Enhancing Information Hiding and Segmentation for Medical Images using Novel Steganography and Clustering Fusion Techniques

by

Hayat Shahir Al-Dmour

A dissertation submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy

School of Biomedical Engineering

Faculty of Engineering and Information Technology

University of Technology Sydney

January 2018

Certificate of Original Authorship

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as part of the collaborative doctoral degree and/or fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Hayat Shahir Al-Dmour
Monday 14th August, 2017

Abstract

In recent years, there has been rapid development in digital medical imaging. The continuous development of medical imaging is expected to make further contributions to healthcare systems, where the increased use of medical imaging in a variety of clinical settings has played an important role in improving health services. The main objective of the research presented in this thesis is to investigate digital image steganography and segmentation in order to offer a systematic way for designing and developing them, with a particular concentration on medical imaging security and magnetic resonance (MR) brain image segmentation.

The first objective presents digital steganography, which refers to the science of concealing important information in digital media such as text, image, audio and video. The importance of this science comes from the fact that if the message is visible, then the attack is highly possible. So, the purpose of digital image steganography is to hide the existence of the secret message from a third party that is unauthorized to see it. The second objective presents digital segmentation, which aims to divide the image into meaningful and non-overlapping regions. The segmentation process is considered an essential process in many important biomedical applications, such as tumour detection, quantitative tissue analysis and computer-integrated surgery.

A major requirement for any steganography method is to minimize the changes that are introduced to the cover image by the data embedding process without compromising the embedding capacity. The main aim of this research is to propose techniques that achieve a high level of capacity, imperceptibility and security. In other words, the proposed methods attempt to reduce the degradation of the stego image to the level that makes the introduced changes not noticeable to the Human Visual System (HVS). Since the HVS is less sensitive to changes in sharp regions of images compared to uniform regions, many researchers have attempted to identify edge pixels and embed the secret message in them in order to enhance imperceptibility and increase the embedding capacity by varying the number of embedded bits per pixel based on edges' strength. However, the identification of edges in steganography systems is usually faced with some challenges that are mainly related to changes that are caused by the embedding process, which lead to having slight difference between the edges of the cover (original) image and the stego image (output of the embedding process). In addition to proposing a method that attempts to resolve this issue, we incorporate coding theory to help in reducing modifications caused by the embedding process.

In medical image security systems, information security schemes are used to conceal coded Electronic Patient Records (EPRs) into medical images. This will help to protect the EPRs' confidentiality without affecting the image quality and particularly the Region of Interest (ROI), which is essential for diagnosis. A method that converts EPR data into ciphertext using private symmetric encryption method is proposed. A simple edge detection method has been developed to embed the confidential information in edge pixels, which will lead to an improved stego image quality. To increase the efficiency, two message coding mechanisms have been utilized to enhance the ±1 steganography. The first one, which is based on Hamming code, is simple and fast, while the other which is known as the Syndrome Trellis Code (STC), is more sophisticated as it attempts to find a stego image that is close to the cover image through minimizing the embedding impact. The proposed steganography algorithm embeds the secret data bits into the Region of Non Interest (RONI), where due to its importance; the ROI is preserved from modifications.

In order to enhance the performance of clustering-based medical image segmentation, an efficient fully-automatic brain tissue segmentation algorithm based on a clustering fusion technique is presented. In the training phase of this algorithm, the pixel intensity value is scaled to enhance the contrast of the image. The brain image pixels that have similar intensity values are then grouped into objects using a superpixel algorithm. Then, three clustering techniques are utilized to segment each object. For each clustering technique, a neural network (NN) model is fed with features extracted from the image objects and is trained using the labels produced by that clustering technique. In the testing phase, a pre-processing step that includes scaling and resizing of the brain image is applied before the superpixel algorithm partitions the image into multiple objects (similar to the training phase). The three trained neural network models are then used to predict the respective class of each object and the obtained classes are combined using majority voting.

The performance of all proposed methods have been tested and evaluated on different datasets using different criteria such embedding rate, mean square error (MSE), peak signal-to-noise ratio (PSNR), weighted peak signal-to-noise ratio (wPSNR), embedding efficiency, jaccard similarity (JS), dice similarity coefficient (DSC), root mean square error (RMSE), accuracy, sensitivity and specificity. Also, the effectiveness of the proposed steganography algorithm is proven using one of the efficient steganalysis techniques. The obtained results showed that our proposed methods outperform some of the well-established methods in the literature.

Acknowledgments

First and foremost, I would like to express my gratitude to Allah (Glorified and Exalted is He) for blessing me with endurance and fortitude to smoothly accomplish this dissertation.

During this research period of four years, there have been many people who have walked alongside me with their guidance, support and inspiration. I am much grateful to my principle supervisor **Dr. Ahmed Al-Ani** for incessant guidance and encouragement. Discussions with him helped me to understand my potential besides producing this high-quality dissertation. Without him, the road towards the completion of this research would have been difficult.

Thanks would be a simple word to applaud the sacrifice of my parents, especially my deceased father. He has been a pillar of support to shape my values, perseverance and in large as a good human. Extended thanks to my sisters and brothers who also supported me in each step of these four years.

I express my profound gratitude to all my colleagues and friends within and outside the premises of UTS, in particular **Obaid Aamir** and **Karthick Thiyagarajan**, for their friendship and support throughout my degree.

Lastly but not the least, I endorse the Mutah University for their commitment to support my Ph.D. candidature with scholarships. This enabled me to realise my long lasting dream into reality.

Dedication

*Every challenging work needs self-efforts as well as guidance and support of parents. This thesis is dedicated to the memory of my father, **Shahir Al-Dmour**, who passed away before I completed my degree. I wish that he could be with me to share the success of my graduation with a Doctor of Philosophy degree.*

*This thesis is also dedicated to my mother, **Khadeejeh Al-Dmour**. This dissertation stands as a testimony for her endless support, prayers, love and beyond to overcome my hardships to complete my degree.*

To my beloved sisters and brothers, for their support and patience throughout these stressful years.

Abbreviations

BPNN	Back Propagation Neural Network
CSF	Cerbuspinal Fluid
DCT	Discrete Cosine Transform
DHHS	Department of Health and Human Services
DICOM	Digital Imaging and Communication In Medicine
DSC	Dice Similarity Coefficient
DWT	Discrete Wavelet Transform
ECC	Error Correction Code
EPR	Electronic Patient Record
FCM	Fuzzy C-means
GM	Gray Matter
HAS	Human Auditory System
HIPAA	Health Insurance Portability and Accountability Act
HVS	Human Visual System
ID	Identity Card
IDCT	Inverse Discrete Cosine Transform
ISP	Internet Service Provider
IWT	Integer Wavelet Transform

JPEG	Joint photographic expert group
JS	Jaccard Similarity
KLD	Kullback–Leibler Divergence
LSB	Least Significant Bit
LSBM	Least Significant Bit Matching
MIS	Medical Information System
MSB	Most Significant Bit
MSE	Mean Square Error
MRI	Magnetic Resonance Image
PACS	Picture Archiving and Communication System
PMM	Pixel Mapping Method
PoV	Pair of Values
PRNG	Pseudo Random Number Generator
PSNR	Peak Signal-to-Noise Ratio
PVD	Pixel Value Difference
RLC	Run Length Coding
RMSE	Root Mean Square Error
ROI	Region of Interest
RONI	Region of Non-Interest
SOM	Self-organized Map
SSIM	Structural Similarity Index
STC	Syndrome Trellis Code
TBPC	Tree-based Parity Check
TPVD	Tri-pixel Value Differencing
VoIP	Voice over Internet Protocol
WM	White Matter
wPSNR	weighted Peak Signal-to-Noise Ratio

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Introduction

Digital medical imaging provides solutions in areas, such as radiography, orthopaedics, and oncology among others in improving patient outcomes. These services prove that imaging technology has gone through a complete revolution, allowing medical professionals to be in a position to improve patient outcomes, which explains why for most economies digital imaging is considered a fundamental medical development based on its immense benefits. Digital medical imaging has provided the health sector with the opportunity to access the latest imaging technologies and services deserved by patients. It has been instrumental in empowering patients and physicians regarding accessibility to important information on human health [2].

In recent years, there has been an explosion in the use and development of digital medical images. Health care systems are making significant use of image processing to improve their services. Image steganography and segmentation are some of the basic concepts that have played a critical role in improving digital medical systems.

Steganography aims to provide invisible communication by concealing the confidential data into other forms of digital media so the information does not attract unauthorized user attention [3]. Image segmentation is considered a critical process in medical image analysis and clinical applications, where it aims at dividing the image into non-overlapping regions [4].

Initially, the internet was used as a paramount academic and military resource. However, Internet usage has grown to become a primary tool utilized by commercial organizations, non-commercial organizations and individuals. Therefore, it is necessary for organizations and individuals to preserve the security, confidentiality, and integrity of data, especially during the transmission process [5]. Cryptography is one of the earliest methods of protecting security and privacy, where it transforms original text into an unreadable form [6].

Image steganography is considered an integral tool used in different applications, such as military communications and medical systems to pass sensitive information over public networks where anyone can access and connect to the Internet. In the digital medical system, distribution and maintenance of medical records is a crucial process. For example, during an orthopaedic surgery, it may be required to send the patient's details and medical images to a pathologist and/or radiologist. This information must be preserved for protecting the privacy of the patient's details. In fact, it is against the law to disclose patient information without the patient's authority as presented under the US Department of Health and Human Services (DHHS) and the Health Insurance Portability and Accountability Act (HIPAA) of 1996 [7].

There are some essential requirements of a successful steganography technique such as imperceptibility, capacity, robustness, and security, which are conflicting. So, one of the most significant challenges that face steganography techniques is embedding a large amount of confidential data without distorting the original image, whereas the capacity and imperceptibility are contradicting requirements of the steganography scheme.

Image segmentation is considered a critical process in medical image analysis and clinical applications used for measuring and visualizing the anatomical structures of the brain, analyzing changes in the brain, surgical planning, image-guided interventions, and delineating the pathological regions.

Brain image segmentation has been instrumental in improving the human understanding of brain anatomy in clinical applications, mainly because of its ability to influence the outcome of brain analysis [8]. The objective of segmentation is to simplify the actual illustration of an image into another format, making it easier to understand and analyze [9].

This chapter is organized as follows. Motivation and research problems are discussed in Section 1.2. Section 1.3 presents the limitations of the existing methods. The research questions and objectives are presented in Sections 1.4 and 1.5 respectively. Thesis organization is summarized in Section 1.6.

1.2 Motivation and Research Problems

In recent years, information security has been proven to be an essential component in the digital era, where the use of digital media such as text, image, audio, and video has been rapidly increasing. Since digital data can be hacked, and any unauthorized intruders can have access to this information during the transmission over the Internet, the need of a secured network is demanding.

Cryptographic methods were introduced to provide integrity, privacy, and confidentiality of the secret data. Cryptography aims to provide confidential communication by encrypting the message using a key. Many real world applications have accepted encryption methods to protect their data. While from a different point of view, the transmission of encrypted text attracts intruders. In addition to the growth of government restrictions and constraints on using encryption methods, this unwanted attention stimulates the researcher to investigate an alternative approach that enables secure communication. Therefore, steganography may be the most secure mode of communication between members of different groups. Digital steganography, particularly digital image steganography, has attracted the research community to protect confidential data during the transmission of images or other media files.

There has been an increased interest in the transmission and exchange of digital medical images between hospitals and clinics over public networks. The Picture Archiving and Communication System (PACS) has been designed to store and transmit digitized medical images for e-health services; however, existing implementations of this system do not pay much attention to the confidentiality and protection of patients' information. Digital steganography has attractive characteristics to offer secure communication for medical system applications.

Despite a large number of publications in the area of steganography, it is still lacking in finding a comprehensive steganography scheme that can achieve good balance between the different requirements of digital steganography. To be more precise, the capacity and imperceptibility are the most significant requirements of the image steganographic system. While increasing the embedding capacity causes noticeable artefacts in the resulting image of the embedding process.

There are a limited number of techniques for medical image steganography, however, most of them use classical steganography techniques, such as Least Significant Bit (LSB), without taking into consideration the appropriateness of the embedding process for hiding the confidential data. These medical image steganography methods define the diagnosis region, known as the Region of Interest (ROI), and protect it from any alternation during the embedding process.

The human brain is one of the most sensitive body organs. Therefore, it is important that the world invests in initiatives aimed at improving the study of brain anatomy and function in a bid to make progress in providing quality care services and treating brain diseases, such as tumours [4]. The brain tumour is considered one of the most dangerous diseases affecting human beings [10].

Advanced imaging technologies, such as CT and MRI have played a paramount role in improving different medical procedures. To be more precise, medical imaging technologies offer powerful methods of examining the internal structures of the human body. Therefore, physicians are able to get multi-dimensional images that facilitate their ability to perform comprehensive analysis and make judgements for diagnosis and treatments.

Image segmentation is an essential process in medical image analysis. Medical image segmentation is difficult because of the pixel intensity inhomogeneity or bias field. Various brain image segmentation methods have been introduced by various researchers to assist physicians and neurosurgeons in the identification and differentiation of normal and diseased tissues. The human brain is comprised of three different tissues: Gray Matter (GM), White Matter (WM) and Cerebrospinal Fluid (CSF). Other tissues, such as a tumour, can be imaged using Magnetic Resonance Imaging (MRI). In reality, the pixel intensities are inhomogeneous and overlap significantly. The absence of distinctly defined edges between neighboring tissues degrades the accuracy of the segmentation process. The reasons mentioned above have motivated the need for developing automatic segmentation methods that are applicable to MR brain images.

1.3 Existing Method Limitations

There are a huge number of steganography and segmentation methods, but most of them suffer from the following drawbacks:

1. Spatial domain steganography techniques provide a large embedding rate compared to transform domain techniques. However, a large embedding capacity affects the visual quality negatively.

2. Transform domain steganography methods provide robustness against attacks, but compromises on both visual quality and embedding capacity.

3. A limited number of steganography publications address the combination of steganography and cryptography, and the effect of embedded encrypted data.

4. Some of the existing segmentation methods require manual/user interaction to initialize some input parameters.

5. The segmentation result depends on the parameters initialization, for example output of the snake model depends on the initial estimation of the curve. Also, the clustering algorithm is dependent upon the initial estimation of the centre values and number of classes.

6. The computational cost and memory are high for some existing methods, such as the hybrid methods.

1.4 Research Question

The following research questions are addressed in this thesis.

- **What is the appropriate steganography model to achieve the best balance between imperceptibility, embedding capacity, and security requirements?**

  This question involves an examination to determine the proper design of a steganographic scheme that can achieve the best image quality without compromising on the embedding capacity or security requirements. The following sub-questions derive for designing the appropriate scheme:

  (a) *Can edge detection algorithms be utilized to achieve a high image quality and embedding capacity?*

  (b) *What is the appropriate coding theory algorithm of the embedding process to improve imperceptibility?*

- **How can patients' confidential information be protected and transmitted securely between different clinics and hospitals?**

  Protecting the confidentiality of patient information during transmission is a challenging problem. Most of the proposed techniques are based on the encryption algorithms. However, the transmission of encrypted data encourages intruders to decrypt it. This research will study the ability to develop a secure medical imaging information system based on the integration of steganography and cryptography techniques.

- **What is the appropriate image segmentation model used for achieving an accurate segmentation result in regards to medical imaging?**

This question addresses different segmentation techniques which can provide accurate results with low computational costs. The following sub-questions derive for designing the appropriate scheme:

(a) *How do we optimally integrate different clustering techniques for brain image segmentation?*

(b) *Can the developed segmentation method handle imbalanced data and overlapping regions?*

1.5 Research Objectives

The aim of this research is to design an image steganographic system, specifically for medical images, in the spatial and transform domains with comprehensible embedding capacity and minimum degradation in the quality of resultant image of the embedding process, which is also known as stego image. To achieve this objective, we develop a new and simple edge detection method that is capable of estimating the exact edge intensities for both the cover and stego images (before and after embedding the secret data). Furthermore, the computation cost of the developed method should be better than other existing techniques. In regards to the security issue, the proposed method obeys an important statistical constraint, where the embedding process does not leave a significant statistical modification on the cover pixels.

The other objective of this research is to develop an efficient fully-automated segmentation method for MR brain images that overcome the overlapping and imbalance issues between brain tissues. To achieve this aim, we introduce a fully-automatic segmentation method for MR images based on the concept of clustering fusion and Neural Networks (NN).

To summarise, our main objectives cover the following:

(1) To develop a new edge detection method.

(2) To improve the embedding efficiency by introducing a new XOR coding operation and modifying the embedding matrix.

(3) To develop an image steganography for health care systems to enhance the confidentiality of information during the exchange process between clinics.

(4) To reduce computational cost for embedding and extraction stages using coding process and block-based edge detection algorithm.

(5) To introduce a complete automated method for the brain tissue segmentation based on clustering techniques.

(6) To create a novel combination between clustering techniques and Neural Network.

(7) To reduce the computational cost of the segmentation process by training neural networks that attempt to imitate the operation of a number of original clustering methods.

1.6 Organization of the Thesis

The thesis is organized as follows:

Chapter 1: This chapter presents an introduction to the research problem. It provides the main motivations and research aims for studying medical image steganography and segmentation.

Chapter 2: This chapter introduces an overview of information security types, in particular steganography. The main differences between steganography, cryptography and watermarking are explained. Digital steganography and its main components are also defined. Next, digital steganography's classification and the requirements of the steganographic system are described. The main evaluation measurements of the steganography method performance are also explained. Several applications employing digital steganography are presented. Finally, the main approaches of steganalysis are explained.

Chapter 3: This chapter contains an extensive literature review of digital steganography methods in the spatial and transform domains. It also discusses state-of-the-art steganography methods and presents existing medical steganography methods.

Chapter 4: This chapter comprises a literature survey of the basic segmentation techniques. It also reviews state-of-the-art segmentation methods for MR brain images.

Chapter 5: This chapter presents a novel image steganography algorithm that combines the strengths of edge detection and XOR coding, to conceal a secret message either in the spatial domain or an Integer Wavelet Transform (IWT) domain of the cover image.

Chapter 6: This chapter introduces a secure medical imaging information system based on steganography and cryptography techniques. This method embeds the encrypted patient's information into region of non interest in order to preserve the diagnosis from any modification.

Chapter 7: This chapter presents a fully-automatic brain tissue segmentation algorithm based on a clustering fusion technique. The proposed method combines the simple linear iterative clustering (SLIC) superpixel, three clustering techniques, and neural network to divide the MR brain image into three tissues of WM, GM and CSF. The method comprises of training and testing stages then it evaluates the accuracy of the proposed method.

Chapter 8: This chapter presents the results of the steganography and segmentation methodologies mentioned in chapters 5, 6 and 7. Several experiments have been carried out to evaluate the performance of the proposed method, and to compare its performance with some of the existing algorithms. A complete description of the employed datasets is given. Afterwards, the results of the steganography methodologies using general and medical datasets for evaluation are presented. Finally, quantitative assessment of the segmentation methodology is carried out different computing metrics.

Chapter 9: This chapter concludes the thesis with a summary of the original contributions and future work.

1.7 Publications

The following papers have been published as a direct result of the research presented in this thesis:

Journals:

1. **H. Al-Dmour** and A. Al-Ani. A clustering fusion technique for MR brain tissue segmentation. In *Neurocomputing*, vol. 275, pages 546–559, 2018.
2. **H. Al-Dmour** and A. Al-Ani. A steganography embedding method based on edge identification and XOR coding. In *Expert systems with Applications*, vol. 46, pages 293–306, 2016.
3. **H. Al-Dmour** and A. Al-Ani. Quality optimized medical image information hiding algorithm that employs edge detection and data coding. In *Computer methods and programs in biomedicine*, vol. 127, pages 24–43, 2016.

Reviewed Conferences:

1. **H. Al-Dmour** and A. Al-Ani. MR brain tissue segmentation based on clustering techniques and neural network. In *International Conference on Image Analysis and Processing (ICIAP '17)*, Springer (2017), pages 225–233, 2017.
2. **H. Al-Dmour** and A. Al-Ani. MR brain image segmentation based on unsupervised and semi-supervised fuzzy clustering methods. In *Digital Image Computing: Techniques and Applications (DICTA '16)*, pages 1–7, 2016.
3. **H. Al-Dmour** and A. Al-Ani. A medical image steganography method based on integer wavelet transform and overlapping edge detection. In *International Conference on Neural Information Processing (ICONIP '15)*, pages 436–444, 2015.
4. **H. Al-Dmour** and A. Al-Ani. Quality optimized medical image steganography based on edge detection and hamming code. In *IEEE 12th International Symposium on Biomedical Imaging (ISBI' 15)*, pages 1486–1489, 2015.

5. **H. Al-Dmour**, N. Ali and A. Al-Ani. An efficient hybrid steganography method based on edge adaptive and tree based parity check. In *International Conference on Multimedia Modeling (MMM '15)*, pages 1–12, 2015.

6. **H. Al-Dmour**, A. Al-Ani and H. Nguyen. An efficient steganography method for hiding patient confidential information. In *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC '14)*, pages 222–225, 2014.

CHAPTER 2

Background and Concept

This chapter presents an overview of the different types of information security and, in particular, steganography. Firstly, the main differences between steganography, cryptography and watermarking are explained. Then, digital steganography and its main components are defined. Next, digital steganography's classification and the requirements of the steganographic system are described. The main evaluation measurements of the steganography method performance are also explained. Several applications employing digital steganography are presented. Finally, the main approaches of steganalysis are explained.

2.1 Introduction

Over the last few decades, the Internet has evolved from an academic and military resource to a public world-wide computer network utilized by numerous commercial and non-commercial organizations and individuals [5]. For example, many institutions such as governments, hospitals and private businesses accumulate an incredible amount of confidential digital data about their employees, customers, patients and products. This data is then often transmitted over insecure and public networks [11]. The evolution in communication has been accompanied by easy-to-access data. For this reason, people are trying to find different ways to maintain security, confidentiality and integrity during data transmission [5]. Additionally, the protection of confidential information is an ethical and legal requirement for many institutions or individuals [11].

Information security is the process of protecting data access, use, destruction, detection, interruption or disruption by intruders [12]. The expressions ’information security’, ’computer security’, and ’information assurance’ are often used interchangeably. These fields are interconnected and have the common objectives of protecting the privacy, integrity and availability of information. However, there are some variances between them in terms of the methodologies used and their areas of interest [13]. For instance, data confidentiality, integrity and availability represent the areas related to information security [13, 14]. On the other hand, computer security does not pay attention to the data processed by the computers, instead concentrating on the correct operation of a computer system and preventing denial-of-service [13].

This chapter is organized as follows. Types of information security are discussed in Section 2.1. Section 2.3 presents history of the steganography. Introduction to digital steganography, steganography method classifications and requirements are introduced in Sections 2.4, 2.5 and 2.6 respectively. Section 2.7 discusses Steganography evaluation Criteria. Sections 2.8, 2.9 and 2.10 present steganography protocols, digital steganography applications and steganalysis respectively. Finally, the summary is given in Section 2.11.

2.2 Types of Information Security

Information security systems can be classified into two main categories: cryptography and information-hiding, as shown in Figure 2.1 [15]. Steganography and watermarking are the two important sub-disciplines of information-hiding [6, 14, 16–18]. A comparison summary between cryptography, steganography and watermarking is presented in Table 2.1.

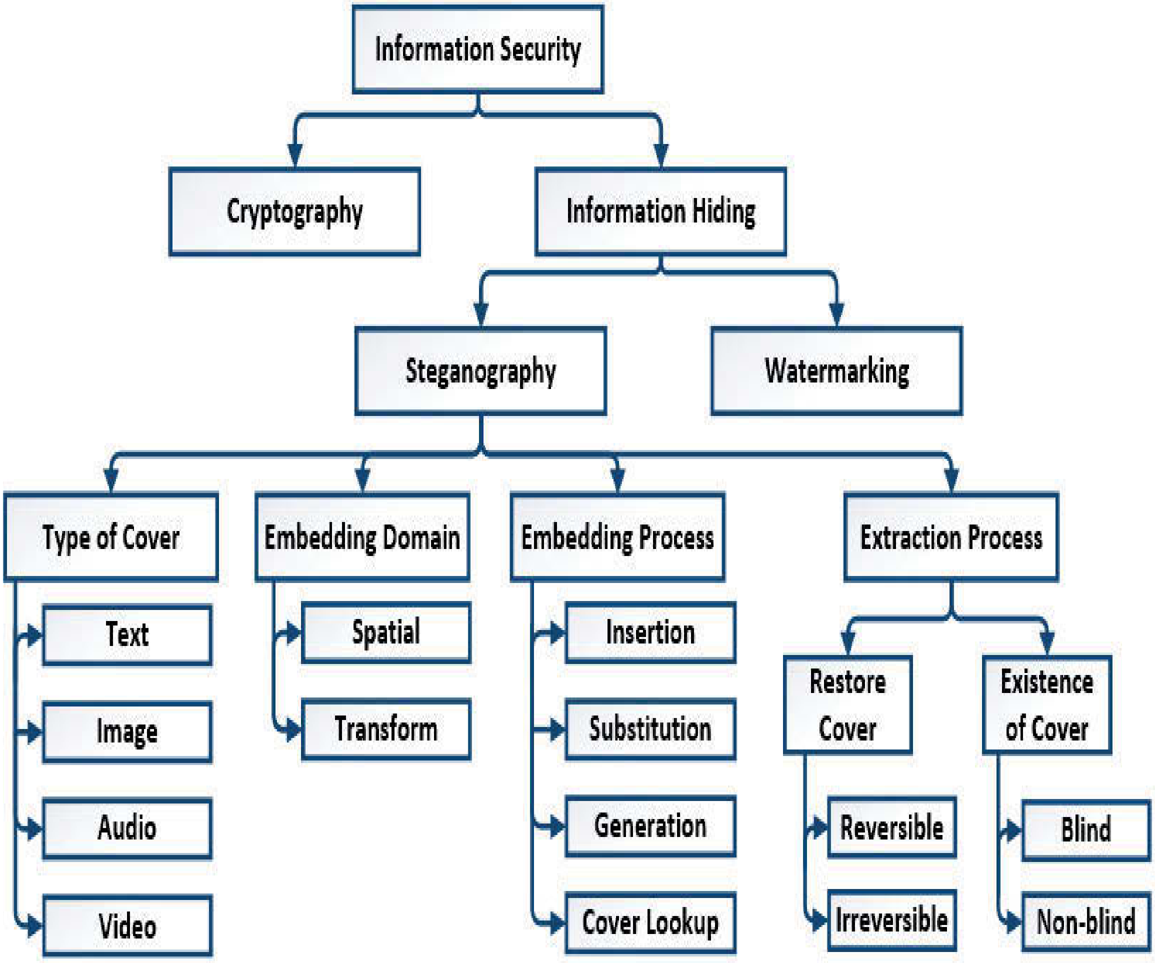


Figure 2.1 Information security system classifications

2.2.1 Steganography and Cryptography

Cryptography and steganography are both methods to protect data from unauthorized users. In cryptography, the secret message is transformed from one form to another in order to make the encrypted data meaningless to intruders. In other words, it encrypts the message to hide its meaning but not its existence. Steganography, in contrast, hides and conceals the secret message within another cover medium to hide all evidence

Table 2.1 Comparison between information security types

Criteria	**Cryptography**	**Steganography**	**Watermarking**
Objective	Data protection	Secure communication	Authentication Copyright
Carrier medium	Text	Any media (text, image, audio and video)	Any media (text, image, audio and video)
Input data	Plaintext	Cover carrier and secret message	Cover carrier and watermark
Output data	Ciphertext	Stego carrier	Watermarked carrier
Key	Required	Optional	optional
Visibility	Visible (hide the meaning but not existence of secret data)	Invisible (hide the existence of the secret data)	Invisible or visible
Extraction type	Blind	Blind	Blind, non-blind and semi-blind
Type of attack	Cryptanalysis	Steganalysis	Image-processing operation
Reversible	Reversible	Reversible or irreversible	Reversible or irreversible
Method requirements	Robustness	Imperceptibility and capacity	Robustness
Broken	If ciphertext is de-ciphered	If secret message is detected	If watermark is removed

of the existence of a secret message during communication. Steganography is utilized to maintain private communication between two parties [6, 19, 20].

Although both cryptography and steganography aim to provide secure communication, they have two different declarations regarding method breaking. If the intruder can decrypt and read the secret message, then a cryptographic scheme is considered cracked. On the other hand, if the intruder detects the existence of a secret message, then a steganographic scheme is considered cracked [6, 21, 22].

2.2.2 Steganography and Watermarking

In steganography, information must never be apparent to a viewer unaware of its presence and modern steganography should be detectable only if the secret key is known. Watermarking, however, can be visible and not necessarily hidden because it is not intended to keep information hidden but rather authenticate the origin of the object tagged. Even though both steganography and watermarking are information-hiding techniques, they have different purposes. Steganography is concerned with hiding the existence of communication by hiding the secret data into another cover carrier, whereas watermarking is concerned with copyright protection and content authentication [14, 23]. Watermarking is used to protect digital contents against the removal of copyright data. Regardless of the fact that somebody realizes that a watermark (i.e., noticeable watermarking) exists in a given article, it must be difficult to remove the copyright information from the cover carrier without creating a degradation in the watermarked carrier [21, 22, 24, 25].

2.3 History of Steganography

Steganography comes from Greek words "steganos" (covered or secret) and "graphy" (writing or drawing). It describes the ancient art of covering messages in a secret way such that only the receiver knows the existence of messages. Steganography can be classified into linguistic steganography and technical steganography [21–23].

The Histories of Herodotus is one of the primary documents that describes the history of steganography. The history of steganography can be traced to 400 BC when Herodotus describes two examples in ancient Greece. In ancient Greece, when someone wanted to transmit a private message they wrote text on a tablet and then it covered it with beeswax. Herodotus tells how Demeratus sent a notification to Sparta about an attack from Xerxes. He wrote the secret message on a wooden tablet and then concealed it with wax. Upon inspection by Xerxes' soldiers, the tablets appeared blank and were allowed to pass [26].

Another example is Histiaeus. He shaved the head of his slave and wrote a message on it. After that, he waited for the slave's hair to grow back, therefore concealing the message and allowing the messenger to transfer their message without impediment [22, 23].

During World War II, it was necessary to invent ways of sending secure messages. The French Resistance sent secret messages written on the backs of messengers using invisible ink. The Germans added letters to the transmitted messages where only certain letters of a transmission formed the real message [27].

Steganography has been far less researched by industry and academics than cryptography. This has changed over recent years. In 1996 the first academic conference on the subject was organized. This was followed by several other conferences focusing on information-hiding as well as watermarking [28, 29].

2.4 Digital Steganography

Digital steganography is the process of embedding data in another cover medium to provide insensible communication. The cover medium may be text, image, audio or video. The output obtained after hiding data in the cover is called stego and the stego medium is transmitted to a receiver. To provide more security, steganography algorithms use cryptography to encrypt the message then embed it inside the cover [14, 18].

In [30], the principle of a steganography framework is explained. The sender (Alice) sends a message (m) to the receiver (Bob) using a random cover medium (c). Alice has an option to embed the message into the cover medium using a stego key (k). The resultant medium, also known as the stego medium (S), should not be differentiated from the original medium (cover) to inhibit the hackers (Wendy) from retrieving the secret message. The stego medium is transmitted to the recipient (Bob) over an insecure channel. The receiver then extracts the secret message since he is aware of the embedding process used by the sender and has the stego key.

The aim of a stego key is to create a secure steganography system. To be more precise, it is possible that Wendy can observe the embedded message in the stego medium and identify the embedding process, but the attacker is unable to retrieve the embedded message without having any knowledge about the stego key. Accordingly, the stego key must be as solid as possible to prevent attackers from breaking the steganography system using all possible stego keys [30].

Steganography system security must fulfil Kerckhoff's principle, which states that in all systems it is assumed that the attacker knows the design of the system, the language used and the algorithm in the system for protection. Therefore, the security of a steganography system should depend on the stego key to ensure that unauthorized users cannot retrieve the secret message without the stego key. When the stego key of the embedding process is similar to the one used in the extraction process, it is referred to as a symmetric key. If they are not similar, it is considered to be asymmetric [30].

Figure 2.2 shows a graphical representation of steganography. A typical steganography system contains two main steps, one for embedding and one for extraction. The embedding algorithm is concerned with inserting the message within a carrier medium such as image, audio or video, where the extraction process retrieves the embedded message from the cover. The extraction algorithm is easier than the embedding algorithm. One of the ways to enhance steganography security is to use the stego key which is required to start the embedding or extraction process. It is utilized to make the extraction process computationally infeasible for unauthorized users [14, 18]. The steganography terminology is listed below:

- Cover object (C): The cover object represents the carrier medium used to hide the secret message (m). Various types of object with redundancy in their representation can be utilized as a cover object, such as text, image, audio and video. It should be undistinguishable from the cover object.
- Stego object (S): The stego object refers to the modified cover object after concealing the secret message. The cover and stego images should have a high degree of similarity to avoid a third party suspecting the existence of the secret message.
- Message (M): This refers to the data that needs to be hidden within the cover object without raising suspicion. Secret data can be any digital data represented in a binary form.

- Key (K): The stego key is an optional component used to control the embedding process. The extraction process is hardly possible without using the stego key [31]. It can be generated using a pseudo-random number generator (PRNG) [32].
- Embedding process (Em): The process of generating a stego object by hiding secret data in the cover object.
- Extraction process (Ex): The process of retrieving secret data from the stego object.

Mathematically, the embedding (or concealing) process can be represented as $S = Em(C, M, K)$, and the extraction process as $\bar{M} = Ex(S, K)$. The extraction process should be reversible to the embedding process. Hence, $Ex(Em(C, M, K), K)$ should be equal to M (or $\bar{M} = M$).

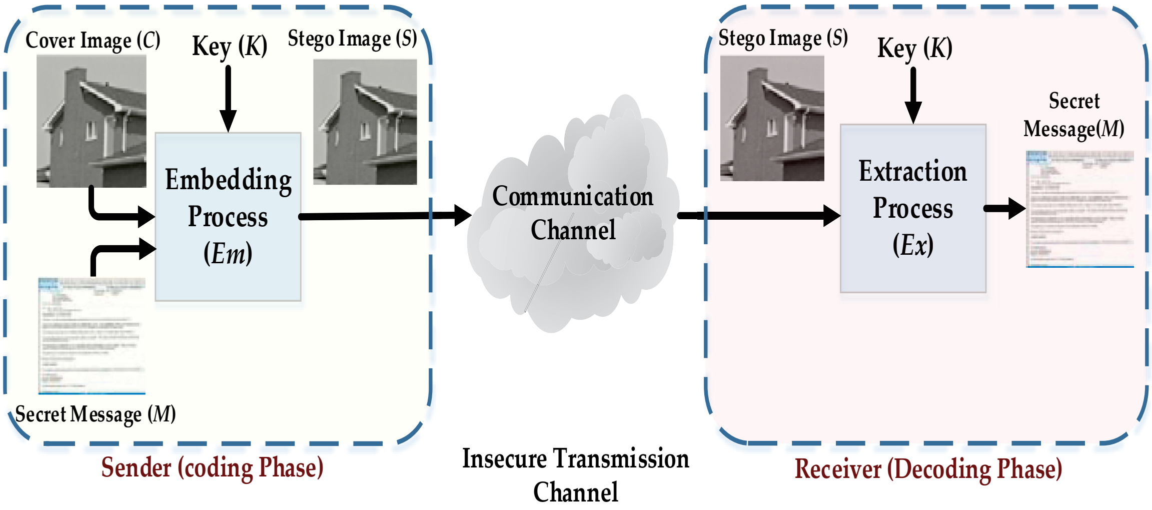


Figure 2.2 Steganography Structure

In recent years, there has been increased interest in developing digital steganography methods [33]. This recent explosion of articles in the field of steganography is due to the following reasons:

- The rapid growth of the Internet and the popular use of digital media by individuals [21].
- There is an urgent demand for copyright marks and serial numbers in digital products from the publishing and broadcasting industries.

- Strict regulations from governments on using encoding methods have stimulated individuals to find new techniques to provide privacy and confidentiality for data transmission [34].

2.5 Steganography Method Classifications

Steganography methods can be classified into various categories as per their application in securing cover files. Each method has different attributes and features [35]. Steganography methods can be classified into four fundamental categories: cover type, embedding domain, embedding and extraction approaches [30, 36]. Figure 2.1 shows a graphical representation of steganography method classifications.

2.5.1 According to the Cover Type

Since different types of digital media can be used as a carrier medium for a secret message, steganography techniques can be classified into four types depending on the cover file format: text, image, audio and video steganography methods. Each type represents the carrier medium where data will be embedded. However, each cover format has different features and these features decide how to embed the secret data in this cover file [14, 24, 37].

The most popular cover type used for embedding secret data is images because of their high degree of redundancy. Also, using images as a cover will not create any suspicion due to their widespread use on the Internet [3].

2.5.2 According to the Embedding Domain

Steganography techniques can be classified into two categories depending on the domain type: spatial and transform. Spatial domain algorithms directly embed the secret data in the cover carrier, while in the transform domain, embedding is carried out on the transform coefficients of the cover carrier. In transform domain methods, various transformations can be used to conceal the secret data, such as discrete cosine transform (DCT) and discrete wavelet transform (DWT). Transform domain algorithms usually have better robustness against attacks than the spatial ones; however, their main limitations are the high computational cost and limited embedding capacity. In comparison, spatial domain algorithms need a shorter execution time and provide a high embedding rate [14, 38]. Table 2.2 presents the differences between image steganography in spatial and transform domains in term of embedding capacity, imperceptibility and robustness [39–41].

Table 2.2 Differentiation between image steganography schemes in the spatial and transform domains

	Spatial domain	**Transform domain**
Advantage	High embedding capacity Shorter computational time High controllable imperceptibility	Robustness against attacks such as geometric attacks and compression
Disadvantage	Vulnerable to geometric attacks	High computational time Limited embedding capacity Lower controllable imperceptibility

2.5.3 According to the Embedding Process

Steganography methods can be divided into four different categories based on the embedding process applied to hide the secret message. These methods are insertions, substitution, generation and the cover lookup [36, 42, 43]. These are discussed below:

(1) Insertion based: The insertion-based method relies on inserting the secret message into specific sections in the cover medium that are neglected by the processing application that reads the cover medium. It does not modify the readable part of the cover file during the embedding process. In this method, the embedding capacity is high and there are no restrictions on the secret message length. However, the cover medium size is smaller than the stego medium size because the embedding process adds the secret data without eliminating or replacing any bit of the original cover file [36].

Embedding the secret data in a Word document between the end-text and begin-text markers is an example of the insertion method. The secret message may not be visible while displaying the file because the Microsoft Word application is designed to disregard any text included between the end-text and begin-text markers [36].

(2) Substitution based: One of the most popular and advanced steganography methods is the substitution-based method [42]. The substitution-based method depends on replacing some part of the cover medium with the secret message [36].

Unlike the insertion-based method, the cover and stego mediums are the same size because the embedding process modifies unimportant parts of the original cover with the secret data without adding any extra data. The insertion-based method, in contrast, adds the secret data into regions ignored by the processor [44].

The two main drawbacks of the substitution method are limited embedding capacity and the artefacts introduced to the stego carrier due to the embedding process which degrade the quality of the stego medium [36].

There are three different ways to select the embedding locations:

- Sequential selection: To embed the secret data, the cover elements are modified individually and in a consecutive manner. This method is simple and easy to implement. However, it has a high probability of detection [45].
- Random selection: The embedding locations are selected in a random manner. The sender and receiver need to utilize a secret key to generate the same random number subset. A pseudo-random number generator (PRNG) has a higher security level than the sequential selection [45].
- Adaptive selection: The adaptive selection rule chooses the cover elements for embedding based on their characteristics. For example, the edge detection method can be applied to the cover image to select the high contrast regions for embedding, which are less detectable than the smooth regions. In terms

of detectability, adaptive selection achieves a better security level than the sequential and random selection methods [45].

(3) Generation based: The generation-based steganography method is different to the insertion and substitution-based methods in terms of the existence of the cover medium. In other words, the cover medium is a fundamental component in all steganography methods except the generation-based method. This is because the secret message is utilized to generate a suitable stego object, therefore the generation-based method cannot be detected by the detection techniques that depend on comparing the stego medium with the cover medium [42]. However, there are a restricted number of stego mediums that can be created from the secret data that also produce a stego medium without meaningful information, such random shapes and colours, which can alert attackers to the existence of the secret message [36].

(4) Cover lookup based: In the embedding process, the cover lookup-based method searches for an existing cover medium and ensures that the cover medium is not changed due to embedding the secret message. It makes an assumption that it can find a convenient cover medium that already consists of the required confidential data. However, this method is impractical when the length of the secret message is increased [42].

2.5.4 According to the Extraction Process

The steganography methods can be classified into two main categories according to the extraction process: blind and non-blind and reversible and irreversible.

2.5.4.1 Reversible and Irreversible Types

Reversible steganography techniques are employed to restore the original image in addition to secret data from the stego image. This type is significant in medical diagnosis, military and remote-sensing applications where retrieving the cover image has the same priority as retrieving the secret data. In contrast, irreversible methods are only concerned with recovering the secret message [46, 47].

2.5.4.2 Blind and Non-blind Types

Steganography methods can be categorized into blind and non-blind based on the requirements of the extraction process.

- Blind steganography: In the blind steganography method, the cover medium is disregarded as it is not required by the receiver. Hence, as the cover medium is not necessary in the extraction process to retrieve the secret data, the sender can use any medium for embedding the secret data [42].
- Non-blind steganography: The non-blind steganography method cannot be applied without the existence of the original cover. The original cover plays an essential role in extracting the secret data in the non-blind method [42].

2.6 Digital Steganography Requirements

There are various characteristics that should be investigated to evaluate the strength as well as the drawbacks of the steganography methods. In general, a reliable steganography algorithm should satisfy some essential requirements which are in conflict. The steganography methods must comply with the features of imperceptibility and embedding capacity as crucial elements. The steganography requirements are summarized as follows [48–50]:

- Imperceptibility (perceptual transparency): Imperceptibility or perceptual transparency refers to the quality of the stego carrier. Even though, the content of the stego carrier will have some difference to the original one, if this difference is not noticeable by the human visual system (HVS) or the human auditory system (HAS), then we can say that this steganography algorithm achieves the imperceptibility requirement [51]. Imperceptibility is the major requirement of any steganography technique [52].

- Capacity (payload): The capacity denotes the number of bits that can be embedded into the cover medium. The embedding capacity commonly suffers against the imperceptibility and robustness requirements. The existing challenge in developing steganography techniques is how to achieve a high embedding capacity while maintaining a high quality and robust system [51, 53].
- Security: Security is an essential demand for steganography as the steganography method should resist steganalytic attacks. A steganography scheme is considered secure if the accuracy value of the classification tool is random guessing [53, 54].
- Robustness (resistance): Robustness refers to the capability of the stego medium to resist various type of manipulations. In other words, the embedded secret data is hard for attackers to remove or modify in an illegal way. Cropping, compression, filtering and noise addition are examples of some attacks which may be used to detect or change the secret data [50, 51].

The steganography method aims to improve its requirements such as imperceptibility, capacity and security. However, improving one particular requirement might negatively influence others; for instance, improving the quality of the stego requires a decrease in the embedding capacity to minimize the artefacts produced by the embedding process [44].

2.7 Steganography Evaluation Criteria

Each steganography method has strong and weak points. Therefore, it is important to evaluate the performance based on some criteria to utilize the most appropriate algorithm for each application. This evaluation process is vital when choosing which method is better in comparison to other existing methods. Unfortunately, there are no commonly accepted criteria to assess the effectiveness of steganographic methods. However, there are general guidelines that should be considered when developing a steganographic method. As shown in Figure 2.3, four essential requirements can be used to assess the steganography performance: imperceptibility, payload, security and computational cost [16, 42].

The similarity between cover and stego mediums, the length of the secret data and the detection of the secret data's existence or contents are the fundamental parameters that should be measured to assess the performance of the steganographic system [36].

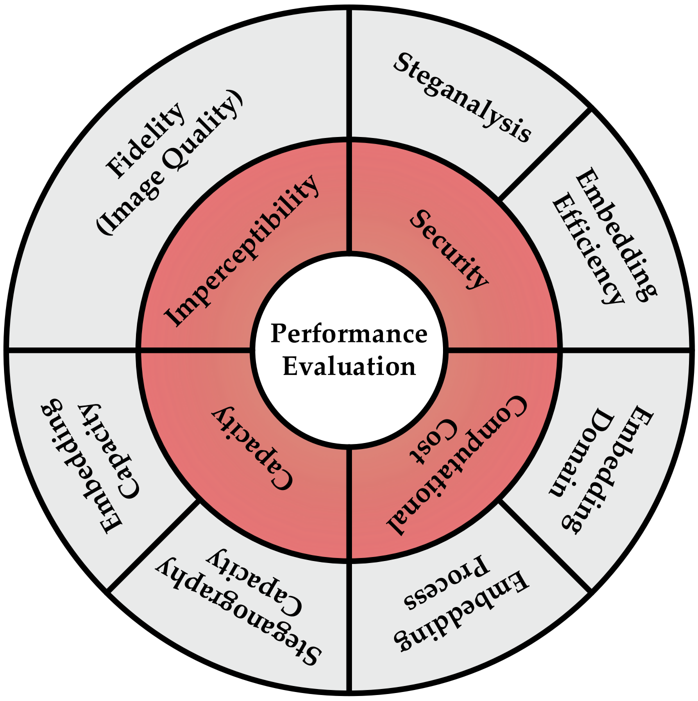


Figure 2.3 Performance evaluation criteria of steganography methods

2.7.1 Imperceptibility Evaluation

Imperceptibility, also known as perceptual transparency, is the main requirement of the steganography method. Two types of imperceptibility can be evaluated: fidelity and quality. Fidelity denotes the perceptual similarity between the cover and stego objects, whereas the quality is an absolute measure of the object appeal. For instance, secret data has been embedded into low resolution video and the stego video is almost identical to the cover video. In this case, the object has high fidelity because it is indistinguishable from the cover video. On the other hand, it is of low quality. It is necessary to use a good quality cover to avoid grabbing the attention of an unauthorized party [44].

Typical metrics of steganography fidelity are peak signal-to-noise ratio (PSNR), mean square error (MSE) and structural similarity (SSIM). PSNR and MSE measure the amount of distortion added to the original object due to the embedding process, while SSIM measures the similarity between the original and stego carriers [44].

2.7.2 Capacity Evaluation

Steganography methods are mainly used to provide secret communications by embedding secret data into a cover object. Therefore, it is important to calculate the data length that can be concealed into the cover. According to Cox et al. [32], steganographic and embedding capacities are not equivalent. The embedding capacity is the maximum number of bits that can be embedded in the cover object, while the steganographic capacity is the maximum number of undetectable bits in the cover object, where the detection probability is negligible. Generally, the steganographic capacity is less than the embedding capacity. It is difficult to determine the maximum number of undetectable embedded bits.

2.7.3 Security Evaluation

Steganographic security is divided into two categories: statistical steganalysis and embedding efficiency. The steganographic security is computed by estimating the detection probability of the existence of a secret message. The steganography method is secure if the detection probability against the steganalysis method is random guessing. The steganography method is considered weak when the existence of the secret message is detected by a specific steganalysis technique. Therefore, the detection probability of a secret message's existence is computed to determine the resistance degree of the steganography method against the steganalysis method. On the other hand, embedding efficiency represents the number of embedded bits per embedding change (introduced change). In other words, a high embedding efficiency value leads to minimizing the embedding distortion [55].

2.7.4 Computational Cost Evaluation

The computational cost of a steganography system mainly depends on several factors, such as the domain of embedding and the embedding process. The secret data may either be embedded by altering the original cover object (spatial domain) or by modifying the transformed coefficient (transform domain). It is clear that the computational cost of the transform domain is higher than the spatial domain [56].

2.8 Steganography Protocols

There are three protocol types of steganography: pure, secret key and public key steganography.

2.8.1 Pure Steganography

The pure protocol refers to a steganography system where the sender and intended recipient do not need to share any secret information such as the stego key before starting the embedding and extraction processes. The security of pure steganography is based on the privacy of the embedding and extraction procedures. The mathematical representation of the pure steganography embedding and extraction functions are described in Eq. 2.1 [30].

$$\begin{aligned} Em &: C \times M \rightarrow S \\ Ex &: S \rightarrow M \end{aligned} \tag{2.1}$$

where M is the secret message and C and S are the cover and stego mediums respectively.

To ensure the security of the pure steganography method, the embedding and extraction procedures should not be available to any unauthorized users except the two communication parties. However, pure steganography is relatively insecure according to Kerckhoff's principle, which states clearly that the embedded algorithm should be known by the third party [57].

2.8.2 Secret Key Steganography

If an attacker knows the embedding and extraction procedures (as per Kerckhoff's principle), then it is possible for them to extract the embedded message from the stego medium. To prevent attackers and unauthorized users from having access to the secret message, a stego key is required to provide the security of the exchanged information between the two communication parties. The mathematical representation of the secret key steganography embedding and extraction functions are described in Eq. 2.2 [30].

$$\begin{aligned} Em &: C \times M \times K \rightarrow S \\ Ex &: S \times K \rightarrow M \end{aligned} \tag{2.2}$$

where M is the secret message, K is the stego key and C and S are the cover and stego mediums respectively.

The sender and receiver need to exchange the stego key before starting the embedding process. Consequently, the separate transmission of the stego key conflicts with the fundamental objective behind steganography. This issue can be solved if the sender and receiver agree to use a stego key before detainment [58].

2.8.3 Public Key Steganography

Public key steganography refers to the system that has two mathematically related keys: the public and private. Public key steganography is similar to the public cryptography system where it is introduced to avoid the addition transmission of the stego key (private) between the sender and receiver. The public key is available to everyone via a publicly accessible repository, while the private key must remain confidential to its respective owner. The mathematical representation of the secret key steganography embedding and extraction functions are described in Eq. 2.3 [30].

$$\begin{aligned} Em &: C \times M \times K_p \to S \\ Ex &: S \times K_r \to M \end{aligned} \tag{2.3}$$

where M is the secret message, K_p is the public key, K_r is the private key and C and S are the cover and stego mediums respectively.

2.9 Digital Steganography Applications

Steganography is utilized in different fields as data privacy and confidentiality are significant issues due to the growth in Internet communication technologies. In recent years, many applications have employed steganography to conceal their data during transmission, for example human rights defenders in situations where some governments and Internet service providers (ISPs) have imposed strict regulations to forbid individuals from employing data encryption [34], improving the robustness of image search engines, analyzing the network traffic of specific clients to embed a unique number into an image, and smart identity card (ID) applications, where personal information is hidden in a photograph [59, 60].

Digital steganography also has attractive characteristics that fit within real-time applications. Therefore, a massive number of steganography techniques have been designed to adapt Voice over IP (VoIP) services. VoIP steganography has grown because IP telephony is very popular [61]. Also, short VoIP connections do not give eavesdroppers enough time to discover any irregularities because of the embedded message [62]. VoIP steganography is different than using a traditional file format such as text, image or audio. It is a real-time scheme, which uses VoIP signals to conceal the existence of the secret data in the real-time communication [42, 61].

In [30], the author mentions some modern domains that integrate steganography into their systems. The digital steganography method has played an essential role inside medical information systems (MIS) in terms of protecting electronic patient records' (EPRs') confidentiality. The basic application of steganography in medical imaging systems was proposed to provide a solution for the authentication problem, where sometimes the relation between the patient's information and their image is lost. Therefore, steganography is employed to embed patients' information and diagnosis reports inside their medical images. A survey of the effect of information security and confidentiality on designing telemedicine application is accessible [63].

In recent years, business security has become essential to the security of countries, as they deal with large transactions that need to be confidential. Each organization must preserve data from potential attackers with the aid of steganography methods.

2.10 Steganalysis

Nowadays, various image steganography systems are available for individuals. Subsequently, there is increased interest regarding how we can differentiate whether an image has secret embedded data to ensure the development of steganography is not utilized for improper intent. This counter-activity is referred to as steganalysis [64].

Steganalysis is the science of discovering the presence of secret messages or extracting data that is embedded within the stego medium without requiring prior information, such as the secret key or the embedding process that has been utilized. In other words, steganalysis refers to the study of breaking steganography methods. It starts by determining the artefacts that exist in the file which was created due to the embedding process. A steganography system is not only considered broken if the steganalysis technique is able to retrieve the embedded message, but also if the steganalysis method is able to detect the existence of the secret message within the stego file [65].

The rapid development in steganography techniques motivated a researcher to implement reliable steganalysis algorithms [66]. However, it is a difficult and challenging process to propose steganalysis techniques because Kersckhoff's principle is not applicable where the information about the embedding process is not available [67]. In general, most steganography methods leave a distortion in the stego file, and these distortions in the structure can facilitate the detection of the presence of secret data even though it is not distinguishable by humans. Changing any area in the cover file will also modify the properties of the cover, which can be an indication of the existence of secret data. Hence, a straightforward correlation between the stego file and its relating cover file may uncover the presence of the embedded message inside the stego file. The absence of the cover file exposes the weakness of the steganalysis. Therefore, the cover file should not be declared publicly or destroyed after the embedding process to avoid the comparison.

Steganalysis methods have been designed for different purposes. For example, some algorithms are intended to identify the absence/presence of secret data while other algorithms aim to extract the embedded data from the stego file. Depending on the output of the method, steganalysis can be arranged into passive and active techniques [58].

Passive steganalysis is the most popular technique for identifying the presence/absence of secret data and/or determining the algorithm of the embedding process [68]. Depending on the steganography method that was used for embedding, passive steganalysis extracts either first-order or second-order statistical features. Then, a classifier technique needs to be trained on the features of the cover and stego images to differentiate between them [58].

In contrast to passive steganalysis, active steganalysis is more complex and less popular because it attempts to estimate the embedded message length, locations of the embedded message, secret key and/or extract the embedded message [68].

2.10.1 Steganalysis Approaches

Steganalysis can be classified into two main types depending on the detection method used to discover the distortion created by the embedding process [69]. The steganalysis methods can be classified into visual and statistical approaches.

2.10.1.1 Visual Steganalysis

The detection of confidential data is done using human senses such as hearing or seeing. For this reason, it is considered to be the simplest method of steganalysis. Although visual steganalysis cannot be used for the JPEG steganography method, it can be used to identify the distortion of the simplest steganography method, which is called least significant bit (LSB) [69]. Moreover, if the image contains uniform regions then it is easy to break steganography methods using a visual attack.

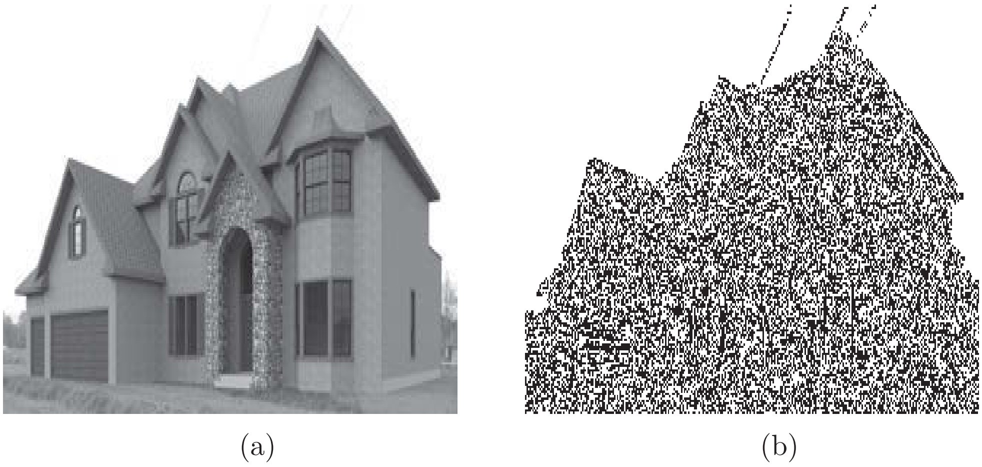


Figure 2.4 (a) Cover image and (b) first LSB plane of the cover image

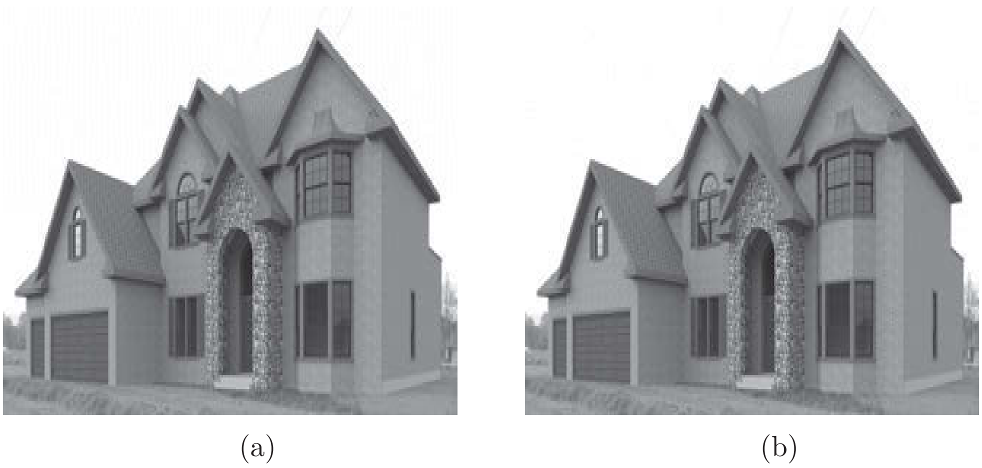
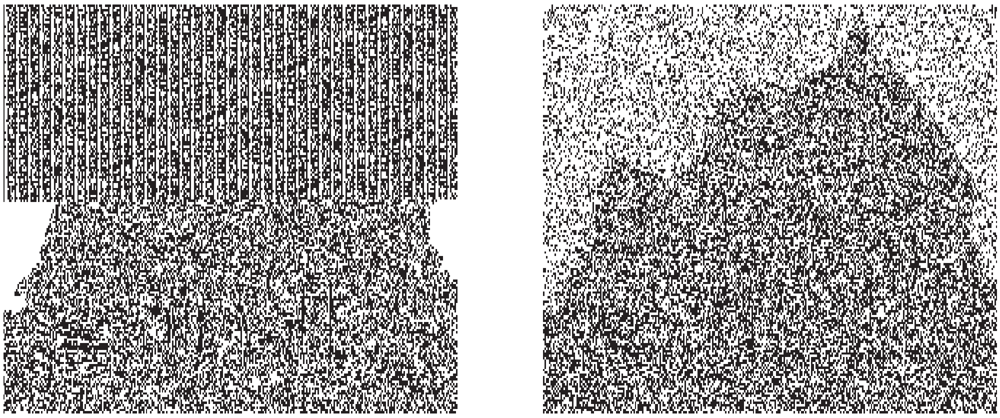


Figure 2.5 The stego images with 47.8% embedding rate using (a) sequential and (b) random embedding locations

Figure 2.4a shows the original (cover) image and Figures 2.5a and 2.5b show the corresponding stego images obtained by embedding 3.16 Kbyte using the LSB



(a) The first LSB plane of stego images (b) The first LSB plane of stego images

Figure 2.6 The first LSB plane of stego images with 47.8% embedding rate using (a) sequential and (b) random embedding locations

steganography method in sequential and random embedding locations respectively. Figures 2.4b, 2.6a and 2.6b show the first LSB plane of the cover and the corresponding stego images. It is easy for the human visual system (HVS) to observe the variation between the original plane and the two stego planes where the flat region in the cover image appears as a noise in the stego images, even after choosing random pixel for embedding.

2.10.1.2 Statistical Steganalysis

Statistical steganalysis detects confidential data based on the mathematical statistic properties of the carrier contents. Statistical steganalysis is more robust than visual steganalysis due its property of detection even when a carrier content contains only a small alteration [69]. However, it does not expose the steganography technique that was used for alternation, which is the main limitation of this method [70]. For instance, in order to consider the statistical properties of an image, the image histogram can be utilized as a statistical tool to detect if there is any colour variation compared to the original image histogram [44].

The chi-square (x^2) test is the easiest statistical attack that has been applied by [71] to determine the randomness in a sequence of data. It is used to assess the goodness of fit of the observed data to the expected data to differentiate between the original and random sequences based on the assumption that the embedded data is random while the original values are not.

2.11 Summary

In this chapter, the fundamental issues regarding the digital steganography process were introduced. The main components and requirements of the steganographic model were explained. Also, the trade-offs between the different steganographic requirements were discussed.

Information security can be classified into cryptography and information-hiding categories. Steganography is an alternative tool for cryptography to provide secure communication without attracting unauthorized attention.

Imperceptibility, capacity, security and robustness are the requirements for steganography. However, imperceptibility and secret data payload are considered the main requirements for developing a steganography framework.

In addition to this, embedding efficiency and embedding payload are the two essential factors that should be taken into consideration to produce a successful steganography method. First, high embedding efficiency means the stego carrier will be good quality with less distortion in the cover carrier due to the embedding process. Second, the high embedding capacity provides the ability to hide large secret data inside the cover carrier.

A steganography method is considered broken if the steganalysis retrieves the embedded secret data or if the existence of the secret data is discovered by any unauthorized user. Therefore, it is possible to utilize steganalysis to improve steganography security.

The evaluation process of steganography is an essential step for selecting a suitable steganography technique for a specific application. There are general guidelines for assessing the developed steganography techniques such as imperceptibility, capacity, security and speed.

CHAPTER 3

Literature Review on Digital Steganography

This chapter is intended to provide an overview of the basic image steganography methods in the spatial and transform domains. This is followed by a comparative analysis of the most popular steganography techniques, then some of the existing biomedical steganography methods are explained.

3.1 Introduction

A steganography scheme is generally applied to conceal the existence of secret data while a cryptography scheme is utilized to protect the content of the secret data by concealing the meaning of the secret data. Both schemes are complementary to each other [24].

In recent years, many steganography methods have been proposed, most of which are based on the substitution system. The substitution method replaces the redundant data of the cover carrier with data from the secret message [72]. This type of steganography technique has a high embedding capacity but is vulnerable to steganalysis attack.

While various other methods have been developed to be more robust against attacks, they cannot conceal a large amount of secret data [73].

There are several approaches for classifying steganography techniques. One classification is according to the type of cover that is used in the embedding process. Another classification is based on the embedding algorithm applied to the cover carrier to conceal the secret data. However, the embedding domain is probably the most popular criteria for group steganography algorithms [74].

All digital formats, such as text, image, audio and video, can be a potential cover medium into which to embed secret data, but the most preferable format is one that has a high degree of redundancy such as an image [74]. Hence, digital image steganography in particular has drawn the attention of a large number of researchers. Moreover, digital image is considered the perfect carrier among others document types due to their high degree of redundancy as well as the characteristics of the human visual system. Additionally, using images as a cover will not create any suspicion due to their widespread use on the Internet [3].

Digital image steganography methods are categorized into spatial and transform domains according to the embedding domain in which the secret data is embedded [14].

This chapter is organized as follows. Section 3.2 presents basic steganographic methods. State-of-the-art research on image steganography based on edge detection, coding theory and wavelet transform methods are introduced in Sections 3.3.1, 3.3.2 and 3.3.3 respectively. Steganography for medical image security is discussed in Section 3.4. Finally, the summary is given in Section 3.5.

3.2 Basic Steganographic Methods

3.2.1 Spatial Domain Steganography

In the spatial domain steganography method, secret data is directly embedded in the cover image. There are numerous methods based on spatial domain, such as least significant bit (LSB), pixel value differencing (PVD) and pixel mapping methods.

3.2.1.1 Least Significant Bit (LSB)

The LSB substitution is the most common and the simplest technique to hide data within the cover image. This method hides data bits in the last significant bit of an image pixel. It is capable of embedding large secret data in a cover without introducing noticeable distortion [24, 29]. LSB steganography works by converting the secret message into a binary bit stream, then replacing the least significant bits of the cover object with the message bits. When LSB replacement is applied, a pixel of odd value will either keep its value or decrease it by one. Nevertheless, it will not be deceased. For even-valued pixels, the inverse is true [75, 76]. Eq. 3.1 presents the embedding process of LSB steganography.

$$S_i = \begin{cases} C_i + 1, & \text{if } m_i \neq LSB(C_i) \text{ and } C_i \text{ is even} \\ C_i - 1, & \text{if } m_i \neq LSB(C_i) \text{ and } C_i \text{ is odd} \\ C_i, & \text{if } m_i = LSB(C_i) \end{cases} \tag{3.1}$$

where C_i is the i^{th} cover pixel value before the embedding, S_i is the i^{th} pixel value after embedding process, and m_i is the i^{th} message bit.

The message can then be extracted from the image by retrieving the pixel LSB and combining every 8 bits together to form single character. The embedding and extraction procedures are illustrated in Algorithms 3.1 and 3.2. The message bits can be placed sequentially by columns or rows of the image pixels [20, 77]. Figure 3.1 shows an example of the LSB substitution scheme.

Algorithm 3.1: LSB Embedding Process

```
Inputs  : Cover Image ($C$), Secret Message ($M$).
Outputs : Stego Image ($S$).
$i \leftarrow 1$ ;
for $i \leq Length(M)$ do
    if $m_i \neq mod\ (C_i, 2)\ and\ mod\ (C_i, 2) = 0$ then
        $S_i \leftarrow C_i + 1$;
    else if $m_i \neq mod\ (C_i, 2)\ and\ mod(C_i, 2) = 1$ then
        $S_i \leftarrow C_i - 1$;
    else if $m_i = mod\ (C_i, 2)$ then
        $S_i \leftarrow C_i$;
```

Algorithm 3.2: LSB Extraction Process

```
Inputs  : Stego Image ($S$).
Outputs : Secret Message ($M$).
$i \leftarrow 1$ ;
for $i \leq Length(M)$ do
    $m\prime_i \leftarrow LSB(S_i)$ ;
```

The embedding rate of the LSB algorithm depends on the size of the original image. For example, it can hide about 32 Kbyte in (512 × 512) grey images. The stego image is also very similar to the original image because the modification occurs in the least significant bit [78].

Figure 3.2 illustrates an example of embedding on the n^{th} bit position from the 1^{st} LSB to the 8^{th} most significant bit (MSB). It can be noticed that embedding in the first, second and third bit position is undetectable visually, while embedding on bits from 4^{th} LSB to 8^{th} MSB produce noticeable distortion. It is observed that embedding in the first LSB changes the texture of the cover image. Therefore, this modification leads to a statistical difference between the original and stego images (texture content).

Figure 3.3 explains the relationship between bit level and imperceptibility requirement. As shown in Figure 3.2 and Figure 3.3, they are contradicting each other, where using the MSB for embedding gives high degradation in the stego image. The converse is also true for using LSB for embedding.

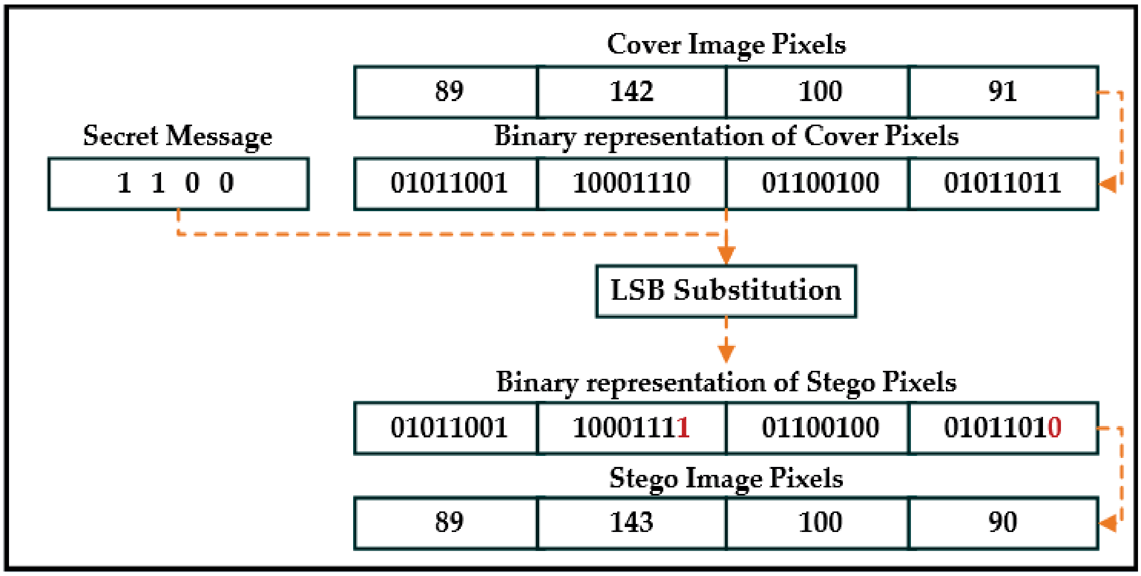
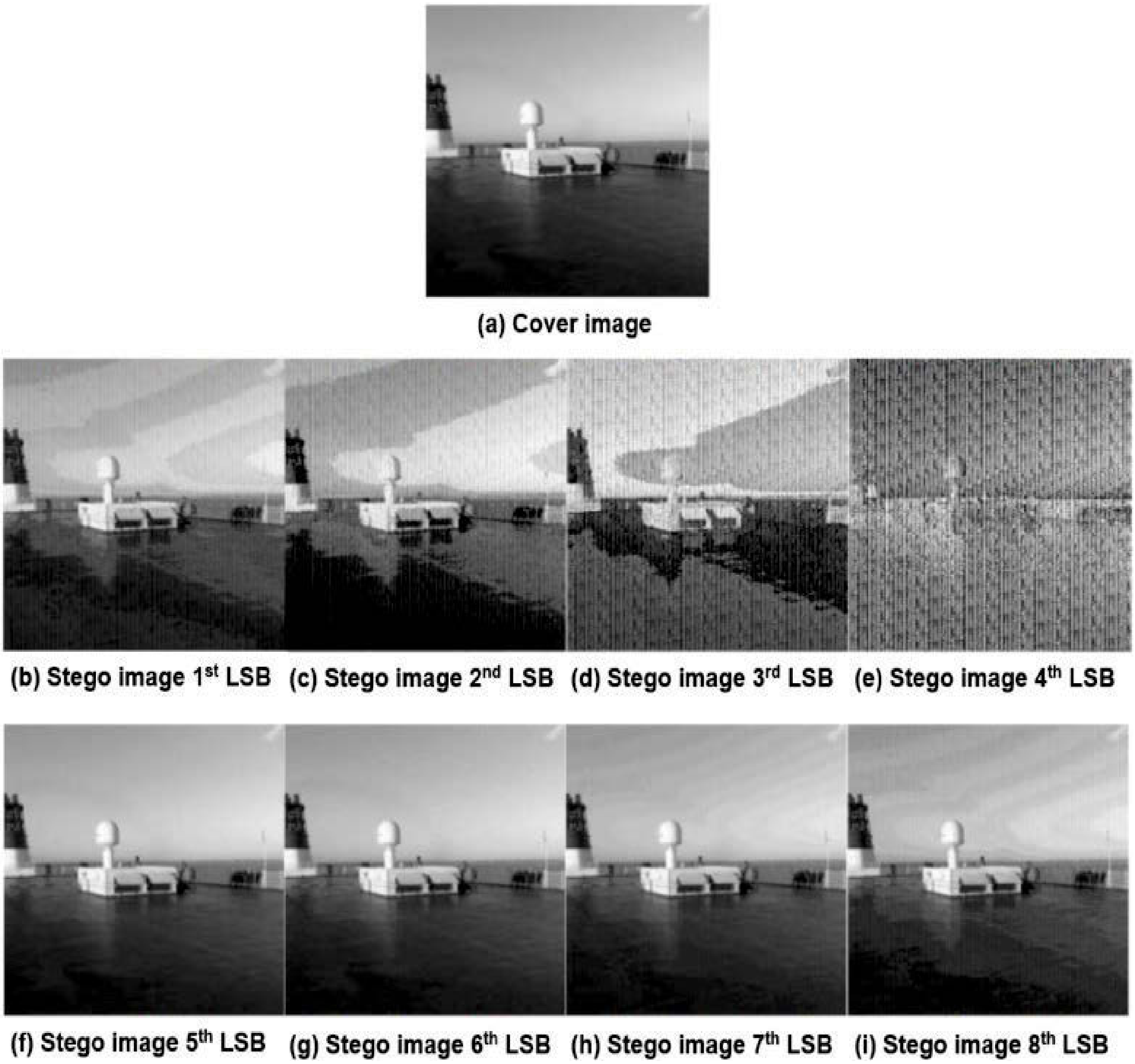


Figure 3.1 An example of the LSB embedding process

On the other hand, the LSB techniques are vulnerable to statistical attacks, which would allow unauthorized users to extract the secret data message. Moreover, LSB does not resist any kind of image-processing operation such as compression and clipping. Attackers could therefore expose a secret message easily if they discovered the stego image [78, 79].

Several improvements have been introduced to the original method, such as the incorporation of pseudo-random number generator (PSNG), which is also known as randomised embedding technique. The secret message is randomly scattered over a cover rather than sequentially embedded. To retrieve the embedded data, only the selected pixels are required to extract the message [37]. A randomized scheme aims to make it difficult for steganalysts to discover the secret message. However, when the size of the secret message is increased, the possibility of selecting the same pixel is increased [80].

LSB matching, also called the ±1 embedding scheme, is a refined version of the LSB method. In this scheme, the pixel value is randomly incremented or decremented to match the secret message bits as shown in Eq.3.2. If the LSB of the cover pixel matches with the secret bit, then pixel value stays as it is. However, if the LSB of the pixel



(a) Cover image

(b) Stego image 1st LSB (c) Stego image 2nd LSB (d) Stego image 3rd LSB (e) Stego image 4th LSB

(f) Stego image 5th LSB (g) Stego image 6th LSB (h) Stego image 7th LSB (i) Stego image 8th LSB

Figure 3.2 (a) Cover image, (b - i) Stego images using n^{th} bits (from 1-LSB to 8-LSB)

does not match with the secret bit then, according to Eq. 3.2, ±1 is added to the pixel value. Figure 3.4 shows an example of the LSB matching method.

$$S_i = \begin{cases} C_i + 1, & \text{if } m_i \neq LSB(C_i) \text{ and } (k > 0 \text{ or } C_i = 0) \\ C_i - 1, & \text{if } m_i \neq LSB(C_i) \text{ and } (k < 0 \text{ or } C_i = 255) \\ C_i, & \text{if } m_i = LSB(C_i) \end{cases} \tag{3.2}$$

where k is a random variable with constant distribution $\{+1, -1\}$, C_i is the i^{th} cover pixel value before the embedding, S_i is the i^{th} pixel value after the embedding process, and m_i is the i^{th} message bit.

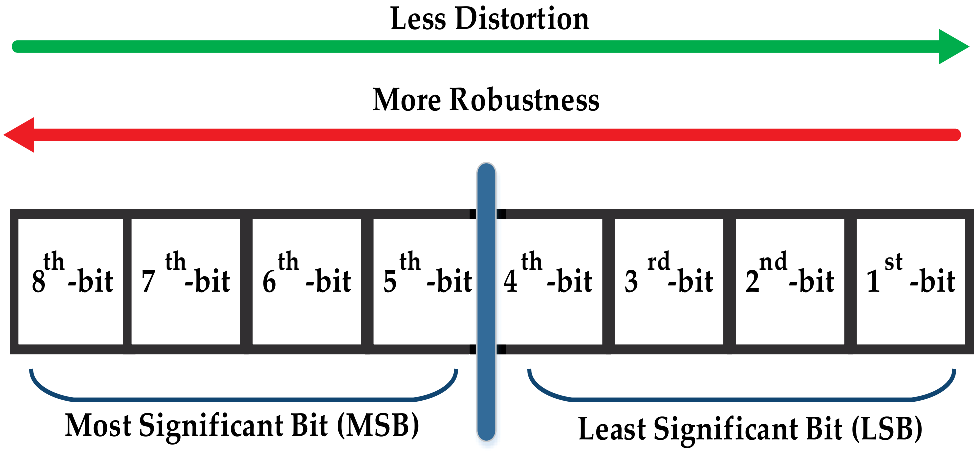


Figure 3.3 Binary representation of grey-scale pixel shows the relationship between distortion and bit level

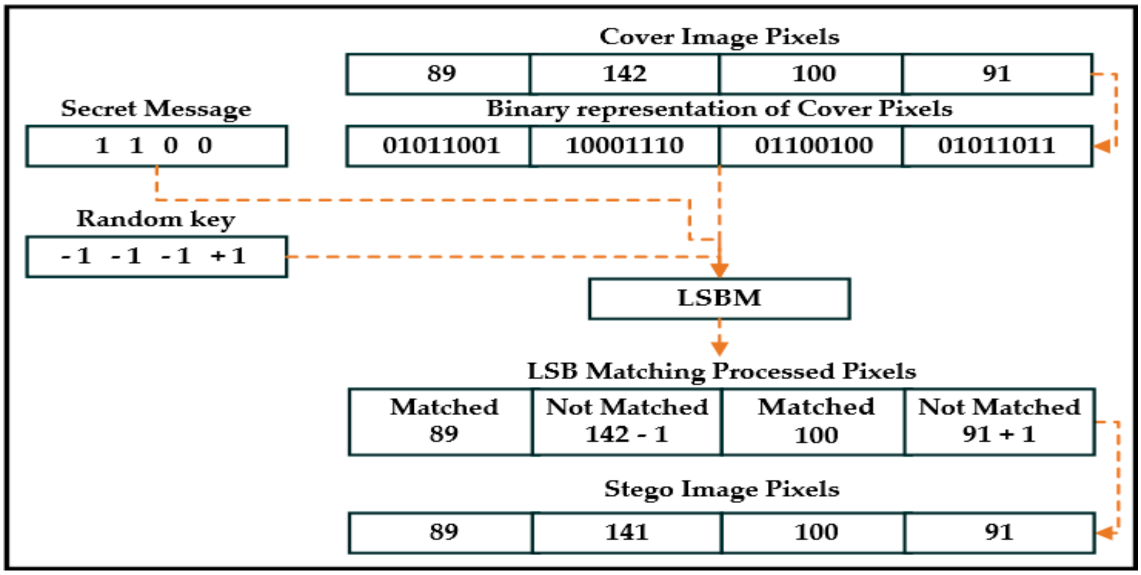


Figure 3.4 An example of the LSBM embedding process

3.2.1.2 Pixel Value Differencing (PVD)

Pixel value differencing (PVD) is another powerful spatial domain image steganography technique using block-based to embed the secret data directly in the cover pixels. Wu et al. [81] proposed the first version of PVD to provide both a high embedding capacity and imperceptibility by dividing the cover image into non-overlapping blocks of two consecutive pixels. A non-fixed number of message bits are embedded inside the cover.

Later on, other authors introduced modified versions of PVD to improve the embedding capacity of PVD such as tri-pixel value differencing (TPVD) [82–84].

The PVD technique is based on human vision's sensitivity to grey variations from smooth to high contrast regions. The embedding rate in a smooth area is less than in a complex area. It starts by distributing the grey image into blocks of two pixels and calculates the difference value of two consecutive pixels in each block. Each block is classified based on the difference of the grey values of the two pixels in the block. A large difference value indicates that the block is in a sharp region while the small difference value indicates that the block is in a smooth region. The human visual system (HVS) is used as a measurement to decide the appropriate areas in the cover image to embed more data and not leave any visual perceptible distortion [81]. The structure of the PVD technique is presented in Figure 3.5.

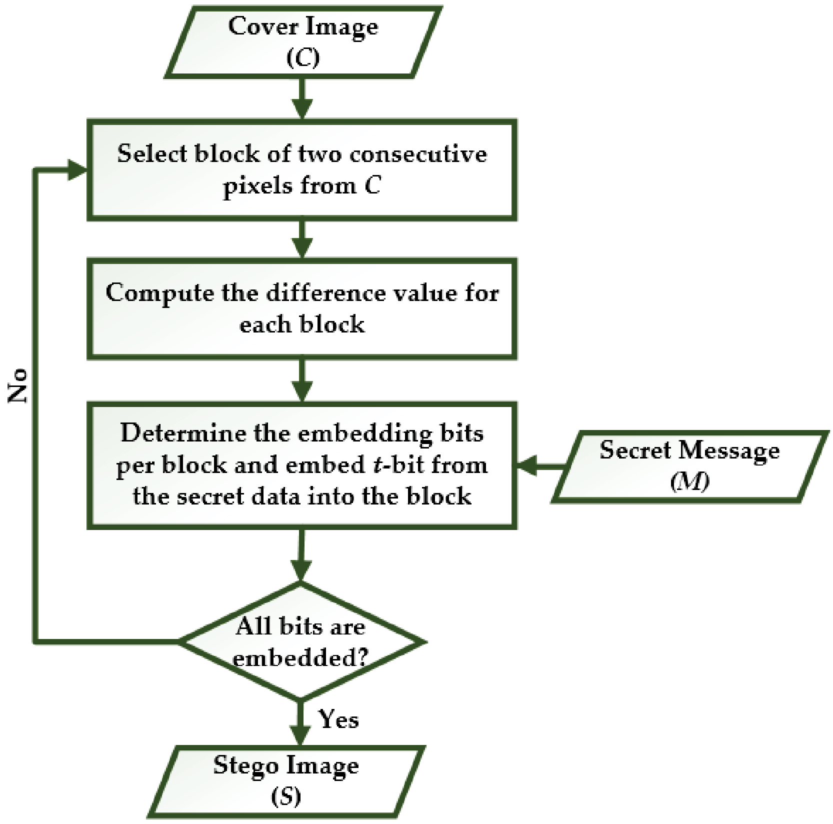

Cover Image (*C*)
Select block of two consecutive pixels from *C*
Compute the difference value for each block
Determine the embedding bits per block and embed *t*-bit from the secret data into the block
Secret Message (*M*)
All bits are embedded?
No
Yes
Stego Image (*S*)

Figure 3.5 Block diagram of PVD method

In the PVD method, a grey-scale image is distributed into non-overlapping blocks of two consecutive pixels (P_i and P_{i+1}). The absolute difference values (d_i) are computed by subtracting P_i from P_{i+1}. The set of all difference ranges from 0 to 255. A block with a large value difference is observed as a block of sharp regions, while a small value difference block is placed in a smooth area. The range table R_k is designed with n contiguous sub-ranges, where $K = 1, 2, \ldots, n$. l_k and u_k represent the lower and upper bound of the sub-range of R_k. The width of the sub-range w_k determines how many bits can be embedded in the block pixels where w_k is computed by $w_k = u_k - l_k + 1$. Figure 3.6 shows an example of the PVD embedding process.

Table 3.1 PVD Range Table (R_k)

Sub-range	R1	R2	R3	R4	R5	R6
Lower - Upper	[0 – 7]	[8 – 15]	[16 – 31]	[32 – 63]	[64 – 127]	[128 – 255]
Hidden bits	3	3	4	5	6	7

The steps of PVD algorithm are described as follows:

Inputs: Cover image (C), secret message (M), range table (R).

Output: Stego image (S).

Step 1 : Divide C into blocks of two adjacent pixels (P_i, P_{i+1}).

Step 2 : Calculate the absolute difference value $d_i = |p_i - p_{i+1}|$, for each block.

Step 3 : Use the range table to find out the sub-range which d_i belongs to. Where $d_i \leq u_i$ for all $K = 1, 2, \cdots, n$, d is used to determine the number of secret bits to embed in each block.

Step 4 : Compute the width w_k of R_k as follows: $w_k = u_k - l_k + 1$, where u_k is the upper bound of R_k and l_k is the lower bound of R_k.

Step 5 : Calculate the number of bits (t) to be hidden in a block. It can be computed using Eq. 3.3.

$$t = \lfloor \log_2 w \rfloor \tag{3.3}$$

Step 6 : Read t-bits from the secret message and convert it into decimal number b.

Step 7 : Find the new difference $d'_i = l_i + b$.

Step 8 : The new values of the pixels can be computed using Eq. 3.4.

Repeat steps 1−8 until all secret data are embedded into the cover image.

$$(P'_i, P'_{i+1}) = \begin{cases} \text{if } p_i \geq p_{i+1} \text{ and } d'_i > d_i, \text{ then } (p_i + \lceil (d'_i - d_i)/2 \rceil, p_{i+1} - \lfloor (d'_i - d_i)/2 \rfloor) \\ \text{if } p_i < p_{i+1} \text{ and } d'_i > d_i, \text{ then } (p_i - \lfloor (d'_i - d_i)/2 \rfloor, p_{i+1} + \lceil (d'_i - d_i)/2 \rceil) \\ \text{if } p_i \geq p_{i+1} \text{ and } d'_i \leq d_i, \text{ then } (p_i - \lceil (d'_i - d_i)/2 \rceil, p_{i+1} + \lfloor (d'_i - d_i)/2 \rfloor) \\ \text{if } p_i < p_{i+1} \text{ and } d'_i \leq d_i, \text{ then } (p_i + \lceil (d'_i - d_i)/2 \rceil, p_{i+1} - \lfloor (d'_i - d_i)/2 \rfloor) \end{cases} \tag{3.4}$$

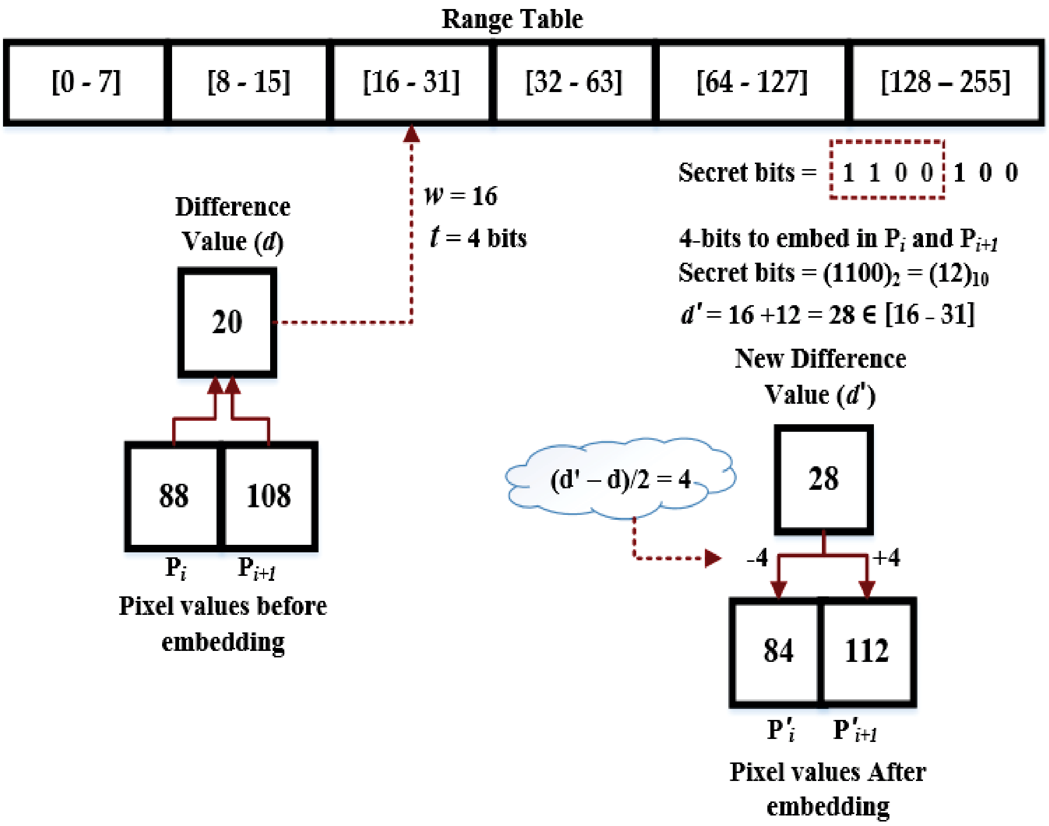

Range Table
[0 - 7]
[8 - 15]
[16 - 31]
[32 - 63]
[64 - 127]
[128 – 255]
Secret bits = 1 1 0 0 1 0 0
w = 16
t = 4 bits
Difference Value (d)
20
4-bits to embed in P_i and P_i+1
Secret bits = (1100)_2 = (12)_10
d' = 16 +12 = 28 ∈ [16 - 31]
New Difference Value (d')
88
108
P_i
P_i+1
Pixel values before embedding
(d' – d)/2 = 4
28
-4
+4
84
112
P'_i
P'_i+1
Pixel values After embedding

Figure 3.6 PVD embedding process example

According to the properties of human vision, eyes can tolerate more changes in sharp-edged areas than smooth areas. Therefore, the PVD method has a high embedding capacity while preserving good quality. However, it cannot utilize all edge directions because it depends on a single direction of either horizontal or vertical edges. Also, PVD can cause significant distortions to the stego image histogram.

3.2.1.3 Pixel Mapping Method

The pixel mapping method (PMM) is an algorithm for hiding data in a grey-scale image. It is used to enlarge the capacity of the embedded data without introducing a visual perception to the stego image. It starts by partitioning the cover into blocks. Every block has a seed pixel which determines the number of bits to embed [85, 86].

Embedding pixels are selected based on a mathematical function which depends on the pixel intensity value of the seed pixel. Its eight neighbours are selected in a counter-clockwise direction. Data embedding is done by mapping each two or four bits of the secret message in each of the neighbour pixels based on some features of that pixel [85, 86].

3.2.2 Transform Domain Steganography

In transform domain steganography, the cover image is transformed to another domain, and then secret data is embedded in the coefficient of the cover image. Spatial domain techniques are based on embedding the secret message within the original cover directly. It is also characterized by the ability to include a huge amount of secret data within the cover in contrast to the transform domain method. However, most spatial domain methods are affected by any modification operations on the original cover. In other words, secret data is lost and destroyed when an attacker performs any image-processing operations such as compression, clipping or cropping [14, 23].

The order of objectives to be achieved from steganography varies from one application to another. Some applications, such as the military ones, aim to protect data from any modification during transmission. Transform domain techniques promise to provide secure and robust steganography methods, but this is done at the expense of data embedding efficiency. Instead, transform domain steganography is concerned with converting the original cover to another form and hiding data within the new domain.

Transform domain steganography is more resistant to sabotage and imperceptible to the naked eye. There are many transform domain functions that can be used to convert the cover pixels to frequency value and embed secret data [23, 77].

Transform domain steganography techniques embed the secret data by altering certain coefficients in the transform domain of the image after applying one or more transforms such as discrete cosine transform (DCT), discrete fourier transform (DFT) or discrete wavelet transform (DWT) [77].

3.2.2.1 Discrete Transform Domain

Joint photographic expert group (JPEG) is the most common image format used for sharing images among people and over the Internet. It is designed to support a wide range of applications. JPEG (implemented using DCT) became the most popular image file format due to its high compression ratio and good quality.

Figure 3.7 shows the block diagram of a JPEG image. The JPEG encoding operation is comprised of three main stages: DCT, quantization and entropy encoding. At the beginning, in order to achieve a good compression ratio the components of the RGB image are converted into any Luminance-Chrominance colour space such as YCbCr. After that, the image is broken into block of size 8×8 pixels to transform into 64 DCT coefficients using Eq. 3.5. In the lossy compression process, the 64 DCT coefficients are quantized using the default quantization table as shown in Tables 3.2 and 3.3. Each DCT coefficient $F_{(u,v)}$ is divided by the corresponding value $Q_{(u,v)}$ from the quantization table. In the next step, zig-zag traversal is performed on the 8×8 block to compress the image using two entropy coding techniques, run length coding (RLC) and Huffman coding [44, 87].

The scaled average value of the 8×8 block intensity is represented in $F_{(0,0)}$, which is called the (DC) coefficient. The other coefficients are called (AC) coefficients.

The JPEG decoding operation is shown in Figure 3.8. It is also comprised of three steps: entropy decoding, dequantization and inverse discrete cosine transform (IDCT). At the beginning, the compressed image is decoded using two entropy coding techniques, run

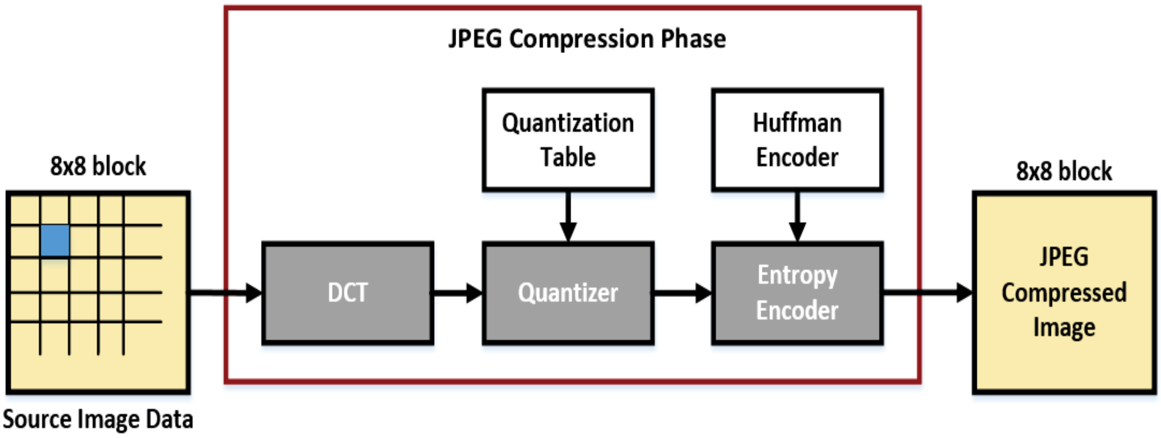


Figure 3.7 JPEG encoding

Table 3.2 The default JPEG quantization table for Luminance

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

length coding (RLC) and Huffman coding, to produce the quantized DCT coefficient $(F(u, v))$. Then, each block of the quantized DCT coefficient is converted to their approximate value using Eq 3.7. Finally, the IDCT is applied to reconstruct the spatial value using Eq. 3.6.

$$F(u,v) = \frac{1}{4}C(u)C(v)\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\left[\frac{\pi(2x+1)u}{16}\right]\cos\left[\frac{\pi(2y+1)v}{16}\right] \tag{3.5}$$

$$F(x,y) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)F(u,v)\cos\left[\frac{\pi(2x+1)u}{16}\right]\cos\left[\frac{\pi(2y+1)v}{16}\right] \tag{3.6}$$

Table 3.3 The default JPEG quantization table for chrominance

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

for $u = 0, \cdots, 7$ and $v = 0, \cdots, 7$

where $C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & for\ u, v = 0 \\ 1 & \text{otherwise} \end{cases}$

$$F(u, v)' = \{F(u, v) \times Q(u, v) | u, v \in \{0, 1, \ldots, 7\}\} \tag{3.7}$$

where $F(u, v)'$ is the approximate DCT value of the quantized DCT coefficient $(F(u, v))$.

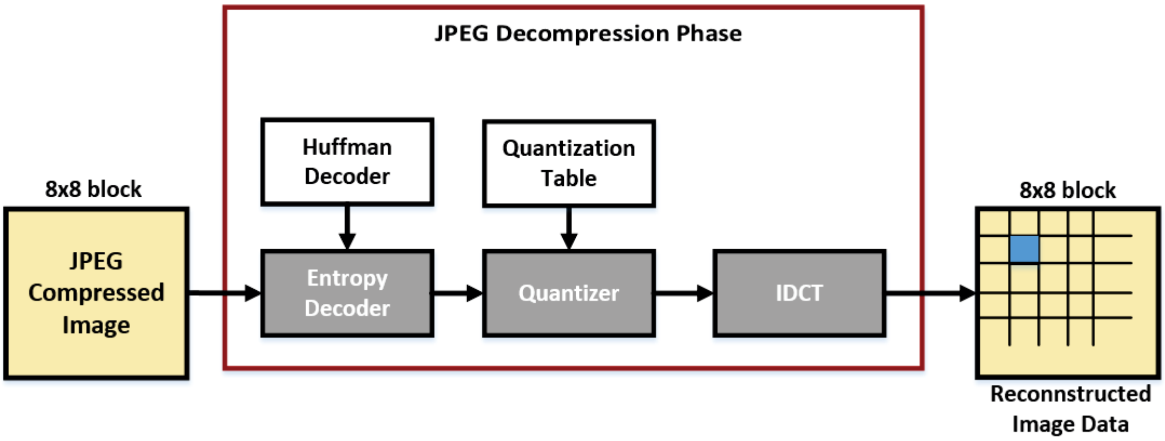


Figure 3.8 JPEG decoding

DCT is considered one of the most important transform domains in steganography in terms of data security because of its high resistance to sabotage. These methods hide messages in more significant areas of the cover and they are typically more robust than other steganography techniques [23, 88]. There are no visual attacks versus the JPEG image because it modifies the content in the frequency domain. Figure 3.9 defines the regions of each frequency in the DCT block, where F$_L$, F$_M$, and F$_H$ represent the lowest, medium, and highest frequencies respectively.

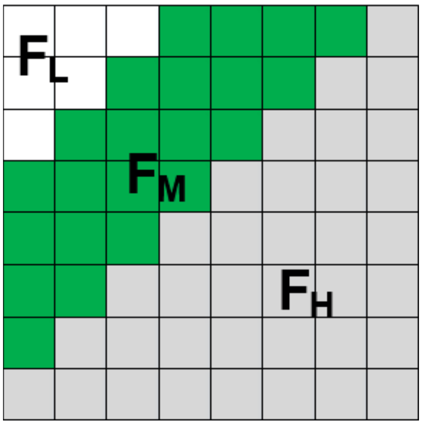


Figure 3.9 DCT regions

3.2.2.2 JPEG Based Steganography

JPEG encoding is the most popular compression standard utilized for still images and a large number of steganography methods have been implemented for the JPEG image file format, such as JSteg, Outguess and F5. The most common algorithm to embed data in quantised DCT coefficients is LSB, which is done by replacing the LSB of DCT coefficients with the secret bit. Figure 3.10 represents the general steganography for JPEG encoding where secret data is embedded after the lossy compression (quantization phase).

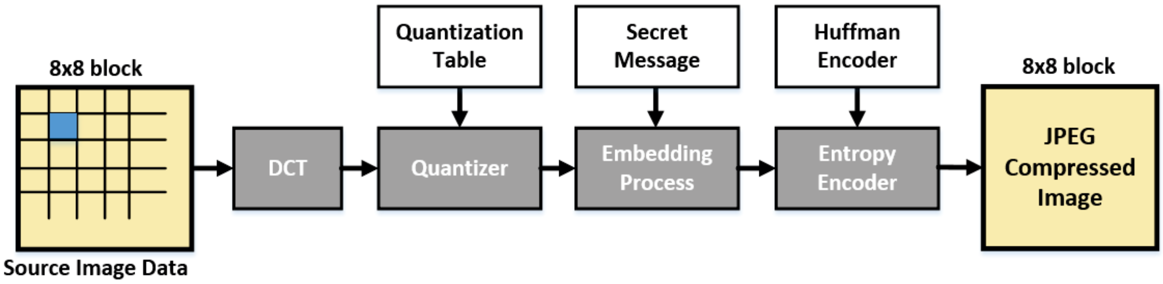


Figure 3.10 Steganography method for JPEG encoding

The JSteg algorithm [89] was developed by Upham and was the first JPEG steganography algorithm. It uses the LSB of the DCT coefficient. The secret data is hidden after compressing the cover image using lossy compression method, where the DCT coefficient is scaled using the default quantization table from the JPEG standard.

Algorithm 3.3 explains Upham's algorithm: the embedding process checks the DCT coefficient in a zig-zag order. JSteg does not utilize 0 or 1 coefficients in the embedding process because it creates a perceptually and statistically artefacts. Otherwise, it will replace the LSB of DCT with the secret message bit [89].

JSteg provides a high compression ratio and accepted capacity of about 12.8% of the cover image size compared to other transform domain steganography algorithms; however, it is still limited since the zero coefficient is large and not used in the embedding process [23, 90].

Algorithm 3.3: JSteg Embedding Process

```
Inputs  : Cover image ($C$), secret message ($M$).
Outputs : Stego image ($S$).
$i \leftarrow 1$ ;
for $i \leq Length(M)$ do
    $C_i \leftarrow$ Coefficient from $C$ ;
    while $C_i = 0$ or $C_i = 1$ do
        $C_i \leftarrow$ next Coefficient from $C$;
    $C_i \leftarrow \mathrm{mod}(C_i, 2) + m_i$ ;
    $S_i \leftarrow C_i$ update stego image coefficient ;
```

The JSteg algorithm is not detected by any visual attacks [71]. However, the modifications caused by the JSteg embedding process deforms the histogram of the DCT coefficients [89]. Westfeld and Pfitzmann [71] presume that the frequency for each pair of values (PoV) is not close to the mean of the PoV. According to the previous assumption, the chi-square test detects the existence of the secret data, because the JSteg embedding algorithm changes the frequency of DCT value $2k$ and $2k + 1$ to be almost similar [52].

Various improved versions have been introduced to avoid histogram symmetry between the two consecutive coefficients or increase the embedding capacity. For instance, the OutGuess algorithm aims to overcome the chi-square test by selecting random locations for embedding instead of using a sequential order [23]. Also, the F5 algorithm was developed by Westfeld to improve the embedding efficiency and capacity of

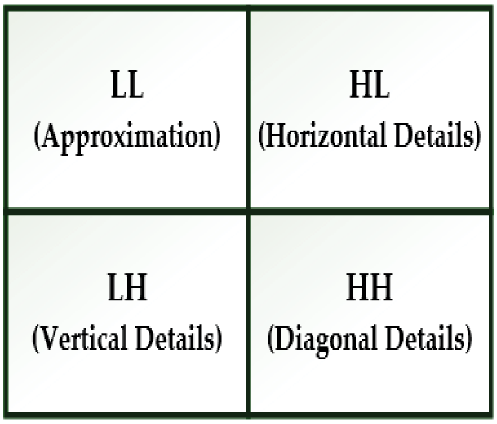
JPEG steganography. It extended another JPEG steganography method by adding permutation straddling characteristic and matrix encoding [90].

3.2.2.3 Wavelet Transform Domain

One of the most commonly used transforms is the discrete wavelet transform (DWT). It converts spatial domain to frequency domain. The wavelet transform is preferred over the cosine transform because it clearly divides the image into different frequency levels [91].

The wavelet transform is a more accurate model aspect of the HVS and requires less computational cost compared to DCT and FFT (Fourier transform). Generally, wavelet transform allows for the embedding of data in high frequency regions where HVS cannot distinguish modifications compared to uniform regions with low frequency.

As shown in Figures 3.11 and 3.12, wavelet transform is a mathematical function that divides an image into levels such as four sub-bands, which are low (LL), low-high (LH), high-low (HL) and high (HH) frequency sub-bands. DWT hides information in the complex region of the cover image because modifications in the edge regions are not detected by the human eye.



LL (Approximation)	HL (Horizontal Details)
LH (Vertical Details)	HH (Diagonal Details)

Figure 3.11 DWT sub-bands

Applying a two-dimensional DWT on an image, it is separated into the following coefficients matrices:

1. CA matrix (LL): It contains the low frequency coefficients which approximate the original image.
2. CH matrix (HL): It contains the high frequency coefficients which are presented in the horizontal details of the original image.
3. CV matrix (LH): It contains the high frequency coefficients which are presented in the vertical details of the original image.
4. CD matrix (HH): It contains the high frequency coefficients which are presented in the diagonal details of the original image.

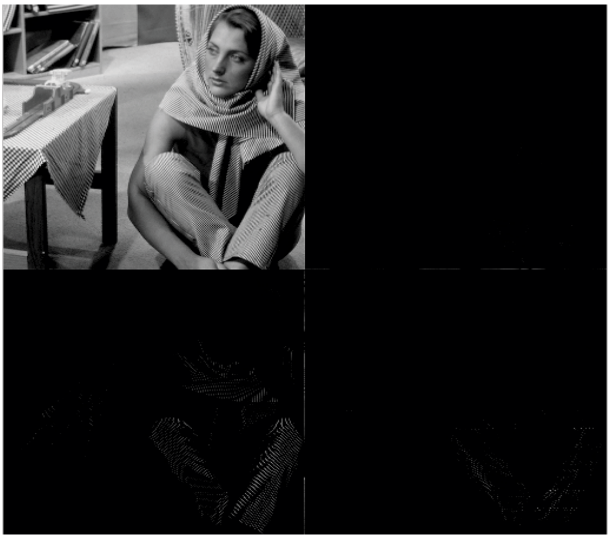


Figure 3.12 An example of the first level of DWT decomposition

Integer wavelet transform (IWT) maps an integer data set to another integer data set. In the case of DWT, the wavelet filters have floating point coefficients. Therefore, any truncations in the DWT coefficient values after concealing data will lead to the loss of the embedded data. In other words, the extraction of the original secret message becomes difficult. However, by introducing a wavelet transform that maps integers to integers, the resulting output can be described in integers without losing any information between forward and inverse transform. The LL sub-band of IWT is close to the original image compared to the LL sub-band of DWT as shown in Figures 3.12 and 3.13.

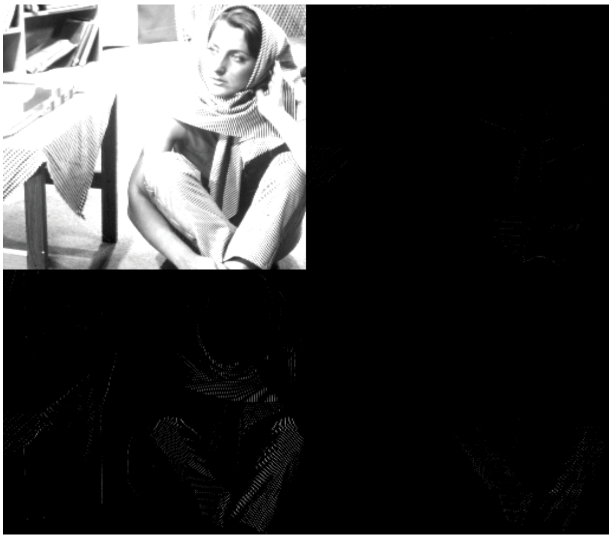


Figure 3.13 An example of the first level of IWT decomposition

3.3 State-of-the-art Steganographic Methods

There is a lot of work related to image steganography in the literature since it is a classic research topic. In this section, some state-of-the-art research on image steganography based on edge detection, coding theory and wavelet transform methods in recent years is reviewed.

3.3.1 Steganography Method Based on Edge Detection

The human visual system (HVS) is less sensitive to changes in high contrast areas compared to the smooth areas. Therefore, image steganography based on the edge detection method has attracted the attention of a large number of researchers. Embedding in the edge pixels aims to improve imperceptibility. Some proposed techniques, such as PVD, differentiate between the smooth and sharp regions without computing the actual edges in the image, while many steganography methods take advantage of the existing edge detection algorithms to compute the actual edges in the image.

Luo et al. introduced an adaptive LSB matching revisited (EALSBMR) method. It improves the detectability probability by integrating LSBMR and edge-based techniques.

It detects the edge regions by computing the difference between two consecutive pixels. A threshold value is used to select the embedding regions according to the length of the secret message. This scheme uses horizontal and vertical edges by dividing the image into blocks then rotating each block by a random angle. However, the relationship between vertical/horizontal pixels could be destroyed because of the rotation process [92].

The utilization of the traditional edge detection algorithms does not guarantee identifying the same edge sets between the cover and stego images. If the secret message is embedded in the edge pixels, it causes changes in the edge regions between the cover and stego images. Therefore, the extracted message will not be identical to the original one. Some of the current edge-based steganography algorithms propose a solutions to ensure the correct identification of edge pixels such as identifying the edge pixels based on the MSB or storing information about edge/non-edge pixels. Li et al. designed a spatial colour image steganography based on the Sobel algorithm. Sobel edge detection was performed on one of the R, G or B channels of the cover image. Embedding locations are chosen based on the largest number of gradients among the R, G and B planes. The LSB of corresponding pixels in different planes are altered to conceal data. Embedding capacity is improved by repeating these phases many times until the secret message is embedded. Finally, the separate planes are integrated to form the stego image [93].

Bassil proposed a colour image steganography that uses canny edge detection to identify the embedding location and LSB techniques to embed the message bits into the cover image. For each edge pixel, three least significant bits are replaced by the secret data bits. The number of edges is also adjusted by three parameters: the size of the Gaussian filter and the low and high threshold values. However, this scheme does not ensure an exact match between the cover and stego edge pixels [94].

Modi et al. applied canny edge detection to a grey-scale image where only the six most significant bits participate to form the edge map. The secret data is concealed in the least two significant bits of every edge pixel. The number of edges are chosen based on the length of the secret data. However, this method embeds the same bit numbers in every edge pixel without taking into consideration if it is a weak or strong edge. In addition to this, the secret data is embedded using the LSB method [95].

Chen et al. introduced a high embedding capacity steganography method based on the hybrid edge detector method. In order to increase the embedding capacity by finding a larger set of edges, the edges identified by the canny and fuzzy edge detector algorithms are combined to generate the final edge image. The edge image is then divided into blocks where each block has n-pixels. For each block, the first pixel (P_1) is used to save the status of the remaining $(n-1)$ pixels, either the edge or non-edge pixel. Finally, the LSB method is applied to embed x bits into the non-edge pixels and y bits into the edge pixels. However, this method creates an unwanted modification in the stego image because $(n-1)$ bits from the first pixel of each block has to be replaced [96].

In order to attain less distortion and higher embedding capacity, Tseng and Leng [97] extended [96] to a block-based design. Four cases of $[x, x+1]$, $[x, x+2]$, $[x, x+3]$, and $[x, x+4]$ are employed to determine the number of embedded bits in the non-edge and edge pixels. For each pair, the first and second elements indicate the bit numbers that can be hidden in the non-edge and edge pixels respectively. In the embedding process, the cover image is divided into non-overlapping blocks of six 4×4 and each block is segmented into four sub-blocks of size 2×2. The upper left sub-block contains the edge/non-edge information of the remaining three sub-blocks. The major limitation of [96] and [97] is that reserving a large space of the cover image to store information about the edge and none-edge pixels creates significant distortion to the quality of stego images. In [97], three-quarters of the whole cover image is utilized just to conceal the secret data.

Sun presented an image steganography based on edge detection and 2^k correction [98]. In the beginning, the secret image is compressed using Huffman coding to minimize the amount of embedded bits in the cover image. Canny edge detector is applied to identify the edge regions which are permuted before the embedding process to enhance the security. This scheme improves the imperceptibility by using a 2^k correction algorithm to reduce the difference between the cover and stego images. However, this method is not capable of identifying the same edge pixels in the cover and stego images. In addition, LSB method is used for embedding the secret data which is detected by most of the structural detectors.

Bai et al. proposed an image steganography based on LSB replacement and edge detection algorithms. To improve the embedding payload, the cover pixels are classified into edge and non-edge pixels using either the canny, Sobel or fuzzy edge detector methods. The LSB replacement method is then performed to hide x bits in the non-edge pixel and y bits in the edge pixel. The value of y is greater than x and y is in the range of 2 to 5 because the HVS can tolerate more changes in the sharp regions than the smooth regions. To obtain an exact match between the cover and stego edges, the edge detector method is applied on the three most significant bits and clears the five least significant bits. The modification rate of this method is 0.5 bpp [99], which is relatively high. Also, as mentioned earlier, the LSB method is vulnerable to statistical attack.

3.3.2 Steganography Based on Coding Theory

Improving the embedding efficiency, which is defined as the number of embedded bits per embedding change (introduced change), is one of the most essential requirements of steganography systems [100, 101]. Developing a steganography method based on coding theory has been considered by a number of researchers, as minimizing the amount of distortion in the stego image caused by the embedding process will improve the imperceptibility and increase the capability of resisting steganalysis [100]. Some of the traditional steganography methods, such as LSB, have a high modification rate. However, the utilization of coding theory enables a decrease in the modification rate [102].

Nowadays, error correction codes (ECCs) are employed to hide the secret data in an image and to retrieve the secret data from the modified image. Some of the famous ECCs are the Bose, Chaudhuri and Hocquenghem (BCH) [103, 104], Hamming [105, 106], Reed-Solomon [107, 108] and Syndrome-Trellis codes (STC) [109, 110].

The idea of employing error correction codes, which is also called matrix embedding, into steganography was firstly introduced by Crandall [102]. The objective of matrix embedding is to achieve a high embedding efficiency by reducing the difference between the cover and stego images. In [102], the XOR operation is utilized to embed two bits of the secret message into a block of three pixels. The maximum embedding rate of this method is 66.67% and the modification rate is 25%.

In [90], F5 is the first implementation of JPEG steganographic scheme based on matrix coding, which resorts to the Hamming codes to minimize the change on the quantized discrete cosine transform (DCT) coefficients of the cover image. Instead of substituting the LSB of the DCT coefficient with the secret bits, it utilizes $(1, n, n - k)$ Hamming code to conceal k bits of secret message into $2^k - 1$ cover bits by changing at most one bit only. Consequently, this method has a limited embedding capacity, for instance when the $(1, 7, 4)$ Hamming code is used, the F5 scheme only hides three secret bits into a block of seven pixels. Also, the computational cost of the Hamming code is high, as matrix multiplication is required [90].

Hou et al. introduced an approach called tree-based parity check (TBPC) that uses a tree structure to enhance the embedding efficiency by reducing deformation on the cover object. The authors proposed a strategy of majority voting for TBPC and argued that this strategy inherited the efficiency of the TBPC method and produced the least deformation. Similar to some of the other coding methods, the drawback of this method is the high computational cost, especially for trees that have multiple levels. The method can hide 2^n bits in a binary tree of n levels. For example, if the binary tree has two levels, then it hides four secret bits into seven pixels [111].

Mstafa et al. developed a video steganography using Hamming code. This method uses an uncompressed video and divides video into frames. The colour space of each frame is converted into YUV components. Each four secret bits are encoded into code of seven bits using (7,4) hamming code. An XOR operation is then performed between the encoded data (four bits of message and three bits of parity) and seven bits of random numbers. Finally, this data is embedded into YUV components [105].

However, Hamming code has been utilized for error correction detection codes and the embedding payload of this method is low at around 0.57 bpp.

Bai and Chang presented a data-hiding scheme based on Hamming code to embed the secret message into the absolute moment block truncation coding (AMBTC) compressed image. The embedding process has two phases. In the first phase, (7,4) Hamming code is applied on the low and high mean values of each compressed block to embed three secret bits and one extra secret bit can be embedded based on the difference between the low and high mean values. In the second phase, another three secret bits are embedded in the AMBTC bitmap. In each phase, the alteration of the AMBTC code can be minimized because only one bit is modified [106]. However, the bit flipping in the AMBTC bitmap might leave noticeable distortions, in particular if the block contains edges [112].

Feng et al. [113] introduced a binary image steganographic technique to reduce the embedding distortion on the texture. The cover vector is generated by splitting the image into superpixels and syndrome trellis code (STC) is utilized to improve the embedding efficiency. The complement, rotation and mirroring-invariant local texture patterns are extracted from the binary image. The changes in complement, rotation and mirroring-invariant local texture pattern distortion show a strong relationship with the detectability of the embedding distortion. The flipping distortion measurement is set with the weighted sum of complement, rotation and mirroring-invariant local texture pattern changes, where the weight is empirically assigned according to the discrimination power of the complement, rotation and mirroring-invariant local texture patterns' histogram.

3.3.3 Steganography Based on Wavelet Transform

There are various steganography schemes based on wavelet transform domain. Wavelet transform offers the opportunity to embed the secret data in regions that the HVS is less sensitive to modifications happening due to the embedding process. The wavelet transform domain provides better imperceptibility in compatibility with the HVS and with higher robustness against image processing attacks.

Ghasemi et al. proposed an image steganography technique based on wavelet transform and genetic algorithm. This method aims to enhance imperceptibility by reducing the difference between the cover and stego images as well as improving the robustness of steganography methods. The cover image was divided into non-overlapping blocks of 4 × 4 pixels and then the frequency representation is computed using 2-D Haar DWT to obtain LL1, LH1, HL1 and HH1 sub-bands. A genetic algorithm-based mapping function is utilized to hide the secret data in the DWT coefficients. The optimal pixel adjustment process is performed after embedding the message [114].

Bhattacharyya and Sanyal presented a steganography method based on discrete wavelet transform difference modulation (DWTDM). The DWTDM algorithm applied DWT to transform the cover image into four sub-bands. Then, each sub-band is divided into block of size 8 × 8 coefficients. After that, four seed pixels are selected from each block to embed four bytes within every block. The DWTDM was developed to overcome limitations in the transform domain techniques. It increases the capacity of embedded data and does not make any visual changes in the cover image [115].

Parul et al. designed a new scheme for image steganography using DWT. Firstly, the cover is separated into three channels (R, G and B), then DWT is applied to each channel before the secret image is modified using the Arnold transform and every colour component of the changed secret images is separated. The secret image is embedded into the high frequency (HL, HH and LH) sub-bands. Finally, the inverse DWT is performed to produce the stego image. This method achieves a good result in term of PSNR and embedding capacity [116].

Reddy and Kumar designed a new information security method that integrates LSB steganography and advanced encryption standard (AES) cryptography to protect the data transmission over unsecured or public networks. The secret data is encrypted using the AES algorithm, then the image is divided into four sub-bands (LL, LH, HL and HH) using the wavelet transformation function. The encrypted text is embedded into the LSB of the LL sub-band. The inverse wavelet transform is applied and the resultant stego image is transmitted to the receiver [117].

Hemalatha et al. introduced a colour image steganography method based on DWT and integer wavelet transform (IWT). It begins by converting the colour cover image into YCbCr colour space. Then, the secret image and Cr component are decomposed into DWT sub-bands. The LL sub-band of the Cr and secret images are divided into non-overlapping blocks of size (2×2). Then, each block (s_i) of the LL sub-band of the secret image is compared with all the LL sub-band blocks (c_i) of the Cr component and the location of the block c_i that has the minimum root mean square error (RMSE) is saved to generate the secret key. The generated key is first compressed using run length encoded (RLE) and then embedded in the LSB of the higher frequency of IWT coefficients of the Cr component. This method hides the generated secret key instead of the actual secret image to improve the security and capacity. A very similar work to Hemalatha et al.'s scheme has been recently proposed in [118], where only the LL sub-band of the secret image is embedded into different sub-bands of the colour cover image [119].

It is clear that most of the proposed methods aim to use wavelet transform to improve security and imperceptibility, in harmony with the HVS and with higher robustness against signal processing attacks. Also, the high frequency sub-bands are employed to carry the secret data.

3.4 Steganography for Medical Image Security

Digital medical images have become an essential part of diagnosis and treatment. However, secure storage processing and analysis of medical images that do not violate the Code of Ethics for Health information Professionals are vital. The Digital Imaging and Communication In Medicine (DICOM) is the international standard for handling, storing, printing and transmitting medical imaging and related information [120]. It defines the format for medical images that can be exchanged with the data necessary for clinical use. However, DICOM was initially introduced without considering network security or data protection [121–123]. A DICOM encoding method was later initiated and has been the only data protection for DICOM for nearly 20 years [124].

3.4.1 Digital Medical Image Steganography

The main aim of utilizing digital steganography for medical images is to increase the security, confidentiality and integrity for both data records and medical images. Medical image steganography is considered a special case of image steganography where medical images have special requirements [125].

In the health care system, the remote exchange of medical images and patient records between clinics has become part of the daily routine. Image steganography is employed to protect the EPR's confidentiality without affecting the medical image quality. It hides the EPR and diagnosis report in their medical image to solve the authentication problem by providing a link between the patient's information and their image [126].

A steganography method consists of two main procedures: embedding and extraction procedures. In the embedding process, the EPR is embedded into the medical image without introducing noticeable distortion on the stego image. The extraction process, on the other hand, is responsible for retrieving the patient's information from the stego image [127].

3.4.1.1 Steganography Advantages

Over the last few years, there has been increasing attention paid to enhancing the privacy of patient information in medical databases by considering steganography schemes as a solution for hiding EPR in the patients' images. This interest has appeared within several steganography techniques that are designed to fit with the requirements of medical images.

Steganography methods are utilized in the health care system for many reasons, These include [128–132]:

- Privacy and confidentiality: Data privacy is a significant issue when medical images and data are exchanged between hospitals over unsecured public networks. To maintain digital medical image and confidential patients' information during the data transmission, steganography coverts the communication channels to avoid drawing the suspicion of an eavesdropper.

- Security: The main characteristic of steganography is to be statistically undiscovered by intruders or unintended users. In steganography, intruders aim to discover the existence of a secret message without having to exactly retrieve the embedded message. To be more precise, steganography security is determined by the supposition that the intruder is not capable of demonstrating whether the cover medium contains secret data or not [133].
- Saving memory and cost: Computational and memory cost are essential requirements for evaluating any information system. Since medical system databases generally have a lot of data to keep, it is reasonable to use procedures that keep the necessary data but require minimum memory space. In the Medical Information System (MIS), the required storage for patient's record can be minimized by hiding EPR within the medical image. However, the computational cost of information-hiding methods should be acceptable compared to other information security methods.
- Availability: Availability is the ability to ensure that authorized users have access to information system information and services at any authorized time.

3.4.1.2 Limitations of Traditional Medical Security Techniques

It is well known that data security in medical systems has been urgently demanded. Different information security techniques are employed in MIS such as virtual private networks, firewalls, digital signatures and encryption methods [123, 128, 134–136]. Nevertheless, traditional security methods have some drawbacks which create a necessarily demand for finding an alternative scheme to protect medical data. These are discussed below.

- Firewall: A firewall is a network security system that is used to control the incoming and outgoing network which is available as software or hardware. It is placed between a trusted internal network and another network. It shields the internal network from interruption and malicious software coming from the external network. An efficient firewall requires three standards: (a) it must act as a door through which all traffic must pass (incoming and outgoing), (b) it must permit only approved traffic to access, and (c) it must be invulnerable to penetration. However, a firewall is not guaranteed to protect an organization network without any other assistance protection. For example, if a firewall is not suitably configured, then it may lead to a denial-of-service (DoS) attack where services are temporarily unavailable to authorized users [134].

- Virtual private network (VPN): A VPN is a network that transports data in a secure way over a public network. It is combined with encryption, authentication and tunnelling to accomplish safe transmission. Various implementations of VPN are available, such as Point-to-Point Tunnelling Protocol (PPTP), Layer 2 Tunnelling Protocol (L2TP), Internet Protocol Security (IPSec), and SOCKS. In order to have access to a VPN, users should have a unique identification and password. In a VPN, the shortage of quality of service (QoS) management over the Internet may lead to packet loss during transmission. Combining a VPN and a firewall do not guarantee protection of the data as it may change before/after any communication takes place. These weakness points mean that different and alternative methods must be found in order to provide secure transmission [134].
- Encryption: Encryption techniques have been approved in different sectors such as the health care system to provide the confidentiality, access control, integrity and privacy of digital data. The encryption process depends on mathematical operations to hide the meaning of the message. In order to prevent intruders from understanding the meaning, encryption methods transform the secret message (plaintext) to an unreadable form (ciphertext). Hence, while people can recognize the existence of a message, it cannot be understood without decryption [137].

  Encryption techniques can be classified into two categories: symmetric (e.g. data encryption standard (DES) and advanced encryption standard (AES)) and asymmetric methods (e.g. (Rivest, Shamir, Adleman) RSA). In a symmetric encryption method, the sender and receiver use one private key for encrypting the plaintext and decrypting the ciphertext. In comparison, asymmetric encryption is a cryptography method in which every user has two different keys (public and private keys) where each key pair is mathematically related. Asymmetric encryption can be employed either for authentication or data secrecy purposes. For authentication, a message is encrypted by the private key of the sender and can only be decrypted by the related public key of the sender, while for data secrecy purpose, a message is encrypted by the public key of the receiver and can only be decrypted by the private key of the sender [137].

Asymmetric methods are preferable because it is not necessary to share the private key with someone else. However, symmetric methods are faster and more common than asymmetric methods. The limitations of using encryption methods are the high computational cost and that the transmission of encrypted text can stimulate intruders' attention, and they may attempt to decrypt it. In general, the strength of encryption method depends on the key length and number of keys [137].

- Hash function: A cryptographic hash function is used to prevent or discover modification during data transmission. It is used to prove the integrity of secret data. The input of the cryptographic hash function is a block of random length, where the output length, also known as hash value, is fixed. In other words, the size of the secret message does not affect the size of the hash value [138, 139]. The hash value is the checksum of the secret data, where the checksum represents the fingerprint of the secret data. The perfect cryptographic hash function has four fundamental characteristics [138]: (a) hash value for any block must be easily computed, (b) it is hard to discover a block that has a given hash, (c) it is hard to alter a block without altering its hash value, and (d) each block message has a unique hash value. Therefore, it is hard to have two different blocks with the same hash value.

  Message digest 4 (MD4), Message digest 5 (MD5) and Secure Hash Algorithm-1 (SHA-1) are the most common known hash algorithms implementation. For example, MD5 has a random input size to produce a hash value of 128-bits. Most of the current cryptographic hash functions methods are susceptible against coincidental alternations.

3.4.2 Region of Interest (ROI)

Medical images contain a pivotal and very significant part which is responsible for diagnosis, i.e., the Region of Interest (ROI). In order not to compromise the diagnosis, the significant part of the medical image, the ROI, should be spared from any modification. Since small changes in the ROI may cause mistakes in diagnosis, it is quite important that secret data be embedded in a region of non-interest in (RONI) [140]. In information-hiding methods, a ROI might be manually or automatically identified either in regular or irregular shapes such as ellipses, rectangles or polygons.

3.4.3 Information Hiding Methods for Medical Images

Many methods have been implemented in the area of medical image information-hiding for different objectives. Al-Qershi classified medical image watermarking schemes into three groups according to their objective: authentication, data-hiding and both authentication and data-hiding [141]. Nevas et. al. suggested three key requirements for EPR data-hiding and transmission: (1) the extraction process of EPR should be blind, (2) EPR data should be extracted with a zero bit error rate (BER) at the receiver side and (3) imperceptibility should not be compromised for any reason. For additional confidentiality, encryption of the EPR can also be used in EPR data-hiding [142]. The required criteria of medical image information-hiding algorithms are clarified by these requirements, for example, information-hiding methods should be blind and invisible. A literature summary that outlines the important aspects of these methods is presented in Table 3.4. It can be observed from Table 3.4 that most of the proposed techniques are blind and combined with cryptography to add more security, while the embedding domains vary between the spatial and transform depending on the objectives behind introducing the method.

Most of the information-hiding methods use LSB because of its simplicity, acceptable distortion of the produced stego images and high embedding rate. However, the LSB technique has many weaknesses [152, 153]. Among the information-hiding methods that utilize LSB are the methods described in [143–146, 150]. Zhou et al. [143] introduced a lossless data embedding scheme for validating the authenticity and integrity of mammography images. This method embeds the encrypted digital signature and patients' information into random pixels of mamograph digital images using the LSB method.

Table 3.4 Literature review for various information-hiding methods

Method	**Image Modality**	**Embedding Domain**	**Embedding Technique**
Zhou [143]	Mammography image (IM)	Spatial	LSB of random pixels
Chao [144]	Hospital mark image	DCT	LSB
Navas [131]	MRI	IWT	LSB
Ali [145]	IRM Echo graphics	Spatial	LSB
Nagaraju[146]	CT MRI US	Spatial	2-LSB
Rahimi [147]	CT MRI	SVD Contourlet Transform	New method
Lou [148]	–	Spatial	Difference Expansion
Memon [149]	CT MRI X-ray US	IWT	New hybrid
Memon [150]	CT	Spatial	LSB

Table 3.4 Literature review for various information-hiding methods

Qershi [151]	US	DWT	Difference Expansion
Method	**Cryptography**	**Secret Data**	**Embedding Region**
Zhou [143]	DES RSA	Patient's data Digital signature	Random pixel in segmented image
Chao [144]	New proposed method	EPR ECG Digital signature	PRNG pixels
Navas [131]	New proposed method	EPR	RONI
Ali [145]	SHA-1	Patient's data Medical Diagonstic	Pixels detected by Harris Corner
Nagaraju[146]	New proposed method	ECG Patient's Info	Whole image
Rahimi [147]	No	Patient's data Signature Watermark	ROI RONI
Lou [148]	No	–	Whole image
Memon [149]	K⊕W	Patient's data Doctor's code LSB of ROI	ROI RONI
Memon [150]	MD5	Patient's data Message Hospital logo Authentication code	RONI

Table 3.4 Literature review for various information-hiding methods

Qershi [151]	MD5	Patient's data Hash ROI ROI embedding map	ROI RONI

Method	**Reversibility**	**Extraction**	**Embedding Rate**
Zhou [143]	Irreversible	Blind	6720 bits
Chao [144]	Irreversible	Non-blind	–
Navas [131]	Reversible	Blind	3400 characters
Ali [145]	Irreversible	Blind	1700 bits
Nagaraju[146]	Irreversible	Blind	(0.04 − 0.97)%
Rahimi [147]	Irreversible	Blind	2010 bits
Lou [148]	Reversible	Blind	Up to 134,898 bits
Memon [149]	Reversible	Blind	32 char doctor code 96 char patient's info 1st LSB plane ROI
Memon [150]	Reversible	Blind	(3528 − 23184) bits
Qershi [151]	Reversible	Blind	10 KB

Method	**Image Quality**	**Objective**	
Zhou [143]	–	Authentication Integrity	
Chao [144]	(33.47 − 42.62) dB	Authentication Integrity Confidentiality	

Table 3.4 Literature review for various information-hiding methods

Navas [131]	44 dB	Data-hiding
Ali [145]	(35-60)dB (50 − 10)% JPEG compression	Integrity Confidentiality
Nagaraju[146]	(70 − 40)dB	Data-hiding
Rahimi [147]	52.2 dB	Data-hiding Integrity
Lou [148]	(21.59 − 48.86) dB	Data-hiding Copyright
Memon [149]	wPSNR (58.44 − 60.94) dB	Authentication Security Confidentiality Integrity Control
Memon [150]	(63.98 − 55.6) dB	Authentication
Qershi [151]	41.25 dB	Authentication Data-hiding

Chao et al. presented a protected information-concealing procedure to generate patients' electronic medical records (EMR) and agent-EMR ciphertext to ensure the confidentiality of patients' EMRs stored in the healthcare database. The method is based on the bipolar multiple-base transformation to permit a mixed of EPR information to be concealed inside the same mark image. This scheme guarantees that only authorized users can gain access to the EMR [144]. However, the extraction process of the watermark needs the original image, a fact that eliminates the value of this method in practice.

Rahimi and Rabbani introduced a dual and blind watermarking technique which embeds the watermark bits in the singular value vectors within the low pass sub-bands in the contourlet transform domain of DICOM images. This method automatically identifies a rectangular ROI and hides a watermark with a different embedding strength in the ROI and RONI, where the RONI is the background region (the black area around the medical information). This technique is tested using CT and MR images. However, the embedding rate of this method is limited because of the way a ROI is selected, where the ROI selected by physician may increase the embedding capacity [147].

Ali et al. proposed a spatial medical image watermarking technique to maintain the integrity of medical images and protect the confidentiality of patients' information during transmission. The digital signature of the hospital data is generated using a secure hash algorithm (SHA1), then it is concatenated to the patients' information. The secret data is embedded in edge pixels using the LSB method [145]. The quality of the watermarked medical images is acceptable but, according to [154], the technique is very fragile and provides a very little security.

Memon et al. developed a hybrid watermarking scheme which hides a robust watermark in the RONI. In this scheme, the medical image is segmented into ROI and RONI. After that, a fragile watermark is hidden into ROI using the LSB technique. The RONI is distributed into blocks of size $N \times N$ and then a location map is generated. A robust watermark is embedded in the RONI coefficients. However, the time complexity of this method is high because of the required calculation to generate the location map [149].

Navas et al. introduced a blind and reversible data-hiding scheme for telemedicine applications that depends on integer wavelet transform (IWT). The ROI is manually identified as a rectangle shape. In order to obviate misdiagnosis, encrypted EPR is embedded in the RONI and the ROI is stored without any noise [131]. However, this method can hide at most 3,400 characters and the computational cost is high.

Nagaraju et al. presented a spatial information-hiding method for medical images. To improve the security of the proposed method, patient information and electrocardiogram (ECG) signals are encrypted before being concealed inside the cover image, then this encrypted data is embedded using the LSB method [146].

The main limitation of the previously mentioned methods is the direct implementation of LSB steganography, which is known for its vulnerability to some steganalysis methods.

In addition to this, most of these methods do not differentiate between the ROI and RONI when embedding the secret message.

Lou et al. introduced a lossless multiple-layer spatial data-hiding scheme for medical images based on pixel-value differencing expansion. This method utilized a reduced difference expansion scheme to conceal the bit stream in the LSBs of the expanded differences. In order to provide a high embedding rate and maintain good quality, the reduced difference expansion method is used [148].

Raul et al. proposed a data-hiding method for radiological medical images which used image moment theory. In order to minimize the size of data to be hidden, the Huffman algorithm was utilized to compress DICOM data. For more security, the compressed data is encrypted using the RC4 method. Finally, secret data bits are embedded in selection pixels with low homogeneity, which can be obtained by scanning the cover image in spiral way using the central pixel [155]. The drawback of Rual et al.'s method is the use of a static key for compression during the embedding and extraction phases.

Bremnavas et al. introduced medical steganography to hide patients' information in text form and image form into the cover images using two different algorithms. The medical details record is converted to UTF format which is then embedded using a LSB method. The medical image is again encrypted using chaos algorithms [156].

Prabakaran et al. introduced a multi secure and robustness steganography technique for medical images. In order to protect MRI images, IWT is performed to embed the secret data bits into a single container image. A dummy container is acquired by applying the flip left operation on the container image, then the Arnold transform is performed on the patient's medical diagnosis image to get a scrambled secret image. The scrambled secret image is hidden in the dummy container and inverse IWT is applied to get the stego image [157]. The computational cost for the embedding and extraction procedures is high.

Tian
(DE) [158]. This has been recently extended in [141, 148, 151] for medical images. Tian's method divides the cover image into non-overlapping blocks of two consecutive pixels. The secret bits are embedded using the difference expansion of each block.

Differences are classified into three groups, while the secret data is embedded into only two groups. Additional information concerning the location map is required to specify which pairs are used. The difference expansion method has an embedding capacity of 0.5 bits per pixel. The limitation of the difference expansion method is the limited embedding rate [158].

3.5 Summary

This chapter provides a brief literature review of the basic steganography methods. The steganography methods can be classified into spatial and transform depending on the embedding domain. The spatial steganography techniques, such as LSB and PVD, have a high embedding capacity and low computational cost compared to the transform steganography techniques. On the other hand, transform domain methods are robust against attacks.

Various transformation algorithms can be used to develop transform domain steganography techniques, such as DCT, DFT, DWT and IWT. Wavelet transform is more popular than DCT and DFT because it decomposes the image into different frequency sub-bands that match the human visual system and offer a high embedding capacity.

The fact that the modification in high contrast regions is less detectable by the human visual system compared to the smooth regions has stimulated researchers to develop image steganography based on edge detection. However, this needs to be applied with care, as the resultant stego image could have edges that are not 100% identical with those of the original cover image.

In addition to this, coding theory algorithms, which are also called matrix embedding, have been utilized in the embedding process in order to improve the embedding efficiency. However, the computational complexity of the coding theory algorithms, such as syndrome-trellis code, is high, especially for high dimensional matrices.

The last part of the chapter discussed steganography for medical images. The main objective of implementing medical image steganography is to provide security, confidentiality and integrity for the medical image and patient records. Steganography for medical image systems is considered as a special case where the ROI should be preserved from any modification to avoid any misdiagnosis.

CHAPTER 4

Image Segmentation Background

This chapter provides a review of the main techniques of image segmentation. Medical image processing and magnetic resonance image are explained. This chapter also discusses state-of-the-art machine learning based brain image segmentation methods.

4.1 Introduction

Image segmentation is considered one of the most significant tools for carrying information that helps in understanding and interpreting images. The interpreted information can be employed for different applications, such as the diagnosis of tumour tissue and object identification and recognition. Image segmentation has been extensively applied in numerous areas such agriculture, medicine and forensics. Figure 4.1 illustrates the level of image segmentation in the image engineering layer. It is the major step for the image analysis (middle layer), where the accuracy of segmentation process has a great impact on the whole process [159].

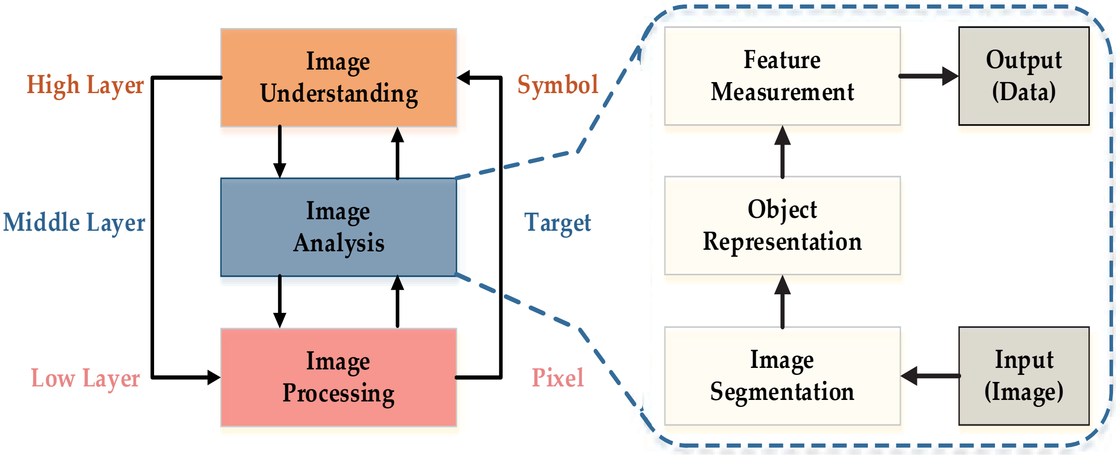


Figure 4.1 Image engineering layer

In general, image segmentation is the process of separating the image (I) into different non-overlapping regions $(R_1, R_2, \cdots, R_n)$ that have a high degree of similarity based on specific criteria; for instance; image luminance, shape and colour components. To be more precise, image segmentation is an essential step in image analysis where the segmentation output affects the accuracy for the entire process. The segmentation process can be formally defined as follows: if $P()$ is a homogeneity predicate defined on groups of connected pixels, then segmentation is a partition of the set I into connected subsets $(R_1, R_2, \cdots, R_n)$ such that $I = \bigcup_{i=1}^{n} R_i$ with $R_i \cap R_j = \Phi$ and $(i \neq j)$. The uniformity predicate $P(R_i)$ is true for all regions R_i and $P(R_i \cap R_j)$ is false when i and j are not equal and R_i and R_j are adjacent [159, 160].

Medical image segmentation has a significant function in analyzing anatomical structure and tissue types. For instance, in magnetic resonance (MR) brain image analysis, the segmentation process is employed for measuring and visualizing the brain's anatomical structure, analyzing brain changes, delineating pathological regions and for surgical planning and image-guided interventions [161]. The objective of segmentation is to simplify the actual illustration of an image into another format which is easier to understand and analyze. Basically, image segmentation is useful for defining boundaries between the brain tissues as well as assigning a unique label to each pixel in the image.

The assigned labels facilitate the classification and grouping of brain tissues where pixels with a similar label represent a certain computed characteristic such as texture, intensity, shape or colour [9].

In recent years, numerous segmentation methods have been introduced to divide the brain image into three tissues with different degrees of accuracy and complexity. However, these methods suffer from various challenging issues such as the development of a common approach that is applicable to all image types [162]. The quality of the image plays an essential role in producing accurate segmentation. However, MR images obtained from different MRI scanners are prone to image intensity-related artefacts, such as image noise or the bias field effect, which are highly dependent on the magnetic field strength. Because of the aforementioned reasons, there is no universally accepted technique for designing MR image segmentation. Accordingly, automated image segmentation is not widely accepted by clinicians [161, 163].

This chapter is organized as follows. Sections 4.2 and 4.3 present classification of segmentation methods based on human interaction and the various automated techniques respectively. Medical image processing in presented in Section 4.4. Magnetic resonance brain image processing is introduced in Section 4.5. MR Brain image segmentation methods are reviewed in Section 4.6. Finally, the summary is given in Section 4.7.

4.2 Classification of Segmentation Methods Based on Human Interaction

Image segmentation techniques can be categorized into three different categories based on the human interaction level: manual, semi and fully automatic.

4.2.1 Manual Segmentation

In manual segmentation, the boundaries of the objects and region of interest are identified and labelled by a human operator [161]. In this method, the human experts use the information presented in the image and add their knowledge to obtain a more accurate segmentation result. Manual segmentation is performed using customized software tools with advanced graphical user interfaces (GUI) to simplify the delineation of regions of interest and display the image [164]. However, manual segmentation is a time-consuming, complex procedure and is prone to errors [161, 165]. For instance to determine the tumour area in a MR brain image, MRI scanners produce various two-dimensional slices and the expert user analyzes the dataset slice by slice to select the best illustrative slice which segments the ROI accurately [166]. Also, in order to delineate the ROI a specialist trained user who is expert in brain structure, such as a radiologist or anatomist, is required. Otherwise, the segmentation results will not be accurate [164].

The drawing of tumour region procedure slice by slice is tedious and can lead to creating jagged images due to limits in the expert rater's view. Thus, the resultant images are not ideal where they demonstrate a stripping effect [167]. Obviously, manual segmentation is operator dependent and suffers from high intra and inter-observer variability [168].

In [169], the delineation of the MRI scans by different expert physicians was evaluated to check if manual segmentation is operator dependent. The variation was as reported 20%±15% within the same physician (intra-rater); estimated through repeating the task of diagnosing a brain tumour and 28%±12% between physicians (inter-rater). Figure 4.2 shows an example of inter-rater inconstancy, where four different specialists played out a manual division of a glioma on a similar slice and patient. The segmentation result of every expert presents outstanding contrasts [1].

Notwithstanding the intra and inter-rater changeability, manual segmentation is usually utilized as a ground truth to qualitatively and quantitatively evaluate the segmentation outputs of the automated algorithms [161].

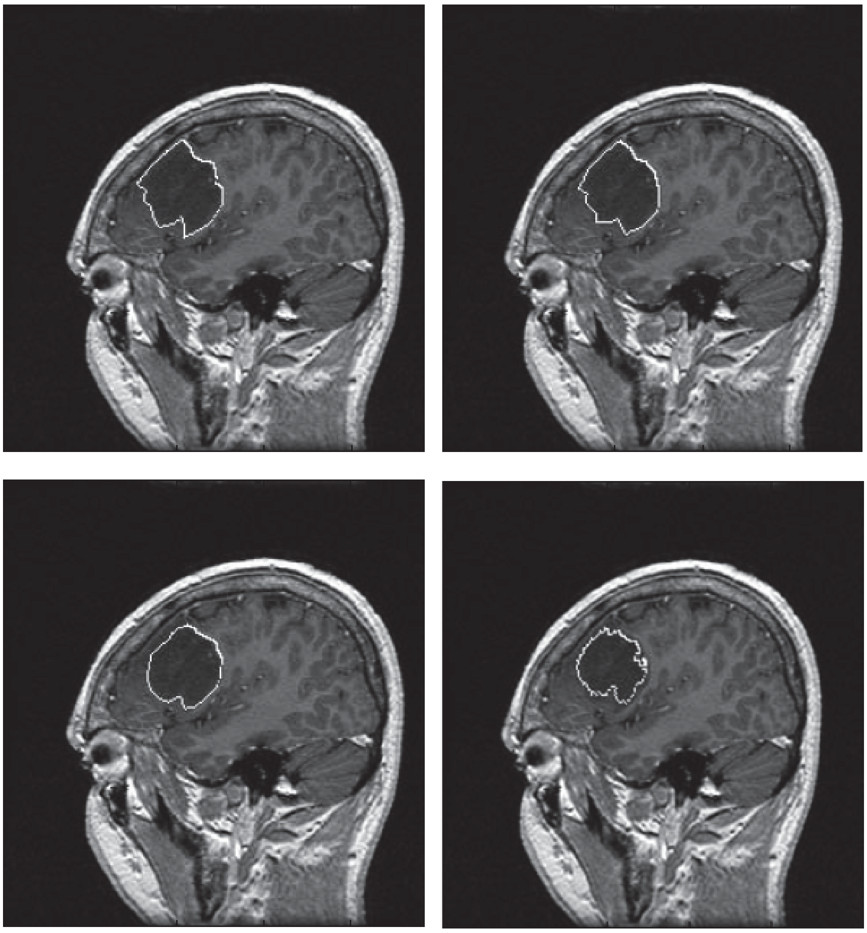


Figure 4.2 Manual segmentation by four different experts manual segmentation by four different experts [1]

4.2.2 Semi-automatic Segmentation

The semi-automatic segmentation technique was designed to engage human interaction while still relying on a computer to perform the segmentation process. It is an intermediate between the manual and fully-automatic segmentation approaches. It might be the ideal approach when a fully-automatic method is unavailable and manual segmentation will be a time-consuming process [170].

In the semi-automatic segmentation method, the involvement of the user is frequently required to input some parameters, verify the efficiency of the result and adjust the segmentation result manually [171]. Recent research has been directed towards developing semi-automatic approaches with the aim of minimizing user interaction. The computational, interactive and user interface are the major elements of an interactive segmentation method. The computational part correlates to one or more fragments of the program that are capable of producing an outline of the object of interest if some parameters are provided. The interactive part mediates information between the user and the computational part by converting the outcome that is produced by the computational part into user visual feedback and the data input entered by the user is converted into program parameters. The output and input devices are responsible for the interaction between the computer and the user through the user interface. The visual information displayed on the screen is analyzed by the user and they react accordingly, providing feedback for the computation [171].

The user interaction in segmentation is classified into three categories [172]:

(1) Initialization: The input of arguments or parameters, image pre-processing to improve the quality and complexity of the image data are evaluated for improving decision-making or the user chooses the object that has to be processed from the first segment of a data set or from a three-dimensional image.

(2) Intervention: Running the process constantly or intermittently towards a suitable output, giving feedback on the outcome of the data from the process, stopping the process in between if undesirable results are attained to correct the mistakes, and then recommencing the process.

(3) Evaluation: Assessing the final outcome of the process to decide if it is satisfactory. If the results are not satisfactory, the process is replicated after modifying the parameters; therefore, the results are altered.

The semi-automatic method employs various strategies for incorporating the computer and the user's expertise. Therefore, the result from employing this method is dependent on two factors: the strategy and the computation. These strategies vary based on the human interaction; that is, whether it will be in the initialization of the segmentation process, keeping the user in control during the whole process, or improving the concept of interaction by including intelligent behaviour. However, like manual segmentation, semi-automatic segmentation faces the limitation of variations between different expert users and within the performance of same user [164].

4.2.3 Fully Automatic Segmentation

The fully-automatic segmentation method refers to the process where the computer identifies the objects without any human interaction to initialize the input parameters or manually correct the segmentation result. In other words, the user just supplies the images that need to be segmented. Artificial intelligence and prior knowledge such as the noise level, appearance and spatial distribution are incorporated into the fully-automatic algorithms. However, it is not easy to develop a fully-automatic method because of image variation and complexity [164].

Recently, there has been increased interest in employing machine learning in the fully-automatic segmentation method to simulate the intelligence of humans to learn effectively. In spite of this, implementing a substantially accurate automatic method is still a challenging task. This can be understood when you consider that humans exploit advanced visual processing and integrate specialized knowledge to segment the image. For instance, the properties of brain structure can be focused on the MR image of the head, which is fairly predictable because the brain is well quantified structurally and the behaviour of various tissues in different MR modalities is well developed. Moreover, there is no sequential component and the brain stays invariable, so, as a result, there is no benefit of being able to visually track objects with time. However, because the human perspective is unable to utilize three-dimensional information in the segmentation process and analyze the data as a sequence of two-dimensional slices, there is a benefit of using a fully-automatic rather than a manual segmentation process [167].

In order to develop a robust automatic segmentation approach, image properties such as size, shape and appearance can be utilized to guide the segmentation method. This knowledge may be integrated into the segmentation model in various ways such as initial conditions, conditions on the model shape parameters or data constraints [164].

4.3 Techniques of Image Segmentation

Different techniques have been proposed to divide the image into non-overlapping regions. However, there is not any single technique that can be considered appropriate for all of the applications and image types [173].

Image segmentation techniques separate the images based on either intensity value similarity (region based) or discontinuity (boundary based) criteria to define the region's border or interior. The segmentation techniques based on similarity divide the image into regions by grouping the similar intensities in one region, while discontinuity criteria aim to identify the isolated values based on the sudden variation in the intensity values such as edges [174, 175]. Figure 4.3 shows three different categories of image segmentation techniques.

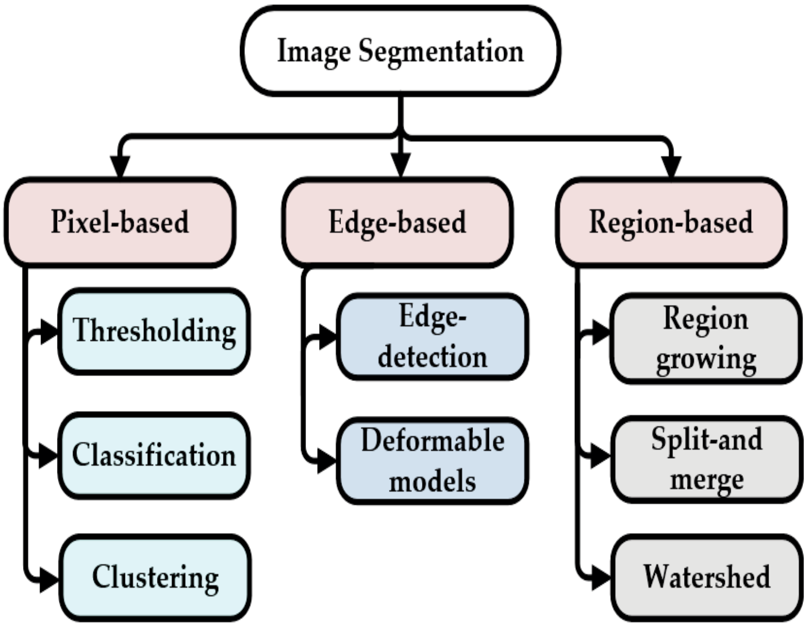
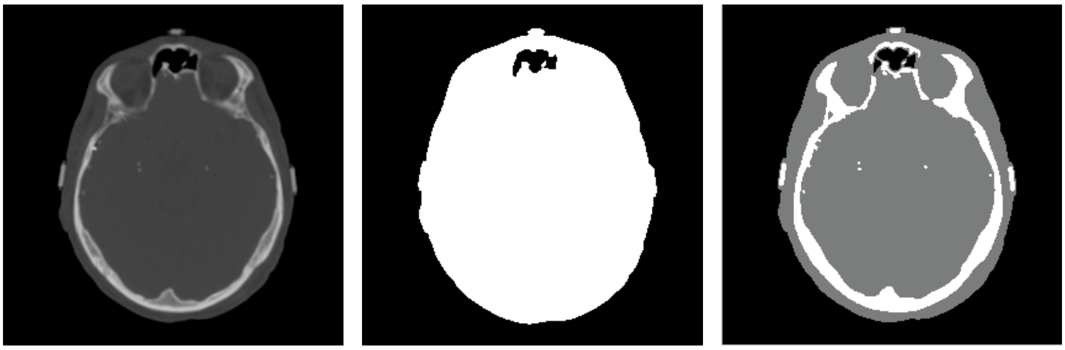


Figure 4.3 Classification of image segmentation techniques

4.3.1 Thresholding-based Segmentation

Thresholding is the most uncomplicated and fast segmentation method that divides the image into regions based on the intensity level. It is assumed that the pixels belonging to a specific range of intensity levels form one region and the rest of the pixels form the other region. Threshold-based methods can be classified into global and local. In global thresholding, the image is only separated into object and background regions, while the local thresholding divides the image into more than two objects [161, 176].



(a) (b) (c)

Figure 4.4 (a) Original image, (b) segmented image using single threshold value and (c) segmented image using multiple threshold values

- Global thresholding: The image histogram represents a binary pattern, in which the pixels in one region have similar intensity values. The easiest way to segment the image is to separate dark and light regions using a single threshold value. Global thresholding constructs a binary image from the grey-scale image by converting all pixels below the single threshold value to zero and the rest into one as shown in Eq 4.1 [176]. Figure 4.4b shows the result of the global thresholding. It segments the image into two regions: foreground and background.

$$g_{(x,y)} = \begin{cases} 1 & \text{if } f_{(x,y)} \geq Th \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

  where $g_{(x,y)}$ is the segmented binary image, $f_{(x,y)}$ is the original pixel value and Th is the threshold value.

  It is possible to achieve an accurate segmentation using the global thresholding method if the object has uniform intensity or the contrast between the object and the background regions is noticeable. Nevertheless, global thresholding fails to provide an acceptable result if there is overlapping between the intensity values of two objects. The main limitation of global thresholding is that it divides the image based on the intensity value without taking into consideration the relationship between the pixels [164].

- Local thresholding: If there are more than two different regions in the image, the segmentation is done through local thresholding. In other words, image segmentation can be accomplished through utilizing multiple threshold values. The threshold value for each object is known as the local threshold. Equation 4.2 represents the local thresholding using $(n-1)$-threshold values [176]. Figure 4.4c shows the result of the multiple thresholding.

$$g_{(x,y)} = \begin{cases} \text{object 1} & \text{If } f_(x,y) \leq T_1 \\ \text{object 2} & \text{If } T_1 < f_(x,y) \leq T_2 \\ \vdots & \vdots \\ \text{object n} & \text{If } f_(x,y) > T_{n-1} \end{cases} \tag{4.2}$$

where $g_{(x,y)}$ is the segmented binary image, $f_{(x,y)}$ is the original pixel value and $(T_1, T_2, \cdots, T_{n-1})$ are the $(n-1)$-threshold values.

A threshold value can be chosen either manually or automatically. Threshold recognition approaches, such as optimal thresholding, p-tile thresholding and histogram shape analysis, are used to select the threshold value automatically [177].

Usually threshold-based segmentation methods, either global or local, are mostly incapable of generating accurate segmentation results and thus it is suggested they are applied as a pre-processing step in the segmentation process [164].

4.3.2 Edge-based Segmentation

Edge detection is a basic and simple algorithm for image segmentation. Edge detection is classified as a boundary-based segmentation method, where it converts the original image into an edge image by identifying the sharp changes in the intensity value. In image processing, edge detection deals with the localization of significant changes of a grey-level image and the recognition of the physical and geometrical characteristics of objects. It is a basic technique which is designed to identify and delineate the object boundary amongst other objects and the background of the image [175, 177].

Edge-detection algorithms are used in various object-detecting applications like medical image processing and biometrics. There are three different types of discontinuities in the grey-level: point, line and edges. Spatial masks can be used to detect all the three types of discontinuities present in an image [178].

There are various edge-detection algorithms for image segmentation such as Roberts, Sobel, Prewitt, Kirsch, Robinson, Marr-Hildreth, LoG and canny edge detection [177, 178]. Edge-detection methods are not appropriate for noisy or complex images where they can generate missing or extra edges [10].

4.3.3 Region-based Segmentation

A region-based segmentation method constructs objects by associating or dissociating neighbour pixels based on the fact that neighbour pixels inside one region are homogeneous and have similar attributes, while there is a high contrast value between two different regions. It is relatively simple and more immune to noise than edge-based segmentation. The region-based technique should be compatible with the following rules for image segmentation [166]:

Suppose I is the original image that is divided into n regions $(R_1, R_2, \cdots, R_n)$, where $i = 1, 2, \cdots, n$.

(1) $I = \bigcup_{i=1}^{n} R_i$, combining the n regions forms the original image.

(2) $R_i \cap R_j = \phi, \forall i \neq j$, the intersection (overlapping) between two different regions should be an be empty set.

(3) $P(R_i) = \text{True}, \forall i = 1, 2, \cdots, n$, the uniformity predicate $P(R_i)$ is true for each region R_i.

(4) $P(R_i \cup R_j) = \text{False}, \forall i, j = 1, 2, \cdots, n$ and $i \neq j$, two adjacent regions should not have the same uniformity predicate.

(5) R_i must be a connected region.

Region-based methods are categorized into region growing, region split and merge and watershed.

- Region-growing method: The region-growing method groups the pixels according to predefined criteria. It begins by defining a set of seed points. Then, these seeds grow by appending each seed to the pixel neighbours that have a high degree of similarity. This process iterates until no pixels can be further included within the region.The similarity criteria is chosen based on the problem and image type. The value of the seed points are selected manually. The benefit of region growing is because it arranges and segments regions that have comparable properties and can produce connected regions. The region-growing method can correctly separate regions and generate a connected region that has identical characteristics [179]. However, the presence of noise in the original image leads to improper initialization of the seed values.
- Split-merge method: The split-merge method, also known as quadrant segmentation, relies on the quadtree division of an image. Firstly, it assigns the root of the tree to the whole input image. If homogeneity (uniformity) is absent, then the tree node is broken into four regions. However, if four son-squares are similar, then they can be assembled as one connected region. This process iterates until no more similar regions need to be merged or heterogeneous regions need to be split [166].
- Watershed: The concept of the watershed method comes from the behaviour of water in a landscape. When it rains, the water drops go downhill, passing by different areas. The water is then collected at the bottom of the valley where the gravitational pull is the strongest. Water from each valley flows in an area that is linked with catchment basins which are only connected to a single basin. The dams are constructed at the points where the water meets from various basins. The landscape is divided into different areas by dams, when the water level has attained the highest peak of the landscape [164].

The watershed method is utilized if the background and foreground of the image can be identified. It can also recognize the weak edges. However, the watershed method has a strong potential for over-segmentation [164].

4.3.4 Machine Learning-based Segmentation

Automatic image segmentation has become an important research area in machine learning. This segmentation scheme depends on pixel classification. The image pixel's properties, such as intensity, local texture and luminance, can be used to generate the feature space where the segmentation process is completed by identifying the similarity between the pixels based on the feature space. Different methods have been developed to segment the image based on machine learning, where it can be categorized into three groups: (1) supervised, (2) semi-supervised and (3) unsupervised learning schemes [180].

The supervised methods use training data that have been manually labelled, while the unsupervised methods use clustering algorithms instead of manual labelling.

4.3.4.1 Supervised Learning (Classification)

The supervised methods, also known as classification, require labelled data to separate the feature space and involve training and testing phases. In the training phase, the manual labelled data is used to build a training model that matches the feature space to the labelled data. The testing phase is used to designate labels to unlabelled data based on the training model. For instance, the simple supervised approach for segmenting a brain tumour image is to use two labels, normal and tumour, and to select the pixel intensities as a feature of the model. In the supervised segmentation method, selecting a suitable training dataset is important because different training datasets can influence the segmentation accuracy results [161].

The k-nearest neighbours (kNN) is one of the simplest classifications among all of the machine-learning algorithms. It classifies the testing dataset by computing the distance between each query of the testing dataset and the training dataset. The query testing data is then assigned to the nearest class [181].

The main limitation of the supervised methods is the necessity of having a large amount of labelled data, which may not always be available, and hence limits its applicability [182]. Also, the use of the same training set for a large number of images can lead to biased results that do not take into account anatomical and physiological variability between different subjects [161].

4.3.4.2 Unsupervised Learning (Clustering)

The unsupervised techniques do basically the same function as the classifier by separating the image into classes based only on the feature space without having training (labelled) data. It manually learns from the available input data to group the similar pixels into one class, while pixels with dissimilar features are grouped into different classes based on similarity/dissimilarity measurement criteria [161].

The most common clustering methods are the k-means [183], fuzzy c-means (FCM) [184], self-organized map (SOM) [185] and the expectation-maximization (EM) [186] algorithms. The K-means clustering algorithm clusters the data by repeatedly computing the mean intensity for each class then classifying the pixel in the class with the closest mean. It is also known as a hard classification method because each pixel should belong to one class. The FCM is a soft classification method, because it is possible for each pixel to belong to multiple classes. The computational cost of the FCM is higher than K-means. The EM algorithm applies the same clustering principles, with the underlying assumption that the data follows a Gaussian mixture model. The drawback of the clustering techniques is the need to determine the number of classes in advance.

4.3.4.3 Semi-supervised Learning

Semi-supervised learning is situated halfway between supervised and unsupervised learning. In semi-supervised clustering approaches, additional information is incorporated to adjust the clustering process in order to improve the segmentation result. There are three different types of additional information: must-link and cannot-link constraints, label data and a pre-defined membership matrix [187]. These are discussed below.

(1) Must-link and cannot-link constraints: Must-link constraints mean to identify the points that must belong to the same class, while cannot-link constraints refer to the points cannot be in the same class.

(2) Label data: A part of data is labelled and others are unlabelled.

(3) Prior membership matrix: The pre-defined membership matrix from an unsupervised clustering algorithm is used to guide the semi-supervised learning.

4.4 Medical Image Processing

Current imaging techniques developed for medical purposes provide a huge insight into detailed images which require a deep analysis in a very short period of time. The medical images are evaluated by specialists through a number of procedures that increase the chance of human errors occurring, which increases the time required for making evaluations and can even lead to wrong interpretations. Medical images are assessed both qualitatively and quantitatively by professionals on the basis of their professional experience, but this analysis has limitations as the assessment is purely based on the human vision system, i.e. human eye vision which can only analyze eight bits of grey-level [188].

Advanced medical imaging systems have the ability to produce images that have up to 65,535 different grey levels. Certain essential data produced through the scanner cannot be examined, let alone be analyzed by a normal human eye. For the in-depth analysis of both high and low resolution medical images, computer-aided diagnosis (CAD) can be employed. CAD is a supportive tool that helps surgeons to conduct examinations of abnormal regions without mistakes. Millions of lives can be saved with early and accurate diagnoses, which are made possible with this technology [188].

Nowadays, image processing is extensively integrated in medical imaging systems due to the rapid improvement and revolution in computerised medical image visualization for image analysis and computer-aided diagnosis [189]. Image division into a group of homogeneous regions is necessary for feature extraction and analysis. Different modalities are available for medical imaging systems, such as computed tomography (CT), magnetic resonance image (MRI), X-radiation (X-ray) and positron emission tomography (PET) [190]. Physicians recognize MRIs as the most versatile medical imaging modality for clinical diagnosis [182].

4.5 Magnetic Resonance Brain Image Processing

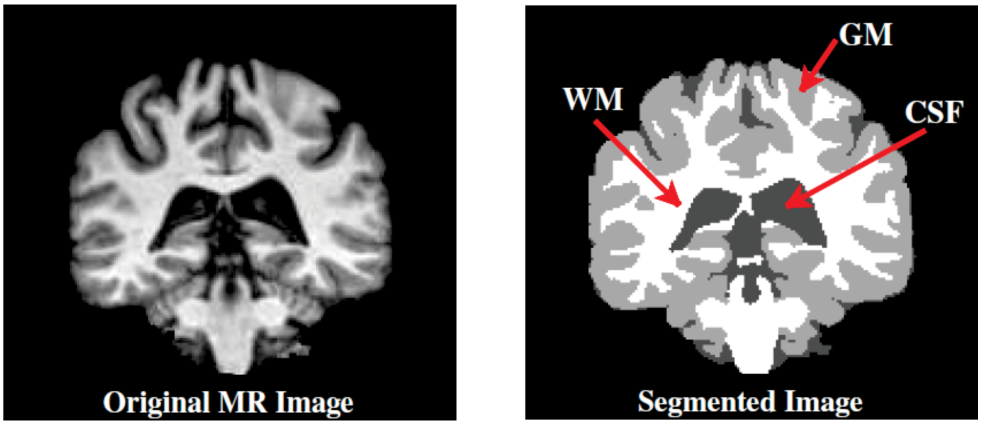
Technological growth has positively improved imaging systems, in particular MRIs. Improvement in the quality and speed of image generation are some of major consequences of these technological developments. An MRI is a medical imaging mechanism employed in radiology for detailed visualizing of the organs and tissues inside the body, particularly for brain imaging. It works by using nuclear magnetic resonance (NMR) to create an image using the nuclei of atoms located in the body. MRI techniques give more distinctive information about the internal structure of organs compared to the other imaging modalities, such as X-rays, ultrasounds or PET, which produce noisy and blurred images [191].

The MRI sequence is an integration of radio frequency (RF) and gradient pulses designed to generate the image. An intense magnetic field is employed by the MRI machine to adjust the magnetization of protons in the body and radio frequency fields are responsible for systematic change in the alignment of this magnetization. Consequently, a rotating magnetic field of larger frequency by the scanner is generated by the protons and this information is recorded to form an image of the scanned area of the body. Strong magnetic field gradients cause nuclei at different locations to rotate at different speeds. Three-dimensional spatial information can be obtained by providing gradients in each direction [192].

Once an MR image is obtained, several image processing techniques can be applied on MR brain images, which would lead to a more efficient patient diagnosis. The MR brain image processing faces certain issues in the research domain. Brain tissue segmentation and the differentiation of normal and diseased tissues present in multi-channelled MR brain images, which are in different sequences and are of multi-axis structures [188].

4.6 MR Brain Image Segmentation Methods

The segmentation process of brain images aims to divide the brain image into three main tissues: white matter (WM), grey matter (GM) and cerebrospinal fluid (CSF) [193]. Figures 4.5a and 4.5b show an original MR image and its corresponding labelled segmented image respectively. The assigned labels facilitate the classification and grouping of brain tissues where pixels with similar labels represent a certain computed characteristic such as texture, intensity, shape or colour [9].



(a) (b)

Figure 4.5 (a) Original MR brain image and (b) segmented image with WM, GM and CSF labels

Numerous methods have been implemented in the area of medical segmentation to provide the accurate segmentation of MR images. Differences in such methods are based in mechanisms that have been utilized to divide the image into non-overlapping regions. Table 4.1 shows some existing research relating to brain image segmentation.

In [194], two unsupervised methods were designed for MR brain image segmentation using SOM. The first approach is termed as histogram fast segmentation SOM (HFS-SOM). In HFS-SOM, features are extracted from the histogram of the image. The feature vectors comprise of intensity occurrence probabilities, the relative position regarding the intensity value, the mean of the probability values over a 3-bin window and the variance of that window, which is used to train the SOM. Then, the k-means is utilized to cluster the SOM output layer. The second approach is known as entropy gradient segmentation (EGS-SOM). First and second-order statistical features are extracted from overlapped windows of size 7 × 7 pixels. In order to train the SOM, a genetic algorithm is applied to select the most discriminative features. Finally, SOM outputs are clustered using the EGS algorithm. The HFS-SOM method is faster because it does not require any parameter setup, while the second approach is more robust under noisy and bad intensity conditions.

Noreen et al. [195] introduced a hybrid MR segmentation method that utilizes DWT and FCM to remove inhomogeneity. Firstly, DWT was applied to the input MR image to obtain LL, LH, HL and HH sub-bands. To obtain a sharpened image, the approximation coefficients of the LL sub-band are set to zero. Then, the IDWT is applied to get a high pass image. The resultant image is segmented using the FCM technique. Finally, Kirch's edge detection mask [196] is performed to fill the missing edge information and enhance the output image.

As a substitute, brain images can be segmented using Gaussian mixture model (GMM) [197], where the pixel intensities of each region are modeled by a Gaussian distribution function [198]. Generally, the GMM parameters are approximated by the EM algorithm [199]. However, the GMM-EM approach does not take into consideration the spatial information and uncertainty of the data. To overcome this limitation, Greenspan et al. [200] and Blekas et al. [201] integrated the spatial constraints into GMM to improve the segmentation accuracy value.

In [202], an automatic brain MRI segmentation method is proposed. For each label, the voxel intensities of all MRI sequences are modelled using Gaussian distributions. The parameters of the Gaussian distributions are evaluated as maximum likelihood estimates and the posterior probability of each label is determined using Bayesian estimation. Regional intensity, texture, the spatial location of voxels and the posterior probability estimates are used as features that are utilized to classify each voxel into one of the four classes (CSF, GM, WM and background) using a multi-category support vector machine (SVM).

Table 4.1 Literature review for various brain image segmentation methods

Author	Segmentation Technique	Description
Selvy et al. (2011) [203]	Four different clustering techniques	The grey MRI is transformed to colour space using pseudo-colour transformation. Then, a clustering technique is applied to segment the image
Ortiz et al. (2013) [194]	HFS-SOM clustering	Features are extracted from the whole volume histogram which is trained by using SOM. Then, the k-means is utilized to cluster the SOM output layer.
Ortiz et al. (2013) [194]	EGS-SOM clustering	Features are extracted from overlapped windows of a size 7 × 7 pixel. Genetic algorithm (GA) is applied to select the most discriminative features to train the SOM.

Table 4.1 Literature review for various brain image segmentation methods

Author	**Segmentation Technique**	**Description**
Ortiz et al. (2013) [204]	SOM-FCM	The 3D statistical features are extracted from the image. The GA-based selection is performed over the extracted features to form an optimized number of feature vectors. These feature vectors are modelled by SOM. Then, FCM is used to compute the degree of voxel modelled by SOM.
Goncalves et al. (2014) [205]	Discriminative clustering (DC) using labels obtained from consistent SOM	A pre-processing is applied to correct field inhomogeneities. Then, this method employed a semi-supervised (DC) method using labels obtained from SOM to segment the brain image.
El-Dahshan et al. (2014) [206]	Feedback pulse-coupled neural network (FPCNN)	The MR image is segmented using the FPCNN. Then, DWT features are extracted from the image and reduced using is principal component analysis (PCA). Finally, the FBPNN is applied to classify the image into pathological or abnormal.

Table 4.1 Literature review for various brain image segmentation methods

Author	Segmentation Technique	Description
Kong et al. (2015) [207]	Information theoretic discriminative segmentation (ITDS)	The simple linear iterative clustering (SLIC) is employed to generate 3D supervoxels for brain MRI. Features are extracted from each supervoxel. ITDS is used for clustering the supervoxels.
Pereira et al. (2016) [208]	Convolutional neural networks (CNN)	Features are extracted using sparse auto-encoder NN. Different CNN architecture segments the brain image, exploring the use of small convolution kernels. Then, post-processing is applied to remove classes that is smaller than a predefined threshold.

4.7 Summary

This chapter presented an overview of image segmentation methods. The segmentation methods can be classified into three categories: manual, semi and fully-automatic methods, depending on the human level of interaction. The main limitations of manual segmentation are time consumption and impracticality. On the other hand, user interaction in automatic segmentation is not necessary, however, automatic segmentation suffers from many challenges that limits its practical applications.

This chapter also discussed the different segmentation techniques, including the low and high level stages. In the discussion of low level image segmentation, edge-based, thresholding-based and region-based are explained. Segmentation results of these method are not very accurate because they do not incorporate high level knowledge about the image. These techniques are usually used in the pre-processing stage of the segmentation process.

MRI is one of the most versatile imaging techniques in medical analysis, such as in the examination and treatment of brain tissues, where it has helped in the identification and differentiation of normal and diseased tissues.

Finally, this chapter reviewed various methods that have been implemented in the area of brain image segmentation to provide accurate segmentation of MR images.

CHAPTER 5

Image Steganography based on Edge Detection and Coding[1]

This chapter presents a novel image steganography algorithm that combines the strengths of edge detection and XOR coding, to conceal a secret message either in the spatial domain or an Integer Wavelet Transform (IWT) based transform domain of the cover image. In order to enhance the imperceptibility, edge detection algorithm identifies sharp edges in the cover image for embedding to cause less degradation to the image quality compared to embedding in a pre-specified set of pixels that do not differentiate between sharp and smooth areas. In addition, the secret data is embedded in the edge pixels using the XOR operation to minimize the difference between the cover and stego images.

[1] The contents of this chapter have been published in the Journal of Expert Systems with Applications, Vol. 46, (2016), entitled "A Steganography Embedding Method Based on Edge Identification and XOR Coding".

5.1 Introduction

The major challenge in developing the embedding process of steganography is ensuring a high quality of the stego image without compromise the embedding capacity. As mentioned earlier in Section 2.6, several requirements should be taken into consideration during design steganographic system. However, these requirements are strongly dependent upon each other and there is a trade-off between them [44].

The LSB substitution is one of the conventional steganography methods that most of the researchers used due to its simplicity, low computational cost and low memory space. Also, LSB is visually imperceptible based on the assumption that the least significant bit of the pixel value is unimportant. However, the LSB method does not differentiate between the smooth and high contrast regions which highly facilitates the steganalysis process. Currently, LSB method can be discovered and identified by existing steganalysis method.

In order to improve the detectability and capacity of LSB, many adaptive steganography techniques have been proposed based on the fact that the human visual system is less sensitive to change in the edge areas compared to the smooth area. However, the methods employ the edge detection algorithms, such as Sobel and Canny, for identifying the embedding location are either offer low embedding capacity or not able to identify the same edge between the cover and stego images. Also, some of these algorithms are unable to utilize the four edge directions, such as PVD [81] and EALSBMR [92]. PVD and EALSBMR offer a high embedding capacity, but it does not accommodate with the undetectability requirement.

A novel image steganography method is presented, which conceals secret data either inside the spatial domain or Integer Wavelet Transform (IWT) domain in such a way that it offers a good image quality without compromise the embedding capacity. The new edge detection method ensures the identification of the same edges between the cover and stego images. The probability of detecting the existence of the hidden message is solidly based on the amount of the embedding distortion in the cover image caused by the embedding process. Therefore, designing steganography system with less number of modification for the same embedding capacity is recognized as an essential requirement to minimize the distortion function. In this chapter, a new and simple XOR operation(embedding process) is developed to improve the embedding efficiency and reduce the artefacts occurred because of the embedding process. Moreover, the computational cost for the embedding process is lower than steganography algorithms based on matrix embedding.

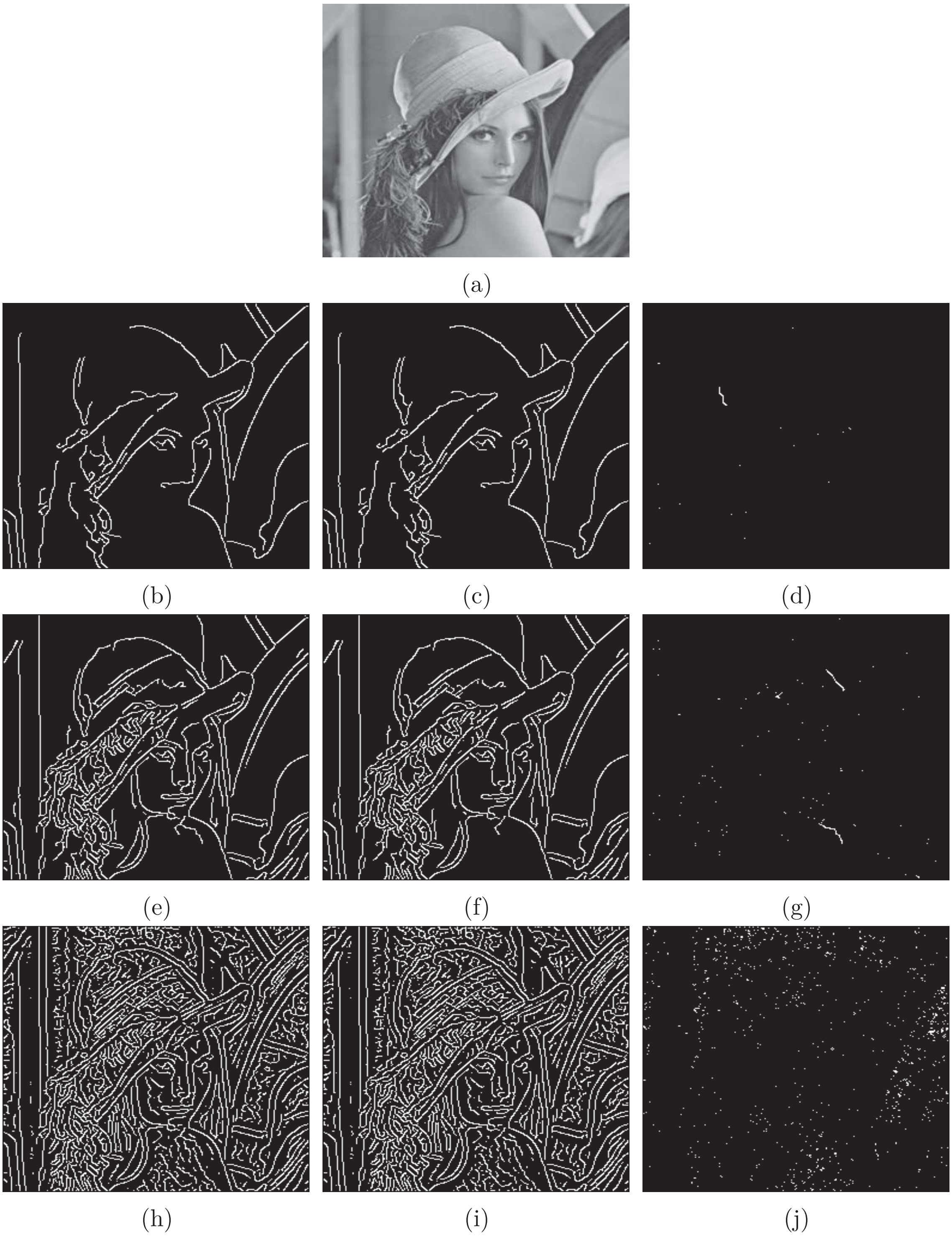
This chapter is organized as follows. Section 5.2.1 introduces the new edge detection algorithm. The Spatial Domain Algorithm using one bit per pixel and n bits per pixel are explained in Sections 5.2.2 and 5.2.3 respectively. Section 5.2.4 presents the integer wavelet transform domain algorithm (n bits per pixel). Finally, the summary is given in Section 5.3.

5.2 The Proposed Methodology

In this section, the image steganography is presented. The framework is based on new edge detection algorithm and XOR operation. Also, the secret data can be embedded whether in the spatial domain or Integer Wavelet Transform (IWT) based transform domain of the cover image. Three different implementations of the proposed method are explained in sections 5.2.2, 5.2.3 and 5.2.4.

5.2.1 Identification of Edges

It is well known that the human visual system is less tangible to changes in image areas that contain edges and sharp transitions in comparison to smooth areas. Accordingly, it is logical to conceal the message in edge areas in order for the steganography algorithm to have a good imperceptibility.



(a)

(b) (c) (d)

(e) (f) (g)

(h) (i) (j)

Figure 5.1 (a) Cover image, (b−d) Edge pixels in a cover image using Canny method, (e−g) Edge pixels in a stego image using Canny method with 3%, 10% and 19% embedding rates and (h−j) Difference between edge pixels in the cover and stego images

The edge image generated by traditional edge detection methods is usually sensitive to changes in the original grey image, even if the changes are minor or not significant. This property limits the utilization of edge detection in steganography, as concealing the message would introduce some changes to the original image. Thus, embedding in pixels identified by one of the existing edge detection methods, such as Canny, cannot guarantee the identification of the exact edge intensities for the cover and stego images. The cover image of size 256 × 256 is shown in Figure 5.1a. Figures 5.1b, 5.1c and 5.1d show the corresponding edge pixels, which are identified by applying Canny edge detection using different threshold values. Edge pixels of three stego images produced after embedding messages of length 2000, 6553 and 12639 bits using the LSB steganography method are shown in Figures 5.1e, 5.1f and 5.1g respectively. Figures 5.1h, 5.1i and 5.1j show the difference between edge pixels in the cover and stego images using low, medium and maximum embedding rates (3%, 10% and 19% respectively), which indicate that edge pixels in the cover and stego images are not identical.

A new and simple edge detection algorithm is proposed to discover the edge (sharp) regions of the cover image, such that the two edge images generated using the original cover image and the stego image are identical. This will enable the correct extraction of the concealed message from the stego image. The algorithm starts by dividing the image into non-overlapping blocks that would be individually evaluated and then categorized as either edge or non-edge blocks. The key idea behind preserving the same edge image is not to embed in the pixels that are used to calculate the edge strength, which are the outer pixels of the block. The edge detection algorithm (5.1) is explained in the following steps:

Step 1: Divide the image C into non-overlapping blocks of the size $n \times n$. Figure 5.2 shows a 3×3 block.

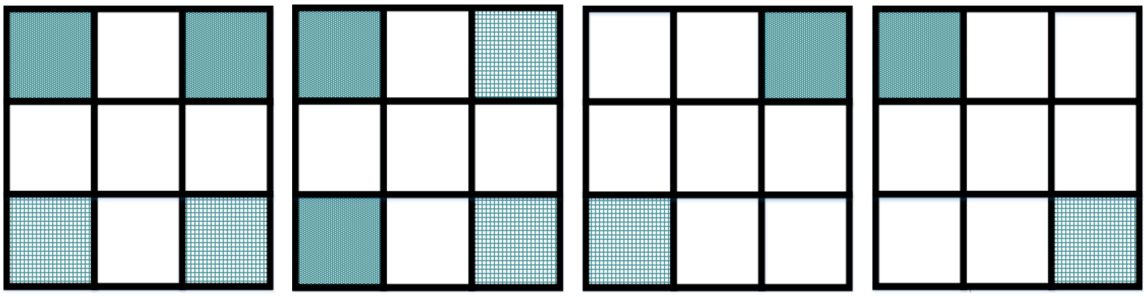
Step 2: Compute the absolute mean difference between the left and right columns of the block (magnitude of vertical edge (VE)). Repeat for horizontal (HE), first diagonal ($D1$) and second diagonal ($D2$) edges. Edge magnitude can be computed using Eq. 5.1.

$$\begin{aligned}
HE &= avg\Big|([P_{(i-1,j-1)}, P_{(i-1,j+1)}] - [P_{(i+1,j-1)}, P_{(i+1,j+1)}])\Big| \\
VE &= avg\Big|([P_{(i-1,j-1)}, P_{(i+1,j-1)}] - [P_{(i-1,j+1)}, P_{(i+1,j+1)}])\Big| \\
D1 &= avg\Big|(P_{(i-1,j+1)} - P_{(i+1,j-1)})\Big| \\
D2 &= avg\Big|(P_{(i-1,j-1)} - P_{(i+1,j+1)})\Big| \qquad (5.1)
\end{aligned}$$

Step 3: Find the maximum of the four values and assign it to e, which is computed using Eq. 5.2. If $e > Th$, then the block is considered to be an edge block, otherwise it is not an edge block. Construct E that contains the calculated e value of each of the edge blocks (which reflects the edge strength), and 0 for non-edge blocks. A binary edge image can also be constructed, which contains 1 for edge blocks and 0 for non-edge blocks.

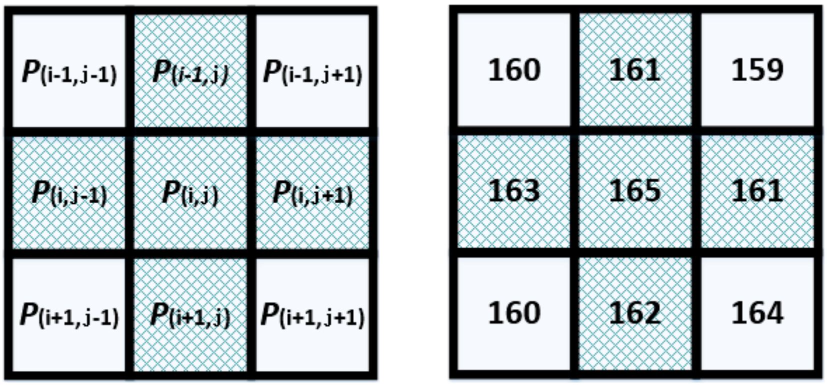
$$e = max\Big\{HE, VE, D1, D2\Big\} \qquad (5.2)$$

Step 4: For the edge blocks, embed in the white 5 pixels, $P_{(i-1,j)}, P_{(i,j-1)}, P_{(i,j)}, P_{(i,j+1)}, P_{(i+1,j)}$, as shown in Figure 5.3a.



(a) (b) (c) (d)

Figure 5.2 An example of 3 × 3 block edges for four directions (a) Horizontal, (b) Vertical, (c) First Diagonal and (d) Second Diagonal



$P_{(i-1,j-1)}$	$P_{(i-1,j)}$	$P_{(i-1,j+1)}$
$P_{(i,j-1)}$	$P_{(i,j)}$	$P_{(i,j+1)}$
$P_{(i+1,j-1)}$	$P_{(i+1,j)}$	$P_{(i+1,j+1)}$

(a)

160	161	159
163	165	161
160	162	164

(b)

Figure 5.3 (a) Selected pixels for embedding 3 × 3 block and (b) a 3 × 3 block of input image

Algorithm 5.1: Edge Detection

```
Inputs   : Cover image ($C$), block size ($n \times n$, which is expected here to be
           $3 \times 3$), threshold value ($Th$).
Outputs : Edge image with edge magnitude ($E$) and Binary edge image ($B$)
$n \leftarrow 3$;
[row,col] $\leftarrow$ size($C$);
$E \leftarrow [\,]$ ; $B \leftarrow [\,]$;
$i \leftarrow 1$ ;
while $i \leq row$ do
    $j \leftarrow 1$ ;
    while $j \leq col$ do
        Subblock $= C(i : i + (n-1), j : j + (n-1))$;
        Compute the four magnitude using Eq. 5.1;
        Find $e$ using Eq. 5.2 ;
        $E(i : i + (n-1), j : j + (n-1)) = e$;
        if $e > Th$ then
            $B(i : i + (n-1), j : j + (n-1)) = 1$;
        else
            $B(i : i + (n-1), j : j + (n-1)) = 0$;
        $j \leftarrow j + n$ ;
    $i \leftarrow i + n$ ;
```

For example to evaluate the block shown in Figure 5.3b as edge or non-edge block, the magnitude of the four directions are computed as follows:

- The magnitude of the horizontal row is $\Big|(160 + 159) - (160 + 164)\Big| = 5$.

- The magnitude of the vertical column $\Big|(160+160)-(159+164)\Big| = 3$.
- The magnitude of the first diagonal is $\Big|159-160\Big| = 1$.
- The magnitude of the horizontal row is $\Big|160-164\Big| = 4$.

Finally, e is the maximum of (5,3,1,4). If $e = 5 > Th$, then this block is categorized as edge region. Otherwise, It is non-edge block.

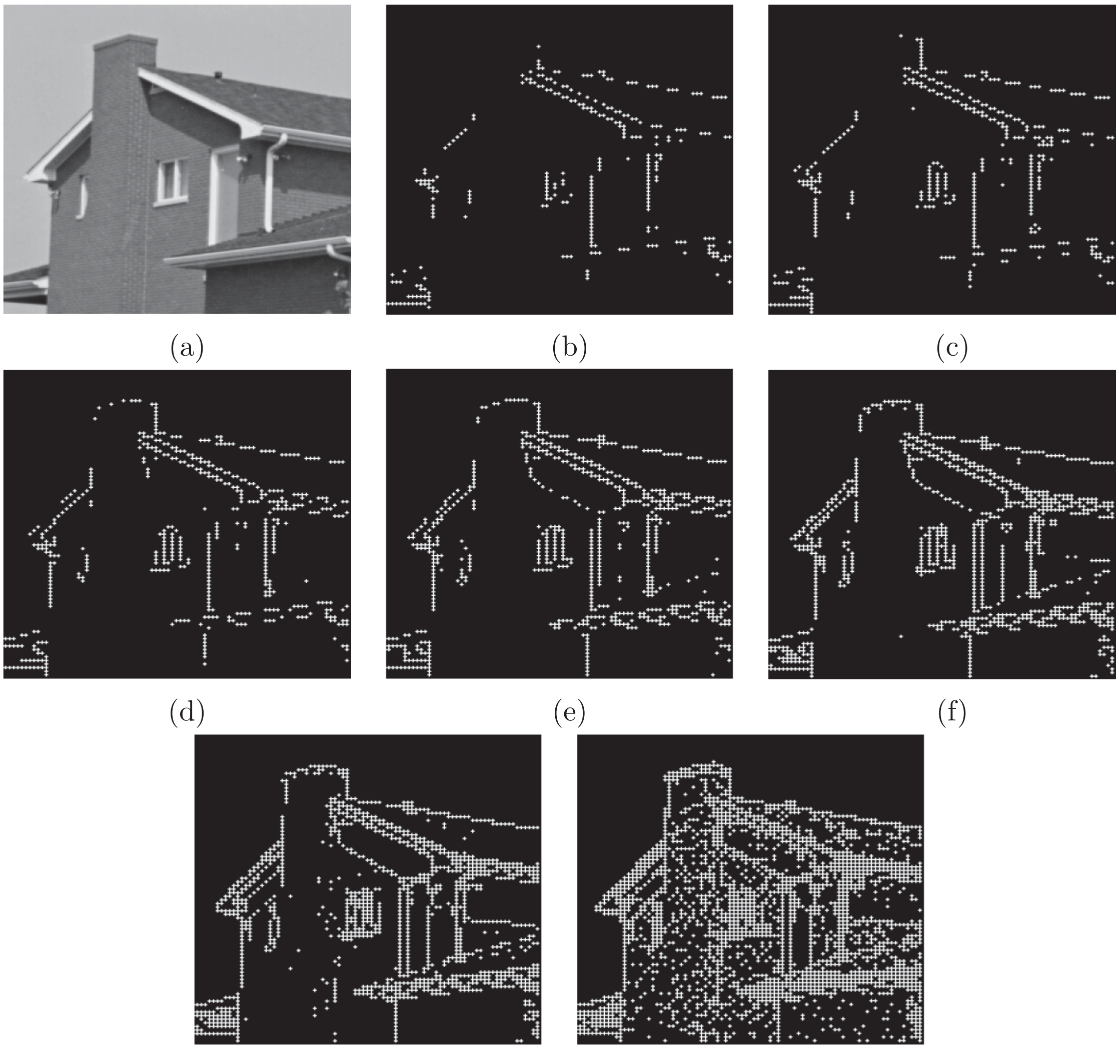
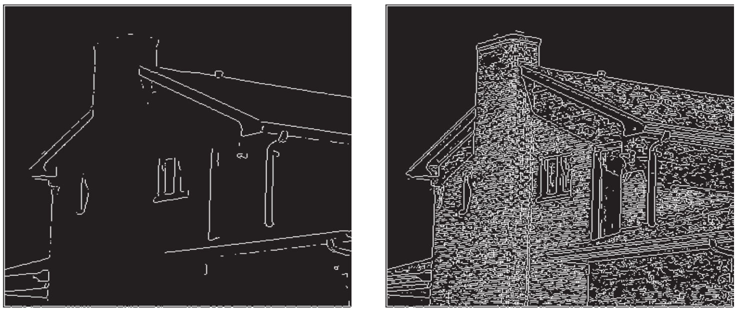


Figure 5.4 (a) Input image, (b) edge image using $Th = 70$, (c) edge image using $Th = 60$, (d) edge image using $Th = 50$, (e) edge image using $Th = 40$, (f) edge image using $Th = 30$, (g) edge image using $Th = 20$, (h) edge image using $Th = 10$



(a) (b)

Figure 5.5 Edge image using Sobel method (a) $Th = 0.1$ and (b) $Th = 0.01$

In order to evaluate the obtained binary edge image of the proposed algorithm, we considered the grey image shown in Figure 5.4a and used different values of threshold in constructing binary edge images using a block size of 3×3, as shown in Figures 5.4b$-$ 5.4h. The edge images indicate the ability of this method in detecting edges with an acceptable accuracy compared to the existing edge detection methods as shown in Figure 5.5. Out of the nine pixels of the block, the five pixels shown in Figure 5.3a will be used for embedding if the block is identified as an edge block. Thus, the four corner pixels that are used for estimating the edge strength will remain unchanged after embedding. This guarantees each block in the cover image to have the same edge strength as its counter part in the stego image.

5.2.2 The Spatial Domain Algorithm (0.75 bit per pixel)

5.2.2.1 The Embedding Process

The flow diagram of our proposed method is illustrated in Figure 5.6. The data embedding process begins with reading the cover image and the secret message. A high threshold (96) is initially considered, which is then adjusted based on the number of pixels needed for embedding the message (identified by the generated binary edge image) and the message length, according to the following condition:

For the given threshold value, if *(no. of edge pixels* $\geq$ *(4 * Message Length) /3))* then the discovered area is enough to embed the secret message.

The embedding process is performed on the detected edge locations using the proposed XOR coding. This method partitions the index table into groups of four pixels and encodes three message bits into the pixels of each group. The XOR operation ensures that the secret message is concealed into the cover with minimum number of pixel changes. Thus, the three secret bits m_1, m_2, and m_3 are embedded in the four LSBs p_1, p_2, p_3, and p_4 (one bit for each edge pixel) according to the following procedure:

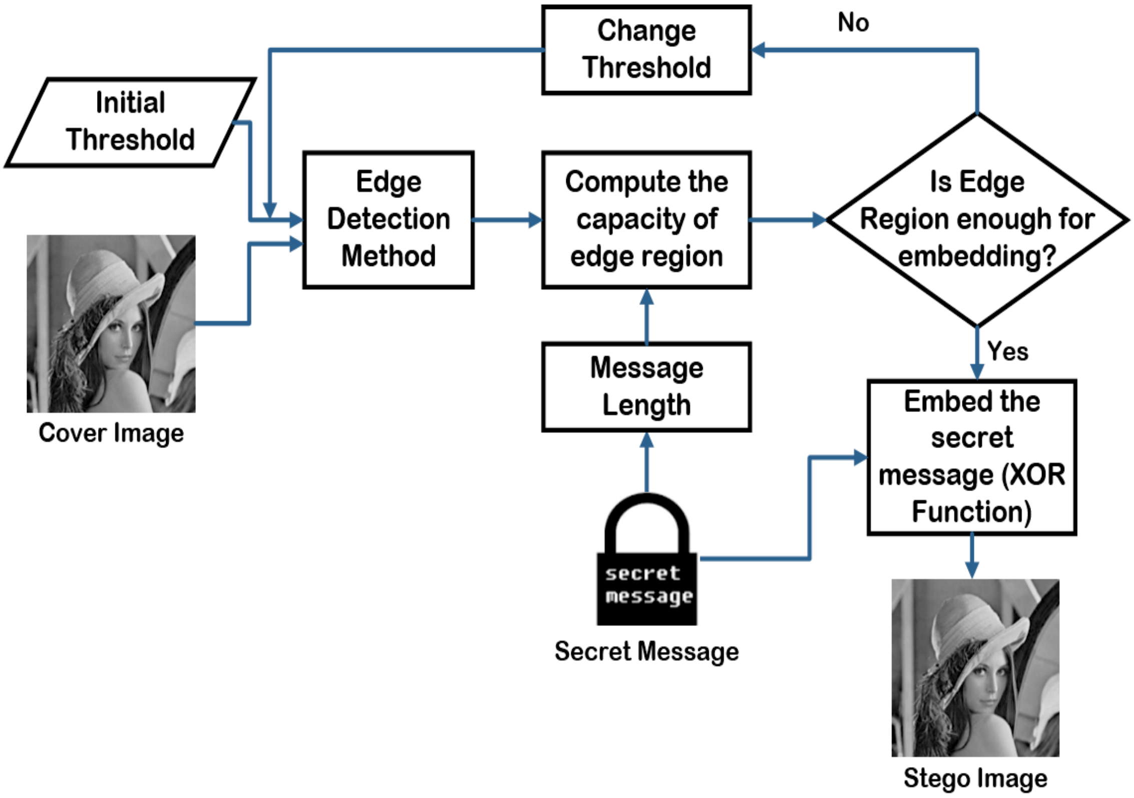


Figure 5.6 Data embedding process in the spatial domain

1. Perform the following three XOR operations

   $k_1 = p_1 \oplus p_2$

   $k_2 = p_3 \oplus p_4$

   $k_3 = p_1 \oplus p_3$

2. To embed the three secret bits m_1, m_2, and m_3, the three calculated bits k_1, k_2 and k_3 are compared with the secret message bits m_1, m_2, and m_3. The result of this comparison, which can take one of eight possibilities, determines which of the four bits p_1, p_2, p_3, and p_4 have to be modified, as shown in Table 5.1.

We will refer to the new four bits of the stego image as q_1, q_2, q_3, and q_4. The table indicates that embedding 3 message bits into 4 cover bits will cause an average modification of 0.3125 bits. For instance, in the following experiment, an embedding process of a secret message of size 3249 bits that are randomly generated into 4332 cover bits modifies 1358 bits. The average modification of this experiment is 1358/4322=0.3135.

Table 5.1 Embedding conditions

Condition			Action to be taken
$m_1 = k_1$	$m_2 = k_2$	$m_3 = k_3$	No change required
$m_1 = k_1$	$m_2 = k_2$	$m_3 \neq k_3$	Complement p_3 and p_4
$m_1 = k_1$	$m_2 \neq k_2$	$m_3 = k_3$	Complement p_4
$m_1 = k_1$	$m_2 \neq k_2$	$m_3 \neq k_3$	Complement p_3
$m_1 \neq k_1$	$m_2 = k_2$	$m_3 = k_3$	Complement p_2
$m_1 \neq k_1$	$m_2 = k_2$	$m_3 \neq k_3$	Complement p_1
$m_1 \neq k_1$	$m_2 \neq k_2$	$m_3 = k_3$	Complement p_2 and p_4
$m_1 \neq k_1$	$m_2 \neq k_2$	$m_3 \neq k_3$	Complement p_1 and p_4

3. The index of the threshold value should also be embedded, as it is needed by the extraction process. In this algorithm, the index of the threshold value is embedded into the four LSBs of the last pixel to avoid making a significant change in the original pixel.

Suppose three message bits $(1, 0, 1)$ needs to be embedded in the edge block as shown in Figure 5.3b. The following steps illustrate the message embedding process:

(1) The LSB values of the four edge pixels (161,163,165 and 161) are $(p_1 = 1, p_2 = 1, p_3 = 1 and p_4 = 1)$ respectively.

(2) The secret message bits are $(m_1 = 1, m_2 = 0 and m_3 = 1)$.

(3) The XOR operation result is $k_1 = 1 \oplus 1 = 0$, $k_2 = 1 \oplus 1 = 0$ and $k_3 = 1 \oplus 1 = 0$.

(4) According to Table 5.1, p_1 is changed $(p_1 = 0)$ because $m_1 \neq k_1$ and $m_3 \neq k_3$.

(5) The stego edge pixels are (160,163,165 and 161).

5.2.2.2 The Extraction Process

The extraction process is easier and faster than the embedding process. Figure 5.7 represents the flow diagram of the extraction process. It starts by retrieving the threshold value. The edge blocks of the stego image are then identified using the retrieved threshold, which will return the same edge image as the one obtained using the cover image. This will be followed by dividing the LSBs of the edge pixels into groups of four. Finally, for each of the four stego edge bits q_1, q_2, q_3, and q_4 the XOR operations listed below are used to retrieve three message bits m_1, m_2, and m_3

$m_1 = q_1 \oplus q_2$

$m_2 = q_3 \oplus q_4$

$m_3 = q_1 \oplus q_3$

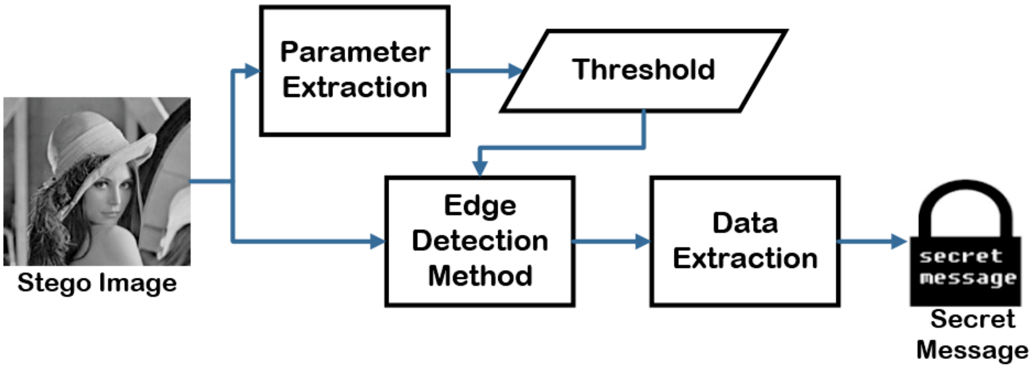


Figure 5.7 Data extraction process in the spatial domain

When considering any combination of $m_1, m_2, m_3, p_1, p_2, p_3$, and p_4 to verify the embedding and extraction processes, one can find that the extraction process truly restores the original message.

5.2.3 The Spatial Domain Algorithm (n bits per pixel)

5.2.3.1 The Embedding Process

In order to improve the embedding capacity, we present here an extension of our 0.75 bpp algorithm to embed n bits in each edge pixel. The value of n is to be determined based on the edge mean value of each block. Thus, strong edges will enable the embedding of more bits than the less strong ones. Hence, unlike the embedding of one bit per pixel that only considers the existence of an edge in a block, this algorithm utilizes the edge strength of each block, e. Embedding n bits per pixel, where n varies from one block to another, may also improve the security of the message, as in this case n needs to be correctly calculated for each block in order to successfully reveal the message.

The data hiding process begins with reading the cover image and the secret message. The new edge detection is then applied to produce the edge strength, e, of each block. In order to specify the number of bits to embed, n, the edge pixels are classified into five groups (G_1, G_2, G_3, G_4 and G_5) based on the edge strength, e, of the block. The chosen of the group width depends on the sensitivity of human visual system to the alternations of gray value from smooth to sharp area. Table 5.2 lists the range of each of the five groups, and Eq. 5.3 determines the length of the message that can be embedded into the edge bits. If the identified edge pixels are not enough for embedding the whole message, then adjust the threshold and repeat the process until the actual length of the message satisfies Eq. 5.3.

$$\begin{aligned}(4 \times \text{Msg Length})/3) \leq \quad & \text{No. of Edge Bits}\Big[(3 \times G_5\, pixel) + (3 \times G_4\, pixel) + \\ & (3 \times G_3\, pixel) + (2 \times G_2\, pixel) + (1 \times G_1\, pixel)\Big] \end{aligned} \tag{5.3}$$

where the multiplier of each term represents the number of secret bits to embed in the edge pixel based on the group it belongs to.

Table 5.2 Number of bits can be utilized from each edge pixel according to the group it belongs to

Group	Group 1	Group 2	Group 3	Group 4	Group 5
n (bpp)	1	2	3	3	3
Range	[4, 7]	[8, 15]	[16, 31]	[32, 63]	[64,255]

The embedding process starts with the G_5 pixels and then moves to the remaining groups, where after it completes G_4 it moves to G_3 then G_2 and finally G_1 to embed 3, 3, 3, 2 and 1 bits in each of the corresponding edge pixels of these groups.

5.2.3.2 The Extraction Process

The retrieving process starts with performing the edge detection algorithm described earlier to get the edge strength, which would be used to categorize the edge blocks into group. Then, the XOR extraction operations are applied to extract three bits from the G_5 pixels, and then it respectively considers the G_4, G_3, G_2 and finally G_1 pixels to extract the corresponding number of bits from each of them.

5.2.4 The Integer Wavelet Transform Domain Algorithm (n bits per pixel)

5.2.4.1 The Embedding Process

The flow diagram of our proposed Integer Wavelet Transform (IWT) based embedding is illustrated in Figure 5.8. The process starts by converting the cover image to the wavelet domain using IWT. Since the HVS is sensitive to small modification into the lower frequency band compared to the higher frequency, the secret data is embedded only in the high frequency sub-bands of the IWT domain to achieve a high robustness and imperceptibility results. In other words, data hiding is carried out in the three sub-bands HH, LH and HL (the LL sub-band is excluded). Similar to the spatial domain embedding, the XOR operation is also utilized here.

The embedding process begins with HH sub-band and identify the edge coefficients to start embedding with the strongest edges to the weakest edges. If the HH sub-band is not enough to embed the secret message, then the process moves to the LH sub-band, and then to the HL sub-band.

The implementation of the embedding process is explained in the following steps:

1. Read the cover image and the secret message.

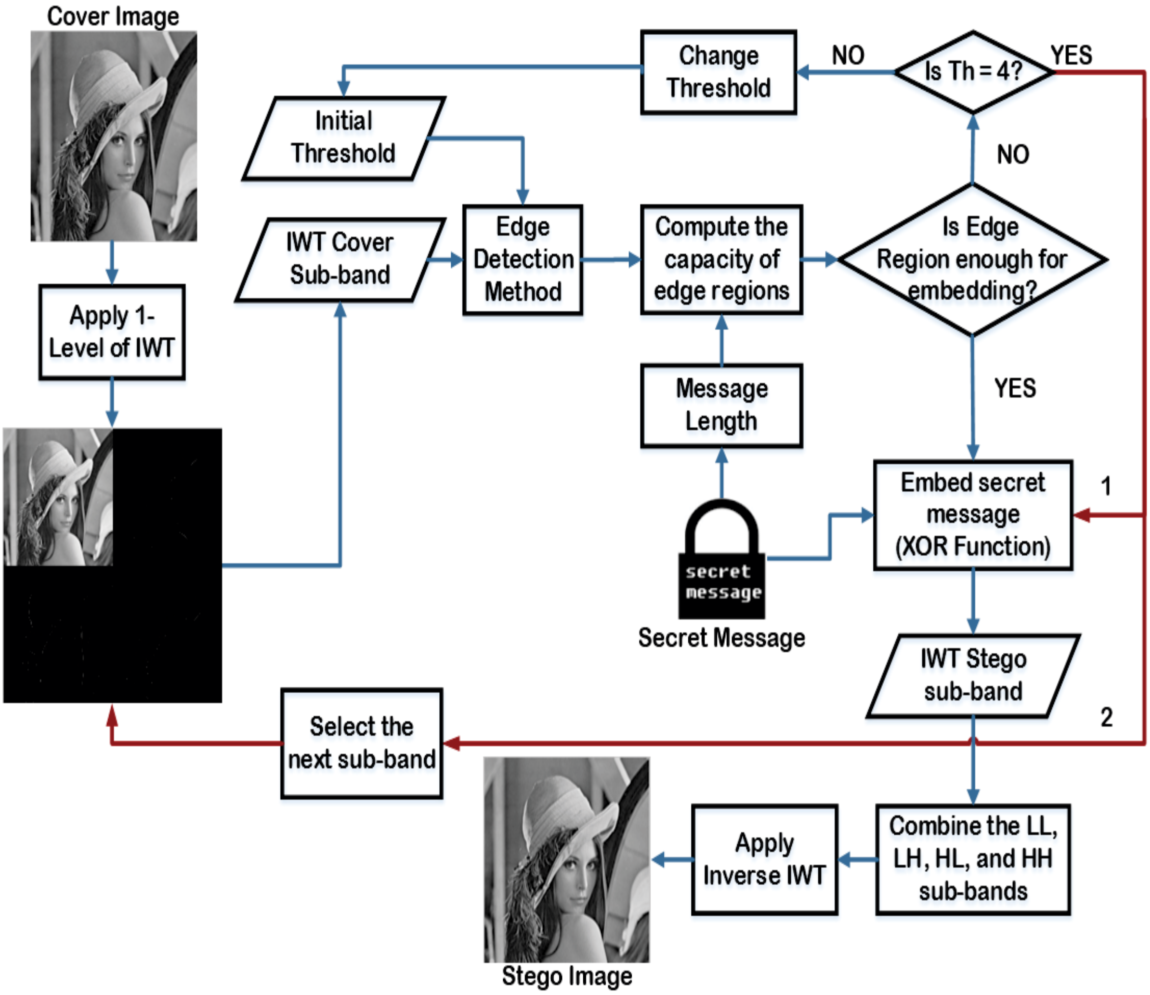


Figure 5.8 Data embedding process in the Integer Wavelet Transform domain

2. Apply the First-Level of IWT on the cover image to decompose the cover image into four sub-bands (LL, HL, LH and HH).

3. Identify edge regions in the high frequency sub-bands (HL, LH and HH). To increase the embedding payload of the wavelet transform method, n LSB from each edge coefficients are utilized in embedding. A higher threshold value (Th) is initialized, which is then decreased based on the number of coefficients needed for embedding and the message length. To identify the edge regions, HH sub-band is divided into non-overlapping blocks of 3×3 coefficients as shown in Figure 5.3a. For each block, the average value (avg) of the four non-shaded coefficients

$(P_{i-1,j-1}, P_{i-1,j+1}, P_{i+1,j-1}, P_{i+1,j+1})$ is calculated. Finally, if the average value (avg) $\geq$ Threshold, the block is selected for embedding.

4. Arrange the edge coefficients into five groups, as shown in Table 5.2. According to Eq. 5.3, if there are enough coefficients to embed the secret message, then embedding process is performed using XOR operation. Otherwise, repeat step 3 after adjusting the threshold value. This process is repeated on the other two sub-bands (LH and HL) until finding enough area for embedding the whole message.

5.2.4.2 The Extraction Process

The extraction process begins with retrieving the threshold value to apply the edge detection method. Edge detection method is performed on the high frequency sub-bands by dividing the sub-band into non-overlapping blocks of size 3×3 to identify the edge area that has been utilized in the embedding process. For each of the three high frequency sub-band (HH, LH and HL), edge blocks are arranged into five groups according to the edge strength. Then, the XOR extraction operations are performed to retrieve n bits from each group as described in Table 5.2.

5.3 Summary

This chapter presented an efficient steganography method that makes use of the fact that the human visual system is less sensitive to changes in high contrast areas of the image and therefore attempts to embed the secret message into edge pixels.

The main contribution of the proposed method is introducing new and efficient edge detection algorithm using non-overlapping blocks that estimates the same edge intensities for the cover and stego images. Also, the incorporation of coding theory makes the embedding more efficient. The proposed method that has been implemented in the spatial and wavelet transform domains to ensure the balance between embedding rate, imperceptibility and security.

CHAPTER 6

Combined Cryptography and Coding based Steganography for Medical Images[1]

This chapter presents an information security scheme conceals coded Electronic Patient Records (EPRs) into medical images in order to protect the EPRs' confidentiality without affecting the image quality and particularly the Region of Interest (ROI), which is essential for diagnosis. The secret EPR data is converted into ciphertext using private symmetric encryption method. Then, the encoded data is embedded in edge pixels of the Region of Non Interest (RONI), which will lead to an improved stego image quality and preserve the ROI from any modification. Two message coding mechanisms have been utilized to enhance the ±1 steganography. The first one, which is based on Hamming code, is simple and fast, while the other which is known as the Syndrome Trellis Code (STC), is more sophisticated as it attempts to find a stego image that is close to the cover image through minimizing the embedding impact.

[1] The contents of this chapter have been published in the Journal of Computer methods and programs in biomedicine, Vol. 127, (2016), entitled "Quality optimized medical image information hiding algorithm that employs edge detection and data coding".

6.1 Introduction

Digital medical images are essential for diagnosis and treatment of many diseases. Therefore, it is extremely important to guarantee secure storage, processing and analysis of medical images without violating the Code of Ethics for Health Information Professionals [209]. As the ever-growing numbers of digital medical images and the necessity to transmit them between different hospitals and clinics for precise diagnosis and treatment planning demand that patients' confidential data to be preserved. In response to this demand, the Digital Imaging and Communication in Medicine (DICOM) standard accepts different encryption methods such as Data Encryption Standard (DES), Triple-DES, and RSA (Rivest, Shamir, Adleman) to protect the privacy and confidentiality of health information [124]. However, encryption methods do not ensure confidentiality of important data, because the transmission of encoded text certainly stimulates intruders' attention, whom may attempt to decrypt it [152]. Information hiding on the other hand is the process of embedding information inside another medium for secure transmission.

Digital steganography has different attractive features to complete the current security measures that can improve the protection for diverse applications. However, image steganography schemes require to be utilized with special consideration for medical imaging systems. Firstly, the steganography method should not compromise the quality of the image, in particular region of interest. Secondly, secret patient data concealed within the cover image should be perfectly extracted [153].

In this chapter, a secure digital medical imaging information system based on a combined steganography and cryptography techniques is introduced. The proposed steganography methodology represents an integration of two main components: two different syndrome codes (STC or Hamming code) that have been utilized to enhance the embedding efficiency by minimizing the distortion function caused due to data embedding, and an accurate method to identically identify sharp regions in both cover and stego images for improving the imperceptibility and to embed larger payload.

This chapter is organized as follows. Section 6.2 introduces the Syndrome Trellis Code (STC). Section 6.3 presents the Hamming code. The proposed methodology is described in Section 6.4. Finally, the summary is given in Section 6.5.

6.2 Syndrome Trellis Code (STC)

Convolutional codes are introduced in [210] and considered one of the most common Error Correcting Codes (ECC). Basically, the encoder of the convolutional codes has memory and outputs depend on the current and previous inputs. A binary convolutional code C is specified by three parameters (N, K, h), where K is the number of inputs, N is the number of outputs and h is the constraint height which represents number of shift registers.

Filler et al. [109, 110] proposed an efficient coding method for steganography, which is called the Syndrome-Trellis Code (STC). It aims to minimize the embedding distortion by finding the closest stego image to the cover image. The STC, which is classified as convolutional code class, represents the codeword by the parity-check matrix. In the binary syndrome-trellis code, the parity-check matrix $\mathbf{H} \in \{0,1\}^{k \times n}$ of size $k \times n$ is represented by placing a small sub-matrix $\hat{\mathbb{H}}$ of size $h \times w$ and shifting it down by one row for a number of times. Equation 6.1 shows an example of a parity-check matrix $\mathbf{H}$ with $k = 4$ and $n = 8$ formed from the sub-matrix $\hat{\mathbb{H}}$ $(h = 2, w = 2)$.

$$\hat{\mathbb{H}} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad , \quad \mathbf{H} = \begin{bmatrix} 1 & 0 & & & & & & \\ 1 & 1 & 1 & 0 & & & & \\ & & 1 & 1 & 1 & 0 & & \\ & & & & 1 & 1 & 1 & 0 \end{bmatrix} \tag{6.1}$$

It is worth to mention that the constraint height h influences the embedding process speed (typically, $6 \leq h \leq 15$).

The syndrome-trellis code is a graph compromising a number of blocks, where each block of the trellis represent one sub-matrix and has $2^h(w + 1)$ nodes structured in a grid of 2^h rows and $w + 1$ columns. A bipartite graph is created between the nodes of the two adjacent columns.

Suppose the cover vector $\mathbf{X} \in \{0, 1\}^n$ is changed to the stego vector $\mathbf{Y} \in \{0, 1\}^n$ after embedding a secret data vector $\mathbf{m} \in \{0, 1\}^k$. The extraction process of the syndrome code is calculated as shown in Eq. 6.2.

$$Ext(y) = Hy^T \tag{6.2}$$

There are many solutions of y that can satisfy Eq. 6.2. The set of all possible solutions of y is called coset of m, which is identified by $C(m) = \{z \in \{0, 1\}^n \mid Hz = m\}$. To select the best solution of y that achieves the minimum distortion, the embedding method computes the additive distortion function for each y from the coset using Eq. 6.3. The additive distortion function D_{st} is used to identify the total effect of the embedding modifications caused by the embedding process, such that the lower value of D_{st} the less detectable by steganalysis [109, 110, 211].

$$D_{st}\left(x, y\right) = \sum_{i=1}^{n} \rho\left(x_i, y_i\right) \tag{6.3}$$

where $\rho\left(x_i, y_i\right)$ is the cost of altering x_i with y_i and $d_i \in [0, \infty]$. To improve the embedding efficiency, the syndrome embedding process is to select y that minimizes the embedding distortion using Eq. 6.4.

$$Emb(x, m) = arg \underset{y \in C(m)}{min} D\left(x, y\right) \tag{6.4}$$

The embedding process is comprised of two stages: a forward and backward stages. The forward stage involves constructions of the trellis based on $H_{k \times n}$ and $Ext(y)$, while the identification of the closest codeword is implemented in the backword stage.

Each path in the trellis begins in the leftmost all-zero state and extends to the right. The edges represent adding $\left(y\left[i\right] = 1\right)$ or not adding $\left(y\left[i\right] = 0\right)$ the ith column of $\mathbf{H}_{k \times n}$ to the current partial syndrome. The calculation of the syndrome trellis is explained step-by-step on each path. For example, in Figure 6.1, the first two edges that connect

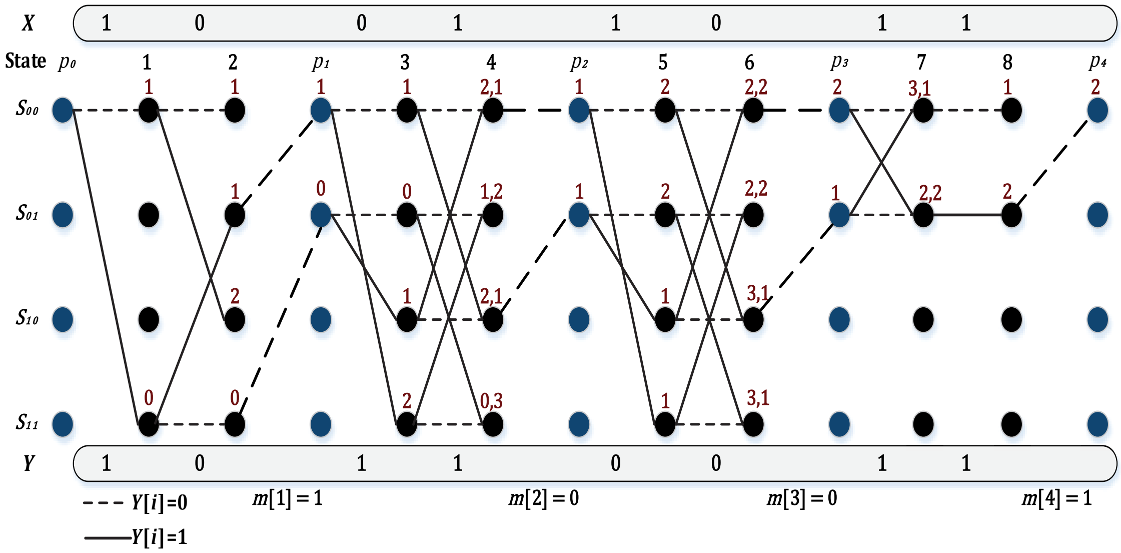


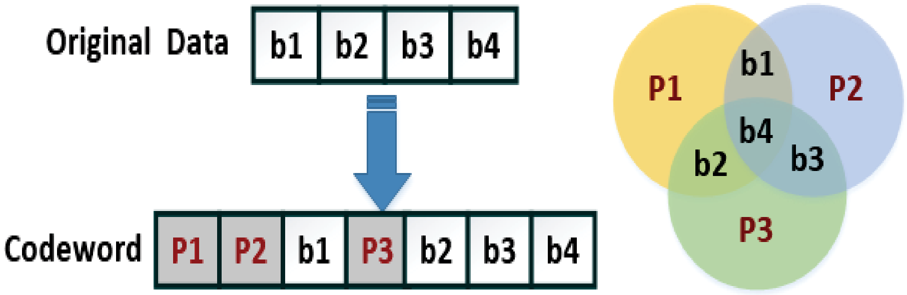
Figure 6.1 Example of STC embedding

the state 00 (S_{00}) from column P_0 with states 11 (S_{11}) and 00 (S_{00}) in the next column, correspond to adding or not adding the first column of $\mathbf{H}[.,1]$ to S_{00}. At the end of the first block, all paths for which the first bit of the partial syndrome does not correspond to the message bit m_1 are finished. Then, we get a new column P_1 of the trellis, which will be utilized as the starting column of the next block. The previous step is repeated at each sub-block of the matrix **H**. To achieve the best match between the stego and cover bits, a weight is computed to each edge in the trellis. Therefore, the path that has the minimum weight is considered the closest match between the cover and stego bits. It is easy to find this path using the backward path from the rightmost state using the edges that were not terminated and create the stego bits (y).

Figure 6.1 represents an example to embedding process using syndrome-trellis when the secret message (m) is 1001 and the cover vectors (X) is 10011011. The path with the minimum weight using the backward step is used to produce the stego vector (Y) which is 10110011.

6.3 Hamming Code

Hamming code is a linear error-correcting codes that detects and corrects one-bit error at the receiver. To fulfil this objective, extra bits (parity check bits) should be added to the original data before transmission to form a codeword. Encoding 4 bits using (7,4) Hamming code requires 3 additional parity check bits; i.e. the original 4 bits are expanded to 7 bits. Figure 6.2 indicates the relationship between original and parity check bits, where b$_i$ refers to an original bit and P$_i$ to a parity check bit. Parity check bits $P_1, P_2, and\, P_3$ are computed using Eq. 6.5.



(a)

(b)

Figure 6.2 (a) Encoding of 4 bits using (7,4) Hamming code (b) The relationship between the original and parity check bits

$$\begin{aligned} P_1 &= b_1 \oplus b_2 \oplus b_4 \\ P_2 &= b_1 \oplus b_3 \oplus b_4 \\ P_3 &= b_2 \oplus b_3 \oplus b_4 \end{aligned} \tag{6.5}$$

Decoding procedure aims to detect and correct errors, and then retrieve the original bits. For error detection and error correction, a syndrome vector (SR) is computed by multiplying the parity check matrix and the received codeword using Eq. 6.6, where the multiplication operation performed using a bitwise AND between two bits followed by an XOR between the terms. If the syndrome vector is not equal to zero, then there exist an error in the received codeword and the decimal value of the syndrome vector indicates error location in the codeword. According to the codeword arrangement, the variable in the fourth digit is inverted to its complement.

$$
\begin{aligned}
SR &= H \times W^T \qquad (6.6)\\
\text{where } H &= \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}_2
\end{aligned}
$$

6.4 The Proposed Methodology

Since the Human Visual System (HVS) is less sensitive to modification in sharp image regions compared to uniform regions, the proposed method hides the EPR in the sharp regions by utilizing a simple edge detection method. In order to maintain good quality of the generated stego images a syndrome code is utilized. Moreover, to add a second layer of security, we introduce an efficient and simple encryption method to conceal the meaning of EPR.

The flow diagram of our proposed method is illustrated in Figure 6.3. The proposed method begins with encrypting the secret data using the symmetric encryption algorithm, then embeds the ciphertext into the cover image. To retrieve the secret data, the encrypted data is extracted from the stego image using the extraction procedure. Finally, the decryption procedure is utilized to retrieve the original secret data.

The proposed method comprises five main procedures: encryption, edge detection, embedding, extraction, and decryption.

6.4.1 The Encryption Process

To enhance the confidentiality of the patient information records, secret data is encrypted using the symmetric encryption approach before embedding in the medical image. However, the existing encryption methods have many limitation such as high computational cost. Therefore, we present a new and simple encryption approach which comprises three main stages: first round permutation, substitution and second

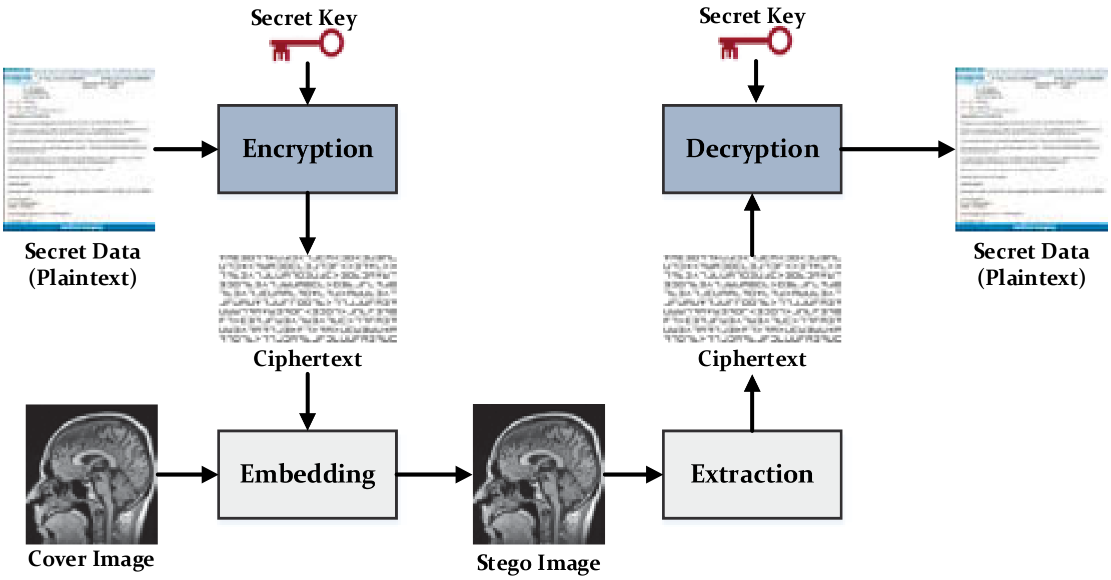


Figure 6.3 The block diagram of the proposed method

round permutation. Permutation operation is a core element of cryptography methods that aim to generate shuffled order of the plaintext. While the substitution operation is the backbone of almost every cryptography method. It seeks to make the statistical relationship between the plaintext and ciphertext as complex as possible.

The implementation of this process is explained in the following steps:

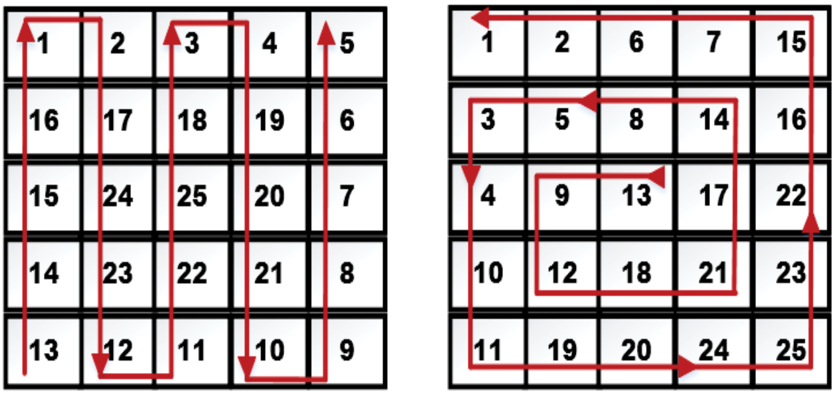
Algorithm 1: Encryption Process.

Inputs: Plaintext (PT), key1 (X), key2 (Y), key3 (Z).

Output: Ciphertext (CT).

(1) Permutation 1$^{\text{st}}$ Round (T_1): The first round permutation is illustrated in Figure 6.4a with 25 characters organized into a 5×5 matrix. Each entry in the matrix represent a character index within a 25 data block. These indices are written in spiral order, then read column by column as shown in Figure 6.4a.

(2) Substitution Round: We present a simple mathematical function to implement a poly-alphabetic substitution cipher, where each ASCII code is mapped to many



1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

(a)

1	2	6	7	15
3	5	8	14	16
4	9	13	17	22
10	12	18	21	23
11	19	20	24	25

(b)

Figure 6.4 Permutation boxes (a) box-1 and (b) box-2

ASCII code symbols using Eq. 6.7.

$$CT = \left[\Big((PT - X) \times Y \Big) \, mod \; 256 \right] \oplus Z \tag{6.7}$$

where CT is the ciphertext, PT is the plaintext, X is a group of four values ranging from 1 to 255, repeated as many times as necessary, Y is the second key.

Y value is between 1 and 255, such that Y and 256 is relatively prime, i.e. greatest common divisor (Y, 256) = 1. Finally, Z is a sequence of binary digits that is extracted from the cover pixels. To reduce the shared information between sender and receiver, key values are derived from the cover image.

(3) Permutation 2^{nd} Round (T_2): The order of the output from the substitution stage are rearranged using Box-2 as shown in Figure 6.4b. For example, we want to encrypt the following secret data "There is a negative sign.". We select four pixels from specific locations to be assigned to the first key (X), and $Y = 3$. An illustration of the data encryption process is shown in Table 6.1.

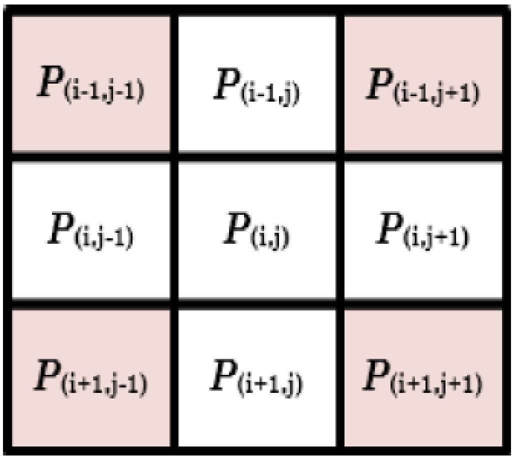
6.4.2 Edge Detection

It is well known that embedding data in sharp contrast areas of an image is less noticeable by the human eye compared to uniform areas [81, 212]. Consequently, patient's information is embedded in edge areas in order to produce a high quality stego

Table 6.1 An illustration of data encryption process

PT	Box-1 (T$_1$)	ASCII (PT)	X	PT−X	(PT−X)*Y	[(PT−X)*Y] mod 256	Z	CT	Box-2 (T$_2$)
T	e	101	113	−12	−36	220	139	87	185
h	g	103	92	11	33	33	148	181	143
e	a	97	10	87	261	5	132	129	179
r	t	116	1	115	345	89	130	219	205
e	T	84	113	−29	−87	169	122	211	157
	h	104	92	12	36	36	115	87	71
i	i	105	10	95	285	29	120	101	198
s	n	110	1	109	327	71	107	44	44
	g	103	113	−10	−30	226	109	143	211
a	n	110	92	18	54	54	114	68	129
		32	10	22	66	66	130	192	219
n	i	105	1	104	312	56	139	179	68
e	.	46	113	−67	−201	55	142	185	192
g	v	118	92	26	78	78	136	198	188
a	e	101	10	91	273	17	137	152	182
t	r	114	1	113	339	83	151	196	198
i	e	101	113	−12	−36	220	155	71	164
v		32	92	−60	−180	76	129	205	148
e	s	115	10	105	315	59	135	188	204
	a	97	1	96	288	32	150	182	196
s		32	113	−81	−243	13	144	157	152
i	s	115	92	23	69	69	137	204	101
g	i	105	10	95	285	29	137	148	87
n		32	1	31	93	93	155	198	181
.	e	101	113	−12	−36	220	120	164	87

image. If adopting one of the existing edge detection methods to identify and embed in the edge regions, then the generated stego image will be slightly different from the original cover image. Thus, when trying to perform edge detection on the stego image to identify the edges (or produce the edge image) for the purpose of extracting the message, some of the identified edges will not exactly match the original ones, and hence, there is no guarantee that all pixels used to extract the message will be identical to those used in embedding it. Therefore, we utilized the proposed edge detection algorithm (Section 5.2.1) to identify the edge regions on the cover image, such that



$P_{(i-1,j-1)}$	$P_{(i-1,j)}$	$P_{(i-1,j+1)}$
$P_{(i,j-1)}$	$P_{(i,j)}$	$P_{(i,j+1)}$
$P_{(i+1,j-1)}$	$P_{(i+1,j)}$	$P_{(i+1,j+1)}$

Figure 6.5 An example of a 3 × 3 block

the two edge images generated using the cover and stego images are identical. This will enable the correct extraction of the concealed message from the stego image. The algorithm starts by dividing the image into non-overlapping blocks that would be individually evaluated and then categorised as either edge or non-edge blocks.

To discover the edge regions, the cover image (C) is divided into non-overlapping blocks of size 3×3 pixels, an example of which is shown in Figure 6.5. The edges in an image are categorized into horizontal (H), vertical (V), and two diagonal directions ($D1$ and $D2$). For each block, the magnitude for each direction can be computed by finding the absolute mean difference between the four shaded pixels (for H and V), or between two of the shaded pixels (for $D1$ and $D2$). Edge magnitude can be computed using Eq. 5.1. The edge magnitude of each block (e) is the maximum of the four edge values. Finally, if $e > Threshold$, the block is considered as an edge block, otherwise it is not an edge block.

6.4.3 The Embedding Process

The flow diagram of the embedding procedure is illustrated in Figure 6.6. The embedding procedure starts by dividing the cover image into ROI and RONI. To identify the edge region, a high value is initially assigned to the threshold variable, which is then modified based on the message length and the number of pixels needed. A STC is used to encode the secret data and then embed it into the identified edge pixels. The implementation of the embedding process is explained in the following steps.

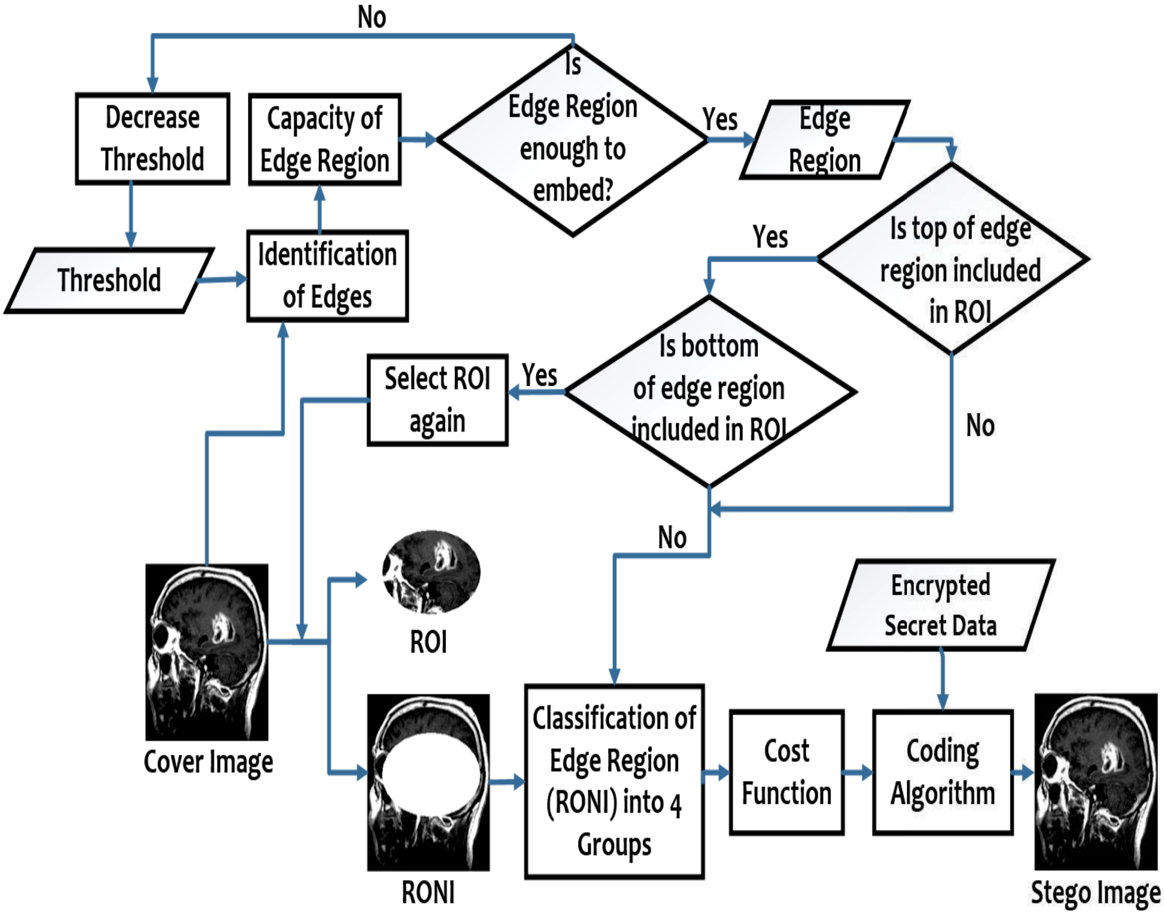
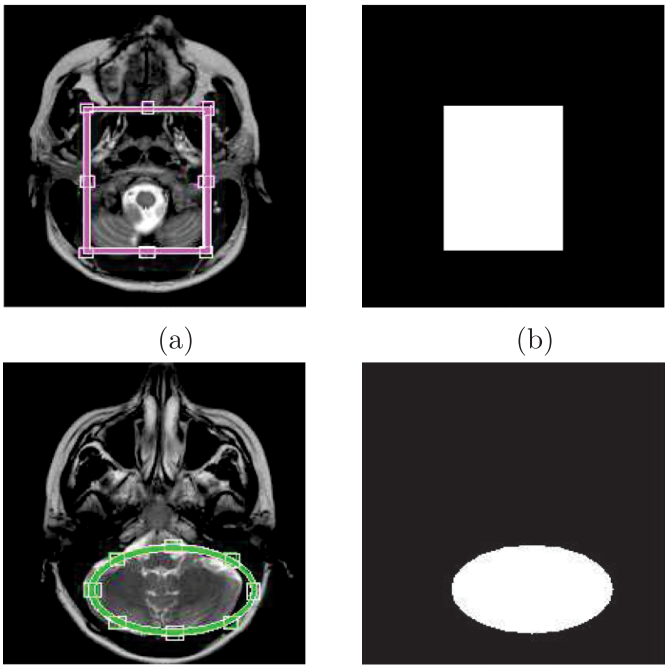


Figure 6.6 The block diagram of embedding process

Algorithm 2: The Embedding Procedure.

(1) Identification of ROI and RONI: Medical image usually contains a particular region that is referred to as the Region of Interest (ROI). It is quite important to protect this region from any modifications in order not to compromise the diagnosis. A rectangular or ellipse ROI is identified by the user manually using four-element vector that specifies its initial coordinates and size. After the identification of ROI, the medical image is converted into a binary image, where pixels that have a value of 1 belong to ROI, while those that have a value of 0

belong to RONI, as shown in Figure 6.7. The secret data is embedded only in edge blocks of RONI to avoid making any change to the ROI. The extraction process requires ROI coordinates to identify the embedding region, and therefore, the coordinates of ROI are concatenated with the secret message to form the data that will be embedded.



(a) (b)

(c) (d)

Figure 6.7 (a) and (c) ROI of MRI cover images. (b) and (d) corresponding Binary Image of ROI

(2) Identification of Edges: In order to increase the embedding rate, n LSBs from each edge pixel are used in embedding. Algorithm 5.1 *(Edge Detection)* is applied to detect edge regions. The magnitude value (e) of each block determines the number of bits, n, to be utilized from each pixel as shown in Table 6.2. Accordingly, a high mean value which represents strong edges will carry more bits than a lower mean value. Based on this, we may not need to embed in all blocks of the RONI. The magnitude of each block is compared to the initial threshold value. However, to reduce the computational cost of edge detection algorithm, initial threshold value is set to a value of four. Then, edge blocks are classified into four groups according to the magnitude value (e) as shown in Table 6.2. According to Eq.

6.8, three secret bits are embedded into four bits of the cover image.

$$(4 \times Message\ Length)\,/3) \leq Number\ of\ edge\ bits \tag{6.8}$$

where edge bits are the utilized n LSBs from each edge pixel as shown in Table 6.2.

Table 6.2 Numbers of bits that can be embedded in each of pixels of an edge block based on the group it belongs to

Group	–	G1	G2	G3	G4
Range of group	[0–3]	[4–7]	[8–15]	[16–63]	[64–255]
Threshold	0	4	8	16	64
n (bpp)	0 bit	1 bit	2 bits	3 bits	3 bits
Region	Smooth Region (Unused)	Smooth Region	Smooth Region	Sharp Region	Sharp Region

(3) Comparison between ROI and edge region locations: As the ROI coordinates are embedded in the initial edge blocks, a comparison between the first/last row of the ROI with the first/last edge regions is required to ensure the initial edge blocks are excluded from ROI pixels. The comparison is explained in the following cases:

- Case 1: If the location of the first row of the edge region is below than that of the first row of ROI, the secret data is embedded from top to bottom of the cover image, and the cover pixel $C(1,1)$ is changed to even value to determine the extraction process direction.
- Case 2: If the location of the first row of ROI is below than or equal to the first row of edge region and the location of the last row of ROI is below than that of the last row of the edge region, the secret data is embedded from bottom to top of the cover image, and the cover pixel $C(1,1)$ is changed to odd value to determine the extraction process direction.
- Case 3: If the top and bottom rows of the edge regions are completely included in the ROI, return to step 1 of the embedding process to re-identify the ROI.

In case 3, steps 1 - 2 will be repeated, otherwise we proceed to step 4. The first pixel $C_{(1,1)}$ must be modified so that it indicates the direction of the extraction process. Extract the secret message from top to bottom of the stego image if the $C_{(1,1)}$ pixel is even. Otherwise, extraction process will start from bottom to top.

(4) Classification of edge pixels: To improve the embedding capacity, RONI pixels are classified into four groups ($[4-7], [8-15], [16-63]$, and $[64-255]$) based on the magnitude value, e, of their corresponding edge blocks. Thus, more bits can be embedded in sharp edges compared to less strong edges. Another advantage of this approach is that embedding different number of bits per pixel may improve the security of the message.

(5) Linear Cost Function: Embedding capacity and image quality are two key factors that should be considered when implementing a steganography method. However, some of these factors are conflicting. For example, improving the image quality generally implies decreasing the embedding capacity. We propose here an optimization function that balances between the embedding capacity (secret message length) and number of bits utilized from each group. For example, if the message length is less than the number of pixels that belong to Group 4, then we utilize only 1-bit from Group 4 instead of using 3-bits to improve the imperceptibility. In other words, If we utilize 3 bits from each pixel in Group 4, then the difference between the cover and stego pixels will be in the range $[0-7]$ according to the weight of each digit in the binary system, where the first LSB is of weight 2^0, the second LSB is of weight 2^1, and the third LSB is of weight 2^2.

$$\begin{aligned}\text{Number of edge bits} = \quad & (a \times G_4 \text{ pixels}) + (b \times G_3 \text{ pixels}) + (c \times G_2 \text{ pixels}) \\ & + (d \times G_1 \text{ pixels})\end{aligned} \tag{6.9}$$

where a, b, c and d are integer numbers ranging from 0 to n.

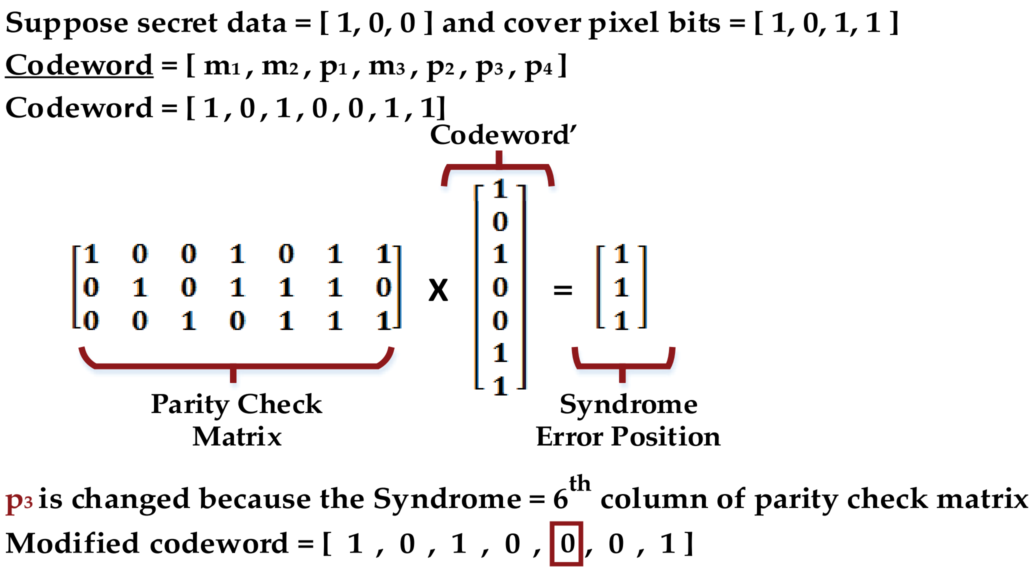
(6) Embedding function: To modify the cover bits according to the message bits, either Hamming or Syndrome-Trellis codes have been utilized to find the stego

bits. Those two coding methods have been considered as one of them is simple and fast while the other is more sophisticated and computationally more expensive.

a) Embedding using a (7,4) Hamming Code: Hamming code is utilized to minimize the number of bits that need to be modified in the stego image. In this stage, Hamming code is used to hide each 3 bits of the secret data (m_1, m_2 and m_3) into 4 cover bits (b_1, b_2, b_3 and b_4) that come from two embedding pixels (selected as described in the previous step). A codeword, W, is formed by arranging the seven bits in the following order: $W = [m_1, m_2, b_1, m_3, b_2, b_3, b_4]$. The parity check matrix and the codeword are multiplied to determine which of the cover pixels need to be modified, as shown in Eq. 6.6. SR indicates the bits that need to be modified based on the codeword W. If SR is equal to the 3rd, 5th, 6th or 7th column of H, then one bit is changed from the cover pixels. For example, when $SR = [0, 1, 1]$ then b_1 is changed. If SR is equal to the 1st, 2nd or 4th column of H, then two bits of cover pixels are changed. For example, if $SR = [0, 0, 1]$ then b_3 and b_4 are changed. If b_1, b_2 or b_4 are converted to its complement, then m_2 or m_3 are modified while they do not require any modification. So, we have selected two cover bits to change given that one of them should be used to compute m_1, m_2, and m_3, and the other cover bit is used to compute m_2 and m_3. If $SR = [0, 1, 0]$ then b_2 and b_4 are changed. Finally, if $SR = [0, 1, 1]$ then b_1 and b_4 are changed. An example of embedding 3 secret bits is shown in Figure 6.8.

b) Embedding using Syndrome-Trellis Code: The syndrome-trellis coding (STC) [109, 110] is utilized for data hiding using Eq. 6.4. As described in section 6.2, the framework of steganography based on STC, the additive distortion function (D_{st}) is defined to choose the codeword having the lowest distortion.

(7) Update the stego image: Modify the stego image using Least Significant Bit Matching (LSBM) as defined in the following embedding function:



Suppose secret data = [1, 0, 0] and cover pixel bits = [1, 0, 1, 1]

Codeword = [m_1 , m_2 , p_1 , m_3 , p_2 , p_3 , p_4]

Codeword = [1 , 0 , 1 , 0 , 0 , 1 , 1]

$$\underbrace{\begin{bmatrix}1 & 0 & 0 & 1 & 0 & 1 & 1\\0 & 1 & 0 & 1 & 1 & 1 & 0\\0 & 0 & 1 & 0 & 1 & 1 & 1\end{bmatrix}}_{\text{Parity Check Matrix}} \times \overbrace{\begin{bmatrix}1\\0\\1\\0\\0\\1\\1\end{bmatrix}}^{\text{Codeword'}} = \underbrace{\begin{bmatrix}1\\1\\1\end{bmatrix}}_{\text{Syndrome Error Position}}$$

p_3 is changed because the Syndrome = 6th column of parity check matrix

Modified codeword = [1 , 0 , 1 , 0 , $\boxed{0}$, 0 , 1]

Figure 6.8 An illustration of embedding 3 secret bits into 4 cover bits using Hamming code

$$Sb_i = \begin{cases} b_i + 2^{pos-1}, & if m_i \neq b_i \quad and\,(k > 0\, or\, b_i = 2 \times (pos-1)) \\ b_i - 2^{pos-1}, & if m_i \neq b_i \quad and\,(k < 0\, or\, b_i = 255 - 2 \times (pos-1)) \\ b_i, & if m_i = b_i \end{cases} \qquad (6.10)$$

where Sb_i is the i^{th} stego bit obtained by using LSBM , b_i is the modified cover bit produced from step 6, k is the random variable with uniform distribution on $\{+1, -1\}$ and pos the right most i^{th} LSBs of b_i ranging from 1 to 3. Note that LSBM has been chosen over LSB replacement, as according to [213] it proved to be more efficient.

6.4.4 The Extraction Process

Algorithm 3: The Extraction Procedure.

(1) Extraction of the shared information: Identify the extraction direction from the first pixel $S(1,1)$.

(2) Identification of Edges: Identify edge regions in the stego image using Algorithm 5.1 (*EdgeDetection*) and the extracted threshold value.

(3) Identification of ROI and RONI: To divide the stego image into ROI and RONI, coordinates of ROI should be extracted from the initial edge blocks using the following XOR operations:
$m_1 = b_1\prime \oplus b_2\prime \oplus b_4\prime$
$m_2 = b_1\prime \oplus b_3\prime \oplus b_4\prime$
$m_3 = b_2\prime \oplus b_3\prime \oplus b_4\prime$
Where $(b_1\prime, b_2\prime, b_3\prime\, and\, b_4\prime)$ are the stego bits and $(m_1, m_2\, and\, m_3)$ are the secret bits.

(4) Classification of edge region: RONI pixels are classified into groups based on the edge mean value of each block as shown in Table 6.2.

(5) Linear Cost Function: Apply the cost function to determine how many bits are used from each group.

(6) *Secret message extraction:* extract the secret data bits (m_1, m_2 and m_3) using the previous XOR operations when the embedding function applied the Hamming code. Otherwise, if the STC was performed, then extract the secret message using Eq. 6.2.

6.4.5 Decryption Process

Algorithm 4: The Decryption Procedure.

(1) Inverse Permutation of Round 2 (T_2^{-1}): Apply the inverse permutation (T^{-1}) on the extracted ciphertext to get the original order. The inverse permutation is defined in the following sequence.

T$_2^{-1}$ = [25, 24, 10, 11, 9, 23, 22, 8, 2, 12, 13, 3, 1, 7, 21, 20, 6, 4, 14, 15, 5, 19, 18, 16, 17].

(2) Inverse Substitution: An inverse substitution function should be applied to replace the ciphertext with the original text (plaintext). The corresponding decryption function is defined in Eq. 6.11.

$$PT = \left(\left[(\, CT \oplus Z \,) \times Y^{-1} \right] + X \right) mod\ 256 \tag{6.11}$$

where CT is the ciphertext, PT is the plaintext, X is a group of four values ranging from 1 to 255, repeated as many times as necessary, and Y^{-1} is the multiplicative inverse of the second key (Y), in the range 1 to 255, such that $\mathrm{Y.Y}^{-1} \equiv 1$ mod 256.

(3) Inverse Permutation of Round 1 (T_1^{-1}): Finally, inverse permutation of round 1 is performed to get the correct order of the plaintext. The inverse permutation is defined in the following sequence.

T_1^{-1} = [5, 6, 15, 16, 25, 24, 23, 22, 21, 20, 11, 10, 1, 2, 3, 4, 7, 14, 17, 18, 19, 12, 9, 8, 13].

6.5 Summary

This chapter handles the security issue of patient's information in the digital medical system. It presented an efficient combination between cryptography and information hiding techniques in order to ensure the security and privacy of patients' information through concealing the meaning of the secret data and its existence. Because medical images have to be carefully processed, as introducing modifications to their important regions, known as the region of interest (ROI), may impact diagnosis of patients' conditions, we have refrained from making any modifications to the ROI and developed our algorithm to conceal the secret data in the Region of Non Interest (RONI). Moreover, based on the characteristic of the human visual perception, we focused on embedding data into the sharp edges of the RONI, as this would attract less attention from intruders about the existence of secret data in the image. To further enhance the embedding efficiency and increase data security, we incorporated a coding algorithms that helped in reducing modifications to the original (cover) images.

CHAPTER 7

Medical Image Segmentation based on Clustering Fusion[1]

This chapter presents an efficient fully-automatic brain tissue segmentation algorithm based on a clustering fusion technique. In the training phase of this algorithm, the pixel intensity value is scaled to enhance the contrast of the image. The brain image pixels that have similar intensity are then grouped into objects using a superpixel algorithm. Further, three clustering techniques are utilized to segment each object. For each clustering technique, a neural network (NN) model is fed with features extracted from the image objects and is trained using the labels produced by that clustering technique. In the testing phase, pre-processing step includes scaling and resizing the brain image are applied and then, the superpixel algorithm partitions the image into multiple objects (similar to the training phase). The three trained neural network models are then used to predict the respective class of each object and the obtained classes are combined using majority voting.

[1] The contents of this chapter have been published in the Journal of Neurocomputing, Vol. 275, (2018), entitled "A Clustering Fusion Technique for MR Brain Tissue Segmentation"

7.1 Introduction

The segmentation process of brain images aims to divide the brain image into three main tissues: white matter (WM), grey matter (GM) and cerebrospinal fluid (CSF) [193]. The objective of segmentation is to simplify the actual illustration of an image into another format, which is easier to understand and analyze [9]. Basically, image segmentation is useful to define boundaries between the brain tissues as well as assigning a unique label to each pixel in the image.

Despite image segmentation has distinct benefits, there are various challenging issues associated with image segmentation algorithms, such as development of a common approach that can be used to all image types and applications. The image quality plays an important role in producing accurate segmentation. However, MR images obtained from different MRI scanners are prone to image intensity-related artefacts, such as image noise or the bias field effect, which are highly dependent on the magnetic field strength [161, 163]. Also, the selection of a suitable method for certain image types can be quite challenging. Because of the aforementioned reasons, there is no universally accepted technique for designing MR image segmentation [162].

In recent years, there has been a growing interest in developing an image segmentation technique based on clustering techniques. According to [214], clustering is considered one of the most popular and efficient techniques for image segmentation. Common clustering methods are the k-means, the fuzzy c-means (FCM), Gaussian mixture model (GMM), and self-organized map (SOM). Each one of these methods has its own advantages and disadvantages. For instance, the k-means method is fast and easy to implement, but it is strongly affected by outlier points and sensitive to initialization [215]. FCM, on the other hand, is less sensitive to initialization, while SOM achieves comparatively better results for overlapped classes when compared to the k-means method; however, the computational cost of FCM and SOM is high. Obviously, it is difficult to have one clustering method that combines all the strength points of the existing clustering techniques in a single method. This motivated this research to explore the possibilities of incorporating numerous clustering techniques to produce the final segmentation result that possesses two main properties; namely, enhanced accuracy and reduced computational cost for the testing phase.

In order to attain the salient features of a number of clustering methods, a fully-automatic segmentation method for MR images based on the concept of clustering fusion and neural networks (NN) is presented. In this process, the image is first divided into superpixels which are used by a number of different clustering algorithms to produce the segmentation results. A neural network model is then trained using the results of each clustering algorithm and the obtained results of the different neural network models are combined to produce the final clustering results for each superpixel.

This chapter is organized as follows. Section 7.2 presents three different clustering techniques. Section 7.3 introduces the Artificial Neural Networks (ANNs). The proposed methodology is described in Section 7.4. Finally, the summary is given in Section 7.5.

7.2 Clustering Techniques

Clustering techniques are mostly unsupervised learning algorithm not reliant on labelled data. The clustering algorithm divides the image into non-overlapping classes with similar intensities based on image features. Moreover, the unsupervised clustering technique strongly based on the initialization and image features to obtain an appropriate result. Usually, the clustering algorithms used are k-means, fuzzy c-mean (FCM) and self-organizing map (SOM).

7.2.1 K-means

K-means is one of the most popular unsupervised clustering technique that separates the input data into groups based on their distance from each other. It is simple and fast to apply on images with large data points [216]. It consists of two main steps: (i) generating a new grouping by assigning each data point to the closest cluster centre and (ii) calculating the k centroid values [217]. The k-means method is described in

Algorithm 7.1.

Algorithm 7.1: k-means

```
Inputs   : Data points ($X$) of size $n \times r$ and number of clusters ($k$).
Outputs : Cluster indices ($C$) of size $n \times 1$.
Randomly initialize cluster centres ($C_j$), where $C_j = \{c_1, c_2, \cdots, c_k\}$;
repeat
    Calculate the distance between each data point of $X$ and cluster centre of
     $C$ using Eq. 7.1;
    Assign each data point ($x_i$) to cluster $c_j$ which has the closest centroid;
    Calculate the new cluster centre using Eq. 7.2;
until the cluster centres no longer change;
```

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} \left\| x_i - c_j \right\|^2 \tag{7.1}$$

where n is the number of data points $(x_1, x_2, \cdots, x_n)$, and k is the number of cluster centres.

$$c_j = \frac{1}{m_i} \sum_{x \in c_j} \mathbf{x} \tag{7.2}$$

c_j is the j^{th} cluster centre and m_i is the number of data points (x) in the j^{th} cluster centre.

7.2.2 Fuzzy c-mean

Fuzzy c-mean (FCM) [184] depends on the primary idea of the k-means (hard) clustering with some modifications. In the hard clustering, each data point should belong to only one cluster, while in the FCM, each data point can belong to more than one cluster according to the degree of membership associated with each data point. FCM aims to improve the membership matrix and cluster centres. The objective function is calculated as shown in Eq. 7.3. The FCM method is described in Algorithm 7.2.

```
Algorithm 7.2: Fuzzy c-means
Inputs  : Data points $(X)$ of size $n \times r$, number of clusters $(c)$, the fuzzifier
          $(m)$, threshold $(\epsilon)$ and maximum number of iterations.
Outputs : Membership matrix $U$ and cluster centres $C$.
$t \leftarrow 0$;
$u_{ij}^t$ initialized randomly;
repeat
    $t \leftarrow t + 1$;
    Calculate $C_j^t$ using Eq. 7.4;
    Calculate $u_{ij}^t$ using Eq. 7.5;
until $\|u^t - u^{t+1}\| < \epsilon$, or max iteration is reached;
```

$$J = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^{m} \times \left\| x_i - c_j \right\|^2 \tag{7.3}$$

where $\sum_{i=1}^{c} u_{ij} = 1 \,\text{and}\, u_{ij} \in [0,1]\,, \forall j = 1, \cdots, n$

n is the number of data points $(x_1, x_2, \cdots, x_n)$, c is the number of cluster centres and u_{ij} is the membership degree of data point x_i to cluster c_j.

The cluster centre and membership matrix values are defined in Eqs. 7.4 and 7.5 respectively.

$$c_j = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m} \tag{7.4}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \tag{7.5}$$

where m is the fuzzier that determines the level of cluster fuzziness (in our proposed method, m is set to 2). Also, $m \in \mathbb{R}$ with $m \geq 1$.

7.2.3 Self Organizing Map

Self-organizing map (SOM) [185] is an unsupervised learning neural network. SOM has a feed-forward structure and, like other types of neural network, does not require the targeted output be specified because it divides data point into groups by learning from the data itself instead of constructing a rule set. The map contains two layers: the input and output (competitive) layers. In the input layer, each node corresponds to single input data, while the output layer is organized into a two-dimensional grid of competitive neurons, as shown in Figure 7.1. Each input node is connected to the output node by adjustable weight vector and is updated in each iterative process.

The winner neuron, also is known as best matching unit (BMU), is determined by selecting the minimum Euclidean distance between the input data and weight vector at each iteration. SOM also utilizes the neighbourhood function. So, when the node wins a competition, the node neighbours are also updated. Let $X = \{x_1, x_2, \cdots, x_N\}$ be the input data of size $N \times 1$ and w_{ij} is the weight vector of the node x_i. The winner neuron is computed using Eq. 7.6.

$$c = \|x_i - w_{ij}\| \tag{7.6}$$

The winner neuron and its neighbors are updated using Eq. 7.7.

$$w_{ij}(t+1) = w_{ij}(t) + H_{ci}\Big[x_i(t) - w_{ij}(t)\Big] \tag{7.7}$$

where w_{ij} and $w_{ij}(t{+}1)$ are the old and new adjusted weight for the node x_i respectively, t describes the iteration number of the training process, $x_i(t)$ is the input data at iteration t, and H_{ci} is the neighbourhood function for the winner neuron c, which is calculated using Eq. 7.8.

$$H_{ci} = \alpha\left[\exp^{(-\frac{\|r_c - r_i\|^2}{2\sigma(t)^2})}\right] \tag{7.8}$$

where α is learning rate, r_i and r_c are the positions of the node i and the winner node c in the topological map (output space) respectively, $\|r_c - r_i\|$ is the distance between the i and winning neurons and σ is the search distance (the number of neighbourhood pixels).

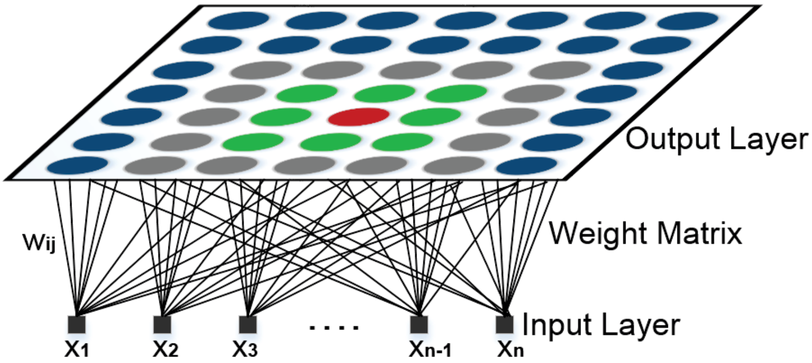


Figure 7.1 Self-Organizing Map (Rectangular Topology)

The SOM algorithm can be summarized as follows:

1. Initialization phase: Initialize random weight to all the nodes .
2. Competitive phase: Select the BMU by examining all the grid nodes with the input data using Eq. 7.6. The minimum Euclidean distance presents the highest matching.
3. Cooperation: The neighbourhood function of the BMU is computed using Eq. 7.8. The radius of neighbours decreases over iteration.
4. Learning stage: Update the weight vector of the winning neuron and its neighbours using Eq. 7.7.

7.3 Artificial Neural Networks

An artificial neural network (ANN) is an information processing mechanism which imitates the biological nervous system in processing information. It comprises of large number of interconnected processing elements, which is also known as neurons, which are working to resolve a particular issue. An ANN system learns by sitting different examples, in order to be configured for a later specific application during a learning process. Learning within ANNs systems includes modulation to the existence connections between the neurons, which is called the synaptic connections [218].

As shown in Figure 7.2, NNs are arranged into layers, where each layer consists of a number of interconnected nodes which contains an activation function. Patterns are introduced to the network via the input (first) layer, which connects to one or more hidden layers and the processing is done through weighted connections. Also, the hidden layers are connected to the output layer where the ANNs output is considered as weighting graphs.

The supervised learning process comprises of having the desired output for specific input set. In other words, each training sample contains the input data and its corresponding (desired) outputs [219].

In the supervised learning strategy, the weights and thresholds of the NN are consistently modified through the use of comparative activities, performed by the learning process itself, that observe the inconsistency between the generated and desired outputs, utilizing this distinction on the alteration process. The network is counted trained when this error is within a reasonable range [219].

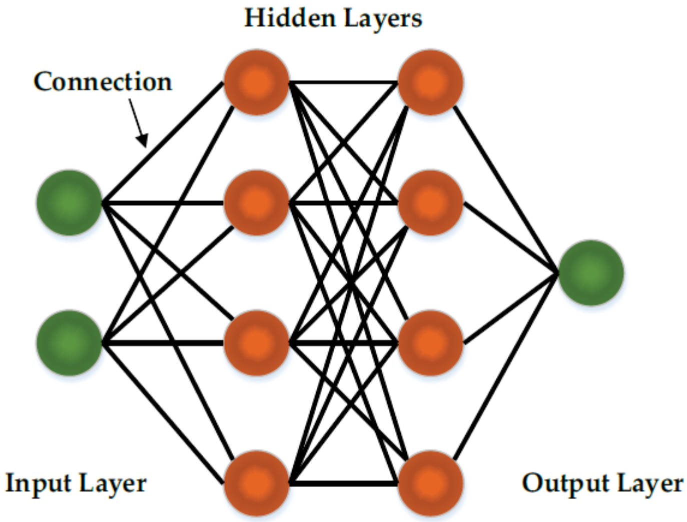


Figure 7.2 Architecture of neural network

7.4 The Proposed Methodology

This chapter demonstrates a brain tissue segmentation method by incorporating various clustering techniques and an artificial neural network (ANN). This chapter demonstrates a brain tissue segmentation method by incorporating various clustering techniques and an ANN. In general, the goal of the proposed method is to generate individual segmentation using neural networks that learn from each clustering technique separately and then combine the segmentation results of those neural networks using a majority voting approach. This will help to enhance the overall segmentation performance and reduce the testing computational cost in comparison to the original clustering algorithms. The proposed method is categorized into two main phases: training and testing. Each phase involves several steps as shown in Figures 7.3 (training) and 7.10 (testing) respectively.

7.4.1 Training Stage

7.4.1.1 Pre-processing

The pre-processing phase is mainly introduced to enhance the MR segmentation of the cerebrospinal fluid (CSF) region. The preprocessing and enhancement step consists of image scaling and resizing.

- Image scaling: In order to enhance the contrast of the image, the image value distribution is scaled to cover a wide range from 0 to 255. Eq. 7.9 is applied to generate the new image.

  $$\begin{aligned} \text{Scale pixel} &= a + ((\text{Original pixel} - c) * F) \\ \text{where } F &= \frac{b-a}{d-c} \end{aligned} \tag{7.9}$$

  where a and b are the target minimum and maximum grey-levels respectively and the original grey-levels fall in the range $[c, d]$.

- Image resizing: The dimension of the original and ground truth images are doubled (from 256×256 to 512×512) by replicating each pixel into a block of size

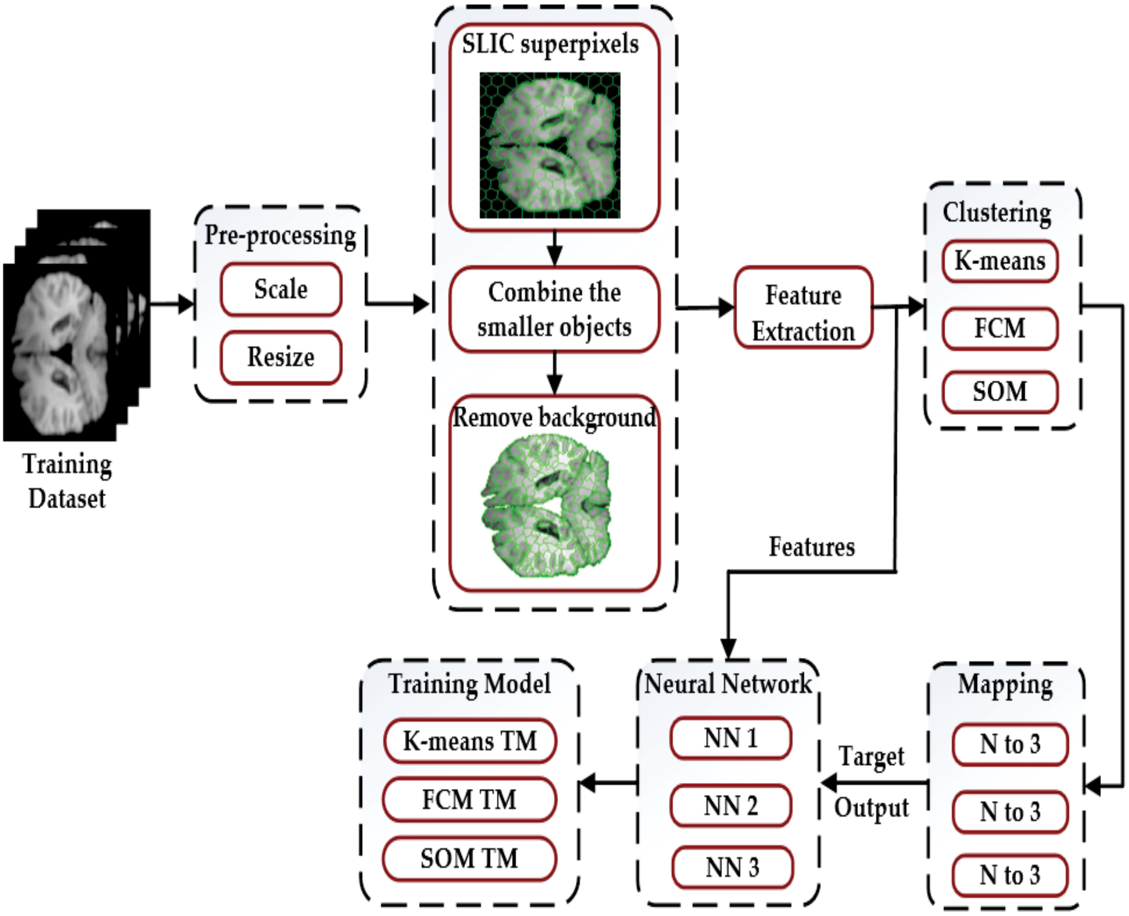


Figure 7.3 The training phase of the proposed method

2 × 2. Most of the existing algorithms have poor CSF classification performance compared to that of GM and WM. The CSF class covers a small area that has an average ratio of 2.10 in the brain image compared to the GM and WM, whose average ratio is 54.22 and 43.68 respectively. When using the superpixels algorithm, classes might be combined with other classes. Therefore, the objective of doubling the size of the original and ground truth images is to prevent the merging of CSF objects with the other objects to make it clearer for the clustering step.

7.4.1.2 Pre-segmentation

- Simple linear iterative clustering (SLIC) superpixels [220, 221]: The SLIC performs a constrained search space in the neighbourhood of the cluster centre by integrating intensity and spatial location to generate the superpixels.

  The scaled image is over-segmented into objects using the SLIC superpixels algorithm [220] as shown in Figure 7.6a. In the beginning, the required number of superpixels, k, is manually defined. The initial superpixel cluster centres $C_j = [I_j, x_j, y_j]^T$ with $j = \{1, 2, \cdots, k\}$, where I_j is the pixel intensity and (x_j, y_j) is the coordinator of the centre. The initial cluster centres are sampled on a regular grid spaced S pixels. In order to have approximately equal size of superpixel, the grid size is set to $\sqrt{\frac{N}{k}}$, where N is the total number of image pixels. Then, the centres are moved to the lowest gradient position in a 3×3 neighbourhood to prevent centring a superpixel on an edge or a noisy pixel. The pixels are grouped based on intensity similarity. After each iteration, the cluster centres are modified based on the pixels assigned to that cluster. The iterations continue until the superpixel centres do not change.

  Figures 7.4 and 7.5 illustrate the impact of the compactness and number of superpixel parameters on the boundary adherence ability of superpixels. The compactness parameter (m) controls the shape regularity of the superpixel region (flexibility of superpixel boundaries). It is clear that higher compactness values generate more regular and smoother shapes as shown in Figures 7.4a - 7.4d. However, a higher value of compactness affects the boundary adherence of superpixels. In our experiments, the superpixel size (k) and the compactness (m) set to 1500 and 5 respectively. Also, A small number of superpixels would lead to a large superpixel size as shown in Figure 7.5a.

- Merge the small objects: After dividing the image into several groups based on the SLIC superpixels algorithm, a small object is merged with its neighbour object according to some rules. In the beginning, we need to predict the small objects which have less than 30 pixels. Then, neighbours of the small objects are identified and sorted in ascending order based on their sizes. Finally, the

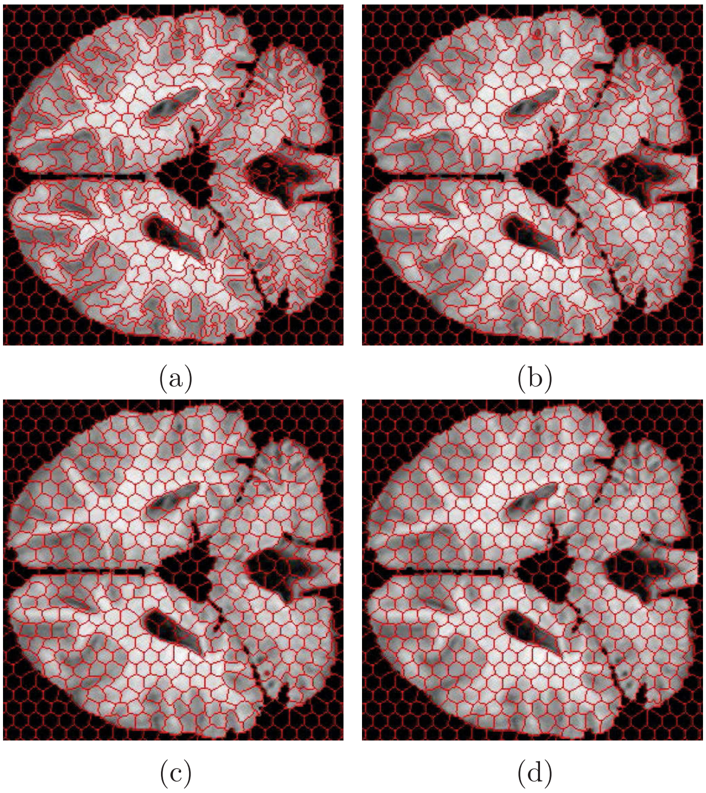


Figure 7.4 Visual illustration of the effect of SLIC superpixel parameters (number of superpixel (k) and compactness (m)) in brain tissue segmentation: (a) $k = 2000$ and $m = 5$, (b) $k = 2000$ and $m = 10$, (c) $k = 2000$ and $m = 20$, (d) $k = 2000$ and $m = 30$

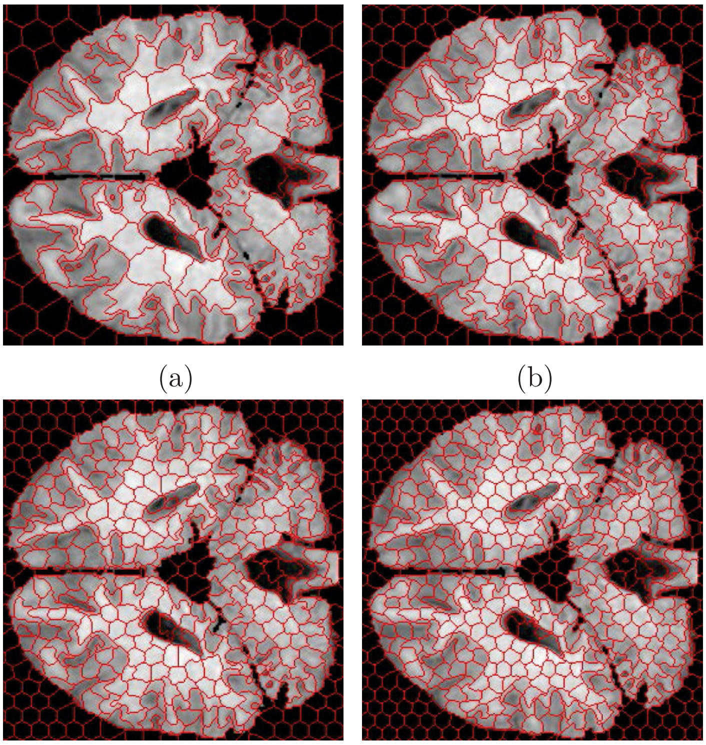
small objects are merged with the biggest neighbour. Figure 7.6b presents the over-segmented image after the merging step.

- Remove the background: The merged image is divided into background and object regions based on Eq. 7.10.

$$g_{(x,y)} = \begin{cases} 1 & \text{if } f_{(x,y)} \geq Th \\ 0 & \text{otherwise} \end{cases} \tag{7.10}$$

Where $g_{(x,y)}$ is the segmented binary image, $f_{(x,y)}$ is the original pixel value and Th is the threshold value.

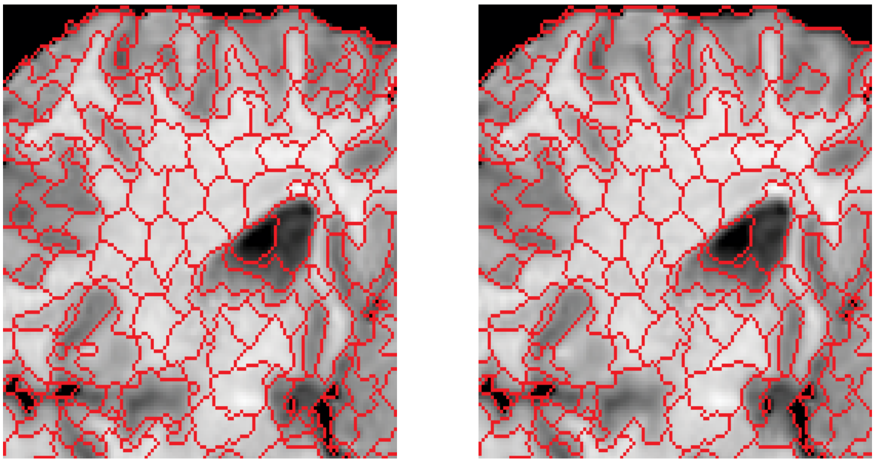
The global threshold value is selected based on the dataset that used for evaluation. The histogram of the background of the training samples are examined to decide the threshold value. The threshold value is identified by $Th = \lfloor($ max. pixel value $-$ min. pixel value $) \times 3\%\rfloor$. The $\lfloor \ \rfloor$ operation is the floor function.



(a) (b)

(c) (d)

Figure 7.5 Visual illustration of the effect of SLIC superpixel parameters (number of superpixel (k) and compactness (m)) in brain tissue segmentation: (a) k = 500 and m = 10, (b) k = 1000 and m = 10, (c) k = 1500 and m = 10 and (d) k = 2000 and m = 10

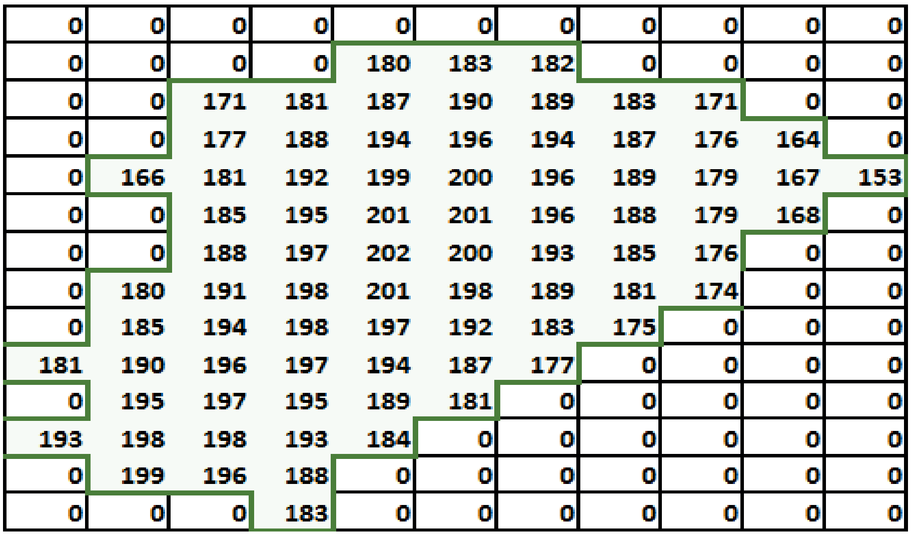


(a) (b)

Figure 7.6 (a) Zoomed area from the SLIC superpixels algorithm and (b) zoomed area of the SLIC superpixel after merging the small objects

7.4.1.3 Feature Extraction

Pixel intensity features are extracted for all objects and fed to three different clustering techniques simultaneously. Hence, the sizes of the over-segmented objects are varied. For each object, the histogram is identified and sorted in ascending order. The five highest frequencies of pixel intensity are extracted. The size of the feature matrix is $k \times 5$, where k is number of objects and five is number of features for each object. Figure 7.7 shows pixel intensities of one object and Figure 7.8 shows its histogram, where the first five shaded intensities (highest frequencies) are the extracted features of that object.

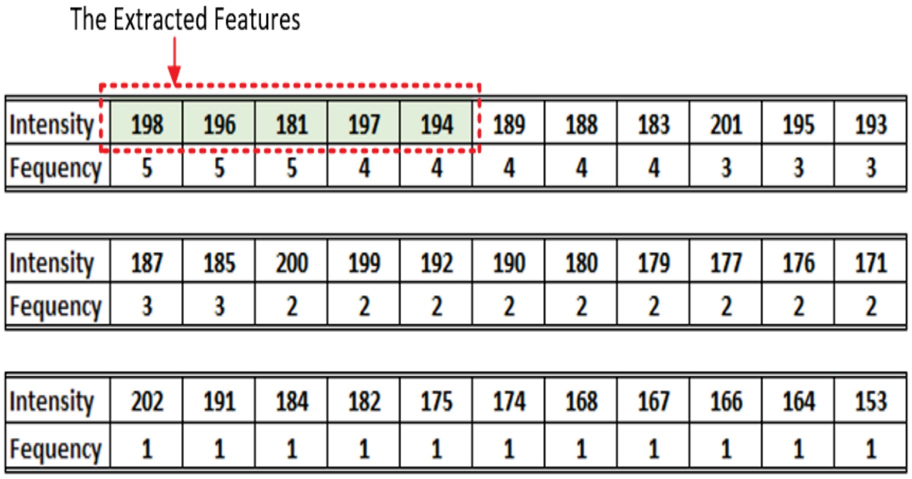


0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	180	183	182	0	0	0	0
0	0	171	181	187	190	189	183	171	0	0
0	0	177	188	194	196	194	187	176	164	0
0	166	181	192	199	200	196	189	179	167	153
0	0	185	195	201	201	196	188	179	168	0
0	0	188	197	202	200	193	185	176	0	0
0	180	191	198	201	198	189	181	174	0	0
0	185	194	198	197	192	183	175	0	0	0
181	190	196	197	194	187	177	0	0	0	0
0	195	197	195	189	181	0	0	0	0	0
193	198	198	193	184	0	0	0	0	0	0
0	199	196	188	0	0	0	0	0	0	0
0	0	0	183	0	0	0	0	0	0	0

Figure 7.7 An example of object pixel intensities

7.4.1.4 Clustering Techniques

A number of base clustering techniques are utilized to divide the brain image into groups. The inputs for each clustering algorithm are features, centre values and the class label (there are three classes for the MR brain images excluding the background). In order to enhance the segmentation accuracy, the number of classes is increased to six to address the issue of intra-class variability. This will help in differentiating between objects that belong to different classes and yet have small differences between their intensity levels (e.g. some regions of the CSF and GM classes). Three base clustering



The Extracted Features

Intensity	198	196	181	197	194	189	188	183	201	195	193
Fequency	5	5	5	4	4	4	4	4	3	3	3

Intensity	187	185	200	199	192	190	180	179	177	176	171
Fequency	3	3	2	2	2	2	2	2	2	2	2

Intensity	202	191	184	182	175	174	168	167	166	164	153
Fequency	1	1	1	1	1	1	1	1	1	1	1

Figure 7.8 The histogram of Figure 7.7

algorithms are chosen to be used in this work, which are k-means, FCM and SOM. A brief description of each of these three algorithms is given below.

7.4.1.5 Matching Classes

For the brain tissue dataset, this step aims at forming three classes from the six classes that were produced in the previous step, where the desired number of brain tissues is three (WM, GM and CSF). In other words, the six classes produced by the segmentation algorithms are grouped to form three classes. The mapping function is described in Algorithm 7.3.

7.4.1.6 Back Propagation Neural Network (BPNN)

The BPNN is considered to be one of the simplest methods for supervised training of multi-layered neural networks. It estimates the non-linear relationship between the input and the output by adjusting the weight values. The back propagation method is a generalization of the least mean square (LMS) algorithm that updates network weights to reduce the mean squared error between the network and targeted outputs. The supervised network is trained using two inputs: the input data points (extracted features) and the targeted outputs. The BPNN algorithm can be summarized as follows [222]:

Algorithm 7.3: Mapping n-classes to 3

```
Inputs  : $k$ objects, clustering output, ground truth image.
Outputs : new class.
$cu \leftarrow \{1, 2, \ldots, 10\}$;
$i \leftarrow 1$;
for $i \leq 10$ do
    Find which objects from the clustering output are equal to class $cu(i)$
      $[row, col] \leftarrow$ find (clustering output == $cu(i)$);
    $class \leftarrow [\ ]$;
    $j \leftarrow 1$;
    for $j \leq 10$ do
        GT $\leftarrow$ Find the class value of object($row_j, col_j$) in the ground truth
          image;
        class $\leftarrow$ [class ; GT];
    class no $\leftarrow$ 3;
    class value $\leftarrow$ [128, 196, 254];
    m $\leftarrow$ 1;
    for $m \leq$ class no do
        [R, C] $\leftarrow$ find (GT == class value($m$));
        Percentage ($m$) $\leftarrow$ Length ($R$) divide by Frequency of class value ($m$) in
          the ground truth;
        Accuracy ($m$) $\leftarrow$ Percentage ($m$) divide by Total of frequency for all
          classes in the ground truth image;
    Assign for $cu_i$ the maximum accuracy value;
```

1. Initialize weights and node offset to random values.

2. Present a training example and propagate it through the network. The training example includes a input vector $(x_0, x_1, \cdots, x_{n-1})$ and desired output vector $(d_0, d_1, \cdots, d_{n-1})$.

3. Compute the actual output (y) using the sigmoid non-linearity function.

4. Adjust weights starting from the output layer and working backwards (backward pass) to the first hidden layer using Eq. 7.11.

$$w_{ij}(t+1) = w_{ij}(t) + \mu \delta_j x\prime_i \tag{7.11}$$

where $w_{ij}(t+1)$ is the weight from hidden node i or from an input to node j at time t, $x\prime_i$ is either the output of node i or is an input, μ is the gain term and δ_j is an error term for node j. δ_j is calculated using Eq. 7.12.

$$\delta_j = \begin{cases} y_j(1-y_j)(d_j-y_j) & \text{If node } j \text{ is an output node} \\ x_j(1-x\prime_j)\sum_k \delta_j w_{jk} & \text{If node } j \text{ is an internal hidden node} \end{cases} \tag{7.12}$$

where d_j is the desired output of node j, y_j is the actual output of node j, k is over all nodes in the layers above node j and x_j is the input of node j.

5. If the LMS is greater than a predefined threshold value then repeat step 2 to 5.
6. Stop the training process and store the optimal weights.

The three BPNN models are trained to imitate the three clustering algorithms, as this would enable the use of trained models instead of the original clustering methods in the testing stage, which helps reduce the complexity of the system. Therefore, after segmenting the training dataset using the three clustering methods, target outputs that represent the clustering labels for each method are used to train the neural networks that are fed with the extracted features of the superpixel objects as shown in Figure 7.9. The first five images of the dataset are used for training. Thus, a model for brain tissue segmentation is learned from the training dataset for each clustering technique independently.

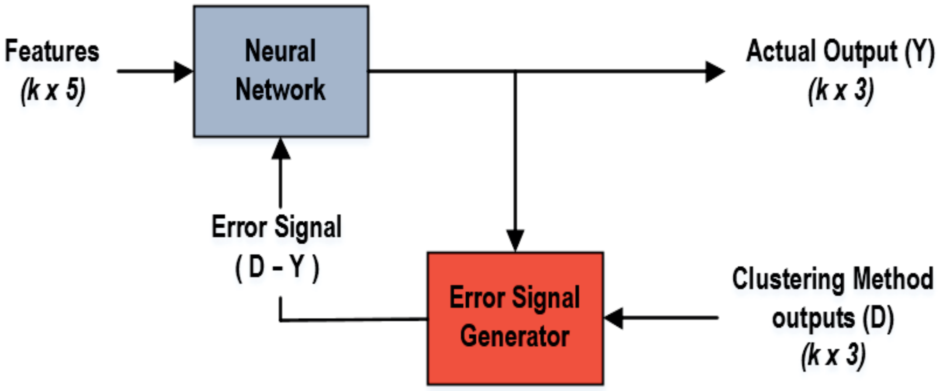


Figure 7.9 The training model of NN under supervised learning

7.4.2 Testing Stage

The flow diagram of our testing stage is illustrated in Figure 7.10. The first three steps are identical to the training stage. The process begins by performing the pre-processing step, which includes contrast enhancement and image resizing. In the next step, the SLIC superpixels algorithm divides the brain image into multiple objects. The subsequent step involves feature extraction, where the five highest frequencies of pixel intensity are extracted from each object. For each clustering technique, the training model is utilized to predict the class for each object. Below are the detailed steps of the testing stage:

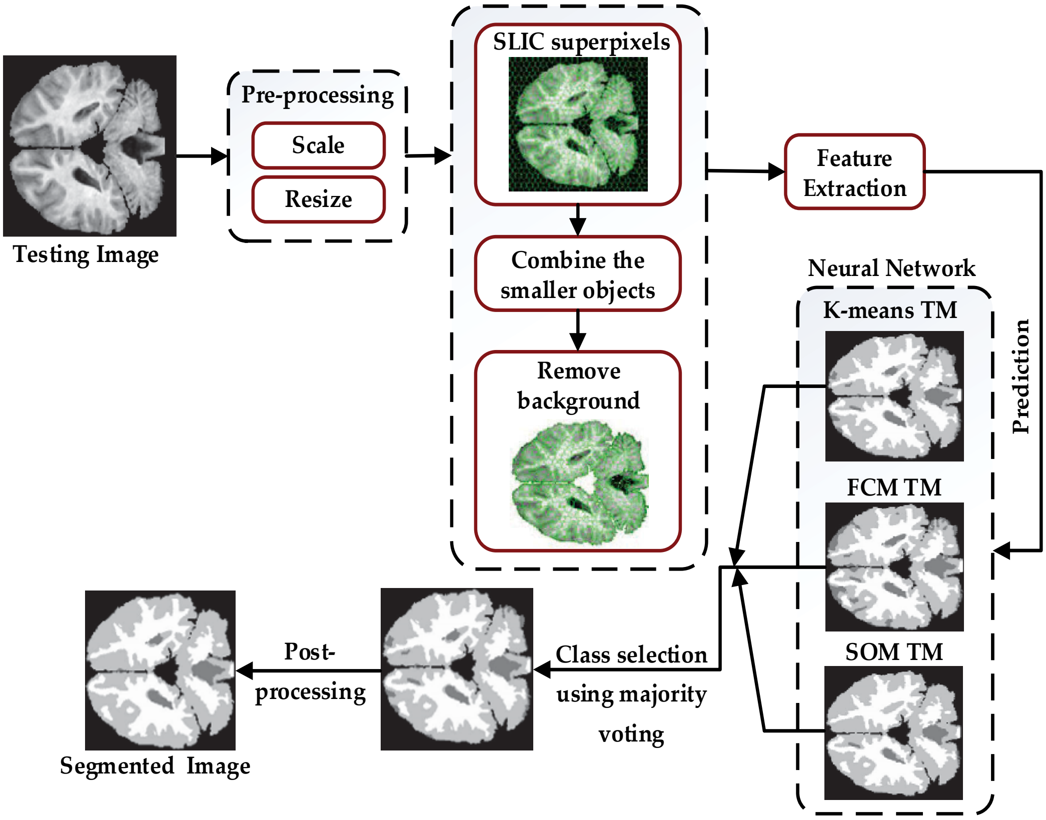


Figure 7.10 The testing phase of the proposed method

1. Pre-processing step: The pixel intensity of the test image is modified using Eq. 7.9 to improve the image contrast. The image size is then increased to help in accurately identifying the CSF region.

2. Pre-segmentation step: The SLIC superpixels algorithm is applied to divide the image into objects. Then, any small object that has less than 30 pixels is merged with its biggest neighbour. Finally, the background is excluded using a global threshold value.

3. Feature extraction: For each object, the five highest frequencies of pixel intensity are extracted.

4. Neural network: The trained NN models are used to predict the segmented image for each of the clustering methods separately.

5. Select the final segmented image: The segmented image is combined using a majority voting approach, where each object is assigned to the class that receives the highest votes.

6. Post-processing step: Correction of boundary labels may be required, especially for under-represented classes, such as the CSF that was found to be confused with GM. To correct the partition result, we need to identify the CSF region and identify their distance from the centre of the image using Eq. 7.13. If the distance (D) of a CSF object is greater than a certain threshold (set here to 80), then this object is reassigned to the GM class.

$$D = \sqrt{\left(\frac{\text{image row}}{2} - \text{avg.}O_{ir}\right)^2 + \left(\frac{\text{image column}}{2} - \text{avg.}O_{ic}\right)^2}, \forall i = 1, \cdots, t \tag{7.13}$$

where O_i is the CSF object, O_{ir} is the row of object O_i, and O_{ic} is the column of object O_i.

Figure 8.15 shows an example of the segmented image before and after applying the post-processing step. Figures 7.11a and 7.11b are identical, while Figures 7.11c and 7.11d are different.

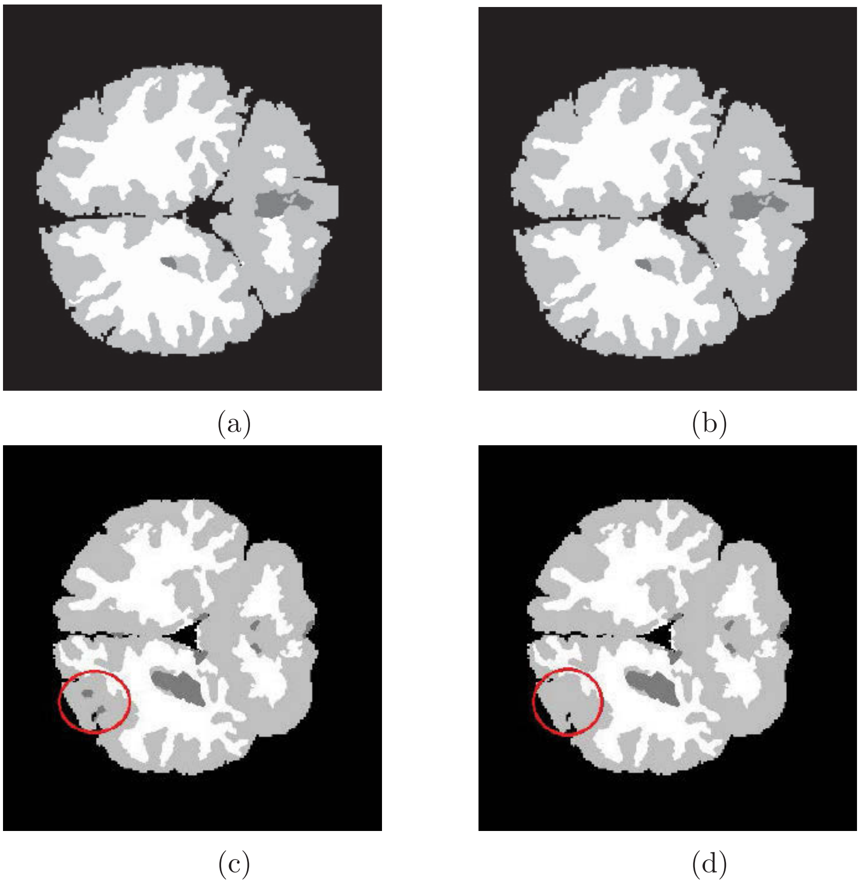


Figure 7.11 Subject 111-2, slice 20: (a) without the post-processing step, (b) with the post-processing step, Subject 205-3, slice 20: (c) without the post-processing step, and (d) with the post-processing step

7.5 Summary

In this chapter, a segmentation method was developed that combines the SLIC superpixel, three clustering techniques and a neural network to divide the MR brain image into three tissues of WM, GM and CSF. The method comprises two stages: training and testing. Both stages start with a pre-processing step to improve the image contrast, as the brain structure is not realized by unique intensities in MR images. This step also incorporates image scaling and resizing, was it has been found that this helps to achieve a better clustering outcome, particularly for the under-represented class of CSF. Then, the brain image is partitioned into objects using the SLIC superpixels algorithm, where the utilization of superpixels proved useful in segmenting complex objects, such as those of brain images. The training stage involves the training of a NN model for each base clustering of the proposed method to imitate its segmentation outcome. Features extracted from the frequencies of pixel intensities are used for this purpose. In the testing stage, the trained NN model of each clustering technique was utilized to predict the class for each object. The outcome of the segmentation results are then combined using majority voting and a post-processing step is used to correct the boundary labels.

The main contribution of the proposed method is introducing an efficient ensemble-based clustering method to enhance the segmentation of MR brain tissues. The testing stage of the proposed method is computationally efficient and more accurate than the segmentation obtained using a single clustering technique.

CHAPTER 8

Experimental Results and Discussions

This chapter presents the results of the steganography and segmentation methodologies mentioned in chapters 5, 6 and 7. Several experiments have been carried out to evaluate the performance of the proposed method, and to compare its performance with some of the existing algorithms. In the beginning, a complete description of the employed datasets is given. Afterwards, the results of the steganography methodologies using general and medical datasets for evaluation are presented. Finally, quantitative assessment of the segmentation methodology is carried out different computing metrics.

8.1 Steganography Performance Evaluation

8.1.1 Image Dataset

BOWS2 (abbreviations of Break Our Watermarking System) database [223] contains 10,000 grey-scale natural images of size 512 × 512 (http://bows2.ec-lille.fr/).

8.1.2 Evaluation

Different evaluation metrics are typically used to measure the performance of steganography techniques. The embedding capacity, embedding distortion (objective test) and security metrics are used to evaluate the overall performance of the proposed methodologies. The proposed methodologies are implemented in MatlabR2014b.

8.1.2.1 Embedding Capacity Evaluation

Embedding capacity is an essential measurement to evaluate the performance of steganography methods. It refers to the amount of bits that can be embedded into the cover image. High embedding capacity is an attractive characteristic that most steganography methods strive to achieve. Embedding capacity is computed using Eq. 8.1 [224]. In our experiments, a uniformly distributed random message is generated.

$$E = \frac{K}{WH}(bpp) \tag{8.1}$$

where K is the maximum number of secret message bits that can be embedded in the cover image, and W and H are the cover image width and height respectively.

8.1.2.2 Embedding Distortion Evaluation

There is no unique method to measure imperceptibility of steganography methods. One of the commonly used measures of imperceptibility is the peak signal-to-noise ratio (PSNR) between the cover and stego images, which is calculated using Eq. 8.2.

$$PSNR = 10\log_{10}\left[\frac{255^2}{MSE}\right](dB) \tag{8.2}$$

where MSE is the mean square error between cover and stego images, which is defined as:

$$MSE = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}(C_{ij} - S_{ij})^2 \tag{8.3}$$

where C_{ij} and S_{ij} are the grey values of pixel (i, j) of the cover and stego images. W and H are the width and height of the cover image (the stego image has the same size).

$PSNR$ quality measures the distortion occurred on the cover image and it does not take HVS into consideration. The weighted peak signal-to-noise ratio ($wPSNR$) is an alternate measurement quality, which is defined using Eq. 8.4. It utilizes an extra parameter called noise visibility function (NVF). $wPSNR$ is roughly equivalent to $PSNR$ for flat areas because NVF is close to one in smooth areas. But for edge regions, $wPSNR$ is higher than $PSNR$, because NVF is close to zero for complex regions, and hence it attempts to reflect how the HVS perceives images.

$$wPSNR = 10\log_{10}\left(\frac{\max(x)^2}{\|NVF(C-S)\|^2}\right)(dB) \tag{8.4}$$

where NVF is defined as:

$$NVF_{(i\,,j\,)} = \frac{1}{1+\sigma^2_{L_{(i\,,j\,)}}} \tag{8.5}$$

where $\sigma^2_{L_{i,j}}$ denotes the local variance of an image in a window of size (3×3) centred on the pixel with coordinates (i, j).

The average difference is a simple and popular image quality evaluation criterion. It is computed by averaging the absolute difference between the cover and stego images, which is calculated as shown in Eq. 8.6 [225].

$$\text{Average Difference} = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H} \mid c_{ij} - s_{ij} \mid \tag{8.6}$$

8.1.2.3 Security Evaluation

Security is an important issue in steganography systems. The goal of steganography is reducing the distortion caused by the embedding process, and accordingly prevent statistical detection of differences between the altered (stego) and original (cover) images. The proposed methodology is evaluated under blind steganalysis method.

Li-110D [226] is one of the most efficient steganalyzers used for detecting spatial domain steganography.

In [226], the blind image steganalysis method is considered as a texture classification. Image textures are classified into regular, near regular, irregular and stochastic. The proposed approach is based on the fact that the embedding process affects the texture of the original (cover) image by generating more stochastic (random) textures.

Li-110D [226] extracts statistical moment features of probability density function (PDF) from the normalized histogram of the local linear transform (LLT) coefficients of the image. These features aim to detect particular alternations of the local texture caused by the embedding process based on the fact that steganography introduces more stochastic textures to the stego images in a fine scale.

It has been proved that many of the steganography techniques mainly affect the medium and high frequencies of the image [227]. Also, medium and high frequency components can be classified as stochastic textures which might be introduced by the embedding process. Since local DCT can offer meaningful visions regarding the image characteristics of the spatial and frequency domains [228]. Li-110D utilized the medium and high coefficient statistics of the local DCT to recognize between the cover and stego images [226].

Three one-dimensional DCT base vectors, $u_1 = [1, 1, 1]^T$, $u_2 = [1, 0, -1]^T$ and $u_3 = [1, -2, 1]^T$, are used to create a set of two-dimensional DCT (T^{DCT}) masks of size 3×3 using Eq. 8.7 [229]. In [226], six 2D DCT masks were only chosen using Eq. 8.8.

$$T_{ij}^{DCT} = \left\{ u_i \times u_j^T | (i\,, j) \in \{1, 2, 3\} \right\} \tag{8.7}$$

$$T_{ij}^{DCT} = \left\{ u_i \times u_j^T | (i\,, j) \in \{(1, 3), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)\} \right\} \tag{8.8}$$

The six local two-dimensional DCT masks are given as follows:

$$T_{13}^{DCT} = \begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{bmatrix} \quad T_{22}^{DCT} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad T_{23}^{DCT} = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{bmatrix}$$

$$T_{31}^{DCT} = \begin{bmatrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ 1 & 1 & 1 \end{bmatrix} \quad T_{32}^{DCT} = \begin{bmatrix} 1 & 0 & -1 \\ -2 & 0 & 2 \\ 1 & 0 & -1 \end{bmatrix} \quad T_{33}^{DCT} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & -4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

In addition to TDCT masks, another four convolution kernels (W) for local transform are introduced which are equivalent to the second-order derivative (T^{SOD}). The four convolution kernels are identified as follows:

$$T_{1}^{SOD} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} \; T_{2}^{SOD} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \; T_{3}^{SOD} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \; T_{1}^{SOD} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

The final local linear transform (LLT) mask includes ten masks, i.e., $T^{LLT} = T^{DCT} \cup T^{SOD}$. Then, the features ($F$) are extracted by convolving the input image (I) with the final local linear transform masks (T^{LLT}) using Eq. 8.9. Finally, the probability mass function (PMF) are produced by calculating the normalized histogram of the extracted features (F).

$$F_i = \left\{ I * T_i^{LLT} | i \in \{1, 2, 3, \cdots, 10\} \right\} \tag{8.9}$$

where the symbol $*$ refers to the convolution operation.

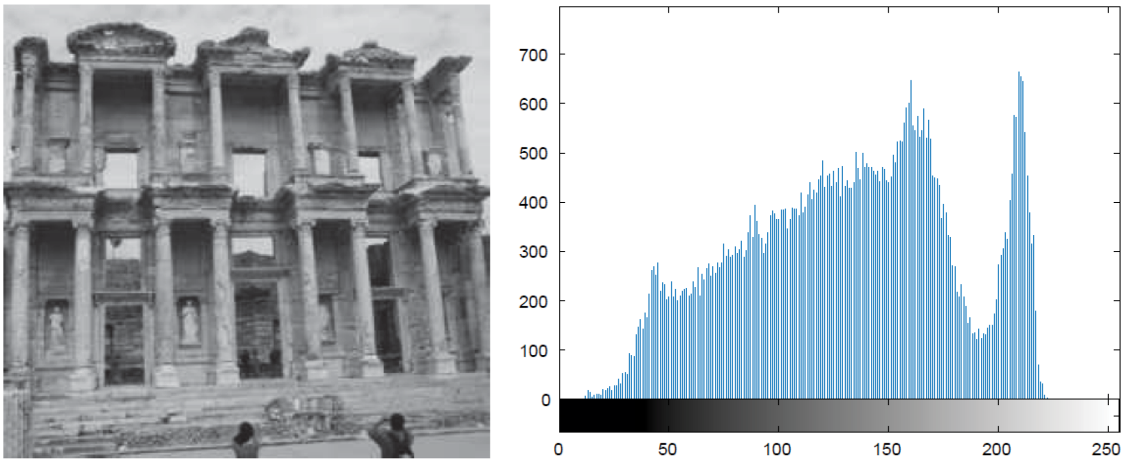
Kullback–Leibler Divergence (KLD) [230] is one of the common security measures to analyze the steganography system. It measures the difference between two probability distributions P and Q. Suppose the cover and stego images are represented by C and S respectively and the probability distribution function of C and S are denoted by p_c and q_s. KLD between two probability functions is computed using Eq. 8.10. The smaller KLD value introduced by the embedding process is the less detectable by steganalysis method.

$$KLD(p_c \parallel q_s) = \sum_{g \in G} p_c(g) \log \frac{p_c(g)}{q_s(g)} \tag{8.10}$$

where $g \in G = \{0, 1, 2, \cdots, 255\}$ is the grey-scale of the cover and stego images.

8.1.3 Image Steganography Methodology Results

This section presents the experimental results of the proposed methodologies, which is introduced in Sections 5.2.2, 5.2.3 and 5.2.4. The proposed methods have been tested using the BOWS2 database [223].



(a) (b)

Figure 8.1 (a) Cover image 512 × 512 and (b) Cover image histogram

8.1.3.1 Embedding Capacity and Distortion Evaluations

Some steganography methods, such as LSB, provide fixed embedding rate. The embedding payload of the proposed method differs from one image to another, and hence, the embedding rate depends on the contents of the cover image and the threshold value used to discover edges.

Figures 8.1a and 8.1b show one of the cover image used in the experiment and its histogram. Figures 8.2a, 8.2c and 8.2e are the stego images resulting from the 1 bpp embedding algorithm (Section 5.2.2) in spatial domain for 5%, 20% and 40% embedding rates. The visual differences between the cover and stego images cannot be discovered by the human eye, and even the histograms of the stego images (illustrated in Figures 8.2b, 8.2d and 8.2f are quite similar to that of the cover image. Figure 8.3 shows the KLD of Figure 8.1a using the 1bpp proposed method with different embedding rates. The difference between the probability distribution of the histogram of the cover and stego images is small, which indicates a high security.

Figures 8.4a-8.4c show the cover and stego images obtained from 1bbp embedding algorithm for 10% and 30% embedding rates. Smoother and textural parts from both the cover and stego images are zoomed as shown in Figures 8.4d - 8.4f. It can be observed that is hard to notice the difference between the cover and stego images.

Figures 8.5a, 8.5c and 8.5e show the stego images obtained from applying the n bpp embedding algorithm (Section 5.2.3) for 5%, 20% and 40% embedding rates. The visual difference between the cover and stego images cannot be discovered by the human eye. The stego histogram of the n bpp embedding algorithm shown in Figures 8.5b, 8.5d and 8.5f indicate a very high degree of similarity with the histogram of the cover image. Figure 8.6 shows the KLD of the image shown in Figure 8.1a using the proposed Nbpp method with different embedding rates. The results reveal that the difference between the probability distribution of the histogram of the cover and stego images is small, which indicates a high security. The KLD of the proposed Nbpp method is smaller than that of the proposed 1bpp method because the Nbpp method utilizes less number of pixels to embed the same message. Figures 8.7a-8.7c represent the difference between the cover and stego images resulting from N bpp embedding algorithm with 5%, 20% and 40% embedding rates respectively. The white pixels denote the pixels that have been changed after the embedding process.

Figures 8.8a, 8.8c and 8.8e show the stego images obtained from applying the n bpp embedding algorithm in IWT domain (Section 5.2.4) for 5%, 20% and 40% embedding rates. The stego histogram of the n bpp embedding algorithm shown in Figures 8.8b,

Table 8.1 Image quality evaluation with various 1-bpp steganographics methods in the spatial domain and embedding rates over 10,000 stego images. The red values indicate the best result

Embedding Rate	Method		MSE	PSNR	wPSNR	Avg. difference
5%	LSB-based	TBPC [111]	$0.0269(\pm 4.6\times 10^{-4})$	$63.82(\pm 0.099)$	$64.84(\pm 1.43)$	$0.0269(\pm 4.6\times 10^{-4})$
	Edge-LSB	EALSB-MR [92]	$0.0322(\pm 3.8\times 10^{-3})$	$63.07(\pm 0.425)$	$69.99(\pm 2.88)$	$0.0301(\pm 1.8\times 10^{-3})$
		Proposed	$0.0207(\pm 3.3\times 10^{-4})$	$64.98(\pm 0.069)$	$71.01(\pm 2.15)$	$0.0207(\pm 3.2\times 10^{-4})$
10%	LSB-based	TBPC [111]	$0.0415(\pm 7.6\times 10^{-4})$	$61.94(\pm 0.086)$	$62.98(\pm 1.42)$	$0.0415(\pm 7.7\times 10^{-4})$
	Edge-LSB	EALSB-MR [92]	$0.0578(\pm 9.8\times 10^{-3})$	$60.55(\pm 0.574)$	$66.23(\pm 2.96)$	$0.0531(\pm 4.4\times 10^{-3})$
		Proposed	$0.0413(\pm 4.6\times 10^{-4})$	$61.98(\pm 0.049)$	$66.83(\pm 1.74)$	$0.0413(\pm 4.6\times 10^{-4})$
20%	LSB-based	TBPC [111]	$0.0809(\pm 1.1\times 10^{-3})$	$59.04(\pm 0.068)$	$60.32(\pm 1.36)$	$0.0809(\pm 1.2\times 10^{-3})$
	Edge-LSB	EALSB-MR [92]	$0.1088(\pm 2.5\times 10^{-2})$	$57.85(\pm 0.807)$	$62.14(\pm 2.91)$	$0.0969(\pm 1.1\times 10^{-2})$
		Proposed	$0.0826(\pm 6.6\times 10^{-4})$	$58.96(\pm 0.034)$	$62.92(\pm 1.52)$	$0.0826(\pm 6.6\times 10^{-4})$
25%	LSB-based	TBPC [111]	$0.1012(\pm 9.7\times 10^{-4})$	$58.99(\pm 0.077)$	$60.32(\pm 1.39)$	$0.1012(\pm 1.1\times 10^{-3})$
	Edge-LSB	EALSB-MR [92]	$0.1369(\pm 3.5\times 10^{-2})$	$56.87(\pm 0.907)$	$60.69(\pm 2.89)$	$0.1198(\pm 1.5\times 10^{-2})$
		Proposed	$0.1033(\pm 7.4\times 10^{-4})$	$57.99(\pm 0.031)$	$61.77(\pm 1.45)$	$0.1033(\pm 7.4\times 10^{-4})$
30%	LSB-based	TBPC [111]	$0.1230(\pm 1.1\times 10^{-3})$	$57.23(\pm 0.073)$	$58.77(\pm 1.37)$	$0.1230(\pm 1.2\times 10^{-3})$
	Edge-LSB	EALSB-MR [92]	$0.1790(\pm 4.9\times 10^{-2})$	$55.73(\pm 1.012)$	$59.02(\pm 2.87)$	$0.1529(\pm 2.2\times 10^{-2})$
		Proposed	$0.1239(\pm 8.1\times 10^{-4})$	$57.19(\pm 0.028)$	$60.98(\pm 1.36)$	$0.1239(\pm 8.1\times 10^{-4})$
40%	LSB-based	TBPC [111]	$0.1652(\pm 1.5\times 10^{-3})$	$55.95(\pm 0.045)$	$57.85(\pm 1.09)$	$0.1652(\pm 1.5\times 10^{-3})$
	Edge-LSB	EALSB-MR [92]	$0.2448(\pm 6.6\times 10^{-2})$	$54.38(\pm 1.079)$	$57.20(\pm 2.78)$	$0.2022(\pm 2.7\times 10^{-2})$
		Proposed	$0.1624(\pm 9.3\times 10^{-4})$	$56.12(\pm 0.024)$	$60.59(\pm 1.09)$	$0.1652(\pm 9.2\times 10^{-4})$

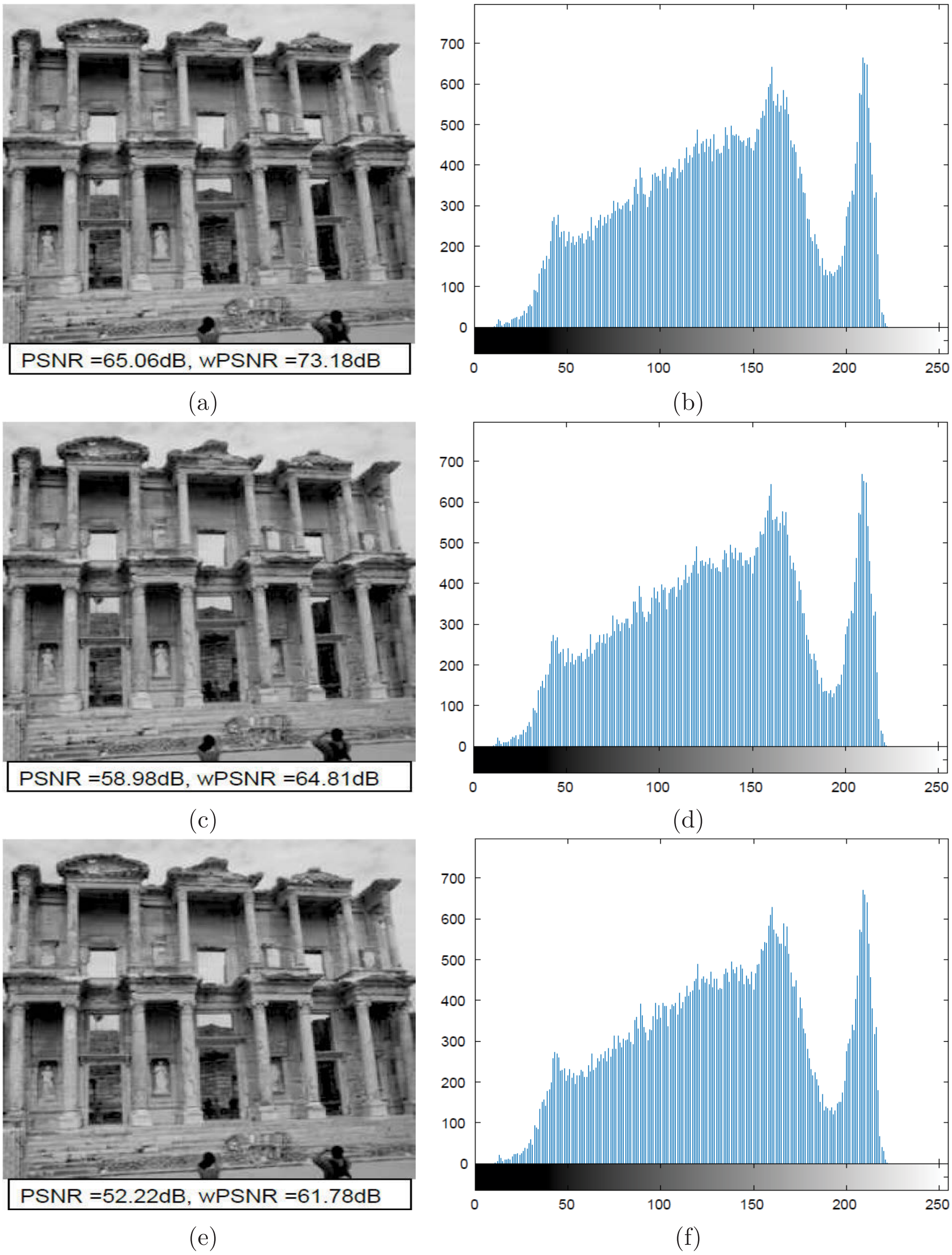


Figure 8.2 (a), (c) and (e) Stego images using the 1bpp proposed algorithm (Section 5.2.2) in the spatial domain with 5%, 20% and 30% embedding rate, and (b), (d) and (f) Histograms of the corresponding stego images

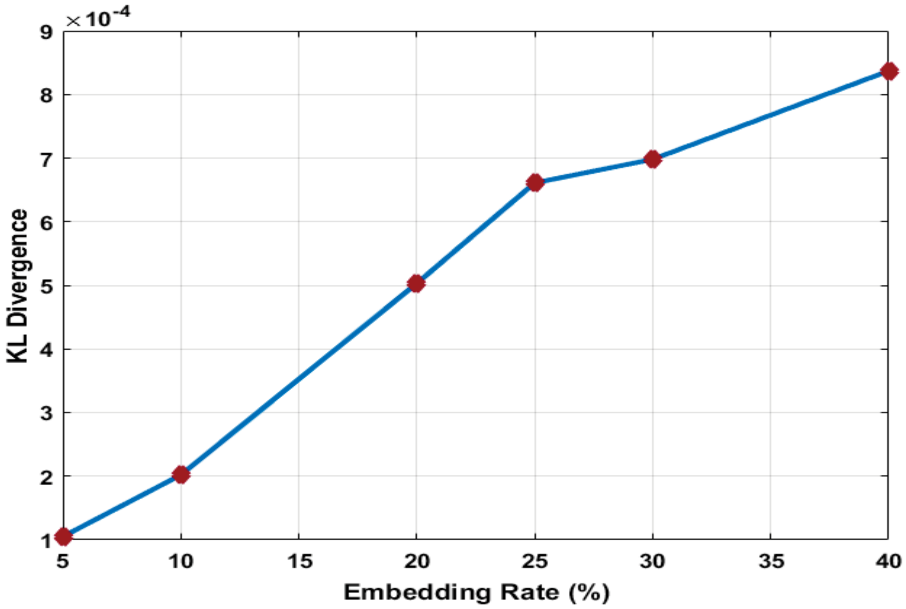


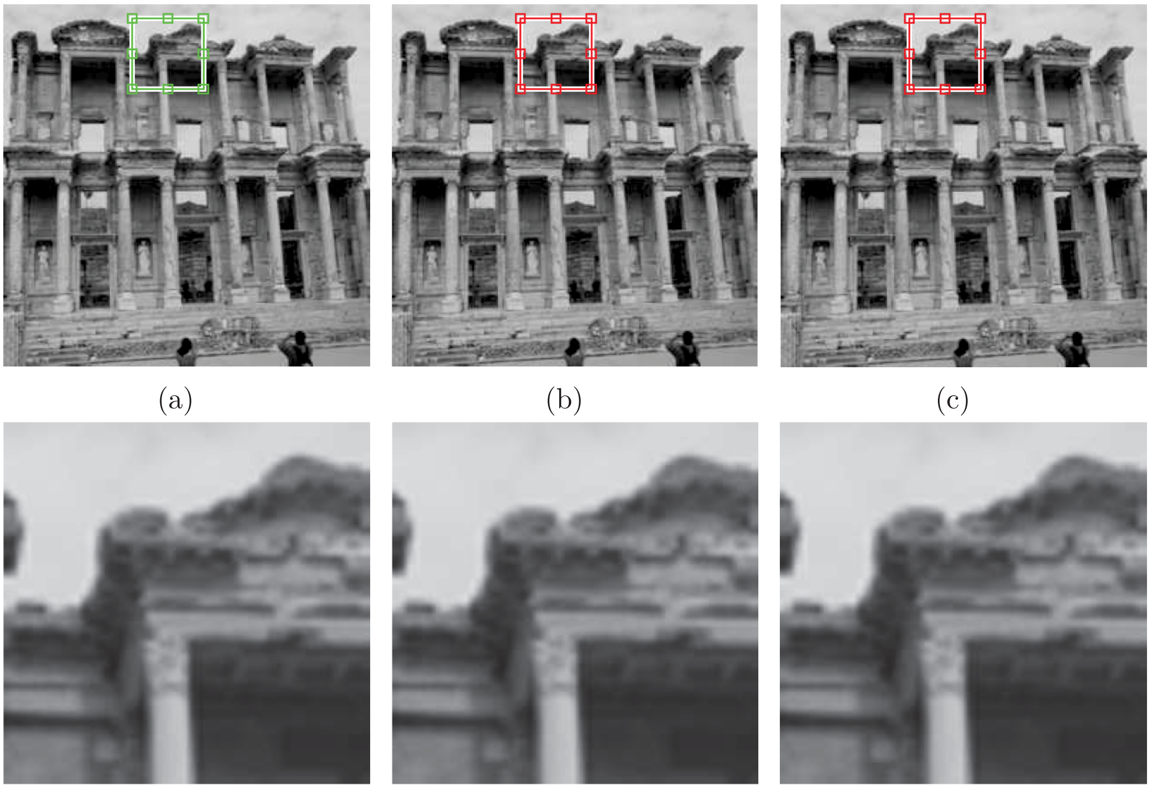
Figure 8.3 KLD for Figure 8.1 using 1bpp proposed method with various embedding rates

Table 8.2 The computational cost of various 1-bpp steganographic methods in the spatial domain to embed 12902 bits

	TBPC [111]	EALSB-MR [92]	Proposed
Computational cost (sec)	23.87	0.104	2.033

8.8d and 8.8f indicate a very high degree of similarity with the histogram of the cover image.

Two measures are commonly used to estimate the quality of the stego images with respect to cover images, which are Peak Signal-to-Noise ration (PSNR) and its weighted version wPSNR. Despite of being widely used, the PSNR quality metric does not take into consideration the HVS characteristics. It calculates the degradation in all regions in the same way. wPSNR has been introduced to give a more accurate image quality measurement. Table 8.1 presents the quality of the stego images using different 1-bpp steganography methods with embedding rates ranging from 5% to 40%. It is observed that, the proposed 1 bpp method (5.2.2) obtained the best image quality compared to EALSBMR [92] since it utilizes the XOR operation to reduce the difference between the cover and stego images. Also, EALSBMR performs readjustment operation in



(a) (b) (c)

(d) (e) (f)

Figure 8.4 (a) Cover image, (b-c) Stego Images using the 1bpp proposed algorithm in the spatial domain with 10% and 30% embedding rate, (d) zoomed area from the cover image, and (e-f) zoomed area from the stego image with 10% and 30% embedding rate

some cases to guarantee the extraction of the exact secret message in the receiver side. The PSNR values of the TBPC [111] and the 1 bpp proposed method are found to be close because both methods use coding to reduce the difference between the cover and stego images, where the TBPC uses a tree structure and the proposed method uses XOR operations. However, the wPSNR values of the proposed method are better than that of the TBPC. The reason is that the proposed method embeds the secret data on the edge regions. Also, the computational cost of the TBPC method is high because of the use of tree structure as shown in Table 8.2.

On the other hand, Table 8.3 presents the results of embedding n-bpp using TPVD, edge adaptive PVD, edge adaptive n-LSBs and our proposed method (5.2.3). In edge adaptive PVD, the secret data is embedded in the edge regions according to the difference value between each two adjacent pixels. In edge adaptive n-LSBs method,

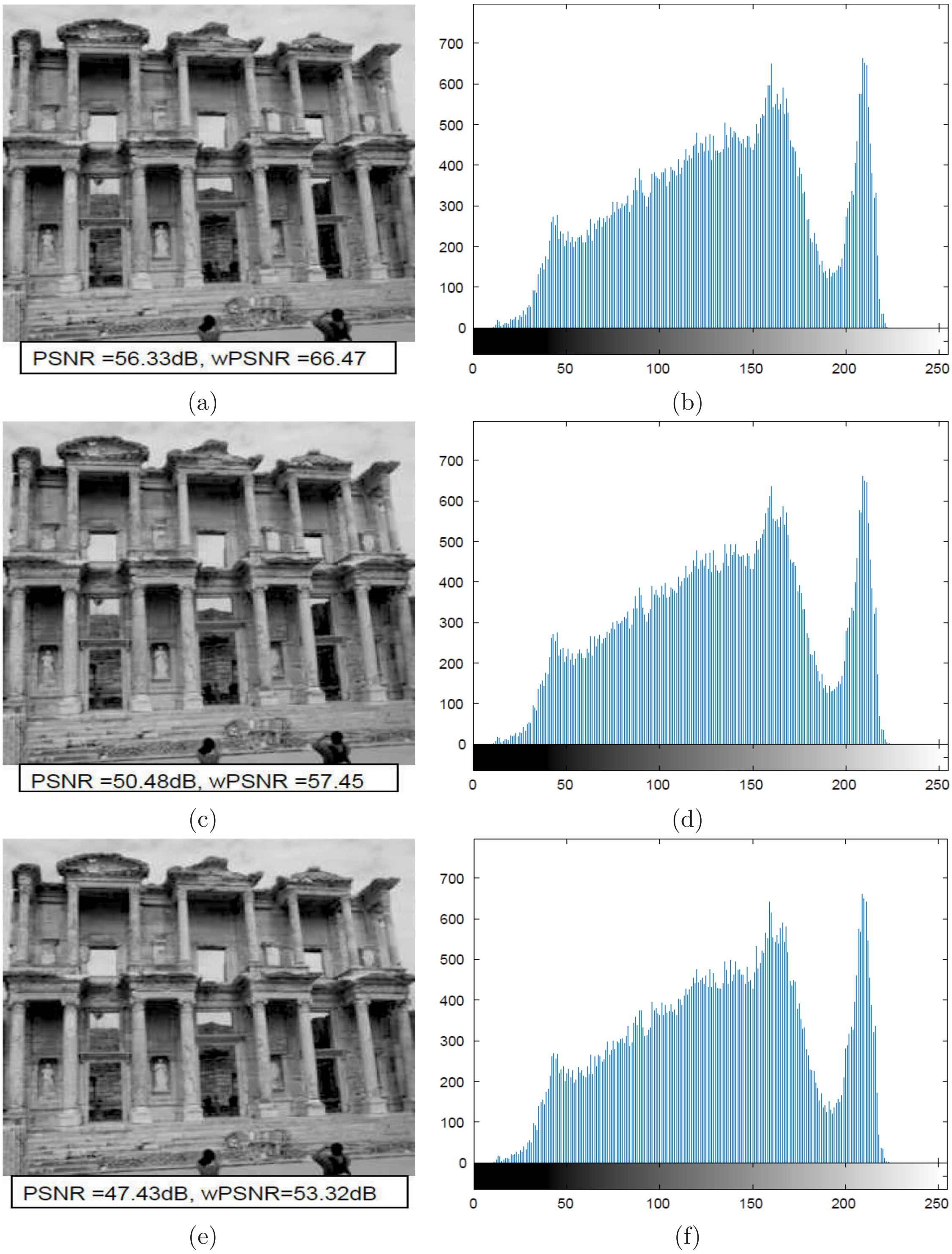


Figure 8.5 (a), (c) and (e) Stego images using the Nbpp proposed (5.2.3) algorithm in the spatial domain with 5%, 20% and 40% embedding rate and (b), (d) and (f) Histograms of the corresponding stego images

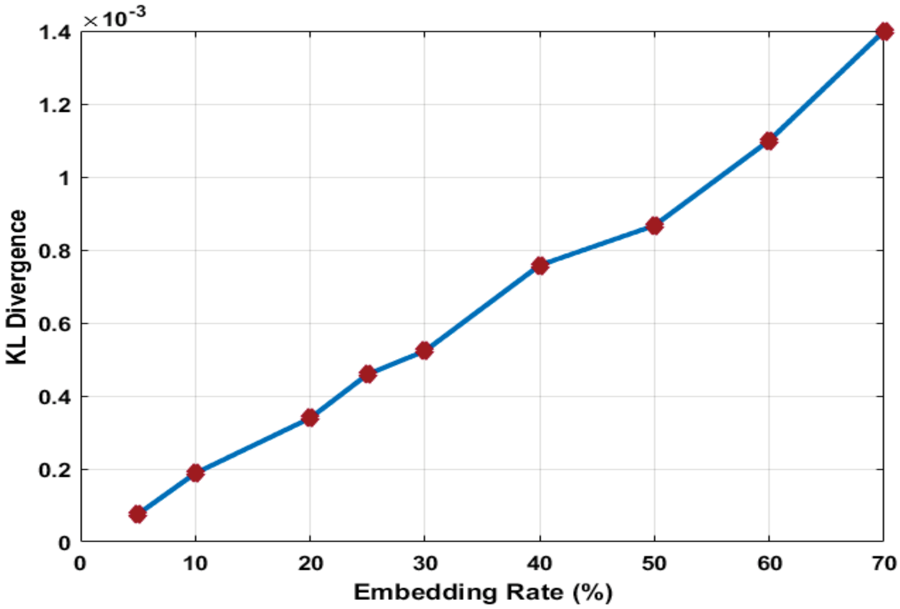


Figure 8.6 KLD for Figure 8.1 using Nbpp proposed method with various embedding rates

the LSB steganography method is expanded and the proposed edge detection method is used to discover the sharpest regions for the embedding process. It is similar to the embedding method excluding the XOR operations. The visual quality results are noticeably high for the proposed method compared to the other methods.

The embedding rates listed in Table 8.3 indicate that the embedding payload of the n bpp algorithm is about double that of the 1 bpp algorithm and it exceeds 70% of the cover image size. This is achieved with reduction in image quality, as indicated by the PSNR and wPSNR values (shown in Table 8.3).

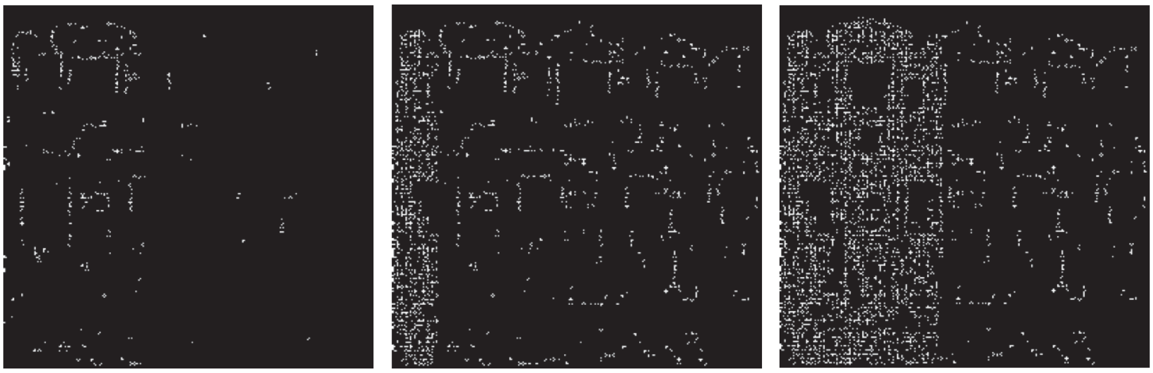
Table 8.4 presents the visual quality performance results of the proposed method in the Integer Wavelet Domain (5.2.4). The results show that the proposed method obtain a good PSNR and wPSNR values for different embedding capacity. The PSNR values are between 61.37 dB and 48.45 dB with 5% - 50% embedding rates, where the minimum acceptable value of the PSNR is 35 dB. Moreover, the maximum average difference between the cover and stego images is up to 0.461.

In order to have a comparison between the two proposed N-bpp methods (spatial and IWT domains), a graphical representation of the PSNR and wPSNR values are shown

Table 8.3 Image quality evaluation with various N-bpp steganographics methods in the spatial domain and embedding rates over 10,000 stego images. The red values indicate the best result

Embedding Rate	Method		MSE	PSNR	wPSNR	Avg. difference
10%	Edge-based	Adaptive PVD [231]	0.675	49.84	58.50	0.1401
		TPVD [232]	0.998	48.14	52.11	0.1598
		Adaptive N-LSB	0.307	53.26	60.69	0.0842
	Edge-XOR-based	Proposed	**0.289**	**53.53**	**60.43**	**0.0849**
25%	Edge-based	Adaptive PVD [231]	0.957	48.32	54.74	0.4669
		TPVD [232]	2.845	43.59	47.64	0.5154
		Adaptive N-LSB	0.853	48.82	54.73	0.2216
	Edge-XOR-based	Proposed	**0.694**	**49.72**	**55.13**	**0.2085**
40%	Edge-based	Adaptive PVD [231]	1.167	47.46	52.60	0.3784
		TPVD [232]	5.742	40.54	45.14	0.6853
		Adaptive N-LSB	1.334	46.88	52.37	0.3446
	Edge-XOR-based	Proposed	**1.072**	**47.83**	**52.86**	**0.3282**
50%	Edge-based	Adaptive PVD [231]	**1.292**	**47.02**	51.59	0.4492
		TPVD [232]	6.654	39.90	43.85	0.8914
		Adaptive N-LSB	1.675	45.89	51.35	0.4292
	Edge-XOR-based	Proposed	1.316	46.94	**51.98**	**0.4049**
60%	Edge-based	Adaptive PVD [231]	1.706	45.81	50.58	0.5557
		TPVD [232]	11.833	37.40	42.78	1.1052
		Adaptive N-LSB	2.014	45.09	50.43	0.5173
	Edge-XOR-based	Proposed	**1.571**	**46.17**	**51.14**	**0.4842**
70%	Edge-based	Adaptive PVD [231]	**1.816**	**45.54**	50.04	0.6178
		TPVD [232]	17.381	35.73	41.70	1.3555
		Adaptive N-LSB	2.350	44.42	49.69	0.6027
	Edge-XOR-based	Proposed	1.833	45.50	**50.53**	**0.5651**

in Figures 8.9a and 8.9b. It is clear that the stego images of the proposed IWT method achieves a higher visual quality compared to those obtained using the spatial domain method. However, the spatial domain method provides a higher embedding capacity compared to the IWT method, i.e., 70% compared to a maximum of 50% based on Eq. 8.1 because the LL sub-band is excluded from the embedding process.



(a) (b) (c)

Figure 8.7 (a-c) Difference between the cover and stego images using the Nbpp proposed algorithm in the spatial domain with 5%, 20% and 40% embedding rate

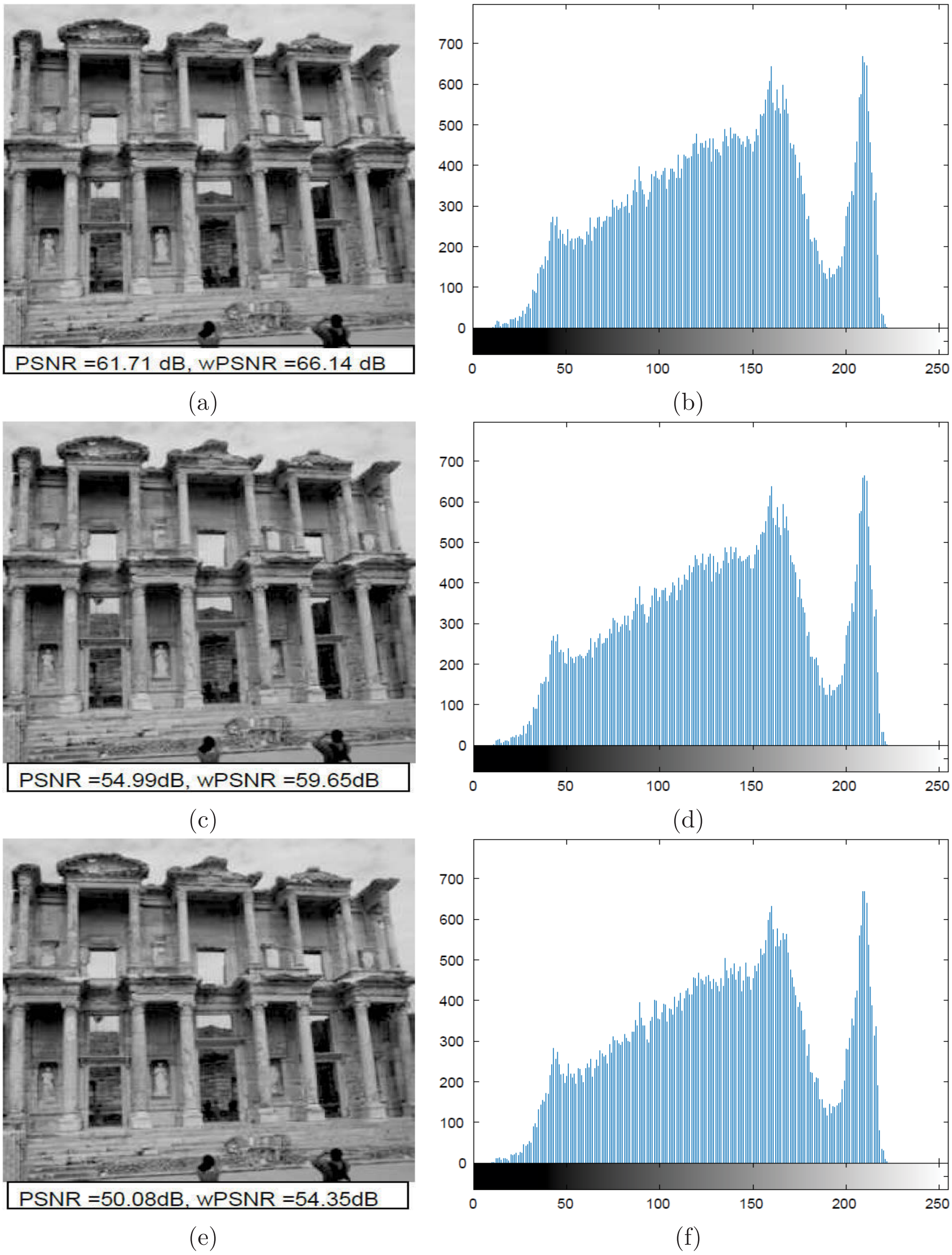
Table 8.4 Image quality evaluation of the N-bpp IWT proposed method with embedding rates over 10,000 stego images

Embedding Rate	MSE	PSNR	wPSNR	Avg. Difference
5%	0.047	61.37	65.49	0.041
10%	0.103	58.14	62.04	0.081
25%	0.369	52.46	56.44	0.225
30%	0.487	51.30	55.37	0.274
40%	0.710	49.62	53.86	0.372
50%	0.929	48.45	52.95	0.461

The effect of utilizing one or n bits per pixel of the proposed method is illustrated in Tables 8.1, 8.3 and 8.4. If only the first bit is only used in the embedding process, then the error will be in the range $[-1, +1]$ based on the value of the secret bit. On the other hand, the N-bpp proposed method utilizes three bits from the sharp region, which increases the error of the pixel value to be in the range $[-7, +7]$. The degradation of stego image quality increases when using more bits.

8.1.3.2 Security Evaluation

Analysis of the proposed method is performed by extracting feature sets from the original and stego images. These features are used to train an SVM (support vector

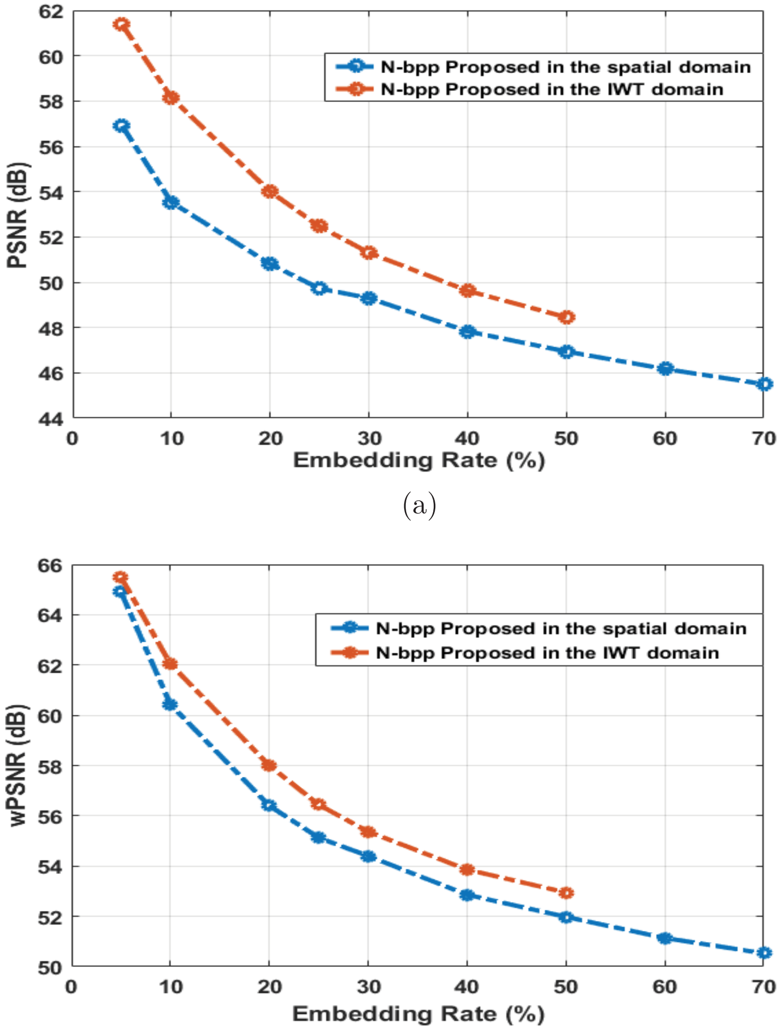


(a) (b)

(c) (d)

(e) (f)

Figure 8.8 (a), (c) and (e) Stego images using the Nbpp proposed (5.2.4) algorithm in the integer wavelet domain with 5%, 20% and 40% embedding rate and (b), (d) and (f) Histograms of the corresponding stego images



(a)

(b)

Figure 8.9 (a) PSNR values and (b) wPSNR values of the proposed N-bpp in the spatial and wavelet domains

machine) classifier to learn the difference in features produced by data embedding. In the experiments, the linear function kernel with the default parameter setting is selected and the Matlab implementation of SVM is used. Also, Testing is done in two-fold cross-validation, i.e., half of the cover and stego images are randomly selected for training, and the remaining images are used for testing. This test is repeated twenty times, and the average of the obtained accuracy values (shown in Tables 8.5 and 8.6) indicate that TBPC, EALSB-MR, TPVD, edge adaptive PVD and edge adaptive n LSBs are detected with an accuracy greater than the proposed method for most of the embedding rates (since the number of stego and cover images are equal, the random guess accuracy is 50%). Moreover, when the embedding rate increased, the accuracy value of the SVM classification increased. For example, when the embedding rate is 40% using 1 bpp method, the accuracy value is 60.23%. Even though the obtained classification accuracy values of the proposed method are higher than that of the random guess, these values are not high enough to enable the differentiation between the cover and stego images with an acceptable precision. This however is not the case for other steganography methods, especially with the high embedding rates. Please note that the IWT steganography algorithm has not been tested using this steganalysis method, as the Li-110D has been mainly developed to detect spatial steganography.

Table 8.5 The average accuracy value (for 10,000 cover images and their corresponding stego images) against Li-110D with various 1-bpp methods. The red values indicate the best result

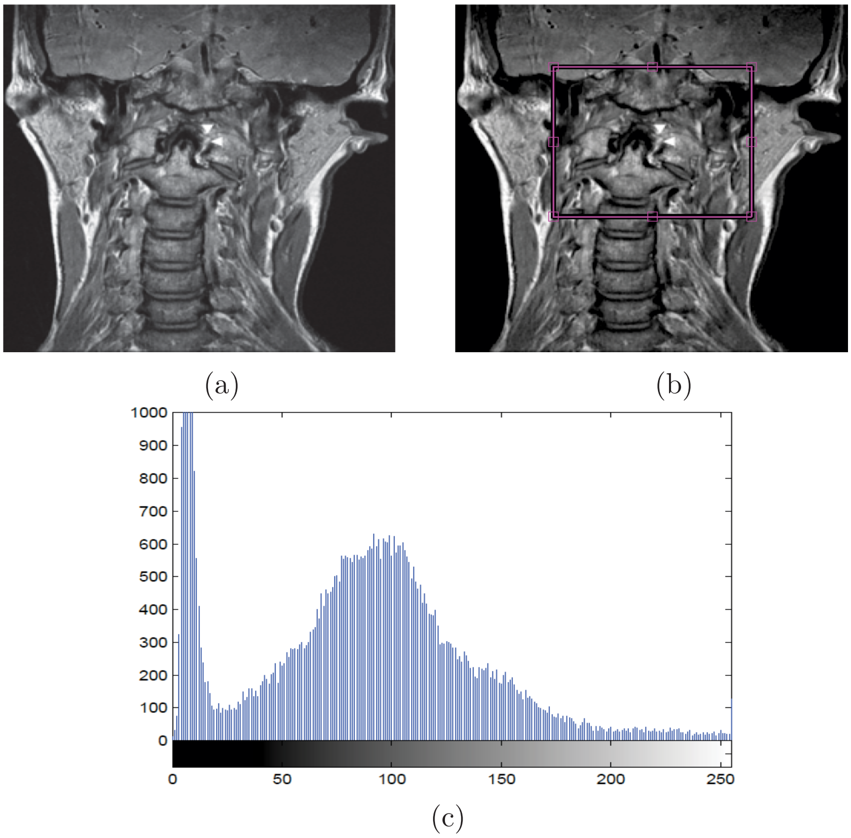
Embedding Rate	5%	10%	20%	25%	30%	40%	Avg
TBPC [111]	65.51	74.87	81.10	84.06	85.26	89.09	79.98
EALSB-MR [92]	49.74	52.14	57.02	59.40	63.21	69.87	58.56
1bpp Proposed	49.58	51.27	54.37	55.69	57.45	60.23	54.78

Table 8.6 The average accuracy value (for 10,000 cover images and their corresponding stego images) against Li-110D with various N-bpp methods. The red values indicate the best result

Embedding Rate	10%	25%	40%	50%	60%	70%	Avg
Adaptive-PVD [231]	51.89	61.56	74.28	80.44	84.48	86.69	73.22
TPVD [232]	68.35	84.90	90.50	92.24	93.55	94.58	87.35
Adaptive-N LSB	54.73	59.02	62.35	67.66	69.50	76.37	64.93
N-bpp Proposed	52.80	58.83	62.74	63.27	64.72	66.01	61.39

8.1.4 Medical Image Steganography Methodology Results

This section presents the experimental results of the proposed medical image steganography method, which is introduced in Section 6.4. The proposed methodology is evaluated using 100 MRI cover images (all of them are grey-level of size 255 × 255). Figures 8.10a and 8.10b represent one of the cover images used in the experiment and its ROI. Figure 8.10c shows the corresponding histogram of the cover image.


(a)

(b)

(c)

Figure 8.10 (a) MRI cover images, (b) ROI of the cover image, and (c) histogram of the cover image

8.1.4.1 Embedding Capacity and Distortion Evaluation

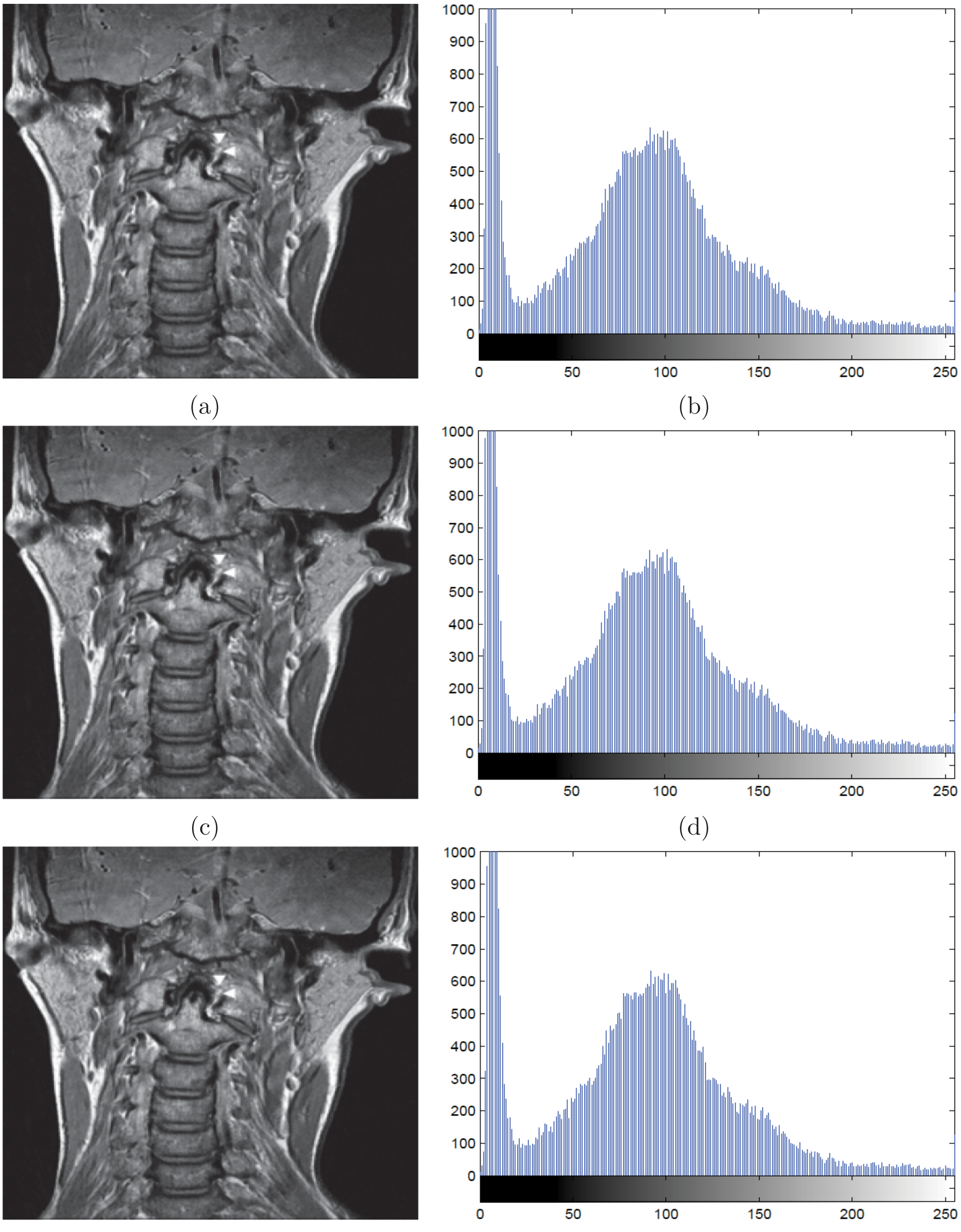
The quality of the embedding process is often evaluated by the embedding efficiency (e). It represents the number of embedded bits per embedding change (introduced change), where a maximized embedding efficiency value is related to a minimized embedding distortion. It can be determined by $e = K/d$, where K is the length of the secret message and d is the number of bit changes [109, 110].

The stego images produced with embedding rates of 5%, 20% and 40% using the proposed method using STC are shown in Figures 8.11a, 8.11c and 8.11e. One can notice that it is difficult for the human eye to differentiate between the original and stego images. Moreover, the histogram of the stego images (represented in Figures 8.11b, 8.11d and 8.11f) are quite similar to that of the cover images. It is important to mention that for a 255×255 image, which contains 65025 pixels, an embedding rate of 40% means embedding $(40/100) * 65025 = 26010$ bits, which is equivalent to 3,715 ASCII characters. For a 512×512 image, an embedding rate of 40% means embedding 104757 bits or 14,979 ASCII characters.

In order to have a comprehensive comparison with the proposed method, PVD, TPVD, edge adaptive PVD and edge adaptive n-LSBs methods are implemented. In edge adaptive PVD, the secret data is embedded in the sharp regions first based on the difference value between each two consecutive pixels (without overlap). In edge adaptive n-LSBs method, the LSB steganography method is expanded and the proposed edge detection method is used to discover the sharpest regions for the embedding process. It is similar to the proposed embedding method excluding the coding algorithm and cost function. The visual quality performance results are shown in Table 8.7. It is clear that the proposed method using STC obtained the best image quality in all image metric measurements compared to the other methods, followed by the Hamming code implementation of the proposed algorithm.

Similar to other measures, the average difference increases with the increase of the embedding rate, but for the proposed method using STC an average difference of 0.21 is obtained using an embedding rate of 50%. The average difference for the Hamming code is found to be the second best, with a maximum of 0.323 for 50% embedding rate. These results indicate that the difference between the cover and stego images obtained using the proposed method is generally small. The proposed method and the edge adaptive n-LSB results confirm the effectiveness role of using the coding theory, which aims to reduce the difference between the cover and stego images.

A graphical representation of the PSNR and wPSNR values are shown in Figures 8.12a and 8.12b respectively. Those two figures demonstrate the superior performance of the proposed method using STC, and to a slightly less degree using Hamming code, in comparison to the other methods with PSNR values greater than 50 dB and wPSNR



(a) (b)

(c) (d)

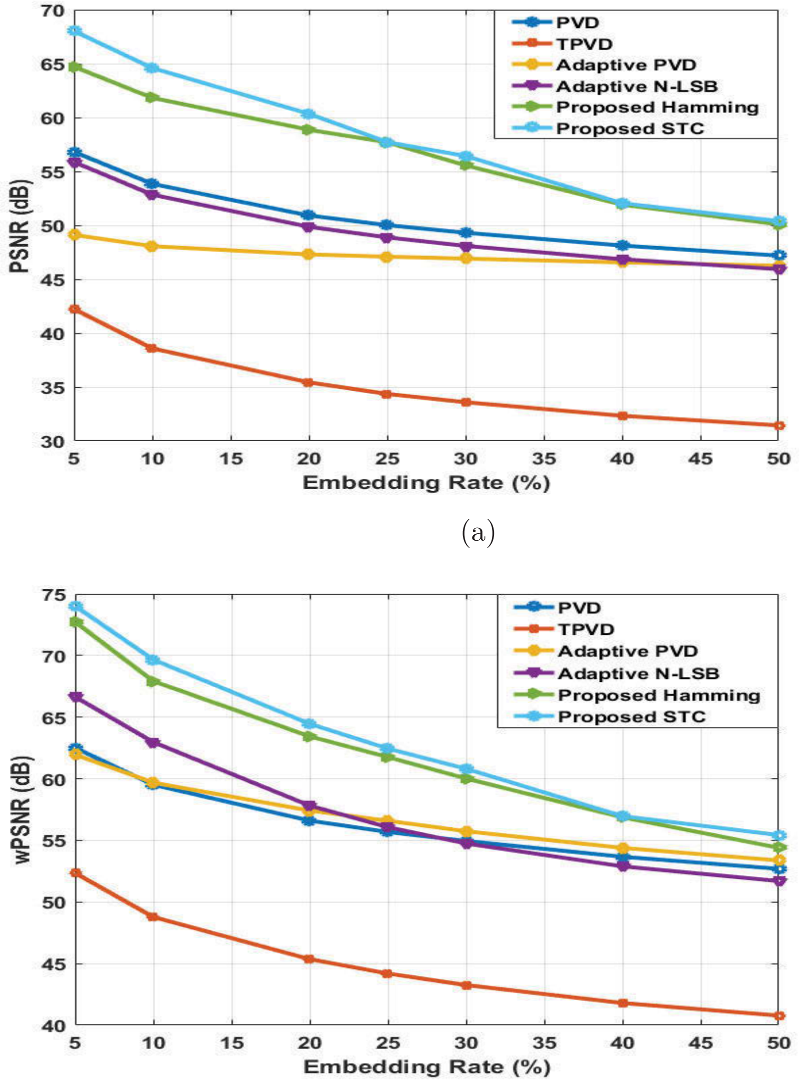
(e) (f)

Figure 8.11 Stego images produced by STC (a) 5%, (c) 20% and (e) 40% embedding rate, (b), (d) and (f) Histogram of the corresponding stego images

Table 8.7 Comparison of the results of PVD, TPVD, Adaptive PVD, Adaptive N-LSB and the N-bpp proposed methods using XOR, Hamming and STC. The red values indicate the best result

Method	Embedding rate	5%	10%	25%	30%	40%	50%
PVD [81]	Avg. Diff	0.044	0.088	0.219	0.259	0.344	0.428
	Avg. MSE	0.136	0.269	0.635	0.762	1.025	1.242
	Avg. PSNR	56.81	53.84	50.01	49.31	48.12	47.19
	Avg. wPSNR	62.51	9.51	55.68	54.95	53.67	52.69
TPVD [232]	Avg. Diff	0.146	0.292	0.754	0.912	1.228	1.515
	Avg. MSE	3.891	9.017	23.83	28.45	38.03	46.67
	Avg. PSNR	42.23	38.58	34.36	33.59	32.33	31.44
	Avg. wPSNR	52.35	48.78	44.19	43.26	41.80	40.78
Adaptive PVD [231]	Avg. Diff	0.104	0.164	0.284	0.318	0.386	0.453
	Avg. MSE	0.798	1.016	1.271	1.325	1.436	1.542
	Avg. PSNR	49.11	48.06	47.09	46.91	46.56	46.25
	Avg. wPSNR	61.96	59.68	56.59	55.74	54.39	53.37
Adaptive N-LSB	Avg. Diff	0.043	0.086	0.215	0.258	0.343	0.425
	Avg. MSE	0.168	0.338	0.842	1.010	1.343	1.661
	Avg. PSNR	55.87	52.84	48.88	48.09	46.85	45.93
	Avg. wPSNR	66.65	62.97	56.07	54.75	52.89	51.70
Proposed Method Edge-XOR	Avg. Diff	0.043	0.086	0.216	0.258	0.344	0.421
	Avg. MSE	0.147	0.295	0.739	0.886	1.175	1.421
	Avg. PSNR	56.45	53.43	50.44	48.76	47.43	46.61
	Avg. wPSNR	67.65	63.30	56.01	54.85	53.26	52.18
Proposed Method Hamming code	Avg. Diff	0.021	0.042	0.106	0.145	0.237	0.323
	Avg. MSE	0.022	0.043	0.111	0.181	0.419	0.637
	Avg. PSNR	64.71	61.82	57.70	55.56	51.91	50.09
	Avg. wPSNR	72.73	67.92	61.76	60.02	56.87	54.40
Proposed Method STC	Avg. Diff	0.009	0.022	0.071	0.091	0.164	0.210
	Avg. MSE	0.010	0.023	0.111	0.150	0.422	0.594
	Avg. PSNR	68.02	64.59	57.71	56.43	52.05	50.39
	Avg. wPSNR	74.02	69.66	62.45	60.81	56.97	55.43

values greater than 55 dB, which means the hidden data is undetectable according to the human visual perception.



(a)

(b)

Figure 8.12 (a) PSNR values and (b) wPSNR values for PVD, TPVD, edge adaptive PVD, edge adaptive n-LSB and the proposed method using Hamming and trellis codes

Table 8.8 presents the quality of the stego images using different 1 bpp steganography methods and the proposed method using the XOR and STC with embedding rates ranging from 5% to 30%. It is observed that, the proposed 1 bpp method using STC obtained the best image quality compared to the proposed method using XOR operation, EALSBMR [92] and Bassil [94] since it utilizes the trellis code to reduce the difference between the cover and stego images.

Table 8.8 Comparison of the results of EALSB-MR, LSB Canny, the 1-bpp proposed method using XOR and the 1-bpp proposed method using STC. The red values indicate the best result

Method	Embedding rate	5%	10%	20%	25%	30%
EALSB-MR [92]	Avg. Diff	0.030	0.053	0.097	0.120	0.153
	Avg. MSE	0.032	0.058	0.109	0.137	0.179
	Avg. PSNR	63.07	60.55	57.85	56.87	55.73
	Avg. wPSNR	69.99	66.23	62.14	60.69	59.02
LSB Canny [94]	Avg. Diff	0.030	0.053	0.101	0.126	0.148
	Avg. MSE	0.030	0.053	0.101	0.126	0.148
	Avg. PSNR	63.28	60.85	58.10	57.13	56.42
	Avg. wPSNR	69.57	65.23	60.70	60.42	57.73
Proposed Method XOR (1 bpp)	Avg. Diff	0.021	0.042	0.083	0.104	0.125
	Avg. MSE	0.019	0.041	0.079	0.103	0.124
	Avg. PSNR	64.94	61.93	58.93	57.96	57.17
	Avg. wPSNR	72.54	68.11	63.79	62.06	60.81
Proposed Method STC (1 bpp)	Avg. Diff	0.009	0.019	0.045	0.056	0.062
	Avg. MSE	0.010	0.019	0.046	0.057	0.061
	Avg. PSNR	68.28	66.48	62.55	61.02	60.36
	Avg. wPSNR	72.99	70.48	65.22	64.75	62.17

Table 8.9 Comparison of embedding efficiency for STC and Hamming code with different payloads

Method	Payload	Embedding Efficiency
STC	20%	4.47
	50%	4.44
Hamming code	20%	2.39
	50%	2.40

Table 8.9 shows the embedding efficiency of the proposed algorithm for the two coding techniques of STC and Hamming Code. The results indicate that STC achieves a better embedding efficiency (causes less modifications to the cover image) compared to the Hamming code. On the other hand, it is worth mentioning that due to the

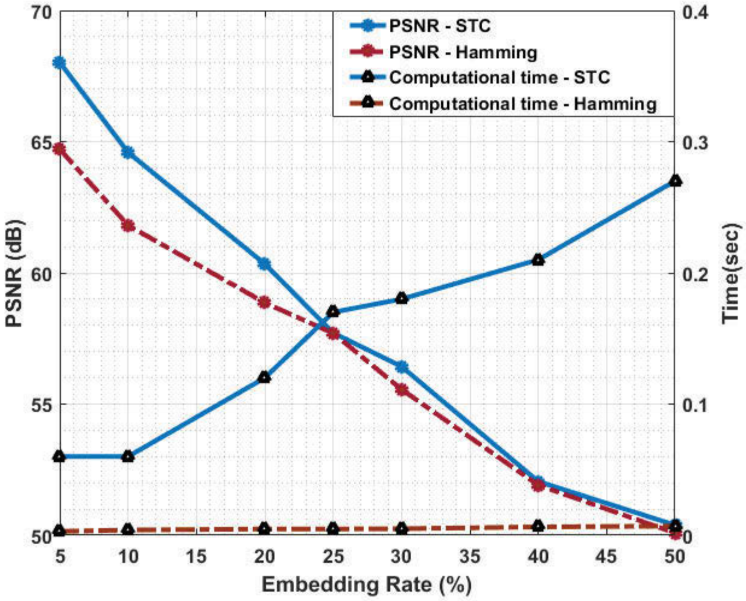


Figure 8.13 The computational cost and PSNR values for the proposed method using Syndrome-Trellis and Hamming codes

shift operations, the STC has noticeably higher computational time compared to the Hamming code. Figure 8.13 illustrates the computational time and PSNR values of the STC and hamming code for various embedding rates. The implementation of the proposed methodology was running on a PC with Intel I5-3470 Quad core 3.20GHz and 8GB RAM. According to [109, 110], the constraint height is usually ranged between 6 to 15, and therefore, a value of 12 is chosen in implementing the STC. The figure shows that the difference in PSNR between the two methods gets smaller for higher embedding rates (25% or more), whereas the difference in computational time gets bigger, where it exponentially increases with the increase of the embedding rate for the STC method. Therefore, based on the length of the message to be embedded, i.e., embedding rate, and the need for computational time saving, the choice between the STC and Hamming code implementations can be left to the user.

The effect of varying the size of ROI has also been investigated for different embedding rates, as shown in Figure 8.14. As explained earlier, the proposed method demonstrated a high embedding capacity with a very good visual quality in terms of PSNR and wPSNR. However, as the ROI size gets bigger, the RONI size gets smaller, and hence the number of bits that can be embedded gets smaller. Even if the message can be fully embedded when using a bigger ROI, the algorithm will need to embed in less sharp regions, which will have an impact on PSNR, wPSNR and MSE. For an acceptable visual quality of the MRI dataset, the maximum possible message length which may be embedded around a particular ROI was determined and analyzed, for cover images of size 255 × 255 as shown in Table 8.10. From this evaluation, we observed the following: an ROI of size 105 × 105 that is covering 17% of the original image size was capable of carrying a message length of up to 32,512 bits, whereas for a bigger ROI (165 × 165) covering 41%, the message length reduced to only 19,507 bits.

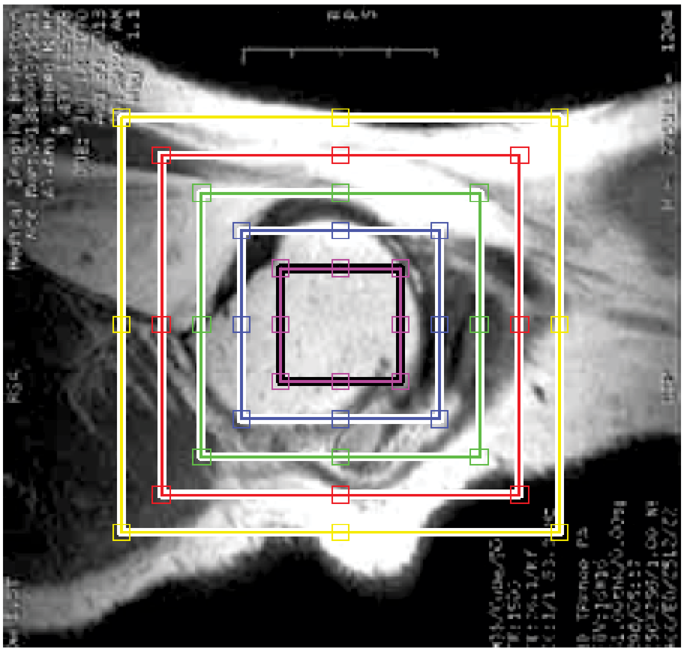


Figure 8.14 Cover Image with different ROI size

8.1.4.2 Security Evaluation

The security of the proposed method has been tested on the medical database, which contains 100 gray-scale images of size 255 × 255. Also, it has been tested on the BOWS2 database [223], which contains 10,000 grey-scale natural images of size 512 × 512 without taking into consideration ROI. It then trains a classifier to differentiate between stego and cover images. In the experiments, stego images are created using the proposed method with different embedding rates ranging from 5% to 50%. Then image features

Table 8.10 The image quality results of the proposed method using STC with different ROI and EPR sizes

Size of ROI	Embedding Rate	5%	10%	20%	25%	30%	40%	50%
45x45	MSE	0.010	0.023	0.052	0.096	0.137	0.334	0.557
	PSNR	68.11	64.60	61.04	58.36	56.78	53.77	50.70
	wPSNR	74.05	69.92	64.75	63.00	61.31	58.43	55.70
	Avg. Diff	0.009	0.022	0.047	0.067	0.087	0.146	0.202
75x75	MSE	0.010	0.023	0.060	0.108	0.144	0.403	0.589
	PSNR	68.05	64.59	60.52	57.83	56.60	52.29	50.44
	wPSNR	74.01	69.73	64.49	62.56	60.96	57.11	55.42
	Avg. Diff	0.009	0.022	0.049	0.070	0.089	0.161	0.208
105x105	MSE	0.010	0.023	0.074	0.116	0.175	0.474	0.611
	PSNR	67.98	64.60	59.58	57.51	55.86	51.42	50.28
	wPSNR	74.01	69.44	64.16	62.14	60.34	56.49	55.37
	Avg. Diff	0.010	0.022	0.053	0.073	0.096	0.172	0.213
135x135	MSE	0.011	0.023	0.096	0.184	0.292	0.532	-
	PSNR	67.66	64.49	58.31	55.83	53.69	50.87	-
	wPSNR	73.82	68.73	63.02	60.52	58.71	55.98	-
	Avg. Diff	0.010	0.022	0.061	0.088	0.117	0.184	-
165x165	MSE	0.012	0.026	0.145	0.247	0.364	-	-
	PSNR	67.48	64.08	56.90	54.45	52.54	-	-
	wPSNR	73.64	68.33	61.75	59.47	57.82	-	-
	Avg. Diff	0.011	0.023	0.070	0.099	0.129	-	-

are extracted from the cover and stego images. An SVM (Support Vector Machine) classifier is utilized to train these features to learn the difference in features produced by data embedding. Testing is done in two fold cross-validation i.e., half of the cover and stego images are selected randomly for training, and the remaining images are used for testing. This test is repeated twenty times, and the averages of the obtained accuracy values are shown in Tables 8.11 and 8.12. It can be noticed that the proposed method using STC outperforms the other methods for most of the embedding rates. When the embedding rate increased the difference between the cover and stego images increased, and consequently the error rate of the SVM classification decreased. Since there are only two classes and they have the same number of images in our problem, an error rate of 50% means random guess. Table 8.12 shows that the proposed method had error rates between 49% and 59% for embedding rates that are 40% and less. This range of error rate indicates that it is hard to make conclusive judgement about the existence of secret data in images. Even when the embedding rate is as high as 50%, the error rate was still quite high (35%) by using STC and (33%) by using Hamming code, which proves the efficiency of the proposed information hiding method.

Table 8.11 The average accuracy value of the proposed method (for 100 medical cover images and their corresponding stego images) against Li-110D steganalysis method

Embedding Rate	5%	10%	20%	25%	30%	40%	50%	Avg
Proposed Method Hamming	44.73	46.02	49.63	51.9	57.3	59.48	68.05	53.87
Proposed Method STC	45.38	44.78	47.8	50.01	53.63	56.73	58.82	51.02

Table 8.12 The average accuracy value of PVD, TPVD, Adaptive PVD, Adaptive N-LSB and the proposed method (for 10,000 cover images and their corresponding stego images) against Li-110D steganalysis method

Embedding Rate	5%	10%	20%	25%	30%	40%	50%	Avg
PVD	58.35	66.20	79.34	80.92	81.33	84.01	85.84	76.57
TPVD	59.26	68.35	80.94	84.90	87.38	90.50	92.24	80.51
Adaptive-PVD	49.86	51.89	57.81	61.56	65.86	74.28	80.44	63.10
Adaptive-LSB	52.34	54.73	57.44	59.02	60.10	62.35	67.66	59.09
Proposed Method Hamming	49.62	51.63	56.04	58.09	59.45	62.41	66.81	57.72
Proposed Method STC	49.02	50.83	53.87	54.66	56.05	59.24	65.40	55.58

8.1.4.3 Evaluation of the Proposed Method with Other Medical Information Hiding

Table 8.13 shows a comparison with an existing medical information hiding methods (Navas [131], Bremnavas [156], Thiyagarajan [233]), in term of embedding process, embedding capacity and encryption process. The extraction process of all methods is classified as blind, i.e., does not need the cover image to extract the secret data. The efficiency of the encryption methods depends on three factors: key generation, substitution and transposition operations. In [131] and [156], mono-alphabet substitution operation was utilized to encrypt the EPR, which replaces each plaintext letter with one ciphertext letter. While the proposed encryption method performs poly-alphabet substitution operation by representing each plaintext letter with multiple cipher letters. In terms of information hiding technique evaluation, the proposed method provides a high embedding rate compared to the other methods, and hence can be used to efficiently embed large amount of secret data.

Similar to other spatial steganography and many of the transform ones, the proposed algorithm is fragile to changes in the stego image (change in size, rotation, etc). However, even in such cases, the hidden information will not be revealed to the intruder, while the original image and EPR data can always be retrieved from the source.

Table 8.13 A comparison between the proposed method various information hiding techniques

	Algorithms			
	[131]	**[156]**	**[233]**	Proposed
Embedding Domain	IWT	Spatial	Spatial	Spatial
Embedding Technique	Difference between coefficients in LH and HL sub-bands	LSB	LSB	LSBM with STC or Hamming code
Encryption	Yes	Yes	NO	Yes
ROI	Yes	No	Yes, RONI is the background region using Canny method	Yes
Shape of ROI	Rectangular	No	Irregular	Rectangular Ellipse
Embedding Rate	8500 bits 13%	Patient record around 20 bytes	(650−1850) bits	(3251−32600) bits (5−50)%
Key	Yes	No	Yes	Yes
Extraction Process	Blind	Blind	Blind	Blind
Shared Info	NA	NA	NA	Coordinates of ROI

8.1.4.4 Encryption Process Evaluation

The transposition process transforms a plaintext by arranging the positions of the secret data digits of the original text without changing the identities of the digits. Let L represents the length of plaintext/ciphertext and K represents all possible permutations of $\{1, 2, \ldots, L\}$. The random key matrix which is used in Figures 6.4a and 6.4b, is a 5×5 matrix. There can be $25! = 1.55\,e + 25$ possible permutations for each stage. If the key is not known to the attacker, the brute force attack has to do a comprehensive search among these possibilities and it requires a large computational cost to find the correct order. Moreover, the hacker has to start in the reverse order to retrieve the original plaintext and the encryption process replaces the digits, where the ciphertext will not have exactly the same letter frequency distribution (the ciphertext is not vulnerable to the frequency distribution analysis methods).

The poly-alphabetic substitution cipher replaces each plaintext letter by a number of ciphertext letters. In the proposed encryption process, there are three different keys (x, y and z), and therefore the number of possible keys is $[(256 \times 51) - 1) \times 2^S]$, where S is the length of the secret data in the binary system. For example, if the secret data length is 500 byte, then the key space is $1.72\,e + 1208$.

In addition, the XOR operation is considered unconditionally secure since the plaintext and key lengths are equal and are only used once and never repeated with different embedding process. It is worth mentioning that state-of-the-art encryption techniques could be more robust against attacks. However, they are usually more computationally expensive, which would increase the overall computational cost of the steganography algorithm, especially that it contains a number of other building blocks.

Table 8.14 illustrates the computational time for two different encryption algorithms and the proposed encryption algorithm. The table shows that the AES and DES methods have higher computational times compared to the proposed encryption method.

Table 8.14 The computational cost for the AES-128, DES and the proposed encryption algorithm

	AES	DES	Proposed encryption method
Plaintext length (bit)	128	64	200
Computational cost (sec)	1.4136	0.6418	0.0374

8.2 Segmentation Performance Evaluation

This section presents the segmentation results obtained by the proposed automatic brain tissue segmentation methodology based on clustering fusion. The proposed methodology is implemented in MatlabR2016b.

8.2.1 MRI Image Datasets

In this study, image datasets are used from the Internet Brain Segmentation Repository (IBSR), which is made available by the Center for Morphometric Analysis, Massachusetts General Hospital (http://www.cma.mgh.harvard.edu/ibsr). The MRI image database consists of two datasets for brain tissue segmentation of normal subjects, one with 20 subjects and another with 18 subjects. The first and second dataset are known as IBSR20 and IBSR18 respectively. IBSR20 and IBSR18 contain a T1-weighted (T1-w) and the manual segmentation (ground truth) for each image [234].

The IBSR20 dataset contains a three-dimensional T1-weighted MRI brain dataset for 20 normal subjects and the corresponding manual segmentation. The size of the volume is 256 × 256 × 65 voxels grid with 3.1 mm slice thickness.

The IBSR18 dataset contains a three-dimensional T1-weighted MRI brain data set for 18 normal subjects and the corresponding manual segmentation. The size of the volume is 256 × 256 × 128 voxels grid with 1.5 mm slice thickness [235]. IBSR18 scans present higher resolution and image quality than IBSR20, with no apparent acquisition artefacts that can bias the accuracy of some scans [236].

8.2.2 Evaluation

The manual segmentation of WM, GM, and CSF is utilized as the ground truth to evaluate the proposed method results. However, the provided labels consider the sulcal cerebrospinal fluid (SCSF) as part of GM not CSF. The difficulty of using the IBSR20 dataset is due to presence of some artefacts and intensity irregularities [236].

The three base clustering methods of k-means, FCM and SOM are applied along with the proposed method to segment a random slices of IBSR18 and slice 20 of the IBSR20 dataset, which was chosen randomly. Firstly, an evaluation about the ability of BPNN in imitating the performance of each of the three clustering methods is presented. As mentioned in Section 7.4.2, the trained BPNN models are chosen to utilize instead of the actual base clustering methods in the testing phase in order to reduce the testing computational complexity. The first five images were used for training and the remaining ones for testing. Differences in performance between the trained BPNN models and the actual base clustering methods are shown in Tables 8.15 and 8.16. Results reveal that the prediction error for the three clustering methods is quite small. Table 8.17 presents the average difference between the trained BPNN model results and the ground truth. The trained model of SOM method achieves the lower prediction error value compared to the other trained models.

Table 8.15 Average difference between each actual base clustering method and its corresponding trained BPNN model results

Base clustering : Trained BPNN	Prediction error
k-means and BPNN	0.131
FCM and BPNN	0.127
SOM and BPNN	0.149

Figure 8.15a shows one of the ground truth images used in the experiment. Figures 8.15b, 8.15c and 8.15d show the segmentation results of the three base clustering methods of k-means, FCM and SOM respectively. Figures 8.15e and 8.15f show the segmentation results of the proposed method after the majority voting and post-processing steps respectively. The figures show that the proposed method segmented the

Table 8.16 Average difference between two different clustering methods results

Clustering 1 : Clustering 2	$\frac{\text{Avg. different pixels}}{\text{Avg. total object pixels}}$
k-means and FCM	3280 : 46688
k-means and SOM	2081 : 46688
FCM and SOM	4840 : 46688

Table 8.17 Average difference between each trained BPNN model results and ground truth

Trained BPNN: Ground truth	Prediction error
Trained k-means and ground truth	0.189
Trained FCM and ground truth	0.193
Trained SOM and ground truth	0.186

brain image more efficiently than the other base methods, as they have a higher incorrect segmentation percentage compared to the proposed method. Different evaluation measurements are used to assess and compare the results, as explained below.

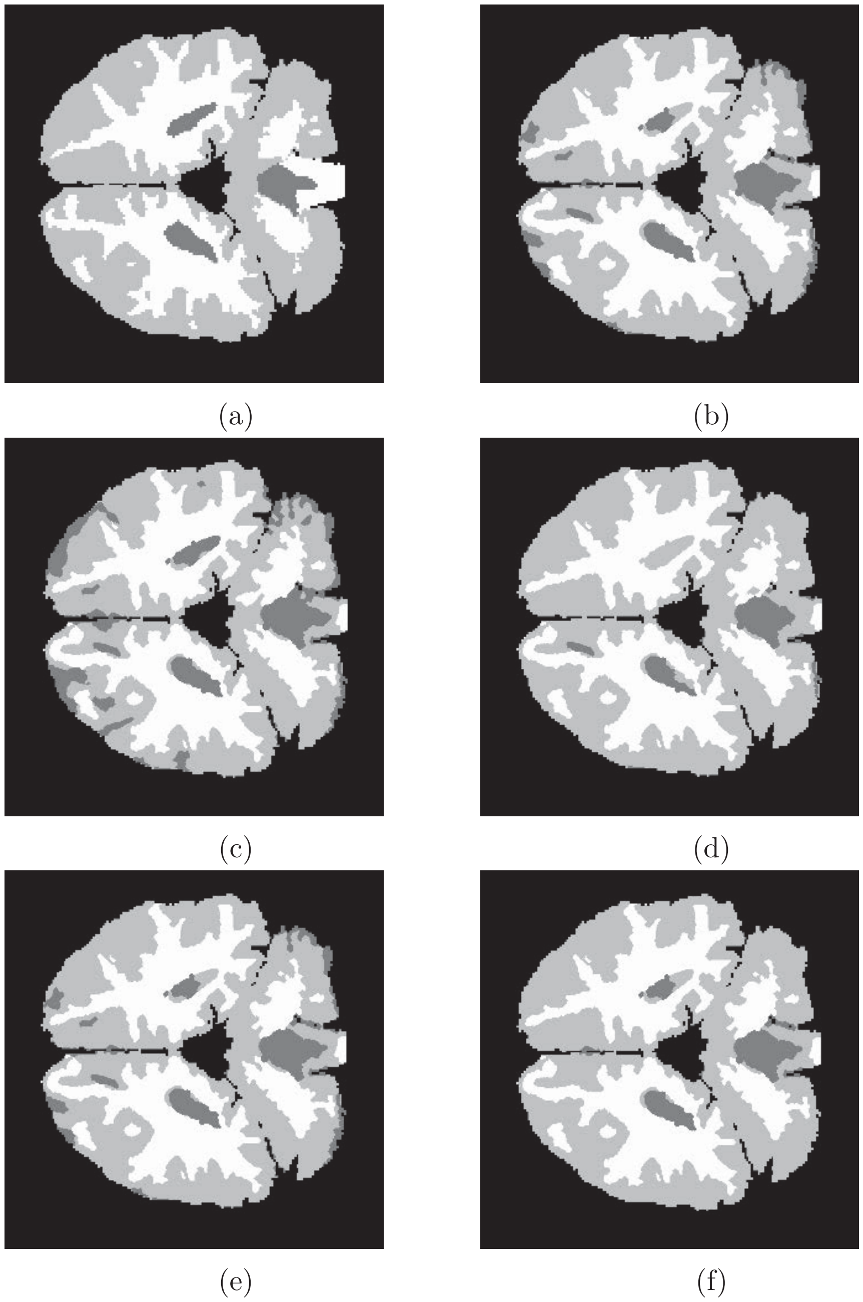
8.2.2.1 Spatial Overlaps

In order to evaluate the accuracy of the segmentation method, the Jaccard similarity (JS) metric is used [237], which is computed as the ratio between the intersection and union of the segmented and ground truth images. The range of JS is between 0 and 1. Higher JS values indicate that the segmented region matches more of the ground truth region. The JS is defined using Eq. 8.11.

$$JS = \frac{S_1 \cap S_2}{S_1 \cup S_2} \tag{8.11}$$

where S_1 is the ground truth image and S_2 is the segmented image.

Table 8.18 presents the results of JS using k-means, FCM, SOM and the proposed method. It is observed that the proposed method achieved a higher degree of similarity for all three classes (WM, GM and CSF) than the other three segmentation methods.



(a) (b) (c) (d) (e) (f)

Figure 8.15 Subject 12-3, slice 20: (a) Ground truth, (b) k-means, (c) FCM, (d) SOM, (e) proposed method after the majority voting step and (f) proposed method after applying the post-processing step

On the other hand, the other methods were found to perform quite well for the WM and GM classes, but the performance was noticeably lower for the CSF class.

Table 8.18 JS values of the proposed method and trained NN of k-means, FCM and SOM using random slice

Dataset	Class	k-means	FCM	SOM	Proposed
IBSR20	WM	0.706±0.099	0.704±0.096	0.706±0.101	**0.712±0.094**
	GM	0.694±0.076	0.633±0.086	0.730±0.076	**0.745±0.079**
	CSF	0.329±0.092	0.270±0.096	0.431±0.122	**0.545±0.105**
	Avg of tissues	0.576±0.055	0.536±0.064	0.622±0.061	**0.668±0.057**
IBSR18	WM	0.767±0.058	0.765±0.081	0.771±0.038	**0.788±0.030**
	GM	0.644±0.143	0.615±0.148	0.641±0.139	**0.759±0.075**
	CSF	0.429±0.083	0.406±0.072	0.432±0.087	**0.641±0.074**
	Avg of tissues	0.614±0.062	0.595±0.062	0.615±0.058	**0.729±0.038**

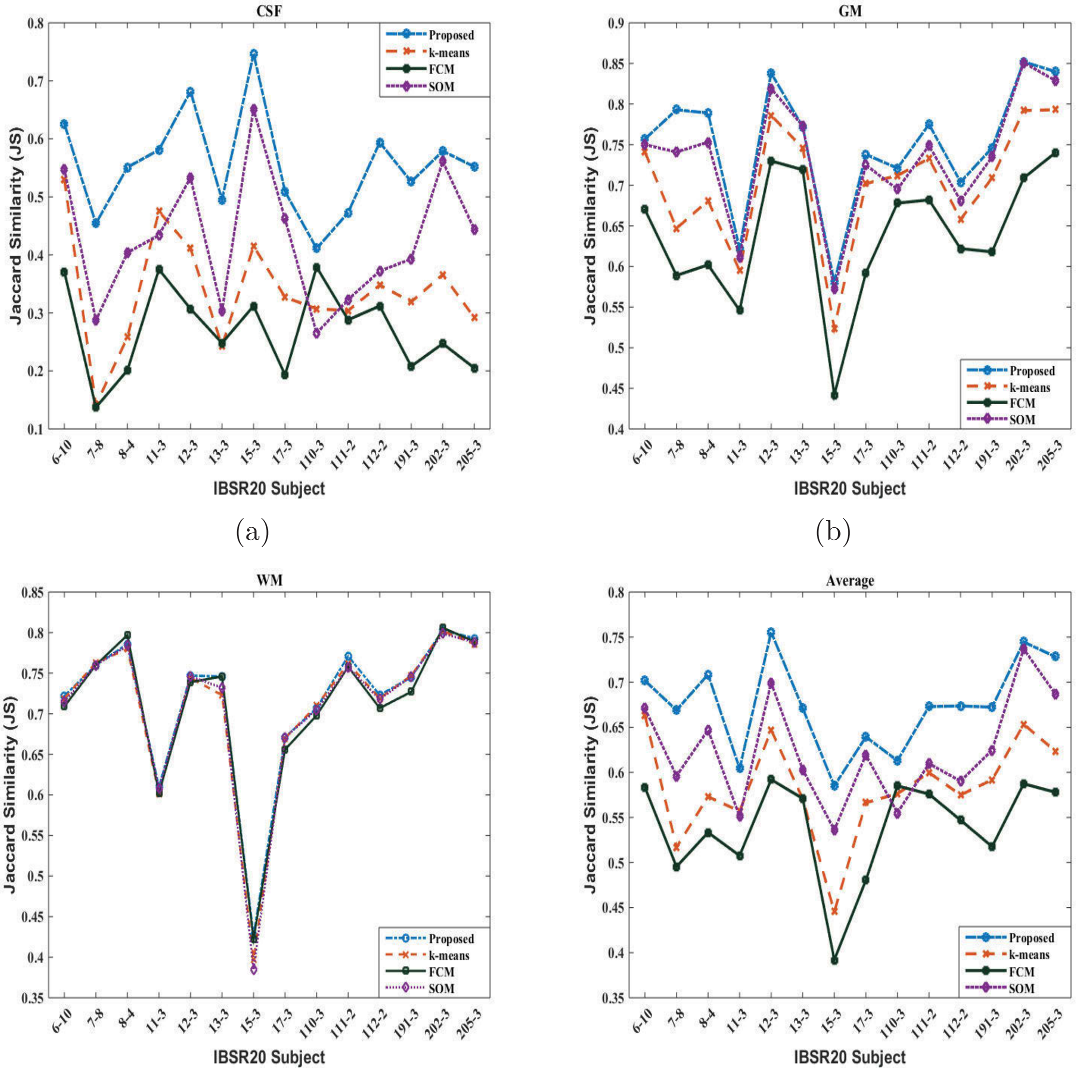
Figures 8.16a - 8.16d show the JS results for the IBSR20 subjects, slice 20. All the methods achieve a good segmentation results for the three classes, expect for some instances. For example, the WM class of subject (15-3). This could be caused by the high level of overlap in pixel intensity between the GM and WM classes as shown in Figure 8.17. Figure 8.16 also shows that the proposed method has better JS results for GM and CSF tissues compared to the other segmentation methods.

The Dice similarity coefficient (DSC) is a statistical validation metric that was proposed to evaluate the accuracy of segmentation methods [236]. It describes the overlap between the segmented and ground truth images using Eq. 8.12:

$$DSC = 2 \times \frac{S_1 \bigcap S_2}{\mid S_1 \mid + \mid S_2 \mid} = 2 \times \frac{TP}{TP + FP + TP + FN} \tag{8.12}$$

where TP (true positive) is the number of common pixels between the segmented and ground truth tissues, FN (false negative) is the number of pixels that are not detected in the segmented tissue and FP (false positive) is the number of pixels incorrectly assigned as tissue in the segmented image compared to the ground truth image as shown in Figure 8.18.

Table 8.19 shows the results of the DSC obtained using the three based segmentation methods and the proposed method. The results are consistent with that of Table 8.18



(a) (b)

(c) (d)

Figure 8.16 JS results for IBSR20 dataset, slice 20: (a) CSF, (b) GM, (c) WM and (d) average

in terms of achieving better segmentation than other methods and for scoring high values for the WM, GM and CSF tissues.

Figures 8.19a - 8.19d show the box plot of DSC for all of the segmentation methods. The proposed method achieves more accurate results compared to the other methods and scores the highest median results for all the tissues. However, the performance of the proposed method for the WM segmentation is only slightly better than the other methods, while the lowest DSC values of the GM and CSF tissues were obtained using

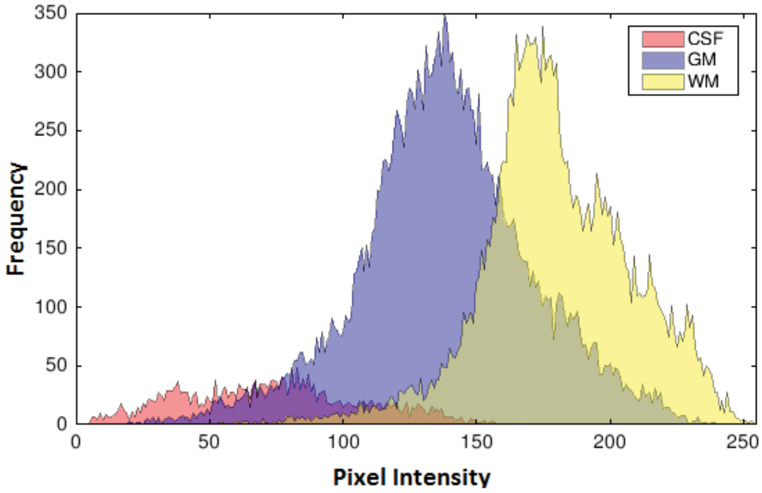


Figure 8.17 Pixel intensity overlapping between the brain tissues

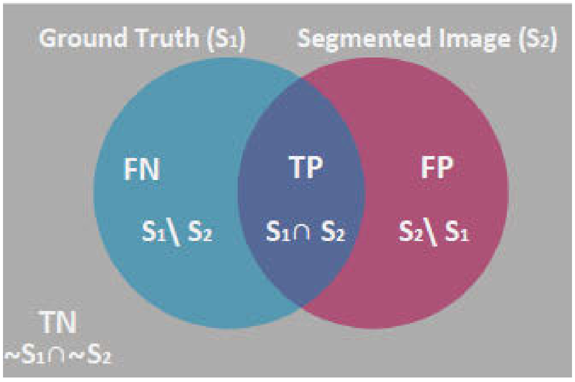


Figure 8.18 Venn diagram of true positive, true negative, false positive and false negative

FCM. Even though all methods achieved comparable results for the segmentation of WM, the proposed method was superior to the base clustering methods for the other two classes of GM and CSF and, accordingly, achieved on average better segmentation results.

The outcome of the segmentation method is also tested using Root Mean Square Error (RMSE). The RMSE is a statistical metric that finds the difference between the ground truth and segmented images. It is computed using Eq. 8.13.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{W}\sum_{j=1}^{H}(S_2 - S_1)^2}{W \times H}} \tag{8.13}$$

Table 8.19 DSC values of the proposed method and trained NN of k-means, FCM and SOM using random slice

Dataset	Class	k-means	FCM	SOM	Proposed
IBSR20	WM	0.824±0.079	0.822±0.074	0.823±0.081	0.829±0.073
	GM	0.818±0.055	0.769±0.066	0.841±0.052	0.852±0.053
	CSF	0.491±0.109	0.409±0.101	0.588±0.112	0.698±0.086
	Avg of tissues	0.710± 0.049	0.667±0.053	0.750±0.048	0.793±0.043
IBSR18	WM	0.885±0.036	0.882±0.051	0.887±0.023	0.891±0.020
	GM	0.784±0.104	0.762±0.112	0.781±0.101	0.861±0.052
	CSF	0.664±0.081	0.648±0.076	0.669±0.085	0.779±0.056
	Avg of tissues	0.777± 0.050	0.764±0.051	0.780±0.050	0.840±0.027

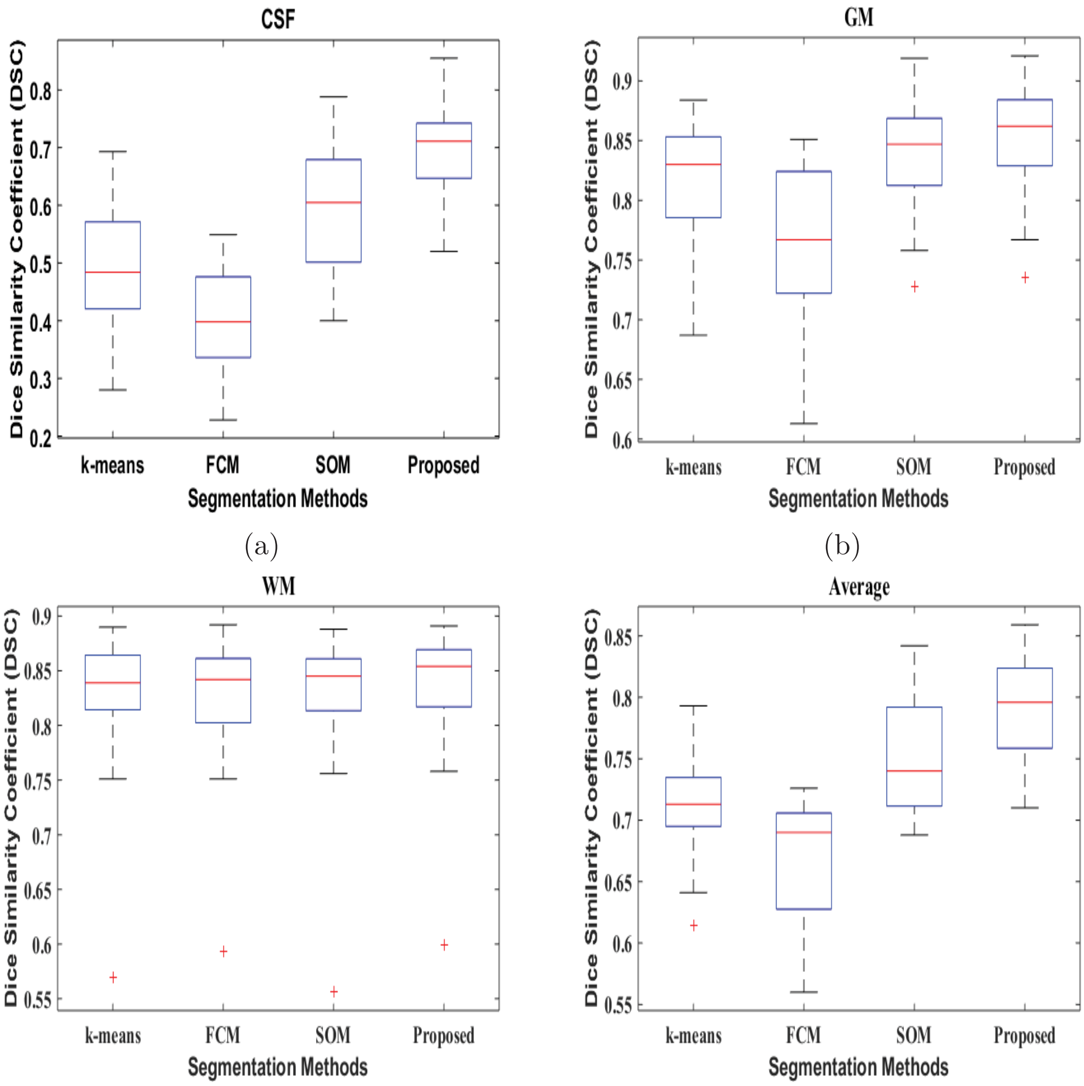
where W and H are the width and height of the ground truth ($S1$) and segmented ($S2$) images.

Table 8.20 shows the experimental results of the proposed method compared to the k-means, FCM and SOM. It can be observed from Table 8.20 that the proposed method obtained better results compared to the other methods in terms of RMSE values.

Table 8.20 RMSE of the proposed method and trained NN of k-means, FCM and SOM using slice number 20

Subject	k-means	FCM	SOM	Proposed
12-3	0.199	0.225	0.179	**0.176**
15-3	0.245	0.267	0.232	**0.230**
205-3	0.161	0.178	0.147	**0.143**
Average RMSE	0.198	0.219	0.182	**0.178**

Table 8.21 presents the results of a single NN that is trained directly using manual annotation (ground truth) and the results obtained using the proposed fusion method. The results show that the single NN that was trained directly using ground truth was not able to perform as well as the proposed method and could not recognize the under-represented class of the CSF tissue in the IBSR20 dataset.



(a)

(b)

(c)

(d)

Figure 8.19 DSC results for IBSR20 dataset,slice 20: (a) CSF, (b) GM, (c) WM and (d) average

8.2.2.2 Accuracy, Sensitivity and Specificity

Accuracy, sensitivity and specificity are used for the qualitative evaluation of the proposed method. Sensitivity refers to the ability of the clustering method to accurately

Table 8.21 A comparison between the proposed method versus a single neural network trained using the ground truth annotation

		NN trained using ground truth			Proposed method		
dataset	Class	JS	DSC	RMSE	JS	DSC	RMSE
IBSR20	WM	0.409	0.558	0.251	0.712	0.829	0.178
	GM	0.639	0.773		0.745	0.852	
	CSF	0.000	0.000		0.545	0.698	
	Average	0.349	0.444	0.251	0.668	0.793	0.178
IBSR18	WM	0444	0.579	0.241	0.788	0.891	0.172
	GM	0.568	0.705		0.759	0.861	
	CSF	0.284	0.412		0.641	0.779	
	Average	0.432	0.571	0.241	0.729	0.840	0.172

identify the tissue regions in the segmented image. It can be defined as shown in Eq. 8.14.

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{8.14}$$

The specificity, which is defined in Eq. 8.15, reflects the ability of the clustering method to accurately identify the non-tissue regions.

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{8.15}$$

where TN (true negative) is the number of pixels not labelled either in the segmented or ground truth tissues.

The accuracy of the clustering method is computed as the rate of correctly predicted over all predicted pixels using Eq. 8.16.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \tag{8.16}$$

Table 8.22 shows the qualitative performance of the proposed method compared to the k-means, FCM and SOM. The proposed method performs more accurately in identifying the WM, GM and CSF tissues than the other methods.

Table 8.22 Mean of Accuracy, Sensitivity and Specificity values of k-means, FCM, SOM and the proposed method using slice number 20

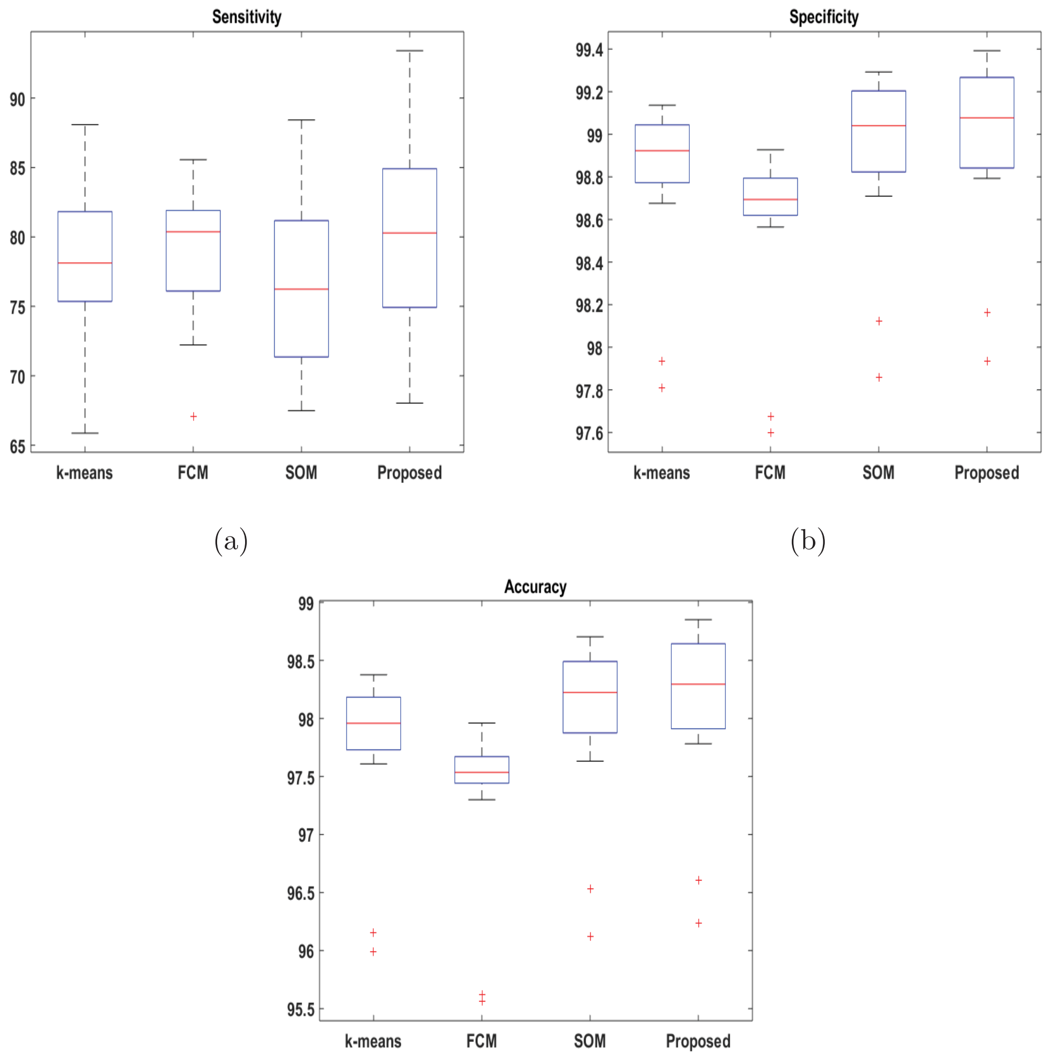
	Sensitivity	**Specificity**	**Accuracy**
k-means	76.77±6.798	98.74±0.437	97.66±0.791
FCM	77.97±5.981	98.52±0.466	97.21±0.859
SOM	75.66±6.975	98.86±0.433	97.91±0.778
Proposed	79.66±7.099	98.97±0.415	98.10±0.752

The spatial overlap metric does not estimate the overall performance in terms of error. Therefore, sensitivity, specificity and accuracy values are presented for each method in Table 8.22. The quantitative results show that the proposed method has better segmentation results for WM, GM and CSF tissues than the three base clustering techniques as shown in Figures 8.20a - 8.20c. The higher levels of qualitative evaluation metrics achieved by the proposed method demonstrate the benefit of the clustering confusion technique. This confirms the initial hypothesis that collecting the segmentation results from different clustering techniques gives the best combination result.

As stated in the experiment results, the segmentation results depends on the image resolution. To be more precise, the performance of the proposed method decreases in the presence of high pixel intensities overlap. It is worth mentioning here that all the images used to evaluate this method were obtained from real brain scan. Therefore, the noise permanently existed on all of the images due to the acquisition process.

8.2.2.3 Impact of SLIC Parameters

Figure 8.21 illustrates the impact of SLIC parameters on the segmentation results. Selecting a suitable number of superpixels is important for improving the overall segmentation accuracy and reducing the computational cost. A small number of superpixels tends to increase the superpixel size, which can guarantee a lower



(a)

(b)

(c)

Figure 8.20 (a) Sensitivity, (b) Specificity and (c) Accuracy results for IBSR20 dataset, slice 20

computational cost. However, a small number of superpixels (with a large superpixel size) may cause boundary superpixels to fall on the edge of two or more different classes, which causes inaccurate features and classification, while a large number of superpixels (with a small superpixel size) has a stronger chance of including only one type of tissue. Moreover, it is evident that the small compactness value ($m = 5$) produces better segmentation results in terms of DSC mainly because the brain tissues possess irregular shape, whereas large compactness values (particularly for $m = 20$) generate more regular shapes.

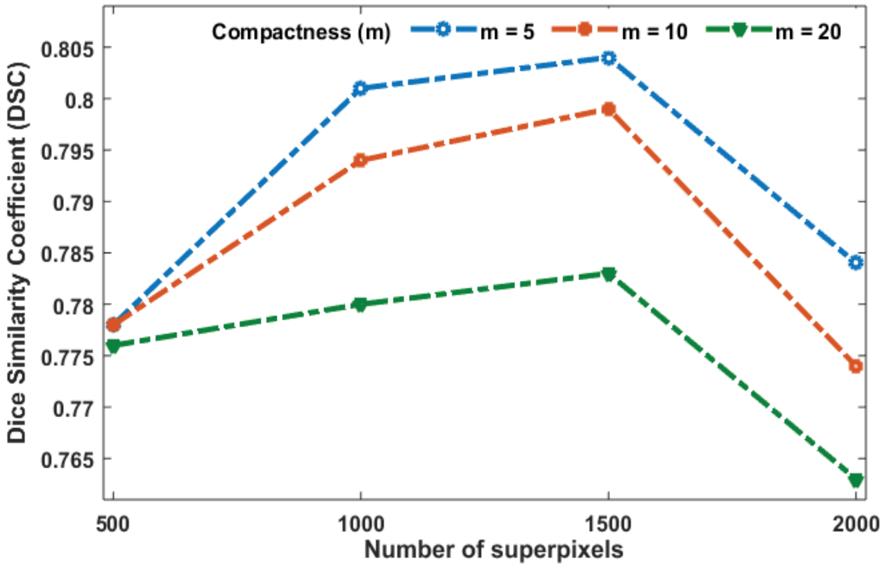


Figure 8.21 The impact of SLIC superpixel parameters (number of superpixels and compactness) on the DSC

Table 8.21 presents the results of a single NN that trained directly using manual annotation (ground truth), and the results obtained using the proposed fusion method. The results show that the single NN that was trained directly using ground truth was not able to perform as good as the proposed method and could not recognize the under-represented class of the CSF tissue in the IBSR20 dataset.

8.2.2.4 Computational Cost

Table 8.23 The computational cost for the k-means, FCM and SOM clustering techniques and the trained NN of single clustering technique in the testing phase

	k-means	FCM	SOM	Trained NN (testing phase)
Computational cost (sec)	1.72	1.86	8.51	0.035

Table 8.23 illustrates the computational time for the three different clustering techniques and the trained NN model of single clustering algorithm in the testing phase (we assume

that the three clustering algorithms would run in parallel). The table shows that the base clustering methods have higher computational times compared to the trained NN model. These results prove that the computational cost of the proposed method is noticeably lower than that of the base clustering methods (even if the three trained NNs are to run in series).

8.3 Summary

The evaluation process of steganography and segmentation is a fundamental step for assessing the performance and selecting the appropriate technique. This chapter utilized different metrics to evaluate the efficiency of the proposed methodologies.

The embedding capacity, imperceptibility and security requirements of the first proposed image steganography method are tested on a large dataset that consists of 10,000 images of size 512 × 512. The results demonstrated that the stego images produced using the proposed IWT algorithm achieved a high level of robustness and visual quality with acceptable embedding capacity compared to 1*bpp* and *nbpp* proposed methods in the spatial domain. On the other hand, the spatial domain methods achieved a good embedding payload and maintained low perceptibility, where it achieved a PSNR value of 50.53 dB with 70% embedding rate. The proposed method demonstrated considerable improvements in term of image quality and security (tested using a well-established textural steganalysis method) compared with other popular steganography methods.

Moreover, the medical steganography methodology has been tested on 100 MRI images of size256 × 256. Experimental results indicated that the proposed algorithm achieved high embedding rate with low level of embedding distortion, and hence provided a good compromise between payload and quality of the stego images. Moreover, steganalysis results of the proposed algorithms obtained using the Li-110D technique proved the superiority of the proposed algorithm compared to the well-known PVD algorithm and two of its variants.

Finally, the performance of MR brain image segmentation method has been evaluated using different statistical measurements that include: jaccard similarity, dice similarity coefficient, root mean square error, accuracy, sensitivity and specificity. The experimental results show that the proposed method can achieve a good segmentation results than the based clustering techniques.

CHAPTER 9

Conclusions and Future Work

This chapter concludes the thesis with a summary of the presented methodologies. It begins by summarizing the research contributions, then outlines future research that would lead to more development of these research fields.

9.1 Conclusions

The main objective of the research introduced in this thesis is to investigate digital image steganography and segmentation in order to offer a methodical way for designing and developing them, with a particular concentration on medical images security and MR brain image segmentation. Digital image steganography is an advancing innovation, which has an incredible potential for handling multimedia information security. On the other hand, digital medical images are the result of the developed technology that has allowed health care systems to exchange medical images and patient records remotely between different clinics and hospitals. In this manner, medical images are vulnerable to increased security threats while being transmitted over public networks. Moreover, traditional security techniques, such as encryption algorithms, do not offer the required protection. Therefore, this research examines digital image steganography in order to address this issue.

Despite of its distinct benefits, digital image steganography has not been extensively accepted in practice. As stated in Section 1.3, digital steganography suffers from some limitations. Additionally, validation of the appropriateness of the existing steganography techniques for specific applications is more difficult because there is no standard model that can evaluate the performance of steganography for particular applications. The steganography requirements are also not well determined for interpreting and reflecting accurate results when developing and applying steganography in practice. To be more precise, there is no any standard definition for the steganography requirements of medical imaging systems.

The most critical requirements of any steganographic system are imperceptibility and capacity. Subsequently, this thesis addresses and enhances these two major requirements of digital steganography techniques: the embedding capacity and stego image quality (imperceptibility). In this thesis, novel and efficient image steganography methods are proposed to improve the stego image quality and increase the embedding capacity. Further, the precision of PSNR and wPSNR are investigated. These two measures are essential for evaluating the resultant (stego) image quality in order to assess the performance of the steganography method.

Computer vision algorithms can also help biomedical image processing to improve and accomplish their tasks, such as restoration, registration, segmentation or tracking. Image segmentation aims to divide the image into meaningful and non-overlapping regions. It is considered an essential process in many important biomedical applications, for example tumour detection, quantitative tissue analysis and computer-integrated surgery. Image segmentation can be categorized based on either the level of human interaction or the technique type, i.e., pixel-based, edge-based or region-based methods. The accurate segmentation of MR brain images is a complicated and challenging problem due to the variable tissue types.

A number of original contributions have been introduced in this thesis as follows:

(1) The development of an edge detection algorithm: The utilization of the traditional edge detection based steganography algorithms does not guarantee identifying the same edge sets between the cover and stego images. In order to enable the correct extraction of the concealed message from the stego image, a new and simple edge detection algorithm is proposed to discover the edge (sharp) regions of the cover image, such that the two edge images generated using the original cover and stego images are identical.

(2) Increasing the embedding capacity without compromising the imperceptibility: A high embedding capacity without compromising the quality of the stego image was achieved by introducing the new edge detection algorithm. It is inspired by the fact that the human visual system (HVS) is less sensitive to changes in the edge areas compared to smooth areas. Thus, strong edges will enable the embedding of more bits than less strong ones.

(3) Improving imperceptibility and security: A high stego image quality was achieved by utilizing coding theory. Coding theory aims to reduce the difference between the cover and stego images.

(4) The development of a steganographic system: The edge detection and XOR coding are combined to develop a steganography algorithm that conceals a secret message in either the spatial domain or an integer wavelet transform (IWT)-based transform domain of the cover image. In order to enhance the imperceptibility, the edge detection algorithm identifies sharp edges in the cover image for embedding in order to cause less degradation to the image quality than embedding in a pre-specified set of pixels that do not differentiate between sharp and smooth areas. In addition to this, the secret data is coded before embedding using an XOR based formula to minimize the difference between the cover and stego images.

(5) The development of a medical image steganographic system: An efficient combination between cryptography and information-hiding techniques is presented in order to ensure the security and privacy of patients' information

through concealing the meaning of the secret data and its existence. Medical images have to be carefully processed, as introducing modifications to their important regions, known as the region of interest (ROI), may impact the diagnosis of patients' conditions, therefore this research refrains from making any modifications to the ROI and the algorithm developed instead conceals the secret data in the region of non interest (RONI). Moreover, based on the characteristic of the human visual perception, this research focused on embedding data into the sharp edges of the RONI, as this would attract less attention from intruders about the existence of secret data in the image. To further enhance the embedding efficiency and increase data security, coding algorithms were incorporated that helped to reduce modifications to the original (cover) images.

(6) The development of brain image segmentation based on clustering fusion: This research presents an efficient fully-automatic brain tissue segmentation algorithm based on a clustering fusion technique. The segmentation method combines the SLIC superpixel, three clustering techniques and a neural network to divide the MR brain image into three tissues of WM, GM and CSF.

9.2 Future Work

Further work includes increasing the embedding capacity of the proposed steganography methods which were discussed in Chapters 5 and 6 by utilizing the edge detection algorithm. Also, there are different directions to extend the proposed method of MR brain image segmentation presented in Chapter 7.

The following directions are recommended for future research.

- The proposed edge detection algorithm can be enhanced through the more accurate identification of edge regions while maintaining the identification of the same edge sets in both the cover and stego images.
- The embedding capacity can be improved by utilizing more pixels per block for embedding.

- Enabling the automatic identification of ROI in medical images.
- Applying the wavelet transform for the medical image steganography method to ensure the robustness against image processing operations.
- Further analysis is needed to optimally identify number of classes of the clustering algorithm.the number of classes.
- In order to separate brain image into the background and object regions using threshold-based segmentation, additional analysis is required to generate dynamic threshold instead of fixed threshold value for the dataset.
- Extending the segmentation method to 3D images.

Bibliography

[1] S. Luo, R. Li, and S. Ourselin, “A new deformable model using dynamic gradient vector flow and adaptive balloon forces,” in *APRS Workshop on Digital Image Computing*, pp. 9–14, 2003.

[2] M. A. Balafar, A. R. Ramli, M. I. Saripan, and S. Mashohor, “Review of brain mri image segmentation methods,” *Artificial Intelligence Review*, vol. 33, no. 3, pp. 261–274, 2010.

[3] T. Morkel, J. H. Eloff, and M. S. Olivier, “An overview of image steganography.,” in *ISSA*, pp. 1–11, 2005.

[4] P. Gupta and E. S. Singh, “Review paper on brain image segmentation using chan-vese algorithm and active contours,” *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 3, pp. 3766–3769, 2014.

[5] N. research Council *et al.*, *Networking health: prescriptions for the internet*. National Academies Press, 2000.

[6] D. Salomon, *Data privacy and security: encryption and information hiding*. Springer Science & Business Media, 2003.

[7] Y. Srinivasan, B. Nutter, S. Mitra, B. Phillips, and D. Ferris, “Secure transmission of medical records using high capacity steganography,” in *Computer-Based Medical Systems, 2004. CBMS 2004. Proceedings. 17th IEEE Symposium on*, pp. 122–127, IEEE, 2004.

[8] J. Wang, J. Kong, Y. Lu, M. Qi, and B. Zhang, “A modified fcm algorithm for mri brain image segmentation using both local and non-local spatial constraints,” *Computerized medical imaging and graphics*, vol. 32, no. 8, pp. 685–698, 2008.

[9] E. Abdel-Maksoud, M. Elmogy, and R. Al-Awadi, “Brain tumor segmentation based on a hybrid clustering technique,” *Egyptian Informatics Journal*, vol. 16, no. 1, pp. 71–81, 2015.

[10] D. Selvaraj and R. Dhanasekaran, “Mri brain image segmentation techniques-a review,” *Indian Journal of Computer Science and Engineering (IJCSE), ISSN*, pp. 0976–5166, 2013.

[11] Y. Yin, I. Kaku, J. Tang, and J. Zhu, *Data mining: Concepts, methods and applications in management and engineering design.* Springer Science & Business Media, 2011.

[12] A. Singh, A. Vaish, and P. K. Keserwani, “Information security: Components and techniques,” *International Journal*, vol. 4, no. 1, 2014.

[13] Y. S. Feruza and T.-h. Kim, “It security review: Privacy, protection, access control, assurance and system security,” *International journal of multimedia and ubiquitous engineering*, vol. 2, no. 2, pp. 17–32, 2007.

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

[15] A. Shaik, V. Thanikaiselvan, and R. Amitharajan, “Data security through data hiding in images: A review,” *Journal of Artificial Intelligence*, vol. 10, no. 1, pp. 1–21, 2017.

[16] B. Li, J. He, J. Huang, and Y. Q. Shi, “A survey on image steganography and steganalysis,” *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, no. 2, pp. 142–172, 2011.

[17] C. Shen, H. Zhang, D. Feng, Z. Cao, and J. Huang, “Survey of information security,” *Science in China Series F: Information Sciences*, vol. 50, no. 3, pp. 273–298, 2007.

[18] M. S. Subhedar and V. H. Mankar, “Current status and key issues in image steganography: A survey,” *Computer Science Review*, vol. 13, pp. 95–113, 2014.

[19] J. Zöllner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G. Wicke, and G. Wolf, “Modeling the security of steganographic systems,” in *Information Hiding*, pp. 344–354, Springer, 1998.

[20] P. Thomas, “Literature survey on modern image steganographic techniques,” *International Journal of Engineering Research & Technology (IJERT) Vol*, vol. 2, pp. 2278–0181, 2013.

[21] A. Rengarajan, “Information security a random image steganography perspective,” *SASTRA University*, 2014. Available at http://hdl.handle.net/10603/17471.

[22] J. Cummins, P. Diskin, S. Lau, and R. Parlett, “Steganography and digital watermarking,” *School of Computer Science, The University of Birmingham*, vol. 14, p. 60, 2004.

[23] N. Provos and P. Honeyman, “Hide and seek: An introduction to steganography,” *Security & Privacy, IEEE*, vol. 1, no. 3, pp. 32–44, 2003.

[24] S. Kumar, *Image Steganography Using Improved LSB And Exor Encryption Algorithm.* PhD thesis, THAPAR UNIVERSITY PATIALA, 2014.

[25] S. Sarreshtedari and S. Ghaemmaghami, "High capacity image steganography in wavelet domain," in *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pp. 1–5, IEEE, 2010.

[26] R. Waterfield and C. Dewald, *The histories.* Oxford Paperbacks, 2008.

[27] J. Watkins, "Steganography-messages hidden in bits," *Multimedia Systems Coursework, Dept of Electronics and CS, University of Southampton, SO17 1BJ, UK*, 2001.

[28] R. J. Anderson and F. A. Petitcolas, "On the limits of steganography," *Selected Areas in Communications, IEEE Journal on*, vol. 16, no. 4, pp. 474–481, 1998.

[29] G. C. Kessler and C. Hosmer, "An overview of steganography," *Advances in Computers*, vol. 83, no. 1, pp. 51–107, 2011.

[30] S. Katzenbeisser and F. Petitcolas, *Information hiding techniques for steganography and digital watermarking.* Artech house, 2000.

[31] R. Chandramouli, M. Kharrazi, and N. Memon, "Image steganography and steganalysis: Concepts and practice," in *International Workshop on Digital Watermarking*, pp. 35–49, Springer, 2003.

[32] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital watermarking and steganography.* Morgan Kaufmann, 2007.

[33] E. Zielińska, W. Mazurczyk, and K. Szczypiorski, "Trends in steganography," *Communications of the ACM*, vol. 57, no. 3, pp. 86–95, 2014.

[34] D. Vitaliev, "Digital security and privacy for human rights defenders.," *The International Foundation for Human Right Defenders*, pp. 77–81, 2007.

[35] N. Provos and P. Honeyman, "Detecting steganographic content on the internet," tech. rep., Center for Information Technology Integration, 2001.

[36] E. Cole, "Hiding in plain sight," *Steganography and the Art of Covert Communication, Wiley*, 2003.

[37] Z. K. Al-Ani, A. Zaidan, B. Zaidan, H. Alanazi, *et al.*, "Overview: Main fundamentals for steganography," *arXiv preprint arXiv:1003.4086*, 2010.

[38] G. Swain and S. K. Lenka, "Classification image steganography techniques in spatial domain: A study," *Int J Comput Sci Eng Tech*, vol. 5, no. 3, pp. 219–32, 2014.

[39] D. Bandyopadhyay, K. Dasgupta, J. Mandal, and P. Dutta, "A novel secure image steganography method based on chaos theory in spatial domain," *International Journal of Security*, 2014.

[40] M. Ghebleh and A. Kanso, “A robust chaotic algorithm for digital image steganography,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 6, pp. 1898–1907, 2014.

[41] M. Hussain and M. Hussain, “A survey of image steganography techniques,” *International Journal of advanced Science and Technology. Vol.*, vol. 54, pp. 113–123, 2013.

[42] I. J. Cox, M. Miller, and J. Bloom, “J. fridrich, t. kalker, digital watermarking and steganography,” 2008.

[43] G. Kipper, *Investigator’s guide to steganography.* Auerbach Publications A CRC Press Company, Boca Raton, London, New York, Washington, DC, 2003.

[44] A. Al-Mohammad, *Steganography-based secret and reliable communications: Improving steganographic capacity and imperceptibility.* PhD thesis, Brunel University, School of Information Systems, Computing and Mathematics Theses, 2010.

[45] O. Khalind and B. Y. Y. Aziz, “Lsb steganography with improved embedding efficiency and undetectability,” *Computer Science & Information Technology*, vol. 5, no. 1, pp. 89–105, 2015.

[46] S. Kumar and S. Muttoo, “A comparative study of image steganography in wavelet domain,” *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 2, pp. 91–101, 2013.

[47] W. Hong, C.-L. Pan, T.-S. Chen, W.-Y. Ji, and Y.-K. Wang, “A novel data embedding method for high payload using improved pixel segmentation strategy,” in *Electrical Power Systems and Computers*, pp. 89–95, Springer, 2011.

[48] W.-J. Chen, C.-C. Chang, and T. H. N. Le, “High payload steganography mechanism using hybrid edge detector,” *Expert Systems with Applications*, vol. 37, no. 4, pp. 3292 – 3301, 2010.

[49] M. Tang, J. Hu, and W. Song, “A high capacity image steganography using multilayer embedding,” *Optik-International Journal for Light and Electron Optics*, vol. 125, no. 15, pp. 3972–3976, 2014.

[50] N. Raftari and A. M. E. Moghadam, “Digital image steganography based on assignment algorithm and combination of dct-iwt,” in *Computational Intelligence, Communication Systems and Networks (CICSyN), 2012 Fourth International Conference on*, pp. 295–300, IEEE, 2012.

[51] D. Singla and M. Juneja, “An analysis of edge based image steganography techniques in spatial domain,” in *Engineering and Computational Sciences (RAECS), 2014 Recent Advances in*, pp. 1–5, IEEE, 2014.

[52] C. A. Stanley, “Pairs of values and the chi-squared attack,” *Department of Mathematics, Iowa State University*, 2005.

[53] A. Al-Ataby and F. Al-Naima, “A modified high capacity image steganography technique based on wavelet transform,” *changes*, vol. 4, p. 6, 2008.

[54] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, “Multimedia data-embedding and watermarking technologies,” *Proceedings of the IEEE*, vol. 86, no. 6, pp. 1064–1087, 1998.

[55] J. Fridrich, P. Lisonˇek, and D. Soukal, “On steganographic embedding efficiency,” in *Information Hiding: 8th International Workshop, IH 2006, Alexandria, VA, USA, July 10-12, 2006, Revised Seleceted Papers*, vol. 4437, p. 282, Springer Science & Business Media, 2007.

[56] R. Roy and S. Changder, “Quality evaluation of image steganography techniques: A heuristics based approach,” *International Journal of Security and Its Applications*, vol. 10, no. 4, pp. 179–196, 2016.

[57] F. Cayre, C. Fontaine, and T. Furon, “Watermarking security part one: theory,” in *Electronic Imaging 2005*, pp. 746–757, International Society for Optics and Photonics, 2005.

[58] O. S. Khalind, *New methods to improve the pixel domain steganography, steganalysis, and simplify the assessment of steganalysis tools.* PhD thesis, University of Portsmouth, 2015.

[59] A. K. Jain and U. Uludag, “Hiding fingerprint minutiae in images,” in *Proceedings of 3rd Workshop on Automatic Identification Advanced Technologies*, pp. 97–102, 2002.

[60] N. F. Johnson and S. Jajodia, “Exploring steganography: Seeing the unseen,” *Computer*, vol. 31, no. 2, pp. 26–34, 1998.

[61] W. Mazurczyk, “Voip steganography and its detection—a survey,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 2, p. 20, 2013.

[62] H. Tian, K. Zhou, H. Jiang, Y. Huang, J. Liu, and D. Feng, “An adaptive steganography scheme for voice over ip,” in *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pp. 2922–2925, IEEE, 2009.

[63] B. Zaidan, A. Zaidan, and M. Mat Kiah, “Impact of data privacy and confidentiality on developing telemedicine applications: A review participates opinion and expert concerns,” *Int. J. Pharmacol*, vol. 7, no. 3, pp. 382–387, 2011.

[64] P. Thiyagarajan, G. Aghila, and V. P. Venkatesan, “Stego-image generator (sig)-building steganography image database,” in *Advances in Digital Image Processing and Information Technology*, pp. 257–267, Springer, 2011.

[65] D. Artz, “Digital steganography: hiding data within data,” *IEEE Internet computing*, vol. 5, no. 3, pp. 75–80, 2001.

[66] Y. Q. Shi, C. Chen, G. Xuan, and W. Su, “Steganalysis versus splicing detection,” in *International Workshop on Digital Watermarking*, pp. 158–172, Springer, 2007.

[67] T. Holotyak, J. Fridrich, and S. Voloshynovskiy, "Blind statistical steganalysis of additive steganography using wavelet higher order statistics," in *IFIP International Conference on Communications and Multimedia Security*, pp. 273–274, Springer, 2005.

[68] S. Trivedi and R. Chandramouli, "Active steganalysis of sequential steganography," in *Electronic Imaging 2003*, pp. 123–130, International Society for Optics and Photonics, 2003.

[69] P. Wayner, *Disappearing Cryptography: Information Hiding: Steganography & Watermarking.* Morgan Kaufmann, 2002.

[70] P. A. Watters, F. Martin, and S. Stripf, "Visual steganalysis of lsb-encoded natural images," in *Information Technology and Applications, 2005. ICITA 2005. Third International Conference on*, vol. 1, pp. 746–751, IEEE, 2005.

[71] A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," in *International workshop on information hiding*, pp. 61–76, Springer, 1999.

[72] M. Köhler, I. Lukić, and V. Križanović Čik, "Protecting information with subcodstanography," *Security and Communication Networks*, vol. 2017, 2017.

[73] M. Zamani, A. A. Manaf, and R. Ahmad, "Knots of substitution techniques of audio steganography," in *The 2009 International Conference on Telecom Technology and Applications*, pp. 415–419, 2009.

[74] G. Sravanthi, B. S. Devi, S. Riyazoddin, and M. J. Reddy, "A spatial domain image steganography technique based on plane bit substitution method," *Global Journal of Computer Science and Technology Graphics & Vision, 12 (15)*, 2012.

[75] A. D. Ker, "A general framework for structural steganalysis of lsb replacement," in *Information Hiding*, pp. 296–311, Springer, 2005.

[76] J. Qin, X. Sun, X. Xiang, and Z. Xia, "Steganalysis based on difference statistics for lsb matching steganography," *Inform. Technol. J*, vol. 8, pp. 1281–1286, 2009.

[77] A. Kanso and H. S. Own, "Steganographic algorithm based on a chaotic map," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 8, pp. 3287–3302, 2012.

[78] C.-S. Chan, "On using lsb matching function for data hiding in pixels," *Fundamenta Informaticae*, vol. 96, no. 1, pp. 49–59, 2009.

[79] R. Roy, S. Changder, A. Sarkar, and N. C. Debnath, "Evaluating image steganography techniques: Future research challenges," in *Computing, Management and Telecommunications (ComManTel), 2013 International Conference on*, pp. 309–314, IEEE, 2013.

[80] Y. Luo, X. Li, and B. Yang, "Locating steganographic payload for lsb matching embedding," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pp. 1–6, IEEE, 2011.

[81] D.-C. Wu and W.-H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognition Letters*, vol. 24, no. 9, pp. 1613–1626, 2003.

[82] K.-C. Chang, P. S. Huang, T.-M. Tu, and C.-P. Chang, "Adaptive image steganographic scheme based on tri-way pixel-value differencing," in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pp. 1165–1170, IEEE, 2007.

[83] K.-C. Chang, C.-P. Chang, P. S. Huang, and T.-M. Tu, "A novel image steganographic method using tri-way pixel-value differencing," *Journal of multimedia*, vol. 3, no. 2, pp. 37–44, 2008.

[84] W. M. Abduallah, A. M. S. Rahma, and A.-S. K. Pathan, "Mix column transform based on irreducible polynomial mathematics for color image steganography: A novel approach," *Computers & Electrical Engineering*, vol. 40, no. 4, pp. 1390 – 1404, 2014.

[85] S. Bhattacharyya and G. Sanyal, "Hiding data in images using pixel mapping method (PMM).," in *Security and management*, pp. 683–689, 2010.

[86] S. Bhattacharyya, L. Kumar, and G. Sanyal, "A novel approach of data hiding using pixel mapping method (PMM)," *International Journal of Computer Science & Information Security*, p. 1, 2010.

[87] H.-W. Tseng and C.-C. Chang, "Steganography using jpeg-compressed images," in *Computer and Information Technology, 2004. CIT'04. The Fourth International Conference on*, pp. 12–17, IEEE, 2004.

[88] F. Alturki and R. Mersereau, "Secure blind image steganographic technique using discrete fourier transformation," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 2, pp. 542–545, IEEE, 2001.

[89] J. Kodovsky and J. Fridrich, "Quantitative structural steganalysis of jsteg," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 681–693, 2010.

[90] A. Westfeld, "F5—a steganographic algorithm," in *Information hiding*, pp. 289–302, Springer, 2001.

[91] I. Banerjee, S. Bhattacharyya, and G. Sanyal, "Hiding & analyzing data in image using extended pmm," *Procedia Technology*, vol. 10, pp. 157–166, 2013.

[92] W. Luo, F. Huang, and J. Huang, "Edge adaptive image steganography based on lsb matching revisited," *Information Forensics and Security, IEEE Transactions on*, vol. 5, no. 2, pp. 201–214, 2010.

[93] L. Li, B. Luo, Q. Li, and X. Fang, "A color images steganography method by multiple embedding strategy based on sobel operator," in *Multimedia Information Networking and Security, 2009. MINES'09. International Conference on*, vol. 2, pp. 118–121, IEEE, 2009.

[94] Y. Bassil, "Image steganography based on a parameterized canny edge detection algorithm.," *International Journal of Computer Applications*, vol. 60, 2012.

[95] M. R. Modi, S. Islam, and P. Gupta, "Edge based steganography on colored images," in *Intelligent Computing Theories*, pp. 593–600, Springer, 2013.

[96] W.-J. Chen, C.-C. Chang, and T. H. N. Le, "High payload steganography mechanism using hybrid edge detector," *Expert Systems with applications*, vol. 37, no. 4, pp. 3292–3301, 2010.

[97] H.-W. Tseng and H.-S. Leng, "High-payload block-based data hiding scheme using hybrid edge detector with minimal distortion," *IET Image Processing*, vol. 8, no. 11, pp. 647–654, 2014.

[98] S. Sun, "A novel edge based image steganography with 2 k correction and huffman encoding," *Information Processing Letters*, vol. 116, no. 2, pp. 93–99, 2016.

[99] J. Bai, C.-C. Chang, T.-S. Nguyen, C. Zhu, and Y. Liu, "A high payload steganographic algorithm based on edge detection," *Displays*, 2017.

[100] J. Fridrich, P. Lisoněk, and D. Soukal, "On steganographic embedding efficiency," in *International Workshop on Information Hiding*, pp. 282–296, Springer, 2006.

[101] C. Liu, X. Li, X. Lu, and B. Yang, "Improving embedding efficiency by incorporating sdcs and wpc," in *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pp. 1018–1021, IEEE, 2009.

[102] R. Crandall, "Some notes on steganography," *Posted on steganography mailing list*, 1998.

[103] R. Zhang, V. Sachnev, and H. J. Kim, "Fast bch syndrome coding for steganography," in *International Workshop on Information Hiding*, pp. 48–58, Springer, 2009.

[104] R. J. Mstafa and K. M. Elleithy, "A high payload video steganography algorithm in dwt domain based on bch codes (15, 11)," in *Wireless Telecommunications Symposium (WTS), 2015*, pp. 1–8, IEEE, 2015.

[105] R. J. Mstafa and K. M. Elleithy, "A highly secure video steganography using hamming code (7, 4)," in *Systems, Applications and Technology Conference (LISAT), 2014 IEEE Long Island*, pp. 1–6, IEEE, 2014.

[106] J. Bai and C.-C. Chang, "A high payload steganographic scheme for compressed images with hamming code.," *IJ Network Security*, vol. 18, no. 6, pp. 1122–1129, 2016.

[107] P. Praveenkumar, K. Thenmozhi, J. Rayappan, and R. Amirtharajan, "Reversible steganography on ofdm channel-a role of rs coding," *Inform. Technol. J*, vol. 13, pp. 2052–2056, 2014.

[108] M. H. Shirafkan, E. Akhtarkavan, and J. Vahidi, "A image steganography scheme based on discrete wavelet transform using lattice vector quantization and reed-solomon encoding," in *Knowledge-Based Engineering and Innovation (KBEI), 2015 2nd International Conference on*, pp. 177–182, IEEE, 2015.

[109] T. Filler, J. Judas, and J. Fridrich, "Minimizing embedding impact in steganography using trellis-coded quantization," in *IS&T/SPIE Electronic Imaging*, pp. 754105–754105, International Society for Optics and Photonics, 2010.

[110] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920–935, 2011.

[111] C.-L. Hou, C. Lu, S.-C. Tsai, and W.-G. Tzeng, "An optimal data hiding scheme with tree-based parity check," *Image Processing, IEEE Transactions on*, vol. 20, no. 3, pp. 880–886, 2011.

[112] W. Hong, T. S. Chen, Z. Yin, B. Luo, and Y. Ma, "Data hiding in ambtc images using quantization level modification and perturbation technique," *Multimedia Tools and Applications*, pp. 1–22, 2016.

[113] B. Feng, W. Lu, and W. Sun, "Secure binary image steganography based on minimizing the distortion on the texture," *IEEE transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 243–255, 2015.

[114] E. Ghasemi, J. Shanbehzadeh, and N. Fassihi, "High capacity image steganography using wavelet transform and genetic algorithm," in *International Multi Conference of Engineers and Computer Scientists*, vol. 1, 2011.

[115] S. Bhattacharyya and G. Sanyal, "A robust image steganography using dwt difference modulation (dwtdm)," *International Journal of Computer Network and Information Security*, vol. 4, no. 7, p. 27, 2012.

[116] M. Parul and D. H. Rohil, "Optimized image steganography using discrete wavelet transform (dwt)," *International Journal of Recent Development in Engineering and Technology*, vol. 2, no. 2, pp. 75–81, 2014.

[117] M. I. S. Reddy and A. S. Kumar, "Secured data transmission using wavelet based steganography and cryptography by using aes algorithm," *Procedia Computer Science*, vol. 85, pp. 62–69, 2016.

[118] D. Baby, J. Thomas, G. Augustine, E. George, and N. R. Michael, "A novel dwt based image securing method using steganography," *Procedia Computer Science*, vol. 46, pp. 612–618, 2015.

[119] S. Hemalatha, D. U. Acharya, A. Renuka, and P. R. Kamath, "A secure color image steganography in transform domain," *International Journal on Cryptography and Information Security*, vol. 3, no. 1, 2013.

[120] M. Onken, M. Eichelberg, J. Riesmeier, and P. Jensch, “Digital imaging and communications in medicine,” in *Biomedical Image Processing*, pp. 427–454, Springer, 2011.

[121] S. Das and M. K. Kundu, “Effective management of medical information through roi-lossless fragile image watermarking technique,” *Computer methods and programs in biomedicine*, vol. 111, no. 3, pp. 662–675, 2013.

[122] O. S. Pianykh, *Digital imaging and communications in medicine (DICOM).* Springer, 2012.

[123] F. Cao, H. Huang, and X. Zhou, “Medical image security in a hipaa mandated pacs environment,” *Computerized Medical Imaging and Graphics*, vol. 27, no. 2, pp. 185–196, 2003.

[124] O. S. Pianykh, *Digital imaging and communications in medicine (DICOM): a practical introduction and survival guide.* Springer Science & Business Media, 2009. [Online; accessed 18-December-2014].

[125] B. M. Irany, *A High Capacity Reversible Multiple Watermarking Scheme-Applications to Images, Medical Data, and Biometrics.* PhD thesis, University of Toronto, 2011.

[126] G. Coatrieux, L. Lecornu, B. Sankur, and C. Roux, “A review of image watermarking applications in healthcare,” in *Engineering in Medicine and Biology Society, 2006. EMBS’06. 28th Annual International Conference of the IEEE*, pp. 4691–4694, IEEE, 2006.

[127] C. I. Podilchuk and E. J. Delp, “Digital watermarking: algorithms and applications,” *Signal Processing Magazine, IEEE*, vol. 18, no. 4, pp. 33–46, 2001.

[128] G. Coatrieux, H. Maitre, B. Sankur, Y. Rolland, and R. Collorec, “Relevance of watermarking in medical imaging,” in *Information Technology Applications in Biomedicine, 2000. Proceedings. 2000 IEEE EMBS International Conference on*, pp. 250–255, IEEE, 2000.

[129] H. Baur, U. Engelmann, F. Saurbier, A. Schröter, U. Baur, and H. Meinzer, “How to deal with security issues in teleradiology,” *Computer Methods and Programs in Biomedicine*, vol. 53, no. 1, pp. 1–8, 1997.

[130] M. A. Epstein, M. S. Pasieka, W. P. Lord, S. T. Wong, and N. J. Mankovich, “Security for the digital information age of medicine: Issues, applications, and implementation,” *Journal of digital imaging*, vol. 11, no. 1, pp. 33–44, 1998.

[131] K. Navas, S. A. Thampy, and M. Sasikumar, “Epr hiding in medical images for telemedicine,” *International Journal of Biomedical Sciences*, vol. 3, no. 1, 2008.

[132] S.-C. Cheung, D. K. Chiu, and C. Ho, “The use of digital watermarking for intelligence multimedia document distribution,” *Journal of theoretical and applied electronic commerce research*, vol. 3, no. 3, pp. 103–118, 2008.

[133] R. A. Sampaio and M. P. Jackowski, "Assessment of steganographic methods in medical imaging," 1985.

[134] J. E. Canavan, "Fundamentals of network security, artech house," *Inc., Norwood, MA*, 2001.

[135] G. Coatrieux, H. Maitre, and B. Sankur, "Strict integrity control of biomedical images," in *Photonics West 2001-Electronic Imaging*, pp. 229–240, International Society for Optics and Photonics, 2001.

[136] S. Goldwasser and M. Bellare, "Lecture notes on cryptography," *Summer course "Cryptography and computer security" at MIT*, vol. 1999, p. 1999, 1996.

[137] K. Stine and Q. Dang, "Encryption basics," *Journal of AHIMA*, vol. 82, no. 5, p. 44, 2011.

[138] B. Preneel, "Cryptographic hash functions," *European Transactions on Telecommunications*, vol. 5, no. 4, pp. 431–448, 1994.

[139] C. Paar and J. Pelzl, *Understanding cryptography: a textbook for students and practitioners.* Springer Science & Business Media, 2009.

[140] A. Wakatani, "Digital watermarking for roi medical images by using compressed signature image," in *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pp. 2043–2048, IEEE, 2002.

[141] O. M. Al-Qershi and B. E. Khoo, "Authentication and data hiding using a hybrid roi-based watermarking scheme for dicom images," *Journal of digital imaging*, vol. 24, no. 1, pp. 114–125, 2011.

[142] K. Navas, S. Nithya, R. Rakhi, and M. Sasikumar, "Lossless watermarking in jpeg2000 for epr data hiding," *Proc. IEEE-EIT 2007*, pp. 697–702, 2007.

[143] X. Zhou, H. Huang, and S.-L. Lou, "Authenticity and integrity of digital mammography images," *Medical Imaging, IEEE Transactions on*, vol. 20, no. 8, pp. 784–791, 2001.

[144] H.-M. Chao, C.-M. Hsu, and S.-G. Miaou, "A data-hiding technique with authentication, integration, and confidentiality for electronic patient records," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 6, no. 1, pp. 46–53, 2002.

[145] M. A. Hajjaji, A. Mtibaa, E.-B. Bourennane, *et al.*, "A watermarking of medical image: Method based'lsb'," *International Journal of Computer Science Issues*, 2011.

[146] C. Nagaraju and S. ParthaSarathy, "Embedding ecg and patient information in medical image," in *Recent Advances and Innovations in Engineering (ICRAIE), 2014*, pp. 1–6, IEEE, 2014.

[147] F. Rahimi, H. Rabbani, *et al.*, "A dual adaptive watermarking scheme in contourlet domain for dicom images," *Biomedical engineering online*, vol. 10, no. 1, p. 53, 2011.

[148] D.-C. Lou, M.-C. Hu, and J.-L. Liu, "Multiple layer data hiding scheme for medical images," *Computer Standards & Interfaces*, vol. 31, no. 2, pp. 329–335, 2009.

[149] N. A. Memon, A. Chaudhry, M. Ahmad, and Z. A. Keerio, "Hybrid watermarking of medical images for roi authentication and recovery," *International Journal of Computer Mathematics*, vol. 88, no. 10, pp. 2057–2071, 2011.

[150] N. A. Memon and S. A. M. Gilani, "Watermarking of chest ct scan medical images for content authentication," *International Journal of Computer Mathematics*, vol. 88, no. 2, pp. 265–280, 2011.

[151] O. M. Al-Qershi and B. E. Khoo, "High capacity data hiding schemes for medical images based on difference expansion," *Journal of Systems and Software*, vol. 84, no. 1, pp. 105–112, 2011.

[152] H. Nyeem, W. Boles, and C. Boyd, "A review of medical image watermarking requirements for teleradiology," *Journal of digital imaging*, vol. 26, no. 2, pp. 326–343, 2013.

[153] S. M. Mousavi, A. Naghsh, and S. Abu-Bakar, "Watermarking techniques used in medical images: a survey," *Journal of digital imaging*, vol. 27, no. 6, pp. 714–729, 2014.

[154] S. Riaz and S.-W. Lee, "A robust multimedia authentication and restoration scheme in digital photography," *Multimedia Tools and Applications*, pp. 1–31, 2013.

[155] R.-C. Raul, F.-U. Claudia, and G. d. J. Trinidad-BIas, "Data hiding scheme for medical images," in *Electronics, Communications and Computers, 2007. CONIELECOMP'07. 17th International Conference on*, pp. 32–32, IEEE, 2007.

[156] I. Bremnavas, B. Poorna, and G. Kanagachidambaresan, "Medical image security using lsb and chaotic logistic map," in *Advances in Recent Technologies in Communication and Computing (ARTCom 2011), 3rd International Conference on*, pp. 229–231, IET, 2011.

[157] G. Prabakaran, R. Bhavani, and P. Rajeswari, "Multi secure and robustness for medical image based steganography scheme," in *Circuits, Power and Computing Technologies (ICCPCT), 2013 International Conference on*, pp. 1188–1193, IEEE, 2013.

[158] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 13, no. 8, pp. 890–896, 2003.

[159] Y.-J. Zhang, "An overview of image and video segmentation in the last 40 years," *Advances in Image and Video Segmentation*, pp. 1–15, 2006.

[160] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.

[161] I. Despotović, B. Goossens, and W. Philips, "Mri segmentation of the human brain: challenges, methods, and applications," *Computational and mathematical methods in medicine*, vol. 2015, 2015.

[162] R. Dass and S. Devi, "Image segmentation techniques 1," *INTERNATIONAL JOURNAL OF ELECTRONICS AND COMMUNICATION TECHNOLOGY (IJECT)*, vol. 3, pp. 66–70, 2012.

[163] M. Havaei, H. Larochelle, P. Poulin, and P.-M. Jodoin, "Within-brain classification for brain tumor segmentation," *International journal of computer assisted radiology and surgery*, vol. 11, no. 5, pp. 777–788, 2016.

[164] N. Gordillo, E. Montseny, and P. Sobrevilla, "State of the art survey on mri brain tumor segmentation," *Magnetic resonance imaging*, vol. 31, no. 8, pp. 1426–1438, 2013.

[165] B. Foster, U. Bagci, A. Mansoor, Z. Xu, and D. J. Mollura, "A review on segmentation of positron emission tomography images," *Computers in biology and medicine*, vol. 50, pp. 76–96, 2014.

[166] K.-P. Wong, "Medical image segmentation: methods and applications in functional imaging," in *Handbook of biomedical image analysis*, pp. 111–182, Springer, 2005.

[167] M. Prastawa, E. Bullitt, N. Moon, K. Van Leemput, and G. Gerig, "Automatic brain tumor segmentation by subject specific modification of atlas priors 1," *Academic radiology*, vol. 10, no. 12, pp. 1341–1348, 2003.

[168] S. Belhassen and H. Zaidi, "A novel fuzzy c-means algorithm for unsupervised heterogeneous tumor quantification in pet," *Medical physics*, vol. 37, no. 3, pp. 1309–1324, 2010.

[169] G. P. Mazzara, R. P. Velthuizen, J. L. Pearlman, H. M. Greenberg, and H. Wagner, "Brain tumor target volume determination for radiation treatment planning through automated mri segmentation," *International Journal of Radiation Oncology* Biology* Physics*, vol. 59, no. 1, pp. 300–312, 2004.

[170] P. A. Yushkevich, Y. Gao, and G. Gerig, "Itk-snap: An interactive tool for semi-automatic segmentation of multi-modality biomedical images," in *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pp. 3342–3345, IEEE, 2016.

[171] S. D. Olabarriaga and A. W. Smeulders, "Interaction in the segmentation of medical images: A survey," *Medical image analysis*, vol. 5, no. 2, pp. 127–142, 2001.

[172] J. L. Foo, "A survey of user interaction and automation in medical image segmentation methods," *Iowa State University, Human Computer Interaction Technical Report ISU-HCI-2006-02*, 2006.

[173] G. K. Seerha and R. Kaur, “Review on recent image segmentation techniques,” *International Journal on Computer Science and Engineering*, vol. 5, no. 2, p. 109, 2013.

[174] K. Rahini and S. Sudha, “Review of image segmentation techniques: A survey,” *International Journal*, vol. 4, no. 7, 2014.

[175] A. Bali and S. N. Singh, “A review on the strategies and techniques of image segmentation,” in *Advanced Computing & Communication Technologies (ACCT), 2015 Fifth International Conference on*, pp. 113–120, IEEE, 2015.

[176] A. Kaur, “A review paper on image segmentation and its various techniques in image processing,” *International Journal of Science and Research*, 2012.

[177] M. J. Kumar, D. G. R. Kumar, and R. V. K. Reddy, “Review on image segmentation techniques,” *International Journal of Scientific Research Engineering & Technology (IJSRET), ISSN*, pp. 2278–0882, 2014.

[178] A. Chaudhary and T. Gulati, “Segmenting digital images using edge detection,” *International Journal of Application of Innovation in Engineering & Management*, vol. 2, no. 5, 2013.

[179] D. J. Withey and Z. J. Koles, “A review of medical image segmentation: methods and available software,” *International Journal of Bioelectromagnetism*, vol. 10, no. 3, pp. 125–148, 2008.

[180] H. Mohsen, E.-S. A. El-Dahshan, E.-S. El-Horbaty, and A.-B. M. Salem, “Atiner’s conference paper series com2016-1994,”

[181] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” *Journal of Machine Learning Research*, vol. 10, no. Feb, pp. 207–244, 2009.

[182] N. M. Portela, G. D. Cavalcanti, and T. I. Ren, “Semi-supervised clustering for mr brain image segmentation,” *Expert Systems with Applications*, vol. 41, no. 4, pp. 1492–1497, 2014.

[183] G. B. Coleman and H. C. Andrews, “Image segmentation by clustering,” *Proceedings of the IEEE*, vol. 67, no. 5, pp. 773–785, 1979.

[184] J. C. Bezdek, R. Ehrlich, and W. Full, “Fcm: The fuzzy c-means clustering algorithm,” *Computers & Geosciences*, vol. 10, no. 2, pp. 191–203, 1984.

[185] T. Kohonen, “Self-organizing maps, vol. 30 of springer series in information sciences,” *ed: Springer-Verlag Berlin*, 2001.

[186] D. L. Pham, C. Xu, and J. L. Prince, “Current methods in medical image segmentation,” *Annual review of biomedical engineering*, vol. 2, no. 1, pp. 315–337, 2000.

[187] T. M. Tuan *et al.*, “A cooperative semi-supervised fuzzy clustering framework for dental x-ray image segmentation,” *Expert Systems with Applications*, vol. 46, pp. 380–393, 2016.

[188] G. Vishnuvarthanan, “Tumor detection and tissue segmentation in magnetic resonance brain images using fuzzy and optimization techniques,” 2015.

[189] G. Dougherty, “Image analysis in medical imaging: recent advances in selected examples,” *Biomedical imaging and intervention journal*, vol. 6, no. 3, 2010.

[190] R. Hiralal and H. P. Menon, “A survey of brain mri image segmentation methods and the issues involved,” in *The International Symposium on Intelligent Systems Technologies and Applications*, pp. 245–259, Springer, 2016.

[191] H. Kekre and S. Gharge, “Segmentation of mri images using probability and entropy as statistical parameters for texture analysis,” *Advances in Computational sciences and Technology (ACST)*, vol. 2, no. 2, pp. 219–230, 2009.

[192] A. W.-C. Liew and H. Yan, “Current methods in the automatic tissue segmentation of 3d magnetic resonance brain images,” *Current medical imaging reviews*, vol. 2, no. 1, pp. 91–103, 2006.

[193] Z. Ji, Y. Xia, Q. Sun, Q. Chen, D. Xia, and D. D. Feng, “Fuzzy local gaussian mixture model for brain mr image segmentation,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 3, pp. 339–347, 2012.

[194] A. Ortiz, J. Górriz, J. Ramírez, D. Salas-Gonzalez, and J. M. Llamas-Elvira, “Two fully-unsupervised methods for mr brain image segmentation using som-based strategies,” *Applied Soft Computing*, vol. 13, no. 5, pp. 2668–2682, 2013.

[195] N. Noreen, K. Hayat, and S. A. Madani, “Mri segmentation through wavelets and fuzzy c-means,” *World Applied Sciences Journal*, vol. 13, pp. 34–39, 2011.

[196] M. Roushdy, “Comparative study of edge detection algorithms applying on the grayscale noisy image using morphological filter,” *GVIP journal*, vol. 6, no. 4, pp. 17–23, 2006.

[197] H. Permuter, J. Francos, and I. Jermyn, “A study of gaussian mixture models of color and texture features for image classification and segmentation,” *Pattern Recognition*, vol. 39, no. 4, pp. 695–706, 2006.

[198] L. Gupta and T. Sortrakul, “A gaussian-mixture-based image segmentation algorithm,” *Pattern Recognition*, vol. 31, no. 3, pp. 315–325, 1998.

[199] W. M. Wells, W. E. L. Grimson, R. Kikinis, and F. A. Jolesz, “Adaptive segmentation of mri data,” *IEEE transactions on medical imaging*, vol. 15, no. 4, pp. 429–442, 1996.

[200] H. Greenspan, A. Ruf, and J. Goldberger, “Constrained gaussian mixture model framework for automatic segmentation of mr brain images,” *IEEE transactions on medical imaging*, vol. 25, no. 9, pp. 1233–1245, 2006.

[201] K. Blekas, A. Likas, N. P. Galatsanos, and I. E. Lagaris, “A spatially constrained mixture model for image segmentation,” *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 494–498, 2005.

[202] R. Katyal, S. Paneri, and M. Kuse, "Gaussian intensity model with neighborhood cues for fluid-tissue categorization of multisequence mr brain images," *Proceedings of the MICCAI Grand Challenge on MR Brain Image Segmentation (MRBrainS'13)*, 2013.

[203] P. T. Selvy, V. Palanisamy, and T. Purusothaman, "Performance analysis of clustering algorithms in brain tumor detection of mr images," *European Journal of Scientific Research*, vol. 62, no. 3, pp. 321–330, 2011.

[204] A. Ortiz, A. A. Palacio, J. M. Górriz, J. Ramírez, and D. Salas-González, "Segmentation of brain mri using som-fcm-based method and 3d statistical descriptors," *Computational and mathematical methods in medicine*, vol. 2013, 2013.

[205] N. Goncalves, J. Nikkilä, and R. Vigario, "Self-supervised mri tissue segmentation by discriminative clustering," *International journal of neural systems*, vol. 24, no. 01, p. 1450004, 2014.

[206] E.-S. A. El-Dahshan, H. M. Mohsen, K. Revett, and A.-B. M. Salem, "Computer-aided diagnosis of human brain tumor through mri: A survey and a new algorithm," *Expert systems with Applications*, vol. 41, no. 11, pp. 5526–5545, 2014.

[207] Y. Kong, Y. Deng, and Q. Dai, "Discriminative clustering and feature selection for brain mri segmentation," *IEEE Signal Processing Letters*, vol. 22, no. 5, pp. 573–577, 2015.

[208] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in mri images," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1240–1251, 2016.

[209] I. M. I. Association *et al.*, "IMIA code of ethics for health information professionals," *Retrieved April*, vol. 14, p. 2004, 2002.

[210] P. Elias, "Coding for two noisy channels," in *Information Theory, Third London Symposium*, vol. 67, London, England, 1955.

[211] A. Quazi and A. Gulve, "A review on uniform embedding for efficient jpeg steganography," *International Journal of Computer Applications*, vol. 114, no. 12, 2015.

[212] N. Maleki, M. Jalali, and M. V. Jahan, "Adaptive and non-adaptive data hiding methods for grayscale images based on modulus function," *Egyptian Informatics Journal*, vol. 15, no. 2, pp. 115–127, 2014.

[213] J. Fridrich, *Steganography in digital media: principles, algorithms, and applications.* Cambridge University Press, 2009.

[214] N. Dhanachandra, K. Manglem, and Y. J. Chanu, "Image segmentation using k-means clustering algorithm and subtractive clustering algorithm," *Procedia Computer Science*, vol. 54, pp. 764–771, 2015.

[215] H. Li, H. He, and Y. Wen, "Dynamic particle swarm optimization and k-means clustering algorithm for image segmentation," *Optik-International Journal for Light and Electron Optics*, vol. 126, no. 24, pp. 4817–4822, 2015.

[216] M. Xess and S. A. Agnes, "Survey on clustering based color image segmentation and novel approaches to fcm algorithm," *International Journal Of Research In Engineering And Technology*, vol. 2013, 2013.

[217] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[218] D. R. Choudhury, P. Bhargava, P. Reena, and S. Kain, "Use of artificial neural networks for predicting the outcome of cricket tournaments," *International Journal of Sports Science and Engineering*, vol. 1, no. 2, pp. 87–96, 2007.

[219] G. Sagar, S. V. Chalam, and M. K. Singh, "Evolutionary algorithm for optimal connection weights in artificial neural networks," *International Journal of Engineering*, vol. 5, no. 5, pp. 333–340, 2011.

[220] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[221] P. Kovesi, "Image segmentation using slic superpixels and dbscan clustering," *University of Western Australia, Center for Exploration Targeting, Image Analysis Group*, 2013. Available at http://www.peterkovesi.com/projects/segmentation/index.html.

[222] R. Lippmann, "An introduction to computing with neural nets," *IEEE Assp magazine*, vol. 4, no. 2, pp. 4–22, 1987.

[223] P. Bas and T. Furon, "Bows-2." http://bows2.ec-lille.fr/, 2007.

[224] E. Satir and H. Isik, "A huffman compression based text steganography method," *Multimedia tools and applications*, vol. 70, no. 3, pp. 2085–2110, 2014.

[225] C.-S. Lu, *Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property.* Igi Global, 2004.

[226] B. Li, J. Huang, and Y. Q. Shi, "Textural features based universal steganalysis," in *Electronic Imaging 2008*, pp. 681912–681912, International Society for Optics and Photonics, 2008.

[227] Y. Wang and P. Moulin, "Optimized feature extraction for learning-based image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, pp. 31–45, 2007.

[228] X. Hou, T. Zhang, G. Xiong, Z. Lu, and K. Xie, "A novel steganalysis framework of heterogeneous images based on gmm clustering," *Signal Processing: Image Communication*, vol. 29, no. 3, pp. 385–399, 2014.

[229] I. Ng, T. Tan, and J. Kittler, "On local linear transform and gabor filter representation of texture," in *Pattern Recognition, 1992. Vol. III. Conference C: Image, Speech and Signal Analysis, Proceedings., 11th IAPR International Conference on*, pp. 627–631, IEEE, 1992.

[230] C. Cachin, "An information-theoretic model for steganography," *Information and Computation*, vol. 192, no. 1, pp. 41–56, 2004.

[231] J. Mandal and D. Das, "Steganography using adaptive pixel value differencing (APVD) of gray images through exclusion of overflow/underflow," *Second International Conference on Computer Science, Engineering and Applications (CCSEA-2012)*, pp. 93–102, 2012.

[232] Y.-P. Lee, J.-C. Lee, W.-K. Chen, K.-C. Chang, I.-J. Su, and C.-P. Chang, "High-payload image hiding with quality recovery using tri-way pixel-value differencing," *Information Sciences*, vol. 191, pp. 214–225, 2012.

[233] P. Thiyagarajan and G. Aghila, "Reversible dynamic secure steganography for medical image using graph coloring," *Health Policy and Technology*, vol. 2, no. 3, pp. 151–161, 2013.

[234] A. Worth, "The internet brain segmentation repository (IBSR)." http://www.cma.mgh.harvard.edu/ibsr/, 2009.

[235] T. Rohlfing, "Image similarity and tissue overlaps as surrogates for image registration accuracy: widely used but unreliable," *IEEE transactions on medical imaging*, vol. 31, no. 2, pp. 153–163, 2012.

[236] S. Valverde, A. Oliver, M. Cabezas, E. Roura, and X. Lladó, "Comparison of 10 brain tissue segmentation methods using revisited ibsr annotations," *Journal of Magnetic Resonance Imaging*, vol. 41, no. 1, pp. 93–101, 2015.

[237] U. Vovk, F. Pernuš, and B. Likar, "A review of methods for correction of intensity inhomogeneity in mri," *Medical Imaging, IEEE Transactions on*, vol. 26, no. 3, pp. 405–421, 2007.