressed prior to hiding. For example, in [31], the payload is encoded with the Huffman algorithm before hiding in the carrier. In all these methods, payload extraction procedures require proper decryption or decoding operations in addition to LSB extraction to obtain readable payloads, which are considered as security and efficiency enhancements for steganography operations.

In the absence of security enhancements, hiding the payload in LSBs of sequential pixels in the carriers can risk the strength of the payload, as the carrier can easily be scanned linearly to obtain a readable payload. In order to resist such extraction methods, the embedding order of the bits can be scrambled through reversible methods. For example, the previously explained method proposed by [28] uses a pseudo-random number generator to scramble the order of the payload. Similarly, [32] uses the RC4 algorithm for random number generation to hide the given payloads in the carriers, where both the payload and the cover are in digital image format. As a different approach, instead of scrambling the order of the bits, [33] proposes a method where bits of the payload are stored in different bits of each pixel, not only the LSB. Evaluation of their method showed that a payload of 278 kilobytes (KB) could successfully be hidden in a 596 KB long uncompressed image without any visual degradation.

LSB steganography methods operate on images with high bpp values, such as grayscale and color images. On the other hand, binary images offer much fewer spatial features due to their nature; therefore, they cannot be used as carriers for the majority of the steganography methods implemented to operate on grayscale or color images. However, the ways the binary images are generated and their structures can be exploited for implementing unique steganography methods. For example, [34] offers a method where all characters in a given plaintext payload are first converted into unique 3x3 patterns, then distributed on the carrier image by finding suitable locations for each pattern. In [35], an edge detection scheme working on binary images for finding suitable hiding spots has been proposed. This method analyses all edges available in a binary image, compares both the old and new visual quality of the carrier after flipping pixels from white to black or vice versa and hides bits of the payload in suitable pixels. In addition, hiding image-type payloads, as well as plaintext payloads, into binary carriers is a widely studied practice.

When compared to grayscale or color images, the distortions created by the payload in binary carriers are more visible due to the higher contrast between each pixel; therefore, precautions for minimizing these distortions must be taken. In [36], this is achieved by the proposed "pixel density selection" concept, which reveals suitable hiding places such as regions with complex textures. Similarly, the concept of optimal dispersion degree is introduced in order to choose appropriate hiding locations in the carrier image in [37]. [38] uses the concepts of pixel density proportion and distortion fusion to find such suitable regions in the carrier to hide the payload. All these algorithms are employed to minimize the visual and statistical adverse effects of flipped bits on the carrier image and to maintain the density of black and white pixels as close as possible to the original image.

Similar to secret sharing [39] and visual cryptography [40], which are schemes that describe how to split a given payload into parts instead of keeping it as a single piece, some methods produce multiple outputs and distribute parts of the given payloads into each output. Some of the proposed methods produce multiple output images that must be stacked over to reveal the payload they are carrying. Among these methods, some as proposed in [41, 42], create carriers where the original image is also required during payload extraction. However, the methods in [43, 44] do not require the original version for extraction. The method proposed in [45] extends this steganography approach such that the images needed to be stacked over can also be from different origins.

3. METHODS

In this chapter, the designed steganography methods will be explained. All steganography methods have been implemented to hide plaintext payloads over binary images. The methods differ from each other according to how the binary image was generated; each of the following sections focuses on a different carrier type and explains both the payload hiding and payload extraction procedures.

3.1. Hiding on Thresholded Images

The proposed method (Appendix 2) takes inspiration from existing ecology-friendly printing methods and produces texts with holes, where the holes are actually bits of provided payloads. The method works by detecting text in the carrier image using optical character recognition and evenly distributing payload bits from the first carrier letter to the last. Figure 3.1 displays a sample output obtained by this approach.

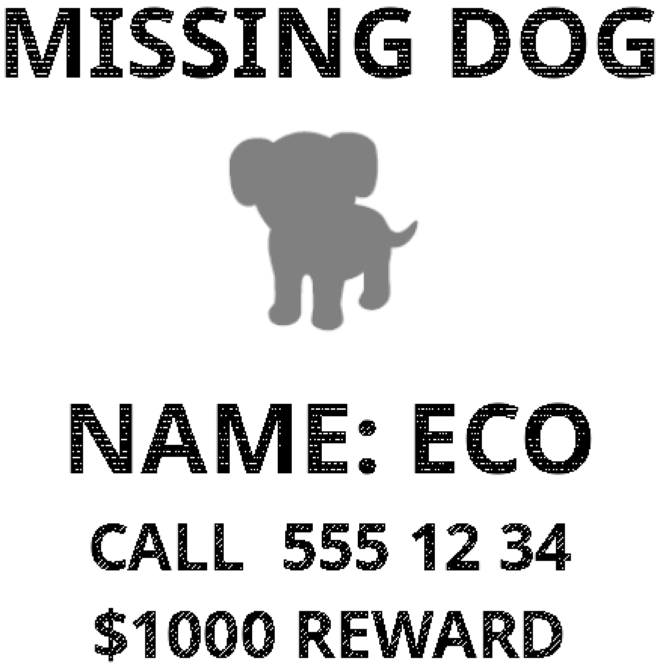
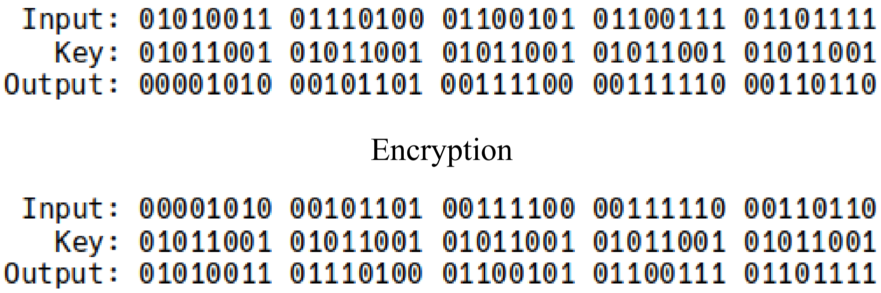


Figure 3.1 Output of thresholded images method

The distance D between each payload bit is determined as the ratio of all usable black pixels in the eroded cover letters to the length of payload bits. Since this value can vary according to the carrier and the payload, it has to be made known to the receiver; this is achieved by turning the first Dth usable black pixel of the first carrier letter to white.

In order to increase the payload's security, we propose applying the Exclusive OR (XOR) cipher on the payload before embedding. XOR cipher is a simple encryption method where the data is bitwise XOR'ed with a chosen key. If the length of the key is shorter than the data, the key can be sequentially repeated to match the data's length. To decrypt the encrypted data, it can be XOR'ed again with the same key to reveal the original data (Figure 3.2).



```
 Input: 01010011 01110100 01100101 01100111 01101111
   Key: 01011001 01011001 01011001 01011001 01011001
Output: 00001010 00101101 00111100 00111110 00110110
```

Encryption

```
 Input: 00001010 00101101 00111100 00111110 00110110
   Key: 01011001 01011001 01011001 01011001 01011001
Output: 01010011 01110100 01100101 01100111 01101111
```

Decryption

Figure 3.2 XOR Cipher

Usually, it would be assumed that all bits extracted from carriers will directly form American Standard Code for Information Interchange (ASCII) letters when grouped together. With XOR cipher, extracted bits would additionally require a key to decipher the extracted bits. Instead of embedding an extra key into the carrier, we have implemented the method to choose the key for all bits in each carrier letter as the previous letter (Figure 3.3). An exception has to be made for bits assigned to the first carrier letter because it is impossible to choose a key letter for those bits. Instead of keeping these bits without encryption or ciphering them with the null character (0x00), which will effectively reveal a critical part of the payload, the algorithm has been implemented such that all bits assigned to the first carrier letter are XOR'ed with the last carrier letter as the key.

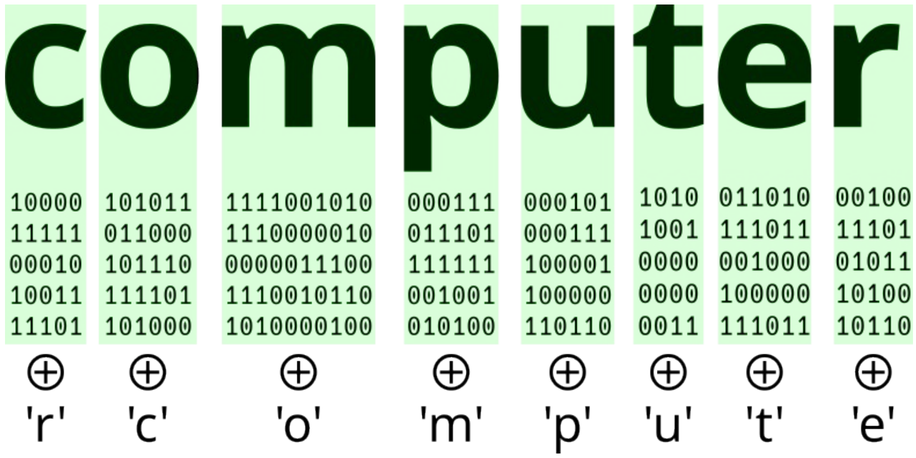


Figure 3.3 Application of XOR Cipher with the proposed method

3.2. Payload Extraction from Thresholded Images

In order to extract the payload hidden in thresholded images, as explained above, it is first necessary to obtain the distance D between the payload bits. After recognizing the letters in the carrier image and eroding them, the position of the first white pixel denotes this distance. After gaining this information, sequentially scanning all letters will allow gathering all XOR’ed payload bits. When a letter is completely scanned, the collected bits are XOR’ed with the key to reveal the actual payload bits. Finally, when all carrier letters have been scanned, translating all collected bits to ASCII will reveal the payload.

This method has been proposed as a preliminary study to determine what kinds of approaches can be performed on binary images before moving to halftone images.

3.3. Hiding on Halftone Images

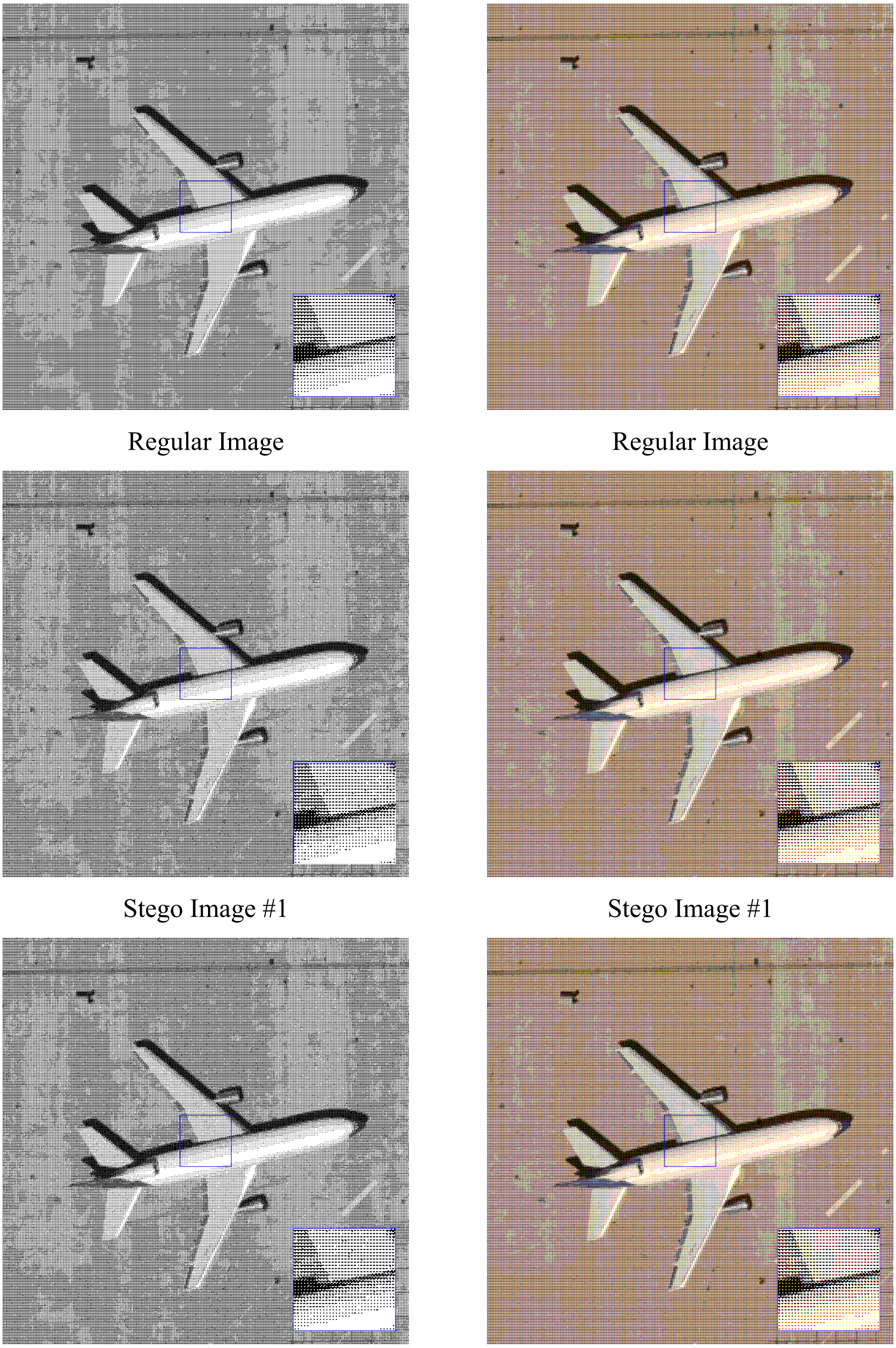
Due to the differences between each halftoning algorithm, multiple variations of the hiding algorithm have been implemented. Instead of changing the halftoning algorithms, the proposed methods focus on integrating the payload hiding scheme into the halftoning algorithms. Each method basically converts a given grayscale image into binary using the desired halftoning method and alters the produced image at the last moment such that a different pixel or pattern is used in the output rather than the one calculated by the halftoning algorithm.

In order to increase the payload's security, the proposed methods take inspiration from the visual secret-sharing method [40]; they produce multiple outputs, and all payload bits are scattered randomly across these output images. According to how the payload is distributed, every image in the set will have slight differences. Additionally, the methods will produce different unique sets of an image due to the randomized nature of the payload distribution process.

3.3.1. Hiding on halftone images using patterns

Hiding payloads in halftone images generated with patterns requires an altered version of the pattern decision procedure of the halftoning algorithm explained previously (Appendix 3). The hiding method starts by dividing the source image into dLen-long blocks of pixels, where dLen is the ratio of total payload capacity to the length of the payload in bits. The method is implemented to hide one payload bit in each block; this is necessary for ensuring the payload is spread across the whole carrier instead. For each payload bit, a random output share is chosen. In order to avoid statistical steganalysis attacks that target every Nth pixel in an image, the pixel which will carry the payload bit is also chosen randomly within the block. Like every other pixel in the source image, the pattern for payload carrier pixels will be chosen according to Equation 2.3. However, instead of using the chosen pattern, the pattern before the chosen pattern is used if the payload bit is 0, or the pattern after the chosen pattern is used instead. This altered pattern is used only in the previously chosen output; the originally chosen pattern is used in all other outputs. This procedure is repeated until all payload bits are processed and hidden in the output shares. Figure 3.4 presents the comparison of both grayscale and color outputs obtained via this method with their regular counterparts.

This method has a slight limitation on the total carrier capacity. Since it is impossible to choose a pattern prior to the first pattern or similarly a pattern next to the last pattern, this method is implemented to avoid hiding bits in the darkest or lightest regions in the source image. This precaution results in lower payload capacity with carriers with such regions; therefore, carriers must be chosen appropriately.



Regular Image

Regular Image

Stego Image #1

Stego Image #1

Stego Image #2

Stego Image #2

Figure 3.4 Comparison of pattern-based steganography outputs

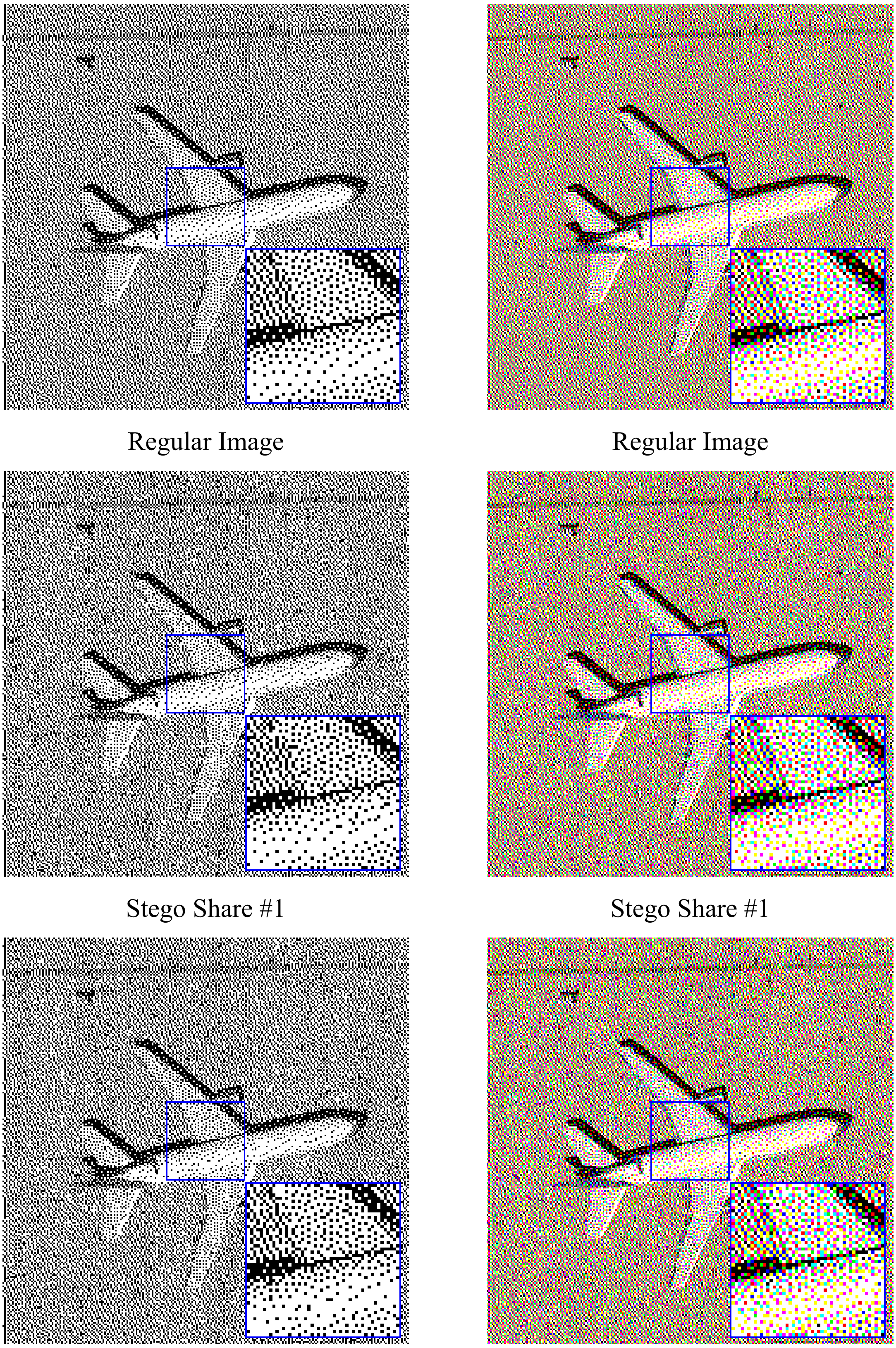
3.3.2. Hiding on halftone images using error diffusion

For the pattern-based hiding algorithm, it is possible to represent source pixels as patterns in output images by choosing among multiple patterns and embedding payload bits in these pixels using prior or successor patterns. This approach is unsuitable for halftone images generated with error diffusion because each pixel can be represented only as black or white in these images. In this variation of the hiding method, when a pixel is chosen as a bit carrier, the randomly chosen share gets either a black or white pixel depending on the payload bit, and all the other shares get the opposite color. All other features of the hiding algorithm are preserved without any changes. Figure 3.5 presents the comparison of both grayscale and color outputs obtained via this method with their regular counterparts.

3.4. Payload Extraction from Halftone Images

It has been previously explained that the hiding methods implemented for halftone images scatter each bit of the payload to a randomly picked output image. The hiding algorithms have been implemented so that when a random output image receives a payload bit, the pattern/pixel at the embedding coordinate will be represented differently from the pattern or pixel positioned at the same coordinate in the other output images. The implemented extraction algorithm (Appendix 4, Appendix 5) basically seeks out these differences; when any different pattern or pixel is encountered during scanning the same coordinate in all available carriers, this will be interpreted as the presence of a payload bit, and these bits will be sequentially collected together to reveal the payload. If all of the patterns/pixels collected from the same coordinate are found to be the same, then it is assumed that no payload bits are available in that coordinate, and that coordinate is skipped. After the last coordinate is processed, converting all collected bits into ASCII characters will reveal the payload.

Obtaining the whole set of output images generated during payload hiding is crucial to successful extraction. Since the payload is embedded as individual bits (rather than bytes), even a single missing carrier image will cause cascaded shifts in random positions of extracted bits, resulting in unintelligible outputs when the set is converted to ASCII



Regular Image

Regular Image

Stego Share #1

Stego Share #1

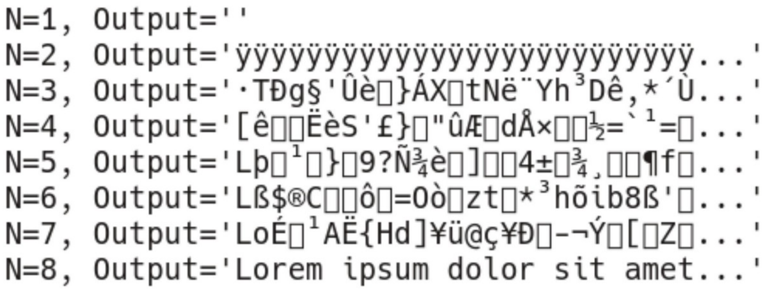
Stego Share #2

Stego Share #2

Figure 3.5 Comparison of error diffusion-based steganography outputs

characters. This outcome has been demonstrated in Figure 3.6; for Figure 3.6, a set of 8 carriers was created, and then the extraction algorithm was run on them eight times using an increased number of carriers on each attempt. Since there are no other carriers to compare with, it can be observed that no payload is extracted when a single carrier is used. The whole payload is successfully extracted when all previously generated carriers are present. All other extraction attempts result in various unintelligible outputs.

It should be noted that since the payload is distributed randomly, there is a slight possibility that available carriers can carry all eight bits of any payload character. A few legit payload characters can actually be extracted, as evidenced by cases where N=5, 6, and 7 in Figure 3.6.



```
N=1, Output=''
N=2, Output='ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ...'
N=3, Output='·TÐg§'Ûè□}ÁX□tNë¨Yh³Dê,*´Ù...'
N=4, Output='[ê□□ËèS'£}□"ûÆ□dÅ×□□½=`¹=□...'
N=5, Output='Lþ□¹□}□9?Ñ¾è□]□□4±□¾¸□□¶f□...'
N=6, Output='Lß$®C□□ô□=Oò□zt□*³hõib8ß'□...'
N=7, Output='LoÉ□¹AË{Hd]¥ü@ç¥Ð□-¬Ý□[□Z□...'
N=8, Output='Lorem ipsum dolor sit amet...'
```

Figure 3.6 Extraction results with different numbers of provided carriers

4. EXPERIMENTS AND RESULTS

In this chapter, the conducted tests for outputs produced using all proposed methods, along with the used test parameters, and both the obtained qualitative and quantifiable results will be presented.

4.1. Evaluation of Thresholded Carriers Method

For the evaluation of the method that generates ecology-friendly-like stego carriers, a short survey was conducted as a subjective evaluation. The outputs were not evaluated against computed quality metrics as the added white holes in the carrier letters would nevertheless cause a decrease in the quality of the stego carriers.

This survey presented the attendees with a simple pair of images consisting of identical texts generated using existing ecology-friendly methods and the proposed steganography algorithm. Also, the attendees were asked to answer how they perceived both images and their opinions about why these images were generated. Seventy different answers were recorded in this survey. The answers recorded for the first question are presented in Table 4.1. The second question received various answers, such as to get people's attention or conduct an eye exam. However, while only five attendees answered that they were generated for ink-saving purposes, none suspected that the second text was generated for secret communication.

Table 4.1 Subjective evaluation results for thresholded carriers

Observed Similarity	**Percentage of Attendees**
Very Different	20%
Different	21%
Neither different nor the same	29%
Similar	19%
The Same	11%

4.2. Evaluation of Halftone Carriers Methods

The methods proposed for halftone carriers have been tested both objectively and subjectively. In addition, their resistance against brute force steganalysis attacks has also been tested.

For objective and subjective tests, three images from UC Merced Land Use Dataset [46] are selected according to spatial textures present in them: airplane80 is chosen for its smooth textures, beach09 is chosen for its mixed textures, and forest22 is chosen for its complex textures. All images provided in the dataset are in 256x256 color format. Grayscale versions of the chosen carriers also had to be generated to assess the effectiveness of the methods that generate binary halftone images. In order to establish a basis for comparison, a total of 12 regular halftone versions (i.e., with no payloads) of the carriers have been generated as the initial step. Finally, a novel method that computes how much of the payload can be revealed by brute-force attacks has been implemented and applied for security tests.

4.2.1. Objective evaluation

The objective evaluation of the proposed methods has been conducted by calculating signal-to-noise ratio (SNR) (Equation 4.1) [47], peak signal-to-noise ratio (PSNR) (Equation 4.2) [48], and structural similarity index measure (SSIM) (Equation 4.3) [49] scores of the generated stego carriers. Among these metrics, SNR and PSNR are used to calculate the effects of added noise on the image, and SSIM is used to calculate perceptual differences between original and modified versions of an image.

$$SNR = 10\log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) \tag{4.1}$$

$$PSNR = 10\log_{10}\left(\frac{MAX^2}{MSE}\right) \tag{4.2}$$

$$SSIM(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \tag{4.3}$$

The proposed methods can use all pixels of carrier images to hide single bits in them; therefore, a 256x256 grayscale image can carry a maximum of 65,536 bits (8,192 bytes), and their color counterparts can carry a maximum of 196,608 bits (24,576 bytes). From this information, two separate payloads that are 25% and 50% of the total grayscale capacity have been generated (2,048 and 4,096 bytes, respectively). The same payloads have been used with both grayscale and color versions of the carriers to compare the results obtained when the same payload is used on different carriers. For the tests, 72 different sets with a total of 576 images have been generated using the parameters presented in Table 4.2. All these tests have been conducted by implementing the algorithms on MATLAB 2022a. Each test has been repeated three times, and average results are calculated; they are presented in Table 4.3 to Table 4.6 and Figure 4.1 to Figure 4.4.

Table 4.2 Objective evaluation parameters

Carrier Images	airplane80, beach09, forest22
Carrier Types	Binary pattern based Color pattern based Binary error diffusion based Color error diffusion based
Number of Output Shares	4, 8, 12
Payload Lengths	2,048 bytes, 4,096 bytes

Table 4.3 Objective evaluation results for pattern-based binary carriers

		airplane80			**beach09**			**forest22**		
NSHARE	**%**	**SNR**	**PSNR**	**SSIM**	**SNR**	**PSNR**	**SSIM**	**SNR**	**PSNR**	**SSIM**
4	**25**	19.4026	21.6214	0.9843	18.6862	21.5858	0.9838	17.6736	21.5862	0.9838
	50	16.3999	18.6187	0.9687	15.6763	18.5759	0.9677	14.6618	18.5745	0.9676
8	**25**	22.4135	24.6323	0.9921	21.6973	24.5969	0.9919	20.6840	24.5967	0.9919
	50	19.4103	21.6291	0.9844	18.6865	21.5861	0.9839	17.6724	21.5851	0.9838
12	**25**	24.1753	26.3941	0.9948	23.4601	26.3597	0.9946	22.4455	26.3582	0.9946
	50	21.1715	23.3903	0.9896	20.4479	23.3475	0.9893	19.4338	23.3464	0.9892

Table 4.4 Objective evaluation results for pattern-based binary color carriers

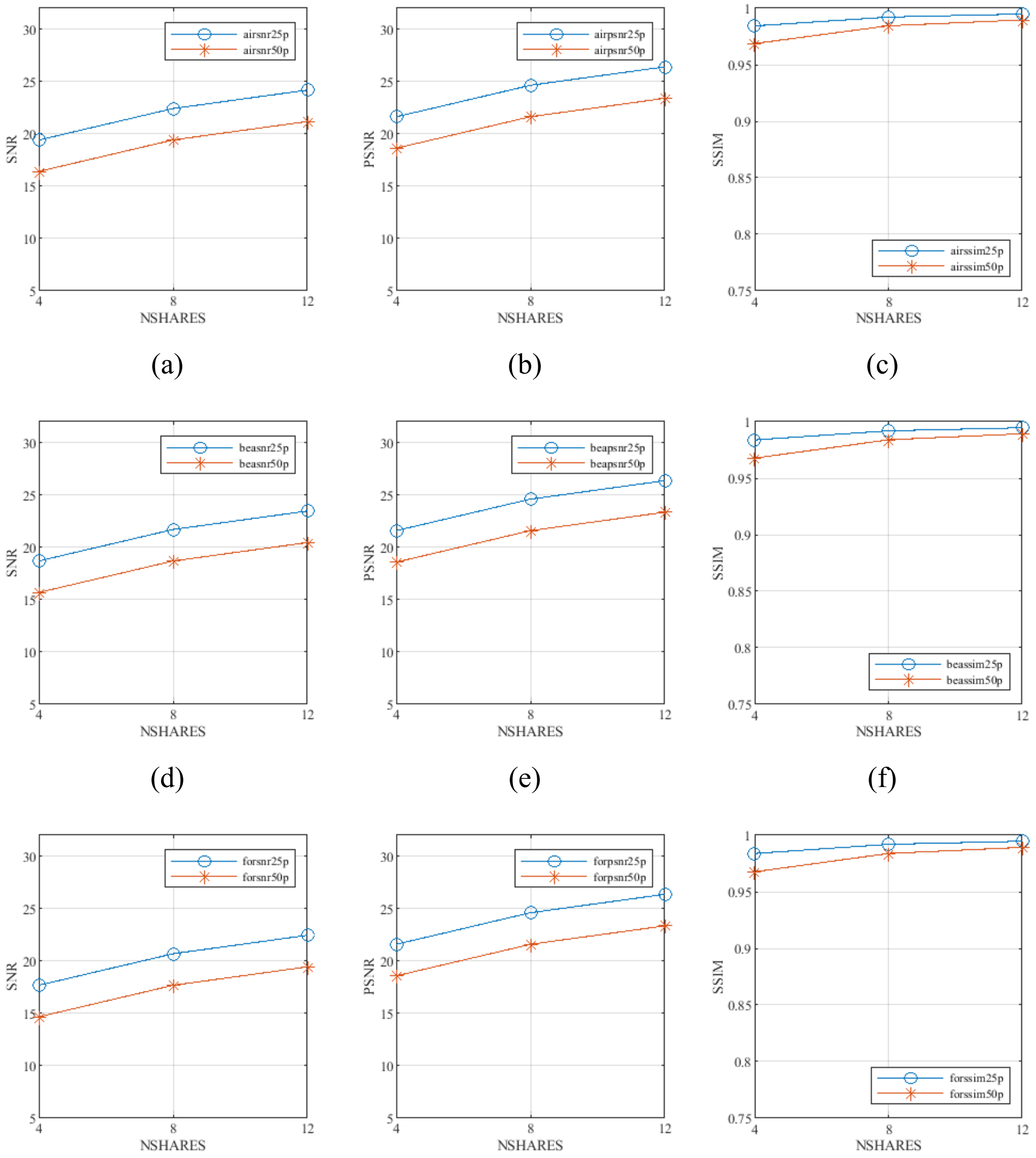
		airplane80			**beach09**			**forest22**		
NSHARE	**%**	**SNR**	**PSNR**	**SSIM**	**SNR**	**PSNR**	**SSIM**	**SNR**	**PSNR**	**SSIM**
4	**25**	24.0480	26.4016	0.9949	23.3402	26.3572	0.9945	22.2351	26.3572	0.9947
	50	21.0492	23.4028	0.9898	20.3308	23.3478	0.9890	19.2237	23.3457	0.9894
8	**25**	27.0621	29.4158	0.9974	26.3516	29.3685	0.9972	25.2468	29.3688	0.9973
	50	24.0633	26.4169	0.9949	23.3424	26.3593	0.9945	22.2343	26.3564	0.9947
12	**25**	28.8207	31.1743	0.9983	28.1131	31.1300	0.9982	27.0072	31.1293	0.9982
	50	25.8215	28.1751	0.9966	25.1035	28.1204	0.9963	23.9962	28.1182	0.9965

Table 4.5 Objective evaluation results for error diffusion-based binary carriers

		airplane80			**beach09**			**forest22**		
NSHARE	**%**	**SNR**	**PSNR**	**SSIM**	**SNR**	**PSNR**	**SSIM**	**SNR**	**PSNR**	**SSIM**
4	**25**	9.9747	12.1439	0.8682	9.2451	12.1378	0.8632	8.3936	12.1465	0.8682
	50	7.4984	9.6676	0.7686	6.8341	9.7267	0.7637	5.9897	9.7426	0.7718
8	**25**	12.9851	15.1544	0.9340	12.2556	15.1483	0.9316	11.4040	15.1569	0.9341
	50	10.5089	12.6781	0.8843	9.8448	12.7374	0.8817	9.0000	12.7529	0.8858
12	**25**	14.7466	16.9159	0.9560	14.0170	16.9097	0.9544	13.1657	16.9186	0.9560
	50	12.2704	14.4396	0.9229	11.6059	14.4985	0.9211	10.7614	14.5142	0.9239

Table 4.6 Objective evaluation results for error diffusion-based binary color carriers

		airplane80			**beach09**			**forest22**		
NSHARE	**%**	**SNR**	**PSNR**	**SSIM**	**SNR**	**PSNR**	**SSIM**	**SNR**	**PSNR**	**SSIM**
4	**25**	14.6643	16.9411	0.9566	14.0000	16.9555	0.9547	12.9906	16.9388	0.9564
	50	12.1335	14.4103	0.9229	11.6672	14.6228	0.9231	10.6407	14.5890	0.9256
8	**25**	17.6665	19.9433	0.9783	17.0225	19.9781	0.9773	16.0028	19.9510	0.9782
	50	15.1378	17.4146	0.9614	14.6962	17.6518	0.9615	13.6574	17.6056	0.9628
12	**25**	19.4294	21.7062	0.9855	18.7741	21.7297	0.9849	17.7551	21.7034	0.9855
	50	16.9010	19.1779	0.9743	16.4456	19.4011	0.9743	15.4136	19.3619	0.9751

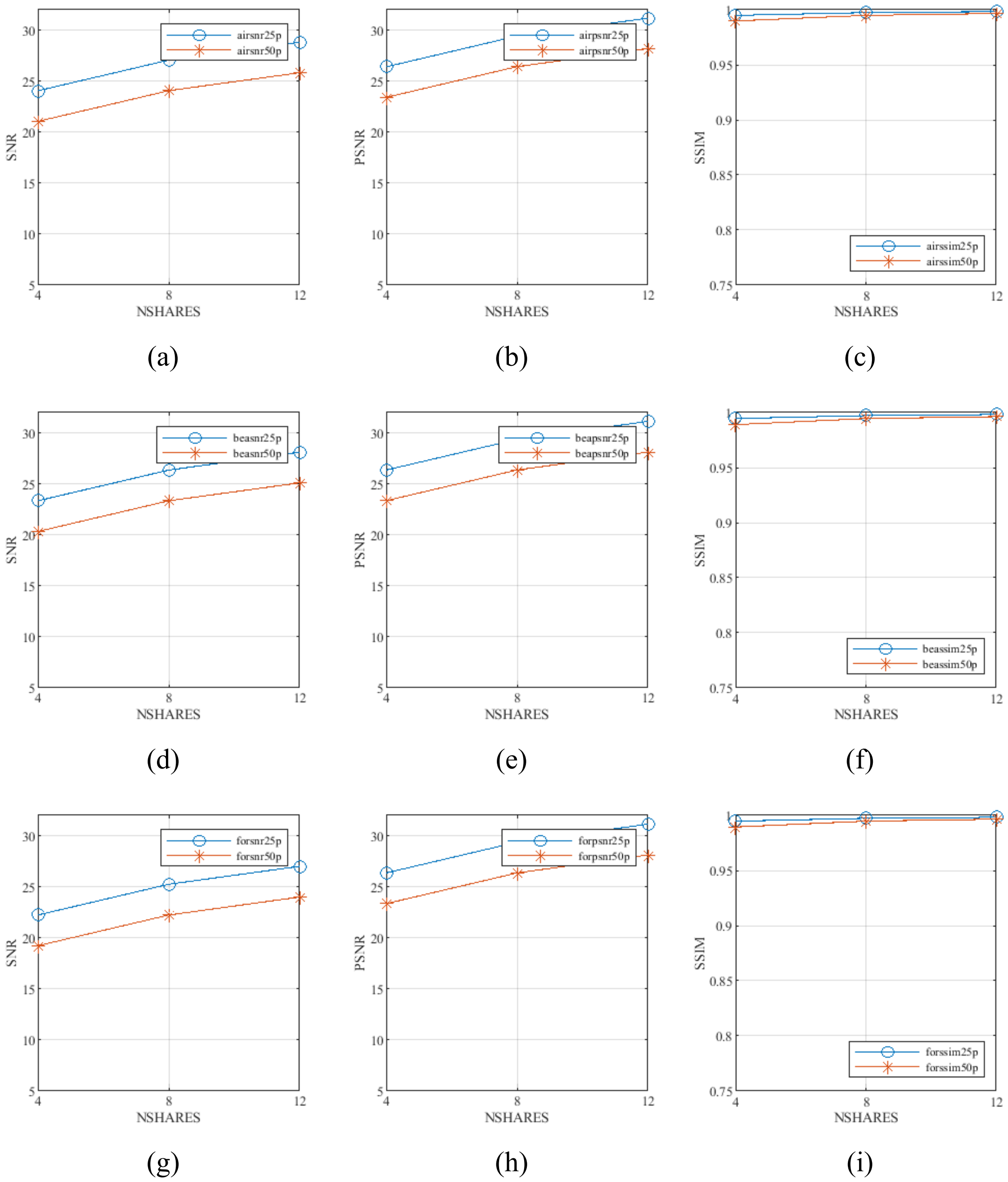


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.1 Objective test results for pattern-based binary carriers

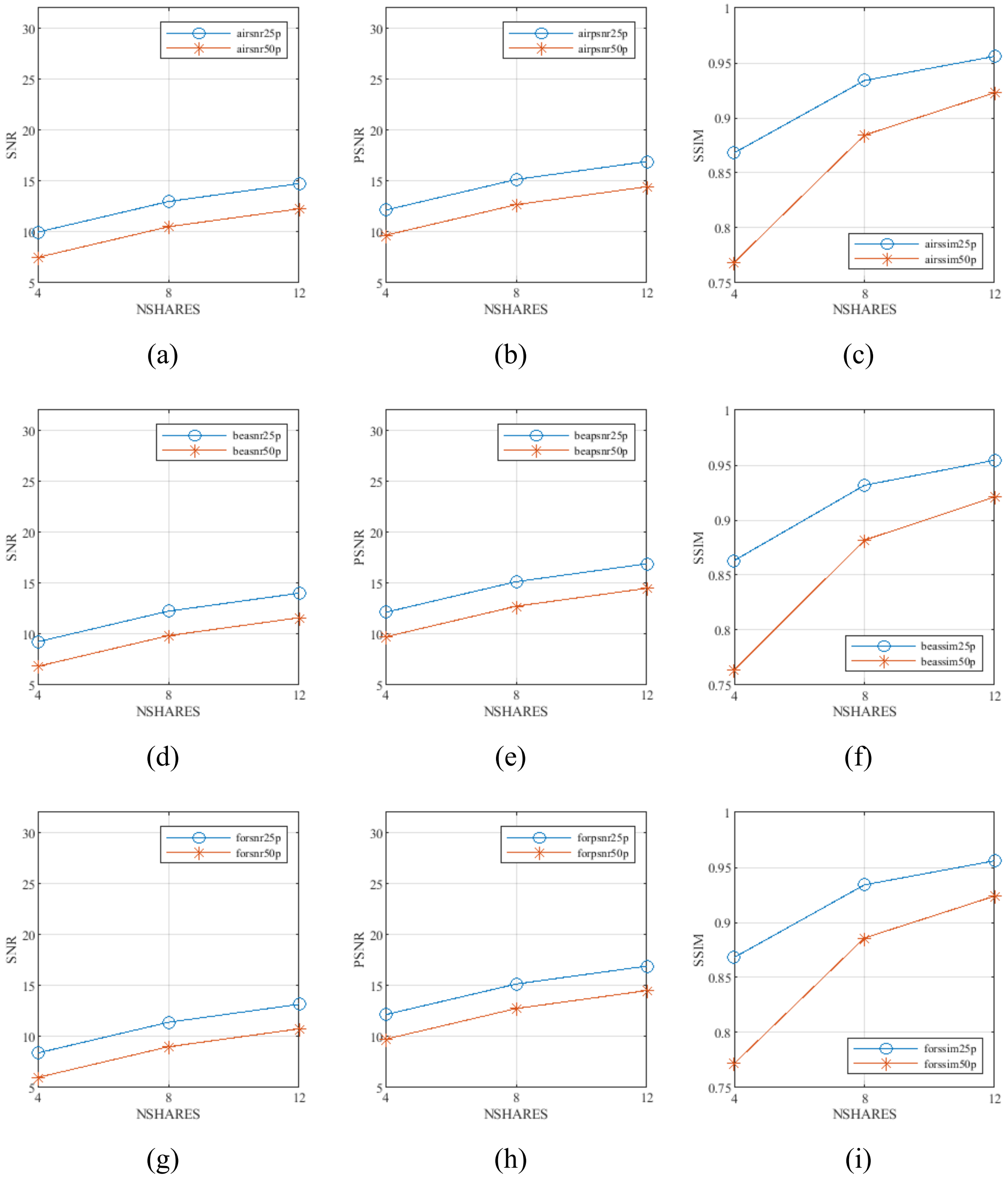


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.2 Objective test results for pattern-based binary color carriers

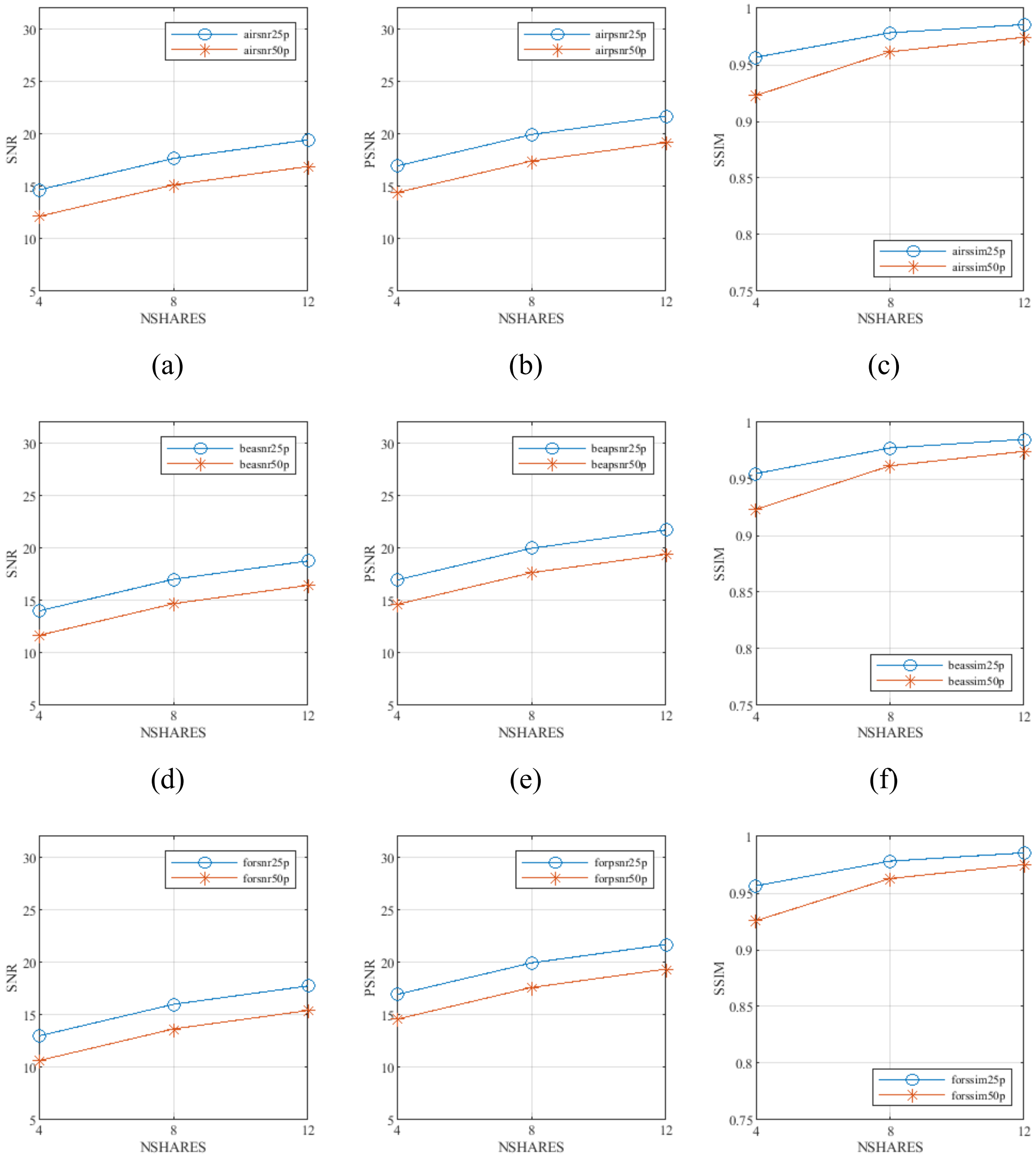


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.3 Objective test results for error diffusion-based binary carriers



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.4 Objective test results for error diffusion-based binary color carriers

In order to observe the relation between the payload size and output quality more precisely, additional tests have also been conducted. In these tests, 256 different payloads of increasing lengths between 32 and 8,192 bytes have been embedded in the previously used airplane80 image. The SNR values obtained from these tests are displayed in Figure 4.5. In digital images, an SNR value of 20 or higher can be considered acceptable; therefore, it can be concluded that for an average carrier, optimum payload lengths should be approximately less than 25%, 60%, 5%, and 10% of total payload capacity for pattern-based binary, pattern-based color, error diffusion-based binary, and error diffusion-based color carriers, respectively.

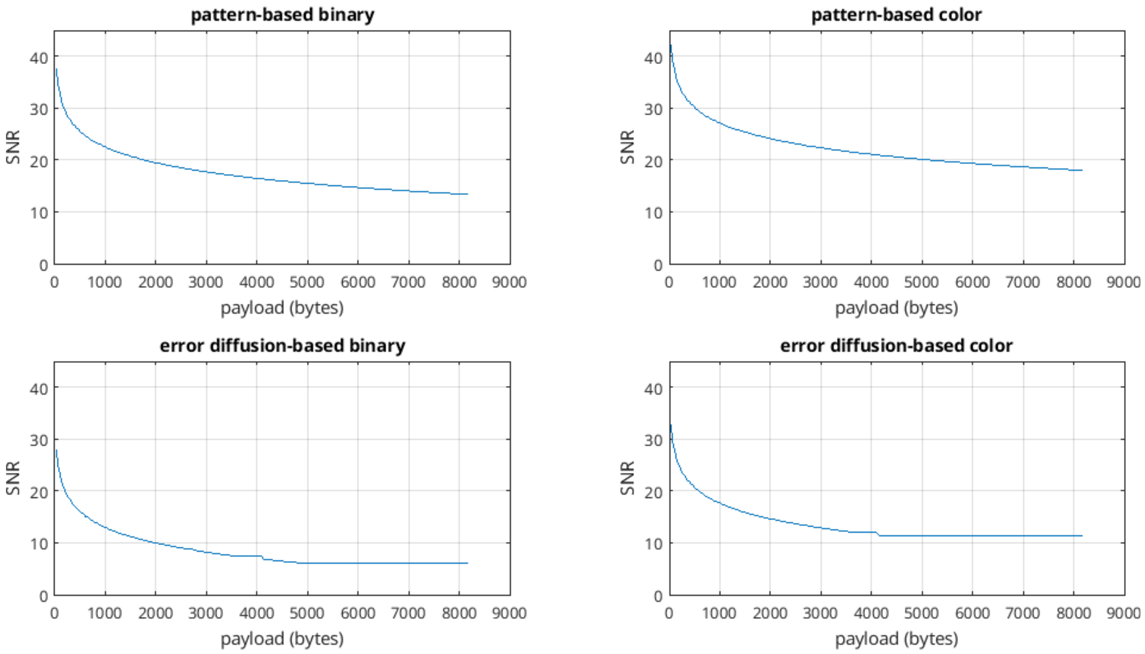


Figure 4.5 Precise SNR results

4.2.2. Subjective evaluation

In addition to machine-made objective assessment, the quality of the outputs has also been evaluated by subjective human perception. For this purpose, a survey, of which 95 people attended, was conducted. In this survey, the attendees were presented with 12 pairs of regular and stego images that were chosen from the images generated for objective evaluation. Due to the reasons explained before, it was not possible to present the color images in this survey in a format suitable for people with color blindness. The attendees were asked to observe and reply with the amount of difference they perceived by visually comparing each image pair. The results of this survey are presented in Table 4.7.

4.2.3. Safety tests

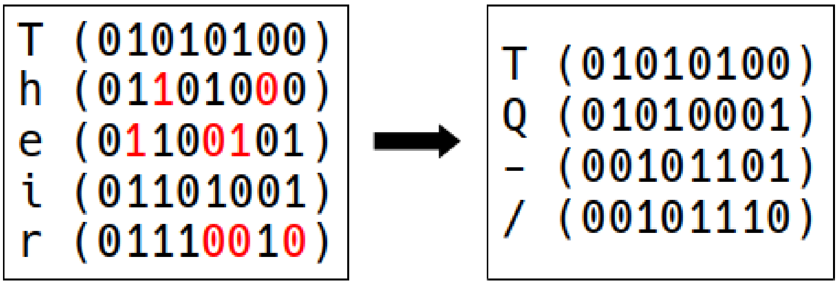
Undetectability is one of the primary requirements for stego carriers; suspicious-looking carriers may get the attention of attackers. These attackers can attempt payload extraction with or without knowing how the payload was hidden. In such cases, the steganography method used to hide the payload must be robust against unwanted payload extraction.

Table 4.7 Subjective evaluation results

	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%

The proposed hiding algorithm for halftone images has been developed to ensure that successful payload extraction can only be accomplished with access to the whole set of carriers. Even a single missing carrier can drastically affect the extraction result. This outcome is achieved by hiding the payload in bits (rather than bytes); any number of missing carriers will cause shifts in the extracted payload, resulting in illegible outputs when the bits are converted back to bytes.

In order to represent the effects of these shifts in bits, a sample scenario is presented in Figure 4.6. In this figure, the "Their" word is chosen randomly and presented with binary representations of each letter in the word (40 bits). In this figure, bits in black should be imagined as payload bits available during extraction, and bits in red should be imagined as bits initially available during hiding but missing during payload extraction. When the bits highlighted with red color are removed from the payload, and the remaining bits are grouped together to form letters, "TQ-/" (32 bits) is obtained as a result (represented along with its bit representation on the right side of the figure). In this example, the "T" letter could successfully be extracted because all bits belonging to a letter may be present in available carriers during extraction and align perfectly to represent the original letter. However, although all bits belonging to the "i" letter are also available during extraction, it is impossible to interpret the original letter meaningfully in the output since its bits are misaligned because of previously missing bits belonging to the "h" and "e" letters.



```
T (01010100)
h (01101000)
e (01100101)
i (01101001)
r (01110010)
```

```
T (01010100)
Q (01010001)
- (00101101)
/ (00101110)
```

Figure 4.6 Effects of missing shares on extracted text

In order to estimate how much of the payload can successfully be revealed with missing shares, multiple tests have been conducted. For these tests, three different sets of carriers with different numbers of shares (4, 8, 12) have been generated. The chosen payload is 2,048 bytes long, and the dimensions of chosen carrier image are 256x256. Then, several extraction attempts with different numbers of input shares were conducted (e.g., 3 of 8, 11 of 12). Since the algorithm is nondeterministic and produces totally different outputs for the same inputs each time it has been executed, this procedure has been repeated three times, and average results have been calculated from obtained results. The results of these tests are presented in Table 4.8 to Table 4.10.

Table 4.8 Payload extraction attempt results for 4 carriers

NSHARES	**Number of bytes extracted**	**Length of longest revealed payload**
1	0	0
2	996 – 1,012	0
3	1,498 – 1,509	2
4	2,048	2,048

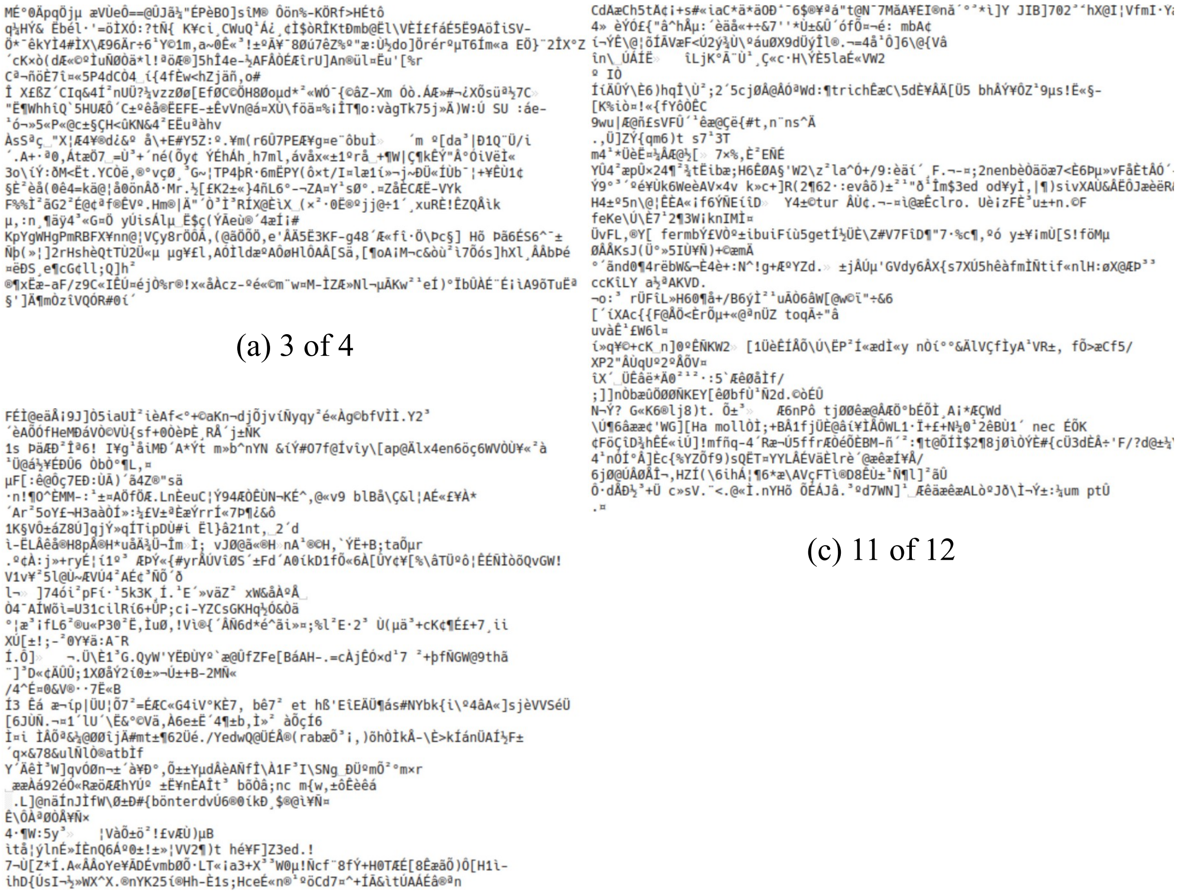
Table 4.9 Payload extraction attempt results for 8 carriers

NSHARES	Number of bytes extracted	Length of longest revealed payload
1	0	0
2	485 – 502	0
3	737 – 742	2
4	994 – 997	2
5	1,241 – 1,244	3
6	1,491 – 1,499	2
7	1,739 – 1,750	3
8	2,048	2,048

Table 4.10 Payload extraction attempt results for 12 carriers

NSHARES	Number of bytes extracted	Length of longest revealed payload
1	0	0
2	326 – 331	0
3	491 – 496	2
4	658 – 661	3
5	826 – 833	2
6	992 – 999	3
7	1,160 – 1,160	3
8	1,326 – 1,331	2
9	1,496 – 1,506	2
10	1,664 – 1,673	2
11	1,831 – 1,834	5
12	2,048	2,048

Although the number of extracted bytes may appear high, it is actually expected since every image is expected to get roughly an N^{th} of the payload. These extracted bytes mostly have no meaning and are unreadable, as they cannot correctly form ASCII letters because of missing bits, as explained previously. The most critical point here is the lengths of pieces of successfully extracted payload. The results in Table 4.8 to Table 4.10 show that the chance of revealing a meaningful payload is very low, even when only a single carrier is missing during extraction. Out of all extraction results, three sample outputs where only one carrier was missing from each carrier set are presented in Figure 4.7.



(a) 3 of 4

(c) 11 of 12

(b) 7 of 8

Figure 4.7 Sample faulty extraction results obtained with single missing carriers

5. DISCUSSIONS

This chapter will comment on the produced outputs and evaluation results for the methods that produce halftone carriers. Several results have been obtained from both objective and subjective evaluations; some of these are common for both evaluations, but results unique to each evaluation have also been obtained. Results gained from both evaluations are consistent with each other. There are no outliers in either evaluation, and the scores tend to get higher or lower together in both evaluations.

One shared result gained from objective and subjective evaluations is that the perception of the payload in color carriers is always lower than their binary counterparts. From the subjective evaluation perspective, this can mainly be attributed to the contrast among neighboring pixels. In binary halftone images, each pixel can be represented as either white or black, which results in high contrasts among pixels. On the other hand, color halftone images often have visually similar neighboring pixels (e.g., yellow and white, red and magenta), which contributes to lower contrast values. From these findings, it can be deduced that the HVS is more prone to detecting noises caused by the payload in binary images. Therefore, color halftone images score higher.

One result unique to the subjective evaluation is the difference in scores between carriers according to their generation method. When compared to carriers generated with error diffusion, the overall percentage of payload visibility is lower in carriers generated with patterns. This difference can be attributed to how many pixels are affected for each payload bit. In carriers generated with error diffusion, every pixel is responsible for whether to receive a payload bit or not. However, pixels in images generated with patterns can not be considered alone, as they belong to groups of 3x3 pixels, and each payload bit flips only one pixel in the pattern. Because of this, flipped pixels can get close in error diffusion images, resulting in getting lower scores by observers.

Another significant result obtained from the subjective evaluation is that the carriers with large regions of heterogeneous textures (e.g., forest22) can hide the provided payloads better, as it becomes challenging to distinguish pixels carrying payload bits by the HVS in such images. This finding is supported by the fact that scores of carriers with less heterogeneous regions (e.g., airplane80) are lesser.

Generally, the results that are presented in Figure 4.1 to Figure 4.4 and Table 4.3 to Table 4.6 show that the quality of the generated stego carriers increases when any of the following conditions meet:

- shorter payloads are hidden,
- number of carrier shares increases,
- color carriers are chosen rather than binary images.

In conclusion, it can be deduced from both the objective and subjective evaluations that color halftone images generated with patterns are the most suitable carriers for the proposed steganography algorithm for halftone images.

6. CONCLUSION

This study proposes new steganography methods that use binary images as carriers for plaintext payloads. The proposed methods can hide the provided payloads into either binary images that carry text data (such as banners) or halftone images generated from grayscale and color sources. The method that hides in images carrying visual text data encrypts and hides the payload into outputs visually similar to materials printed through ecology-friendly printing appearing as tiny holes. The other methods that hide the payloads in halftone images distribute their payloads into multiple copies of the generated output. The carriers can be generated either via predefined patterns or error diffusion methods. In either case, the payloads hidden in these images can be extracted when all the produced outputs are gathered back together again only.

The carriers with visual text data can be transferred digitally or printed with laser printers for sharing, but printing them with inkjet printers will cause the payload to be lost; therefore, this should be avoided. Evaluations made for halftone carriers show the methods can produce outputs that can be considered successful and satisfying both statistically (from objective evaluations that scored at least 90% successful) and visually (from subjective evaluations that scored as high as 93% successful) for payloads occupying up to 50% of their carriers' capacity. These generated carriers can be shared among several individuals where each individual gets a different piece of the payload hidden in the carrier they have received. Additional security tests conducted for halftone carriers show that the carriers resist unwanted payload extraction without the complete carrier sets.

Although the halftone carriers already performed successfully during initial evaluations, their quality can be improved in a further study by determining safe payload lengths for provided carriers before hiding. Automatic selection of appropriate carriers before hiding the payload can also have positive outcomes. Finally, since digital videos consist of sequences of images, it is possible to apply the proposed steganography methods to digital videos created using binary images, as long as the videos are stored without compression, and the pixels in each frame retain their original values.

REFERENCES

[1] D. Davies, “A brief history of cryptography,” Information Security Technical Report, vol. 2, no. 2, pp. 14–17, 1997.

[2] P. Kadian, S. M. Arora, and N. Arora, “Robust digital watermarking techniques for copyright protection of digital data: A survey,” Wireless Personal Communications, vol. 118, no. 4, pp. 3225–3249, 2021.

[3] G. J. Simmons, “The prisoners’ problem and the subliminal channel,” in Advances in Cryptology. Springer, 1984, pp. 51–67.

[4] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, “Digital image steganography: Survey and analysis of current methods,” Signal processing, vol. 90, no. 3, pp. 727–752, 2010.

[5] A. Kumar and K. Pooja, “Steganography-a data hiding technique,” International Journal of Computer Applications, vol. 9, no. 7, pp. 19–23, 2010.

[6] T. Morkel, J. H. Eloff, and M. S. Olivier, “An overview of image steganography.” in ISSA, vol. 1, no. 2, 2005, pp. 1–11.

[7] D.-C. Lou, J.-L. Liu, and H.-K. Tso, “Evolution of information-hiding technology,” in Information Security and Ethics: Concepts, Methodologies, Tools, and Applications. IGI Global, 2008, pp. 144–154.

[8] S. Bansod and G. Bhure, “Data encryption by image steganography,” Int. J. Inform. Comput. Technol. Int. Res. Publ. House, vol. 4, pp. 453–458, 2014.

[9] S. Roy and M. Manasmita, “A novel approach to format based text steganography,” in proceedings of the 2011 international conference on communication, Computing & Security, 2011, pp. 511–516.

[10] K. S. K. Arumugam, “Covert communication over multi-user channels,” Ph.D. dissertation, Georgia Institute of Technology, 2019.

[11] BT, R. I. R. et al., “Studio encoding parameters of digital television for standard 4: 3 and wide-screen 16: 9 aspect ratios,” Int. Radio Consultative Committee Int. Telecommun. Union, Switzerland, CCIR Rep, pp. 624–4, 2011.

[12] R. Ulichney, Digital halftoning. MIT press, 1987.

[13] D. L. Lau and G. R. Arce, Modern digital halftoning. CRC Press, 2018.

[14] Y. Zhang et al., Image processing. Walter de Gruyter GmbH & Co KG, 2017.

[15] B. E. Bayer, “An optimum method for two-level rendition of continuous tone pictures,” in IEEE International Conference on Communications, June, 1973, vol. 26, 1973.

[16] R. W. Floyd, “An adaptive algorithm for spatial gray-scale,” in Proc. Soc. Inf. Disp., vol. 17, 1976, pp. 75–77.

[17] J. F. Jarvis, C. N. Judice, and W. Ninke, “A survey of techniques for the display of continuous tone pictures on bilevel displays,” Computer graphics and image processing, vol. 5, no. 1, pp. 13–40, 1976.

[18] J.-N. Shiau and Z. Fan, “Set of easily implementable coefficients in error diffusion with reduced worm artifacts,” in Color Imaging: Device-Independent Color, Color Hard Copy, and Graphic Arts, vol. 2658. SPIE, 1996, pp. 222–225.

[19] P. Stucki, “Mecca: a multiple-error correction computation algorithm for bi-level image hardcopy reproduction,” Ph.D. dissertation, Verlag nicht ermittelbar, 1981.

[20] S. Arivazhagan, W. S. L. Jebarani, S. A. Roy, and E. Amrutha, “Improving quality of stego images through dithering techniques for pixel pair matching steganographic schemes.”

[21] R. Milošević, U. Nedeljković, B. Banjanin, D. Novaković, and N. Kašiković, “the analysis of ink jet printed eco-font efficiency,” Journal of Graphic Engineering and Design, vol. 7, no. 1, p. 13, 2016.

[22] "Ecofont: easy earnings, good cause," Ecofont, 2017. Accessed: Aug. 12, 2022. [Online]. Available: https://www.ecofont.com/

[23] A. D. Ker, “Improved detection of lsb steganography in grayscale images,” in International workshop on information hiding. Springer, 2004, pp. 97–115.

[24] D. Neeta, K. Snehal, and D. Jacobs, “Implementation of lsb steganography and its evaluation for various bits,” in 2006 1st international conference on digital information management. IEEE, 2006, pp. 173–178.

[25] W. Luo, F. Huang, and J. Huang, “Edge adaptive image steganography based on lsb matching revisited,” IEEE Transactions on information forensics and security, vol. 5, no. 2, pp. 201–214, 2010.

[26] C. Wang, W. Zhang, J. Liu, and N. Yu, “Fast matrix embedding by matrix extending,” IEEE Transactions on Information Forensics and Security, vol. 7, no. 1, pp. 346–350, 2011.

[27] R. Dumre and A. Dave, “Exploring lsb steganography possibilities in rgb images,” in 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT). IEEE, 2021, pp. 1–7.

[28] K. Kordov and S. Zhelezov, “Steganography in color images with random order of pixel selection and encrypted text message embedding,” PeerJ Computer Science, vol. 7, p.e380, 2021.

[29] L. Voleti, R. Balajee, S. K. Vallepu, K. Bayoju, and D. Srinivas, “A secure image steganography using improved lsb technique and vigenere cipher algorithm,” in 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS). IEEE, 2021, pp. 1005–1010.

[30] M. Kumar, A. Soni, A. R. S. Shekhawat, and A. Rawat, “Enhanced digital image and text data security using hybrid model of lsb steganography and aes cryptography technique,” in 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS). IEEE, 2022, pp. 1453–1457.

[31] D. Darwis, A. T. Priandika, A. Surahman, A. F. O. Pasaribu, A. Junaidi et al., “Combination of modified lsb steganography and huffman compression for data security,” in 2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE). IEEE, 2021, pp. 218–224.

[32] N. Akhtar, P. Johri, and S. Khan, “Enhancing the security and quality of lsb based image steganography,” in 2013 5th International Conference and Computational Intelligence and Communication Networks. IEEE, 2013, pp. 385–390.

[33] M. Sutaone and M. Khandare, “Image based steganography using lsb insertion technique,” in 2008 IET International Conference on Wireless, Mobile and Multimedia Networks. IET, 2008, pp. 146–151.

[34] L. Cruz, B. Patrão, N. Gonçalves, O. Diamanti, and A. Vaxman, “Halftone pattern: A new steganographic approach.” in Eurographics (Short Papers), 2018, pp. 21–24.

[35] H. Cao and A. C. Kot, “On establishing edge adaptive grid for bilevel image data hiding,” IEEE transactions on information forensics and security, vol. 8, no. 9, pp. 1508–1518, 2013.

[36] W. Lu, Y. Xue, Y. Yeung, H. Liu, J. Huang, and Y.-Q. Shi, “Secure halftone image steganography based on pixel density transition,” IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 3, pp. 1137–1149, 2019.

[37] Y. Xue, W. Liu, W. Lu, Y. Yeung, X. Liu, and H. Liu, “Efficient halftone image steganography based on dispersion degree optimization,” Journal of Real-Time Image Processing, vol. 16, no. 3, pp. 601–609, 2019.

[38] M. Yu, X. Yin, W. Liu, and W. Lu, “Secure halftone image steganography based on density preserving and distortion fusion,” Signal Processing, vol. 188, p. 108227, 2021.

[39] A. Shamir, “How to share a secret,” Communications of the ACM, vol. 22, no. 11, pp. 612–613, 1979.

[40] M. Naor and A. Shamir, “Visual cryptography,” in Workshop on the Theory and Application of of Cryptographic Techniques. Springer, 1994, pp. 1–12.

[41] M. S. Fu and O. C. Au, “Steganography in halftone images: conjugate error diffusion,”Signal Processing, vol. 83, no. 10, pp. 2171–2178, 2003.

[42] S.-C. Pei and J.-M. Guo, “Data hiding in halftone images with noise-balanced error diffusion,” IEEE signal processing letters, vol. 10, no. 12, pp. 349–351, 2003.

[43] J. Rosen and B. Javidi, “Hidden images in halftone pictures,” Applied Optics, vol. 40, no. 20, pp. 3346–3353, 2001.

[44] X. Yan, S. Wang, X. Niu, and C.-N. Yang, “Halftone visual cryptography with minimum auxiliary black pixels and uniform image quality,” Digital Signal Processing, vol. 38, pp. 53–65, 2015.

[45] Z. Wang and G. R. Arce, “Halftone visual cryptography through error diffusion,” in 2006 International Conference on Image Processing. IEEE, 2006, pp. 109–112.

[46] Y. Yang and S. Newsam, “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, 2010, pp. 270–279.

[47] Y. Xu, J. Li, J. Feng, H. Zhang, W. Xu, and J. Duan, “Lévy noise-induced stochastic resonance in a bistable system,” The European Physical Journal B, vol. 86, no. 5, pp. 1–7, 2013.

[48] U. Sara, M. Akter, and M. S. Uddin, “Image quality assessment through fsim, ssim, mse and psnr—a comparative study,” Journal of Computer and Communications, vol. 7, no. 3, pp. 8–18, 2019.

[49] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” IEEE transactions on image processing, vol. 13, no. 4, pp. 600–612, 2004.

APPENDIX

APPENDIX 1: THRESHOLDING ALGORITHM

```
Input: source image I, threshold value T
Output: binary image B

for i = 1 to $I_{height}$, do
  for j = 1 to $I_{width}$, do
    if $I_{i,j} < T$, then
      $B_{i,j} = 0$
    else
      $B_{i,j} = 1$
    end if
  end for
end for
```

APPENDIX 2: TEXT CARRIER METHOD ALGORITHM

```
Input: source image I with alphanumerical letters, payload P
Output: stego carrier I

cl = number of alphanumerical characters in the carrier
bp = total number of usable black pixels in characters after erosion
sd = bp / length($P_{binary}$) % distance between payload bits

for i = 1 to cl, do
  erode next alphanumerical character in I
  for j = 1 to sd, do
    embed next payload bit
    skip sd pixels
  end for
  restore character
end for
```

APPENDIX 3: HIDING ALGORITHM FOR PATTERN CARRIERS

```
Input: source image I, payload P, number of output shares N
Output: a set of stego images R

let R be a set of N empty images
pbLen = $P_{binary}$.length
dLen = I.capacity / pbLen
if dLen < 1, then                     % terminate if payload is
  return                              % larger than carrier capacity
end if

for each dLen-long blocks in I as block, do
  c = $R_{random}$                          % choose a random share
  l = $block_{random}$                      % choose a random pixel in block
  b = $P_{binary}$.next
  $R_{c,block}$ = embed()                   % hide payload bit in chosen pixel
end for
```

APPENDIX 4: PAYLOAD EXTRACTION ALGORITHM FOR PATTERN BASED CARRIERS

Input: a set of stego carriers S

Output: extracted payload P

```
Let P be empty string
let C be empty pattern set
for i = 1 to $S_{any}$.height, do
  for j = 1 to $S_{any}$.width, do
    for k = 1 to $S_{any}$.channels, do
      C ← Ø
      for l = 1 to S.length, do
        $C_{l,k}$ ← $S_{l,i,j,k}$
      end for
      if count(unique(C)) = 2, then
        if count(prev(C)) > count(next(C)), then
          P += "1"
        else
          P += "0"
        end if
      end if
    end for
  end for
end for
return ASCII(P)
```

APPENDIX 5: PAYLOAD EXTRACTION ALGORITHM FOR ERROR DIFFUSION CARRIERS

```
Input: a set of stego carriers S
Output: extracted payload P

Let P be empty string
let C be empty pixel set
for i = 1 to $S_{any}$.height, do
  for j = 1 to $S_{any}$.width, do
    for k = 1 to $S_{any}$.channels, do
      C ← ∅
      for l = 1 to S.length, do
        $C_{l,k} \leftarrow S_{l,i,j,k}$
      end for
      if count(unique(C)) = 2, then
        if count(black(C)) > count(white(C)), then
          P += "1"
        else
          P += "0"
        end if
      end if
    end for
  end for
end for
return ASCII(P)
```

APPENDIX 6: THE LIST OF THE ARTICLES PUBLISHED AND THE CONFERENCE PROCEEDINGS PRESENTED WITHIN THE SCOPE OF THE THESIS STUDY

ARTICLES

E. Çiftci and E. Sümer, “A novel steganography method for binary and color halftone images,” PeerJ Computer Science, vol. 8, p. e1062, 2022. DOI: 10.7717/peerj-cs.1062 (SCI-E)

CONFERENCE PROCEEDINGS

E. Çiftci and E. Sümer, “A novel steganography method for halftone images,” in 2022 30th Signal Processing and Communications Applications Conference (SIU). IEEE, 2022, pp. 1–4. DOI: 10.1109/SIU55565.2022.9864763