T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

A NEW LSB BASED IMAGE STEGANOGRAPHY METHOD

Asmaa Hakeem Naeem AL MAGED

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

09-2023
KONYA

TEZ KABUL VE ONAYI

Asmaa Hakeem Naeem AL MAGED tarafından hazırlanan “A New Lsb Based Image Steganography Method” adlı tez çalışması 19/09/2023 tarihinde aşağıdaki jüri tarafından oy birliği ile Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Prof. Dr. Halife KODAZ	…………………..
Danışman Assoc. Prof. Kemal TÜTÜNCÜ	…………………..
Üye Dr. Öğr. Üyesi Murat KÖKLÜ	…………………..

Yukarıdaki sonucu onaylarım.

Prof. Dr. ……. ……..
FBE Müdürü

Bu tez çalışması ……………. tarafından ………… nolu proje ile desteklenmiştir.

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Asmaa Hakeem Naeem AL MAGED

Tarih:19.09.2023

ÖZET

YÜKSEK LİSANS TEZİ

GÖRÜNTÜ STEGANOGRAFİSİNDE LSB TABANLI YENİ BİR YÖNTEM

Asmaa Hakeem Naeem AL MAGED

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Kemal TÜTÜNCÜ

2023, 100 **Sayfa**

Jüri
Prof. Dr. Halife KODAZ
Doç. Dr. Kemal TÜTÜNCÜ
Dr. Öğr. Üyesi Murat KÖKLÜ

Bu çalışma blok tabanlı En Küçük Önemli Bit (LSB) görüntü steganografisi için yeni yaklaşımlar önermektedir. Bilgi aktarımında hem güvenliği hem de algılanamazlığı artırmayı amaçlamaktadır. Resimlerin yönetilebilir 16x16 piksel ve 32x32 piksel parçalarına bölünmesine ve ardından verileri yerleştirmek için yüksek LSB, Mutlak Ortalama Parlaklık Sapması (AMBD) ve Entropi değerlerine sahip blokların seçilmesine dayanmaktadır. Veri gömme ise tek pariteye dayalı LSB ile sağlanmaktadır. Veriler seçilen blokların içerisinde yer alan piksellerin tek pariteyi sağlaması durumunda yalnızca yeşil kanallarına gömülmektedir. Ortalama Karesel Hata (MSE), Tepe Sinyal Gürültü Oranı (PSNR), Evrensel Kalite Endeksi (UQI), Yapısal Benzerlik Endeksi (SSIM), Sürekli Kalite Ölçümü (CQM), Mutlak Fark (AD), Yapısal İçerik (SC), Normalize Çapraz Korelasyon (NCC) ve Normalize Mutalk Hata (NAE) metrikleri bu yöntemlerin performansını analiz etmek için kullanılmıştır. Veri kapasitesine karşılık algılanmazlıkta oluşan değişimi gözlemleyebilme amaçlı olarak LSB, AMBD ve Entropi yöntemlerine dayalı blok seçimleri, 30 byte, 200 byte, 400 byte ve 800 byte metin mesajlarını 7 farklı görüntüye LSB tabanlı yöntemle gömme için kullanılmıştır. Literatürde olduğu gibi görüntünün orijinal boyutları ne olursa olsun, metin verilerini gömmek için 256x256 piksel, 512x512 piksel ve 1024x1024 piksele ölçeklendirilmiştir. Sonuçlar, veri yerleştirme için LSB, AMBD ve Entropi tabanlı blok seçiminin PSNR ve MSE açısından literatürden daha iyi ve birbirine benzer sonuçlar verdiğini göstermiştir.

Anahtar Kelimeler: AMBD, Entropi, LSB, görüntü steganografisi, PSNR, MSE

ABSTRACT

MS THESIS

A NEW LSB BASED IMAGE STEGANOGRAPHY METHOD

Asmaa Hakeem Naeem AL MAGED

THE DEGREE OF MASTER OF SCIENCE IN COMPUTER ENGINEERING

Advisor: Assoc. Prof. Dr. Kemal TÜTÜNCÜ

2023, 97 Pages

Jury
Prof. Dr. Halife KODAZ
Assoc. Prof. Dr. Kemal TUTUNCU
Assist. Prof. Dr. Murat KOKLU

This paper proposes new approaches for block-based Least Significant Bit (LSB) image steganography. It aims to improve security, imperceptibility and data capacity in information transmission. It is based on dividing images into manageable 16x16 pixel and 32x32 pixel chunks and then selecting blocks with high LSB, Absolute Mean Brightness Deviation (AMBD) and Entropy values to embed data. Data embedding is achieved by LSB based on single parity. Data is embedded only in the green channels of the selected blocks if the pixels in the selected blocks fulfil single parity. (Mean Square Eror) MSE, Peak signal-to-noise ratio (PSNR), Universal Quality Index (UQI), Structural Similarity Index (SSIM), Continuous Quality Measure (CQM), Absolute Difference (AD), SC, (Normalized Cross-Correlation) NCC and (Normalized Absolute Error) NAE metrics are used to analyse the performance of these methods. In order to observe the change in imperceptibility with respect to data capacity, block selections based on LSB, AMBD and Entropy methods were used to embed 30byte, 200byte, 400byte and 800byte text messages into 7 different images using LSB based method. As in the literature, whatever the original dimensions of the image are, they are scaled to 256x256 pixels, 512x512 pixels and 1024x1024 pixels to embed the text data. The results show that LSB, AMBD and Entropy based block selection for data embedding gives better and similar results than the literature in terms of PSNR and MSE.

Keywords: AMBD, Entropy, LSB, image steganography, PSNR, MSE.

ÖNSÖZ

İlk önce, hem kötü hem de iyi zamanlarımda beni destekledikleri için aileme teşekkür ederim. Ayrıca, danışmanım Doç. Dr. Kemal TÜTÜNCÜ'ye engin bilgisi, desteği ve sabrından dolayı teşekkür ederim. Son olarak, diğer arkadaşlarıma tez çalışmam süresince bana moral ve destek oldukları için ve yapıcı çalışma ortamı sağladıkları için teşekkür ederim.

Asma Hakeem Naeem AL MAGED
KONYA- 2023

İÇİNDEKİLER

SİMGELER VE KISALTMALAR

Kısaltmalar

ABC:	Artificial Bee Colony
AES:	Advanced Encryption Standard
ASCII:	American Standard Code for Information Interchange
AMBD:	Absolute Mean Brightness Deviation
AD:	Absolute Difference
BMP:	Microsoft Windows Bitmap
BWT:	Burrows Wheeler Transform
CPU:	Computer Processing Unit
CQM:	Continuous Quality Measure
dB:	Decibel
DPI:	Pixels Per Inch
ECC:	Elliptic Curve Cryptography
JPEG:	Joint Photographic Experts Group
LSB:	Least Significant Bit
LZW:	Lempel–Ziv–Welch
MSE:	Mean Squared Error
MP3:	MPEG-1 Audio Layer 3
MSB:	Most Significant Bit
MSSIM:	Mean Structural Similarity Index
NCC:	Normalized Cross-Correlation
NAE:	Normalized Absolute Error
PMM:	Pixel Mapping Method
PNG:	Portable Network Graphic
PPI:	Pixels Per Inch
PSNR:	Peak Signal-to-Noise Ratio
PVD:	Pixel Value Differencing
PLS:	Pixel Array Embedding
RGB:	Red-Green-Blue
RPE:	Random Pixel Embedding
RS:	Regular-Singular
RSA:	Rivest Shamir Adleman
SC:	Structural Content
SSIM:	Structural Similarity Index
STW:	Spatial-Orientation Tree Wavelet
TIFF:	Tagged Image File Format
UQI:	Universal Quality Index

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Introduction

Nowadays, ensuring the security of data and the transmission of information via the Internet has become a critical and significant concern. As technology advances, and various methods of storing and exchanging information, commonly known as network data transmission, become more prevalent, it is essential to safeguard this information from potential threats posed by hackers and attackers (Neamah et al., 2020). Consequently, there is a data to prevent unauthorized access by intruders. Additionally, it has become increasingly crucial to guarantee the reliability and security of these communications (Subramanian et al., 2021).

There are many areas of security technology that deals with the protection of secret data; the most important of these methods are cryptography and steganography (Neamah et al., 2020). Cryptography is the study of secure information communication, involving encryption and decryption processes. Encryption transforms plain text into unreadable cipher text using an encryption key, while decryption reverses this process to retrieve the original message. It safeguards data from unauthorized access, ensuring its protection. However, the cipher text remains visible, and only those with the proper decryption key can accurately decipher the encrypted information (AL-Windawi and Al-Jarrah, 2017). While steganography is a method used to hide smaller multimedia data within larger multimedia data, ensuring that the presence of the hidden information remains undetectable. This process involves manipulating cover data, such as images, text, audio, or video files, to create a secret channel for transmitting messages. Steganography falls under the umbrella of information hiding, which includes practices like watermarking and data management. The main goal of steganography is to prevent the concealed data from being visible or apparent in any way within the cover medium (Bhattacharyya, 2011). The main difference between cryptography and steganography is that cryptography changes the data into a form that can't be understood, while steganography hides the data itself (AL-Windawi and Al-Jarrah, 2017).

Image steganography has seen significant advancements with digital technology, but the challenge lies in balancing data hiding capacity, security, and perceptual integrity of the cover image. Researchers are devoted to exploring innovative approaches to improve efficiency, robustness, and imperceptibility of steganography algorithms (Subramanian et al., 2021). This involves developing new methods, refining existing methods, and employing advanced strategies to achieve an optimal trade-off between embedding capacity and security. The ultimate goal is to create highly reliable and undetectable steganography methods that are virtually invisible to human observers. Ongoing research in image steganography holds the potential to transform secure communication, confidential data exchange, digital forensics, and other domains that rely on covert information transmission (Bhardwaj and Sharma, 2016).

1.2. Literature Review

Many research have been done in the field of image steganography, showing different developments and methods. Several notable studies have contributed to the comprehension and development of image steganography methods to hide data, which are illustrated below:

In 2016, the authors used Spatial-orientation Tree Wavelet (STW) compression and Rivest Shamir Adleman (RSA) cryptography to conceal both Red-Green-Blue (RGB) and gray-scale images. The MATLAB software was used to hide eight different images (4 RGB and 4 gray-scale images), but the size of the concealed data was not specified. Analysis showed a Peak signal-to-noise ratio (PSNR) improvement of almost 50% over the traditional Least Significant Bit (LSB) method, with values ranging from 77 to 83 Decibel (dB) in both RGB and gray-scale images (Rajput et al., 2016). Furthermore, another study concealed a message by applying Advanced Encryption Standard (AES) 256 bit encryption using the classical LSB method. They evaluated the performance of their method and the classical LSB method based on PSNR and Mean Square Error (MSE) criteria. The evaluation was conducted on two different colored images. The researchers concluded that their method offered improvements in terms of security and embedding capacity (Arora et al., 2016). While another researchers used the LZW method to compress the message, which was encrypted using AES and Elliptic curve cryptography (ECC) before embedding with the traditional LSB approach. They achieved a PSNR of 80 dB by hiding 32-bit messages on distinct pictures. Then, a message with a length ranging from 32 to 288 bits was buried on the same picture, yielding a PSNR of 77-81 dB. Furthermore, they state that the message is encrypted twice, providing double security (Manjula and Shivakumar, 2016). Whereas another research combined the classical LSB replacement method with the RSA encryption algorithm. They successfully concealed a 90x90 pixel gray-scale image within a 256x256 pixel color image, achieving a significant increase of 3 dB in PSNR

(56 dB) compared to the classical LSB method (Zhou et al., 2016). As well as another study introduced a new method based on the classical LSB replacement method. The RGB image was divided into 3 blocks, where the message bits were embedded by substituting the last bit of each channel. An 8-bit message was concealed, with the 9th bit determining the embedding direction (right or left). Experimental results on 10 different images demonstrated an enhancement in the PSNR value compared to the conventional LSB method (Sugathan, 2016). Finally, another study investigated a method that leverages the difference between the Most significant bits (MSBs). By comparing the difference between the 5th and 6th bits with the message bit, they manipulated the 5th bit to match the desired difference. The method's effectiveness was evaluated by concealing 32 kB of data within 2 gray-scale and 2 RGB images, resulting in higher PSNR values in comparison to existing MSB replacement-based methods (Islam et al., 2016).

In 2017, the authors combined the traditional LSB method with AES cryptography and Deflate compression. Random LSB embedding using a pseudo-random generator was employed instead of sequential embedding. Three existing methods were compared to the proposed approach based on histogram analysis. Their research successfully concealed 3264 bits of data in a 512x512 pixel color image. The study highlighted the superior security of their method compared to the other methods used (Chikouche and Chikouche, 2017). While, another study utilized chaos-based logistic mapping encryption in n-LSB image steganography. The secret message underwent one-dimensional logistic mapping encryption before applying the n-LSB method. This enhanced the Stego-image security, yielding a 6.6% higher PSNR compared to classical LSB. Their approach showcased the effectiveness of chaos-based encryption in steganography for improved security and quality (Cataltas and Tutuncu, 2017b). Whereas, another research explored the impact of various compression methods on LSB-based image steganography. They employed 4 compression algorithms to compress 3 distinct hidden messages in separate images. The deflate algorithm achieved a data embedding rate of 113% at 213.5% compression, resulting in a PSNR value of 60.29 dB (Cataltas and Tutuncu, 2017b). Lastly, another study proposed a new symmetric key-based image hiding method using LSB steganography. The method relied on pseudo-random keys generated from a 1D logistic map, serving as chaotic sequences, to determine pixel positions for hiding secret data. While the stego-image exhibited enhanced visual quality, it had a reduced payload capacity, and the system's robustness was significantly compromised when subjected to lossy compression (Rajendran and Doraipandian, 2017).

In 2019, authors utilized the Burrows Wheeler Transform (BWT) for image steganography, hiding confidential information based on block sequence. The method was applied to images of size 256x256 and 512x512. The proposed method achieved PSNR of 51.5-51.12 dB with a payload of 2359296 bits at three bits per pixel. The stego images showed high correlation with the originals, maintaining image quality and security. The proposed method offered high capacity embedding, improved PSNR, and low complexity (Thahab, 2019).

In 2020, the researchers in their research, data embedding was performed using a random method and the exclusive OR method in selected pixels based on a long stego key. The proposed method achieved a PSNR value of 49.2048 dB and an Structural Similarity Index (SSIM) value of 0.9999, indicating high image quality. The resistance criteria, such as Regular-singular (RS) value (0.011197) and relative Entropy value (0.00004), further demonstrated the effectiveness of the approach in resisting attacks (Rafat, 2020).

In 2021, authors implemented Pixel Array Embedding (PLS) in LSB steganography to enhance security. Data embedding was conducted on random pixels using LSB method, and the PLS file was encrypted with AES for additional protection. The developed system achieved MSE and PSNR values of 0.95186 and 48.34506, respectively. Furthermore, ChiSquare, RS, and Sample pairs tests were employed for steganalysis analysis of the system (Tiwari and Gangurde, 2021). While, another research the contact information was concealed within biomedical images using LSB method. The resulting stego image's feature vector was converted to quantitative data and stored in an Excel file. Using the Artificial bee colony (ABC) method, if the user information and quantities matched, the person's image was displayed on the screen. The system achieved an accuracy rate of up to 93.65% (Khudher, 2021). Finally, another researchers utilized one of the R, G, or B channels as an indicator within the RGB pattern and performed LSB-based data hiding in the other two channels. The data hiding process depended on the last 2 bit values of the indicator channel, determining whether data was hidden in both channels, only one channel, or not hidden in the other channels. The authors claimed enhanced security for their proposed method, although specific metrics such as PSNR, SSIM, and MSE values were not provided to evaluate the system's performance (Sasmal and Mula, 2021).

1.3. Research Problem

a. Steganography hides data within images, making it undetectable. However, current methods may lack robustness in preserving data integrity and detecting unauthorized modifications by third parties. Ensuring reliability and integrity poses a challenge in steganography.
b. The increasing reliance on digital communication and the widespread use of the internet highlight the need for enhanced data security during transmission.
c. One aspect of a steganography system is its capacity, which refers to the amount of information that can be concealed within a carrier. Increasing the capacity of a system is challenging due to the need to balance it with imperceptibility.
d. LSB concealing, used to hide secret messages, directly affects the least significant bits of picture pixels, reducing image quality and susceptibility to assaults. This reduces picture resolution and raises the risk of discovery and illegal access to the secret content.

1.4. Aim of The Thesis

The objectives of the thesis is to:

a. Utilizing color images, particularly aerial map images, to improve data capacity, security and imperceptibility in steganography.
b. Implements image block division for efficient data hiding, utilizing methods based on LSB, pixel variance and brightness deviation, and Entropy to select appropriate blocks. This approach maximizes data hiding effectiveness by leveraging image characteristics, resulting in improved overall performance.
c. Employing LSB with odd numbers and green channel to minimize noticeable changes in the cover image.
d. Evaluating the performance of the proposed approach using various image quality metrics to compare its effectiveness with previous studies.

1.5. The Scope of Thesis

This thesis will mainly focus on:

1. Using well known images including aerial map images as cover images for steganography embedding due to benchmark characteristics and large data size.
2. Dividing the chosen images into blocks of both 16x16 and 32x32 pixels, based on the image size, to facilitate analysis and data hiding.
3. Converting the text data to binary form and using LSB method, with a preference for odd numbers, to embed the data into the selected blocks. This approach minimizes noticeable changes in the cover image.
4. Assessing three methods for block selection: LSB method, AMBD method, and Entropy-based method. The evaluation focuses on their effectiveness in preserving image quality and ensuring secure data hiding.
5. Evaluating the proposed approach's performance using various image quality metrics, such as MSE, PSNR, Universal Quality Index (UQI), Mean Structural Similarity Index (MSSIM), CQM, Absolute Difference (AD), Structural Content (SC), Normalized Cross-Correlation (NCC), and Normalized Absolute Error (NAE), to determine the best results and compare them with previous studies.

1.6. Thesis Layout

This thesis consists of six chapters including this chapter. These chapters are structured as follows:

- Chapter 2 ("Theoretical Background"): This chapter provides an overview of steganography, covering its definition, types, and methods. It explores the various aspects of steganography, offering a solid foundation for understanding the field and its advancements.
- Chapter 3 ("Methodology"): This chapter provides an overview of steganography, covering its definition, types, and methods. It explores the various aspects of steganography, offering a solid foundation for understanding the field and its advancements.
- Chapter 4 ("Experimental Result and their Discussions"): This chapter contains the results of the proposed approach, data representations and their discussions.
- Chapter 5 ("Conclusion and Future Work"): Summary of the major conclusions and recommendations for the future work are included in this chapter.

2. THEORETICAL BACKGROUND

2.1. Introduction

The rapid advancement of information technologies has led to a significant increase in the collection, storage, and manipulation of numerical data. Among this data, a major portion consists of images obtained digitally in various information technologies. The emergence and popularity of mobile devices and cameras have resulted in the replacement of traditional analog images with digital ones. Additionally, the widespread usage of the internet has facilitated the effortless sharing, acquisition, and storage of digital images (Neamah et al., 2020).

This section is dedicated to discussing digital images, their types, and formats. It also emphasizes the significance of encryption (cryptography), watermarking, and steganography in the realm of data hiding. Furthermore, the purpose of utilizing steganography methods specifically for color and grayscale images, which serve as the fundamental focus of this thesis, will be explored.

2.2. Digital Images

A digital image is a representation of visual information that can be stored and processed using digital technology such as cameras and computers. It can be defined as a two-dimensional function f (x, y), where x and y represent spatial plane coordinates. The amplitude of f at any (x, y) coordinate corresponds to the gray level or intensity of the image at that point. When x, y and f have finite and discrete values, the image is referred to as a digital image (Subramanian et al., 2021).

Digital images is created through various methods, each employing a unique methods tailored to capture, process, and convert visual information into digital representations (AL-Windawi and Al-Jarrah, 2017). The most common methods to represent digital images are:

1. Photographs: images captured using digital cameras or film cameras with digital sensors. The process involves exposing the camera's sensor to light from the scene, which is converted into numerical values representing the intensity of light for each pixel (Subramanian et al., 2021).
2. Computer-generated Images: Images created entirely through computer graphics methods, involving the design of virtual 3D models, defining materials, lighting, and rendering to generate the final image (AL-Windawi and Al-Jarrah, 2017).

3. Scanned Images: Images created by digitizing physical documents, photographs, or artwork using scanners, which capture the image by scanning line by line and converting it into a digital representation (AL-Windawi and Al-Jarrah, 2017).
4. Graphics and Illustrations: Images created using graphic design software or drawing tools, involving the design and manipulation of shapes, lines, colors, and textures to produce logos, icons, illustrations, infographics, and digital artwork (AL-Windawi and Al-Jarrah, 2017).
5. Satellite and Remote Sensing Images: Images captured by sensors on satellites or aerial platforms, used in fields like geography, geology, agriculture, and environmental monitoring. Sensors capture electromagnetic radiation from the Earth's surface, and the collected data is processed to generate digital images providing valuable information about the Earth's features and conditions (AL-Windawi and Al-Jarrah, 2017).

Each component of a digital image, referred to as a pixel, represents a specific point with a finite position and value. A pixel is also known as a picture element, image element, pel, or pixel (Bhattacharyya, 2011). The digital image is composed of these pixels, which are visualized as dots. In Figure 2.1, the origin, situated at the upper left corner of the image, serves as the reference point where the *x* and *y* directions of the pixels begin. The position of pixels within the image is determined by their *x* and *y* sequence numbers, denoted as (x, y). The origin is located at (0, 0), which is also known as a coordinate on the two-dimensional plane. Mathematically, the coordinate of a pixel is expressed as $f(x, y)$. The digital image model can be mathematically represented which shown in equation 2.1.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ F(M-1,0) & F(M-1,1) & \dots & F(M-1,N-1) \end{bmatrix} \tag{2.1}$$

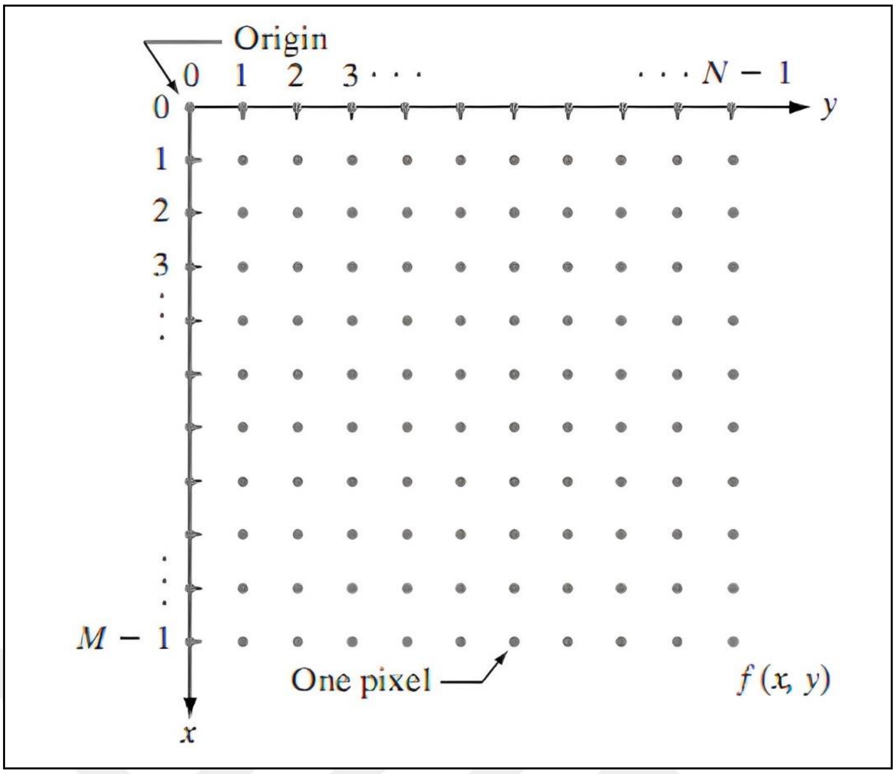


Figure 2.1. Digital image display model (Subramanian et al., 2021)

2.2.1. Digital image types

There are three general types of digital image:

A. Binary image

A binary image is a digital image with pixels that can be either 1 or 0, representing white and black regions respectively as can be seen in Figure 2.2. It is stored in memory as a bitmap using one bit per pixel (Bhardwaj and Sharma, 2016). Binary images simplify complex images into a binary representation, making them useful for tasks like object recognition and segmentation in image processing. By setting target region pixels to 1 and others to 0 through methods like binarization or thresholding, binary images provide a straightforward means to analyze the presence or absence of objects or features. Their simplicity, efficient storage, and ability to capture object information through their binary nature make them valuable in various image processing and steganography applications (Subramanian et al., 2021).

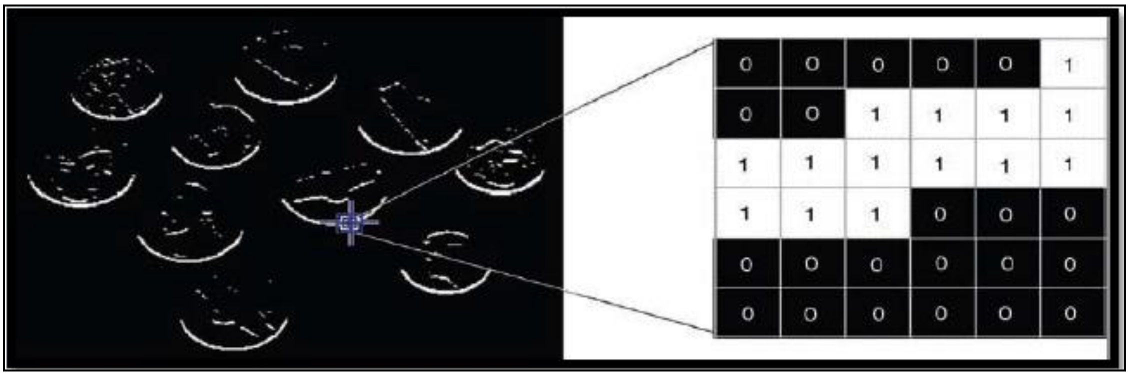


Figure 2.2. Binary image (Bhardwaj and Sharma, 2016).

B. Gray Scale Image

A grayscale image is a digital image with pixels representing shades of gray ranging from 0 to 255 as can be seen from Figure 2.3. The lower values correspond to darker shades, while higher values indicate brighter shades. Grayscale images are typically displayed in gray tones but can also be shown using alternative color schemes. Each pixel in a grayscale image holds a single sample, representing its brightness level (Bhardwaj and Sharma, 2016). With 256 different shades of gray, the transition between intensities is smooth and nuanced. Grayscale images find extensive use in applications like medical imaging and photography, effectively conveying contrast, texture, and depth details. Their 8-bit per pixel storage allows for a broad range of intensity values, enabling precise analysis and interpretation of visual content. Overall, grayscale images simplify complex information, focusing solely on brightness, and facilitating comprehensive steganography methods (Rajput et al., 2016).

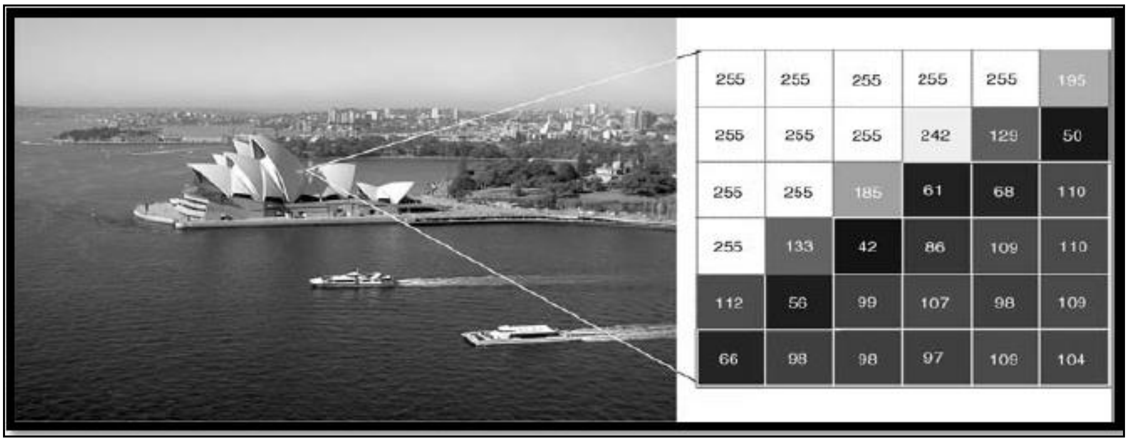


Figure 2.3. Gray scale image (Bhardwaj and Sharma, 2016)

C. RGB color image

In color images, the color value of each pixel is obtained through the combination of three primary colors: (red, green, and blue) RGB. These colors are mixed in varying proportions to produce a wide range of colors. Color images can be represented in different bit depths, such as 16-bit, 24-bit, or 32-bit data as can be seen in Table 2.1. In a 16-bit image, also known as a high-color image, each pixel has 5 bits for red, 6 bits for green, and 5 bits for blue, resulting in a total of 65,536 possible color combinations. True color images, represented as 24-bit data, allocate 8 bits (1 byte) to each color channel (red, green, and blue), allowing for 16,777,216 different colors. This has become the standard for color images, where each pixel requires 3 bytes of memory (Arora et al., 2016). In 32-bit color images, an additional 8 bits is used to represent transparency or opacity. This extra channel determines the level of transparency for each pixel. Color images are typically stored as raster maps, two-dimensional arrays of integer triples, or as separate raster maps for each color channel. In image steganography, the RGB format's extensive color range enables the discreet hiding of information within the image, leveraging the abundant color data available (Manjula and Shivakumar, 2016). A digital image can be described depending on the number of bits used for pixel representation in Table 2.2.

Table 2.1. Mixing ratios of the colour values of the pixels in the colour picture

NO.	Colour name	RGB triplet	Color
1	Black	(0, 0, 0)	
2	Green	(0, 128, 0)	
3	Gray	(128, 128, 128)	
4	Brown	(150, 75, 0)	
5	Orange	(255, 127, 0)	
6	Cyan	(0, 255, 255)	
7	Fuchsia	(255, 0, 255)	
8	Yellow	(255, 255, 0)	
9	White	(255, 255, 255)	

Table 2.2. The coloured area of the image corresponds to the depth a little bit

NO.	Image properties	Bits resolution	Color space
1	Binary image (black and white)	1	2 colors
2	Gray scale (monochrome)	8	256 gray levels
3	Colored image	8	256 colors
4	Colored image	16	65,536 colors
5	True color (RGB)	24 or 32	16,777,216 colors

2.2.2 Color Models

A greyscale image, also known as a panchromatic or black and white image, represents shades of gray without color as can be seen in Figure 2.4. A typical greyscale image has an 8-bit color depth, allowing for 256 different shades of gray. In contrast, a true color image has a 24-bit color depth, resulting in approximately 16 million colors. Greyscale images can have varying levels of gray, such as 16-bit images with 65,536 shades.

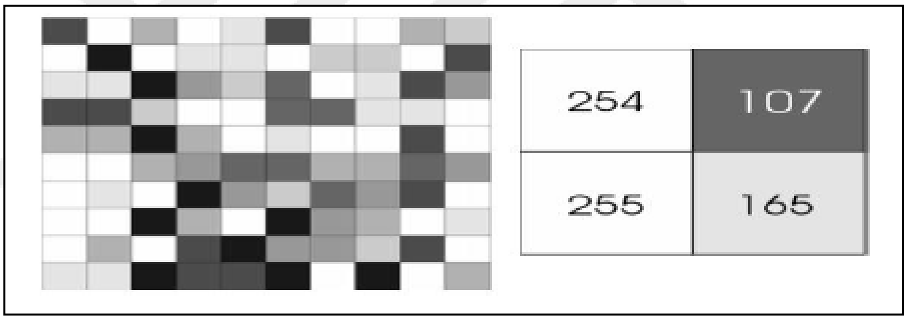


Figure 2.4. An 8 bit grayscale image (pixel value ranges between 0 (black) and 255 (white) (Sugathan, 2016)

The RGB colour model can be visualized using a colour cube, as illustrate in Figure 2.5. In this model, grayscale is represented by the line where R (red), G (green), and B (blue) values are equal. The RGB model is designed based on human colour perception and is commonly used for displaying images on monitors, TVs, and other colour systems (Islam et al., 2016).

The size of an RGB digital image remains constant and is determined by the number of bits used for quantization. For example, an 8-bit image has pixel values ranging from 0 to 255. In the RGB model, a value of 0 (at coordinate 0) signifies complete darkness or obscurity of colour, while a value of 1 (at coordinate 255) represents the maximum intensity of colour. The RGB values (0, 0, 0) correspond to black, while (1, 1, 1) corresponds to white (Zhou et al., 2016).

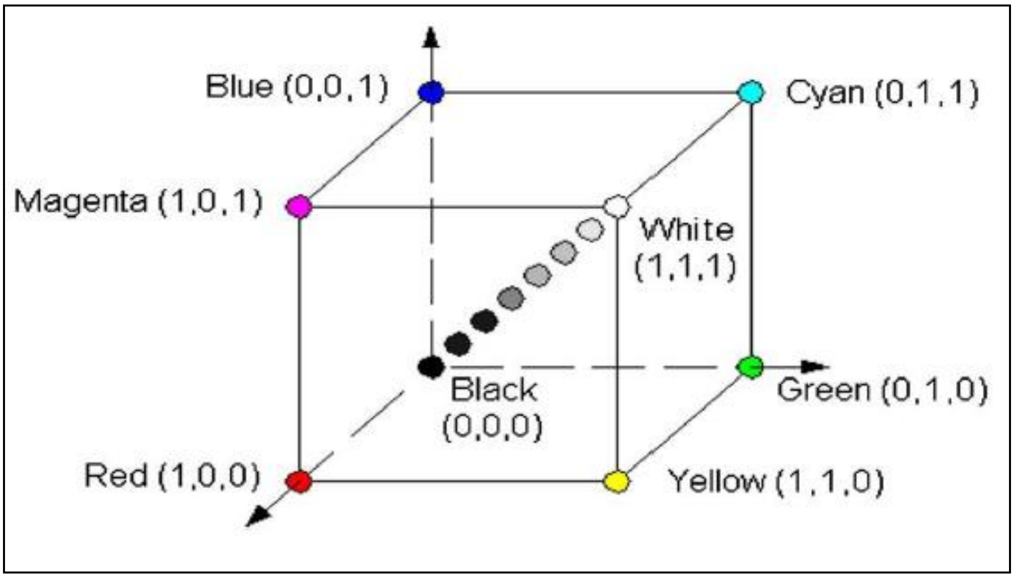


Figure 2.5. Colour cube used for representing true colour images (RGB) (Zhou et al., 2016)

The CMY colour model is derived from the merging of two of the primary colours, namely red (R), green (G), and blue (B). When the G and B colours are combined, we get cyan. When the R and B colours are merged, we obtain magenta, and when R and G are merged, we get yellow. The white colour is achieved by combining all three colours together (Islam et al., 2016). This concept is illustrated in Figure 2.6.

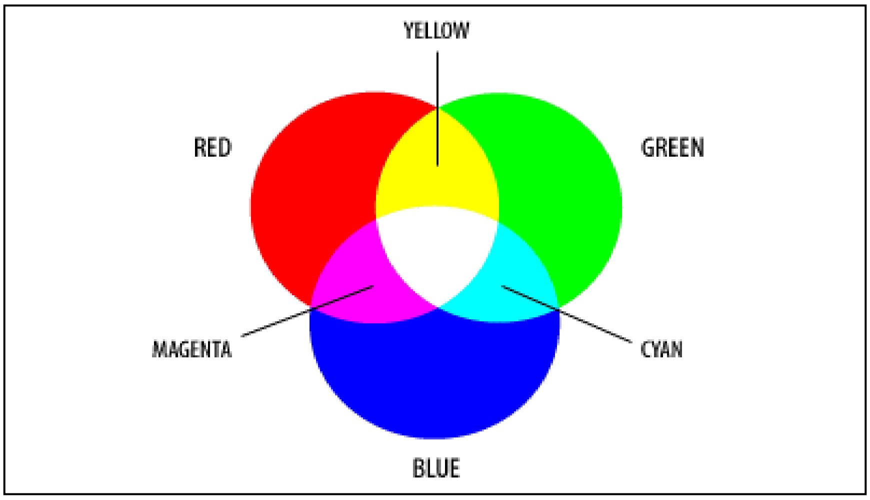


Figure 2.6. Merging the three primary colours in the RGB colour model to obtain the colours used in the CMY colour model (Sugathan, 2016)

2.2.3 Resolution Concept

Resolution refers to the level of detail present in a digital image. It is determined by the number of pixels that compose the image. In simpler terms, resolution indicates the image's width and height in terms of pixels (Chikouche and Chikouche, 2017). For instance, an image with a resolution of 448x336 implies that it consists of 448 pixels horizontally and 336 pixels vertically, resulting in a total of 150,528 pixels. Higher resolution corresponds to a closer representation of the actual image, as greater pixel density contributes to enhanced clarity. In summary, increasing resolution improves image quality and enhances visual fidelity (Subramanian et al., 2021).

When the resolution of a digital image is enhanced, the file size, which represents the memory space required to store the image, also increases. The calculation of the file size of the digital image is given in equation 2.2.

$$\text{File size} = (\text{horizontal pixels} \times \text{vertical pixels} \times \text{colour depth}) / 8 \qquad (2.2)$$

In digital imaging, sampling refers to the measurement of dots or pixels per inch (DPI or PPI) to express image resolution. Increasing the sampling frequency enhances image resolution. DPI signifies the number of dots or pixels within a one-inch region, serving as a scale for digital image resolution. For instance, an image with a 256x256 DPI has 256 dots per inch in both width and length (Cataltas and Tutuncu, 2017a). Figure 2.7 showcases the impact of different image sizes on various DPI values.

2.2.4 Cover Image Formats

Cover images can be stored in a variety of image file formats, which fall into two categories (lossless and loss formats). The following is a list of the most common image file formats classified into two distinct categories (Cataltas and Tutuncu, 2017a).

Figure 2.7. Effect of DPI value on resolution (a) 256×256 pixels 72dpi (b) 128×128 pixels 36dpi (c) 64×64 pixels 18 dpi (d) 32×32 9dpi (Subramanian et al., 2021)

2.2.4.1 Lossless image representation formats

Lossless image formats are file formats that maintain the original image quality without any loss of data (Cataltas and Tutuncu, 2017a). Here are three main lossless formats commonly used for cover images:

1. Microsoft Windows Bitmap (BMP): Developed by Microsoft, BMP is a file format that can store images with different color depths (1, 4, 8, or 24 bits per pixel). The image data is stored in a scan line order, from bottom to top (Sugathan, 2016).
2. Tagged Image File Format (TIFF): TIFF was introduced in 1986 and is widely used for its versatility. It supports monochrome, grayscale, and 8/24-bit color images. TIFF files can be saved in various color formats and compression methods. However, many graphics programs do not compress TIFF files, resulting in larger file sizes (Sugathan, 2016).
3. Portable Network Graphic (PNG): PNG was developed as a more advanced replacement for the GIF format. It offers support for an alpha channel (transparency) and the RGB color space. PNG files can contain up to 256 colors but utilize more

efficient color information storage. It also supports 8-bit transparency. PNG is suitable for palette-based, grayscale, and RGB images.

These formats ensure that the cover images retain their original quality and can be used in various applications without any loss of data or image degradation (Sugathan, 2016).

2.2.4.2 Lossy image compression formats

Lossy image compression formats are file formats that use compression methods that sacrifice some image details to achieve smaller file sizes (Cataltas and Tutuncu, 2017b). Here is the main lossy compression format commonly used for cover images:

1. Joint Photographic Experts Group (JPEG): JPEG is a widely used format for compressing images with 24-bit color depth or grayscale images. It employs a flexible compression method that allows adjusting the compression rate. When an image is compressed using JPEG, some information is permanently lost, resulting in a decrease in image quality. However, the compressed image file size is significantly smaller compared to the original. The JPEG standard offers two basic compression methods that can be tailored to meet the specific requirements of different applications (Sugathan, 2016).

It's important to note that while JPEG compression reduces file size, it also introduces some degree of image degradation. The level of compression applied can be adjusted to strike a balance between file size and acceptable image quality for a given application (Rajendran and Doraipandian, 2017). Summary of image file formats are listed in Table 2.3.

Table 2.3. Summary of image file formats

Image format	Name	Compression	Color Depth	Typical Usage
BMP	Microsoft Windows Bitmap	Lossless	1, 4, 8, 24	General-purpose, uncompressed images
TIFF	Tagged Image File Format	Lossless (with options for lossy compression)	1, 8, 24	Professional photography, high-quality image archiving
PNG	Portable Network Graphic	Lossless (with options for lossy compression)	1, 2, 4, 8, 16, 24, 48	Web graphics, transparent images, logos
JPEG	Joint Photographic Experts Group	Lossy	24	Digital photography, web images, social media sharing

2.3 Information Hiding

Throughout history, secret communication has been vital for humans. As technology advanced, the methods for concealing information evolved. In today's interconnected world, confidential communication has become increasingly important. Its primary objective is to deliver information to the intended recipient before it reaches unauthorized parties or becomes comprehensible to them. Cryptography, the science of encryption, transforms plain text into unreadable cipher text, ensuring the information's confidentiality (Thahab, 2019). However, encryption alone may not suffice. Data hiding methods like steganography and watermarking complement encryption. These methods, collectively known as data hiding, involve embedding information within other data or objects in a way that remains inconspicuous to observers. The goal is to protect the embedded data or securely transmit it without arousing suspicion. Information hiding finds applications in diverse areas such as bolstering security, safeguarding copyrights, managing digital rights, and enabling covert communication. Its purpose is to enhance privacy, protect sensitive information, and ensure the integrity of digital content (Rafat, 2020). Figure 2.8 show the data hiding used in applications requiring information security.

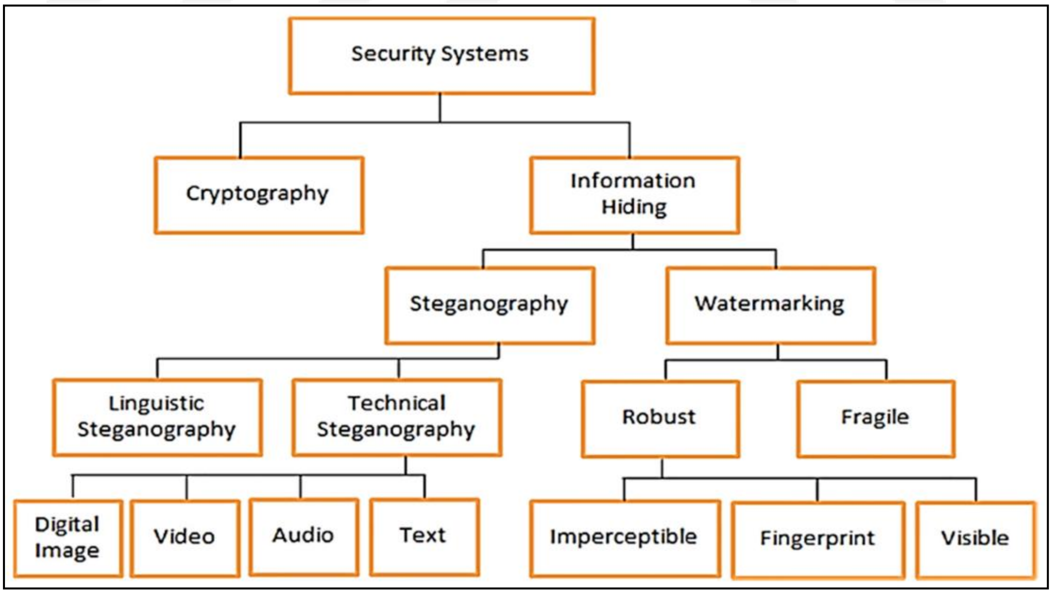


Figure 2.8. Classification tree of a general data security system (Tiwari and Gangurde, 2021)

2.4 Cryptography

Cryptography is the practice of securing communication and information by converting it into an unintelligible form using mathematical algorithms. It involves encryption, which transforms readable data into encoded cipher text, and decryption, which reverses the process to retrieve the original plaintext. Cryptography employs methods such as symmetric and asymmetric key algorithms, hash functions, and digital signatures to ensure data confidentiality, integrity, and authentication. With applications in secure communication, financial transactions, data privacy, and more, cryptography safeguards sensitive information, protecting it from unauthorized access and tampering in our interconnected digital world (Khudher, 2021).

2.5. Watermarking

Watermarking is a method used to embed imperceptible and persistent markers or identifiers into digital media, such as images, audio, video, or documents. The purpose of watermarking is to assert ownership, provide copyright protection, and track unauthorized use or distribution of the media. Watermarks can be visible or invisible, with visible watermarks serving as overt indicators of ownership and invisible watermarks designed to be undetectable to the human eye. By adding these markings, either through modifications to the media itself or by altering the media's properties, watermarking enables content creators and owners to assert their rights and deter unauthorized use, while allowing the media to remain functional and accessible for legitimate purposes (Sasmal and Mula, 2021).

2.6. Steganography

Steganography is a scientific topic as well as an artistic method for hiding information within an object. It operates on the concept that only the intended receiver should be aware of the hidden information, while others should be unconscious of its presence. The name "steganography" derives from the ancient Greek terms "steganos" for "covered" and "graphic" for "writing," which translates as "covered writing." This method of covert communication involves concealing a secret message behind another seemingly innocuous piece of information, referred to as a cover. The idea is to keep the hidden information hidden without raising any suspicions in sight of those who view it (Morkel et al., 2005). The primary goal of employing steganography is to avoid drawing attention to the transmission of secret information. However, if there is more uncertainty about the contents of a document, the purpose of concealing a hidden secret inside that document becomes less likely to fulfill its objectives because if an observer notices any change in the delivered document, he will attempt to discover the concealed information inside the document (Aydoğan, 2016).

2.6.1. Basic of steganography system

Steganography systems typically consist of several key components:

- Cover message: The cover message serves as the carrier or container for the secret message. It can be any form of digital media, such as an image, video, audio file, text document, or any other type of file (Dougherty, 1994).
- Secret message: The secret message is the information that needs to be concealed within the cover message. It could be sensitive data, confidential communication, or any other type of hidden information (Dougherty, 1994).
- Secret key: The secret key is used to control access to the hidden data. It is a piece of information or code that is required to encrypt or decrypt the secret message. The key ensures that only authorized individuals can extract or access the hidden information (Dougherty, 1994).
- Embedding algorithm: The embedding algorithm is the method used to hide the secret message within the cover message. It involves manipulating the cover message in a way that conceals the secret information while maintaining the overall appearance or functionality of the cover message. Various methods, such as LSB substitution or spread spectrum modulation, can be employed for embedding (Dougherty, 1994).
- Extracting algorithm: The extracting algorithm is the process used to retrieve the hidden data from the stego file, which is the modified cover message containing the secret message. This algorithm reverses the embedding process and extracts the concealed information using the secret key or specific extraction method. The extracting algorithm should accurately recover the secret message without any loss or distortion (Dougherty, 1994).

In a steganography system scenario, the sender begins by selecting the appropriate message carrier, such as an image, video, audio, or text file, followed by choosing effective secret messages and a strong password known to the receiver. They then carefully choose a suitable steganography method capable of securely encoding the message. Once prepared, the sender can transmit the stego file via email, chatting, or other modern methods, which serves as the cover document carrying the hidden message. Upon receiving the stego file, the receiver uses the same password and an extracting algorithm to decode the message, unveiling the secret information (Cataltas and Tutuncu, 2017a). Figure 2.9 illustrates the general steganography system involved in concealing and revealing hidden messages.

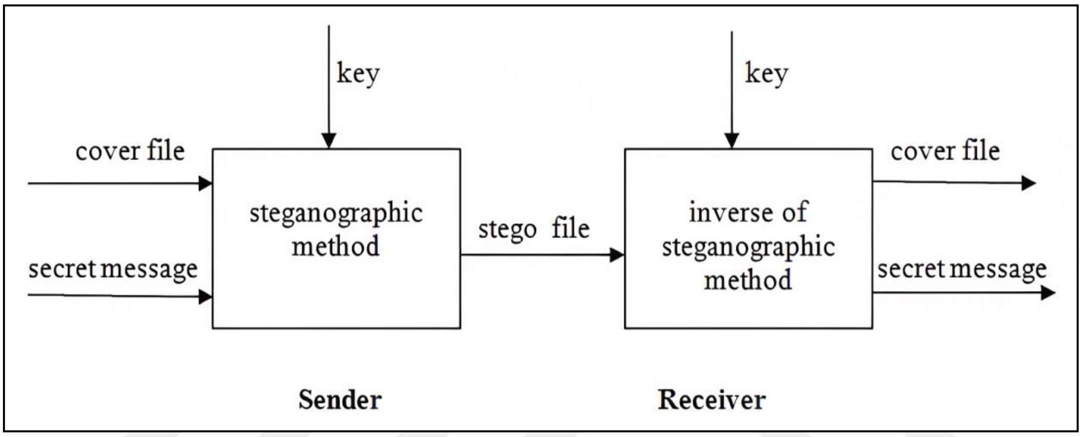


Figure 2.9. A generic steganography system (Gonzales and Wintz, 1987)

2.6.2. Type of steganography

Steganography embeds secret information into digital carriers such as text, image, audio, video, and so on, making it difficult for humans to recognize it. Steganography is classified into four kinds based on the carrier object used to insert the hidden data (Dougherty, 1994). Steganography is classified as follows:

A. Text steganography: involves hiding a secret message within a larger body of text, is a complex method. It is challenging because text files have minimal redundant data to conceal the hidden message effectively. The disadvantage lies in the ease with which unauthorized parties can modify the text or its format, compromising the integrity of the hidden information. To address this, one effective method is to add white spaces and tabs to the ends of lines in a document. Since these elements naturally occur in any text, they blend seamlessly and do not raise suspicion. This method helps maintain the covert nature of the hidden message by utilizing inconspicuous modifications.

B. Image steganography: is a widely used method that involves hiding information within the pixel intensity of an image. It is commonly used to hide text, audio, or video within images and is popular on the internet. The most common method is LSB insertion, where secret data bits are embedded by modifying the least significant bits of the pixel values. This modification is subtle and not easily noticeable. Other methods include masking, filtering, and using algorithms or transformations to hide images within other images or mediums. Overall, image steganography allows for covert communication and is widely employed in various fields (Dougherty, 1994).
C. Audio steganography: conceals secret messages within digital sounds by subtly altering the binary sequence of an audio file. It is more challenging than other forms of steganography. Various methods allow information embedding in audio formats like AU, WAV, MPEG-1 Audio Layer 3 (MP3), etc. Concealing messages in audio is harder than in images. Methods range from hiding information in noise patterns to advanced signal processing for accurate concealment. Audio steganography provides covert communication through manipulation of digital sound files (Dougherty, 1994).
D. Video steganography: involves hiding information within video files, utilizing methods that are applicable to both sound and image-based steganography. This system allows for the concealment of large amounts of data with minimal distortion. Using video files as carriers for steganography is advantageous and often preferred over other approaches. It offers a suitable platform for embedding secret messages, making it an effective method for covert communication and data hiding (Dougherty, 1994).

Steganography types can be assess based on capacity, security, integrity, robustness, transparency, and tamper resistance as represented in Table 2.4. Where capacity refers to hidden information volume, Security ensures protection against unauthorized access, Robustness gauges resistance to modifications without data compromise, Transparency concerns visibility of hidden information and Tamper resistance evaluates resilience against manipulation or destruction. These characteristics are crucial for comparing and evaluating steganography methods and systems (Islam et al., 2016).

Table 2.4. Comparison of different steganography methods

Techniques-Features	Text-steganography	Image-steganography	Audio-steganography	Video-steganography
Security	High	High	Low	High
Capacity	Low	High	Low	High
Transparency	Low	Low	Low	High
Integrity	Low	High	Low	Low
Temper resistance	High	High	High	High
Robustness	Low	High	Low	Low

2.7. Image Stenography Methods

Image steganography methods can be classified into two types:

A. Spatial domain method: Spatial domain involve direct manipulation of image pixels to achieve desired enhancements. These methods, including logarithmic transforms, power law transforms, and histogram equalization, rely on directly changing the pixel values in the image. By altering individual pixel values, these methods can effectively enhance the overall contrast of the entire image. This approach enables precise control over the pixel values and allows for targeted modifications to achieve the desired visual improvements (Cataltas and Tutuncu, 2017a). Spatial domain methods are broadly classified into LSB, Pixel value differencing (PVD), Edges based data imbedding (EBE), Random pixel embedding method (RPE), Pixel Mapping Method (PMM), Labeling or connectivity method (CONNECT), and Gray level value based method (GLV) (Tyagi, 2018).

B. Transform domain method: Transformation operations are utilized to enable data hiding in the frequency domain. This transformation yield image components, and the data hiding process focuses on manipulating the transformation coefficients associated with these components. These coefficients comprise zero and non-zero values, where the zeros indicate areas imperceptible to the human visual system, typically disregarded during lossy compression. On the other hand, non-zero values represent regions visible to humans, serving as prime candidates for data hiding. By embedding data within these perceptible regions, the hidden information remains resilient throughout subsequent transformations, regardless of whether they are lossy or lossless. Consequently, this frequency-based data hiding approach enhances the durability of the hidden data against image attacks (Cataltas and Tutuncu, 2017b). Transform domain methods are broadly classified into Discrete Cosine Transform (DCT) method, Discrete Fourier Transform (DFT) method, Discrete Wavelet Transform (DWT) method, Integer Wavelet Transform (IWT) method, and Discrete Curvelet Transform (DCVT) method (Tyagi, 2018).

2.8. Block-Based Image Steganography

Block-based image involves dividing an image into smaller square-shaped blocks, typically with a fixed size like 16x16 or 32x32 pixels. Instead of processing the entire image as a whole, operations are performed on each individual block separately. This approach is commonly used in image steganography, by dividing the image into blocks, we can embed or extract the secret information within each block independently (Heilbronner and Barrett, 2013). In figure 2.10 shows the division of the image into blocks.

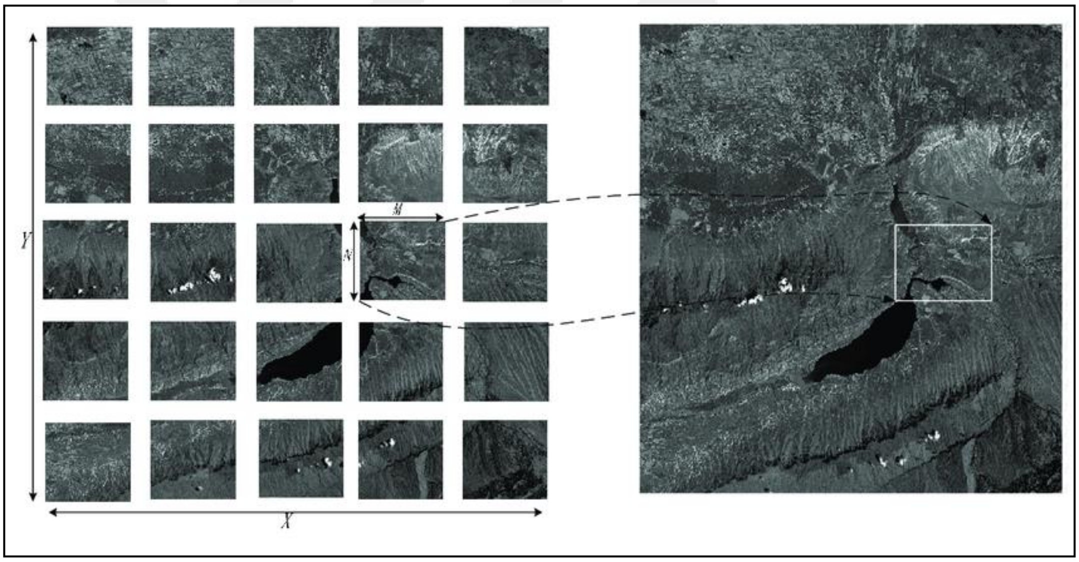


Figure 2.10. Block division schema (Rahman et al., 2023)

Block-based image in image steganography is an effective method that not only offers increased hiding capacity and reduced visual impact but also provides the flexibility to selectively determine which blocks to hide data inside. By dividing the image into blocks, steganography algorithms can strategically choose specific blocks to embed secret data, further enhancing security and covert communication. This targeted approach allows for efficient utilization of blocks and enables customization based on block characteristics, such as complexity or perceptual significance. Thus, block-based image steganography offers a powerful and adaptable method for concealing information within digital images (Heilbronner and Barrett, 2013).

2.9. LSB

LSB data hiding method involves altering LSB of the RGB values of the pixel data in an image by replacing it with a bit from the secret message. This approach takes advantage of the fact that the precision level of image formats is typically much higher than what can be perceived by the human eye. By modifying LSB, which has minimal visual impact, it becomes difficult for humans to detect the hidden data, making this method effective for concealing information within an image (Stanković et al., 2012).

In this method, LSB of each pixel in the cover image is replaced with a bit from the secret message after converting it into binary form. Although these changes cause slight numerical differences in the pixel values and result in subtle alterations in color, they are typically imperceptible to the human eyes. By utilizing the red, green, and blue color channels of each pixel in a 24-bit color image, it becomes possible to hide up to 3 bits of data. For example, Figure 2.11 demonstrates how the letter "A" can be hidden in three pixels of a 24-bit color image. The binary equivalent of the American Standard Code for Information Interchange (ASCII) value for "A" (001000001) is compared with LSB of each pixel, and the pixel values are modified accordingly as shown in Figure 2.11-b. In this example, the least significant bit of the 1st pixel, the red and blue color values of the 2nd pixel, and the green color value of the 3rd pixel are altered (Subramanian et al., 2021).

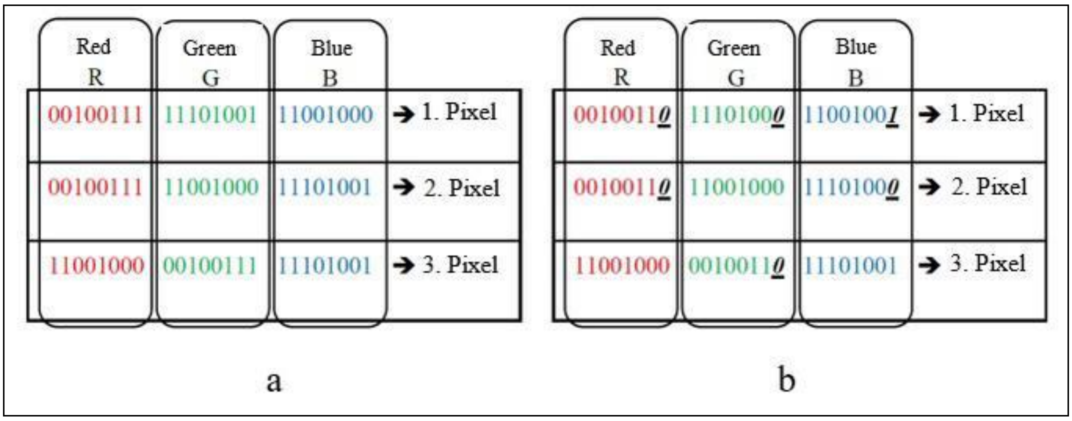


Figure 2.11. Hiding the letter ‘A’ in pixels of an image (a) bit values of pixels before data hiding, (b) bit values of pixels after data hiding (Subramanian et al., 2021)

The LSB method in image steganography has three important aspects: capacity, visibility, and robustness. It allows for a high data capacity but has a lower capacity due to using only one bit per pixel. It maintains good visibility, minimizing changes to the stego image's quality. However, it lacks robustness as the hidden data can be easily retrieved, and the presence of hidden information can potentially be detected by analyzing LSBs.

2.10. AMBD

The AMBD is a measure used in image steganography to identify pixels that are less likely to attract attention when modified. It is calculated by comparing the brightness value of each pixel to the average brightness of the entire image. The AMBD is obtained by calculating the average AD between the brightness value of each pixel and the average brightness of the image. Pixels with higher AMBD values have brightness values that closely align with the average brightness, making them less noticeable when modified (Shawkat, 2007).

To calculate AMBD, start by finding the average brightness (Avg) of the image after transform to gray-scale image as described in equation 2.3. This is done by adding up the brightness values of all the pixels and dividing the sum by the total number of pixels. Next, determine the AD between the brightness value of each pixel and the average brightness in equation 2.4. Summing up these ADs gives the SumDiff, which represents the overall deviation from the average brightness as shown in equation 2.5. Finally, divide the SumDiff by the total number of pixels to obtain AMBD value which shown in equation 2.6 (Ibraheem et al., 2012).

$$Avg = (1/n) * \sum(B(i)) \; for \; i = 1 \; to \; n \tag{2.3}$$

$$Diff(i) = |B(i) - Avg| \; for \; i = 1 \; to \; n \tag{2.4}$$

$$SumDiff = \sum(Diff(i)) \; for \; i = 1 \; to \; n \tag{2.5}$$

$$AMBD = SumDiff \,/\, n \tag{2.6}$$

The AMBD value serves as a measure of the extent to which each pixel's brightness deviates from the average brightness of the entire image. Higher AMBD values indicate pixels that are less likely to be visually noticeable when modified, making them suitable for embedding hidden information in image steganography (Ibraheem et al., 2012).

2.11. Entropy

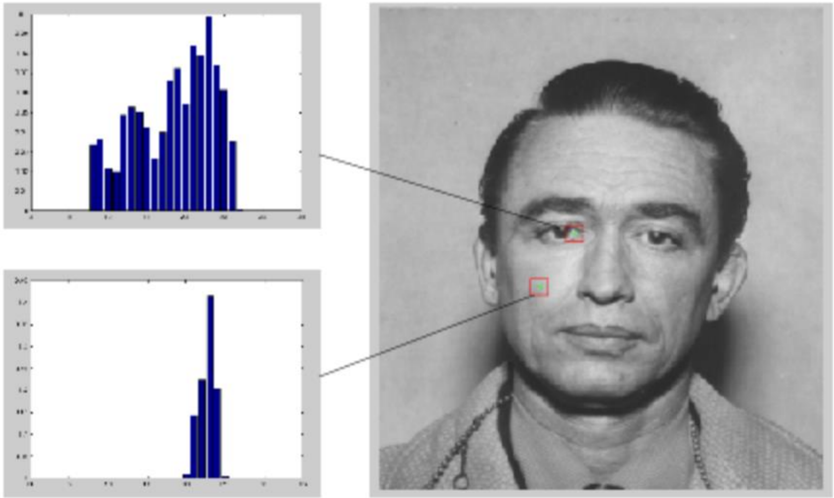
Entropy based embedding in an image involves a mechanism where information is concealed within the pixel values of an image using the concept of Entropy. Image Entropy plays a crucial role in quantifying the information content of an image. By characterizing the complexity and randomness of pixel intensity distribution, Entropy provides valuable insights into the visual intricacy and information richness of images. Its significance extends across multiple domains, including image processing and data hiding (Peterson and Wolffsohn, 2005).

To calculate Entropy of an image, the image is first converted to grayscale to simplify the process. It is then divided into smaller blocks, and for each block, a histogram is created to capture the frequency distribution of pixel intensities. The histogram is normalized to represent the probability distribution within the block. Using the normalized histogram, Entropy of each block is calculated using the Shannon Entropy formula as described in equation 2.7.

$$H(x) = -\sum_{x \in X} q(x) \log q(x) \tag{2.7}$$

Finally, the entropies of all the blocks are aggregated to obtain the overall Entropy of the image. Entropy calculation involves probability and logarithmic operations, with the formula representing the sum of the products of probabilities and their logarithms. This process provides insights into the information content and complexity of the image. In image steganography, high Entropy regions are often preferred for embedding hidden data as they possess increased randomness and complexity, making it harder to detect the presence of concealed information (Yalman, 2010).

For example, in Figure 2.12-a the intensities in the eye region are flatter, it suggests that the pixel values in that region are highly unpredictable. This corresponds to a high Entropy in the eye region, indicating a greater range of pixel intensity values and increased randomness. While in figure 2.12-b the intensities in the cheek region of an image are highly predictable, it indicates a low Entropy in that specific region. This means that the pixel values in the cheek region exhibit less variation and are more concentrated around certain values (Mousa, 2017).



(a) (b)

Figure 2.12. (a) High Entropy value (b) Low Entropy value (Mousa, 2017)

2.12. Evaluation of Steganography Techniques

In the evaluation of steganography methods, the properties of imperceptibility, robustness, and hiding capacity are essential factors to consider (Dougherty, 1994), which described in following:

1. Imperceptibility refers to the main goal of steganography, which is to hide information within a cover object in such a way that it remains undetectable to human observers. When a person views the cover object, they should not be able to distinguish whether it contains embedded information or not. The goal is to make the cover medium appear identical before and after the information is hidden (Dougherty, 1994).
2. Robustness is the degree of resistance or difficulty required to destroy or remove the embedded data without causing noticeable damage to the cover image. A robust steganography method ensures that even if the stego-object (the cover object with hidden information) is subject to various attacks or modifications, the embedded data can still be extracted reliably (Tiwari and Gangurde, 2021).
3. Capacity refers to the maximum amount of information that can be hidden inside the cover image. It is typically measured in bits per pixel (bpp) and represents the tradeoff between the amount of information to be concealed and the quality and fidelity of the cover image. Higher hiding capacity allows for more information to be hidden, but it may also increase the chances of detection or reduce the imperceptibility (Parmar and Pancholi, 2015).

2.13. Quality Evaluation Metrics

Quality Evaluation Metrics in image Steganography assess the effectiveness of hiding data while maintaining the original image's visual integrity. These metrics ensure hidden data remains undetectable and minimize noticeable distortions. By evaluating factors like PSNR, SSIM, MSE, etc., the quality of steganography methods can be measured objectively. The aim is to achieve secure data hiding without compromising the image's appearance (Bayer and Widenfors, 2002). To evaluate the quality of steganography, various metrics are utilized. These metrics include:

1. Mean Squared Error (MSE) quantifies the overall distortion between the pixel values of images. It is calculated by taking the average of the squared differences between corresponding pixel values using the equation (2.8), a lower MSE value indicates better similarity and less distortion between the images, providing a measure of the quality and fidelity of image compression (Bayer and Widenfors, 2002).

$$MSE = \left(\frac{1}{mn}\right) \sum_m \sum_n (A_{mn} - B_{mn})^2 \tag{2.8}$$

2. Peak Signal-to-Noise Ratio (PSNR) quantifies the difference between the original image and the stego image after data embedding, providing a measure of the quality and fidelity of the steganography process. PSNR is typically expressed in decibels (dB) and represents the ratio between the maximum possible power of a signal (in this case, the original image) and the power of the difference signal (the distortion or error introduced during embedding).it is calculated using the equation (2.9), a higher PSNR value indicates better quality and less visible distortion (Al-Saedi, 2016).

$$PSNR = 10 * \log_{10}\left(\frac{MAX^2}{MSE}\right) \tag{2.9}$$

3. UQI assesses the similarity between the images. It considers factors such as contrast, structural similarity, and luminance distortion. UQI ranges from 0 to 1, with 1 indicating perfect image quality. The UQI equation is described in equation 2.10, a higher UQI value signifies better similarity and higher quality between the

images (Majeed et al., 2021).

$$\text{UQI} = \frac{4\sigma_{XY}\bar{x}\bar{y}}{(\sigma_{x^2}+\sigma_{y^2})(\bar{x}^2+\bar{y}^2)} \tag{2.10}$$

4. SSIM is a method used to compute the similarity or structural changes between two images based on luminance, contrast, and structural information. The goal is to measure how close the parameter values are to those of a standard/reference image, indicating a higher degree of similarity (Mishra and Bhanodiya, 2015). The SSIM is calculated in equation 2.14.

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2+\mu_y^2+c_1} \tag{2.11}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2+\sigma_y^2+c_2} \tag{2.12}$$

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y+c_3} \tag{2.13}$$

$$SSIM(x,y) = l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma} \tag{2.14}$$

5. MSSIM is an extension of the SSIM that considers multiple scales of the image. It calculates the SSIM at different scales by downscaling the images and comparing them. By comparing images at different scales, the MSSIM captures more comprehensive information about their structural similarity (Evsutin et al., 2020). The MSSIM is calculated in equation 2.15.

$$MSSIM(X,Y) = \frac{1}{wh}\sum_x \sum_y SSIM(x,y) \tag{2.15}$$

6. SC assesses how well the compressed image preserves the important structural elements and features present in the original image. SC focuses on capturing the spatial relationships and patterns within the image.it calculated in equation (2.16), a less value of SC shows that image is of good quality (Bayer and Widenfors, 2002).

$$SC = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(A_{ij})^2}{\sum_{i=1}^{m}\sum_{j=1}^{n}(B_{ij})^2} \tag{2.16}$$

7. NCC quantifies the similarity between two images based on their cross-correlation. It measures the linear relationship between the pixel values of the original and

compressed images. NCC is calculated by taking the cross-correlation between the two images and normalizing it to a range of -1 to 1, where 1 represents perfect similarity (Bayer and Widenfors, 2002; Yiğit and Karabatak, 2019). It's calculated in equation 2.17.

$$NCC = \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{(A_{ij} * B_{ij})}{A_{ij}^2} \tag{2.17}$$

8. NAE is a metric used to evaluate the similarity or dissimilarity between two images. It measures the average AD between corresponding pixel values in the two images, normalized by the average pixel value of the reference image. It's calculated in equation 2.18, where higher NAE value shows that image is of poor quality (Bayer and Widenfors, 2002; Mishra and Bhanodiya, 2015).

$$NAE = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} (|A_{ij} - B_{ij}|)}{\sum_{i=1}^{m} \sum_{j=1}^{n} (A_{ij})} \tag{2.18}$$

9. AD is a metric commonly used to assess the dissimilarity or error between two images. It measures the absolute value of the pixel-wise difference between corresponding pixels in the two images. It's calculated in equation 2.19, where less AD value indicate that image is of good quality (Matted et al., 2021).

$$Ad(x, y, d) = I_l(x, y) - I_r(x, y, d) \tag{2.19}$$

3. THE PROPOSED APPROACH

3.1. Introduction

This chapter focuses on the approach proposed in this study. It proposes a new approach in image steganography, which involves the embedding of text within a cover image using one of the three different block selection methods: LSB, AMBD, and Entropy method. The selected blocks are used for embedding secret message by LSB method. Only the green channel of each pixel in the selected block is used for data embedding. The parity check is also used to increase the security of the system. Furthermore, the quality of the resulting images, after embedding text using these methods for block selection, are evaluated using various image quality metrics. The chapter is divided into two main parts. Firstly, a simple explanation of the proposed approach is provided through the use of a diagram. Secondly, the embedding process using the three aforementioned method for block selection and using LSB as embedding is demonstrated, followed by a detailed description of the extraction process.

3.2. The proposed approach

This thesis proposes a new approach for embedding a secret message within a cover image by dividing the image into blocks of 16x16 pixels or 32x32 pixels, and then evaluations of the blocks by LSB, AMBD, or Entropy methods. The blocks with the highest values were choosen to embed the secret data by LSB. At this point, it is important to mention that only the green channel values of the pixels in selected blocks will be change depending on the parity. The number of zeros in the binary value of green channel of the related pixel is counted. If the text bit to be embedded is 0 and the count is even, or if the text bit to be embedded is 1 and the count is odd, the pixel is left unchanged. However, if the parity conditions are not met, LSB of the green channel value is changed. The 7 experimental images including satellite images of Iraq used for evaluation of the proposed approach. The implementation of the proposed approach is carried out in the Python language using the PyCharm environment. The implementation is divided into three modules: image preparation, which involves segmenting the image into blocks and converting the secret message into binary bits; embedding, which selects blocks using LSB, AMBD, and Entropy methods to hide the secret message and tests the quality of the stego image compared to the original image using various measures such as PSNR, MSE, UQI, MSSIM, CQM, AD, SC, NCC, and

NAE; and extraction, which extract the secret message from stego image. Figure 3.1 illustrate the methodology of proposed approach.

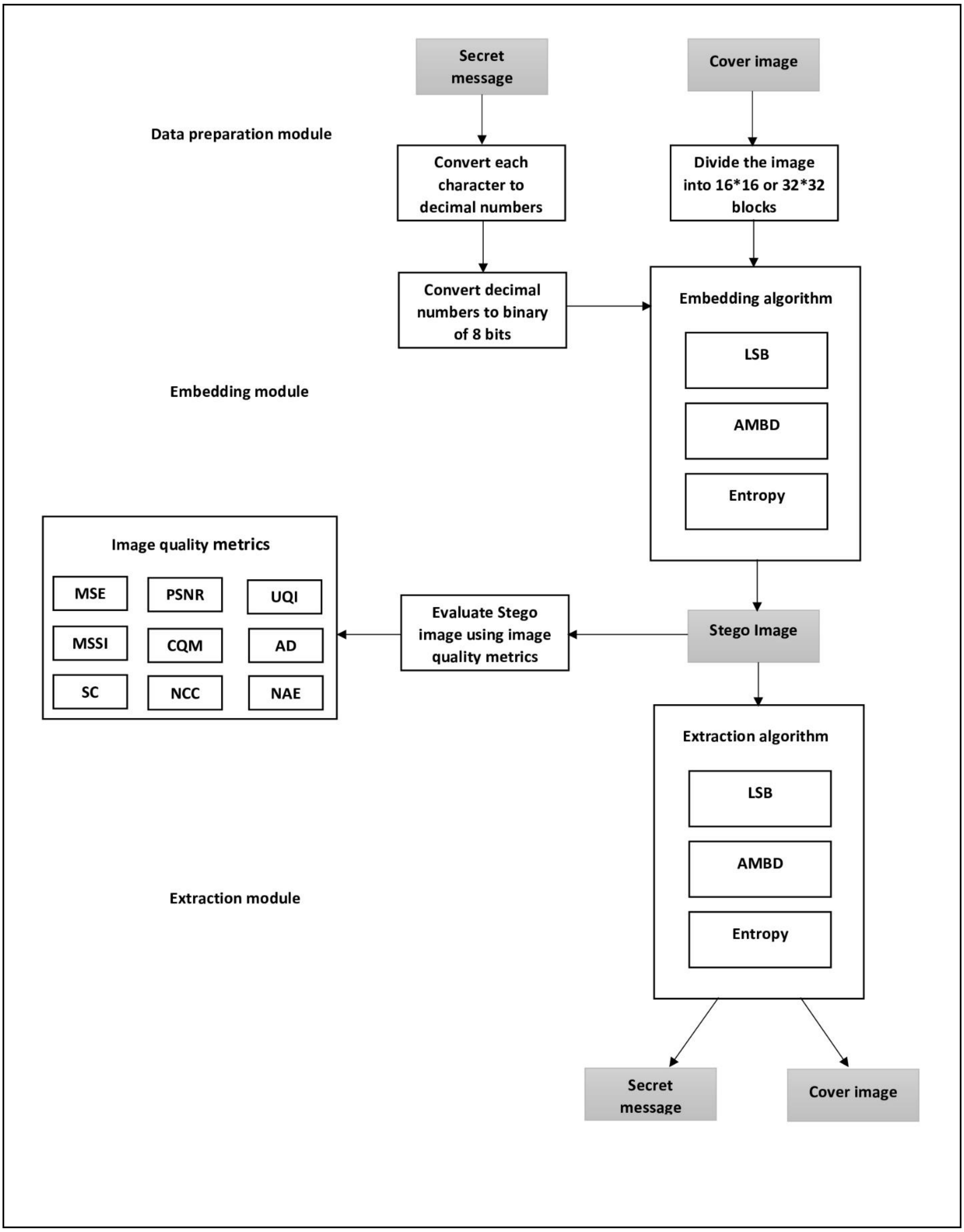
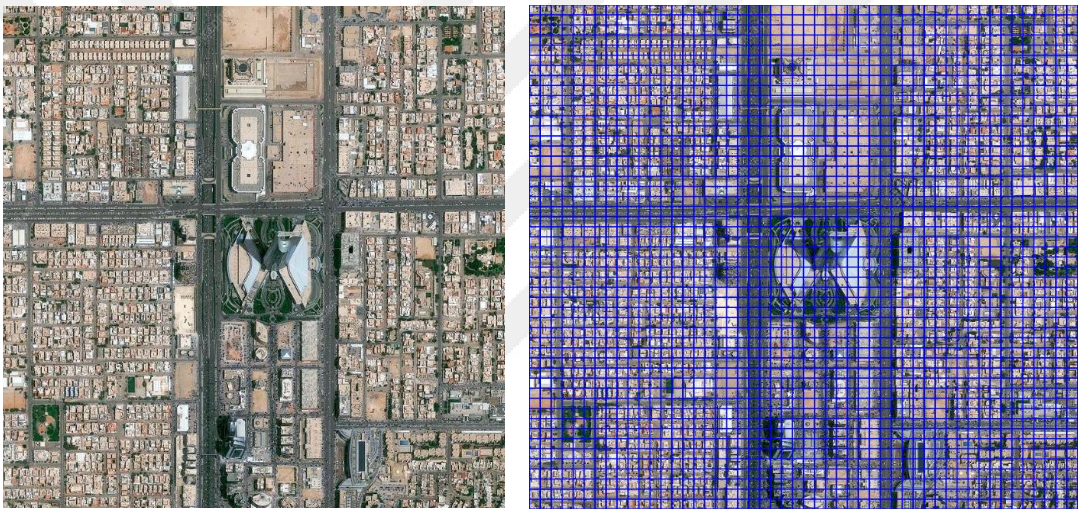


Figure 3.1. Methodology of proposed approach

3.3. Image Preparation

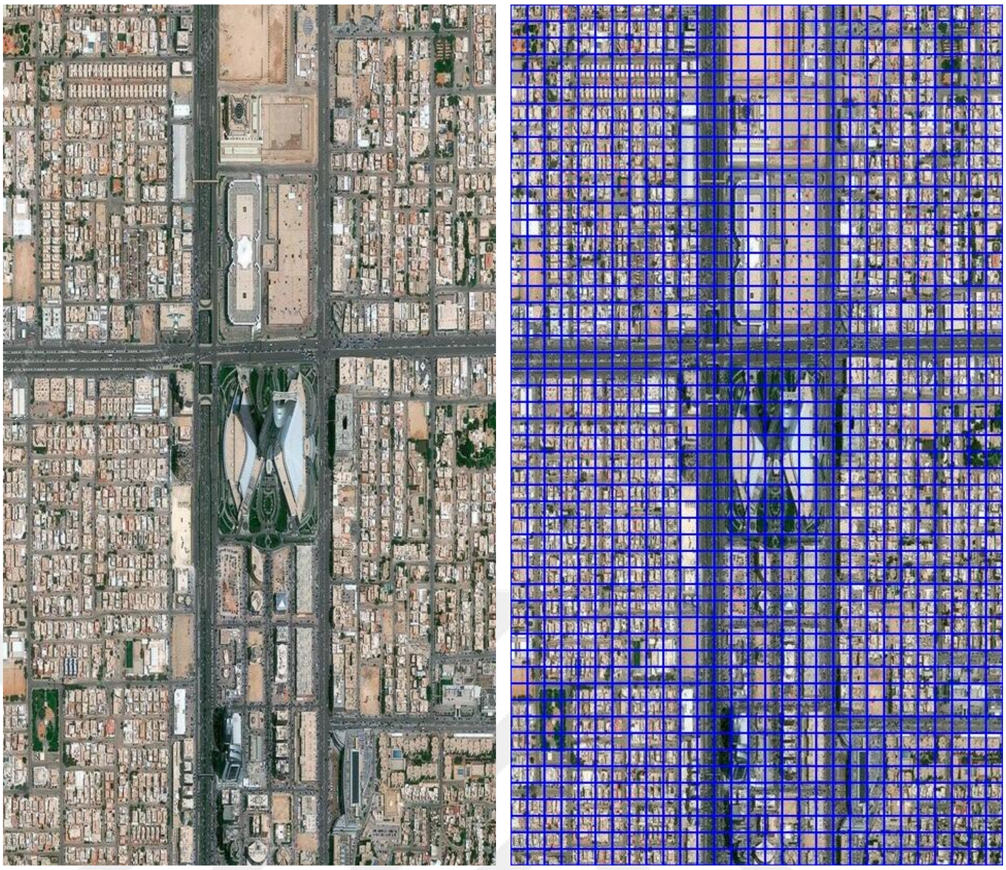
The first step of data preparation involves loading the cover image, which is the image that will be used to hide the data. In this step, the cover image is divided into blocks, typically with sizes of 16x16 pixels and 32x32 pixels block. During the hiding process, the entire image is multiplied in size by the dimensions of these blocks, so if the original image size is, for example, 800x800 pixels, dividing it into blocks with the size of 16x16 pixels will result in 2500 blocks. This calculation is performed by dividing the width of the image (800 pixels) by the size of the block (16 pixels) and multiplying it by the height of the image divided by the height of the block (800/16). Each block contains a fixed number of pixels, which in this case is 1024x1024 pixels, as shown in Figure 3.2.



(a) **(b)**

Figure 3.2. Cover image (a) original state (b) divided by blocks of 16×16 pixels

When the size of a cover image is not evenly divisible by the block size (e.g., block with the size of 16x16 pixels), it poses a problem as there will be areas of the image that cannot be captured by complete blocks. To address this issue, the image will be resize so that its dimensions align with the closest values that are divisible by the block size. By doing so, all blocks in the image will have consistent dimensions, allowing for proper processing without encountering visual artifacts or distortions. Figure 3.3 illustrate the cover image size before and after applying block with the size of 16x16 pixels.



(a) **(b)**

Figure 3.3. Cover image (a) original image size of 478x835 pixels (b) resized image of size 464x832 pixels after applying block of 16x16 pixels size.

A secret message refers to text data that is meant to be concealed within an image. Prior to the embedding process, the text is converted into binary form to facilitate its integration into the image pixels. Initially, the text is stored as an array of characters. Each character is then converted into its corresponding ASCII number representation and stored in an array. Once all the text is converted into ASCII numbers, a zero is added at the end of the decimal array. These zero serves as a stop key that aids in the decoding process when extracting the data from the stego image.

Subsequently, these ASCII numbers are further converted into 8-bit binary form. The resulting binary values are discreetly embedded within LSB of the green channel in an RGB image. By modifying LSB of the green channel, the changes made to the pixel values are typically imperceptible to the human eye, ensuring that the secret message remains hidden discreetly within the image. The imperceptible nature of these alterations allows for the covert hiding of the secret message within the image, without arousing suspicion. Figure 3.4 illustrate an example of the convert process.

3.4. Embedding Module

After dividing the cover image into blocks, the method aims to find suitable blocks for hiding a secret message without significantly altering the appearance of the image. To ensure a seamless integration, it employs three effective methods: LSB, AMDB, and Entropy. LSB method evaluates LSB of each pixel in the given block. AMDB method evaluates the average brightness levels within blocks to ensure they offer enough variability for embedding the message without causing noticeable visual distortions. Furthermore, Entropy method measures the complexity and randomness of pixel values within blocks, prioritizing blocks with higher Entropy for better message concealment. By leveraging these methods, the approach identifies appropriate pixels for embedding the secret message, while preserving the cover image's overall appearance and its color histogram shape.

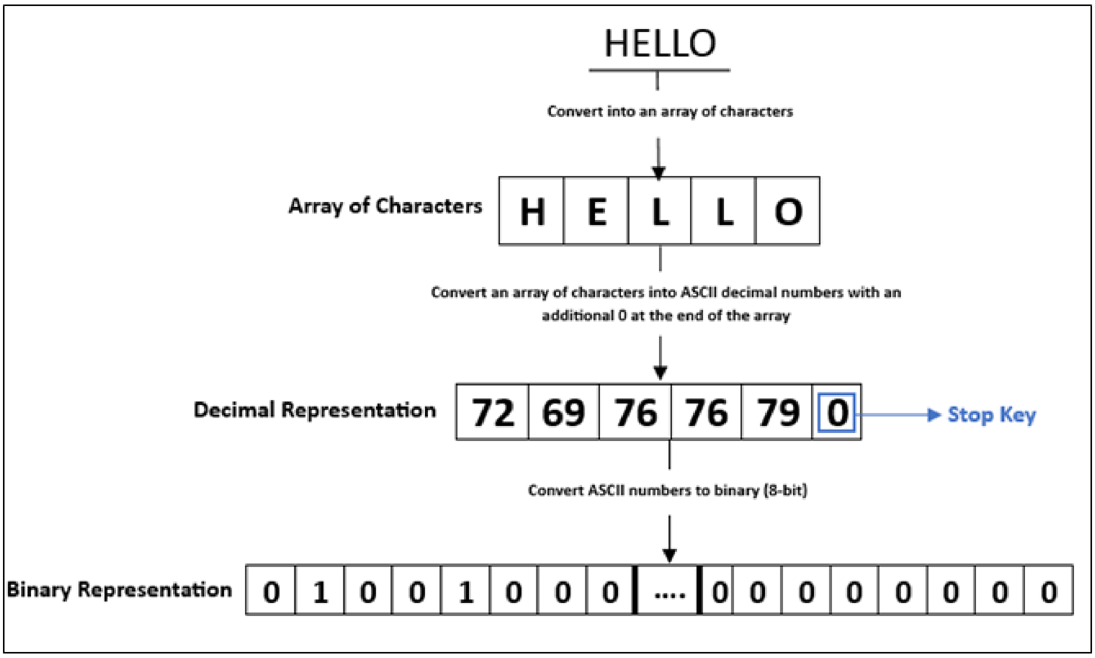


Figure 3.4. Example of the converting process

3.4.1. LSB based block selection

This method involves several steps to calculate LSB value for each block. Firstly, the block is converted to grayscale, simplifying subsequent analysis. Then, the approach proceeds to extract LSB from each pixel within the grayscale block. This is achieved by performing a bitwise AND operation with the value 1, effectively isolating

LSB of each pixel. Figure 3.5 illustrate an example of calculating LSB value for block A with size of 4x4 pixels.

Once LSB values are extracted for all pixels within the block, the algorithm shifts its focus to identifying the blocks that possess the highest LSB values. It accomplishes this by sorting LSB values in descending order. Through this sorting process, the algorithm can determine which blocks exhibit the most potential for effectively hiding the secret message. The top blocks with the highest LSB values are then selected for further processing. Figure 3.6 illustrates a selected 16x16 pixels block with the highest LSB values within an image of 800x800 pixels, which is the best location to hide data.

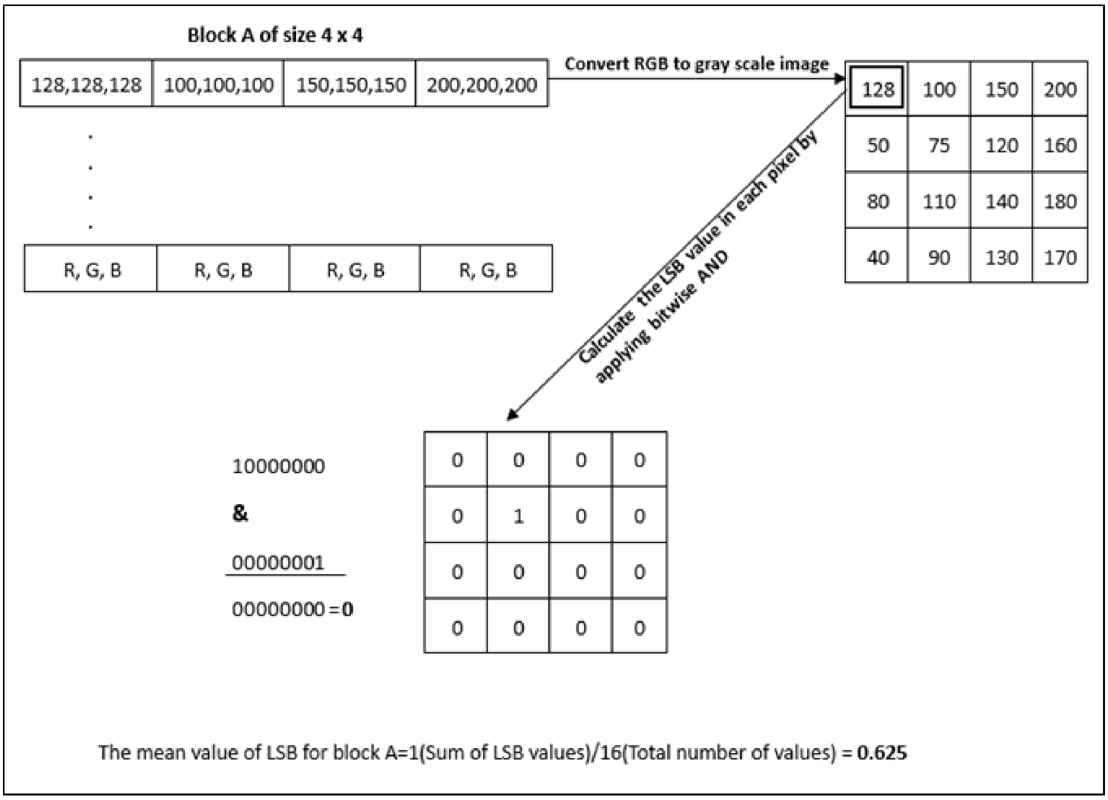


Figure 3.5. Calculating LSB value for block of 4x4 pixels

Figure 3.6. The blocks with highest LSB values

3.4.2. AMBD based block selection

The AMBD method is the second method used to embed text within an image by analyzing the brightness levels of the different blocks within the image. To apply AMBD method in each block, the method calculates the average brightness value by determining the mean of the brightness values of all the pixels within that specific block. The brightness value is represented as a single value for each color channel (red, green, and blue) in the RGB color model. This average brightness value gives an indication of the overall brightness of the block. After averaging the brightness values of the individual block, AMBD value is determined. The AMBD value represents the absolute average of all mean brightness values obtained from the single block. Figure 3.7 illustrate an example of calculating AMBD value for block A with size of 4 x 4 pixels. Once AMBD values are extracted for each block, the algorithm shifts its focus to identifying the blocks that possess the highest AMBD values. It accomplishes this by sorting AMBD values in descending order. Through this sorting process, the algorithm can determine which blocks exhibit the most potential for effectively hiding the secret message. The top blocks with the highest AMBD values are then selected for further processing. Figure 3.8 illustrates a selected 16x16 pixels size block with the highest AMBD values within an image of 800x800 pixels, which is the best location to hide data.

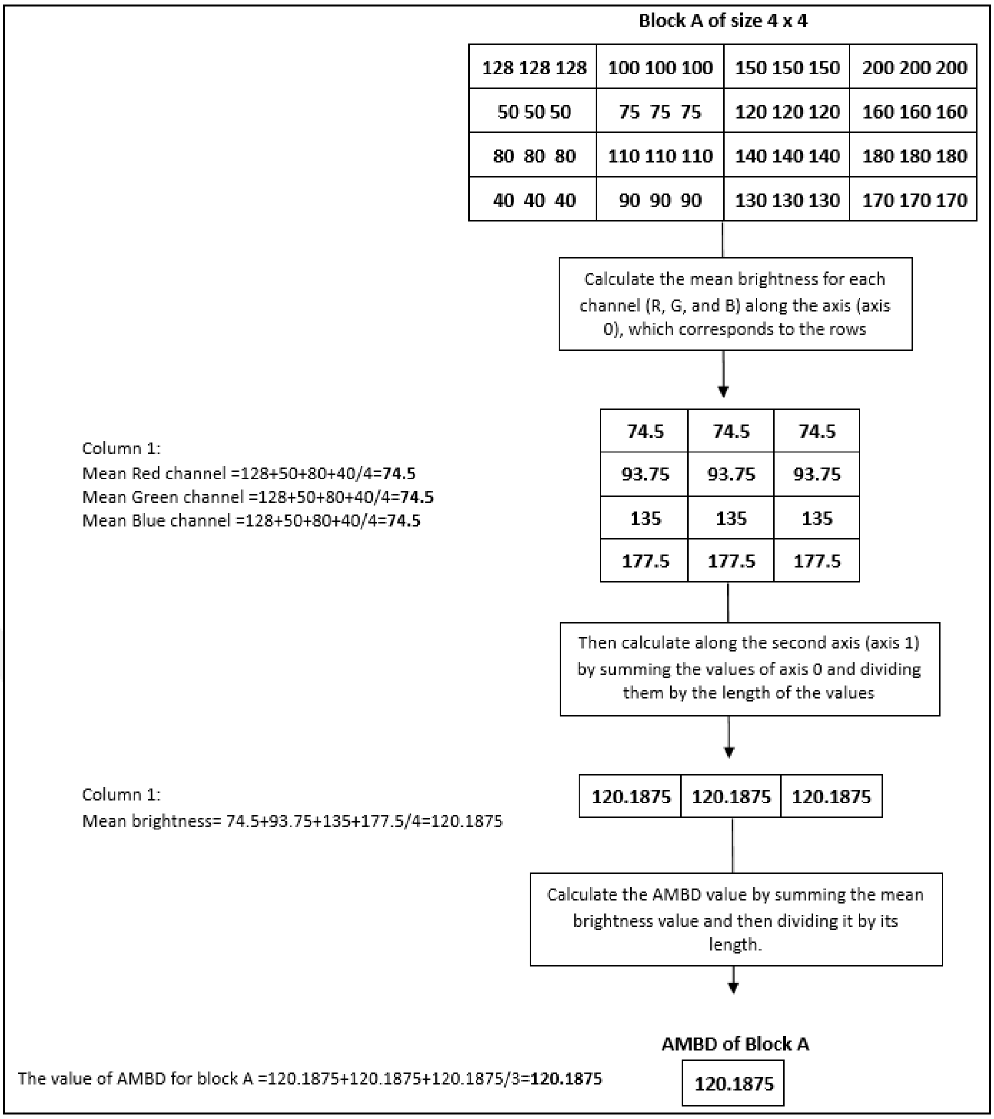


Figure 3.7. Calculating AMBD values for block of 4x4 pixels

Figure 3.8. The block with highest AMBD values

3.4.3. Entropy based block selection

Entropy-based method is another approach used to embed text within an image by analyzing the complexity and information content of different blocks. Entropy represents the degree of randomness or uncertainty in a set of data. In the context of image processing, Entropy provides a measure of the diversity and intricacy present in an image block. To calculate Entropy, a histogram of pixel values is created for each block. This histogram tallies the occurrences of each pixel value within the block. From the histogram, the probabilities of each pixel value are computed by normalizing the histogram values. Next, Entropy value is obtained by summing the product of each probability and its logarithm. To ensure a positive value, the result is negated. This calculation enables an estimation of the level of information or complexity present in the block. Figure 3.9 illustrate an example of calculating Entropy value for block A with size of 4 x 4 pixels.

Once Entropy values are extracted for each block, the algorithm shifts its focus to identifying the blocks that possess the highest Entropy values. It accomplishes this by sorting Entropy values in descending order. Through this sorting process, the algorithm can determine which blocks exhibit the most potential for effectively hiding the secret message. The top blocks with the highest Entropy values are then selected for further processing. Figure 3.10 illustrates a selected 16x16 pixels block with the highest Entropy values within an image of 800x800 pixels, which is the best location to hide data.

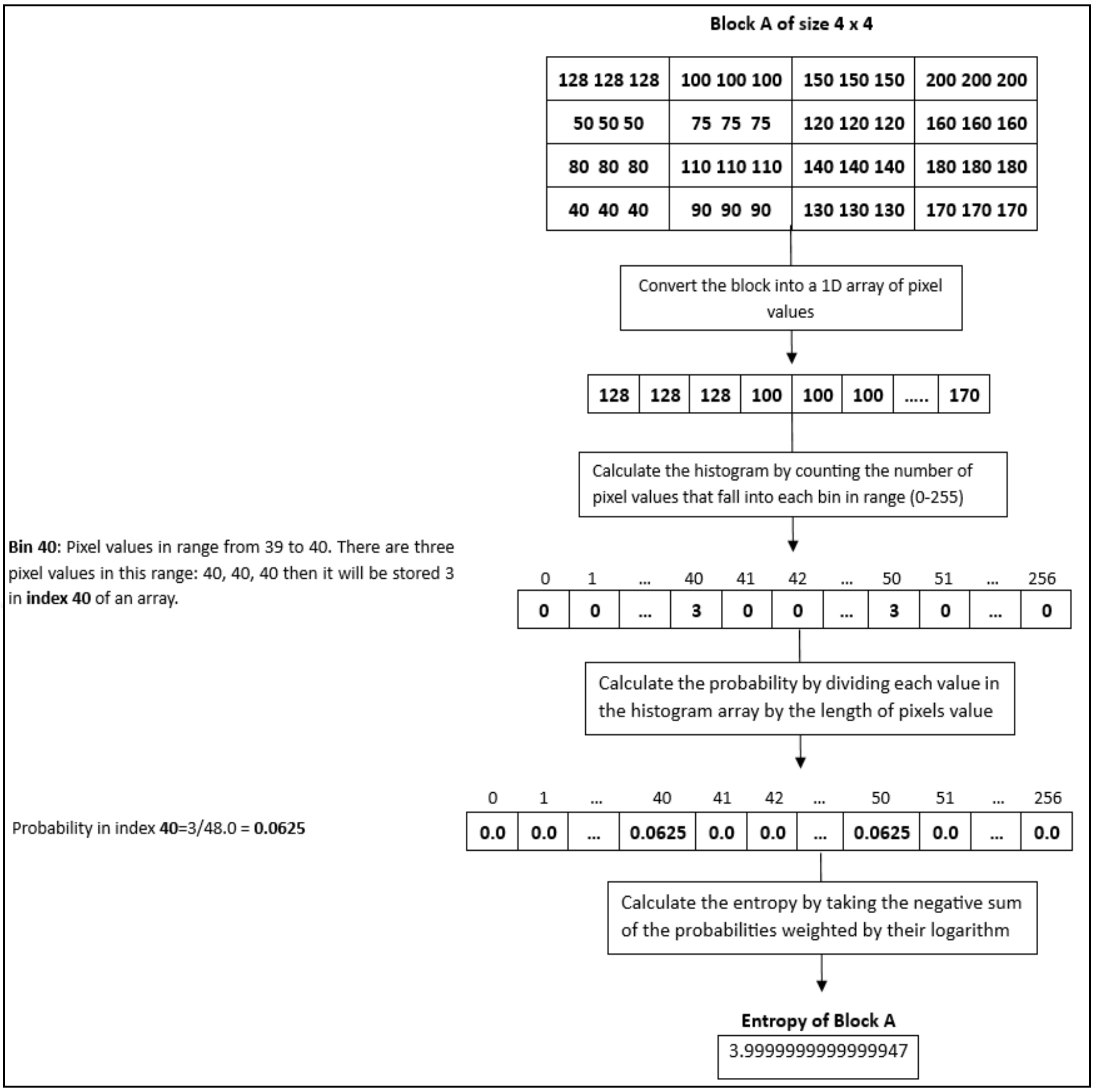


Figure 3.9. Calculating Entropy value for block of 4x4 pixels

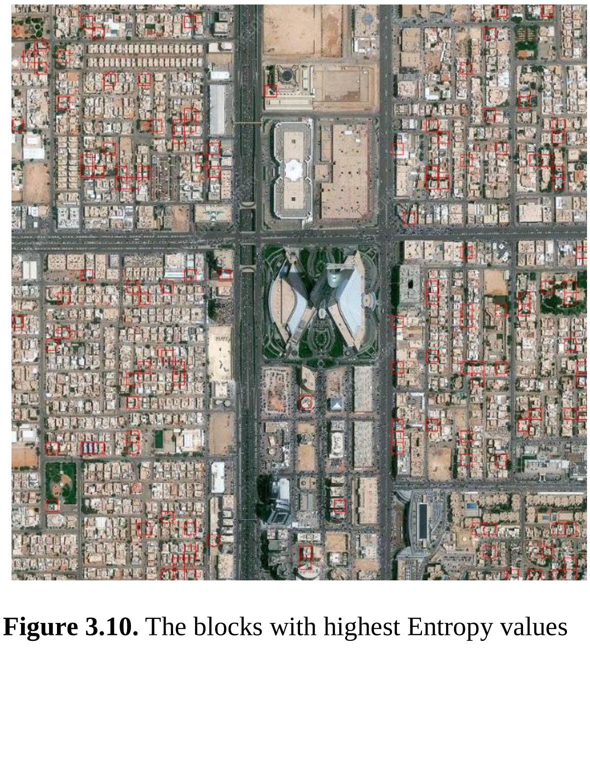


Figure 3.10. The blocks with highest Entropy values

3.4.4. The process of embedding text within a cover image

The process of embedding text within blocks that were selected based on LSB, AMBD, or Entropy methods involves modifying the **green channel values** of the selected blocks in the image. It starts by iterating over each pixel within the chosen blocks. For each pixel, the green channel value is retrieved and converted to a binary representation. The number of zeros in the binary value is then counted. Next, it checks the **parity of the zero count** to determine if the current text bit should be encoded. If the text bit is 0 and the count is even, or if the text bit is 1 and the count is odd, the pixel is left unchanged. However, if the parity conditions are not met, LSB of the green channel value is changed. The modified green channel value is converted back to decimal and updated in the pixel. This process continues until all text bits have been encoded or the selected blocks have been processed. Figure 3.11 illustrates an example of the embedded text ‘h’ with a block.

After the embedding process, the image will be reconstructed using these modified blocks, resulting in a stego image that contains the hidden secret message. This stego image can then be tested using quality image metrics to compare it with the original image. Figure 3.12 illustrate the flow chart of embedding process.

Text bit: **‘h’** as binary is **‘01101000’**

Block A of size 4 x 4

128 **128** 128	100 **100** 100	150 **150** 150	200 **200** 200
50 **50** 50	75 **75** 75	120 **120** 120	160 **160** 160
80 **80** 80	110 **110** 110	140 **140** 140	180 **180** 180
40 **40** 40	90 **90** 90	130 **130** 130	170 **170** 170

Block A with embedded text

128 **129** 128	100 **100** 100	150 **151** 150	200 **201** 200
50 **50** 50	75 **75** 75	120 **120** 120	160 **160** 160
80 **80** 80	110 **111** 110	140 **141** 140	180 **180** 180
40 **40** 40	90 **90** 90	130 **130** 130	170 **170** 170

First pixel within the block:

The pixel data is [128, 128, 128].
The green value data only: **128**
Green pixel as binary: 1000000**0**
Text bit: **0** (first bit)

- The number of zeros in the green channel is 7, which is odd, and the text bit is 0, then the least significant bit of the green channel value is flipped to 1.

New green pixel as binary: 1000000**1**.
The new green value as a decimal: **129**
The new pixel data: [128, 129, 128]

Same process for the rest pixels until all text bits embeds within the block.

Figure 3.11. Example of embedding text to a block of size of 4x4 pixels

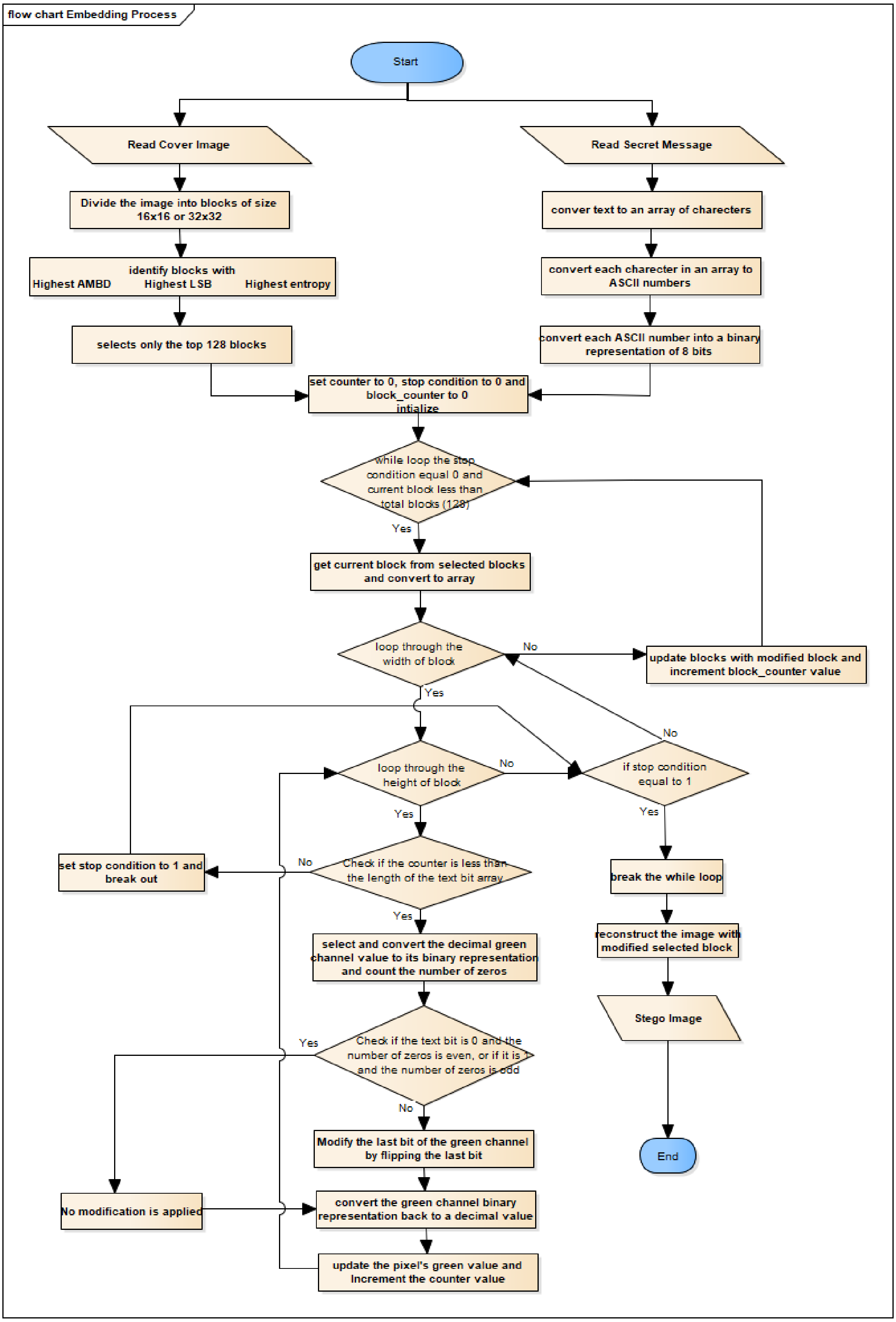


Figure 3.12. Flowchart of the embedding process

3.5. Extraction Process

The process of extracting text from blocks selected based on LSB, AMBD, or Entropy methods involves modifying the green channel values of the selected blocks in the image. The algorithm iterates over each pixel within the chosen blocks and extracts the green value, which contains the hidden text. This green value is then converted into its binary representation. The algorithm counts the number of zeros in the binary value and checks LSB. If the count of zeros is even, it appends 0; if it is odd, it appends 1 and stores it inside a binary text list. This process continues for all the pixels in the block until a complete character, consisting of 8 bits, is obtained and stored within the binary text list. Once a complete character is obtained, it is converted back to decimal form, representing an ASCII value, which is then appended to a text integer list. If the extracted ASCII value is 0, signifying the end of the hidden text, the extraction process stops. Otherwise, the process continues to extract more characters from the remaining blocks. Figure 3.13 illustrate an example of extracting text "h" within a block of 4x4 pixel size.

After the extraction process, the decimal ASCII values stored in the text integer list will be converted into a string format. This conversion allows the hidden texts to be represented as readable text. The algorithm then proceeds to print the extracted texts, revealing the contents that were previously hidden within the image. Figure 3.14 illustrate the flow chart of the extraction process.

Block A with embedded text

128 **129** 128	100 **100** 100	150 **151** 150	200 **201** 200
50 **50** 50	75 **75** 75	120 **120** 120	160 **160** 160
80 **80** 80	110 **111** 110	140 **141** 140	180 **180** 180
40 **40** 40	90 **90** 90	130 **130** 130	170 **170** 170

First pixel within block:

The pixel data is [128, 129, 128]
The green value data only: **128**
Green pixel as binary: 10000001

- The number of zeros in the green binary representation is **6**, which is even, so append **0** to a binary text list

.
.
.

Third pixel within block:

The pixel data is [150, 151, 150]
The green value data only: **151**
Green pixel as binary: 10010111

- The number of zeros in the green binary representation is 3, which is odd, so append 1 to a binary text list

.
.
.

- The process continues until it obtains a binary text list with 8 bits, which is [0,1,1,0,1,0,0,0].
- Converted into a decimal value, which is 104, then as a string, which then represents '**h**'.

Figure 3.13. Example of extracting text within a block 4x4 pixels

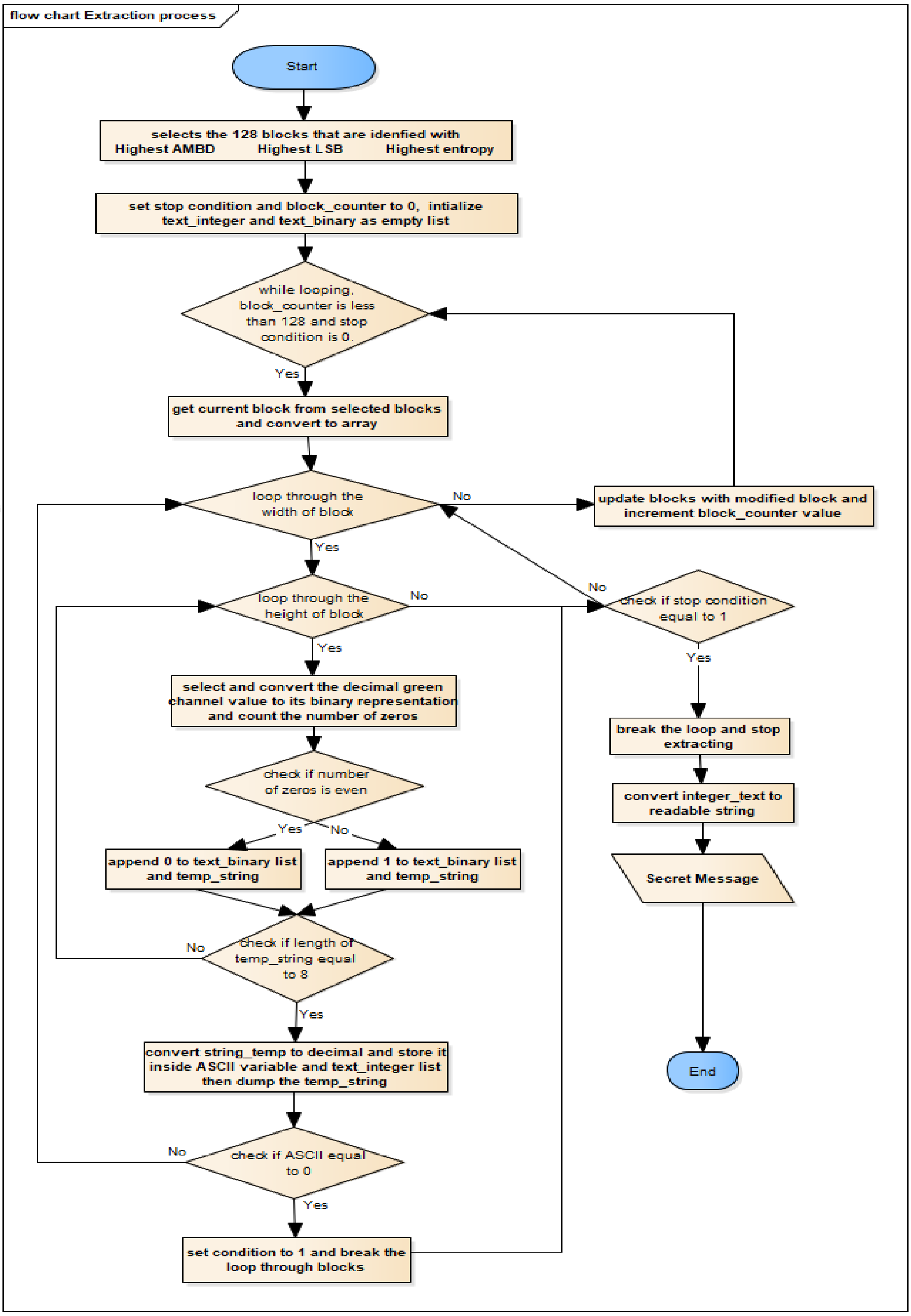


Figure 3.14. Flow chart of the extraction process

4. PROPOSED SYSTEM IMPLEMENTATION, RESULTS AND DISCUSSION

4.1. introduction

In this chapter, we describe the experiments conducted to evaluate the effectiveness of the LSB, AMBD, and Entropy algorithms in the context of steganography. We also introduce the measures that were taken into consideration to assess the efficiency and effectiveness of our steganographic system. The evaluation aims to provide a comprehensive understanding of the algorithms' performance by considering various aspects relevant to steganography. By analyzing the results, we aim to determine the overall quality and robustness of the proposed techniques, ensuring a thorough assessment of their capabilities and potential for practical steganographic applications.

4.2. Execution Environment

The requirements (hardware and software) that have been used are as follows:

1. Hardware:
   - Laptop
   - Processor type Intel Core i5-8265U Computer Processing Unit (CPU), speed 1.80 GHz
   - GPU: GeForce MX130
   - Memory (RAM) capacity of 8 GB
2. Software Environment:
   - Windows 11
   - Python Language.
   - PyCharm program.

In this study, we implemented three techniques: the proposed method, LSB (Least Significant Bit), AMBD (Average Multi-Block Difference), and Entropy technique, all using the Python programming language, specifically version 3.10, within the PyCharm IDE. The validation of algorithm efficiency was performed on 24-bit colored JPG images.

4.3. Image Steganography System Implementation

In this section, we will provide a detailed overview of the implementation and thorough experimentation of an image steganography system that includes three approaches: AMBD, LSB, and Entropy-based approaches. The procedural steps for implementation the image steganography approaches are as follows:

Step one: When the code is run following window appears and the process starts by selecting one of the image saved in jpeg or jpg format, which is required to hide the text within it. Figure 4.1 illustrates the selection of a Lena image with dimensions of 512 x 512 for the implementation of the proposed image steganography system.

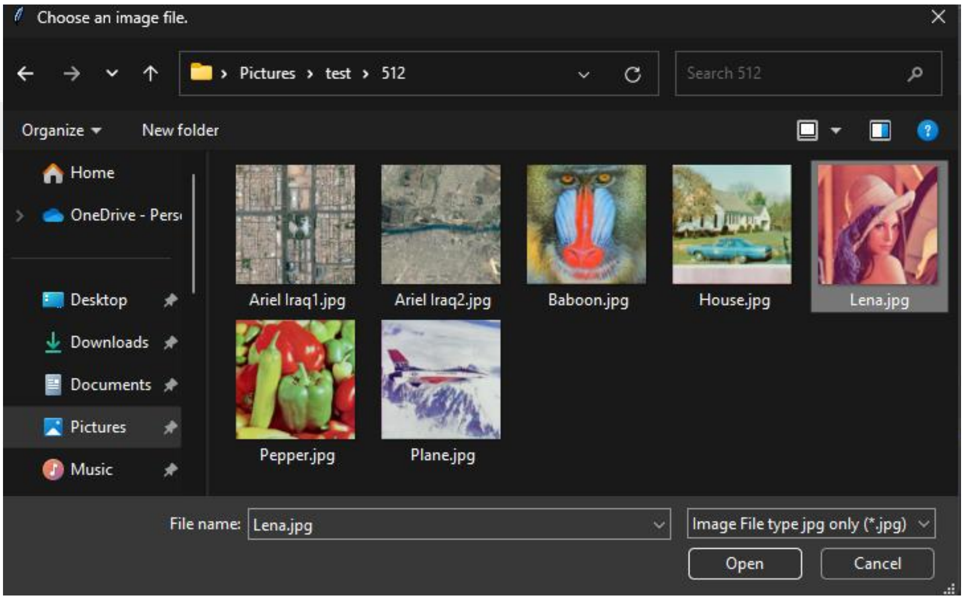


Figure 4.1. Selecting an image in .jpg format

Step two: After selecting the image, it will be divided into blocks of 16x16 pixels and 32x32 pixels. Figure 4.2 shows the division of the Lena image into blocks of size 16 × 16.

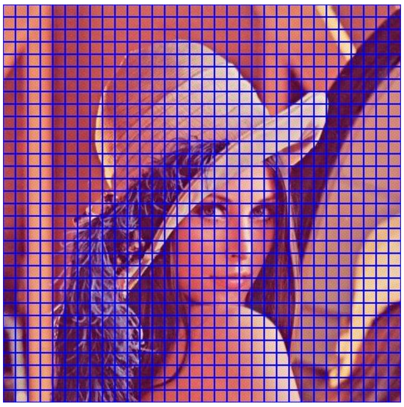
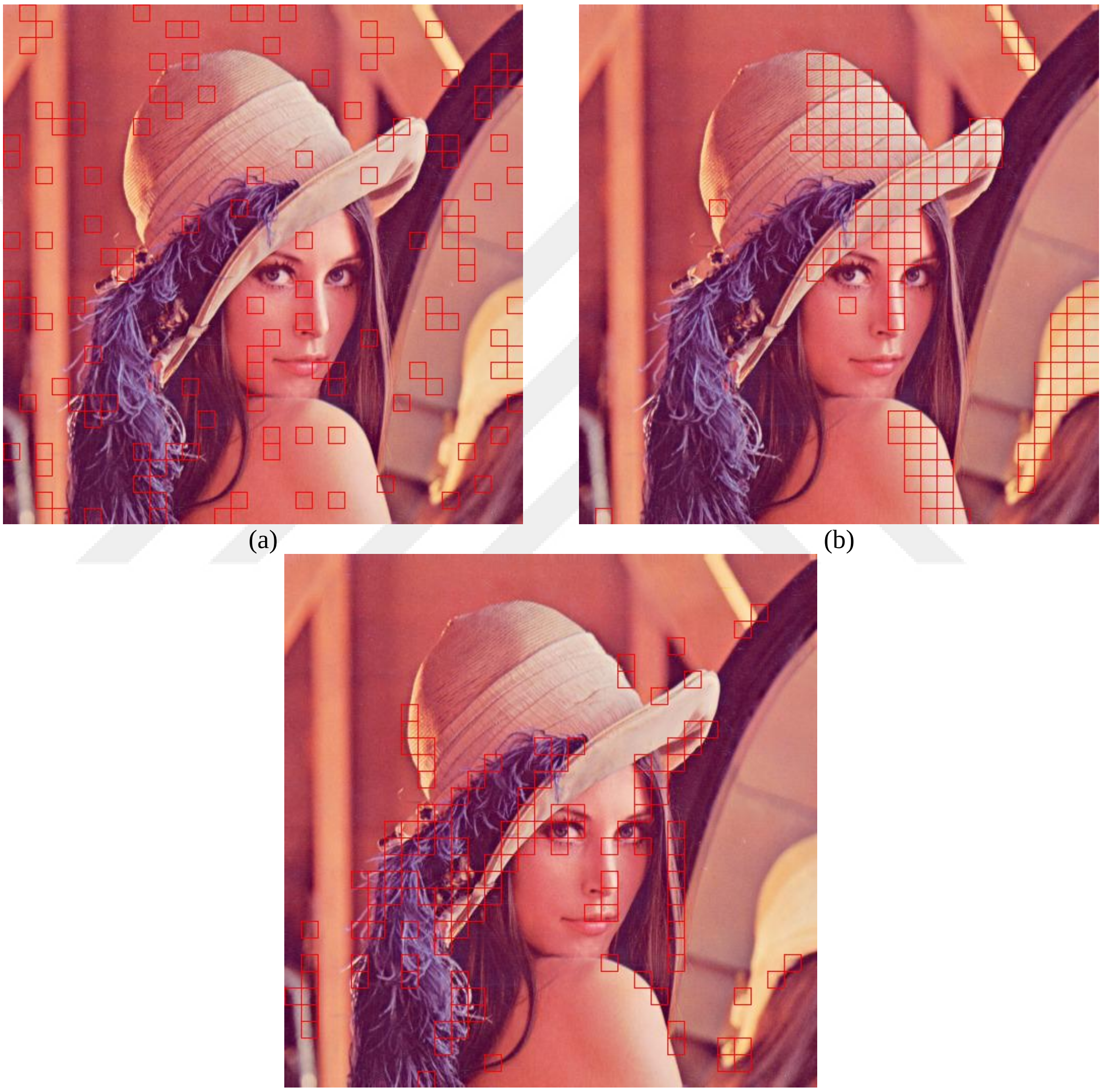


Figure 4.2. Image with 16×16 block size

Step three: In this step, 128 blocks are chosen according to the specific approach used for hiding text within these selected blocks. In the LSB approach, the selection is based on the top blocks with the highest LSB values. For the AMBD approach, the blocks with the highest AMBD values are chosen. Lastly, in the entropy approach, blocks with the highest entropy values are selected. Figure 4.3. illustrates the blocks that have been selected based on the LSB, AMBD, and Entropy algorithms by developed system.



(a) (b)

(c)

Figure 4.3. Selected blocks with highest (a) LSB (b) AMBD (c) Entropy

Step four: After all the blocks were choosen for three approaches, the user chooses the text file to be hidden inside the image, as shown in Figure 4.4.

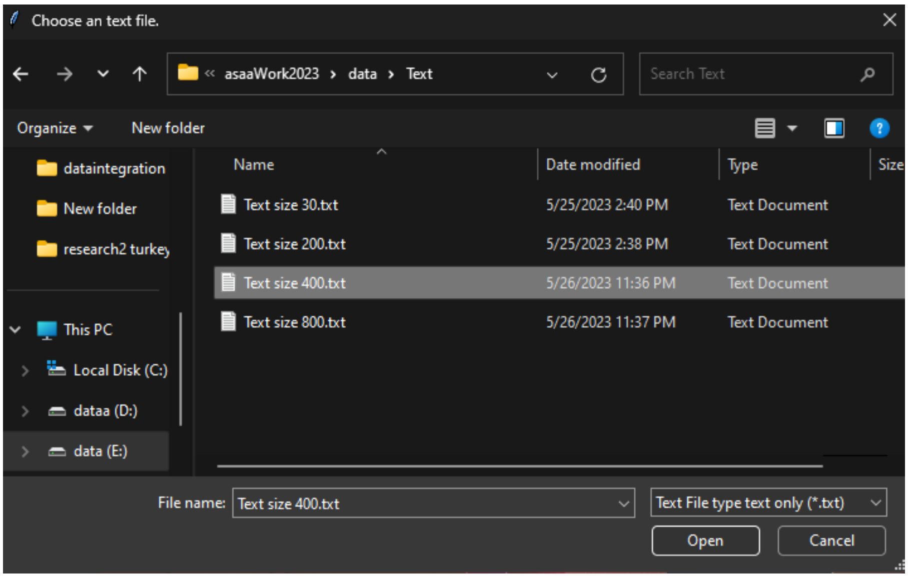
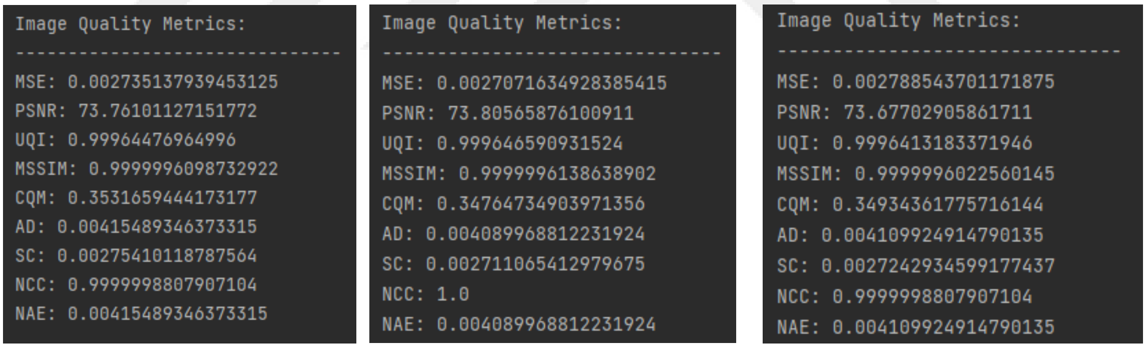


Figure 4.4. Selecting a text file in .txt format

Step five: The system embeds text within the selected blocks, subsequently presenting the stego image alongside an assessment of its image quality. Figure 4.5 illustrates the assessment of image quality for the stego image resulting from the implementation of each of the three approaches.



(a) (b) (c)

Figure 4.5. Image quality assessment for stego image obtained using (a) LSB approach (b) AMBD approach (c) Entropy algorithm

Step six: Finally, the system executes the text extraction process from the stego image, revealing the concealed text within. Figure 4.6 demonstrates the extracted text from the stego image.

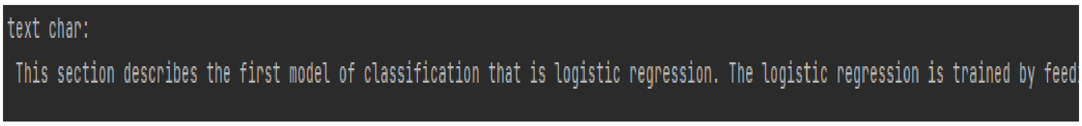


Figure 4.6. Extracted text from the stego image

4.4. Experimental Result

This study's experiments included a detailed comparison of the original image file (cover image) and the stego image created using three suggested techniques: LSB, AMBD, and Entropy. The assessment encompassed the analysis of hidden text both before transmission and after reception to measure the level of distortion introduced during the embedding of secret text into the original cover file. Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), UQI, Multi-Scale Structural Similarity (MSSIM), Pearson Correlation Coefficient-based Quality Metric (CQM), Average Difference (AD), SC, NCC, and NAE were calculated to assess the effectiveness of the proposed approaches (NAE). The quality and similarity of the original and stego images were revealed by these parameters, which was extremely helpful.

The experiments utilized colored images in .jpeg or .jpg format, varying in resolution with sizes of 256x256, 512x512, and 1024x1024 pixels. Different text sizes of 30, 200, 400, and 800 bytes were embedded into the images during the experimentation. Prior to applying the proposed techniques (LSB, AMBD, and Entropy), the images were divided into blocks of either 16x16 or 32x32 pixels. Figure 4.7 visually represents the images that were used for the comprehensive experimental analysis, providing a clear overview of the images involved in the study.

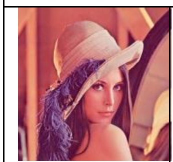
Color image of size dimensions (256*256, 512*512, 1024*1024) and file format(.jpeg)						
						
Lana	Jet	Pepper	Baboon	House	Aerial Iraq1	Aerial Iraq2

Figure 4.7. Cover file specifications used in the experiments

4.4.1. Experimental results of the proposed LSB approach

In this experimental study, the Least Significant Bit (LSB) technique was employed for embedding text in images. Initially, the images were partitioned into blocks of either 16x16 or 32x32 pixels. For each block, the LSB values were computed, and those with the highest LSB values were identified. A total of 128 blocks with the highest LSB values were selected for embedding a secret message, with the message sizes ranging from 30 to 800 bytes. To assess the effectiveness of the LSB technique, a comparison was made between the resulting stego image and the original image. Both images were converted to grayscale, allowing for an analysis of their pixel intensities. Histograms of the pixel values were plotted, providing a comprehensive understanding of the distribution and intensity of pixels in both images. This analysis offered valuable insights into the impact of the data embedding process on the final stego image, shedding light on potential changes in pixel values due to the secret message insertion. Such analysis is crucial for evaluating the efficacy and robustness of the LSB technique in maintaining the visual quality of the stego image while concealing the secret message.

In Figure 4.8, the Lina image is processed using the LSB technique with a block size of 16x16. A visual comparison between the original and processed images shows a remarkable similarity, indicating the effective preservation of essential image properties. Although a detailed histogram analysis reveals minor distortions with a few frequency increases in pixel density and slight variations in pixel intensity values.

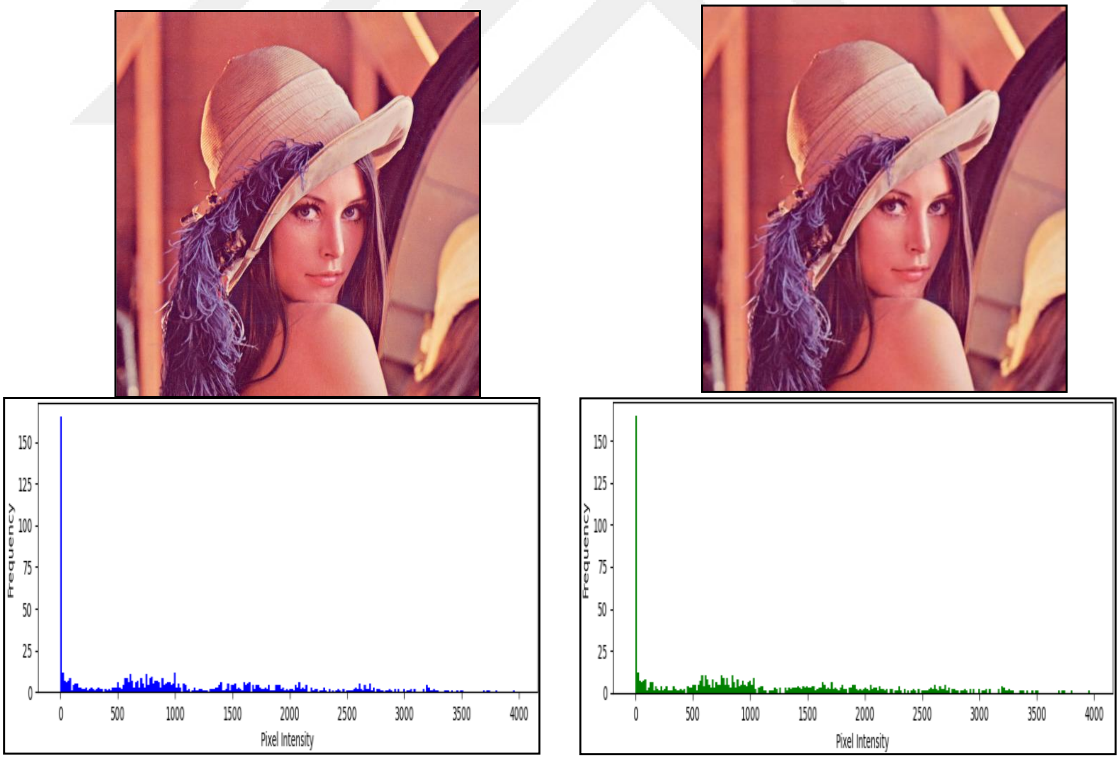


Figure 4.8. Histogram graph of the original and stego of the image named Lena using the LSB approach with block size of 16x16

In Figure 4.9, displays the results of using the LSB approach with a 32x32 block size on the Lina image for the data embedding procedure. The histogram distribution of pixel density in the image processed with a block size of 32x32 mimics that of the image processed with a block size of 16x16 quite closely. While the main patterns of the two distributions are almost identical, there are some slight differences.

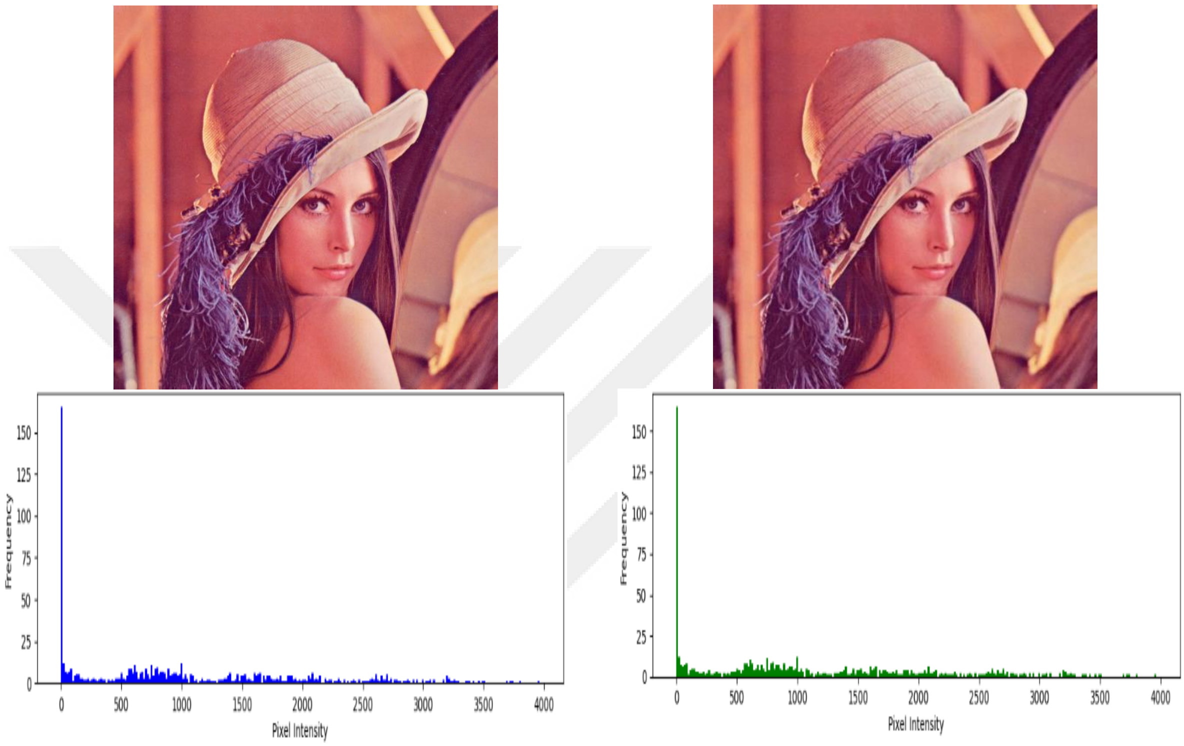


Figure 4.9. Histogram graph of the original and stego of the image named Lena using the LSB approach with block size of 32x32

Table 4.1 presents the results of using the LSB method with a block size of 16x16 on color images of different resolutions. Increasing the text size used for data hiding consistently decreased the PSNR values, indicating a trade-off between data capacity and image quality. As more data was embedded into the images, the least significant bits underwent more modifications, leading to increased distortion and lower PSNR values. The PSNR values for color images at resolutions 256, 512, and 1024 ranged from 64.92 to 78.86, 70.90 to 85.07, and 76.99 to 91.34, respectively. This reveals that higher-resolution images generally achieved higher PSNR values, but the overall trend of decreasing PSNR with increasing text size remained consistent across all tested images.

In contrast, the MSE values for all the examined color images increased as the text size increased. In general, a smaller MSE value suggests that the image steganography approach is of greater quality. The smallest MSE values were found for 30-byte text sizes. The MSE values for color images with resolutions of 256, 512, and 1024 varied between 0.0008 and 0.0204, 0.0002 and 0.005, and 0.00004 and 0.0012, respectively. Additionally, it can be determined that the MSE is connected with the number of color variations in the cover image, with greater color variations resulting in lower MSE values.

The UQI and MSSIM metrics both took a value of one, indicating a perfect similarity between the original and stego images. The CQM metric demonstrated that lower values for text size 30 indicated reduced distortion and better quality in the stego images. Furthermore, the AD, SC, and NAE metrics exhibited low values, suggesting that the stego images closely resembled the original images with well-preserved structural similarity and content.

Table 4.1. Results of statistical parameters obtained from applying the LSB approach on color images with different text sizes of block size of 16x16

Image name	Text Size (byte)	PSNR (dB)	MSE	UQI	MSSIM	CQM	AD	SC	NCC	NAE
Image resolution 256*256										
Lena	30	78.71	0.0008	1	1	0.110	0.001	0	1	0.001
	200	71.10	0.005	1	1	0.635	0.007	0	1	0.007
	400	67.76	0.010	1	1	1.386	0.016	0	1	0.016
	800	**64.92**	**0.020**	**1**	**1**	**2.674**	**0.031**	**0**	**1**	**0.031**
Baboon	30	**78.86**	**0.0008**	**1**	**1**	**0.096**	**0.001**	**0**	**1**	**0.001**
	200	70.86	0.005	1	1	0.673	0.079	0	1	0.079
	400	67.85	0.010	1	1	1.408	0.016	0	1	0.016
	800	65.01	0.020	1	1	2.679	0.031	0	1	0.031
Pepper	30	78.63	0.0008	1	1	0.090	0.001	0	1	0.001
	200	70.83	0.005	1	1	0.664	0.007	0	1	0.007
	400	67.74	0.010	1	1	1.465	0.017	0	1	0.017
	800	65.00	0.020	1	1	2.631	0.0309	0	1	0.030
House	30	78.71	0.0008	1	1	0.104	0.001	0	1	0.001
	200	70.80	0.005	1	1	0.670	0.007	0	1	0.007
	400	67.75	0.010	1	1	1.401	0.016	0	1	0.016
	800	64.94	0.020	1	1	2.612	0.030	0	1	0.030
Plane	30	78.81	0.0008	1	1	0.110	0.001	0	1	0.001
	200	70.79	0.005	1	1	0.688	0.008	0	1	0.008
	400	67.71	0.011	1	1	1.426	0.016	0	1	0.016
	800	64.94	0.020	1	1	2.680	0.031	0	1	0.031
Ariel Iraq1	30	78.56	0.0009	1	1	0.118	0.001	0	1	0.001
	200	70.81	0.005	1	1	0.683	0.008	0	1	0.008
	400	67.69	0.011	1	1	1.389	0.016	0	1	0.016
	800	64.96	0.020	1	1	2.656	0.031	0	1	0.031
Ariel Iraq2	30	78.78	0.0008	1	1	0.109	0.001	0	1	0.001
	200	70.80	0.005	1	1	0.675	0.007	0	1	0.007

	400	67.64	0.0115.	1	1	1.443	0.016	0	1	0.016
	800	64.94	0.020	1	1	2.688	0.031	0	1	0.031
Image resolution 512*512										
Lena	30	84.70	0.0002	1	1	0.027	0.0003	0	1	0.0003
	200	76.89	0.001	1	1	0.175	0.002	0	1	0.002
	400	73.76	0.0027	1	1	0.353	0.004	0	1	0.004
	800	71.02	0.005	1	1	0.672	0.007	0	1	0.007
Baboon	30	84.41	0.0002	1	1	0.026	0.0003	0	1	0.0003
	200	77.00	0.001	1	1	0.163	0.001	0	1	0.001
	400	73.73	0.002	1	1	0.349	0.004	0	1	0.004
	800	**70.90**	**0.005**	**1**	**1**	**0.673**	**0.007**	**0**	**1**	**0.007**
Pepper	30	84.55	0.0002	1	1	0.031	0.0003	0	1	0.0003
	200	76.67	0.001	1	1	0.189	0.002	0	1	0.002
	400	73.67	0.002	1	1	0.371	0.004	0	1	0.004
	800	70.98	0.005	1	1	0.663	0.007	0	1	0.007
House	30	84.32	0.0002	1	1	0.032	0.0003	0	1	0.0003
	200	76.72	0.001	1	1	0.175	0.002	0	1	0.002
	400	73.56	0.002	1	1	0.359	0.004	0	1	0.004
	800	71.05	0.005	1	1	0.641	0.007	0	1	0.007
Plane	30	84.63	0.0002	1	1	0.027	0.0003	0	1	0.0003
	200	76.72	0.001	1	1	0.177	0.002	0	1	0.002
	400	73.66	0.0027	1	1	0.354	0.004	0	1	0.004
	800	71.00	0.005	1	1	0.658	0.007	0	1	0.007
Ariel Iraq1	30	**85.07**	**0.0002**	**1**	**1**	**0.023**	**0.0002**	**0**	**1**	**0.0002**
	200	76.17	0.001	1	1	0.200	0.002	0	1	0.002
	400	73.35	0.002	1	1	0.385	0.004	0	1	0.004
	800	70.99	0.005	1	1	0.672	0.007	0	1	0.007
Ariel Iraq2	30	84.58	0.0002	1	1	0.031	0.0003	0	1	0.0003
	200	76.81	0.001	1	1	0.166	0.001	0	1	0.001
	400	73.77	0.002	1	1	0.346	0.004	0	1	0.004
	800	71.07	0.005	1	1	0.657	0.007	0	1	0.007
Image resolution 1024*1024										
Lena	30	90.38	0.00005	1	1	0.007	0.00008	0	1	0.00008
	200	83.02	0.0003	1	1	0.041	0.0004	0	1	0.0004
	400	79.77	0.0006	1	1	0.086	0.001	0	1	0.001
	800	77.01	0.0012	1	1	0.165	0.001	0	1	0.001
Baboon	30	90.82	0.00005	1	1	0.007	0.00008	0	1	0.00008
	200	82.98	0.0003	1	1	0.040	0.0004	0	1	0.0004
	400	79.67	0.0007	1	1	0.090	0.001	0	1	0.001
	800	77.05	0.0012	1	1	0.160	0.001	0	1	0.001
Pepper	30	90.85	0.00005	1	1	0.008	0.00008	0	1	0.0001
	200	82.86	0.0003	1	1	0.043	0.0005	0	1	0.0005
	400	79.55	0.0007	1	1	0.096	0.001	0	1	0.001
	800	77.03	0.0012	1	1	0.173	0.002	0	1	0.002
House	30	**91.34**	**0.00004**	**1**	**1**	**0.007**	**0.00008**	**0**	**1**	**0.00008**
	200	83.13	0.0003	1	1	0.036	0.0004	0	1	0.0004
	400	79.79	0.0006	1	1	0.066	0.0007	0	1	0.0007
	800	77.11	0.0012	1	1	0.118	0.001	0	1	0.001
Plane	30	90.50	0.00005	1	1	0.009	0.0001	0	1	0.0001
	200	82.78	0.0003	1	1	0.044	0.0005	0	1	0.0005
	400	79.72	0.0006	1	1	0.084	0.0009	0	1	0.0009
	800	**76.99**	**0.0012**	**1**	**1**	**0.168**	**0.001**	**0**	**1**	**0.001**
Ariel Iraq1	30	90.65	0.00005	1	1	0.007	0.00008	0	1	**0.00008**
	200	82.97	0.0003	1	1	0.042	0.0004	0	1	0.0004
	400	79.80	0.0006	1	1	0.088	0.001	0	1	0.001
	800	77.02	0.0012	1	1	0.168	0.001	0	1	0.001
Ariel Iraq2	30	90.55	0.00005	1	1	0.006	0.00007	0	1	**0.00007**
	200	82.91	0.0003	1	1	0.041	0.0004	0	1	0.0004
	400	79.78	0.0006	1	1	0.085	0.001	0	1	0.001

	800	77.02	0.0012	1	1	0.163	0.001	0	1	0.001

Figures 4.10 and 4.11 show the best PSNR and MSE values obtained as a result of the data hiding process at a text size of 30 bytes on seven different colored images with different resolutions using the LSB method. The analysis reveals that the PSNR values exhibit a range of 78.56 dB to 91.34 dB across the different images with different resolutions using block size of 16x16.

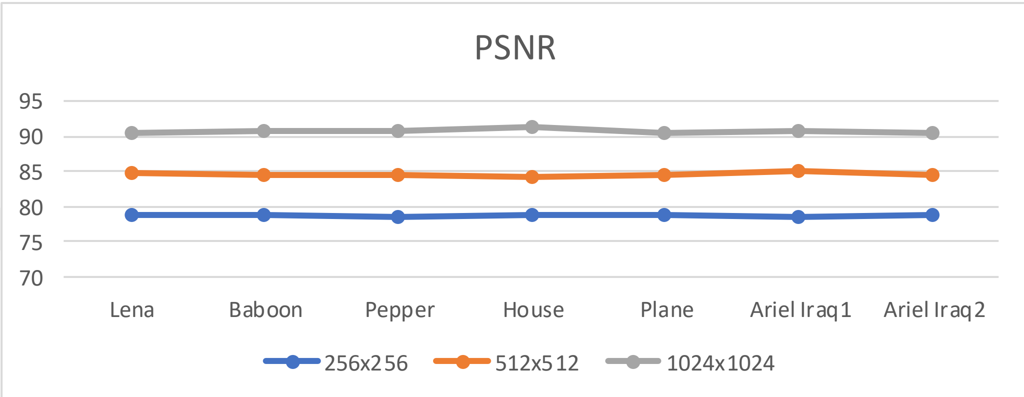


Figure 4.10. Best PSNR achieved using LSB approach in seven coloured images at different resolutions using block size of 16x16

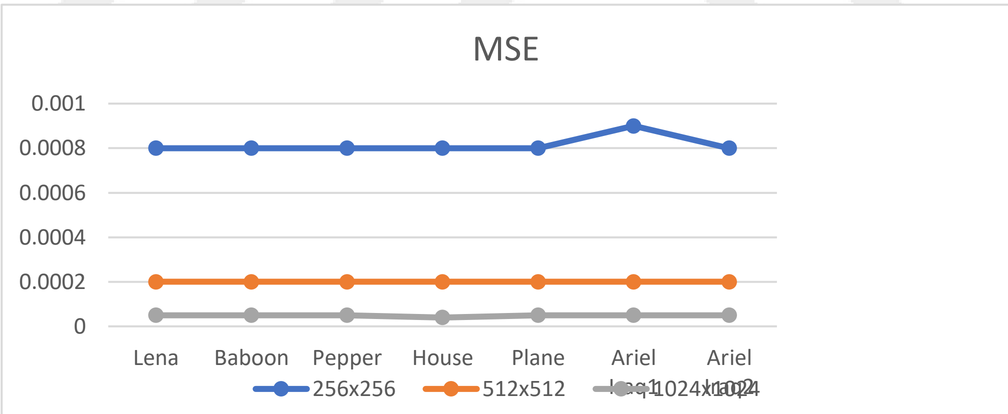


Figure 4.11. Best MSE achieved using LSB approach in seven colored images at different resolutions using block size of 16x16

Table 4.2 presents the results of applying the LSB method with block size of 32x32 to color images of different resolutions. The findings indicate that the results obtained with a block size of 32x32 are very similar to those obtained with a block size of 16x16 for all tested color images. This similarity suggests that increasing the block size from 16x16 to 32x32 did not significantly impact the performance of the LSB steganography method on these images. The observed consistency in results between the two block sizes implies that the LSB method is robust and stable regarding the choice of block size.

Table 4.2. Results of statistical parameters obtained from applying the LSB approach on color images with different text sizes of block size of 32x32

Image Name	Text Size (byte)	PSNR (dB)	MSE	UQI	MSSIM	CQM	AD	SC	NCC	NAE
Image resolution 256*256										
Lena	30	77.92	0.001	1	1	0.132	0.001	0	1	0.001
	200	70.89	0.005	1	1	0.657	0.007	0	1	0.007
	400	67.70	0.011	1	1	1.406	0.016	0	1	0.016
	800	65.04	0.020	1	1	2.617	0.030	0	1	0.030
Baboon	30	**78.91**	**0.0008**	**1**	**1**	**0.100**	**0.001**	**0**	**1**	**0.001**
	200	70.84	0.005	1	1	0.702	0.008	0	1	0.008
	400	67.64	0.011	1	1	1.472	0.017	0	1	0.017
	800	65.02	0.020	1	1	2.675	0.031	0	1	0.031
Pepper	30	78.30	0.0009	1	1	0.127	0.001	0	1	0.001
	200	70.77	0.005	1	1	0.713	0.008	0	1	0.008
	400	67.70	0.011	1	1	1.433	0.016	0	1	0.016
	800	**64.92**	**0.020**	**1**	**1**	**2.805**	**0.033**	**0**	**1**	**0.033**
House	30	78.83	0.0008	1	1	0.126	0.001	0	1	0.001
	200	70.80	0.005	1	1	0.700	0.008	0	1	0.008
	400	67.83	0.010	1	1	1.334	0.015	0	1	0.015
	800	65.11	0.020	1	1	2.556	0.030	0	1	0.030
Plane	30	78.91	0.0008	1	1	0.109	0.001	0	1	0.001
	200	70.90	0.005	1	1	0.754	0.008	0	1	0.008
	400	67.84	0.010	1	1	1.411	0.016	0	1	0.016
	800	64.93	0.020	1	1	2.736	0.032	0	1	0.032
Ariel Iraq1	30	78.56	0.0009	1	1	0.127	0.001	0	1	0.001
	200	70.93	0.005	1	1	0.667	0.007	0	1	0.007
	400	67.66	0.011	1	1	1.429	0.016	0	1	0.016
	800	64.98	0.020	1	1	2.660	0.031	0	1	0.031
Ariel Iraq2	30	78.68	0.0008	1	1	0.113	0.001	0	1	0.001
	200	70.76	0.005	1	1	0.694	0.008	0	1	0.008
	400	67.83	0.010	1	1	1.376	0.016	0	1	0.016
	800	64.92	0.020	1	1	2.606	0.030	0	1	0.030
Image resolution 512*512										
Lena	30	84.75	0.0002	1	1	0.027	0.0003	0	1	0.0003
	200	77.03	0.0012	1	1	0.158	0.001	0	1	0.001
	400	73.82	0.0026	1	1	0.345	0.004	0	1	0.004
	800	71.04	0.0051	1	1	0.648	0.007	0	1	0.007
Baboon	30	84.88	0.0002	1	1	0.030	0.0003	0	1	0.0003
	200	76.90	0.0013	1	1	0.176	0.002	0	1	0.002
	400	73.71	0.0027	1	1	0.362	0.004	0	1	0.004
	800	**70.94**	**0.0052**	**1**	**1**	**0.686**	**0.008**	**0**	**1**	**0.008**
Pepper	30	84.51	0.0002	1	1	0.032	0.0003	0	1	0.0003
	200	76.74	0.0013	1	1	0.179	0.002	0	1	0.002
	400	73.66	0.0027	1	1	0.352	0.004	0	1	0.004
	800	70.96	0.0052	1	1	0.659	0.007	0	1	0.007
House	30	85.07	0.0002	1	1	0.024	0.0002	0	1	0.0002
	200	76.84	0.0013	1	1	0.173	0.002	0	1	0.002
	400	73.68	0.0027	1	1	0.370	0.004	0	1	0.004
	800	85.07	0.0002	1	1	0.024	0.0002	0	1	0.0002
Plane	30	**85.32**	**0.0001**	**1**	**1**	**0.024**	**0.0002**	**0**	**1**	**0.0002**
	200	76.90	0.0013	1	1	0.166	0.001	0	1	0.001

	400	73.69	0.0027	1	1	0.353	0.004	0	1	0.004
	800	70.95	0.0052	1	1	0.676	0.007	0	1	0.007
Ariel Iraq1	30	84.55	0.0002	1	1	0.028	0.0003	0	1	0.0003
	200	76.95	0.0013	1	1	0.175	0.002	0	1	0.002
	400	73.71	0.0027	1	1	0.358	0.004	0	1	0.004
	800	71.02	0.0051	1	1	0.656	0.007	0	1	0.007
Ariel Iraq2	30	84.46	0.0002	1	1	0.028	0.0003	0	1	0.0003
	200	77.03	0.0012	1	1	0.155	0.001	0	1	0.001
	400	73.75	0.0027	1	1	0.351	0.004	0	1	0.004
	800	71.03	0.0051	1	1	0.674	0.007	0	1	0.007
Image resolution 1024*1024										
Lena	30	90.65	0.00005	1	1	0.008	0.00009	1	1	0.00009
	200	83.00	0.0003	1	1	0.044	0.0005	0	1	0.0005
	400	79.82	0.0006	1	1	0.088	0.001	0	1	0.001
	800	77.02	0.0012	1	1	0.164	0.001	0	1	0.001
Baboon	30	90.80	0.00005	1	1	0.007	8.517	1	1	0.00008
	200	82.94	0.0003	1	1	0.043	0.0005	0	1	0.0005
	400	79.79	0.0006	1	1	0.090	0.001	0	1	0.001
	800	77.03	0.0012	1	1	0.168	0.001	0	1	0.001
Pepper	30	90.75	0.00005	1	1	0.007	0.00008	1	1	0.00008
	200	82.90	0.0003	1	1	0.040	0.0004	0	1	0.0004
	400	79.81	0.0006	1	1	0.085	0.001	0	1	0.001
	800	77.10	0.0012	1	1	0.161	0.001	0	1	0.001
House	30	90.45	0.00005	1	1	0.007	0.00009	1	1	0.00009
	200	83.11	0.0003	1	1	0.039	0.0004	0	1	0.0004
	400	79.74	0.0006	1	1	0.086	0.001	0	1	0.001
	800	77.05	0.0012	1	1	0.163	0.001	0	1	0.001
Plane	30	90.95	0.00005	1	1	0.006	7.280	1	1	0.00007
	200	82.76	0.0003	1	1	0.042	0.0004	0	1	0.0004
	400	79.70	0.0006	1	1	0.087	0.001	0	1	0.001
	800	**76.94**	**0.0013**	**1**	**1**	**0.169**	**0.001**	**0**	**1**	**0.001**
Ariel Iraq1	30	**91.01**	**0.00005**	**1**	**1**	**0.006**	**0.00007**	**1**	**1**	0.00007
	200	82.86	0.0003	1	1	0.043	0.0005	0	1	0.0005
	400	79.74	0.0006	1	1	0.086	0.001	0	1	0.001
	800	77.08	0.001	1	1	0.162	0.001	0	1	0.001
Ariel Iraq2	30	90.82	0.00005	1	1	0.006	0.00007	1	1	0.00007
	200	82.96	0.0003	1	1	0.040	0.0004	0	1	0.0004
	400	79.86	0.0006	1	1	0.084	0.0009	0	1	0.0009
	800	76.97	0.0013	1	1	0.166	0.001	0	1	0.001

Figures 4.12 and 4.13 show the best PSNR and MSE values obtained as a result of the data hiding process at a text size of 30 bytes on seven different colored images with different resolutions using the LSB method. The analysis reveals that the PSNR values exhibit a range of 77.92 dB to 91.01 dB across the different images with different resolutions using block size of 32x32.

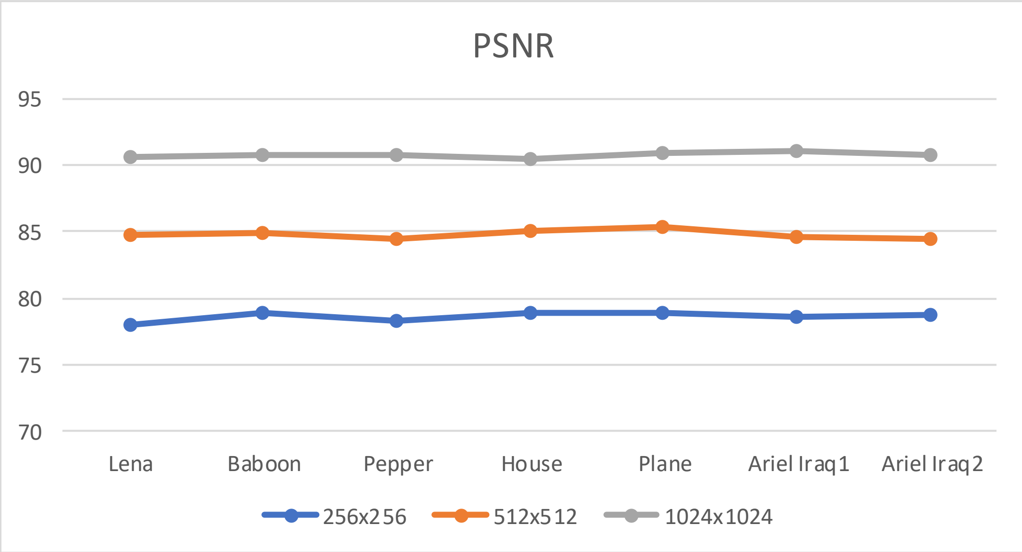


Figure 1.12. Best PSNR achieved using LSB approach in seven colored images at different resolutions using block size of 32x32

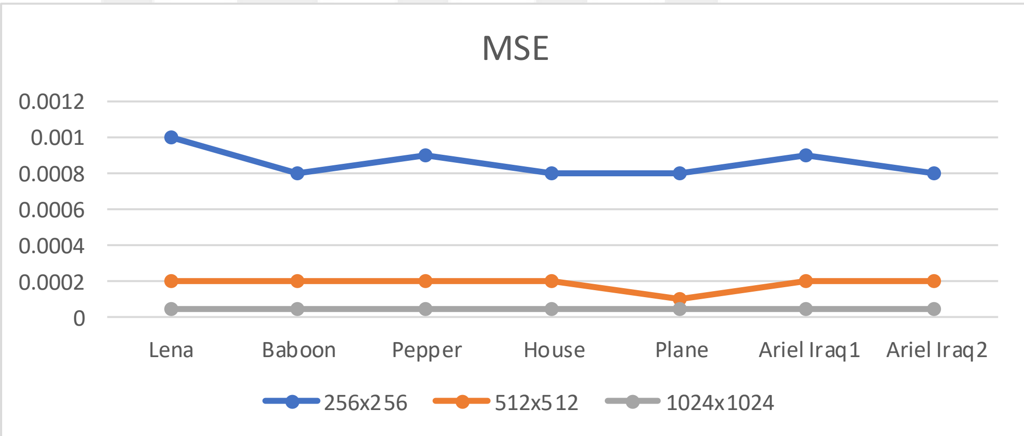


Figure 4.13. Best MSE achieved using LSB approach in seven colored images at different resolutions using block size of 32x32

4.4.2 Experimental results of the proposed AMBD approach

In the experimental study, the Absolute Mean Brightness Deviation (AMBD) technique was employed for embedding text in images. Similar to the LSB technique, the images were divided into blocks of either 16x16 or 32x32 pixels. Out of these blocks, 128 blocks with the highest AMBD values were chosen to embed secret messages ranging in size from 30 to 800 bytes. To evaluate the effectiveness of the AMBD technique, a comparison was made between the resulting stego image and the original image. Both images were converted to grayscale to examine their pixel intensities. Histograms of pixel values were plotted to comprehensively assess the distribution and intensity of pixels. This analysis provided valuable insights into the impact of the data embedding process on the final image.

Figure 4.14 shows the results of processing the Lina image using the AMBD technique with a block size of 16x16. A visual comparison between the original and processed images demonstrates a remarkable similarity, indicating the effective preservation of essential image properties. However, a detailed histogram analysis reveals some minor distortions, with a few frequency increases in pixel density and slight variations in pixel intensity values.

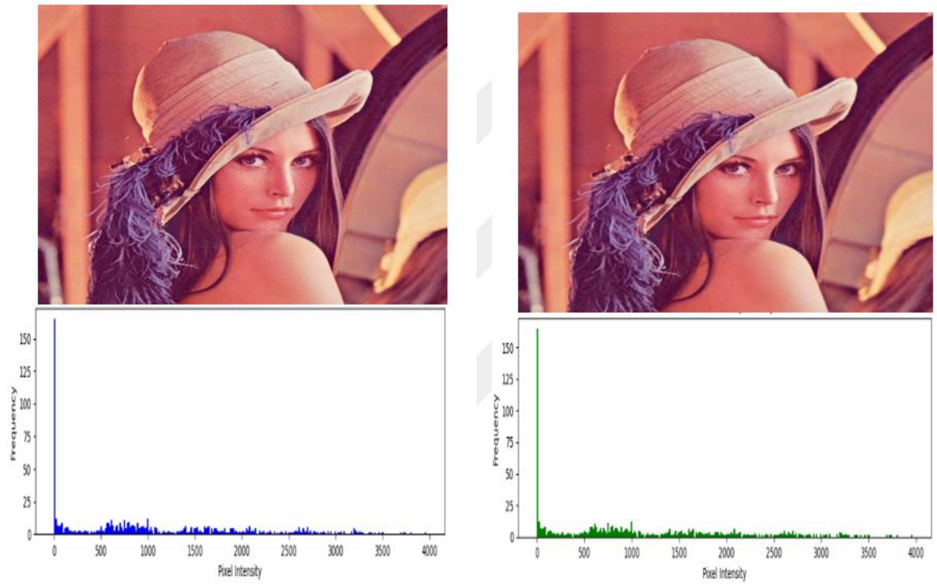


Figure 4.14. Histogram graph of the original and stego of the Iimage named Lena using the AMBD approach with block size of 16x16

In Figure 4.15, the results of applying the AMBD technique with a block size of 32x32 to the Lina image for the data embedding process are displayed. The histogram distribution of pixel density in the image processed with a block size of 32x32 closely resembles the histogram distribution in the image processed with a block size of 16x16. While the overall patterns are almost the same, there are some minor discrepancies between the two distributions.

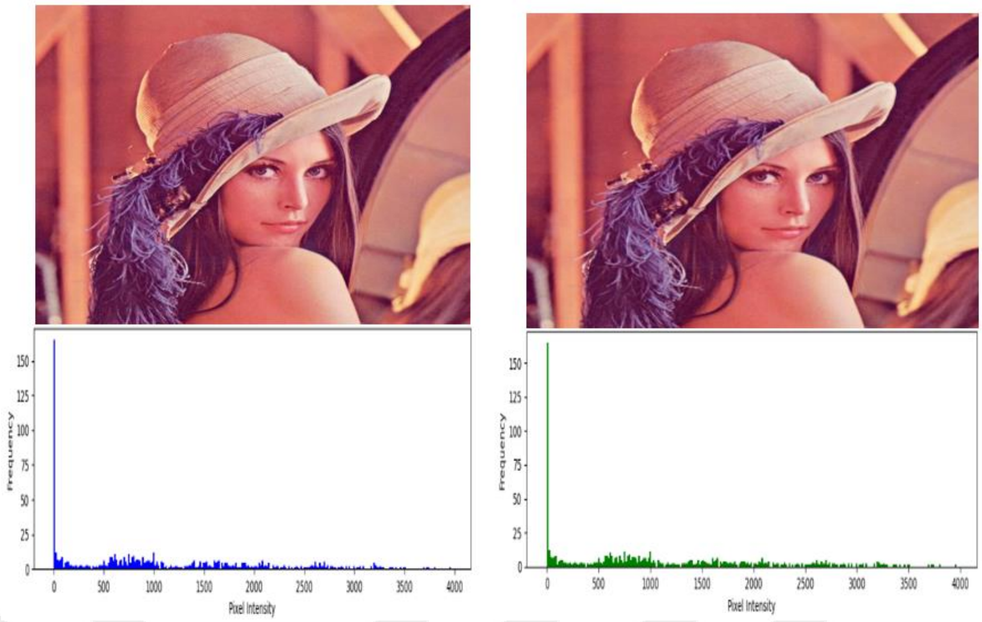


Figure 4.15. Histogram graph of the original and stego of the image named Lena using the AMBD approach with block size of 32x32

Table 4.3 presents the experimental results of using the AMBD method with a block size of 16x16 on color images of different resolutions. Increasing the text size used for data hiding consistently decreased the PSNR values, indicating a trade-off between data capacity and image quality. As more data was embedded into the images, the least significant bits of the green channel underwent more modifications, leading to increased distortion and lower PSNR values. The PSNR values for color images at resolutions 256, 512, and 1024 ranged from 64.903 to 78.97, 70.90 to 85.01, and 72.56 to 91.23, respectively. This suggests that higher-resolution images generally achieved higher PSNR values, but the trend of decreasing PSNR with increasing text size remained consistent across all tested images.

On the other hand, for all the studied color images, the MSE values rose as text size did. A lower MSE score often denotes a higher level of image steganography quality. The text sizes of 30 bytes produced the lowest MSE values. The measured MSE varied from 0.0008 to 0.0210, 0.0002 to 0.0052, and 0.00004 to 0.0036, respectively, for color images at sizes of 256, 512, and 1024. Additionally, it can be deduced that the MSE is connected to the number of color variations in the cover image, with smaller MSE values occurring when color variations are greater.

The UQI and MSSIM metrics both returned a value of one, indicating that the original and stego images were identical. Lower values for text size 30 indicated less distortion and higher quality in the stego images, according to the CQM metric.

Furthermore, the AD, SC, and NAE metrics had low values, indicating that the stego images were structurally and contentally similar to the original images.

Table 4.3. Results of statistical parameters obtained from applying the AMBD approach on color images with different text sizes of block size of 16x16

Image name	Text Size (byte)	PSNR (dB)	MSE	UQI	MSSIM	CQM	AD	SC	NCC	NAE
Image Resolution 256*256										
Lena	30	78.83	0.0008	1	1	0.109	0.001	0	1	0.001
	200	70.98	0.0051	1	1	0.667	0.007	0	1	0.007
	400	67.86	0.0106	1	1	1.371	0.016	0	1	0.016
	800	65.03	0.0203	1	1	2.641	2.641	0	1	0.031
Baboon	30	**78.97**	**0.0008**	**1**	**1**	**0.108**	**0.001**	**0**	**1**	**0.001**
	200	70.76	0.0054	1	1	0.709	0.008	0	1	0.008
	400	67.73	0.0109	1	1	1.410	0.016	0	1	0.016
	800	64.96	0.0207	1	1	2.707	0.031	0	1	0.031
Pepper	30	78.83	0.0008	1	1	0.106	0.001	0	1	0.001
	200	70.91	0.0052	1	1	0.671	0.007	0	1	0.007
	400	67.67	0.0111	1	1	1.441	0.016	0	1	0.016
	800	65.08	0.0201	1	1	2.659	0.031	0	1	0.031
House	30	78.56	0.0009	1	1	0.128	0.001	0	1	0.001
	200	70.74	0.0054	1	1	0.718	0.008	0	1	0.008
	400	67.60	0.0112	1	1	1.441	0.016	0	1	0.016
	800	**64.90**	**0.0210**	**1**	**1**	**2.730**	**0.032**	**0**	**1**	**0.032**
Plane	30	78.91	0.0008	1	1	0.122	0.001	0	1	0.001
	200	70.83	0.0053	1	1	0.699	0.008	0	1	0.008
	400	67.73	0.0109	1	1	1.426	0.016	0	1	0.016
	800	64.96	0.0207	1	1	2.658	0.031	0	1	0.031
Ariel Iraq1	30	78.76	0.0008	1	1	0.093	0.001	0	1	0.001
	200	70.98	0.0051	1	1	0.664	0.007	0	1	0.007
	400	67.62	0.0112	1	1	1.400	0.016	0	1	0.016
	800	64.94	0.0208	1	1	2.687	0.031	0	1	0.031
Ariel Iraq2	30	78.78	0.0008	1	1	0.117	0.001	0	1	0.001
	200	70.82	0.0053	1	1	0.691	0.008	0	1	0.008
	400	67.68	0.0110	1	1	1.464	0.017	0	1	0.017
	800	64.95	0.0207	1	1	2.654	0.031	0	1	0.031
Image Resolution 512*512										
Lena	30	**85.01**	**0.0002**	**1**	**1**	**0.024**	**0.0002**	**0**	**1**	**0.0002**
	200	77.03	0.0012	1	1	0.168	0.0019	0	1	0.0019
	400	73.80	0.0027	1	1	0.347	0.004	0	1	0.004
	800	71.05	0.0050	1	1	0.652	0.007	0	1	0.007
Baboon	30	84.63	0.0002	1	1	0.027	0.0003	0	1	0.0003
	200	76.75	0.0013	1	1	0.178	0.002	0	1	0.002
	400	73.64	0.0028	1	1	0.370	0.004	0	1	0.004
	800	**70.90**	**0.0052**	**1**	**1**	**0.678**	**0.007**	**0**	**1**	**0.007**
Pepper	30	84.80	0.0002	1	1	0.031	0.0003	0	1	0.0003
	200	76.79	0.0013	1	1	0.175	0.002	0	1	0.002
	400	73.69	0.0027	1	1	0.353	0.004	0	1	0.004
	800	70.95	0.0052	1	1	0.653	0.007	0	1	0.007
House	30	84.48	0.0002	1	1	0.041	0.0004	0	1	0.0004
	200	76.81	0.0013	1	1	0.181	0.002	0	1	0.002
	400	73.86	0.0026	1	1	0.358	0.004	0	1	0.004
	800	71.02	0.0051	1	1	0.672	0.007	0	1	0.007
Plane	30	84.73	0.0002	1	1	0.015	0.0001	0	1	0.0001
	200	76.88	0.0013	1	1	0.161	0.001	0	1	0.001
	400	73.80	0.0027	1	1	0.350	0.004	0	1	0.004

	800	71.05	0.0051	1	1	0.641	0.007	0	1	0.007
Ariel Iraq1	30	84.43	0.0002	1	1	0.024	0.0002	0	1	0.0002
	200	76.93	0.0013	1	1	0.170	0.002	0	1	0.002
	400	73.70	0.0027	1	1	0.338	0.003	0	1	0.003
	800	71.05	0.0051	1	1	0.656	0.007	0	1	0.007
Ariel Iraq2	30	84.80	0.0002	1	1	0.026	0.0003	0	1	0.0003
	200	76.88	0.0013	1	1	0.169	0.001	0	1	0.001
	400	73.77	0.0027	1	1	0.357	0.004	0	1	0.004
	800	70.98	0.0051	1	1	0.666	0.007	0	1	0.007
Image Resolution 1024*1024										
Lena	30	90.85	0.00005	1	1	0.007	0.00008	0	1	0.00008
	200	82.92	0.0003	1	1	0.042	0.0004	0	1	0.0004
	400	79.80	0.0006	1	1	0.089	0.001	0	1	0.001
	800	77.12	0.0012	1	1	0.163	0.001	0	1	0.001
Baboon	30	90.82	0.00005	1	1	0.006	0.00007	0	1	0.00007
	200	82.98	0.0003	1	1	0.040	0.0004	0	1	0.0004
	400	79.75	0.0006	1	1	0.087	0.001	0	1	0.001
	800	76.94	0.0013	1	1	0.167	0.001	0	1	0.001
Pepper	30	90.67	0.00005	1	1	0.007	0.00008	0	1	0.00008
	200	82.86	0.0003	1	1	0.043	0.0005	0	1	0.0005
	400	79.63	0.0007	1	1	0.090	0.001	0	1	0.001
	800	77.02	0.0012	1	1	0.167	0.001	0	1	0.001
House	30	90.53	0.00005	1	1	0.008	0.00009	0	1	0.00006
	200	83.13	0.0003	1	1	0.036	0.0004	0	1	0.0004
	400	79.68	0.0006	1	1	0.084	0.0009	0	1	0.0009
	800	77.08	0.0012	1	1	0.159	0.001	0	1	0.001
Plane	30	**91.23**	**0.00004**	**1**	**1**	**0.005**	**0.00006**	**0**	**1**	**0.00006**
	200	82.93	0.0003	1	1	0.031	0.0003	0	1	0.0003
	400	79.65	0.0007	1	1	0.061	0.0007	0	1	0.0007
	800	76.79	0.0013	1	1	0.100	0.001	0	1	0.001
Ariel Iraq1	30	91.17	0.00004	1	1	0.006	0.00007	0	1	0.00007
	200	82.82	0.0003	1	1	0.043	0.0005	0	1	0.0005
	400	79.84	0.0006	1	1	0.084	0.0009	0	1	0.0009
	800	77.09	0.0012	1	1	0.163	0.001	0	1	0.001
Ariel Iraq2	30	90.88	0.00005	1	1	0.006	0.00007	0	1	0.00007
	200	80.28	0.0006	1	1	0.058	0.0006	0	1	0.0006
	400	79.74	0.0006	1	1	0.088	0.001	0	1	0.001
	800	**72.56**	**0.0036**	**1**	**1**	**0.296**	**0.003**	**0**	**1**	**0.003**

Figures 4.16 and 4.17 show the best PSNR and MSE values obtained as a result of the data hiding process at a text size of 30 bytes on seven different colored images with different resolutions using the AMBD method. The analysis reveals that the PSNR values exhibit a range of 78.56 dB to 91.23 dB across the different images with different resolutions using block size of 16x16.

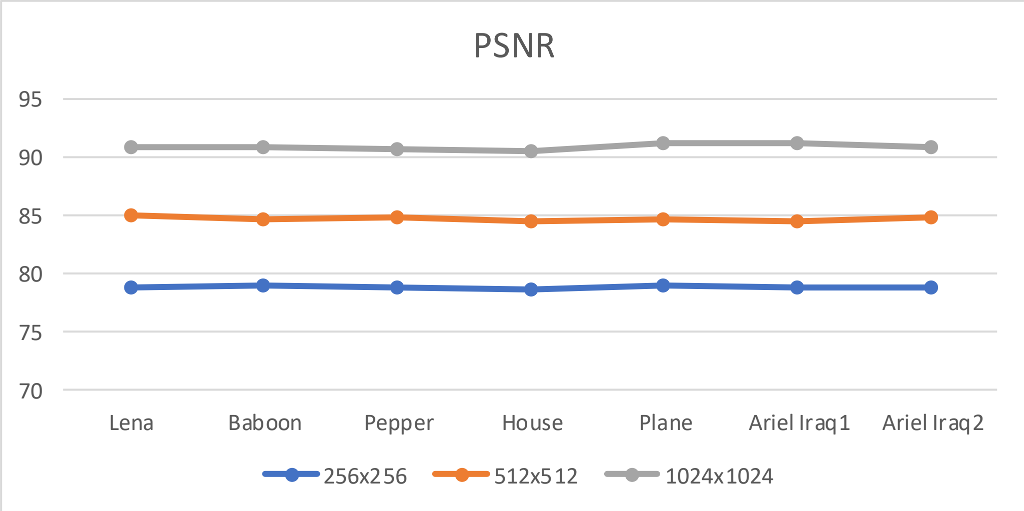


Figure 4.16. Best PSNR achieved using AMBD approach in seven colored images at different resolutions using block size of 16x16

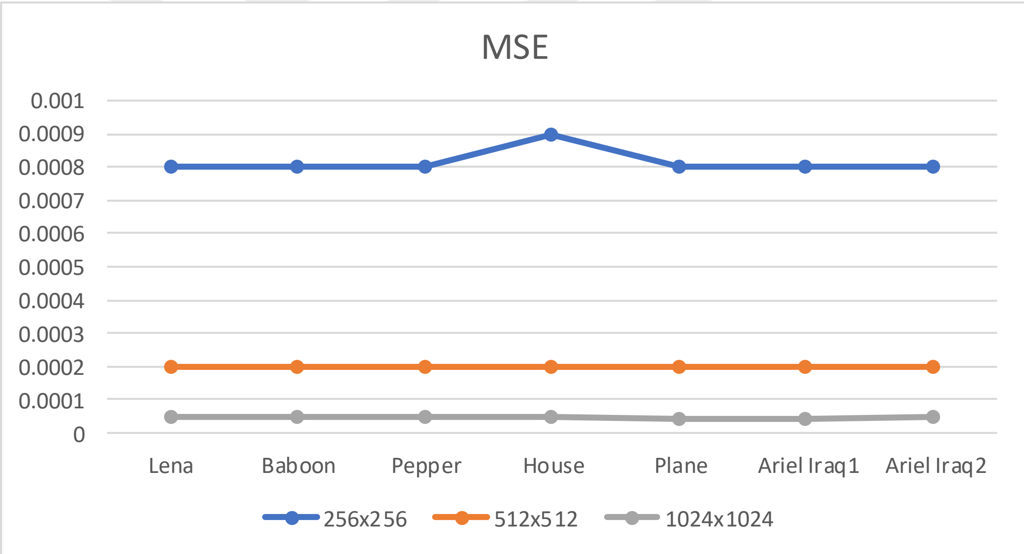


Figure 4.17. Best MSE achieved using AMBD approach in seven colored images at different resolutions using block size of 16x16

Table 4.4 displays the results of utilizing the AMBD method with block sizes of 32x32 on color images of varying resolutions. The findings indicate that the outcomes obtained with a block size of 32x32 are highly similar to those obtained with a block size of 16x16 for all tested color images. This closeness in results suggests that increasing the block size from 16x16 to 32x32 did not have a substantial impact on the performance of the AMBD steganography method for these images. The consistency between the results obtained with both block sizes indicates that the AMBD method is robust and stable, irrespective of the chosen block size.

Table 4.4. Results of statistical parameters obtained from applying the AMBD approach on color images with different text sizes of block size of 32x32

Image name	Text Size (byte)	PSNR (dB)	MSE	UQI	MSSIM	CQM	AD	SC	NCC	NAE
Image Resolution 256*256										
Lena	30	78.99	0.0008	1	1	0.119	0.001	0	1	0.001
	200	71.06	0.0050	1	1	0.631	0.007	0	1	0.007
	400	67.75	0.0108	1	1	1.474	0.017	0	1	0.017
	800	64.95	0.0207	1	1	2.718	0.031	0	1	0.031
Baboon	30	78.41	0.0009	1	1	0.118	0.001	0	1	0.001
	200	70.90	0.0052	1	1	0.695	0.008	0	1	0.008
	400	67.68	0.0110	1	1	1.433	0.016	0	1	0.016
	800	64.95	0.0207	1	1	2.749	0.032	0	1	0.032
Pepper	30	78.78	0.0008	1	1	0.122	0.001	0	1	0.001
	200	70.76	0.0054	1	1	0.743	0.008	0	1	0.008
	400	67.67	0.0111	1	1	1.447	0.017	0	1	0.017
	800	**64.94**	**0.0208**	**1**	**1**	**2.802**	**0.032**	**0**	**1**	**0.032**
House	30	78.68	0.0008	1	1	0.113	0.001	0	1	0.001
	200	70.92	0.0052	1	1	0.678	0.007	0	1	0.007
	400	67.77	0.0108	1	1	1.357	0.015	0	1	0.015
	800	64.98	0.0206	1	1	2.644	0.031	0	1	0.031
Plane	30	79.02	0.0008	1	1	0.098	0.001	0	1	0.001
	200	70.74	0.0054	1	1	0.703	0.008	0	1	0.008
	400	67.70	0.0110	1	1	1.420	0.016	0	1	0.016
	800	65.02	0.0204	1	1	2.575	0.030	0	1	0.030
Ariel Iraq1	30	78.46	0.0009	1	1	0.133	0.001	0	1	0.001
	200	70.87	0.0053	1	1	0.706	0.008	0	1	0.008
	400	67.81	0.0107	1	1	1.442	0.016	0	1	0.016
	800	64.98	0.0206	1	1	2.685	0.031	0	1	0.031
Ariel Iraq2	30	**79.08**	**0.0008**	**1**	**1**	**0.115**	**0.001**	**0**	**1**	**0.001**
	200	70.88	0.0053	1	1	0.684	0.008	0	1	0.008
	400	67.76	0.0108	1	1	1.385	0.016	0	1	0.016
	800	65.03	0.0203	1	1	2.608	0.030	0	1	0.030
Image Resolution 512*512										
Lena	30	84.75	0.0002	1	1	0.027	0.0003	0	1	0.0003
	200	76.81	0.0013	1	1	0.158	0.001	0	1	0.001
	400	73.79	0.0027	1	1	0.318	0.003	0	1	0.003
	800	71.02	0.0051	1	1	0.635	0.007	0	1	0.007
Baboon	30	84.86	0.0002	1	1	0.027	0.0003	0	1	0.0003
	200	76.92	0.0013	1	1	0.165	0.001	0	1	0.001
	400	73.69	0.0027	1	1	0.358	0.004	0	1	0.004
	800	71.07	0.0050	1	1	0.634	0.007	0	1	0.007
Pepper	30	84.63	0.0002	1	1	0.032	0.0003	0	1	0.0003
	200	76.79	0.0013	1	1	0.175	0.002	0	1	0.002
	400	73.77	0.0027	1	1	0.348	0.004	0	1	0.004
	800	71.07	0.0050	1	1	0.651	0.007	0	1	0.007
House	30	84.48	0.0002	1	1	0.028	0.0003	0	1	0.0003
	200	76.76	0.0013	1	1	0.169	0.001	0	1	0.001
	400	73.64	0.0028	1	1	0.363	0.004	0	1	0.004
	800	71.09	0.005	1	1	0.648	0.007	0	1	0.007
Plane	30	**84.91**	**0.0002**	**1**	**1**	**0.027**	**0.0003**	**0**	**1**	**0.0003**
	200	77.01	0.0012	1	1	0.173	0.002	0	1	0.002

	400	73.85	0.0026	1	1	0.340	0.004	0	1	0.004
	800	71.06	0.0050	1	1	0.645	0.007	0	1	0.007
Ariel Iraq1	30	84.68	0.0002	1	1	0.026	0.0003	0	1	0.0003
	200	76.85	0.0013	1	1	0.175	0.002	0	1	0.002
	400	73.68	0.0027	1	1	0.358	0.004	0	1	0.004
	800	**70.95**	**0.0052**	**1**	**1**	**0.678**	**0.007**	**0**	**1**	**0.007**
Ariel Iraq2	30	84.43	0.0002	1	1	0.026	0.0003	0	1	0.0003
	200	76.87	0.0013	1	1	0.170	0.002	0	1	0.002
	400	73.74	0.0027	1	1	0.341	0.004	0	1	0.004
	800	70.98	0.0051	1	1	0.655	0.007	0	1	0.007
Image Resolution 1024*1024										
Lena	30	**91.03**	**0.00005**	**1**	**1**	**0.006**	**0.00008**	**0**	**1**	**0.00008**
	200	82.97	0.0003	1	1	0.041	0.0005	0	1	0.0005
	400	79.78	0.0006	1	1	0.089	0.001	0	1	0.001
	800	77.00	0.0012	1	1	0.166	0.001	0	1	0.001
Baboon	30	90.93	0.00005	1	1	0.007	0.00008	0	1	0.00008
	200	82.91	0.0003	1	1	0.043	0.0005	0	1	0.0005
	400	79.81	0.0006	1	1	0.087	0.001	0	1	0.001
	800	76.97	0.0013	1	1	0.168	0.001	0	1	0.001
Pepper	30	90.72	0.00005	1	1	0.006	0.00007	0	1	0.00007
	200	82.83	0.0003	1	1	0.040	0.0004	0	1	0.0004
	400	79.67	0.0007	1	1	0.086	0.001	0	1	0.001
	800	76.96	0.0013	1	1	0.164	0.001	0	1	0.001
House	30	90.22	0.00006	1	1	0.001	0.00001	0	1	0.00001
	200	82.74	0.0003	1	1	0.032	0.0003	0	1	0.0003
	400	79.83	0.0006	1	1	0.085	0.001	0	1	0.001
	800	76.98	0.0013	1	1	0.161	0.001	0	1	0.001
Plane	30	90.82	0.00005	1	1	0.006	0.00007	0	1	0.00007
	200	82.92	0.0003	1	1	0.040	0.0004	0	1	0.0004
	400	79.64	0.0007	1	1	0.084	0.0009	0	1	0.0009
	800	77.02	0.0012	1	1	0.159	0.001	0	1	0.001
Ariel Iraq1	30	91.01	0.00005	1	1	0.006	0.00007	0	1	0.00007
	200	83.01	0.0003	1	1	0.040	0.0004	0	1	0.0004
	400	79.79	0.0006	1	1	0.084	0.0009	0	1	0.0009
	800	**76.95**	**0.0013**	**1**	**1**	**0.163**	**0.001**	**0**	**1**	**0.001**
Ariel Iraq2	30	90.82	0.00005	1	1	0.006	0.00007	0	1	0.00007
	200	82.86	0.0003	1	1	0.042	0.0005	0	1	0.0005
	400	79.73	0.0006	1	1	0.086	0.001	0	1	0.001
	800	76.97	0.0013	1	1	0.164	0.001	0	1	0.001

Figures 4.18 and 4.19 show the best PSNR and MSE values obtained as a result of the data hiding process at a text size of 30 bytes on seven different colored images with different resolutions using the AMBD method. The analysis reveals that the PSNR values exhibit a range of 78.41 dB to 91.03 dB across the different images with different resolutions using block size of 32x32.

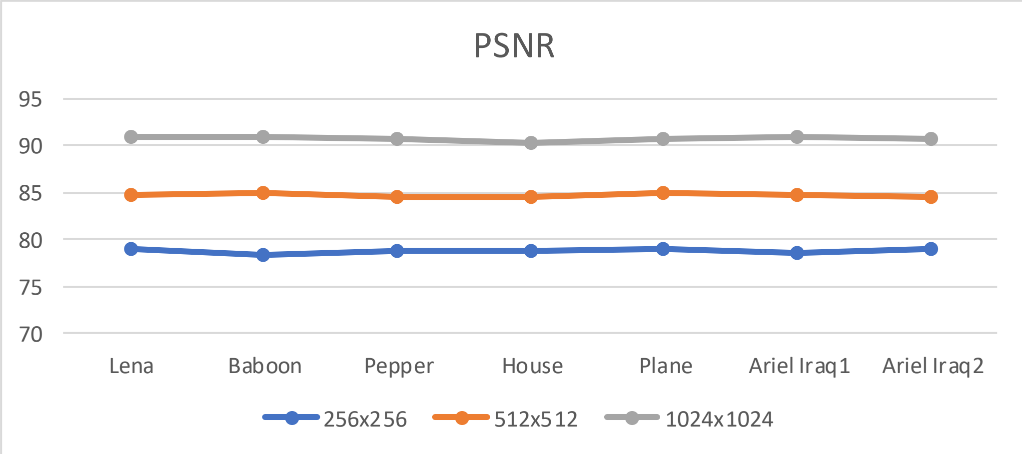


Figure 2.18. Best PSNR achieved using AMBD appraoch in seven colored images at different resolutions using block size of 32x32

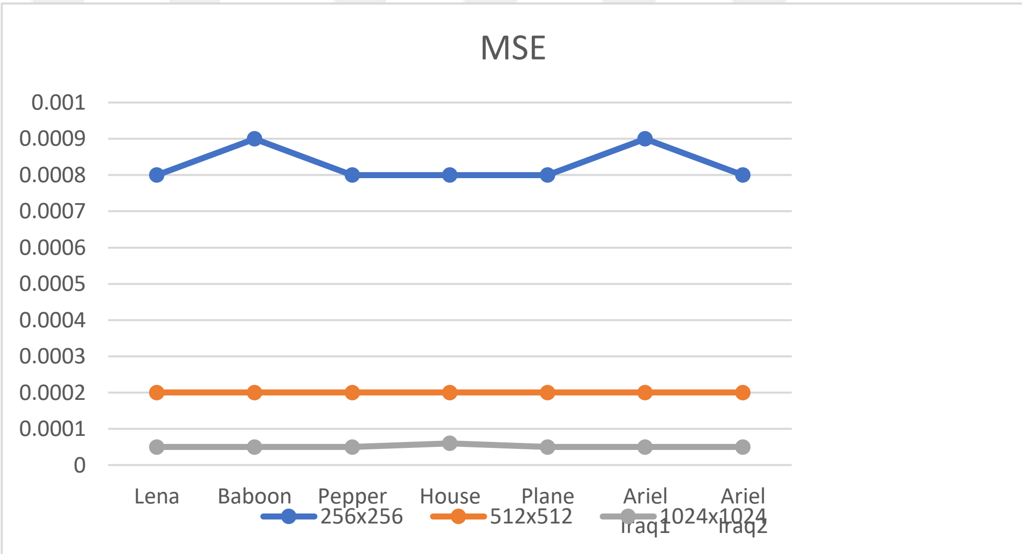


Figure 3.19. Best MSE achieved using AMBD approach in seven colored images at different resolutions using block size of 32x32

4.4.3 Experimental results of the proposed Entropy approach

In the experimental study, the Entropy based method technique was employed for embedding text in images. Similar to the LSB and AMBD techniques, the images were divided into blocks of either 16x16 or 32x32 pixels. Out of these blocks, 128 blocks with the highest Entropy values were chosen to embed secret messages ranging in size from 30 to 800 bytes. To evaluate the effectiveness of the Entropy technique, a comparison was made between the resulting stego image and the original image. Both images were converted to grayscale to examine their pixel intensities. Histograms of pixel values were plotted to comprehensively assess the distribution and intensity of pixels. This analysis provided valuable insights into the impact of the data embedding process on the final image.

Figure 4.20 demonstrates the results of the Entropy technique with a block size of 16x16 applied to the Lina image for the embedding process. A visual examination reveals a remarkable similarity between the original and stego images, indicating successful preservation of essential image properties. However, more detailed histogram analysis reveals minor distortions, including some frequency increases in pixel density and slight variations in pixel intensity values, suggesting some impact on the image due to the data embedding process. Despite these distortions, the overall similarity between the original and stego images highlights the effectiveness of the Entropy technique with a block size of 16x16 in maintaining image quality while hiding the secret message.

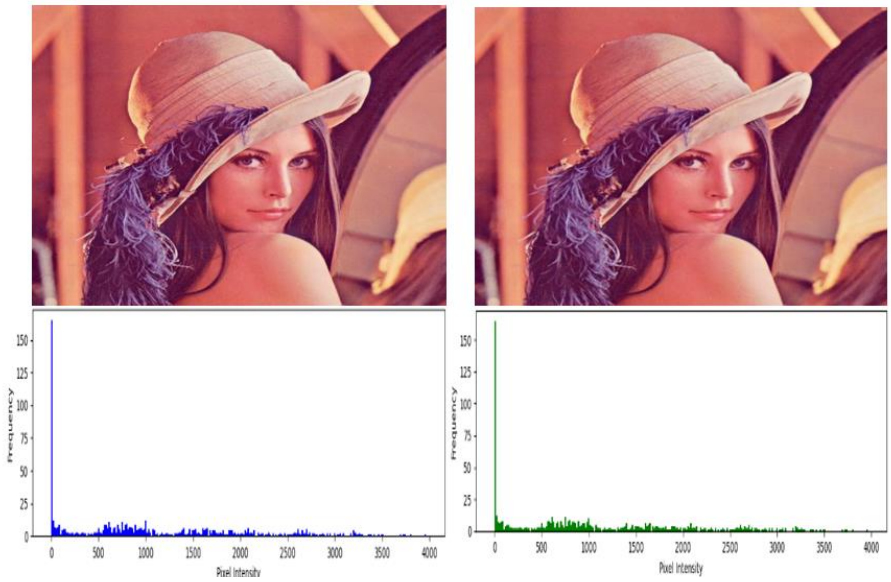


Figure 4.20. Histogram graph of the original and stego of the image named Lena using the Entropy approach with block size of 16x16

In Figure 4.21, the results of applying the Entropy technique with a block size of 32x32 to the Lina image for the data embedding process are displayed. The histogram distribution of pixel density in the image processed with a block size of 32x32 closely resembles the histogram distribution in the image processed with a block size of 16x16. While the overall patterns are almost the same, there are some minor discrepancies between the two distributions.

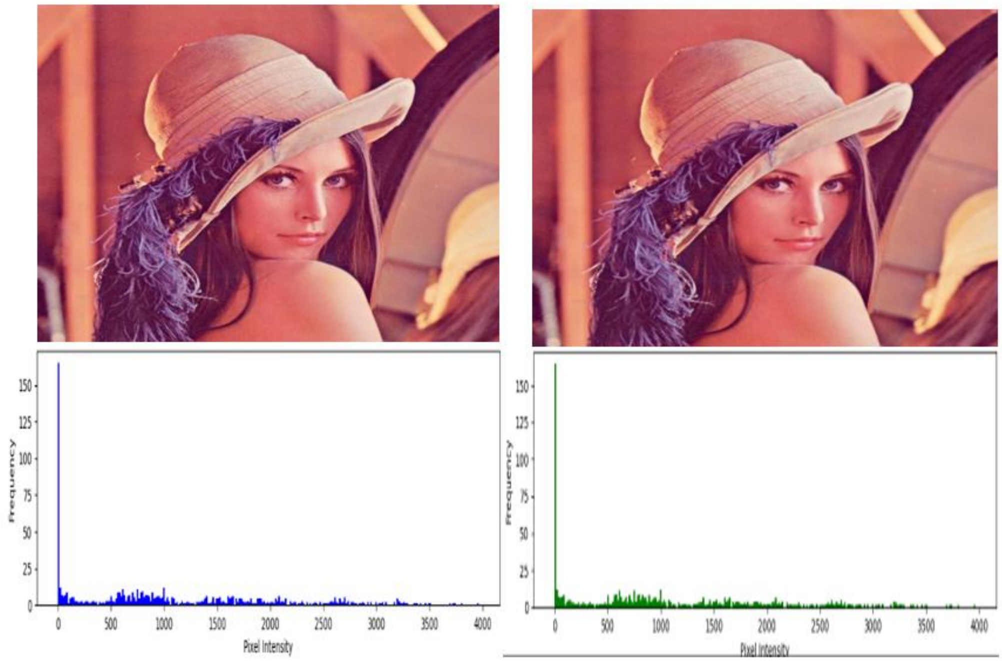


Figure 5.21. Histogram graph of the original and stego of the image named Lena using the Entropy approach with block size of 32x32

Table 4.5 presents the experimental results of using the Entropy method with a block size of 16x16 on color images of different resolutions. The findings show a consistent trade-off between data capacity and image quality, as increasing the text size for data hiding consistently decreased the PSNR values. This is due to increased distortion in the least significant bits of the green channel as more data is embedded. The PSNR values for color images at resolutions 256, 512, and 1024 ranged from 64.88 to 79.16, 70.98 to 85.07, and 76.93 to 90.82, respectively. Higher-resolution images generally achieved higher PSNR values, but the decreasing trend with increasing text size remained consistent across all tested images.

Similarly, the MSE values increased with larger text sizes for all color images. Lower MSE values were obtained for text sizes of 30 bytes. The MSE values for color images at resolutions 256, 512, and 1024 ranged from 0.0007 to 0.0211, 0.0002 to 0.0051, and 0.00005 to 0.0013, respectively. Larger color variations in the cover image correlated with smaller MSE values.

The UQI and MSSIM metrics both took a value of one, indicating a perfect similarity between the original and stego images. The CQM metric demonstrated that lower values for text size 30 indicated reduced distortion and better quality in the stego images. Furthermore, the AD, SC, and NAE metrics exhibited low values, suggesting that the stego images closely resembled the original images with well-preserved structural similarity and content.

Table 4.5. Results of statistical parameters obtained from applying the Entropy approach on color images with different text sizes of block size of 16x16

Image name	Text Size (byte)	PSNR (dB)	MSE	UQI	MSSIM	CQM	AD	SC	NCC	NAE
Image Resolution 256*256										
Lena	30	78.21	0.0009	1	1	0.118	0.001	0	1	0.001
	200	70.67	0.0055	1	1	0.719	0.008	0	1	0.008
	400	67.63	0.0112	1	1	1.446	0.017	0	1	0.017
	800	64.89	0.0210	1	1	2.799	0.032	0	1	0.032
Baboon	30	78.34	0.0009	1	1	0.122	0.001	0	1	0.001
	200	70.89	0.0052	1	1	0.662	0.007	0	1	0.007
	400	67.64	0.0111	1	1	1.436	0.016	0	1	0.016
	800	**64.88**	**0.0211**	**1**	**1**	**2.709**	**0.031**	**0**	**1**	**0.031**
Pepper	30	78.73	0.0008	1	1	0.118	0.001	0	1	0.001
	200	70.93	0.0052	1	1	0.701	0.008	0	1	0.008
	400	67.83	0.0106	1	1	1.413	0.016	0	1	0.016
	800	65.06	0.0202	1	1	2.632	0.030	0	1	0.030
House	30	78.76	0.0008	1	1	0.114	0.001	0	1	0.001
	200	70.83	0.0053	1	1	0.6603	0.007	0	1	0.007
	400	67.86	0.0106	1	1	1.359	0.015	0	1	0.015
	800	65.04	0.0203	1	1	2.690	0.031	0	1	0.031
Plane	30	78.86	0.0008	1	1	0.109	0.001	0	1	0.001
	200	70.89	0.0052	1	1	0.690	0.008	0	1	0.008
	400	67.66	0.0111	1	1	1.447	0.017	0	1	0.017
	800	64.94	0.0208	1	1	2.727	0.032	0	1	0.032
Ariel Iraq1	30	78.66	0.0008	1	1	0.106	0.001	0	1	0.001
	200	70.88	0.0052	1	1	0.665	0.007	0	1	0.0078
	400	67.72	0.010	1	1	1.346	0.015	0	1	0.015
	800	64.97	0.0206	1	1	2.604	0.030	0	1	0.030
Ariel Iraq2	30	**79.16**	**0.0007**	**1**	**1**	**0.106**	**0.001**	**0**	**1**	**0.001**
	200	70.75	0.005	1	1	0.695	0.008	0	1	0.008
	400	67.57	0.0113	1	1	1.472	0.017	0	1	0.017
	800	64.95	0.0207	1	1	2.684	0.031	0	1	0.031
Image Resolution 512*512										
Lena	30	84.63	0.0002	1	1	0.031	0.0003	0	1	0.0003
	200	76.80	0.0013	1	1	0.176	0.002	0	1	0.002
	400	73.67	0.0027	1	1	0.349	0.004	0	1	0.004
	800	71.04	0.0051	1	1	0.655	0.007	0	1	0.007
Baboon	30	84.41	0.0002	1	1	0.026	0.0003	0	1	0.0003
	200	76.87	0.0013	1	1	0.163	0.001	0	1	0.001
	400	73.79	0.0027	1	1	0.336	0.003	0	1	0.003
	800	71.02	0.0051	1	1	0.647	0.007	0	1	0.007
Pepper	30	84.55	0.0002	1	1	0.031	0.0003	0	1	0.0003
	200	76.67	0.0013	1	1	0.189	0.002	0	1	0.002
	400	73.67	0.0027	1	1	0.371	0.004	0	1	0.004
	800	**70.98**	**0.0051**	**1**	**1**	**0.663**	**0.007**	**0**	**1**	**0.007**
House	30	84.32	0.0002	1	1	0.032	0.0003	0	1	0.0003
	200	77.06	0.0012	1	1	0.152	0.001	0	1	0.001
	400	73.79	0.0027	1	1	0.334	0.003	0	1	0.003
	800	71.07	0.0050	1	1	0.631	0.007	0	1	0.007
Plane	30	84.63	0.0002	1	1	0.027	0.0003	0	1	0.0003
	200	75.46	0.0018	1	1	0.184	0.002	0	1	0.002
	400	73.77	0.0027	1	1	0.340	0.004	0	1	0.004

	800	71.08	0.0050	1	1	0.646	0.007	0	1	0.007
Ariel Iraq1	30	**85.07**	**0.0002**	**1**	**1**	**0.023**	**0.0002**	**0**	**1**	**0.0002**
	200	76.94	0.0013	1	1	0.167	0.001	0	1	0.001
	400	73.40	0.0029	1	1	0.375	0.004	0	1	0.004
	800	71.05	0.0051	1	1	0.652	0.007	0	1	0.007
Ariel Iraq2	30	84.58	0.0002	1	1	0.031	0.0003	0	1	0.0003
	200	76.81	0.0013	1	1	0.166	0.001	0	1	0.001
	400	72.31	0.0038	1	1	0.413	0.004	0	1	0.004
	800	71.02	0.0051	1	1	0.636	0.007	0	1	0.007
Image Resolution 1024*1024										
Lena	30	90.55	0.00005	1	1	0.007	0.00008	0	1	0.00008
	200	82.98	0.0003	1	1	0.041	0.0004	0	1	0.0004
	400	79.75	0.0006	1	1	0.089	0.001	0	1	0.001
	800	77.06	0.0012	1	1	0.165	0.001	0	1	0.001
Baboon	30	90.18	0.00006	1	1	0.008	0.00009	0	1	0.00009
	200	82.98	0.0003	1	1	0.040	0.0004	0	1	0.0004
	400	79.70	0.0006	1	1	0.091	0.001	0	1	0.001
	800	**76.93**	**0.0013**	**1**	**1**	**0.170**	**0.002**	**0**	**1**	**0.002**
Pepper	30	90.48	0.00005	1	1	0.007	0.00008	0	1	0.00008
	200	82.91	0.0003	1	1	0.043	0.0005	0	1	0.0005
	400	79.73	0.0006	1	1	0.090	0.001	0	1	0.001
	800	77.13	0.001	1	1	0.161	0.001	0	1	0.001
House	30	90.53	0.00005	1	1	0.007	0.00008	0	1	0.00008
	200	83.13	0.0003	1	1	0.036	0.0004	0	1	0.0004
	400	79.79	0.0006	1	1	0.066	0.0007	0	1	0.0007
	800	77.03	0.0012	1	1	0.163	0.00008	0	1	0.00008
Plane	30	**90.82**	**0.00005**	**1**	**1**	**0.006**	**0.00007**	**0**	**1**	**0.00007**
	200	82.78	0.0003	1	1	0.044	0.0005	0	1	0.0005
	400	79.72	0.0006	1	1	0.084	0.0009	0	1	0.0009
	800	77.09	0.0012	1	1	0.163	0.001	0	1	0.001
Ariel Iraq1	30	90.65	0.00005	1	1	0.007	0.00008	0	1	0.00008
	200	82.97	0.0003	1	1	0.043	0.0005	0	1	0.0005
	400	79.86	0.0006	1	1	0.087	0.001	0	1	0.001
	800	77.02	0.0012	1	1	0.163	0.001	0	1	0.001
Ariel Iraq2	30	90.55	0.00005	1	1	0.006	0.00007	0	1	0.00007
	200	82.91	0.0003	1	1	0.041	0.0004	0	1	0.0004
	400	78.29	0.0009	1	1	0.104	0.001	0	1	0.001
	800	77.03	0.0012	1	1	0.162	0.001	0	1	0.001

Figures 4.22 and 4.23 show the best PSNR and MSE values obtained as a result of the data hiding process at a text size of 30 bytes on seven different colored images with different resolutions using the entropy method. The analysis reveals that the PSNR values exhibit a range of 78.21 dB to 90.82 dB across the different images with different resolutions using block size of 16x16.

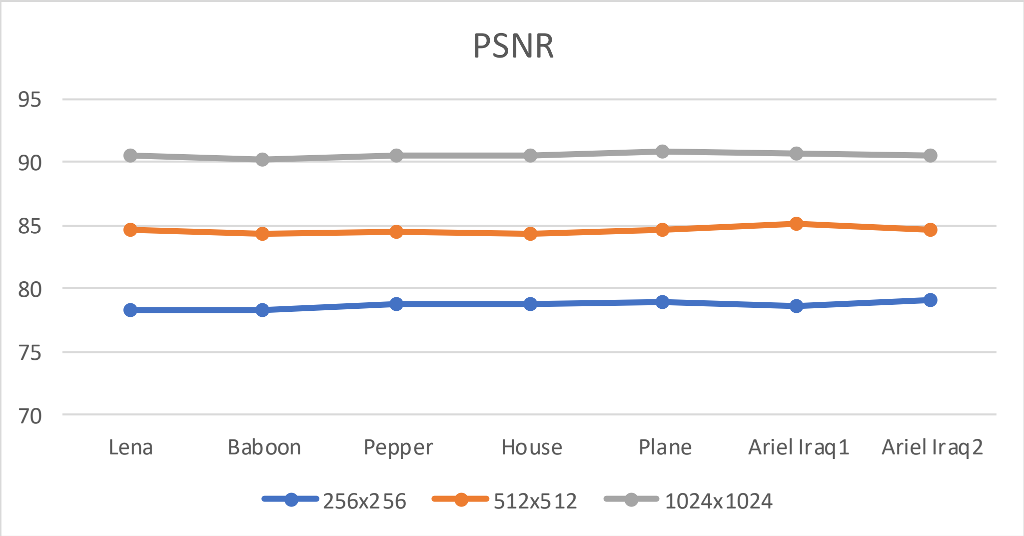


Figure 6.22. Best PSNR achieved using Entropy approach in seven colored images at different resolutions using block size of 16x16

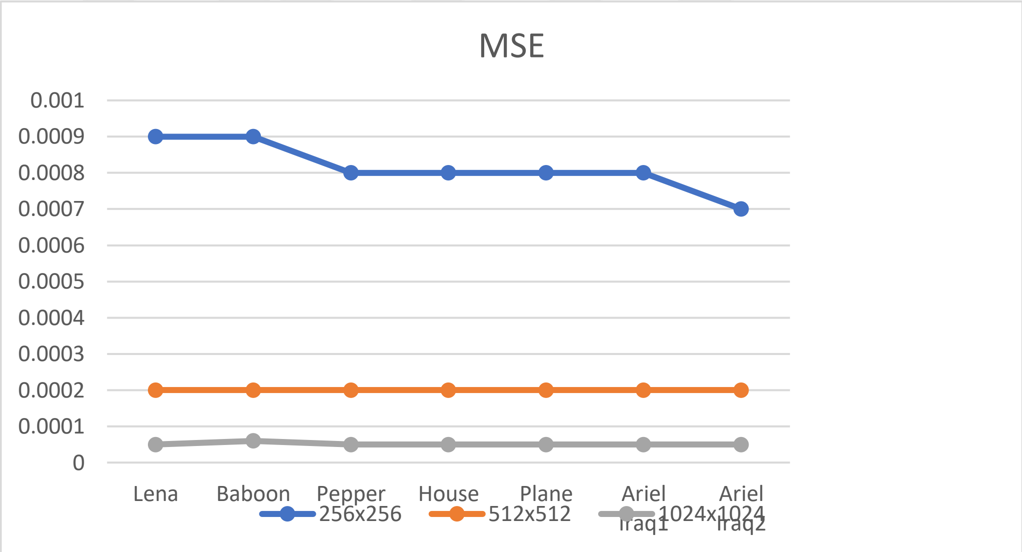


Figure 7.23. Best MSE achieved using Entropy approach in seven colored images at different resolutions using block size of 16x16

Table 4.6 displays the results of utilizing the Entropy method with block sizes of 32x32 on color images of varying resolutions. The findings indicate that the outcomes obtained with a block size of 32x32 are highly similar to those obtained with a block size of 16x16 for all tested color images. This closeness in results suggests that increasing the block size from 16x16 to 32x32 did not have a substantial impact on the performance of the Entropy steganography method for these images. The consistency between the results obtained with both block sizes indicates that the Entropy method is robust and stable, irrespective of the chosen block size.

Table 4.6. Results of statistical parameters obtained from applying the Entropy approach on color images with different text sizes of block size of 32x32

Image name	Text Size (byte)	PSNR (dB)	MSE	UQI	MSSIM	CQM	AD	SC	NCC	NAE
Image resolution 256*256										
Lena	30	78.78	0.0008	1	1	0.126	0.001	0	1	0.001
	200	70.84	0.0053	1	1	0.715	0.008	0	1	0.008
	400	67.83	0.0107	1	1	1.385	0.016	0	1	0.016
	800	65.04	0.0203	1	1	2.639	0.031	0	1	0.031
Baboon	30	78.58	0.0009	1	1	0.117	0.001	0	1	0.001
	200	70.80	0.005	1	1	0.690	0.008	0	1	0.008
	400	67.71	0.0109	1	1	1.377	0.016	0	1	0.016
	800	64.98	0.0206	1	1	2.640	0.031	0	1	0.031
Pepper	30	78.86	0.0008	1	1	0.106	0.001	0	1	0.001
	200	70.83	0.0053	1	1	0.702	0.008	0	1	0.008
	400	67.74	0.0109	1	1	1.376	0.016	0	1	0.016
	800	64.96	0.020	1	1	2.595	0.030	0	1	0.030
House	30	78.58	0.0009	1	1	0.123	0.001	0	1	0.001
	200	70.88	0.0052	1	1	0.643	0.007	0	1	0.007
	400	67.81	0.0107	1	1	1.396	0.0164	0	1	0.0164
	800	64.97	0.0206	1	1	2.738	0.032	0	1	0.032
Plane	30	78.86	0.0008	1	1	0.111	0.001	0	1	0.001
	200	70.74	0.0054	1	1	0.519	0.006	0	1	0.006
	400	67.75	0.0108	1	1	1.183	0.013	0	1	0.013
	800	**64.95**	**0.020**	**1**	**1**	**2.487**	**0.029**	**0**	**1**	**0.029**
Ariel Iraq1	30	78.81	0.0008	1	1	0.109	0.001	0	1	0.001
	200	71.03	0.0051	1	1	0.676	0.007	0	1	0.007
	400	67.74	0.0109	1	1	1.397	0.016	0	1	0.016
	800	64.96	0.0207	1	1	2.639	0.031	0	1	0.031
Ariel Iraq2	30	**78.94**	**0.0008**	**1**	**1**	**0.101**	**0.001**	**0**	**1**	**0.001**
	200	70.85	0.0053	1	1	0.682	0.008	0	1	0.008
	400	67.71	0.011	1	1	1.384	0.016	0	1	0.016
	800	64.98	0.0206	1	1	2.688	0.031	0	1	0.031
Image Resolution 512*512										
Lena	30	84.68	0.0002	1	1	0.025	0.0002	0	1	0.0002
	200	77.03	0.0012	1	1	0.169	0.001	0	1	0.001
	400	73.72	0.0027	1	1	0.356	0.004	0	1	0.004
	800	70.99	0.0051	1	1	0.670	0.007	0	1	0.007
Baboon	30	84.86	0.0002	1	1	0.027	0.0003	0	1	0.0003
	200	76.78	0.0013	1	1	0.181	0.002	0	1	0.002
	400	73.76	0.0027	1	1	0.341	0.004	0	1	0.004
	800	71.02	0.0051	1	1	0.657	0.007	0	1	0.007
Pepper	30	84.70	0.0002	1	1	0.029	0.0003	0	1	0.0003
	200	76.89	0.0013	1	1	0.176	0.002	0	1	0.002
	400	73.65	0.0028	1	1	0.372	0.004	0	1	0.004
	800	70.99	0.0051	1	1	0.679	0.007	0	1	0.007
House	30	84.65	0.0002	1	1	0.028	0.0003	0	1	0.0003
	200	76.84	0.0013	1	1	0.170	0.002	0	1	0.002
	400	73.78	0.0027	1	1	0.350	0.004	0	1	0.004
	800	71.05	0.0051	1	1	0.654	0.007	0	1	0.007
Plane	30	**85.21**	**0.0001**	**1**	**1**	**0.022**	**0.0002**	**0**	**1**	**0.0002**
	200	77.11	0.0012	1	1	0.170	0.002	0	1	0.002
	400	73.72	0.0027	1	1	0.365	0.004	0	1	0.004
	800	71.09	0.0050	1	1	0.656	0.007	0	1	0.007
Ariel Iraq1	30	84.70	0.0002	1	1	0.027	0.0003	0	1	0.0003
	200	76.69	0.0013	1	1	0.169	0.001	0	1	0.001
	400	73.77	0.0027	1	1	0.347	0.004	0	1	0.004
	800	71.03	0.0051	1	1	0.665	0.007	0	1	0.007

Ariel Iraq2	30	84.68	0.0002	1	1	0.028	0.0003	0	1	0.0003
	200	76.81	0.0013	1	1	0.176	0.002	0	1	0.002
	400	73.72	0.0027	1	1	0.362	0.004	0	1	0.004
	800	**70.91**	**0.0052**	**1**	**1**	**0.688**	**0.008**	**0**	**1**	**0.008**
Image Resolution 1024*1024										
Lena	30	90.70	0.00005	1	1	0.007	0.00008	0	1	0.00008
	200	82.98	0.0003	1	1	0.041	0.0004	0	1	0.0004
	400	79.83	0.0006	1	1	0.087	0.001	0	1	0.001
	800	77.03	0.0012	1	1	0.165	0.001	0	1	0.001
Baboon	30	90.55	0.00005	1	1	0.007	8.331	0	1	0.00008
	200	82.94	0.0003	1	1	0.042	0.0004	0	1	0.0004
	400	79.73	0.0006	1	1	0.087	0.001	0	1	0.001
	800	77.03	0.0012	1	1	0.162	0.001	0	1	0.001
Pepper	30	90.93	0.00005	1	1	0.006	0.00007	0	1	0.00007
	200	82.97	0.0003	1	1	0.040	0.0004	0	1	0.0004
	400	79.68	0.0006	1	1	0.089	0.001	0	1	0.001
	800	77.00	0.0012	1	1	0.163	0.001	0	1	0.001
House	30	90.55	0.00005	1	1	0.007	0.00009	0	1	0.00009
	200	82.87	0.0003	1	1	0.044	0.0005	0	1	0.0005
	400	79.73	0.0006	1	1	0.087	0.001	0	1	0.001
	800	77.01	0.0012	1	1	0.166	0.001	0	1	0.001
Plane	30	90.75	0.00005	1	1	0.005	0.00006	0	1	0.00006
	200	82.91	0.0003	1	1	0.043	0.0005	0	1	0.0005
	400	79.75	0.0006	1	1	0.087	0.001	0	1	0.001
	800	77.02	0.0012	1	1	0.166	0.001	0	1	0.001
Ariel Iraq1	30	**91.28**	**0.00004**	**1**	**1**	**0.006**	**0.00007**	**0**	**1**	**0.00007**
	200	83.10	0.0003	1	1	0.039	0.0004	0	1	0.0004
	400	79.80	0.0006	1	1	0.087	0.001	0	1	0.001
	800	77.10	0.0012	1	1	0.164	0.001	0	1	0.001
Ariel Iraq2	30	90.72	0.00005	1	1	0.007	0.00008	0	1	0.00008
	200	82.98	0.0003	1	1	0.041	0.0004	0	1	0.0004
	400	79.72	0.0006	1	1	0.086	0.001	0	1	0.001
	800	**76.95**	**0.0013**	**1**	**1**	**0.161**	**0.001**	**0**	**1**	**0.001**

Figures 4.24 and 4.25 show the best PSNR and MSE values obtained as a result of the data hiding process at a text size of 30 bytes on seven different colored images with different resolutions using the entropy method. The analysis reveals that the PSNR values exhibit a range of 78.58 dB to 91.28 dB across the different images with different resolutions using block size of 32x32.

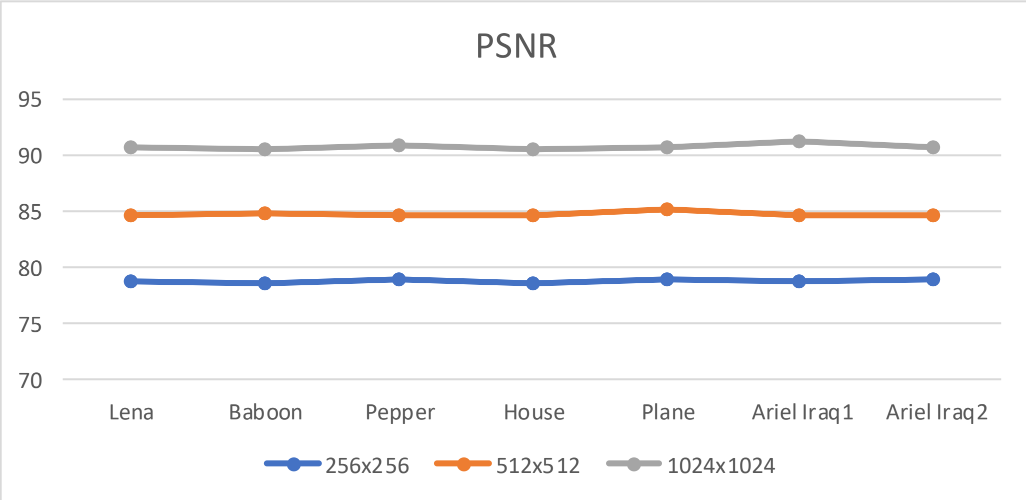


Figure 8.24. Best PSNR achieved using Entropy approach in seven colored images at different resolutions using block size of 32x32

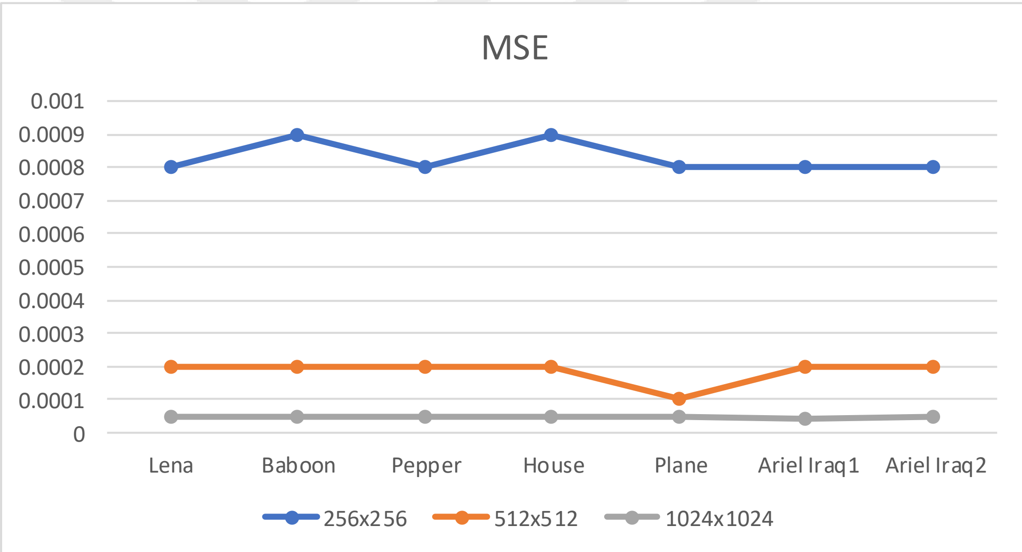


Figure 9.25. Best MSE achieved using Entropy approach in seven colored images at different resolutions using block size of 32x32

4.5. Comparing the Results of our Proposed Techniques with Related Works

The field of image steganography has seen promising results from a variety of published research studies. These experiments have been conducted using a range of methods, with particular focus on LSB method. They have also examined various image resolutions and included various text sizes in their experiments as well. Table 4.7 presents a comparison between the performance results of our proposed approach, and those of other data hiding methods from previous studies.

Only the results of embedding the text size of 30 bytes regarding the proposed approach is presented in Table 4.7 not to extend the Table 4.7 more than necessary. To make the reader to remember the results of the proposed approach for different text sizes for block of 16x16 pixels following paragraphs are added:

The PSNR values for color images at resolutions 256x256 pixels, 512x512 pixels, and 1024x1024 pixels for LSB based block selection approach ranged from 64.92 dB to 78.86 dB, 70.90 dB to 85.07 dB, and 76.99 dB to 91.34 dB, respectively.

The PSNR values for color images at resolutions 256x256 pixels, 512x512 pixels, and 1024x1024 pixels AMDB based block selection approach ranged from 64.903 dB to 78.97 dB, 70.90 dB to 85.01 dB, and 72.56 dB to 91.23 dB, respectively.

The PSNR values for color images at resolutions 256x256 pixels, 512x512 pixels, and 1024x1024 pixels for Entropy based block selection approach ranged from 64.88 dB to 79.16 dB, 70.98 dB to 85.07 dB, and 76.93 dB to 90.82 dB, respectively.

As can be seen from the Table 4.7 and the results presented in upper lines, the proposed approach have compatible results with the previous studies. The dividing the image into the blocks, choosing the most appropriate blocks for data embedding (LSB, AMDB or Entropy), and embedding the secret text into the only green channel of the pixels in the chosen blocks by using odd parity makes the proposed approach more secure and reliable concerning previous studies.

The PSNR ratio is high even the size of the data increases due to selection of appropriate blocks and green channel for data embedding. These two issues increase imperceptibility of the stego images and also allow to embed higher data. Please check study by (Sahil Gangurde et al.,) that resulted in 45.86 dB for 360 bit embedding into Lena image. For the proposed approach, embedding 30 bytes of text into Lena image with resolution of 256x256 resulted around 78 dB for LSB, AMDB and Entropy. For the study of Cataltas and Tutuncu (2017a), PSNR value for embedding 2 kb of text into Lena image with 255x255 resulted in 55.25 dB as highest whereas embedding 30 bytes of text into Lena image with resolution of 256x256 resulted around 78 dB for LSB, AMDB and Entropy by proposed approach. These comparisons can be made with different studies in the literature listed in Table 4.7. It will be seen that proposed approach provides more secure, more flexible, more imperceptibility with higher data embedding capacity.

Table 4.7. PSNR performance results of the proposed approach against other data hiding method

<table>
<tr><th rowspan="2">Approach</th><th rowspan="2">Data size</th><th rowspan="2">Image Name</th><th rowspan="2">Image type</th><th rowspan="2">resolution</th><th colspan="3">PSNR (dB)</th></tr>
<tr><th>LSB</th><th>AMBD</th><th>Entropy</th></tr>
<tr><td rowspan="9">Proposed approach</td><td rowspan="9">30 byte</td><td rowspan="9">Lena
Baboon
Pepper</td><td rowspan="9">RGB</td><td rowspan="3">256x256</td><td>78.71</td><td>78.83</td><td>78.21</td></tr>
<tr><td>78.86</td><td>78.97</td><td>78.34</td></tr>
<tr><td>78.63</td><td>78.83</td><td>78.73</td></tr>
<tr><td rowspan="3">512x512</td><td>84.70</td><td>85.01</td><td>84.63</td></tr>
<tr><td>84.41</td><td>84.63</td><td>84.41</td></tr>
<tr><td>84.55</td><td>84.80</td><td>84.55</td></tr>
<tr><td rowspan="3">1024x1024</td><td>90.38</td><td>90.85</td><td>90.55</td></tr>
<tr><td>90.82</td><td>90.82</td><td>90.18</td></tr>
<tr><td>90.85</td><td>90.67</td><td>90.48</td></tr>
<tr><td rowspan="2">Rajput et al., 2016</td><td rowspan="2">0.177 bpp</td><td>Fabric
Football
Gantrycrane
greens</td><td>RGB</td><td>512x512</td><td colspan="3">77.39
83.98
81.46
82.06</td></tr>
<tr><td>Mountain
Photo101
Turkevich
Wom2</td><td>Gray</td><td>512x512</td><td colspan="3">77.70
82.29
83.54
80.04</td></tr>
<tr><td>Arora et al., 2016</td><td>50000-250000 byte</td><td>RGB image</td><td>RGB</td><td>255x255
270x480
512x512
1200x1920</td><td colspan="3">32.76
37.64
30.19
29.94</td></tr>
<tr><td rowspan="3">Sugathan, 2016</td><td>…</td><td>Lena</td><td rowspan="3">RGB</td><td rowspan="3">360x360</td><td colspan="3">52.17</td></tr>
<tr><td>…</td><td>Pepper</td><td colspan="3">52.17</td></tr>
<tr><td>…</td><td>Baboon</td><td colspan="3">52.16</td></tr>
<tr><td rowspan="4">Islam et al., 2016</td><td>786432 bit</td><td>Lena</td><td rowspan="2">RGB</td><td>512x512</td><td colspan="3">52.34</td></tr>
<tr><td>786432 bit</td><td>Baboon</td><td>512x512</td><td colspan="3">52.68</td></tr>
<tr><td>262144 bit</td><td>Lena</td><td rowspan="2">Gray scale</td><td>512x512</td><td colspan="3">51.17</td></tr>
<tr><td>262144 bit</td><td>Baboon</td><td>512x512</td><td colspan="3">51.18</td></tr>
<tr><td>Chikouche et al., 2017</td><td>3264 bit</td><td>Pepper</td><td>RGB</td><td>512x512</td><td colspan="3">60.61</td></tr>
<tr><td rowspan="2">Cataltas and Kemal, 2017a</td><td>2.71 KB
5.82 KB
18.01KB</td><td rowspan="2">Lena</td><td rowspan="2">RGB</td><td rowspan="2">255x255</td><td colspan="3">55.25
52.03
49.56</td></tr>
<tr><td>2.71 KB
5.82 KB
18.01KB</td><td colspan="3">51.39
48.08
45.71</td></tr>
<tr><td rowspan="3">Cataltas and Tutuncu, 2017b</td><td rowspan="3">56980 bit,
111405 bit,
140364 bit</td><td>Lena</td><td rowspan="3">RGB</td><td>225x255</td><td colspan="3">52.38
49.48
48.47</td></tr>
<tr><td>Baboon</td><td>512x512</td><td colspan="3">59.53
56.61
55.61</td></tr>
<tr><td>Pepper</td><td>600x600</td><td colspan="3">60.91
58.01
57.00</td></tr>
<tr><td rowspan="2">Thahab, 2019</td><td>196608 bits</td><td>Lena
Baboon</td><td rowspan="2">RGB</td><td>256x256</td><td colspan="3">51.13
51.14</td></tr>
<tr><td>786432 bits</td><td>Lena
Baboon</td><td>512x512</td><td colspan="3">51.14
51.14</td></tr>
<tr><td>Rafat, 2020</td><td>2 bpp</td><td>Ariel
Airplane
Gray
Ruler</td><td>Gray scale</td><td>512x512</td><td colspan="3">44.40
43.0812
45.48
49.20</td></tr>
</table>

4.6. Summary

In the summer, the execution environment was described in the begining of this chapter. Following that, the varieties of images and text sizes employed in performing the examined approach were presented. The outcomes of our suggested data concealing approaches (LSB, AMBD, and Entropy) were displayed and explored in depth. Furthermore, the performance of these strategies was compared to earlier study findings. The performance of the suggested approaches was evaluated using nine statistical measures, however only PSNR and MSE discriminated between them. With the suggested methodologies, there were no significant differences in the other investigated parameters. When compared to earlier findings, the quality of our suggested procedures was greater. Furthermore, the results of all three suggested approaches showed significant convergence in the values of PSNR and MSE. This convergence emphasizes the efficiency of each of the three strategies in hiding information inside images, reinforcing their value and dependability even more.

5. CONCLUSION AND FUTURE WORK

5.1. Conclusion

The main advantages of the proposed approach are providing more security, more flexibility, more invisibility, and more data capacity to embed the secret data. In this thesis, we have introduced a new block based image steganography approach for concealing text within images through the selection of blocks, which are selected based on three methods: LSB, AMBD, and Entropy. The embedding and extraction processes were done by utilizing these methods, where the secret text bits are hidden in LSBs of the green channel pixels within chosen blocks. These proposed approach were applied to colored images of varying dimensions and subsequently segmented into blocks sized 16x16 pixels or 32x32 pixels to facilitate the process of selecting blocks and embedding text within blocks. The results indicate that the three proposed methods (LSB, AMBD, and Entropy) to choose the blocks in which the data will be embedded exhibited impressive performance across diverse color image sizes and various text dimensions, as corroborated by an analysis of nine distinct statistical measures (PSNR, MSE, UQI, MSSIM, CQM, AD, SC, NCC, and NAE). Notably, only PSNR and MSE were effective in distinguishing the nuances among the three methods used in proposed approach.

The process of embedding text into images and analyzing the resulting stego images was carried out across three resolutions: 256x256 pixels, 512x512 pixels, and 1024x1024 pixels, utilizing the proposed approach. It was discovered that PSNR values in color images increased as text size decreased. This demonstrates that the similarity between the original image and the stego image decreases as the text size increases. This is because the original image resolution of the manipulated image decreases as more text is embedded, resulting in a lower PSNR value. Moreover, when the green channel color numbers are limited, the PSNR values decrease as text size increases. Notably, PSNR values for various text sizes were higher at an image resolution of 1024x1024 pixels compared to resolutions of 256x256 pixels and 512x512 pixels. This distinction was attributed to the abundance of green channel color pixels within blocks at the higher resolution, which led to the higher PSNR values. Similarly, the MSE values decreased with decreasing text sizes for all color images of various resolutions. This indicates an enhancement in the similarity and fidelity of the stego images to the original image when the text size decreased.

The performances of the three methods used in the proposed approach were further assessed by comparing the results with those achieved using another method, focusing on color images with different text sizes. Our approach demonstrated superior results with higher PSNR and MSE values than the results of previous researches. However, all three methods used in the proposed approach consistently yielded similar results across a range of colored images in varying sizes, subdivided into block sizes of 16x16 pixels or 32x32 pixels. This reliable performance demonstrates the efficiency and effectiveness of these methods in easily concealing information within images. In conclusion, our proposed approach exhibited heightened proficiency in securely concealing private information when compared with the methods used in the literature presented in this study.

5.2. Future work

In terms of future work, our proposed approach for image steganography offers many avenues for improvement and expansion which are described as follows:

1. Enhance system security through advanced encryption and encoding methods for concealed text, bolstering protection against unauthorized access.
2. Improve resilience of embedded text against image modifications, compression, and attacks to ensure system reliability and applicability.
3. Optimize embedding and extraction speed for real-time use, particularly with larger images or videos, enhancing practicality.
4. Extend methods to conceal data in various multimedia formats, such as videos and audio, expanding system versatility while maintaining security and invisibility.
5. Enhance system's global usability by researching cross-language embedding, enabling data concealment in languages beyond English. Develop strategies considering linguistic variations and character sets for improved cultural adaptability while upholding security and effectiveness.

REFERENCES

Al-Saedi, A. K. H., 2016, A method to hide text in image, *Journal of Misan Researches*, 12 (24).

AL-Windawi, Z. T. R. and Al-Jarrah, M., 2017, Security Enhancement of Image Steganography Using Embedded Integrity Features, *Middle East University*.

Arora, A., Singh, M. P., Thakral, P. and Jarwal, N., 2016, Image steganography using enhanced LSB substitution technique, *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 386-389.

Aydoğan, T., 2016, Sayisal Görüntülerde Blok ve Tarama Sirasi Temelli Yeni bir Veri Gizleme Algorïtmasi Tasarimi, *Sakarya Universitesi (Turkey)*.

Bayer, P. and Widenfors, H., 2002, Information Hiding: Steganografic Content in Streaming Media.

Bhardwaj, R. and Sharma, V., 2016, Image steganography based on complemented message and inverted bit LSB substitution, *Procedia Computer Science*, 93, 832-838.

Bhattacharyya, S., 2011, A survey of steganography and steganalysis technique in image, text, audio and video as cover carrier, *Journal of global research in computer science*, 2 (4).

Cataltas, O. and Tutuncu, K., 2017a, Improvement of LSB Based Image Steganography, *International Journal of Electrical, Electronics and Data Communication (IJEEDC), 2017*, pp. 1-5, Volume-5,Issue-9.

Cataltas, O. and Tutuncu, K., 2017b, Implementation and Comparison of Text Compression Algorithms in Image Steganography, *Conference: International Conference on Engineering Technologies (ICENTE'17), Konya, Turkey, December 7-9, 2017*, 944-947.

Chikouche, S. L. and Chikouche, N., 2017, An improved approach for lsb-based image steganography using AES algorithm, *2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B)*, 1-6.

Dougherty, E. R., 1994, Digital Image Processing Methods, *OPTICAL ENGINEERING-NEW YORK-MARCEL DEKKER INCORPORATED-*, 43, 42-42.

Evsutin, O., Melman, A. and Meshcheryakov, R., 2020, Digital steganography and watermarking for digital images: A review of current research directions, *IEEE Access*, 8, 166589-166611.

Gonzales, R. C. and Wintz, P., 1987, Digital image processing, Addison-Wesley Longman Publishing Co., Inc., p.

Heilbronner, R. and Barrett, S., 2013, Image analysis in earth sciences: microstructures and textures of earth materials, Springer Science & Business Media, p.

Ibraheem, N. A., Hasan, M. M., Khan, R. Z. and Mishra, P. K., 2012, Understanding color models: a review, *ARPN Journal of science and technology*, 2 (3), 265-275.

Islam, A. U., Khalid, F., Shah, M., Khan, Z., Mahmood, T., Khan, A., Ali, U. and Naeem, M., 2016, An improved image steganography technique based on MSB using bit differencing, *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, 265-269.

Khudher, I. M., 2021, LSB steganography strengthen footprint biometric template, *Восточно-Европейский журнал передовых технологий*, 1 (9-109), 58-65.

Majeed, M. A., Sulaiman, R., Shukur, Z. and Hasan, M. K., 2021, A review on text steganography techniques, *Mathematics*, 9 (21), 2829.

Manjula, Y. and Shivakumar, K., 2016, Enhanced secure image steganography using double encryption algorithms, *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 705-708.

Matted, S., Shankar, G. and Jain, B. B., 2021, Enhanced image security using stenography and cryptography, *Computer Networks and Inventive Communication Technologies: Proceedings of Third ICCNCT 2020*, 1171-1182.

Mishra, R. and Bhanodiya, P., 2015, A review on steganography and cryptography. Conference Proceeding-2015 International Conference on Advances in Computer Engineering and Applications, ICACEA 2015 (7 2015), 119–122.

Morkel, T., Eloff, J. H. and Olivier, M. S., 2005, An overview of image steganography, *ISSA*, 1-11.

Mousa, S. M. A., 2017, LSBs steganography based on R-Indicator, *M. Sc. Thesis, Faculty of Information Technology, the Islamic University-Gaza*.

Neamah, R. M., Abed, J. A. and Abbood, E. A., 2020, Hide text depending on the three channels of pixels in color images using the modified LSB algorithm, *International Journal of Electrical and Computer Engineering*, 10 (1), 809.

Parmar, C. K. and Pancholi, K., 2015, A Review on Image Compression Techniques, *Journal of Information. Knowledge and Research in Electrical Engineering*, 2 (2), 281-284.

Peterson, R. and Wolffsohn, J., 2005, The effect of digital image resolution and compression on anterior eye imaging, *British journal of ophthalmology*, 89 (7), 828-830.

Rafat, K. F., 2020, Nondeterministic Secure LSB Steganography for Digital Images, *2020 International Conference on Cyber Warfare and Security (ICCWS)*, 1-6.

Rahman, S., Uddin, J., Hussain, H., Jan, S., Khan, I., Shabir, M. and Musa, S., 2023, Multi Perspectives Steganography Algorithm for Color Images on Multiple-Formats, *Sustainability*, 15 (5), 4252.

Rajendran, S. and Doraipandian, M., 2017, Chaotic map based random image steganography using lsb technique, *Int. J. Netw. Secur.*, 19 (4), 593-598.

Rajput, V., Tiwari, S. K. and Gupta, R., 2016, An enhanced image security using improved RSA cryptography and spatial orientation tree compression method, *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 327-331.

Sasmal, M. M. and Mula, M. D., 2021, An enhanced method for information hiding using LSB steganography, *Journal of Physics: Conference Series*, 012015.

Shawkat, S. A., 2007, Enhancing steganography techniques in digital images, *Faculty of Computers and Information, Mansoura University Egypt-2016*.

Stanković, S., Orović, I. and Sejdić, E., 2012, Multimedia signals and systems, Springer Science & Business Media, p. 133-139.

Subramanian, N., Elharrouss, O., Al-Maadeed, S. and Bouridane, A., 2021, Image steganography: A review of the recent advances, *IEEE Access*, 9, 23409-23423.

Sugathan, S., 2016, An improved LSB embedding technique for image steganography, *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, 609-612.

Thahab, A. T., 2019, A secure image steganography based on burrows wheeler transform and dynamic bit embedding, *International Journal of Electrical and Computer Engineering*, 9 (1), 460.

Tiwari, K. and Gangurde, S. J., 2021, LSB steganography using pixel locator sequence with AES, *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*, 302-307.

Tyagi, V., 2018, Understanding digital image processing, CRC Press, p.

Yalman, Y., 2010, Sayısal Görüntüler İçin Histogram Temelli Veri Gizleme Yöntemi Ve Uygulama Yazılımı. Kocaeli Üniversitesi, *Fen Bilimleri Enstitüsü, Elektronik ve Bilgisayar Eğitimi Anabilim Dalı, Doktora Tezi*.

Yiğit, Y. and Karabatak, M., 2019, A stenography application for hiding student information into an image, *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*, 1-4.

Zhou, X., Gong, W., Fu, W. and Jin, L., 2016, An improved method for LSB based color image steganography combined with cryptography, *2016 IEEE/ACIS 15th international conference on computer and information science (ICIS)*, 1-4.

EKLER

EK-1 Algorithm (3.1): LSB based block selection algorithm

INPUT: RGB image blocks
OUTPUT: Image with highest LSB blocks
Begin

For 1 to (number of RGB image Blocks) **do**

Step 1: Convert RGB block to gray scale block.

Step 2: Represent the grayscale block to array matrix of pixel values.

Step 3: Perform a bitwise AND operation between the pixel value and 1 then obtain LSB pixel value.

Step 4: Calculate LSB of a block by summing all LSB pixels' values, then dividing by its length.

End for

Step 5: Sorting LSB values in descending order to obtain the highest LSB blocks (only 128 blocks is selected)

Step 6: Show the image with the highest LSB blocks.

End

EK-2 Algorithm (3.2): AMBD based block selection algorithm

INPUT: RGB image blocks
OUTPUT: Image with the highest AMBD blocks
Begin
For 1 to (number of RGB image Blocks) **do**
Step 1: Convert the RGB block to an array.
Step 2: Calculate the mean brightness values along each color channel (axis= (0, 1)).
Step 3: Calculate the absolute mean of the brightness for the block by summing the mean brightness values and dividing it by its length.
End for
Step 5: Sorting AMBD values in descending order to obtain the highest AMBD blocks (only 128 blocks are selected)
Step 6: Show the image with the highest AMBD blocks.
End

EK-3 Algorithm (3.3): Entropy based block selection algorithm

INPUT: RGB image blocks
OUTPUT: image with the highest Entropy blocks
Begin
For 1 to (number of RGB image Blocks) **do**
Step 1: Convert the RGB block to 1-D array of pixels values
Step 2: Calculate the histogram of the pixel values with 256 bins and range from 0 to 256.
Step 3: Calculate the probabilities by dividing the histogram values by the total number of pixel values.
Step 4: Calculate Entropy by taking the negative sum of the probabilities weighted by their logarithm
End for
Step 5: Sorting Entropy values in descending order to obtain the highest Entropy blocks (only 128 blocks is selected)
Step 6: Show image with the highest Entropy blocks.
End

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	Asmaa Hakeem Naeem AL MAGED
Uyruğu	**:**	IRAK

EĞİTİM

Derece		**Adı, İlçe, İl**	**Bitirme Yılı**
Lise	:	El Yakadah kız lisesi	2014
Üniversite	:	Al Qalam Üniversitesi	2018
Yüksek Lisans	:		
Doktora	:		

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**

UZMANLIK ALANI

YABANCI DİLLER

BELİRTMEK İSTEĞİNİZ DİĞER ÖZELLİKLER

YAYINLAR

Asmaa Hakeem Naeem AL MAGED **"Block Based Image Stenography**" Published in International Journal of Trend in Research and Development (IJTRD), ISSN: 2394-9333, Volume-10 | Issue-4 , August 2023