Sudan University of Science and Technology

College of Graduate Studies

A Novel Image Steganography Method Using Fuzzy Logic and Edge Detection

طريقة جديدة لإخفاء الصور بإستخدام المنطق الضبابي وإكتشاف الحواف

Dissertation

Submitted in Partial Fulfilment of the requirements

for the degree of Doctor of Philosophy inComputer Science and Information Technology

By:

Hala Salih Yusuf Salih

Supervisor:

Professor. Hani Hagras

March-2022

الآيـــــــــــــــة

قال تعالي:﴿وَقُلْ رَبِّ زِدْنِي عِلْمًا﴾ سورة طه114

قال تعالي:﴿ يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ﴾ سورة المجادلة11

ABSTRACT

Steganography is one of the information hiding techniques that hides a message inside another message without drawing any suspicion. For hiding messages, various types of media are used. Image steganography is the technique that uses an image file to conceal information. In recent years, different methods have been proposed, which combined steganography and edge detection, have been proposed.

This thesis presents a novel Image Steganography method using least significant bit (LSB) and fuzzy logic. Firstly, gradient type-1 fuzzy logic (T1FLS) edge detector has been proposed to make disclosing the existence of a secret message a hard operation. The proposed system processes the image in two phases (named fuzzy phase and embed phase). In the fuzzy phase based on the gradient approach and T1FLS, the edge detector is calculated. In the embedding phase, exploiting the edge image that has been obtained from the previous phase in embedding more secret bits in edge pixels than in non-edge pixels. The proposed system is developed on two sides, the sender's side which deals with the embedding process, and the receiver's side which deals with extraction processes.

Secondly, the gradient T1FLS edge detector has been improved by using gradient type-2 fuzzy logic(T2FLS) edge detector due to their ability to handle the high level of uncertainty present in images. The enhanced system processes the image in two phases (called the fuzzy phase and embed phase). The enhanced gradient T2FLS edge detector has the same steps as the gradient T1FLS edge detector, except for the use of T2FLS instead of T1FLS in the fuzzy phase.

Many experiments were conducted to measure the performance of the proposed methods. For the proposed gradient T1FLS edge detector, the experimental results demonstrate the performance of the proposed T1FLS on six 128×192 RGB color images from the BSD300 dataset.

For the enhanced gradient T2FLS edge detector, three experiments were conducted on different image datasets based on the image size. In the first experiment, the proposed T2FLS will apply to six 128×192 RGB color images from the BSD300 image. In the second experiment, the proposed T2FLS will be implemented on six 256×256 RGB color images from the USC-SIPI image dataset. And in the third experiment, the proposed T2FLS will be implemented on six 512×512 RGB color images from the CSIQ image dataset. The PSNR and HVS have been used to measure the quality of the stego image in each experiment.

When the results of the proposed method were compared with previous studies, the results showed that the proposed system provides better stego image quality, as well as higher embedding capacity than previous works. Metrics like peak signal to noise ratio (PSNR) and the human visual system (HVS), have been used to measure the quality of the stego image.

المستخلص

يُعد علم الأخفاء أحد تقنيات إخفاء المعلومات التي تخفي رسالة داخل رسالة أخرى دون إثارة أي شك. يمكن إستخدام أنواع مختلفة من الوسائط لإخفاء الرسائل. إخفاء الصور هو الأسلوب الذي يستخدم ملف صورة لإخفاء المعلومات. في السنوات الأخيرة ، تم إقتراح طرق مختلفة، والتي جمعت بين علم إخفاء المعلومات وإكتشاف الحواف.

تُقدم هذه الدراسة طريقة جديدة للإخفاء في الصور بإستخدام البت الأقل أهمية (LSB) Least Significant Bit والمنطق الضبابي. أولاً ، تم إقتراح كاشف حافة المنطق الضبابي من النوع 1 المتدرج (T1FLS) لجعل الكشف عن وجود رسالة سرية عملية صعبة. يعالج النظام المقترح الصورة على مرحلتين (هما المرحلة الضبابية ومرحلة التضمين). في المرحلة الضبابية وإعتماداً علي منهج التدرج وT1FLS ، يتم حساب كاشف الحافة. في مرحلة التضمين ، ثم استغلال صورة الحواف التي تم الحصول عليها من المرحلة السابقة في تضمين المزيد من البتات السرية في بكسلات الحافة مقارنةً بالبكسل غير الحواف. تم تطوير النظام المقترح من جانبين ، جانب المرسل الذي يتعامل مع عملية التضمين ، وجانب المستلم الذي يتعامل مع عمليات الاستخراج (الاسترجاع).

ثانيًا ، تم تحسين كاشف حافة التدرج T1FLSبإستخدام كاشف حافة المنطق الضبابي من النوع الثاني المتدرج(T2FLS)، نظرًا لقدرته على التعامل مع المستوى العالي من عدم اليقين الموجود في الصور. يُعالج النظام المحسن الصورة على مرحلتين (هما المرحلة الضبابية ومرحلة التضمين). كاشف حافة التدرج المحسن T2FLS له نفس خطوات كاشف حافة التدرج T1FLS ، بإستثناء إستخدام T2FLS بدلاً من T1FLS في المرحلة الضبابية.

أُجريت العديد من التجارب لقياس أداء الطرق المقترحة. بالنسبة إلى كاشف حافة التدرج T1FLS المقترح ، توضح النتائج التجريبية أداء T1FLS المقترح على ست صور ملونة بحجم 128 × 192 RGB من مجموعة البيانات BSD300.

بالنسبة إلى كاشف حافة التدرج المحسن T2FLS ، تم إجراء ثلاث تجارب على مجموعات بيانات صور مختلفة إعتمادًا على حجم الصورة. في التجربة الأولى ، تم تطبيق T2FLS المقترح على ست صور ملونة بحجم 128 × 192 RGB من مجموعة بيانات صورBSD300. في التجربة الثانية ، تم تنفيذ T2FLS المقترح على ست صور ملونة بحجم 256 × 256 RGB من مجموعة بيانات صور USC-

SIPI. وفي التجربة الثالثة ، تم تنفيذ T2FLS المقترح على ست صور ملونة بحجم 512 × 512 RGB من مجموعة بيانات صورCSIQ.

عندما تم مقارنة نتائج الطريقة المقترحة بالدراسات السابقة ، أظهرت النتائج أن النظام المقترح يوفر جودة صورة أفضل ، بالإضافة إلى قدرة تضمين أعلى من الأعمال السابقة. تم استخدام PSNR و HVS لقياس جودة صورة stego في كل تجربة.

ACKNOWLEDGEMENTS

Firstly I want to thanks Allah who give us Knowledge and success, and I would like to thanks several persons who helped and encouraged me during accomplish the research.

My great thanks to my supervisor Prof. Hani Hagras the supervisor of the research for his valuable time and guidance. I also thank all my friends and colleagues who have more or less contributed to achieving this research. I also extend my thanks to my family for their support.

This research indeed helped me to extend my knowledge in this field and in other fields, and I am sure this experience will help me in my life.

DECLARATION

I hereby declare that this thesis is the result of my own investigation, except where otherwise stated. I also declare that it has not been previously or concurrently submitted as a whole for any other degrees at Sudan University of Science and Technology or other institutions.

Hala Salih Yusuf Salih

Signature________________ Date___________________

TABLE OF CONTENTS

LIST OF TABLES

Table No. **Page No.**

LIST OF FIGURES

Figure No. **Page No.**

LIST OF ABBREVIATIONS

BPP	Bits Per Pixel
FL	Fuzzy Logic
FLS	Fuzzy Logic System
FOU	Footprint Of Uncertainty
GA	Genetic Algorithm
HVS	Human Visual System
LMF	Lower Membership Function
LS	Learning System
LSB	Least Significant Bit
MF	Membership Functions
MSB	Most-Significant-Bit
MSE	Minimum Mean Square Error
PSNR	Peak Signal-To-Noise Ratio
PVD	Pixel-Value Differencing
RGB	Red Green Blue
T1FIS	Type-1 Fuzzy Inference System
T2FIS	Type-2 Fuzzy Inference System
UMF	Upper Membership Function

LIST OF PUBLICATIONS

1. H. S. Yusuf, H. Hagras, “Towards Image Steganography Using Type-2 Fuzzy Logic and Edge Detection,” *Proceedings of the 2018 International Conference on Computer Science and Electronic Engineering, Colchester , UK* ,pp. 75–78, September 2018.
2. H. Yusuf, H. Hagras, "High Payload Image Steganography Method Using Fuzzy Logic and Edge Detection", ***International Journal of Computer Science Trends and Technology***, Vol.8, No.5, pp. 123-133, August 2020

3. H. Yusuf, H. Hagras, "A novel Payload Image Steganography Using Type-2 Fuzzy Logic and Edge Detection", ***International Journal of Computer Science Trends and Technology***, vol. 9, no. 2, pp. 89–98, 2021

SECTION ONE INTRODUCTION

CHAPTER ONE: INTRODUCTION

CHAPTER ONE INTRODUCTION

1.1 Introduction

The purpose of computer security includes protecting information and property from theft, corruption, or natural disaster while allowing information and assets to remain accessible and productive to its intended users. By using only classic cryptography, encrypted messages become scrambles data that cannot pass the checkpoint on the network. Steganography provides another way of protecting the secret message through embedding it in other media so that transmitted data is meaningful and not noticeable. Compared with cryptography techniques attempting to conceal the secret message, steganography conceals the existence of the secret message[1]. The Figure1.1 shows various disciplines of information hiding[2].

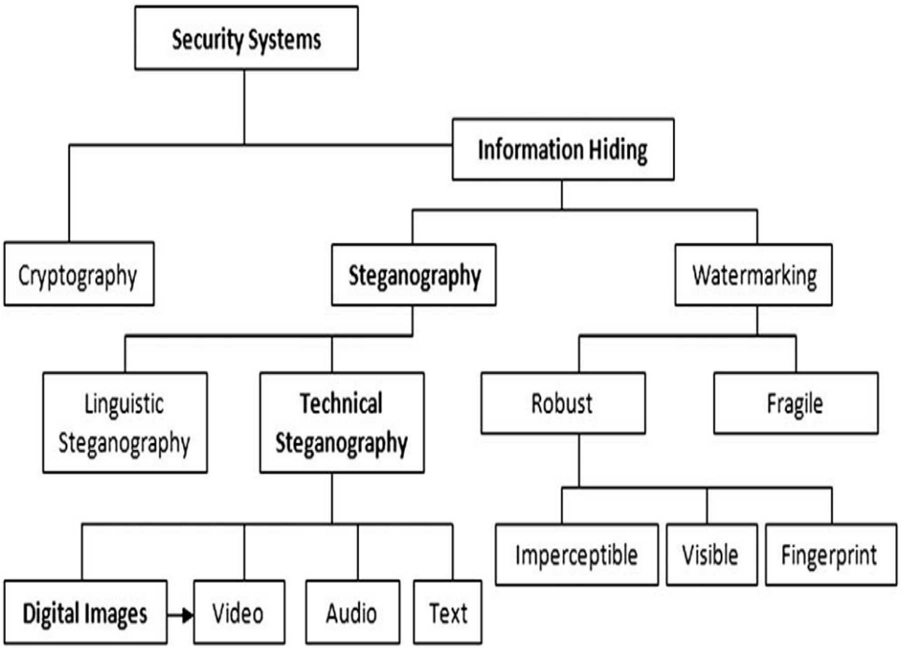


Figure 1.1: various disciplines of information hiding[2]

Steganography is an art and science that hides the existence of secret communications, where only both the sender and the receiver know of the presence of this hidden part[3][4]. It is derived from the Greek word stegano meaning “covering” and graphia meaning “writing or drawing”[5]. The main principle of steganography is the existence of a secret message is unknown, but in cryptography, the secret message is transformed into a different form [6]. Cryptography was created as an encryption technique to protect the secrecy of communication, and many various methods are designed to encrypt and decrypt data to keep the message secret[7][8]. Steganography and cryptography share the same goal, and both concepts are closely related[1]. Unfortunately, sometimes it is not enough to keep the content of the message secret; it may also be necessary to keep the presence of the message confidential. Steganography is one of these techniques for solving the problem. The concealment of secret messages is achieved by embedding them into other seemingly-innocent host mediums. Figure 1.2 shows the basic idea of any steganography process[9].

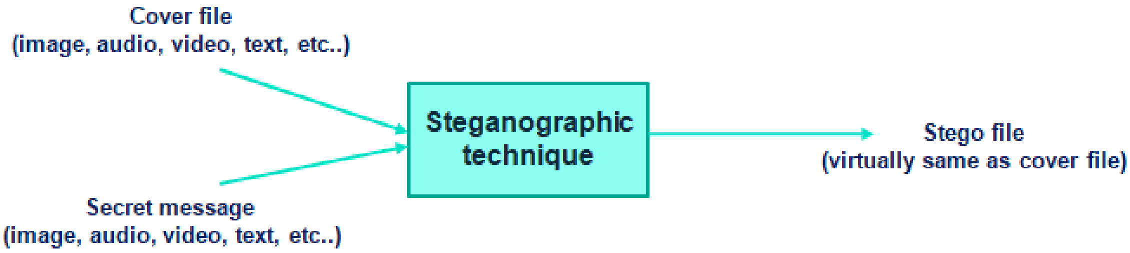

Cover file
(image, audio, video, text, etc..)
Steganographic technique
Stego file
(virtually same as cover file)
Secret message
(image, audio, video, text, etc..)

Figure 1.2: Fundamental scheme of steganography process[9].

Both cryptography and steganography are considered cousins in the family of spy arts. Cryptography scrambles the message so it cannot be understood, while steganography conceals it so that it cannot be seen[10].

To practice steganography, one has to enter data (cover object and message object). The cover object is the element which is used to hide the secret message. Steganography can apply to various cover objects like text, image, audio, or video, etc. A good choice of cover can make the steganography better and harder to detect. The cover object is also called the carrier object of the stenographic method. The message object is the secret message that needs protection. It can also be of types like text, image, audio, or video etc. After applying the steganography method, the output file is referred to as the stego object.

The information hidden in the cover file is known as embedded data. Stego files contain both the cover file and embedded information. Embedding means logically, entering hidden or embedded data into a cover file. In steganography, a cover file is sometimes referred to as a carrier file.

Depending on the nature of carrier object, steganography can be divided into five types(show in Figure 1.3) [11][12]:

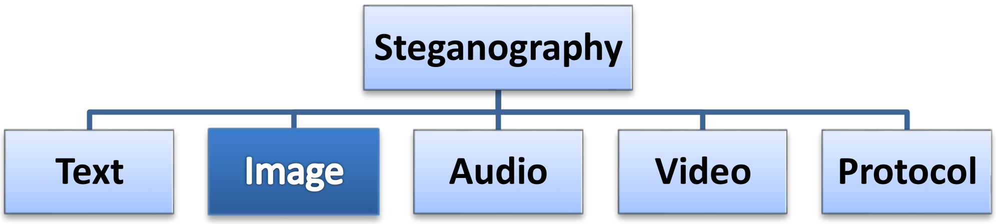


Figure 1.3: Steganography Types depends on the cover object[11][13].

1.2 Problem Statement and Its Significant

The main problems targeted by the thesis are:

1. Capacity of the covert media always small.
2. Robustness of stego object.
3. Imperceptibility in the process of embedding a secret message into image file should be made imperceptible to the human eyes.

Today, it is really necessary to develop steganography techniques to keep the transferred data out of the attention of malicious users. Figure 1.4 shows the inter-relationship between these three requirements and the way they constrain each other. This triangle representation summarizes the tradeoff between embedding capacity and the robustness under a certain degree of attacks while keeping imperceptibility (the quality of the stego image) at a relatively high level can be seen as an optimization problem. It is an interesting issue, how to balance these three requirements in the fields of information hiding[14].

From the previous studies, after taking both Tseng and Leng[15], J. Bai et al [14] and Chen et al [16] schemes into careful account, we observe that these two schemes used grayscale images that give them a little number of pixels to exploit in embedding the secret message, and the quality of their stego image also significantly lower. The proposed system will be designed to increase capacity, robustness, and imperceptibility by using type-2 fuzzy logic systems, edge detection, and LSB substitution to embed secret messages on RGB color images.

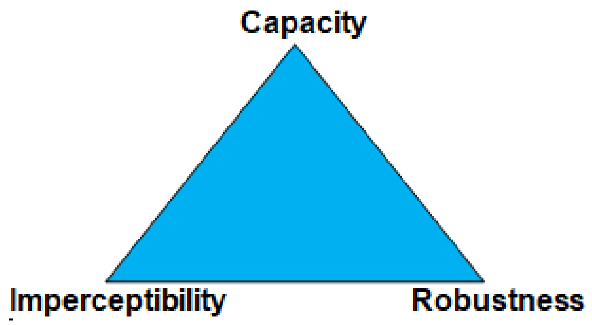


Figure 1.4: Magic triangle model of steganography[14].

1.3 Research Objectives

The main aim of this thesis is to develop an image steganography method to embed a secret message in an image file using type-2 fuzzy logic systems and edge detection to increment the capacity, robustness, and quality of the stego-object. To achieve this aim, the objectives have been listed as follows:

1. Contribute to give more complex and secure system.
2. To study the available methods used for image steganography.
3. To study Fuzzy Logic Systems and use them in edge detection.
4. To develop image steganography method using type-1 fuzzy logic systems and edge detection.
5. To enhance the image steganography method by using type-2 fuzzy logic systems and edge detection.

1.4 Research Methodology

From previous studies that have been done in Fuzzy logic, edge detection, and Image steganography, the strength and weakness points of each method are recognized. Based on this, a fuzzy logic type-2 image steganography method will be employed due to their ability to handle the high level of uncertainty present in images. The type-2 fuzzy systems will be employed in conjunction with LSB substitution and edge detection, to classify cover image pixels into edge pixels and non-edge pixels.

The performance of the proposed methodology will be tested by comparing the original image and the stego image by PSNR and HVS.

1.5 Thesis Contributions

New methods for Image Steganography to embed a secret message in image file using type-2 fuzzy logic systems and edge detection.

1.6 Research Organization

The organization of the following chapters in this research is as follows: Chapter 2 presents the literature on image steganography and edge detection. Chapter 3 presents the literature on the fuzzy logic domain and the basic concepts that are used within this study. Chapter 4 presents the related work. Chapter 5 presents the details of the proposed image steganography algorithm using gradient edge detector and Type-1 Fuzzy Logic System. Chapter 6 presents the details of the enhanced image steganography algorithm using gradient edge detector and Type-1 Fuzzy Logic System. Chapter 7 presents the conclusions of the research and the potential future work.

SECTION TWO THEORETICAL BACKGROUND

CHAPTER TWO: IMAGE STEGANOGRAPHY AND EDGE DETECTION

CHAPTER THREE: FUZZY LOGIC SYSTEMS

CHAPTER FOUR: LETRATURE REVIEW ON STEGANOGRAPHY

CHAPTER TWO IMAGE STEGANOGRAPHY AND EDGE DETECTION

2.1 Introduction

Due to the advancement of both the Internet and computer technologies in recent decades, information security has become an increasingly important factor of communication and information technology. Because of that, we must take measures to protect the secret information. In general, secret information can be protected in one of two ways, either by cryptography or steganography. In cryptography, the secret message is coded in a way that can't be understood, while in steganography, the secret message is hidden.[7][12][17][18][19][20][21].

In the digital world of today, steganography applies to a wide variety of data formats. .bmp, .png, .doc, .gif, .jpeg, .mp3, .txt, and .wav are the most popular data formats. The main reason for this is their popularity on the Internet and the ease with which they can be used by steganographic tools. Moreover, these formats are popular because redundant or noisy data can be removed from them and replaced with a hidden message relatively easily[8]. Steganographic technologies are an integral part of the future of Internet security and privacy on open networks like the Internet. In an open-systems environment, the main driver of steganographic research is the lack of strength in cryptographic systems on their own. Some governments have created laws that limit or prohibit cryptosystems entirely. This was done for fear that law enforcement wouldn't be able to gather intelligence through wiretaps, etc. Unfortunately, this leaves the majority of the Internet community either with relatively weak and often breakable encryption algorithms or no encryption algorithms at all. Advocates of civil liberties argue that "these limitations are an attack on privacy". Here is where Steganography comes into play[8][12][22].

2.2 History of Steganography

Steganography was described by the Greek historian Herodotus in his chronicles known as "Histories" around 440 BC. During this period in Greece, Herodotus recorded two stories about Steganographic techniques. The first report states that Darius the king of Susa shaved off the head of one of his captives and wrote a message on his head. As the prisoner's hair grew, he was sent to the law of King Aristogoras in Miletus unseen. The second story is from Herodotus, who claimed that Demeratus had sent a message to Sparta that Xerxes intended to invade Greece. At that time, writing was the practice of writing on waxed tablets. Demeratus removed the glue from the tablet, wrote a secret message on the underside of the tablet, returned the tablet with wax to make it look like an empty tablet, and finally sent the document without detected[7][8].

Romans used invisible inks, which were primarily based on natural materials like milk and fruit juices. This was achieved by heating the hidden message to return it again. Invisible inks have become much more developed and are still in limited use today. During the fifteenth and sixteenth centuries, many writers along with GaspariSchotti (writer of Steganographica) and Johannes Trithemius (writer of Steganographia) wrote about Steganagraphic strategies along with coding techniques for textual content, invisible inks, and incorporating hidden messages music[7][8].

Nowadays, in the digital world, Steganography is being used all around the international on PC systems. Many technologies and gear have been developed that take gain of old steganographic techniques which include null ciphers, hiding a message in an image, audio, video, or microdot[7][8].

2.3 Image Definition

A computer sees an image as a collection of numbers that represent various intensities of light in various areas of the image. This numeric representation forms a grid and the individual points are referred to as pixels. Most Internet images consist of a rectangular map of the image's pixels (represented as bits) that shows where each pixel is located and it'scolor. The pixels are arranged horizontally in rows and columns. The number of bits in a color scheme, called the bit depth, refers to the number of bits used for every pixel. In current color schemes, the smallest bit depth is 8, meaning that each pixel is described by 8 bits. Monochrome and grayscale images consist of 8-bit pixels and display 256 different colors or shades of grey. Digital color images use the RGB color model, also known as true color, and it is usually stored in 24-bit files. In a 24-bit image, all color variations are derived from three primary colors: red, green, and blue, and each primary color is represented by 8 bits[11].

2.4 Cryptography Vs Steganography

In general, both methods are used to ensure data transmission is secure. However, there are a few differences:

1. While cryptography involves hiding information from others, steganography involves hiding the very existence of information.
2. Cryptography hides data by changing it, while steganography, on the other hand, hides data by masking it (unless it is part of an encryption process).

3. Cryptography's main challenge to a hacker is to crack its algorithm while steganography removes suspicion that data exists.
4. The data encoding process is an important aspect of steganography, and if the system is familiar, it can be mastered[23][24][25][20].

2.5 Steganography Vs Watermarking

While steganography and watermarking share many characteristics, watermarking is primarily used to preserve the ownership rights of the creator. Any alteration of watermarked data then will damage this pattern. It is possible to combine steganography, watermarking, and cryptography to create a more secure system. By combining steganography and cryptography for secure media communication, we can add watermarking to increase security targeted at data hiding[26][27].

2.6 Categorization of Image Steganography

As shown in Figure 2.1, Image Steganographic techniques can be divided into two groups[6]: the Spatial domain technique group and the Transform domain group[28]. The Spatial domain technique embeds information in the intensity of the pixels directly, while the Transform domain technique embeds information in the frequency domain of previously transformed images [4][6][5][17][29].

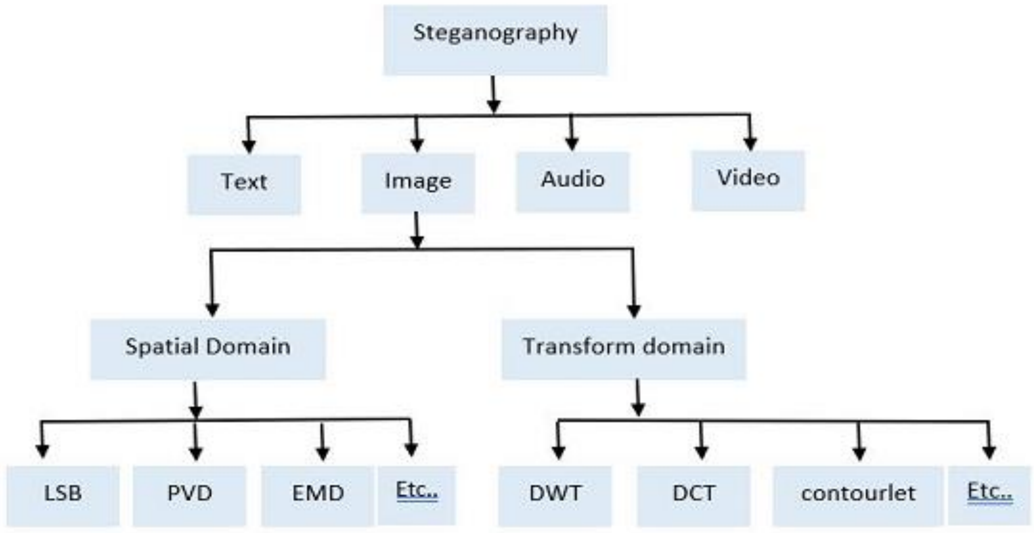


Figure 2.1: Categories of image steganography[30]**.**

2.7 Spatial Domain

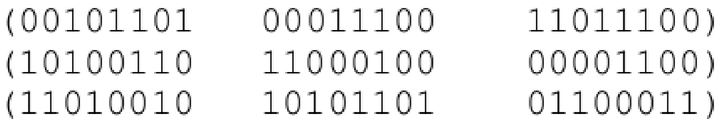
In the spatial domain, the information is hidden straight in the cover. One of the most common spatial techniques is LSB, where the hidden image's pixels are distributed inside the cover image bits by replacing the last bit in each pixel[26][28][29][30].

2.7.1 Spatial Domain Techniques Are

1. Least Significant Bit (LSB).
2. Pixel Value Differencing (PVD).
3. Edges Based data Embedding (EBE).
4. Random Pixel Embedding (RPE).
5. Pixel Mapping Method (PMM).

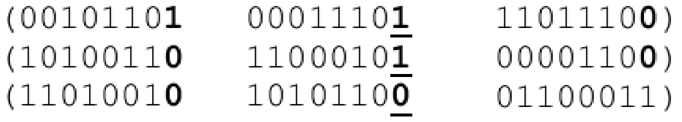
2.7.2 Least Significant Bit

The least-significant-bit (LSB)-based approach is a popular type of steganographic algorithms in the spatial domain[31],[32][23][29]. LSB insertion is a common, simple approach to embedding information in a cover image[19][33][27]. The least significant bit (in other words, the 8-bit) of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the red, green and blue color components can be used, since they are each represented by a byte. For example a grid for 3 pixels of 24-bit image can be as follows:[11][19]



```
(00101101   00011100    11011100)
(10100110   11000100    00001100)
(11010010   10101101    01100011)
```

When the number 200, which binary representation is 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as follows:



(0010110**1**   0001110**1**   1101110**0**)
(1010011**0**   1100010**1**   0000110**0**)
(1101001**0**   1010110**0**   01100011)

2.8 Transform Domain

During the 19th century, began what was named wavelet was developed, and wavelet transformations were important tools for image processing. The advantage of Wavelet Transformation (WT) is its ability to recognize small details in the signal. Wavelets can be used to identify major details in a signal while tiny wavelets can mask minor details[6][26][28][29][30].

2.8.1 Transform Domain Techniques Are

1. Discrete Cosine Transform (DCT).
2. Discrete Fourier Transform (DFT).
3. Discrete Wavelet Transform (DWT).
4. Integer Wavelet Transform (IWT).
5. Discrete Curvelet Transform (DCVT).

2.9 Image Types Utilized In Steganography

The formats used in steganography fall into two major categories[26][28][30]**:**

1. The raw image format: is referred to as a "lossless" format because it preserves the original image information after compression and decompression.
2. Lossy image formats: do not preserve the original information of the image.

2.9.1 Raw image types

1. Bitmap (Bmp): created by Microsoft in its laboratories. Color information for each pixel is stored in BMP formats without being compressed; for example, an 8*8 pixel Bitmap image contains 64 pixels of color information[26][28][30].
2. Tagged Image File Format (TIFF): Introduced in the late 19th century. A TIFF file can store any type of image ("Grayscale, 8-Bit, RGB") without any loss, and a TIFF file requires a high amount of storage[26][28][30].

3. Portable Network Graphic (PNG): First introduced as an improved alternative to Graphics Interchange Format (GIF). In a PNG file format, images are stored in 24-bit RGB, 8-bit grayscale, and RGBA palettes[26][28][30].

2.9.2 Lossy Image Types

Joint Photographic Expert Group (JPEG): Introduced to allow for the compression of RGB and grayscale images. JPEG has the benefit of being flexible regarding compression percentages[26][28][30].

2.10 Human Visual System (HVS)

Image steganography takes the advantage of the limited power of the human visual system (HVS) [34]. According to research, the human eye is more sensitive to changes in the brightness (luminance) of a pixel than to changes in its color (chrominance) [11].

2.11 Digital Image Processing

An image can be characterized as a two-dimensional function, where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is referred to as its intensity. We call an image a digital image when x, y, and the intensity values of f are all discrete, finite quantities. Digital image processing is the art of processing digital images using a computer.A digital image consists of a finite elements number, each of which has a particular value and location. These elements are called image elements, picture elements, and pixels. The term pixels are commonly used to denote the elements of a digital image[35][36].

There are fields such as computer vision whose basic goal is to use computers to mimic human vision, including learning and being able to infer information and take actions based on visual inputs. This area is a branch of artificial intelligence (AI) whose objective is to mimic human intelligence[36].

On the other hand, there are no obvious boundaries in the continuum from image processing at one end to computer vision at the other. However, one beneficial paradigm is to consider three types of computerized operations in this continuum: low-, mid-, and high-level operations. However, one beneficial paradigm is to consider three types of computerized operations in this continuum: low-, mid-, and high-level operations. Low-level operations include primitive operations such as image pre-processing to contrast enhancement, and image sharpening, and decreased noise. A low-level process is characterized by the truth that both its inputs and outputs are images.Mid-level processes are characterized by the fact that their inputs are generally images, but their outputs are attributes extracted from those images (e.g., edges, contours, and the identity of individual objects). As a final point, higher-level processes involve "making sense" of an ensemble of recognized objects, such as in image analysis, as well as performing cognitive functions that are normally associated with vision[36].

2.11.1 Common image formats include

1. 1 sample per point (Black& White or Grayscale)
2. 3 samples per point (Red, Green, and Blue)
3. 4 samples per point (Red, Green, Blue, and “Alpha”, a.k.a. Opacity)

2.12 Edge Detector

Edge detection is a process applied to digital image processing, particularly in the areas of feature extraction, to refer to algorithms and tools that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are usually systematic into a set of curved line segments termed edges[37][38][39][40]. There are many (classical) standard edge detection algorithms such as Sobel, Prewitt, Roberts, Laplacian and Canny operators [16][36][41][42].

2.12.1 Gradient Edge Detector

There are some methods to perform the edge detection process; most of them are based on image gradient magnitude, which are calculated with the first derivative of an image. In this thesis, the edge detection method is performed by calculating the image gradients with the Euclidean distance; which is the most used and the most important method. This operation includes calculating four image gradients to indicate the edge direction based on a 3×3 matrix (Di, for i =1 . . . 4) (Figure. 2.2 illustrates this). Each matrix position (Di), of Figure 2.2, is represented in Figure. 2.3, where f indicates the image, x-axis the rows and y-axis the columns[43][44].

According to these positions, the Euclidean distance is applied to calculate the gradients Di using the eq (2-1). Gradient magnitude, or The edges E, can be calculated with the eq(2-2)[43][44][45].

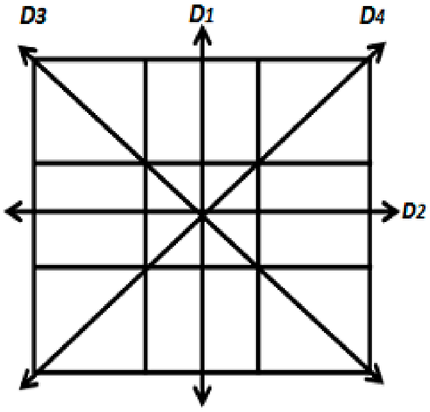
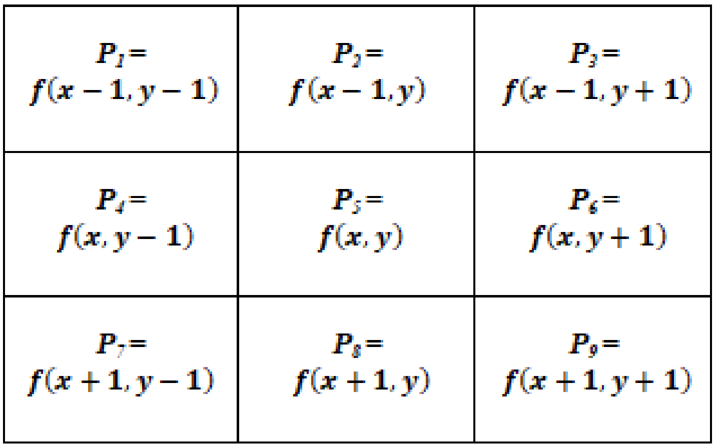


Figure 2.2: 3×3 Matrix indicating direction of the four gradients Di



$P_1 =$ $f(x-1, y-1)$	$P_2 =$ $f(x-1, y)$	$P_3 =$ $f(x-1, y+1)$
$P_4 =$ $f(x, y-1)$	$P_5 =$ $f(x, y)$	$P_6 =$ $f(x, y+1)$
$P_7 =$ $f(x+1, y-1)$	$P_8 =$ $f(x+1, y)$	$P_9 =$ $f(x+1, y+1)$

Figure 2.3: Matrix position

$$D_1 = \sqrt{(p_5 - p_2)^2 + (p_5 - p_8)^2} \tag{2-1}$$

$$D_2 = \sqrt{(p_5 - p_4)^2 + (p_5 - p_6)^2}$$

$$D_3 = \sqrt{(p_5 - p_1)^2 + (p_5 - p_9)^2}$$

$$D_4 = \sqrt{(p_5 - p_3)^2 + (p_5 - p_7)^2}$$

$$\text{Edge} = D_1 + D_2 + D_3 + D_4 \tag{2-2}$$

2.13 Summary

This chapter presented the basic concepts of steganography. Firstly, it begins with definitions and historical information about steganography. Then we made comparisons between steganography vs. cryptography and steganography vs. watermarking.

Image steganographic techniques can be categorized into two groups: the Spatial domain technique group and the Transform domain group. The Spatial domain technique embeds information in the intensity of the pixels directly, while the Transform domain technique embeds information in the frequency domain of previously transformed images.

Secondly, we reviewed the basic concepts about the science of image processing and the definition of the digital image, and then the definition of the edge detector and the traditional types that are used in generating edges.

Finally, we reviewed methods to perform the edge detection process; most of them are based on image gradient magnitude, which is calculated with the first derivative of an image. In this study, the edge detection method is performed by calculating the image gradients with the Euclidean distance; which is the most used and the most important method.

CHAPTER THREE FUZZY LOGIC SYSTEMS

3.1 Introduction

The concept of Fuzzy Logic (FL) was initially introduced by Lotfi Zadeh, the founding father of the entire field, in the 1960s[46][41][47]. Fuzzy logic attempts to mimic human decision-making through the use of fuzzy set theory[44], [48][49].

Based on fuzzy logic, we are able to calculate a medium value between absolute true and absolutely false, with values ranging between 0.0 and 1.0. Using fuzzy logic, it is possible to calculate the degree to which an item is a member. Fuzzy logic computes shades of grey between black/white and true/false[46]. Many words and evaluations, we use in our daily reasoning are not clearly defined mathematically. This is why FL is needed. Figure 3.1 and figure3.2 show the difference between the traditional logical systems or the crisp set and the fuzzy set.

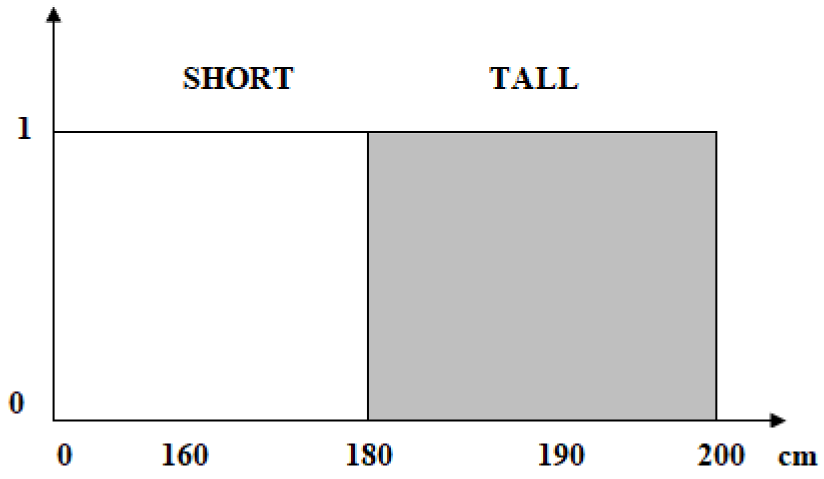


Figure 3.1: Boolean Logic and Crisp Sets.

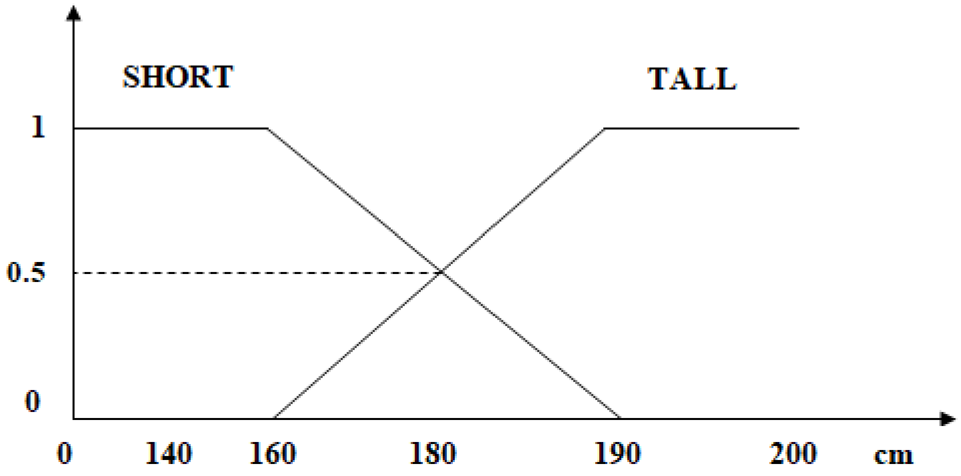


Figure 3.2: Fuzzy Set.

Fuzzy sets allow us to take advantage of the fact that there are no sharp boundaries between sets. There is overlap between the fuzzy sets in certain areas, creating non-crisp or fuzzy boundaries. In addition, note that a given value on the x-axis can belong to more than one fuzzy set, with different membership values.

As shown in Figure 3.1, traditional logical systems use Boolean logic or crisp sets which have sharp boundaries between custom sets. Figure 3.2 illustrates the smooth transition between the fuzzy sets.

From Figure 3.2 we can realize the likeness between human thinking and mathematical expression, so the human always tries to describe the short person from first sight by saying “he/she is a short man” we are not saying “he/she length is 140 cm”.

3.2 Linguistic Variables

In 1975, Zadeh defines linguistic variables as variables whose values are not numbers but words or sentences in a natural or artificial language. The reason for using words or sentences instead of numbers is that linguistic characterizations tend to be less precise than numerical ones[50]. This means that if a variable can take words in natural languages as its values, in this case, it is called a linguistic variable. The words are characterized by fuzzy sets that are defined in the discourse universe in which the variable is defined. Each linguistic variable is characterized by a quintuple (x, T(x), X, G, M) in which x is the name of the variable; T(x) is the linguistic terms set of x, that refer to a base variable whose values being a fuzzy number defined on X; G is a syntactic rule for generating the names of values of x, and M is a semantic rule for associating with each linguistic term $t \in T$ its meaning,*M(t),* which is a fuzzy set on X, that is, *M: T→ F(X)*, where *F(X)* denote s the set of fuzzy sets of X, one fuzzy set for each $t \in T$[51][52].

For example, speed can be interpreted as a linguistic variable and it can be decomposed into the following terms: T (speed) = {slow, moderate, fast, very slow, more or less fast,...}. Where each term in T (speed) is characterized by a fuzzy set in a universe of discourse U = [0, l00] as shown in Figure 3.3.

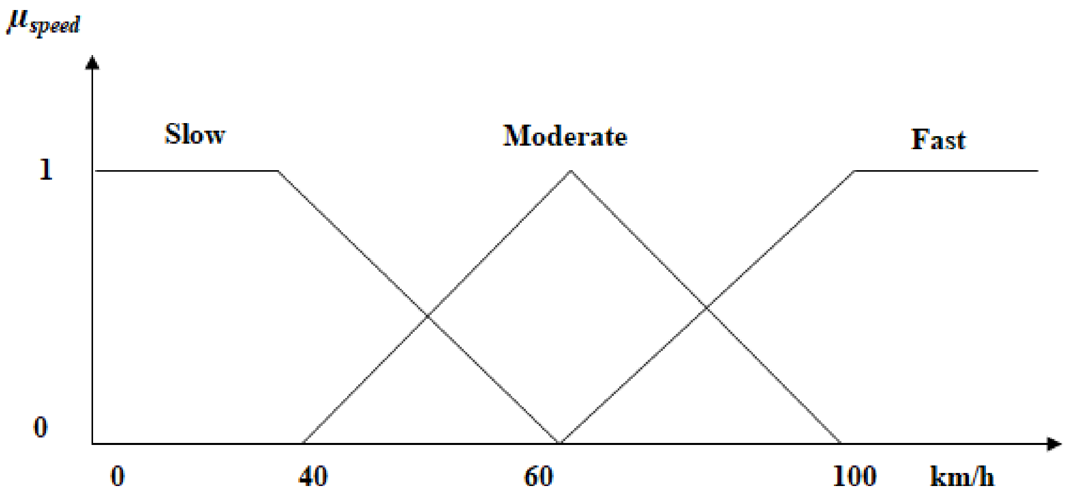


Figure 3.3: Membership Functions for T (speed).

3.3 Membership Functions

Membership functions (MFs), $\mu_F(x)$, in fuzzy logic systems, are associated with the linguistic variables, which help to define the range of values that can be associated with that linguistic label and the degree to which it should be associated. There are various types of membership functions that can be used such as triangular, trapezoidal, Gaussian function, and Bell-shaped[51][52][53]. Figure 3.4 shows an example of define MF.

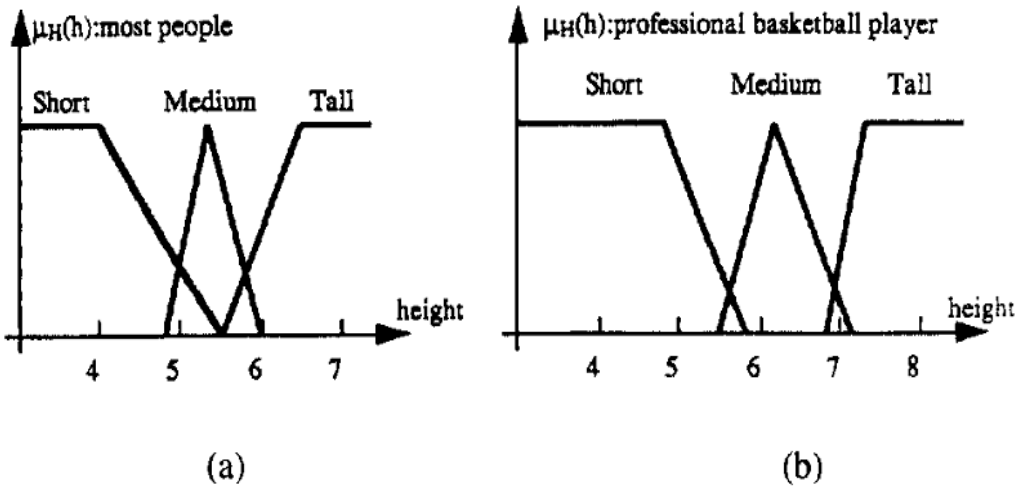


Figure 3.4: Membership functions for T(height) = (short men, medium men, tall men). (a) Most people’s membership functions, and (b) professional basketball player’s membership functions

3.4 Fuzzy Logic and Probability

According to some people, fuzzy logic and probability can't be distinguished. But there are differences between fuzzy logic and probability. The uncertainty concept can be defined using both probability and fuzzy logic. However, the way they deal with uncertainty differs. Probability measures the uncertainty that exists in the occurrence of an event. In contrast, fuzzy logic measures the amount of uncertainty about the characteristics of the event that has occurred. In other words, it measures the grade to which an event occurs, not whether it occurs[52][54].

3.5 Types of Fuzzy Logic Sets

3.5.1 Crisp Sets

In a universe of discourse U, a crisp set A (the set of possible values for a variable) can be identified either by listing its members or by identifying its elements $x \subset A$[51][52][53].One way to do the latter is to specify a condition by which $x \subset A$; thus A can be defined as:

$$\boldsymbol{A = \{x \mid x \text{ meets some condition}\}} \qquad (3\text{-}1)$$

Alternatively, we can introduce a zero-one membership function (also called a characteristic function, discrimination function, or indicator function) for A, denoted by [52]:

$$\mu_A(x), \text{ such that } A \Rightarrow \mu_A(x) = 1 \text{ if } x \in A \text{ and } \mu_A(x) = 0 \text{ if } x \notin A \qquad (3\text{-}2)$$

3.5.2 Type-1 Fuzzy Logic Sets

Type-1 fuzzy sets are the most common and they are commonly known as classical fuzzy sets or fuzzy sets in a traditional context. The following is the formal definition of a type-1 fuzzy set:

Let F be a fuzzy set on a universe of discourse U is characterized by a membership function µF(x) which takes on values in the interval [0, 1]. We can represent the fuzzy set F in U as a set of ordered pairs of a generic element x as[51][52][53]:

$$F = \{(x, \mu_F(x)) \mid x \in U\} \qquad (3\text{-}3)$$

When U is continuous (e.g., the real numbers), F is commonly written as:

$$F = \int_U \mu_F(x) / x \quad (3\text{-}4)$$

In this equation, the integral sign does not denote integration; it denotes the collection of all points$x$∈U with associated membership function $\mu_F(x)$[51][52].

When U is discrete, F is commonly written as:

$$F = \sum_U \mu_F(x) / x \quad (3\text{-}5)$$

In this equation, the summation sign does not denote arithmetic addition; it denotes the collection of all points$x$∈U with associated membership function $\mu_F(x)$ [51][52].

There are different examples of commonly used MFs that are used to describe the type-1 fuzzy sets, such as triangular MF, trapezoidal MF, and Gaussian MF. It can be seen, type-1 fuzzy sets have crisply defined membership functions or degrees of membership [51][55].

3.5.3 Shapes for Membership

1. Triangular: The triangular membership function is represented by Equation (3-6), and it is illustrated in Figure 3.5. This Membership function depends on three parameters (a, b, and c). Where a defined as the starting point, b is defined as the vertex, and c is defined as the endpoint of the triangle. The triangle MFs is defined as[56]:

$$f(x\colon a, b, c) = \begin{cases} \frac{x-a}{b-a} & a \le x \le b \\ \frac{c-x}{c-b} & b \le x \le c \\ 0 & otherwise \end{cases} \quad (3\text{-}6)$$

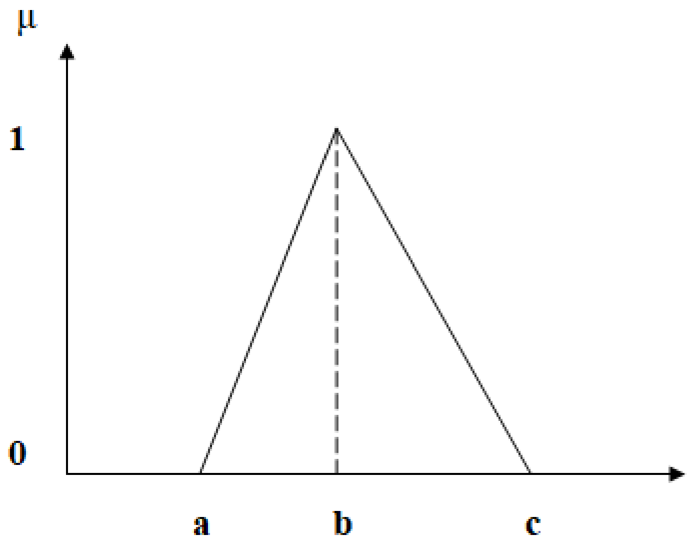


Figure 3.5: Triangular Membership Function.

2. Trapezoidal: The trapezoidal membership function is represented by Equation (3-7), and it is illustrated in Figure 3.6. This Membership function depends on three parameters (a, b, c, and d). Where a and b are defined as the start and end points, c and d are defined as the two points in between. The trapezoidal MF is defined as[56]:

$$f(x\colon a,b,c,d) = \begin{cases} \frac{x-a}{b-a} & a \le x \le b \\ 1 & b \le x \le c \\ \frac{d-x}{d-c} & c \le x \le d \\ 0 & otherwise \end{cases} \tag{3-7}$$

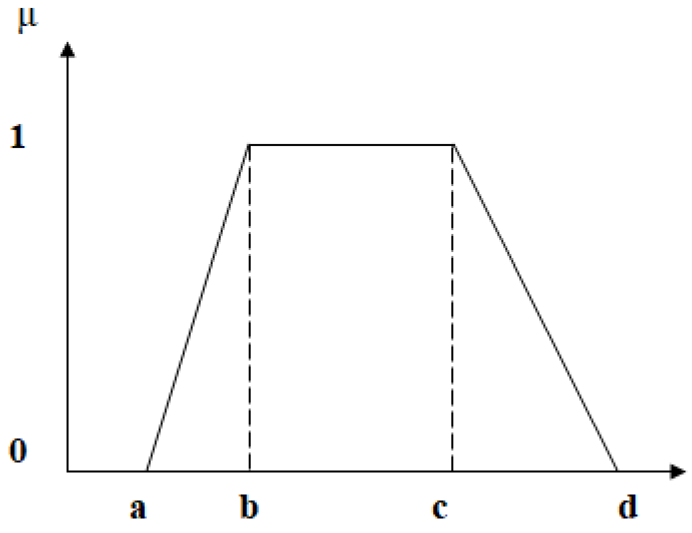


Figure 3.6: Trapezoidal Membership Function.

3. Gaussian: The Gaussian membership function is represented by Equation (3-8), and it is shown in Figure 3.7. This Membership function depends on two parameters (σ and m). Where σ is the standard deviation and m is defined as the mean. The Gaussian MF is given by:

$$f(x{:}\, m, \sigma) = exp\left[\frac{-(x-m)^2}{2\sigma^2}\right] \qquad (3\text{-}8)$$

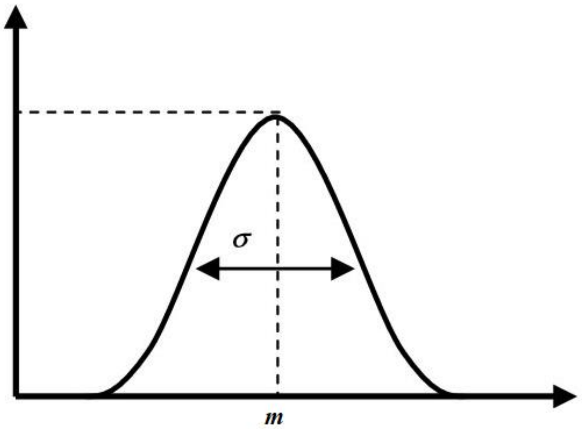


Figure 3.7: Gaussian Membership Function.

3.5.4 Type-2 Fuzzy Logic Sets

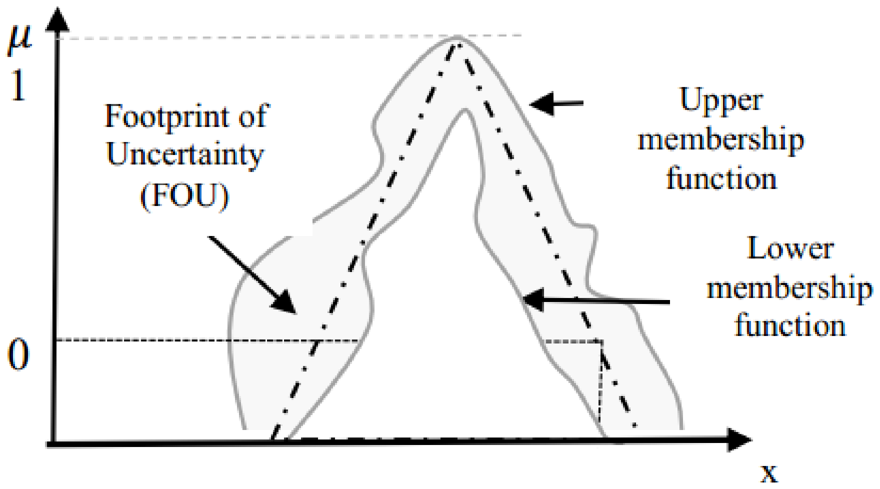
Type-1 fuzzy sets are used when the membership of an element in a set cannot be determined as ordinary sets [0 or1], then fuzzy sets can be used. The fuzzy set of type 2 is a concept that was first introduced in 1975[57][58][59][60]. Type-2 fuzzy sets are useful when it is difficult to determine the precise membership function grade even as a crisp number in the [0, 1] range[47][51][61][62].

The type-1 fuzzy set cannot handle (model) the high level of uncertainties. Type-2 fuzzy sets are an extension of type-1 fuzzy. Type-2 fuzzy sets have a three-dimensional membership function and a Footprint of Uncertainty (FOU) located between the upper member and the lower membership, which provide additional degrees of freedom to handle and model higher degrees of uncertainties[47][49][51][62]. Figure 3.8 shows the membership function and a Footprint of Uncertainty (FOU) of the Type-2 fuzzy set. It is possible to use interval type-2 fuzzy sets to characterize inputs or/and outputs FLS by considering the degrees of freedom and complexity of a type-2 set[63][64][65][66][67].

$$\tilde{A} = \{((x,u),\mu_{\tilde{A}}(x,u)) \mid \forall x \in X, \forall u \in J_x \subseteq [0,1]\} \qquad (3\text{-}9)$$

In which $0 \leq \mu\tilde{A}(x, u) \leq 1$. Ã can also be expressed as:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x,u) \,/\, (x,u) J_x \subseteq [0,1]\} \qquad (3\text{-}10)$$



μ
1
0
Footprint of Uncertainty (FOU)
Upper membership function
Lower membership function
x

Figure 3.8: Membership of a type-2 fuzzy set[47][62]**.**

3.6 Type-1 Fuzzy Logic System (T1FLS)

A type-1 fuzzy logic system maps crisp inputs into crisp outputs by using the theory of fuzzy sets. The Mamdani fuzzy model consists of four major components as shown in Figure 3.9, which are fuzzifier, inference engine, rules, and defuzzifier.[44][48][49][51][42]. Type-1 fuzzy sets handle the uncertainties related to FLS antecedents and Consequents by using precise and crisp membership functions[68][69][64]. When the type-1 membership functions have been selected, all the uncertainty disappears, due to type-1 membership functions being completely precise[66].

3.6.1 Fuzzifier

The fuzzifier maps a crisp input x=(x_1,...x_p)^T ∈ $X_1 \times X_2 \times ... \times X_p \equiv$ X into fuzzy sets A_x in X and this mapping can be expressed as y=f(x). There are two types of fuzzifier, a singleton which is used with a precise value, and a non-singleton can be used with imprecise measures, which means the probabilistic fuzzifier can be used when receiving input data from any inaccurate devices or sensors. It can also be used when the data is disturbed by random noise[51][70].

3.6.2 Rules

The rules are the core of FLS. Each If-Then statement in a rule has antecedents and consequents. Antecedents and consequents are represented through the fuzzy sets of input linguistic variables and output linguistic variables, respectively. Rule Base can be obtained using data from the system, or designed by experts or consultants[51][70][71][72][73].

3.6.3 Fuzzy Inference Engine

The fuzzy inference engine(which is labeled Inference in Figure 3.9), plays a very essential role. It employs the Rule Base to transform fuzzy input sets received from the fuzzifier into fuzzy output sets. In addition, it controls the process of selecting and combining rules from the rule-base[51][70].

3.6.4 Defuzzifier

The defuzzifier is considered as a concluding unit in the fuzzy logic system. It converts the fuzzy sets obtained by the inference engine to crisp output[47][48][51][42][70].

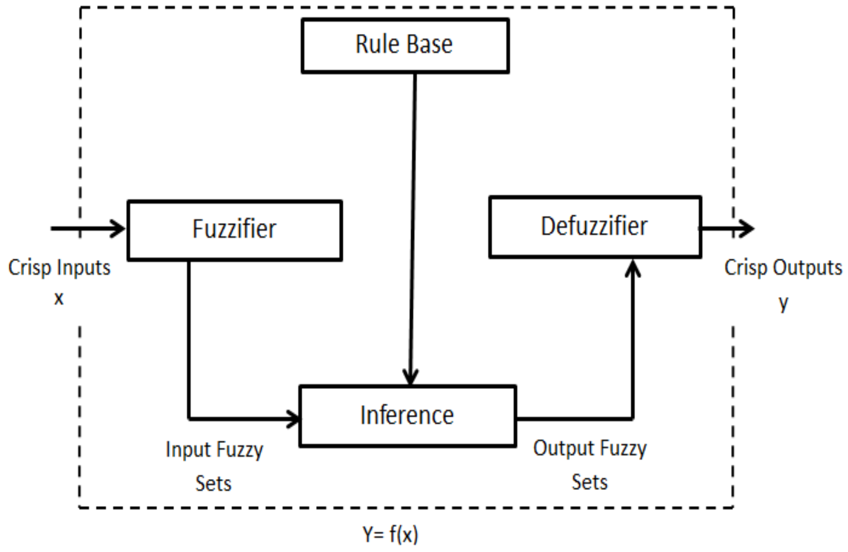


Figure 3.9: Type-1 Fuzzy Logic System[51]**.**

3.7 Type-2 Fuzzy Logic Systems (T2FLS)

The type-2 fuzzy logic system shown in Figure 3.10 was introduced by Lotfi Zadeh in 1975[74][59][75][76][77]. T2FLS consists of five major components, which are the fuzzifier, inference engine, rules, type-reducer, and defuzzifier. The type-2 fuzzy sets are an extension of ordinary type-1 fuzzy sets where a type-2 fuzzy set can fully handle the high levels of uncertainties associated with control applications. A type-2 fuzzy set whose membership function (MF) grades themselves are type-1 fuzzy sets. A type-2 membership grade can be any subset in [0, 1] which is called the primary membership; and corresponding to each primary membership, there is a secondary membership grade that is a crisp number in [0,1][54][59][70][75][63][77]. As shown in Figure 3.11, the type-2 fuzzy set has a three-dimensional membership function and includes a footprint of uncertainty (FOU). A FOU can be described in terms of an upper membership function and a lower membership function; that gives additional degrees of freedom to make it reasonable to handle a high level of uncertainties [78][79][80]. The type-2 fuzzy sets are beneficial where it is difficult to determine the exact and precise membership functions [76][81].

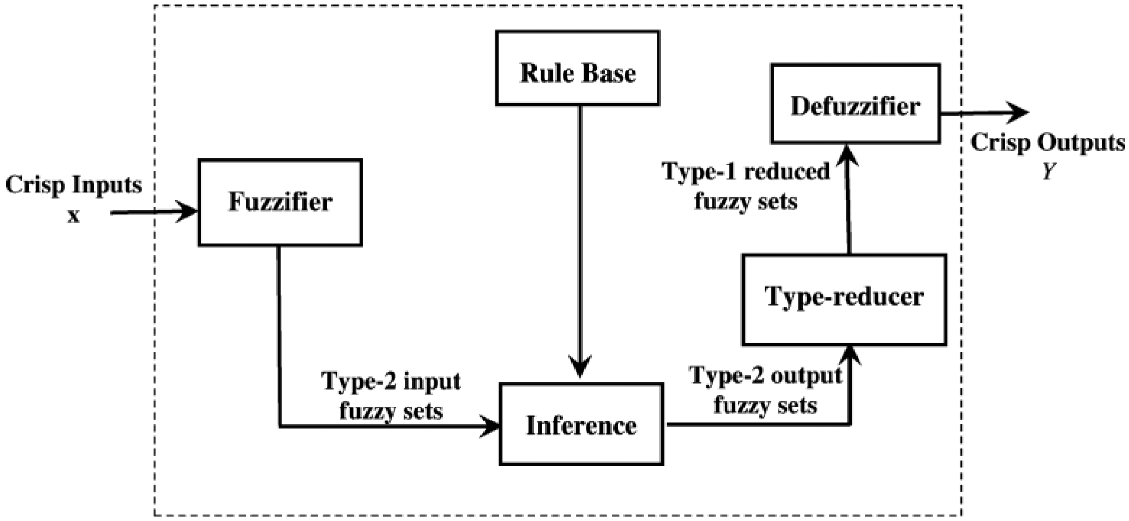


Figure 3.10: Type-2 Fuzzy Logic System[51]**.**

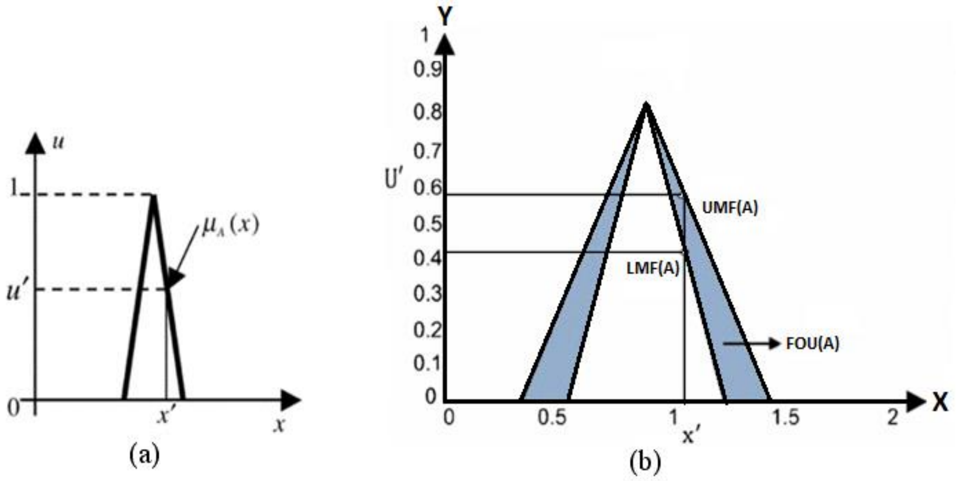


Figure 3.11: (a) A type-1 fuzzy set the membership grade for each element is a crisp number in [0,1].(b) A type-2 fuzzy set is characterized by a three dimensional membership function and a Footprint of Uncertainty (FOU). Interval Type-2 Fuzzy Set. LMF = Lower Membership Function. UMF = Upper Membership Function. FOU = Footprint of Uncertainty.

3.7.1 Fuzzifier

Similarly, as in type-1, the fuzzifier in T2FLS maps a crisp input $x=(x_1,..,x_p)^T \in X_1 \times X_2 \times ... \times X_p \equiv X$ into a type-2 fuzzy set. The resulting fuzzified value is a type-1 fuzzy set representing the primary grades for each input; corresponding to each primary membership, there is a secondary membership grade that is a crisp number in [0, 1]. For a type-2 fuzzy system, three kinds of fuzzifiers are possible: singleton, type-1 non-singleton, and interval type-2 non-singleton[51][70].

3.7.2 Rules

In the type-2 FLS, the rules are exactly the same as in the type-1 fuzzy system. The only distinction between them is that some or all of the fuzzy sets are type-2 fuzzy sets[51][70][71][72].

3.7.3 Fuzzy Inference Engine

As the same as the inference engine of type-1 FLS, the fuzzy inference engine of type-2 FLS plays an important role. Where it converts fuzzy input sets received from the fuzzifier into fuzzy output sets using the rule base[51][70][82][83].

3.7.4 Type reducer

The type reduction receives type-2 fuzzy sets, which are obtained from the inference engine step, and returns type-1 fuzzy sets as outputs[48][51][70][82].

3.7.5 Defuzzifier

As part of the T2FLS, the defuzzifier is regarded as the final unit as the same as the T1FLS. Where it converts the type-1 fuzzy sets obtained by the type reducer to crisp output[47][48][51][70][82].

3.8 Summary

In this chapter, the basic concepts related to fuzzy logic were presented. First, we started with the difference between the crisp sets and fuzzy sets; then it reviewed the types of membership functions that are commonly used to describe the fuzzy sets, such as triangular MF, trapezoidal MF, and Gaussian MF. Finally, we defined type -1 Fuzzy Logic Systems and Type-2 Fuzzy Logic Systems.

CHAPTER FOUR LITERATURE REVIEW ON STEGANOGRAPHY

4.1 Introduction

In this chapter, the literature reviews were classified depending on the techniques used as follows:

4.2 Traditional Image Steganography Techniques

In [84], Wu et alproposeda steganographicmethod based on least-significant-bit (LSB) replacement and pixel-value differencing (PVD) method. Firstly, they obtain a different value from two consecutive pixels though utilizing the PVD method. The location of small difference value can be on a smooth area and the large one can be located on an edged area. In the smooth areas, they hid the secret data into the cover image by LSB method while using the PVD method in the edged areas.

A novel steganography approach based on the combination of LSB substitution mechanism and edge detection was proposed by Bai et al[14].The cover pixels were classified by edge areas and non-edge areas. Then, pixels that belong to the edge area are used to carry more secret bits. Moreover, further increase the payload as well as preserve good image quality.

In [17], A. Kaur et al proposed an image steganography scheme which is a kind of spatial domain technique. In order to embed a secret message in a cover image, the first component alteration technique is used. Their proposed Techniques focus only on the two or four bits of a pixel in an image (at the most five bits at the edge of an image) which results in less peak to signal noise ratio and high root mean square error. They used in their technique; 8 bits of blue components of pixels are replaced with secret data bits.

A. Ioannidou et al. [6] presented a novel technique for image steganography which one of the techniques taking advantage of sharp areas in images in order to hide a more amount of secret data. Specifically, the technique is based on the edges present in an image. A hybrid edge detector is used for their purpose. The edges found by a fuzzy edge detector and the Canny edge detector are unified in order to find a larger set of edges. Moreover, a high payload technique for color images is exploited. These two techniques are combined in order to produce a new steganographic algorithm. Experimental results show that the new method achieves a higher peak signal-to-noise ratio for the same number of bits per pixel of an embedded image.

4.3 Genetic Algorithm Steganography Techniques

AM Fard et al. [34] proposed a novel genetic algorithm evolutionary process to create a secure steganographic encoding on JPEG images. Their steganography step depended on OutGuess which is proved to be the least vulnerable steganographic system. They used message embedding positions as their search space and then applied the genetic algorithmic operators to find the best combination of message and image.

In[85], Wang et al presented a new steganography based on genetic algorithm. After embedding the secret message in LSB (least significant bit) of the cover image, the pixel values of the stego-image are changed by the genetic algorithm to keep their statistic characters. Thus, the existence of the secret message is hard to be detected by the RS analysis. Their proposed algorithm led to better visual quality achieved. The experimental results show the proposed algorithm's effectiveness in resistance to steganalysis with better visual quality.

In [86], HR Kanan et al proposed a tunable visual image quality and data lossless method in spatial domain based on a Genetic Algorithm (GA). The proposed technique lied on modelling the steganography problem as a search and optimization problem. Experimental results, compared to other currently popular steganography techniques, show that the proposed algorithm not only achieves high embedding capacity but also enhances the PSNR of the stego image.

A. Conci et al. [87] presented the AES cryptography algorithm, to improve the hidden data security in two methodologies for steganography: the genetic algorithm and path relinking. It also combines them proposing a new hybrid approach that outperforms the LSB (least significant bits) substitution technique presented in works cited in the literature concerning the quality of a stego image. It improves the possibility of hiding data inside color images significantly, increasing the space available for information by more than three times when compared to the usual steganography approach used by grayscale images. Moreover, all types of digital information from text and compressed files to even executable programs can be hidden inside the cover image. This considerably increases the scope of application of the technique for transmitting information inside a typical image, hiding the data from intruders.

Genetic Algorithms are also used in securing data in steganographic schemes. A.Khamrui et al. [88] propose a steganographic model based on the concept of GA in the frequency domain. In this scheme, four frequency components are generated through the process of DCT that is applied on a 2 × 2 submask of the carrier image.

Due to the small size of the masks and the low embedding capacity of DCT, this scheme can hide a small size of secret data. By hiding large data, the distortion of the stego becomes noticeable by human eyes.

4.4 Neural Network Steganography Techniques

In[89], A. Rana et al proposed image steganography method based on kohonen neural network. Kohonen network is trained due to the absolute contrast sensitivity of pixels present in cover image. Trained network classify the pixels in various classes of sensitivity. Data embedding is performed in less sensitive pixels by LSB substitution method. The observed was that the capacity and security increase with acceptable PSNR in the proposed algorithm compared to the existing algorithm.

N. N. El-Emam et al. [90] proposed three-phase intelligent technique to develop the data-hiding algorithm in colour images with imperceptibility. The first phase of the learning system (LS) has been firstly applied, whereas the other phases have been applied later after the hiding process. The first phase has been constructed to estimate the number of bits to be hidden at each pixel; this phase is based on adaptive neural networks with an adaptive genetic algorithm. The LS of the second phase has been introduced as a detector to check the performance of the proposed steganographic algorithm by establishing rich images model from available cover and stego images. The LS of the last phase has been implemented through three steps, and it is based on a concurrent approach to develop the stego image and protect against attacks. The results of the proposed algorithm can efficiently embed a large quantity of data, up to 12 bpp (bits per pixel), with better image quality.

4.5 Fuzzy Logic Steganography Techniques

In [16] Chen et al. presented a high embedding capacity steganography scheme with a hybrid edge detector. Their scheme was constructed by a combination of the fuzzy edge detector and the Canny edge detector, using the grayscale image to create the hybrid edge image. The embedding operation consists of dividing the edge image into a set of blocks. Each block contains n pixels. They used the first pixel to store the status of the rest pixels. The status of each pixel is classified as ‘1’ or ‘0’ if the pixel is an edge pixel, or non-edge pixel, respectively. Depending on this classification, the x secret bits are embedded into the edge pixel, and y secret bits are embedded into the non-edge pixels, using the LSB substitution technique. For example, take a block B = [P1, P2, P3, P4] pixels, with n = 4, x=1 and y=3. The binary values of these pixels are {[10101010], [10000000], [11111100], [00001111]} respectively, with the secret message S = ‘0110101’. Assume P2 and P4 are edge pixels. Depending on this, the status of P2, P3 and P4 is ‘101’. Replace three LSB in pixel P1 with ‘101’ to store the status of the rest pixels into the block. The new value of pixel P1 is [10101101]. Then, P2 and P4 are used to embed three secret message bits (y = 3), while P3 is used to embed one secret message bits (x = 1). The new values of pixels P2, P3and P4 after embedding process are [10000011], [11111100] and [00001101], respectively. In the extraction operation, the inverse process is executed to retrieve the secret message from the stego image according to the status of each pixel, stored in the first pixel of each block. If the pixel status is an edge pixel, extract three LSB, or extract one LSB, if it is a non-edge pixel.

Tseng and Leng[15] proposed a block-based scheme using a hybrid fuzzy edge detector, which extended [6] to achieve minimal distortion. the proposed scheme has one parameter x, for which the number of embedding secret message bits of non-edge pixels are represented, instead of using the two parameters x and y, as was done in Chen et al.'s scheme. The number of edge pixel embedded should be greater than x , so it uses two bits to present four cases of [x, x+1], [x, x+2], [x, x+3], and [x, x+4]. The second element represents the number of secret bits that should be embedded in edge pixels. In order to achieve the minimal distortion, one of the four cases is chosen by calculating minimum mean square error (MSE) for each 4×4 pixels block.

In [14] J. Bai et al. proposed a scheme based on the LSB methodology, combined with the edge detector, which uses the principle that edge areas can tolerate a larger number of embedded bits more than smooth areas according to HVS. In their scheme, they use the cover image to generate Most-Significant-Bit (MSB) image by clearing the last 5 LSBs of each pixel in the original image for edge detection. The 5 LSBs are employed for embedding the secret data while 3 MSBs of all pixels remain unchanged. They categorize the pixels of the cover image into two categories, which are non-edge pixels and edge pixels, respectively. Each cover pixel in the first category contains 'x' secret message bits, and the second category contains 'y' secret message bits, using LSBs substitution. For these two categories, pixels are embedded by the k-LSB substitution, where the value k equals either x or y, which is decided by the edge information. The secret key K is shared between the sender side and the receiver side. For example, suppose the block is = 4 pixels, that is, P1 = [10011011], P2 = [01111110], P3 = [01011000], P4 = [10011100], x = 2, y = 4, then the secret bit S = ' 101001111110 '. Based on the edge information of these four pixels, we know that P1 and P4 are edge pixels and P2 and P3 are non-edge pixels. P1 and P4 pixels will include 4 bits of a secret message, while P2 and P3 will include 2 bits of a secret message. These four pixels will switch to P1'= [1001 1010], p2'= [01111101], P3'=

[010110 11] and P4' = [1001 1110]. In the extraction phase, the receiver retrieves the two parameters x and y from the last four pixels of the image. And also, the edge information is determined the same as in the embedding phase.

In [43]CI Gonzalez et al. proposed a new general type-2 fuzzy logic method for edge detection applied to color format images. The proposed algorithm combines the methodology based on the image gradients and general type-2 fuzzy logic theory to provide a powerful edge detection method. General type-2 fuzzy inference systems are approximated using the α-planes approach. The edge detection method is tested on a database of color images with the idea of illustrating the advantage of applying the fuzzy edge detection approach on color images against grayscale format images, and also when the images are corrupted by noise. They compare the proposed method based on general type-2 fuzzy logic with other edge detection algorithms, such as ones based on type-1 and interval type-2 fuzzy systems. Simulation results show that edge detection based on a general type-2 fuzzy system outperforms the other methods because of its ability to handle the intrinsic uncertainty in this problem.

4.6 Summary

In this chapter, we have presented the literature reviews into four classifications depending on the techniques. Firstly, we started with the traditional image steganography techniques and their simplicity in solving the problems. Secondly, we reviewed the genetic algorithm techniques and neural network techniques and their ambiguous in solving problems. Finally, we presented the fuzzy logic techniques and their abilities in solving problems.

SECTION THREE PROPOSED METHODOLOGY

CHAPTER FIVE: THE PROPOSED EDGE DETECTOR BASED ON THE GRADIENT APPROACH AND TYPE-1 FUZZY LOGIC SYSTEM

CHAPTER SIX: THE PROPOSED ENHANCED EDGE DETECTOR BASED ON THE GRADIENT APPROACH AND TYPE-2 FUZZY LOGIC SYSTEM

CHAPTER FIVE THE PROPOSED EDGE DETECTOR BASED ON THE GRADIENT APPROACH AND TYPE-1 FUZZY LOGIC SYSTEM

5.1 Introduction

We present a new LSB steganography method in this chapter, which uses a hybrid edge detector to make disclosing the existence of a secret message a hard operation.

5.2 The Proposed Type-1 Fuzzy Logic Embed System

Figure 5.1 shows the proposed fuzzy logic embedding system. In the fuzzy phase, the edge detector is calculated based on the Gradient Approach and Type-1 Fuzzy Logic System. In the embedding phase, exploiting the edge image that has been obtained from the previous phase in embedding more secret bits in edge pixels than in non-edge pixels. The proposed systems developed into two sides, the sender's side that treats with the embedding process, and the receiver's side that treats with extraction processes. On the receiver's side, only the stego image will be the input of the extraction algorithm to get the secret message back again.

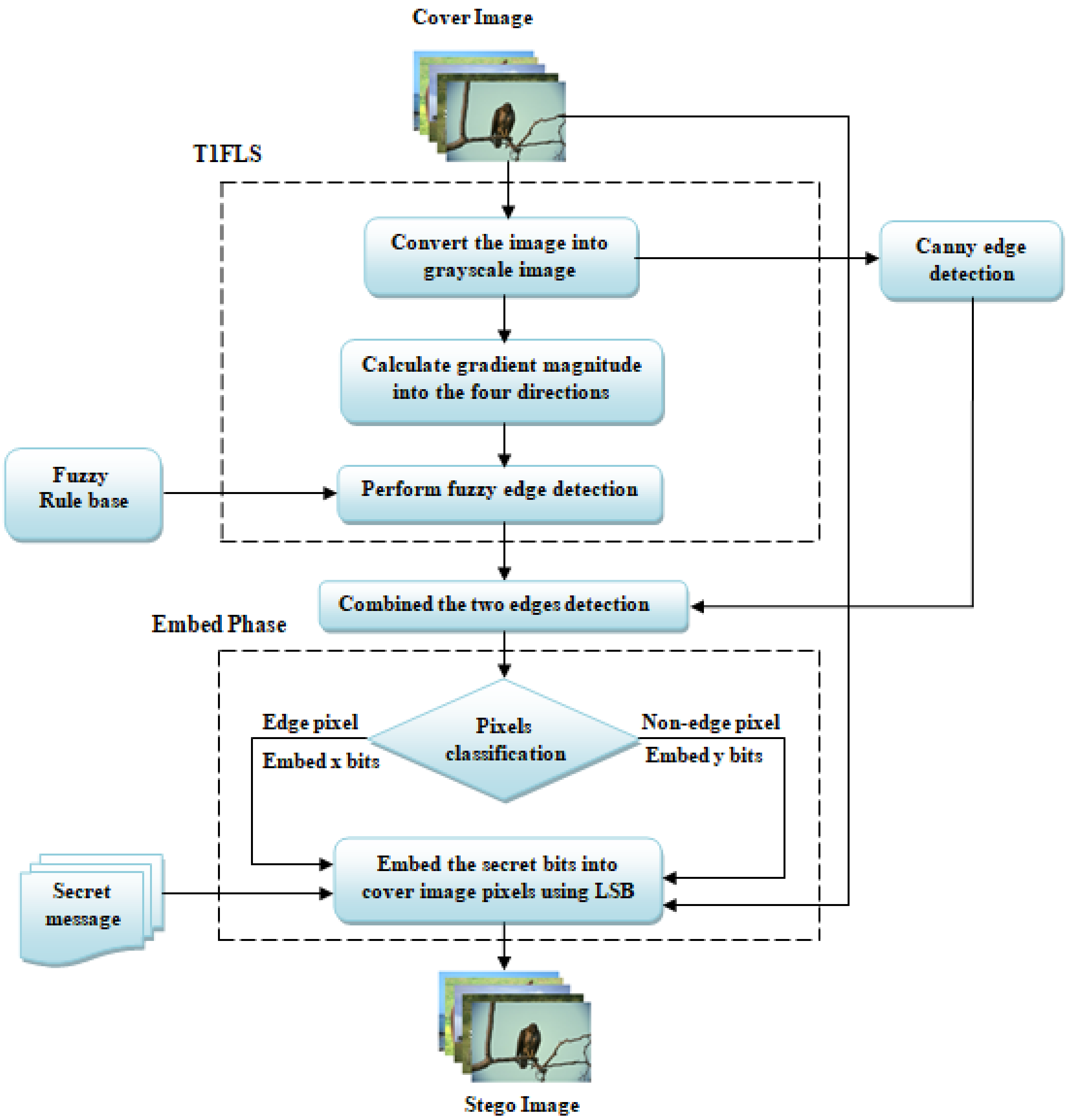

Cover Image
T1FLS
Convert the image into grayscale image
Canny edge detection
Calculate gradient magnitude into the four directions
Fuzzy Rule base
Perform fuzzy edge detection
Combined the two edges detection
Embed Phase
Edge pixel
Embed x bits
Pixels classification
Non-edge pixel
Embed y bits
Secret message
Embed the secret bits into cover image pixels using LSB
Stego Image

Figure 5.1: proposed Type-1 Fuzzy logic Embedding System.

5.2.1 FLS Phase

5.2.1.1 Pre-processing Operation

Some processes are done prior to initiating the embedding operation. On the sender's side, the cover image will be converted into grayscale images, pass a copy of this grayscale image to the canny edge detector, and another copy to the gradient type-1 fuzzy logic system edge detector.

5.2.1.2 Calculate The Gradient T1FLS Edge Detector

The fuzzy logic methodology for edge detection using gradient magnitude consists of using Eq (2-1) To get the gradients in the four directions (D1, D2, D3, D4) and use them as inputs to a fuzzy inference system (FIS), instead of the Eq (2-2). Several kinds of membership functions exist to represent T1FIS, such as Triangular, Trapezoidal, Gaussian, etc. The Gaussian membership function of the proposed T1FLS is illustrated in Figure 5.2.

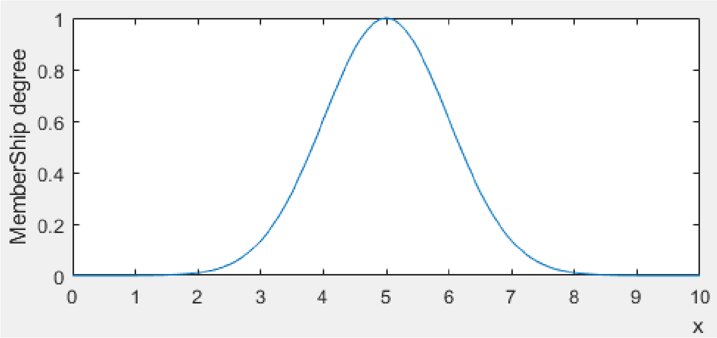


Figure 5.2: Type-1 membership function[44]

In this study, the T1FLS is a singleton Mamdani type, which was designed with four inputs (D1, D2, D3, and D4), and one output. The inputs and output are fuzzified using Gaussian membership functions with uncertain mean; each input has three linguistic values (low, medium and high) to determine the grade to which the evaluated gradient corresponds, to be the output edge. Each output has two linguistic values (edge and background) to produce the gradient magnitude edge detector.

For each D input, the Gaussian membership functions were obtained with (5-1) to (5-6), and the centers of each function were obtained with (5-1) to (5-3), as shown in Figure 5.3. For the output E (the edges), we can obtain these membership functions directly with (5-7) to (5-8)[67][45].

$$\mathrm{low} = \mathrm{mi\,n}(\mathrm{D_i}) \tag{5-1}$$

$$\mathrm{high} = \mathrm{ma\,x}(\mathrm{D_i}) \tag{5-2}$$

$$\mathrm{medium} = \mathrm{low} + (\mathrm{high} - \mathrm{low})/\,2 \tag{5-3}$$

$$\sigma = \mathrm{high}/8$$

$$\mu(\mathrm{low}) = e^{\frac{-(x-\mathrm{low})^2}{2(\sigma)^2}} \tag{5-4}$$

$$\mu(\mathrm{high}) = e^{\frac{-(x-\mathrm{high})^2}{2(\sigma)^2}} \tag{5-5}$$

$$\mu(\mathrm{medium}) = e^{\frac{-(x-\mathrm{medium})^2}{2(\sigma)^2}} \tag{5-6}$$

$$\mu(\mathrm{background}) = e^{\frac{-(x-\mathrm{black})^2}{2(\sigma)^2}} \tag{5-7}$$

$$\mu(\mathrm{edge}) = e^{\frac{-(x-\mathrm{white})^2}{2(\sigma)^2}} \tag{5-8}$$

$$\mathrm{wherewhite} = 255, \sigma = \mathrm{white}/8$$

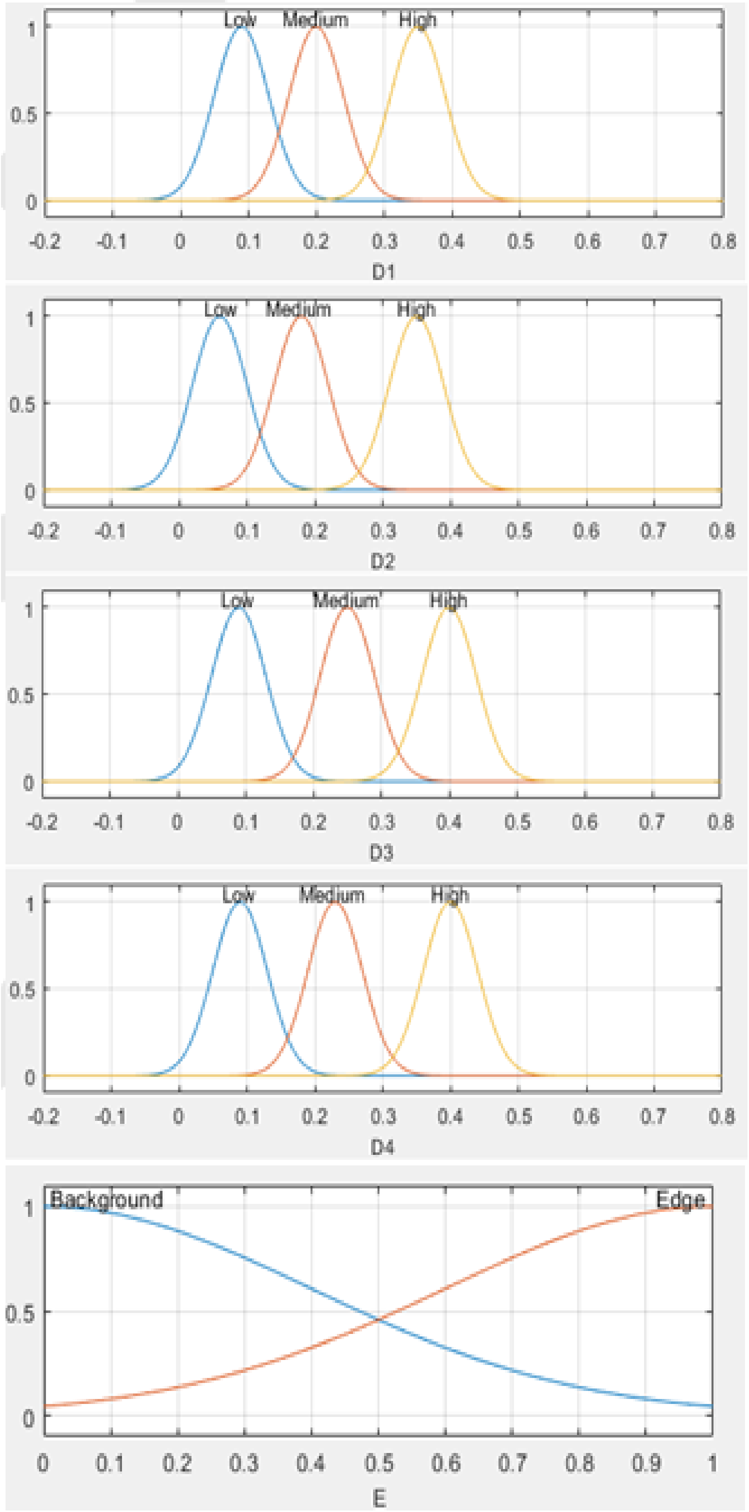


Figure 5.3: T1FIS membership function for the inputs (D1, D2, D3 and D4) and the output E[44]

The important part of a fuzzy system is the fuzzy rules, for this proposed method the fuzzy rules consider various combinations of the gradients inputs Di to produce the gradient magnitude output. Fuzzy rules presented in Table 5.1.

Table 5.1: Three Fuzzy Rules for Edge Detection[45]

Fuzzy Rules
1. If (D1 is HIGH), or (D2 is HIGH), or (D3 is HIGH), or (D4 is HIGH), then (E is Edge.) 2. If (D1 is MEDIUM), or (D2 is MEDIUM), or (D3 is MEDIUM), or (D4 is MEDIUM), then (E is Edge.) 3. If (D1 is LOW), (D2 is LOW), (D3 is LOW) and (D4 is LOW), then (E is Background.)

The first rule tests the four directions (D1, D2, D3, and D4) if it is high this means an edge. The second rule tests the four directions (D1, D2, D3, and D4) if it is medium this means also an edge. The third rule is only to confirm the first two, because if the four directions are low this mean there is no edge in this pixel[45].

5.2.2 Combined the Edge Detections

After that, the two edge detectors (canny and T2FLS edge detector) will combine together to have a new hybrid edge image. The hybrid edge image, cover image and the secret message will be the inputs of (embedding algorithm) the LSB substitution algorithm.

5.2.3 Embed Phase

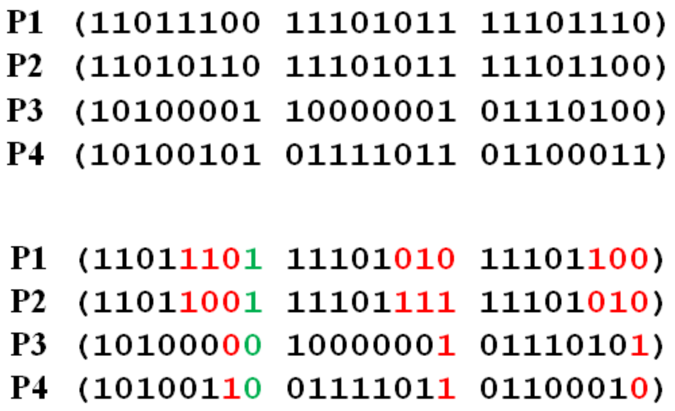
5.2.3.1 Pixels Classifications and Embed Operation

Figure 5.1 shows that the embedding operation is responsible for hiding the secret message into the cover image file, using the proposed LSB method that uses the spatial domain of the RGB color image. On the sender's side, the secret message will be embedded into the cover image file and obtained the stego image file as output.

The operation of embedding a secret message in a cover image depends on the proposed hybrid edge image obtained from the combined edge detection step. The number of bits that should be embedded in each pixel is determined by the category of the pixel in the cover image. We utilize two parameters: x=9 and y=3. If the pixel is an edge pixel, the number of secret bits to be embedded will be x bits, and if the pixel is a non-edge pixel, the number of secret bits to be embedded will be y bits. The first bit in the red color of the cover image will be 0 or 1 to indicate that it is a non-edge or edge pixel respectively. Pixels are embedded by the M-LSB substitution, where the value M equals either x or y, which is decided by the edge information.

5.3 Example

Assume that 4 pixels [P1, P2, P3, and P4], are read from the cover image I. According to the edge information of these four pixels, we know that both the first and second pixels are edge pixels, and that the third and the fourth pixels are non-edge pixels. Consider that the secret bit stream s='110010100100111010011110'. The 9 LSBs of the first and the second pixels are replaced with the corresponding secret bits, and the first bit of red will be 1 (because it is an edge pixel). Similarly, the 3 secret bits are embedded into the third and the fourth pixels by the LSB substitution method, and the first bit of red will be 0, because it is a non-edge pixel. The values of the pixels before and after the embedding operation will be shown in figure 5.4:



```
P1  (11011100 11101011 11101110)
P2  (11010110 11101011 11101100)
P3  (10100001 10000001 01110100)
P4  (10100101 01111011 01100011)

P1  (11011101 11101010 11101100)
P2  (11011001 11101111 11101010)
P3  (10100000 10000001 01110101)
P4  (10100110 01111011 01100010)
```

Figure 5.4: Example of the Proposed Embedding Operation[42]**.**

5.4 Extracting Operation

In the extraction operation, first, the receiver retrieves the number of pixels exploited to embed the secret message from the header of the image. After that, the receiver extracts the first bit in red color if it is 0 or 1 to determine the parameter value of m. If the first bit of red color is 0, this means the value of m will be 3 bits (1bit in red, 1bit in green and1bit in blue color.), and if it is 1, this means the value of m will be 9 bits (3bits in red, 3bits in green and 3bits in blue color). Thus, the secret data will be accurately extracted.

5.5 Result and Discussion

The previous sections have shown the descriptive details of the proposed method. In this section, experimental results are performed to demonstrate the performance of the proposed method by applying the proposed T1FLS on six 128×192 RGB color images from the BSD300 dataset, three of the training images, and three from the testing, respectively as shown in Figure 5.6.

The payload (also called embedding capacity) is measure using the maximum number of embedded bits per pixel (bpp). Its formula is defined as follows:

$$\mathrm{bpp} = \frac{\mathrm{MaximyalEmbeddingbits}}{\mathrm{H} \times \mathrm{W}}, \qquad (5\text{-}9)$$

where H and W, respectively, are the height and width of the original cover image.

We use two viewpoints to measure the quality of stego image. The first one is **the peak signal-to-noise ratio (PSNR)** metrics, which calculate the difference between the stego and cover images, where a higher PSNR means better quality than the stego image.

In the second one, we evaluate the quality of the stego image against that of the cover image, as seen by the **human visual system (HVS).**PSNR formula is defined as follows:

$$\mathrm{PSNR} = 10.\log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right)(\mathrm{dB}), \qquad (5\text{-}10)$$

where the (MSE) is the mean square error between the cover image and stego image. For a cover image with width W and height H, the MSE formula is defined as follows:

$$\mathrm{MSE} = \sum_{i=1}^{H} \sum_{j=1}^{W} \left(p_{ij} - p_{ij}\right)^2 / \left(\mathrm{H} \times \mathrm{W}\right) \qquad (5\text{-}11)$$

Figure 5.5: 'Lena' image[44][42]**.**

3096	42049	253027
113044	249061	253036

Figure 5.6: Figure 5.6: Six128×192 images from BSD300 dataset[44][42]**.**

5.6 Experimental Results for T1FLS Edge Detector

The implementation detail of T1FLS will be presented and discussed by applying different edge detectors, which are shown in Figure 5.8. The PSNR and HVS have been used to measure the quality of the stego image in each experiment. Figure 5.7 illustrates the comparison between the various types of edge detectors. From this comparison, it can be seen that the fuzzy logic edge detector has a larger number of edge pixels.

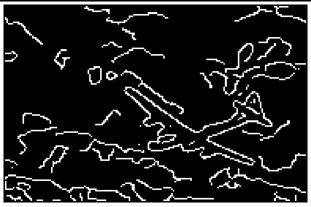
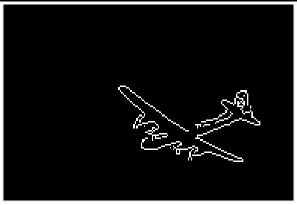
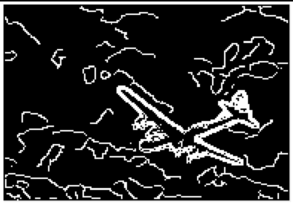
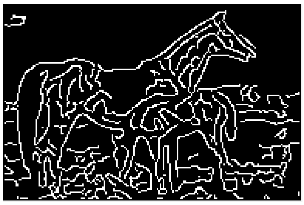
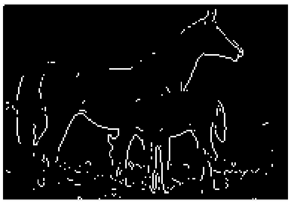
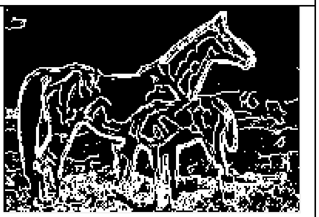
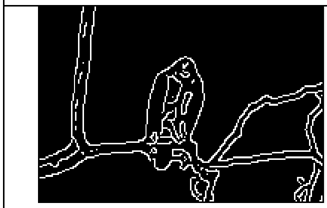
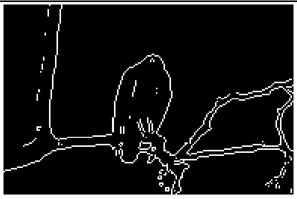
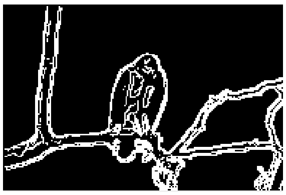
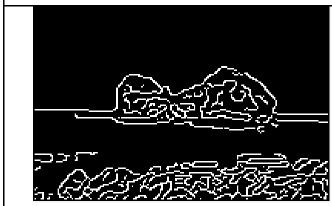
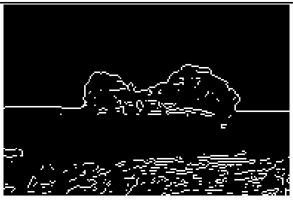
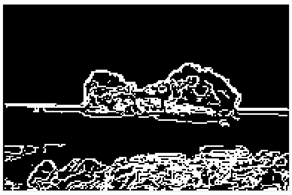
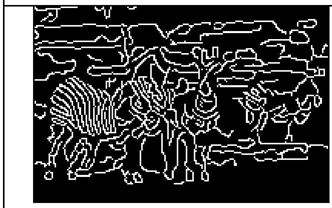
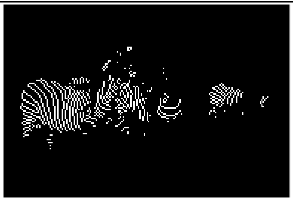
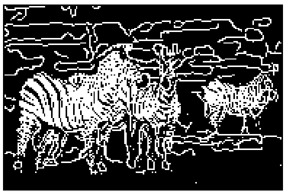
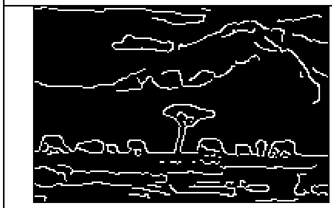
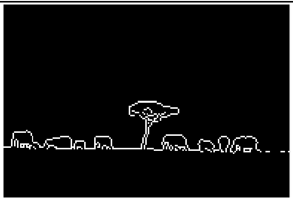
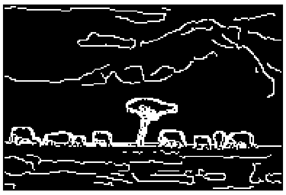
Edge Image		
Canny	**Sobel**	**The Proposed T1FLS**
		
2008	**439**	**2700**
		
3117	**839**	**5717**
		
1624	**1053**	**3432**
		
2203	**1265**	**3867**
		
3874	**1004**	**6467**
		
1995	**536**	**2600**

Figure 5.7: The number of edge pixels detected by Canny, Sobel, and proposed T1FLS[44]**.**

Canny edge detection		
Image name = 3096	Image name = 3096	Image name = 3096
PSNR = 48.4630 dB	PSNR = 45.0340 dB	PSNR = 44.5054 dB
payload = 1.3301 bpp	payload = 2.6680 bpp	payload = 3.0010 bpp
Ratio = 32688 bits	Ratio = 65568 bits	Ratio = 73752 bits
Image name = 113044	Image name = 113044	Image name = 113044
PSNR = 48.1963	PSNR = 44.2973	PSNR = 42.7058
payload = 1.3339	payload = 2.6679	payload = 3.6337
Ratio = 32784	Ratio = 65568	Ratio = 89304
Sobel edge detection		
Image name = 42049	Image name = 42049	Image name = 42049
PSNR = 50.0422	PSNR = 47.8200	PSNR = 45.4341
payload = 1.3301	payload = 2.00	payload = 3.0010
Ratio = 32688	Ratio = 49152	Ratio = 73752

Image name = 249061	Image name = 249061	Image name = 249061
PSNR = 50.1927	PSNR = 46.2740	PSNR = 45.3674
payload = 1.3339	payload = 2.6679	payload = 3.0009
Ratio = 32784	Ratio = 65568	Ratio = 73752
proposed T1FIS edge detection		
Image name = 253027	Image name = 253027	Image name = 253027
PSNR = 46.4572	PSNR = 42.2049	PSNR = 40.1558
payload = 1.3301	payload = 2.66	payload = 4.5488
Ratio = 32688	Ratio =65568	Ratio = 111792
Image name = 253036	Image name = 253036	Image name = 253036
PSNR = 48.1633	PSNR = 44.5028	PSNR = 43.3270
payload = 1.3339	payload = 2.6679	payload = 3.3339
Ratio = 32784	Ratio = 65568	Ratio = 81936

Figure 5.8: Experimental results of the proposed T1FLS with different edge detection techniques using BSD300 dataset image[44]**.**

From Figure 5.8, the best PSNR value obtained by the canny edge detector is 48.4630 when the capacity reached the value of 1.3301 bpp, and the worst PSNR value obtained is 40.7058, when the capacity reached the value of 3.6337 bpp.

For the Sobel edge detector, the best PSNR value obtained is 50.1927 when the capacity reached the value of 1.3339 bpp, and the worst PSNR value obtained is 40.7058, when the capacity reached the value of 3.6337 bpp.

The best PSNR value obtained by the proposed T1FLS is 48.1633 when the capacity reached the value of 1.3339 bpp, and the worst PSNR obtained is 40.1558, when the capacity reached the value of 4.5488 bpp.

The results of the experiments indicate that the proposed scheme has a high number of detecting edge pixels, which means embedding more data without seriously influencing the quality of the entire image as seen by HVS and PSNR. So, it can be said the proposed scheme satisfies the two requirements of capacity and imperceptibility in the Magic Triangle Visual Model.

Table 5.2 : Experimental results of the proposed T1FLS comparison with previous studies on 'Lena' image size of 128×128[44]**.**

J. Bai et al's Fuzzy edge detector [14]						**Proposed T1FLS**					
Canny		**Sobel**		**Fuzzy**		**Canny**		**Sobel**		**Fuzzy**	
PSNR	**Payload**	**PSNR**	**Payload**	**PSNR**	**Payload**	**PSNR**	**Payload**	**PSNR**	**Payload**	**PSNR**	**Payload**
48.927	**1.111**	**50.723**	**1.048**	**47.003**	**1.793**	**49.0110**	**1.1255**	**50.2528**	**1.1223**	**47.9370**	**1.2223**
39.256	**2.222**	**42.339**	**2.096**	**34.554**	**3.586**	**45.3841**	**2.2445**	**46.9834**	**2.2510**	**42.2255**	**3.515**
33.176	**3.222**	**37.397**	**3.096**	**26.308**	**4.586**	**44.0979**	**3.0964**	**45.4577**	**3.0964**	**41.5490**	**4.0776**

Table 5.2 demonstrates a comparison between the proposed T1FLS, with the previous scheme in[14]in terms of payload and PSNR, on ‘Lena’ image, with the size of 128×128, using various edge detector techniques. Through this comparison, it can be noticed that the PSNR results achieved by the proposed T1FLS scheme indicate that the stego image quality is better than the previous scheme and that mean is very close to the cover image.

Was selected the scheme in[14]to compare with our proposed T1FLS scheme, because it outperforms all the previous studies, in terms of payload and PSNR.

The main reason the superiority of the proposed T1FLS scheme is that it used the RGB color image, which gave us a large domain amounting reached to 24bits for each pixel. While previous schemes used grayscale images, which means that each pixel has only 8bits that can be used for embedded.

This means that the embedding space in the proposed T1FLS scheme is three times more than the other schemes, and therefore improves image quality.

5.7 Summary

In this chapter, we presented a new LSB steganography method, which uses a hybrid edge detector to make disclosing the existence of a secret message a hard operation. The proposed system processes the image in two phases (called fuzzy phase and embed phase). In the fuzzy phase based on the Gradient Approach and Type-1 Fuzzy Logic System, the edge detector is calculated. In the embedding phase, exploiting the edge image that has been obtained from the previous phase in embedding more secret bits in edge pixels than in non-edge pixels. The proposed system is developed into two sides, the sender's side that deals with the embedding process, and the receiver's side that deals with extraction processes.

Experimental results are performed to demonstrate the performance of the proposed method by applying the proposed T1FLS on six 128×192 RGB color images from the BSD300 dataset.The PSNR and HVS have been used to measure the quality of the stego image in each experiment.

We have conducted comparisons between the proposed T1FLS, with the previous scheme in[14] in terms of payload and PSNR, on ‘Lena’ image, with the size of 128×128, using various edge detectors techniques. Through this comparison, it can be noticed that the PSNR results achieved by the proposed T1FLS scheme indicate that the stego image quality is better than the previous scheme.

CHAPTER SIX THE PROPOSED ENHANCED EDGE DETECTOR BASED ON THE GRADIENT APPROACH AND TYPE-2 FUZZY LOGIC SYSTEM

6.1 Introduction

In this chapter, we aim to improve the LSB method proposed in the previous chapter, which uses a hybrid edge detector to make disclosing the existence of a secret message a hard operation.

6.2 The Proposed Type-2 Fuzzy Logic Embed System

Figure 6.1 shows the proposed fuzzy logic embedding system. In the fuzzy phase, the edge detector is calculated based on the Gradient Approach and Type-2 Fuzzy Logic System. In the embedding phase, exploiting the edge image that has been obtained from the previous phase in embedding more secret bits in edge pixels than in non-edge pixels.

The proposed system is developed into two sides, the sender's side that treats with the embedding process, and the receiver's side that treats with extraction processes.

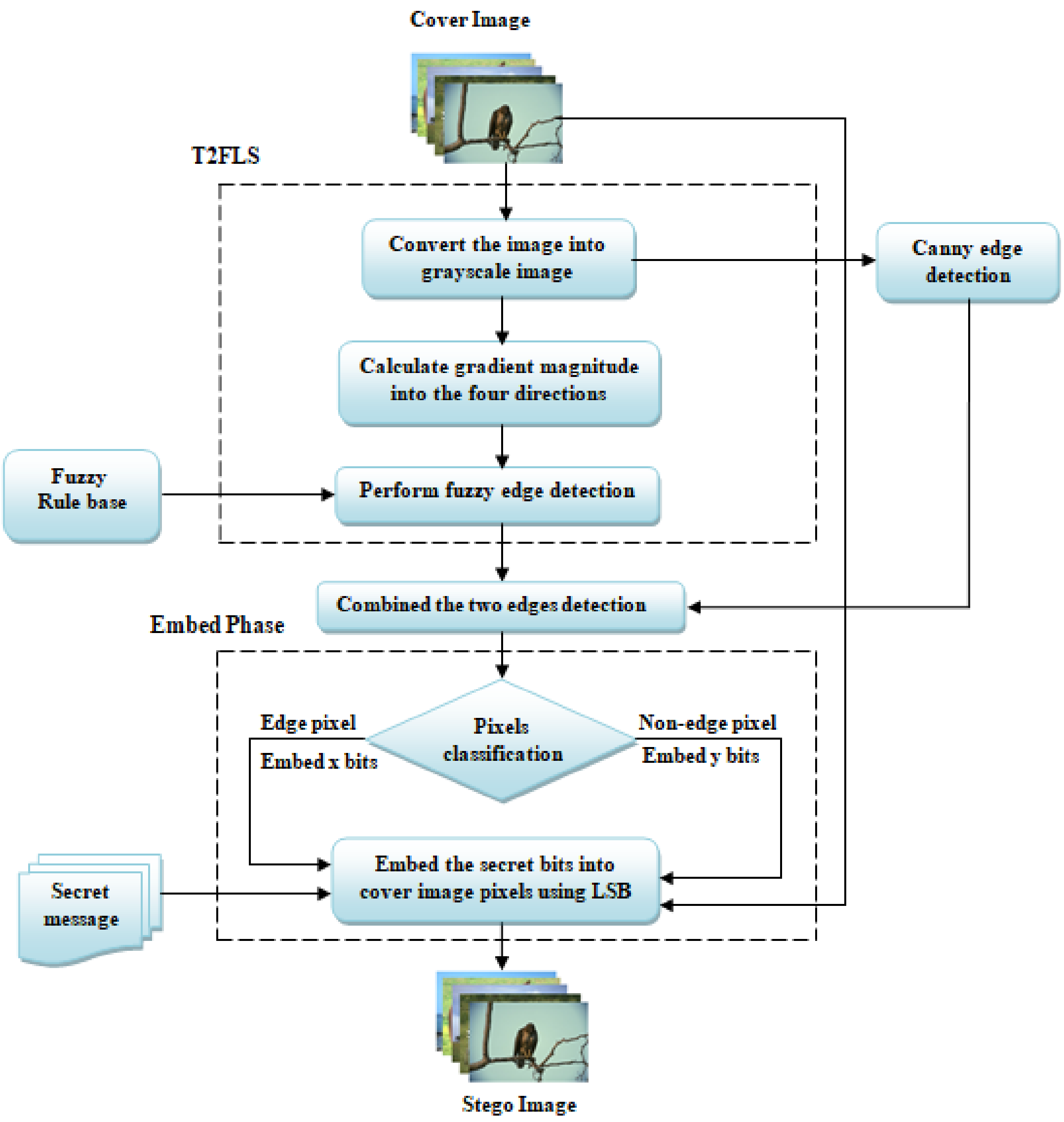

Cover Image
T2FLS
Convert the image into grayscale image
Canny edge detection
Calculate gradient magnitude into the four directions
Fuzzy Rule base
Perform fuzzy edge detection
Combined the two edges detection
Embed Phase
Edge pixel
Embed x bits
Pixels classification
Non-edge pixel
Embed y bits
Secret message
Embed the secret bits into cover image pixels using LSB
Stego Image

Figure 6.1: proposed Type-2 Fuzzy logic Embedding System

6.2.1 FLS Phase

6.2.1.1 Pre-processing Operation

Some processes are done prior to initiating the embedding operation. On the sender's side, the cover image will be converted into grayscale images, pass a copy of this grayscale image to the canny edge detector and another copy to the proposed gradient type-2 fuzzy edge detector algorithm.

6.2.1.2 Calculate Gradient T2FLS Edge Detection

This section presents the fuzzy logic methodology for edge detection, using gradient magnitude consists of using Eq (2-1) to obtain the gradients in the four directions (D1, D2, D3, D4) and employ them as inputs to a fuzzy inference system (FIS), instead of Eq. (2-2).

In this study, the T2FLS has been used a singleton Mamdani type, which was designed to contain four inputs (D1, D2, D3, and D4), and only one output. The inputs and outputs are fuzzified using Gaussian membership functions with uncertain mean; each input has three linguistic values: low, medium, and high to determine the grade to which the evaluated gradient corresponds, to be the output edge. Each output has two linguistic values: edge and background to produce the gradient magnitude T2FLS edge detector. The T2FIS Gaussian membership function is illustrated in Figure 6.1.

The parameters required for the T2FIS MFs are expressed in Equation (6-1), and these are calculated depending on the image gradient values, i.e., considering the image of Figure 4.1(lina), the parameters are obtained with Equations (6-2) to (3-6) [43][44][37].

$$\tilde{\mu}(x,u) = \text{igausmtype2}(x,[\sigma, m1, m2]) \quad (6\text{-}1)$$

$$\text{low} = \text{mi n}(\text{D}_\text{i}) \quad (6\text{-}2)$$

$$\text{high} = \text{ma x}(\text{D}_\text{i}) \quad (6\text{-}3)$$

$$\text{medium} = \text{low} + (\text{high} - \text{low})/\ 2 \quad (6\text{-}4)$$

$$\sigma = \text{high}/8$$

$$\mathrm{m1} \; = \; \mathrm{high} \tag{6-5}$$

$$\mathrm{m2} = \mathrm{m1} + \left(\mathrm{m1} *(\mathrm{FOU}), \mathrm{whereFOUisin}(0,1)\right) \tag{6-6}$$

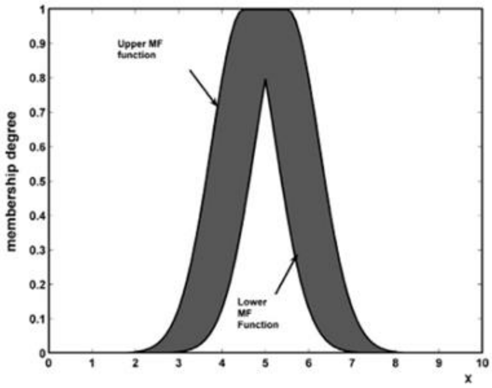


Figure 6.2: Interval Type 2 Fuzzy Set [42]**.**

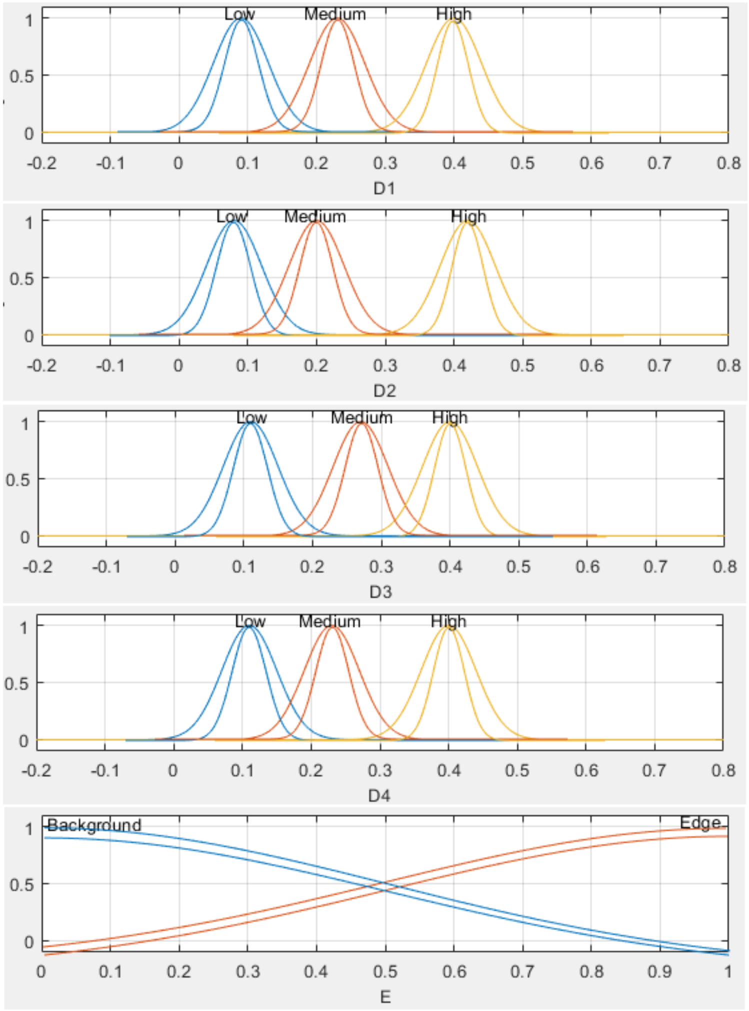


Figure 6.3: T2FIS memberships function for the inputs (D1, D2, D3 and D4) and the output E[42].

6.2.2 The Combined Edge Detections

After that, we will combine the two edge detectors (canny and T2FLS edge detector) to have a new hybrid edge image. The hybrid edge image, the cover image, and the secret message will be the inputs of the (embedding algorithm) LSB substitution algorithm. On the receiver's side, only the stego image will be the input of extraction algorithm to get the secret message back again.

6.2.3 Embed Phase

6.2.3.1 Pixels Classifications and Embed Operation

The embedding operation is responsible for hiding the secret message into the cover image file, using the proposed LSB method that uses the spatial domain of the RGB color image. On the sender's side, the secret message will be embedded into the cover image file and obtained the stego image file as output.

The operation of embedding a secret message in a cover image depends on the proposed hybrid edge image obtained from the combined edge detection step. The number of bits that should be embedded in each pixel is determined by the category of the pixel in the cover image. We utilize two parameters: x=9 and y=3. If the pixel is an edge pixel, the number of secret bits to be embedded will be x bits, and if the pixel is a non-edge pixel, the number of secret bits to be embedded will be y bits. The first bit in the red color of the cover image will be 0 or 1 to indicate that it is a non-edge or edge pixel respectively. Pixels are embedded by the M-LSB substitution, where the value M equals either x or y, which is decided by the edge information.

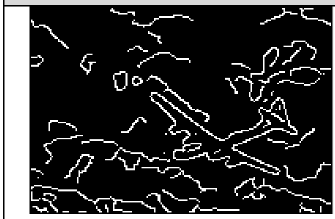
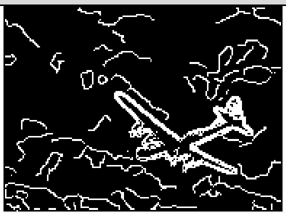
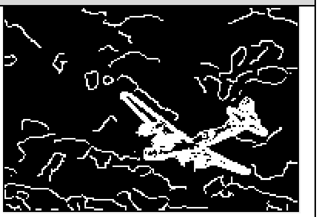
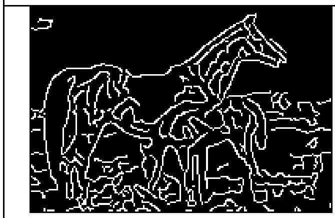
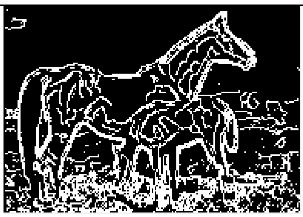
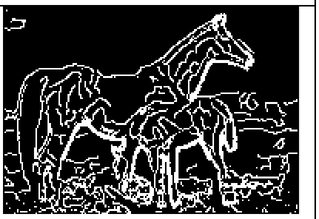
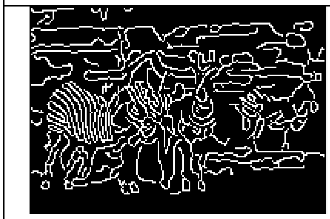
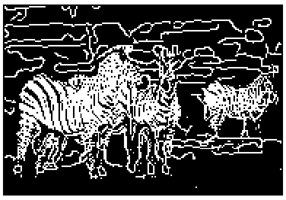
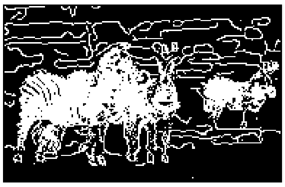
6.3 Experimental Results for T2FLS Edge Detector

In this section, to conduct our results three experiments will be done on different image datasets depending on image size. In the first experiment, the proposed T2FLS will apply to six 128×192 RGB color images from the BSD300 image. In the second experiment, the proposed T2FLS will implement on six 256×256 RGB color images from the USC-SIPI image dataset. And in the third experiment, the proposed T2FLS will implement on six 512×512 RGB color images from the CSIQ image dataset. In each experiment, the performance of the stego image is measured by using the PSNR and HVS.

6.3.1 The First T2FLS Experiment

In the first experiment, the implementation details of the proposed T2FLS method will be presented and discussed by applying different edge detectors, which are shown in Figure 6.4. This experiment has been applied on six 128×192 RGB colorimages from the BSD300 dataset as shown in Figure 4.2.

In Figure 6.3, the proposed T2FLS edge detector was compared with the T1FLS and canny edge detector based on the number of edge pixels that were detected.

Canny	Hala et al's T1FS	Our proposed T2FS
		
2008	2700	3037
		
3117	5717	4326
		
3874	6467	7250

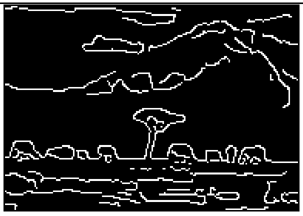
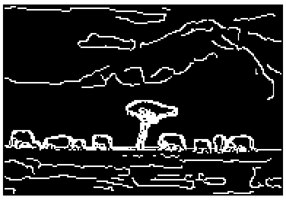
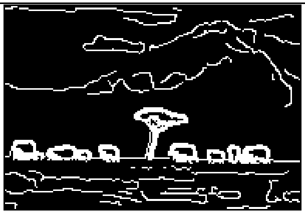
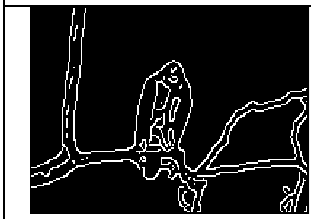
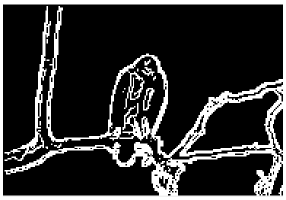
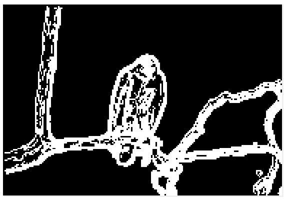
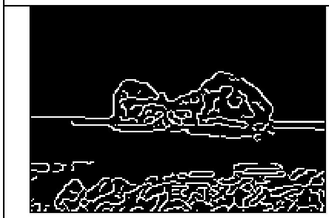
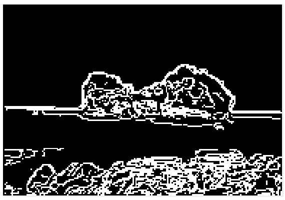
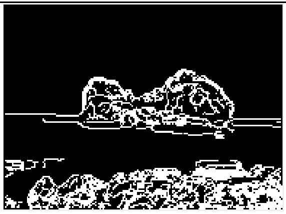
		
1995	2600	2735
		
1624	3432	4568
		
2203	3867	4056

Figure 6.4: The number of edge pixels detected by Canny, Hala et al.'s T1FLS[44]**, and T2FLS edge detector**[42]**.**

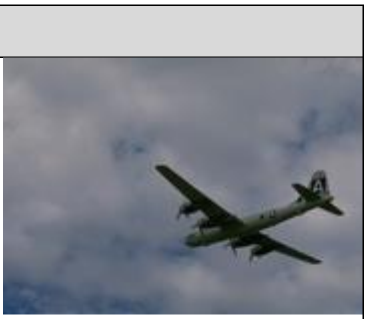
Canny edge detection		
		
Image name = 3096	Image name = 3096	Image name = 3096
PSNR = 48.4630 dB	PSNR = 45.0340 dB	PSNR = 44.5054 dB
payload = 1.3301 bpp	payload = 2.6680 bpp	payload = 3.0010 bpp
Ratio = 32688 bits	Ratio = 65568 bits	Ratio = 73752 bits

Image name = 113044	Image name = 113044	Image name = 113044
PSNR = 48.1963	PSNR = 44.2973	PSNR = 42.7058
payload = 1.3339	payload = 2.6679	payload = 3.6337
Ratio = 32784	Ratio = 65568	Ratio = 89304
Hala et al.' s T1FLS edge detector [6]		
Image name = 253027	Image name = 253027	Image name = 253027
PSNR = 46.4572	PSNR = 42.2049	PSNR = 40.1558
payload = 1.3301	payload = 2.66	payload = 4.5488
Ratio = 32688	Ratio =65568	Ratio = 111792
Image name = 253036	Image name = 253036	Image name = 253036
PSNR = 48.1633	PSNR = 44.5028	PSNR = 43.3270
payload = 1.3339	payload = 2.6679	payload = 3.3339
Ratio = 32784	Ratio = 65568	Ratio = 81936
Proposed T2FLS edge detector		

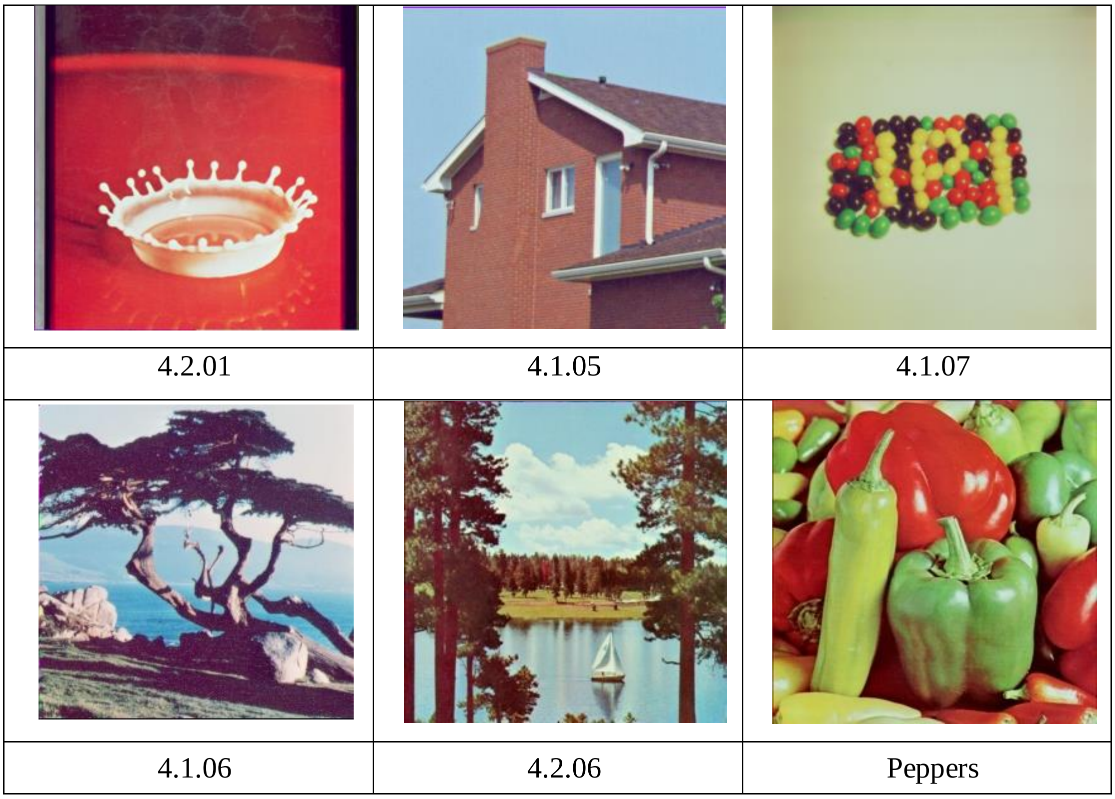
Image name = 42049	Image name = 42049	Image name = 42049
PSNR = 48.3825	PSNR = 42.9341	PSNR = 41.4912
payload = 1.3301	payload = 2.9951	payload = 4.0049
Ratio = 32688	Ratio = 73608	Ratio = 98424
Image name = 249061	Image name = 249061	Image name = 249061
PSNR = 49.4067	PSNR = 44.8596	PSNR = 44.0657
payload = 1.3301	payload = 2.6680	payload = 3.0010 bpp
Ratio = 32688	Ratio = 65568	Ratio = 73752

Figure 6.5: Experimental results of the proposed T2FLS edge detector comparison with Canny and Hala et al.'s T1FLS[44] **edge detection using BSD300 dataset image**[42]**.**

From Figure 6.4, the best PSNR value obtained by the canny edge detector is 48.4630 when the capacity reached the value of 1.3301 bpp, and the worst PSNR obtained is 40.7058, when the capacity reached the value of 3.6337 bpp. For Hala's scheme, the best PSNR obtained is 48.1633 when the capacity reached the value of 1.3339 bpp, and the worst PSNR obtained is 40.1558, when the capacity reached the value of 4.5488 bpp. The best PSNR obtained by the proposed T2FLS scheme is 49.4067 when the capacity reached the value of 1.3301 bpp, and the worst PSNR obtained is 41.4912, when the capacity reached the value of 4.0049 bpp.

6.3.2 The Second T2FLS Experiment

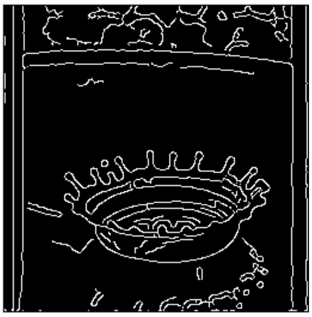
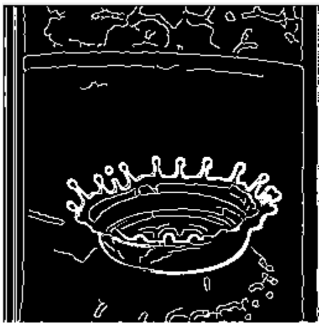
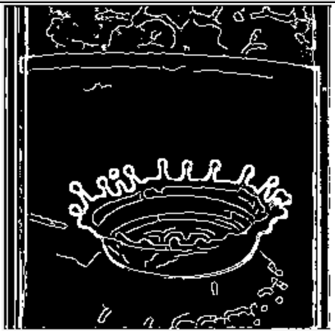
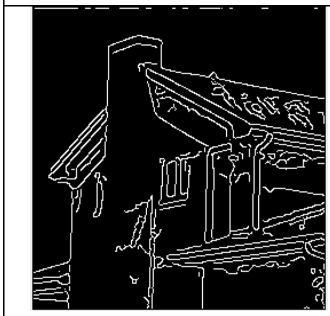
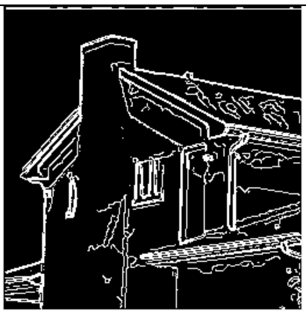
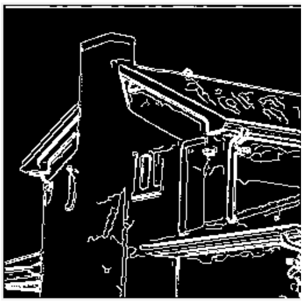
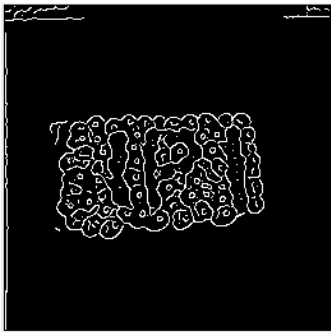
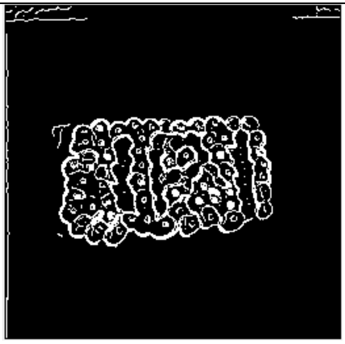
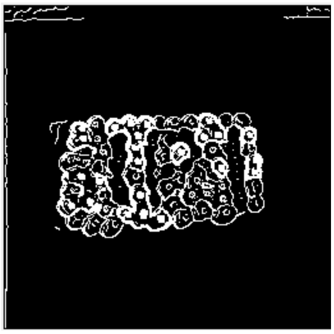
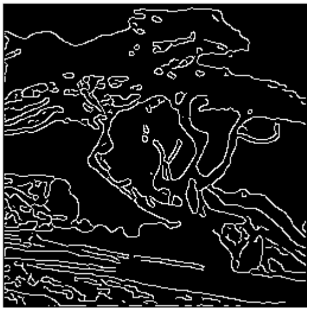
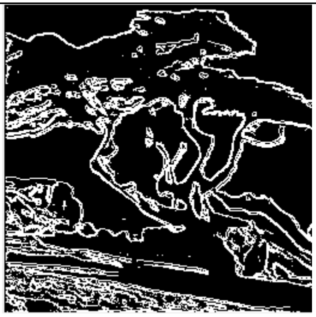
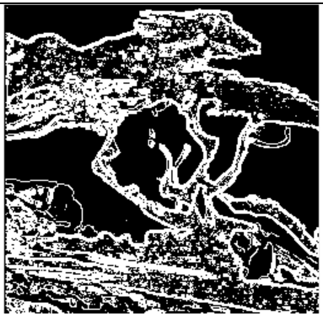
In the second experiment, the implementation results of the proposed T2FLS method will be presented and discussed by applying different edge detectors, which are shown in Figure 6.7. The experiments were applied on six 256×256 RGB color images from the USC-SIPI image dataset which are shown in Figure 6.5. Figure 6.6 illustrates the proposed T2FLS edge detector compared with T1FLS and canny edge detectors. From this comparison, it can seen that the proposed T2FLS edge detector has the largest number of edge pixels.



4.2.01 4.1.05 4.1.07

4.1.06 4.2.06 Peppers

Figure 6.6: Six 256×256 images from USC-SIPI image dataset.

Canny	**Hala et al's T1FS**	**Our proposed T2FS**
		
4799	6755	8237
		
5098	8110	8422
		
3139	5104	5545
		
6448	13122	21486

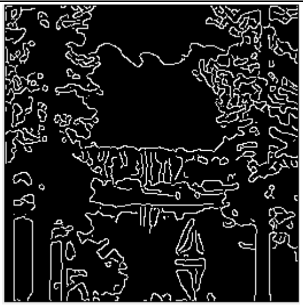
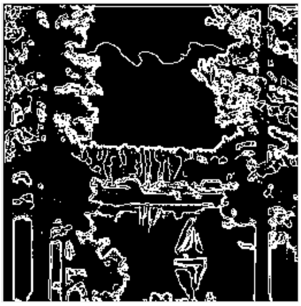
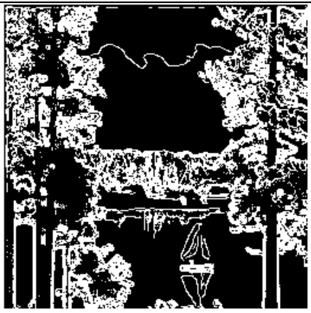
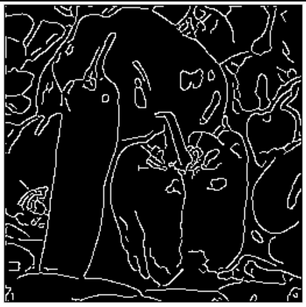
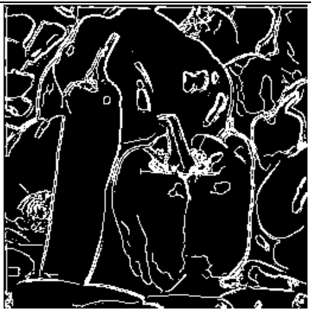
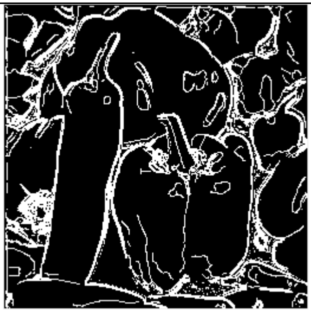
		
7054	13174	20242
		
5753	9852	11765

Figure 6.7: The number of edge pixels detected by Canny, Hala et al.'s T1FLS, and T2FLS edge detector.

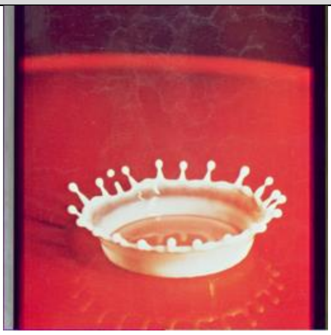
Canny edge detection		
		
Image name = 4.2.01	Image name = 4.2.01	Image name = 4.2.01
PSNR = 49.0505 dB	PSNR = 48.5700 dB	PSNR = 47.8830 dB
payload = 1.00 bpp	payload = 1.25 bpp	payload = 1.54 bpp
Ratio = 65536 bits(8k)	Ratio = 81920 bits(10k)	Ratio = 100762bits(12.3k)

Image name = 4.1.05	Image name = 4.1.05	Image name = 4.1.05
PSNR = 50.3272 dB	PSNR = 48.9768 dB	PSNR = 47.6305 dB
payload = 1.00 bpp	payload = 1.25 bpp	payload = 1.58 bpp
Ratio = 65536 bits(8k)	Ratio = 81920 bits(10k)	Ratio = 103219bits(12.6k)
Hala et al.’ s T1FLS edge detector [6]		
Image name = 4.1.07	Image name = 4.1.07	Image name = 4.1.07
PSNR = 51.5831 dB	PSNR = 49.3756 dB	PSNR = 47.6806 dB
payload = 1.00 bpp	payload = 1.25 bpp	payload = 1.54 bpp
Ratio = 65536 bits(8k)	Ratio = 81920 bits(10k)	Ratio = 100762bits(12.3k)

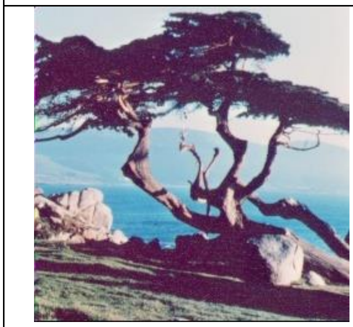



Image name = 4.1.06	Image name = 4.1.06	Image name = 4.1.06
PSNR = 48.6686 dB	PSNR = 46.0894 dB	PSNR = 44.3092 dB
payload = 1.00 bpp	payload = 1.50 bpp	payload = 2.25 bpp
Ratio = 65536 bits(8k)	Ratio = 98304 bits(12k)	Ratio = 147456 bits(18k)
Proposed T2FLS edge detector		

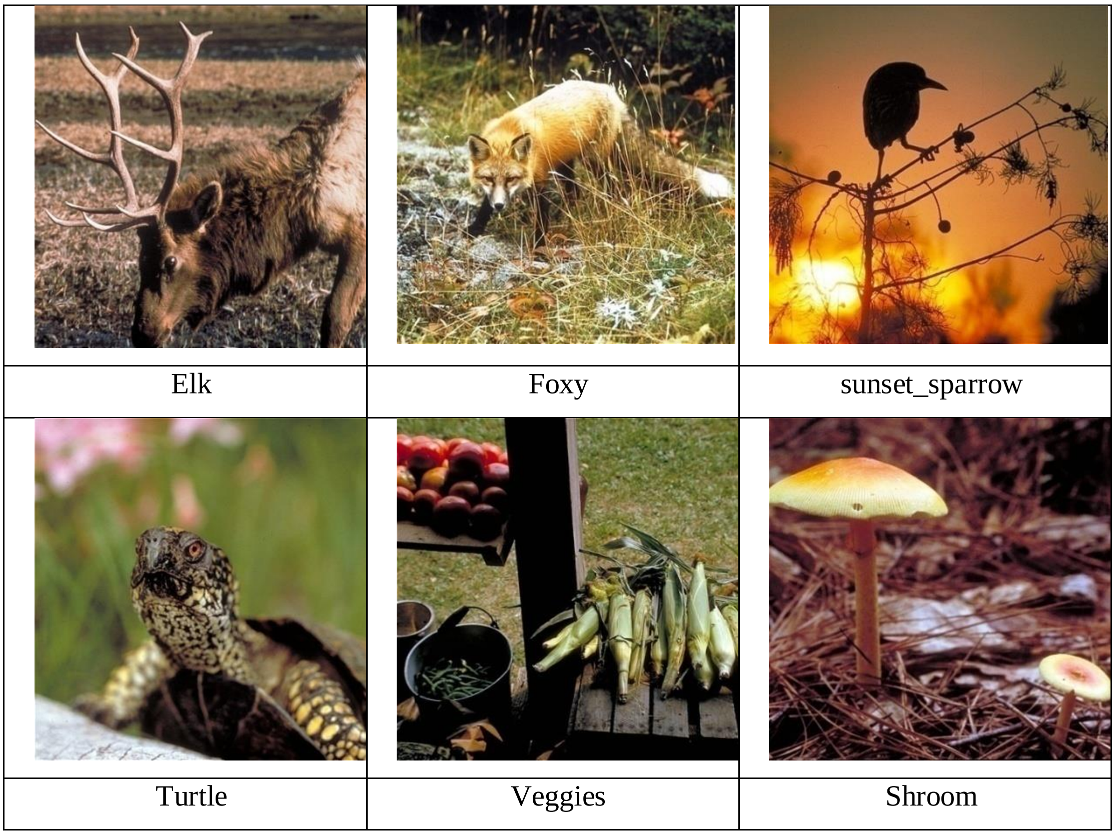
Image name = 4.2.06	Image name = 4.2.06	Image name = 4.2.06
PSNR = 46.5208 dB	PSNR = 42.8142 dB	PSNR = 41.8721 dB
payload = 1.00 bpp	payload = 2.25 bpp	payload = 2.75 bpp
Ratio = 65536 bits(8k)	Ratio = 147456 bits(18k)	Ratio = 180224 bits(22k)
Image name = Peppers	Image name = Peppers	Image name = Peppers
PSNR = 47.4986 dB	PSNR = 45.6524 dB	PSNR = 44.3439 dB
payload = 1.00 bpp	payload = 1.50 bpp	payload = 2.00 bpp
Ratio = 65536 bits(8k)	Ratio = 98304 bits(12k)	Ratio = 131072 bits(16k)

Figure 6.8: Experimental results of the proposed T2FLS edge detector comparison with Canny and Hala et al.'s T1FLS edge detection using USC-SIPI dataset image.

From Figure 6.7, the best PSNR value obtained by the canny edge detector is 51.5831 when the capacity reached the value of 1.00 bpp, and the worst PSNR obtained is 47.6305, when the capacity reached the value of 1.58 bpp. For Hala's scheme, the best PSNR obtained is 51.5831 when the capacity reached the value of 1.00 bpp, and the worst PSNR obtained is 44.3092, when the capacity reached the value of 1.25 bpp. The best PSNR obtained by the proposed T2FLS scheme is 47.4986 when the capacity reached the value of 1.00 bpp, and the worst PSNR obtained is 41.8721, when the capacity reached the value of 2.75 bpp.

6.3.3 The Third T2FLS Experiment

In the third experiment, the implementation results of the proposed T2FLS method will be presented and discussed by applying different edge detectors, which are shown in Figure 6.10. The experiments were applied on six 512×512 RGB color images from the CSIQ image dataset which are shown in Figure 6.8. Figure 6.9 illustrates the proposed T2FLS edge detector compared with T1FLS and canny edge detectors. From this comparison, it can be seen that the proposed T2FLS edge detector has a larger number of edge pixels.



Elk

Foxy

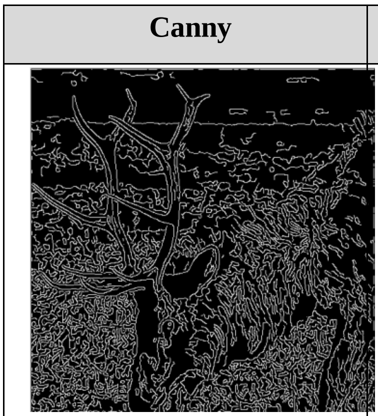
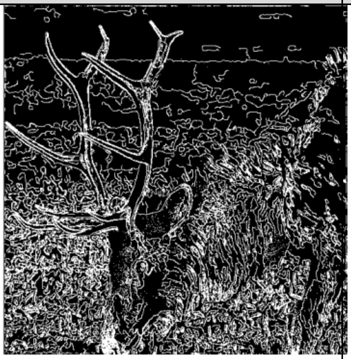
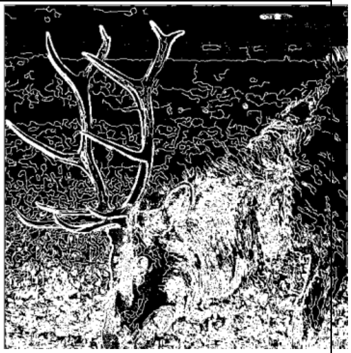
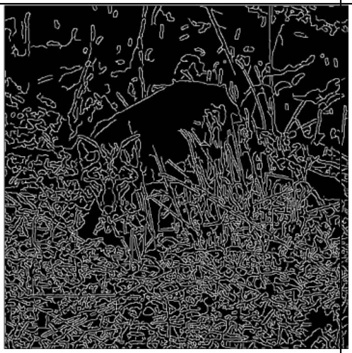
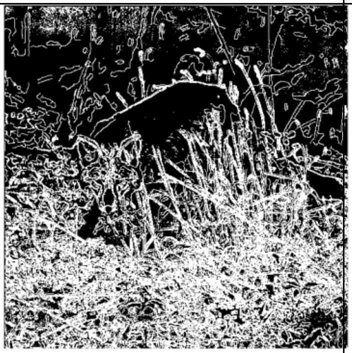
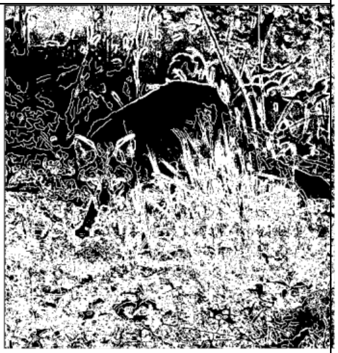
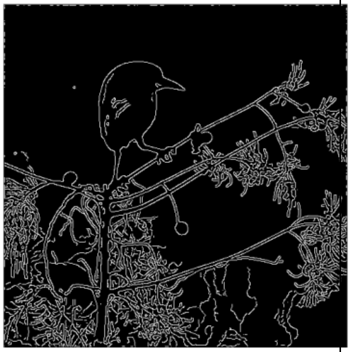
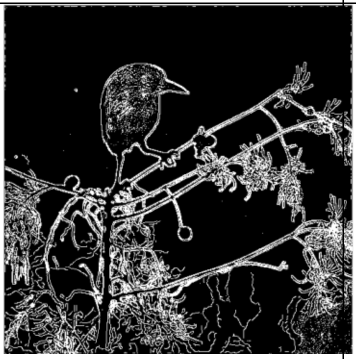
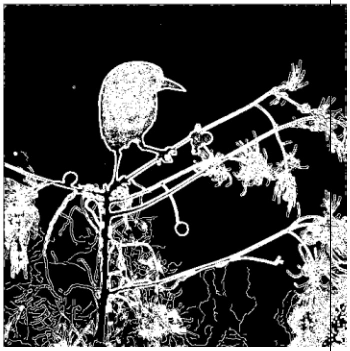
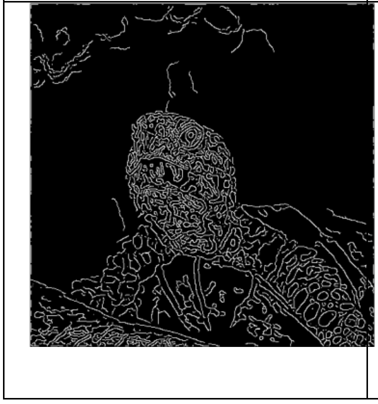
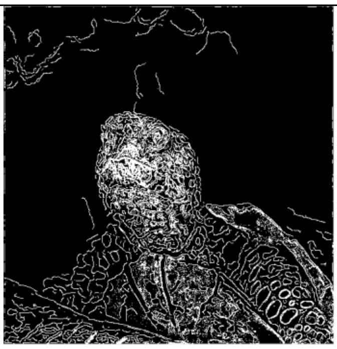
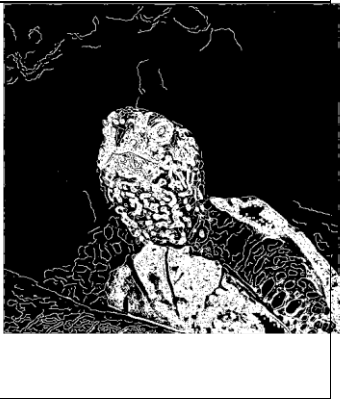
sunset_sparrow

Turtle

Veggies

Shroom

Figure 6.9: Six 512×512 images from the CSIQ image dataset.

Canny	**Hala et al's T1FS**	**Our proposed T2FS**
		
39372	60484	90674
		
47370	105618	133675
		
24063	41678	62602
		

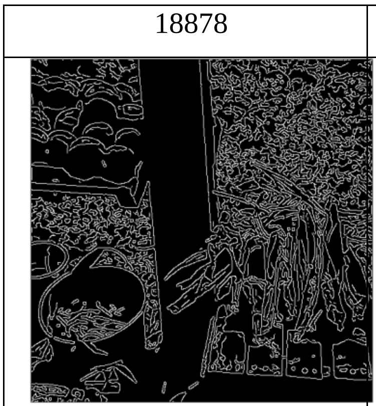
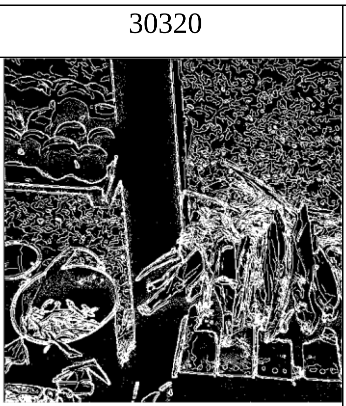
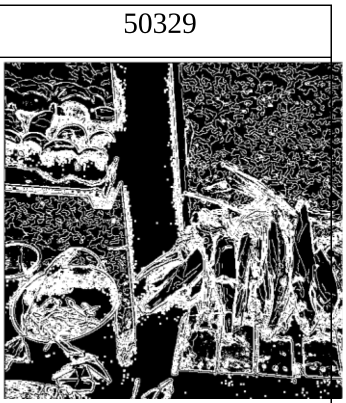
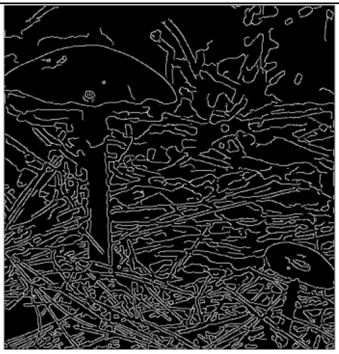
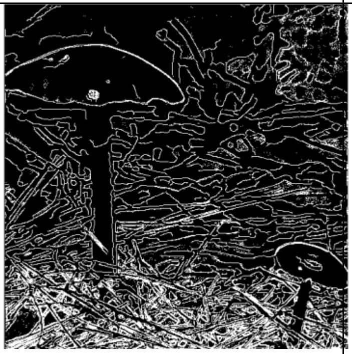
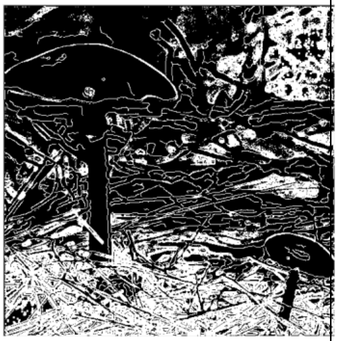
18878	30320	50329
		
35133	69080	93616
		
30940	48958	86688

Figure 6.10: The number of edge pixels detected by Canny, Hala et al.'s T1FLS, and T2FLS edge detector.

Canny edge detection		

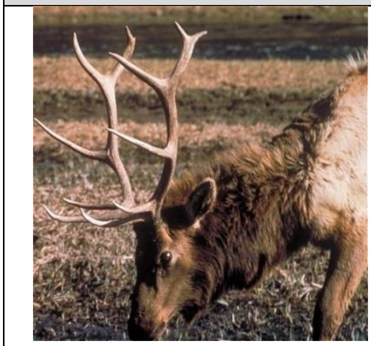



Image name = Elk	Image name = Elk	Image name = Elk
PSNR = 53.2966 dB	PSNR = 49.3432 dB	PSNR = 48.7404 dB
payload = 0.50 bpp	payload = 1.00 bpp	payload = 1.14 bpp
Ratio = 131072 bits(16k)	Ratio = 262144 bits(32k)	Ratio = 299008 bits(36.5k)

Image name = Fox	Image name = Fox	Image name = Fox
PSNR = 52.4022 dB	PSNR = 49.0770 dB	PSNR = 48.4617 dB
payload = 0.50 bpp	payload = 1.00 bpp	payload = 1.125 bpp
Ratio = 131072 bits(16k)	Ratio = 262144 bits(32k)	Ratio = 294912 bits(36k)
Hala et al.' s T1FLS edge detector [6]		
Image name = sunset_sparrow	Image name = sunset_sparrow	Image name = sunset_sparrow
PSNR = 54.8226 dB	PSNR = 50.2858 dB	PSNR = 49.4884 dB
payload = 0.50 bpp	payload = 1.00 bpp	payload = 1.125 bpp
Ratio = 131072 bits(16k)	Ratio = 262144 bits(32k)	Ratio = 294912 bits(36k)

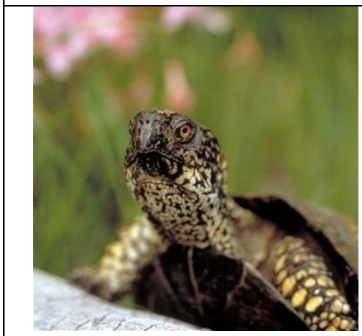



Image name = Turtle	Image name = Turtle	Image name = Turtle
PSNR = 53.3757 dB	PSNR = 51.6088 dB	PSNR = 51.2475 dB
payload = 0.56 bpp	payload = 0.88 bpp	payload = 0.97 bpp
Ratio = 147456 bits(18k)	Ratio = 229376 bits(28k)	Ratio = 253952 bits(31k)
Proposed T2FLS edge detector		

Image name = Veggies	Image name = Veggies	Image name = Veggies
PSNR = 47.0424 dB	PSNR = 43.3191 dB	PSNR = 42.4801 dB
payload = 0.94 bpp	payload = 2.00 bpp	payload = 2.37 bpp
Ratio = 245760 bits(30k)	Ratio = 524288 bits(64k)	Ratio = 620953 bits(75.8k)
Image name = Shroom	Image name = Shroom	Image name = Shroom
PSNR = 47.1916 dB	PSNR = 44.2803 dB	PSNR = 43.7916 dB
payload = 1.00 bpp	payload = 2.00 bpp	payload = 2.24 bpp
Ratio = 262144 bits(32k)	Ratio = 524288 bits(64k)	Ratio = 588185 bits(71.8k)

Figure 6.11: Experimental results of the proposed T2FLS edge detector comparison with Canny and Hala et al.'s T1FLS edge detection using CSIQ dataset image.

From Figure 6.10, the best PSNR value obtained by the canny edge detector is 53.2966 when the capacity reached the value of 0.50 bpp(16kb), and the worst PSNR obtained is 48.4617, when the capacity reached the value of 1.125 bpp(36kb). For Hala's scheme, the best PSNR obtained is 54.8226 when the capacity reached the value of 0.50 bpp (16kb), and the worst PSNR obtained is 49.4884, when the capacity reached the value of 1.25 bpp(36kb). The best PSNR obtained by the proposed T2FS scheme is

47.1916 when the capacity reached the value of 1.00 bpp(32kb), and the worst PSNR obtained is 42.4801, when the capacity reached the value of 2.37 bpp (75.8kb).

Table 6.1: Experimental results of the proposed steganography scheme apply on canny, proposed T1FLS, proposed T2FLS edge detectors with 128×192 images size.

Image size 128×192								
Canny			Proposed T1FLS			Proposed T2FLS		
PSNR	Payload	Ratio	PSNR	Payload	Ratio	PSNR	Payload	Ratio
48.4630	1.3301	32688(3.9kb)	48.1633	1.1633	32784(4kb)	48.3825	1.3301	32688(3.9kb)
45.0340	2.6680	65568(8kb)	44.5028	2.6680	65568(8kb)	42.9341	2.9951	73608(8.9kb)
44.5054	3.0010	73752(9kb)	43.3270	3.3339	81936(10kb)	41.4912	4.0049	98424(12kb)

Table 6.2: Experimental results of the proposed steganography scheme apply on canny, proposed T1FLS, proposed T2FLS edge detectors with 256×256 of images size.

Image size 256×256								
Canny			Proposed T1FLS			Proposed T2FLS		
PSNR	Payload	Ratio	PSNR	Payload	Ratio	PSNR	Payload	Ratio
50.3272	1.00	65536(8kb)	51.5831	1.00	65536(8kb)	47.4986	1.00	65536(8kb)
48.9768	1.25	81920(10kb)	49.3765	1.25	81920(10kb)	45.6524	1.50	98304(12kb)
47.6305	1.58	103219(12.6kb)	47.6806	1.54	100762(12.3kb)	44.3439	2.00	131072(16kb)

Table 6.3: Experimental results of the proposed steganography scheme apply on canny, proposed T1FLS, proposed T2FLS edge detectors with 512×512 of images size.

Image size 512×512								
Canny			Proposed T1FLS			Proposed T2FLS		
PSNR	Payload	Ratio	PSNR	Payload	Ratio	PSNR	Payload	Ratio
53.2966	0.50	131072(16kb)	54.8226	0.50	131072(16kb)	50.3618	0.50	131072(16kb)
49.3432	1.00	262144(32kb)	50.2858	1.00	262144(32kb)	47.1916	1.00	262144(32kb)
48.7404	1.14	299008(36.5kb)	49.4884	1.125	294912(36kb)	44.2803	2.00	524288(64kb)
						43.7916	2.24	588185(71.8kb)

Table 6.1, Table 6.2, and Table 6.3 list different values of payloads and PSNRs according to the change of image size, it can be noticed the PSNR values achieved by the 512×512 size on all experiments have large values of PSNR and large capacity.

Table 6.3 shows that the values obtained when using image size 512x512 achieve the best results when compared to other image sizes.

Table 6.4: Experimental results of the proposed scheme apply on payload values[1.1255, 3.0965, 4.4973, 4.5681] respectively and different values of FOU on 'Lena' image with the size of 128×128[42]**.**

FOU = 0.2	FOU = 0.4	FOU = 0.6	FOU = 0.8
PSNR	PSNR	PSNR	PSNR
48.0248	47.8706	47.8455	47.8557
42.4300	42.3782	42.3229	42.3904
40.7699	40.7028	40.6151	40.5643
-	40.5463	40.5610	40.5026

Table 6.4 lists the different payloads and PSNRs values according to the change of FOU values achieved by the proposed T2FLS edge detector on the 'Lena' image. This difference in values helps us to understand how the changes in the FOU affect the results of the proposed T2FLS edge detector. These changes can expound as follows: the different values of FOU represent various levels of uncertainty, and through this, we can compare the best PSNRs values and then determine the optimal FOU level for using in calculating the T2FLS edge detector. In this experiment, the best PSNR value was the one obtained with the FOU = 0.2.

Table 6.5: Experimental results of the proposed T2FLS comparison with Hala et al.' s T1FLS edge detector [44] and J. Bai et al's Fuzzy edge detector [14] on 'Lena' image with the size of 128×128.

J. Bai et al's Fuzzy edge detector [14]		Hala et al.' s T1FLS edge detector [44]		Proposed T2FLS edge detector	
PSNR	Payload	PSNR	Payload	PSNR	Payload
47.003	1.793	47.9370	1.223	48.0248	1.1255
34.554	3.586	42.2255	3.515	42.4300	3.0965
26.308	4.586	41.5490	4.0776	41.1389	4.0776
-	-	-	-	40.7664	4.5000

Table 6.5 presents the performance in terms of payload and PSNR with two previous schemes [14][44]on the 'Lena' image, with the size of 128×128. The results of comparison demonstrated that the PSNR values achieved by our proposed scheme increase the capacity and improve the quality of the stego image comparison with the previous schemes in [14][44].

6.4 Summary

In this chapter, we improved the image steganography method proposed in the previous chapter by using type-2 fuzzy logic systems and edge detection. The proposed system processes the image in two phases (called the fuzzy phase and embed phase). In the fuzzy phase based on the Gradient Approach and Type-2 Fuzzy Logic System, the edge detector is calculated. In the embedding phase, exploiting the edge image that has been obtained from the previous phase in embedding more secret bits in edge pixels than in non-edge pixels. The proposed system is developed into two sides, the sender's side that deals with the embedding process, and the receiver's side that deals with extraction processes.

To conduct our results three experiments were done on different image datasets depending on image size. In the first experiment, the proposed T2FLS applied to six 128×192 RGB color images from the BSD300 image. In the second experiment, the proposed T2FLS was implemented on six 256×256 RGB color images from the USC-SIPI image dataset. And in the third experiment, the proposed T2FLS was implement on six 512×512 RGB color images from the CSIQ image dataset. In each experiment, the performance of the stego image is measured by using the PSNR and HVS.

From these three experiments, and according to the change in image size, it can be noticed the PSNR values achieved by the 512×512 size on all experiments have large values of PSNR and large capacity.

We presented comparisons between the proposed T2FLS, with the previous scheme in[14][44]in terms of payload and PSNR, on ‘Lena’ image, with the size of 128×128, using various edge detectors techniques. Through this comparison, it can be noticed that the PSNR results achieved by the proposed T2FLS scheme indicate that the stego image quality is better than the previous schemes.

CHAPTER SEVEN CONCULASIONS, RECOMMENDATION AND FUTURE WORK

7.1 CONCLUSIONS

In this thesis, we have proposed an edge detection method based on the gradient technique combines type-1 fuzzy with the canny edge detector to increase the payload while maintaining quality of stego image. We used the (LSB) substitution technique to embed the secret data into the cover image.

The experimental results are performed to demonstrate the performance of the proposed method by applying the proposed T1FLS on six 128×192 RGB color images from the BSD300 dataset. The PSNR and HVS have been used to measure the quality of the stego image in each experiment. We presented comparisons between the proposed T1FLS, with the previous scheme in[14] in terms of payload and PSNR, on 'Lena' image, with the size of 128×128, using various edge detectors techniques.

To enhance our scheme, we improved the proposed edge detection by using gradient type-2 fuzzy logic edge detection due to their ability to handle the high level of uncertainty present in images. Based on this, the type-2 fuzzy system was employed for the detection of more edge pixels which exploit in hiding a large amount of data than non-edge pixels.

To conduct our results three experiments were done on different image datasets depending on image size. In the first experiment, the proposed T2FLS will apply to six 128×192 RGB color images from the BSD300 image. In the second experiment, the proposed T2FLS will implement on six 256×256 RGB color images from the USC-SIPI image dataset. And in the third experiment, the proposed T2FLS will implement on six 512×512 RGB color images from the CSIQ image dataset. In each experiment, the performance of the stego image is measured by using the PSNR and HVS.From these three experiments, and according to the change in image size, it can be noticed the PSNR values achieved by the 512×512 size on all experiments have large values of PSNR and large capacity.

We compared the proposed T2FLS with the previous scheme in [14][44] in terms of payload and PSNR, using various edge-detecting techniques on the ‘Lena’ image, with a size of 128×128. In this comparison, it can be seen that the PSNR results by the proposed T2FLS scheme indicate that the stego image quality is better than the previous scheme.

7.2 RECOMMEDITION AND FUTURE WORK

Further work can be done by making improvements in two directions. The first one is to do enhancement in the LSB image steganography algorithm by the increase the number of bits that carry the secret data.

The second one is to make development in the edge detector algorithm by using the general type-2 fuzzy logic or the genetic algorithm to improve the fuzzy edge.

REFERENCES

[1] M. S. Subhedar and V. H. Mankar, “ScienceDirect Current status and key issues in image steganography : A survey,” *Comput. Sci. Rev.*, pp. 1–19, 2014.

[2] 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.

[3] X. Y. Luo, D. S. Wang, P. Wang, and F. L. Liu, “A review on blind detection for image steganography,” *Signal Processing*, vol. 88, no. 9, pp. 2138–2157, 2008.

[4] F. Al-Shaarani and A. Gutub, “Securing matrix counting-based secret-sharing involving crypto steganography,” *J. King Saud Univ. - Comput. Inf. Sci.*, no. xxxx, 2021.

[5] L. M. Marvel, C. G. Boncelet, and C. T. Retter, “Spread spectrum image steganography,” *IEEE Trans. Image Process.*, vol. 8, no. 8, pp. 1075–1083, 1999.

[6] A. Ioannidou, S. T. Halkidis, and G. Stephanides, “A novel technique for image steganography based on a high payload method and edge detection,” *Expert Syst. Appl.*, vol. 39, no. 14, pp. 11517–11524, 2012.

[7] J. Watkins, “Steganography - Messages Hidden in Bits History of Steganography,” no. December, pp. 1–10, 2001.

[8] B. Dunbar, “Steganographic A detailed look at,” *Sans Inst.*, 2002.

[9] A. Z. Al-othmani, A. A. Manaf, and A. M. Zeki, “A Survey on Steganography Techniques in Real Time Audio Signals and Evaluation,” *Int. J. Comput. Sci. Issues*, vol. 9, no. 1, pp. 30–37, 2012.

[10] S. K. Bandyopadhyay, “A Proposed Method for Image Steganography,” *Res. Med. Eng. Sci.*, vol. 3, no. 4, pp. 2–3, 2018.

[11] T. Morkel, M. S. Olivier, and J. H. . Eloff, “an Overview of Image

Steganography," *Africa (Lond).*, vol. 83, no. July, pp. 51–107, 2005.

[12] P. Techscholar, V. Prof, P. Kumar, and V. K. Sharma, "Information Security Based on Steganography & Cryptography Techniques : A Review," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 4, no. 10, pp. 246–250, 2014.

[13] A. Sharma, R. Kumar, and V. Mansotra, "Proposed Stemming Algorithm for Hindi Information Retrieval," *Int. J. Innov. Res. Comput. Commun. Eng. (An ISO Certif. Organ.*, vol. 3297, no. 6, pp. 11449–11455, 2016.

[14] J. Bai, C. C. Chang, T. S. Nguyen, C. Zhu, and Y. Liu, "A high payload steganographic algorithm based on edge detection," *Displays*, vol. 46, no. December, pp. 42–51, 2017.

[15] H. W. Tseng and H. S. Leng, "High-payload block-based data hiding scheme using hybrid edge detector with minimal distortion," *IET Image Process.*, vol. 8, no. 11, pp. 647–654, 2014.

[16] W. J. Chen, C. C. Chang, and T. H. N. Le, "High payload steganography mechanism using hybrid edge detector," *Expert Syst. Appl.*, vol. 37, no. 4, pp. 3292–3301, 2010.

[17] A. Kaur, R. Dhir, and G. Sikka, "A New Image Steganography Based On First Component Alteration Technique," *J. Comput. Sci.*, vol. 6, no. 3, p. 4, 2010.

[18] S. Dhargupta, A. Chakraborty, S. K. Ghosal, S. Saha, and R. Sarkar, "Fuzzy edge detection based steganography using modified Gaussian distribution," *Multimed. Tools Appl.*, vol. 78, no. 13, pp. 17589–17606, 2019.

[19] S. Prasad and A. K. Pal, "An RGB colour image steganography scheme using overlapping block-based pixel-value differencing," *R. Soc. Open Sci.*, vol. 4, no. 4, 2017.

[20] M. Alkhudaydi and A. Gutub, "Securing Data via Cryptography and Arabic Text Steganography," *SN Comput. Sci.*, vol. 2, no. 1, 2021.

[21] T. Yuvaraja and R. S. Sabeenian, “Performance analysis of medical image security using steganography based on fuzzy logic,” *Cluster Comput.*, vol. 22, pp. 3285–3291, 2019.

[22] S. Kumar, A. Singh, and M. Kumar, “Information hiding with adaptive steganography based on novel fuzzy edge identification,” *Def. Technol.*, vol. 15, no. 2, pp. 162–169, 2019.

[23] “Image Steganography Based Audio Security System,” vol. 03, no. 10, pp. 1917–1921, 2014.

[24] D. Bloisi and L. Iocchi, “Image based steganography and cryptography,” *VISAPP 2007 - 2nd Int. Conf. Comput. Vis. Theory Appl. Proc.*, vol. IFP, no. IA/-, pp. 127–134, 2007.

[25] S. Gupta,Ankur Goyal and B. Bhushan, “Information Hiding Using Least Significant Bit Steganography and Cryptography,” *Int. J. Mod. Educ. Comput. Sci.*, vol. 4, no. 6, pp. 27–34, 2012.

[26] H. Khamis, “Studies on Image Steganography,” no. March, 2021.

[27] K. H. Jung and K. Y. Yoo, “High-capacity index based data hiding method,” *Multimed. Tools Appl.*, vol. 74, no. 6, pp. 2179–2193, 2015.

[28] S. Rustad, D. R. I. M. Setiadi, A. Syukur, and P. N. Andono, “Inverted LSB image steganography using adaptive pattern to improve imperceptibility,” *J. King Saud Univ. - Comput. Inf. Sci.*, no. xxxx, 2021.

[29] M. Hussain, A. W. A. Wahab, Y. I. Bin Idris, A. T. S. Ho, and K. H. Jung, “Image steganography in spatial domain: A survey,” *Signal Process. Image Commun.*, vol. 65, pp. 46–66, 2018.

[30] D. M. Abdullah *et al.*, “Secure Data Transfer over Internet Using Image Steganography: Review,” *Asian J. Res. Comput. Sci.*, no. July, pp. 33–52, 2021.

[31] Z. Zhu, T. Zhang, and B. Wan, “A special detector for the edge adaptive image

steganography based on LSB matching revisited," *IEEE Int. Conf. Control Autom. ICCA*, pp. 1363–1366, 2013.

[32] W. Luo, F. Huang, and J. Huang, "Edge adaptive image steganography based on lsb matching revisited," *IEEE Trans. Inf. Forensics Secur.*, vol. 5, no. 2, pp. 201–214, 2010.

[33] Y. K. Lee and L. H. Chen, "High capacity image steganographic model," *IEE Proc. Vision, Image Signal Process.*, vol. 147, no. 3, pp. 288–294, 2000.

[34] A. M. Fard, M. R. Akbarzadeh-T, and F. Varasteh-A, "A new genetic algorithm approach for secure JPEG steganography," *IEEE Int. Conf. Eng. Intell. Syst. ICEIS 2006*, no. January, 2006.

[35] M. Hagara and P. Kubinec, "About edge detection in digital images," *Radioengineering*, vol. 27, no. 4, pp. 919–929, 2018.

[36] R. C. Gonzalez and R. E. Woods, "Digital Image Processing Third Edition Pearson," pp. 1–076, 2008.

[37] N. Jain, S. Meshram, and S. Dubey, "Image Steganography Using LSB and Edge – Detection Technique," no. 3, pp. 217–222, 2012.

[38] E. B. Kumar, "Comparison and Evaluation of Edge Detection using Fuzzy Membership Functions," no. August, pp. 149–153, 2017.

[39] P. A. Khaire, "A Fuzzy Set Approach for Edge Detection," *Int. J. Image Process.*, no. 6, pp. 403–412, 2012.

[40] R. Lalchhanhima, D. Kandar, and B. Paul, *Performance Analysis of Fuzzy Logic-Based Edge Detection Technique*, vol. 462. Springer Singapore, 2018.

[41] H. S. Yusuf, "Towards Image Steganography Using Type-2 Fuzzy Logic and Edge Detection," *2018 10th Comput. Sci. Electron. Eng.*, pp. 75–78.

[42] H. S. Yusuf and H. Hagras, "A novel Payload Image Steganography Using Type-2 Fuzzy Logic and Edge Detection," vol. 9, no. 2, pp. 89–98, 2021.

[43] C. I. Gonzalez, P. Melin, and O. Castillo, “Edge detection method based on general type-2 fuzzy logic applied to color images,” *Inf.*, vol. 8, no. 3, pp. 1–15, 2017.

[44] H. S. Yusuf and H. Hagras, “High Payload Image Steganography Method Using Fuzzy Logic and Edge Detection,” vol. 8, no. 4, pp. 123–134, 2020.

[45] P. Melin, O. Mendoza, and O. Castillo, “An improved method for edge detection based on interval type-2 fuzzy logic,” *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8527–8535, 2010.

[46] L. A. Zadeh, I. Introduction, and U. S. Navy, “Fuzzy Sets * -,” vol. 353, pp. 338–353, 1965.

[47] S. K. Saeed, “A Fraud-Detection Fuzzy Logic Based System for the Sudanese Financial Sector,” *J. Eng. Comput. Sci.*, vol. 20, no. 1, pp. 17–30, 2019.

[48] H. Hagras, V. Callaghan, M. Colley, W. Park, and U. K. England, “A Fuzzy Genetic Based Embedded-Agent Approach to Learning & Control in Agricultural Autonomous Vehicles . A Fuzzy-Genetic Based Embedded-Agent Approach to Learning & Control in Agricultural Autonomous Vehicles The Computer Science Department , Malcolm Car,” no. October 2014, 1999.

[49] A. Salih and H. Hagras, “A Type-2 Fuzzy Logic Based System for Decision Support to Minimize Financial Default in the Sudanese Banking Sector,” vol. 20, no. 3, 2019.

[50] L. A. Zadeh, “The Concept of a Linguistic Variable and its Application to Approximate Reasoning-I,” vol. 249, 1975.

[51] J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems*. 2017.

[52] J. M. Mendel, “Fuzzy Logic Systems for Engineering : A futorial,” vol. 83, no. 9408047, pp. 345–377, 1995.

[53] S. N. Sivanandam, S. Sumathi, and S. N. Deepa, *Introduction to fuzzy logic using*

MATLAB. 2007.

[54] O. Castillo and P. Melin, “Introduction to type-2 fuzzy logic,” *Stud. Fuzziness Soft Comput.*, vol. 223, pp. 1–4, 2008.

[55] D. S. SLOAN, “ A Review of: ‘ FUZZY SET THEORY AND ITS APPLICATIONS ’ (Second Edition), by H.-J. Zimmermann. Kluwer Publishers, Boston, 1991. ,” *Int. J. Gen. Syst.*, vol. 21, no. 1, pp. 117–119, 1992.

[56] S. S. Jamsandekar and R. Mudholkar, “Self generated fuzzy membership function using ANN clustering technique,” *Int. J. Latest Trends Comput.*, vol. 6, no. 1, pp. 142–152, 2013.

[57] N. N. Karnik, J. M. Mendel, and Q. Liang, “Type-2 fuzzy logic systems,” *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 6, pp. 643–658, 1999.

[58] J. M. Mendel, “Feature Article Some Questions and Answers Type-2 Fuzzy Sets : Feature Article (cont .),” no. August, pp. 10–13, 2003.

[59] N. N. Karnik and J. M. Mendel, “Operations on type-2 fuzzy sets,” vol. 122, pp. 327–348, 2001.

[60] M. Mizumoto and K. Tanaka, “Some properties of fuzzy sets of type 2,” *Inf. Control*, vol. 31, no. 4, pp. 312–340, 1976.

[61] “Hani Hagras,” *IEEE Comput. Intell. Mag.*, no. February, pp. 30–43, 2007.

[62] L. Li, W. H. Lin, and H. Liu, “Type-2 fuzzy logic approach for short-term traffic forecasting,” *IEE Proc. Intell. Transp. Syst.*, vol. 153, no. 1, pp. 33–40, 2006.

[63] J. Andreu-perez, F. Cao, H. Hagras, and G. Yang, “A Self-Adaptive Online Brain Machine Interface of a Humanoid Robot through a General Type-2 Fuzzy Inference System,” vol. 16, no. 1, pp. 1–14, 2016.

[64] M. Antonelli, D. Bernardo, H. Hagras, and F. Marcelloni, “Multi - Objective Evolutionary Optimization of Type - 2 Fuzzy Rule - based Systems for Financial Data Classification,” vol. 25, no. 2, pp. 249–264, 2016.

[65] H. Hagras, M. Colley, V. Callaghan, and M. Carr-West, “Online learning and adaptation of autonomous mobile robots for sustainable agriculture,” *Auton. Robots*, vol. 13, no. 1, pp. 37–52, 2002.

[66] A. Starkey, H. Hagras, S. Shakya, and G. Owusu, “A multi-objective genetic type-2 fuzzy logic based system for mobile field workforce area optimization,” *Inf. Sci. (Ny).*, vol. 329, pp. 390–411, 2016.

[67] A. Sakalli, T. Kumbasar, E. Yesil, and H. Hagras, “Analysis of the performances of type-1, self-tuning type-1 and interval type-2 fuzzy PID controllers on the Magnetic Levitation system,” *IEEE Int. Conf. Fuzzy Syst.*, no. March 2016, pp. 1859–1866, 2014.

[68] H. A. Hagras, “A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots,” vol. 12, no. 4, pp. 524–539, 2004.

[69] C. Lynch and V. Callaghan, “Embedded Interval Type-2 Neuro-Fuzzy Speed Controller for Marine Diesel Engines,” vol. 270.

[70] S. K. Saeed and H. Hagras, “A Big Bang-Big Crunch Type-2 Fuzzy Logic Based System for Fraud-Detection: Case Study Balad Bank in Sudan,” *SUST J. Eng. Comput. Sci.*, vol. 21, no. 3, pp. 16–28, 2020.

[71] J. De Andrés, M. Landajo, and P. Lorca, “Forecasting business profitability by using classification techniques: A comparative analysis based on a Spanish case,” *Eur. J. Oper. Res.*, vol. 167, no. 2, pp. 518–542, 2005.

[72] S. Michael, D. Georgios, M. Nikolaos, and Z. Constantin, “A Fuzzy Knowledge-based Decision Aiding Method for the Assessment of Financial Risk: The Case of Corporate Bankruptcy Prediction,” *Eur. Symp. Intell. Tech.*, no. February, 1999.

[73] D. Kim and C. Kim, “Forecasting time series with genetic fuzzy predictor ensemble,” *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 4, pp. 523–535, 1997.

[74] C. S. Division and C. Sciences, “Fuzzy sets as a basis for a theory of possibility* L.A. zadeh,” vol. 1, no. 1, pp. 3–28, 1978.

[75] Q. Liang, N. N. Karnik, A. Member, and J. M. Mendel, “Connection Admission Control in ATM Networks Using Survey-Based Type-2 Fuzzy Logic Systems,” vol. 30, no. September 2000, 2017.

[76] J. M. Mendel and R. I. B. John, “Type-2 Fuzzy Sets Made Simple,” vol. 10, no. 2, pp. 117–127, 2002.

[77] A. Salih and H. Hagras, “A Big Bang – Big Crunch Optimized Type-2 Fuzzy Logic Based System for Default Prediction in Sudanese Banking Sector,” vol. 8, no. 5, pp. 14–22, 2020.

[78] D. Bernardo, H. Hagras, and E. Tsang, “A Genetic Type-2 Fuzzy Logic Based System for Financial Applications Modelling and Prediction,” no. September 2015, 2013.

[79] C. Wagner, “Towards the Wide Spread Use of Type-2 Fuzzy Logic Systems in Real World Applications,” no. October 2014, 2017.

[80] B. Yao, H. Hagras, and M. J. Alhaddad, “A Fuzzy Logic-Based System for the Automation of Human Behavior Recognition Using Machine Vision in Intelligent Environments A fuzzy logic-based system for the automation of human behavior recognition using machine vision in intelligent environments,” vol. 19, no. April 2016, pp. 499–506, 2014.

[81] A. Bilgin, H. Hagras, A. Malibari, M. J. Alhaddad, and D. Alghazzawi, “Towards a linear general type-2 fuzzy logic based approach for computing with words,” vol. 17, no. 12, pp. 2203–2222, 2013.

[82] F. Liu, “An efficient centroid type-reduction strategy for general type-2 fuzzy logic system,” *Inf. Sci. (Ny).*, vol. 178, no. 9, pp. 2224–2236, 2008.

[83] N. N. Karnik and J. M. Mendel, “Applications of type-2 fuzzy logic systems to forecasting of time-series,” *Inf. Sci. (Ny).*, vol. 120, no. 1, pp. 89–111, 1999.

[84] H. C. Wu, N. I. Wu, C. S. Tsai, and M. S. Hwang, “Image steganographic scheme based on pixel-value differencing and LSB replacement methods,” *IEE Proc.*

Vision, Image Signal Process., vol. 152, no. 5, pp. 611–615, 2005.

[85] S. Wang, B. Yang, and X. Niu, “A secure steganography method based on genetic algorithm,” *J. Inf. Hiding Multimed. Signal Process.*, vol. 1, no. 1, pp. 28–35, 2010.

[86] H. R. Kanan and B. Nazeri, “A novel image steganography scheme with high embedding capacity and tunable visual image quality based on a genetic algorithm,” *Expert Syst. Appl.*, vol. 41, no. 14, pp. 6123–6130, 2014.

[87] A. Conci, A. L. Brazil, S. B. L. Ferreira, and T. Machenry, “AES cryptography in color image steganography by genetic algorithms,” *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA*, vol. 2016-July, 2016.

[88] A. Khamrui and J. K. Mandal, “A Genetic Algorithm based Steganography Using Discrete Cosine Transformation (GASDCT),” *Procedia Technol.*, vol. 10, pp. 105–111, 2013.

[89] A. Rana, N. Sharma, and A. Kaur, “Image Steganography Method Based on Kohonen Neural Network,” vol. 2, no. 3, pp. 2234–2236, 2012.

[90] N. N. El-Emam and M. Al-Diabat, “A novel algorithm for colour image steganography using a new intelligent technique based on three phases,” *Appl. Soft Comput. J.*, vol. 37, pp. 830–846, 2015.