Coventry University

DOCTOR OF PHILOSOPHY

Development of a detection system for colour steganographic images based on extraction of colour gradient co-occurrence matrix features and histogram of difference image

AlJarf, Ahd Mohammad S

Award date:
2017

Awarding institution:
Coventry University

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of this thesis for personal non-commercial research or study
- This thesis cannot be reproduced or quoted extensively from without first obtaining permission from the copyright holder(s)
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Development of a Detection System for Colour Steganographic Images based on Extraction of Colour Gradient Co-occurrence Matrix Features and Histogram of Difference Image

By

Ahd Aljarf

PhD

10/2016

Development of a Detection System for Colour Steganographic Images based on Extraction of Colour Gradient Co-occurrence Matrix Features and Histogram of Difference Image

By

Ahd Aljarf

10/2016

A thesis submitted in partial fulfilment of the University's requirements for the Degree of Doctor of Philosophy

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Certificate of Ethical Approval

Applicant:

Ahd AlJarf

Project Title:

Develop a Detection System for Colour Images using the Colour Gradient Co-occurrence Matrix (CGCM) and Histogram of Difference Image Features

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:

28 September 2016

Project Reference Number:

P45604

Acknowledgment

I would like to express my special appreciation and thanks to my family. Words cannot express how grateful I am to my father Prof. Mohammad Aljarf and my mother Karam Arshad for all of the sacrifices they've made on my behalf, for all the support they gave me to start my PhD, and for their never-failing encouragement. Your prayers for me were what sustained me thus far.

My sincere thanks also goes to my brother Ahamd, who accompanied me during this long journey.

At the end I would like express appreciation to my beloved husband Shadi Badawood, who was always my support in the moments when there was no one to answer my queries.

I would also like to thank my friend Olfat Mirza, who was always there for me.

A special thanks to my director of study Dr. Saad Amin: you have been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a researcher. Your advice on both research as well as on my career have been priceless.

Publications and Achievements

Aljarf, Ahd, Amin, Saad, Filippas, John, and Shuttleworth, James. (2017). 'Detection System for Grey and Colour Images Based on Extracting Features of Difference Image and Renormalized Histogram'. Journal of Information Hiding and Multimedia Signal Processing, 8 (2)

Aljarf, Ahd., Amin, Saad., Filippas, John. and Shuttelworth, James., 2016, August. The Development of an Images Detection System Based on Extracting the Colour Gradient Co-occurrence Matrix Features. In Developments in eSystems Engineering (DeSE), 2016 9th International Conference on (pp. 260-267). IEEE.

Aljarf, Ahd, and Amin, Saad. (2015) 'Filtering and Reconstruction System for Gray Forensic Images. 'World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering 9.1: 20-25.

Aljarf, Ahd and Amin, Saad. (2015) 'Filtering and Reconstruction System for Grey-Level Forensic Images'. 17th International Conference on Image Processing (ICIP 2015) Zurich, Switzerland.

Aljarf, Ahd, Amin, Saad, Filippas, John, and Shuttleworth, James. (2013) 'Develop a Detection System for Grey and ColourStego Images'. International Journal of Modeling and Optimization, 3 (5), 458-461

A Aljarf, S Amin, (2013) 'Develop a Detection System for Grey and Colour Stego Images'. 3rd International Conference on Circuits, System and Simulation (ICCSS 2013) Barcelona, Spain.

Aljarf, Ahd, Saad Amin, and John Filippas. (2013) 'Creating Stego-Images through Hiding Single And Multiple Data Using Different Steganographic Tools'. 10th IASTED International Conference on Signal Processing, Pattern Recognition and Applications, 798

Awarded the title: ‘Coventry University Winner’, during the poster competition event in February 2014 at Warwick University. The competition was organised by the British Computer Society (BCS), Coventry university branch.

Awarded the title: ‘Coventry University Winner’, during the poster competition event in March 2013, which was organised by the British Computer Society (BCS), Coventry university branch.

Presented at the Research Student Symposium in April 2013, and awarded the second prize for the best oral presentations.

Received an award from the Saudi ambassador to UK in London for being titled as a distinguished student.

Table of Contents

List of Figures

List of Tables

Abbreviations

AUC: area under curve.

BMP: bitmap image file.

BPCS Steganography: bit-plane complexity segmentation.

CGCM: colour gradient co-occurrence matrix.

Clean image: any image that does not contain any hidden file.

DCT: discrete cosine transform.

DFT: Discrete Fourier Transform.

HAS: human auditory system.

JPG: joint photographic experts group.

LSB: least signifigant bit.

MLP: multi layer perceptron neural network.

PNG: portable networks graphics.

RBF: radial basis function.

RS Steganalysis: regular singular steganalysis.

SDS: Spatial domain.

Stego image: any image that contain a hidden file.

Stepwise DA: stepwise discriminant analysis classification method.

SVM: support vector machine.

TDS: Transform doman steganography.

Abstract

Steganography is the science of hiding information in some other medium. These media can be text, images, audio or video files. Steganographic analysis (steganalysis), on the other hand, is the science of detecting the existence of hidden information.

Many steganalysis methods have been introduced in the literature. These methods have been developed to combat specific steganography techniques and to detect data hidden in specific image formats. However, no single steganalysis method or tool can detect all types of steganography or support all available image formats.

One of the problems is there a need for more general system to cover different typies of image fromats and detecting wider range of stego images created by many steganography methods blindly. Blind steganalysis means detecting any stego image without knowing the type of the steganography method used or which type of file was embedded.

This thesis focuses on image steganlysis. A detection system is presented that combines three different steganalysis techniques. All technique address blind image steganalysis are rely on the extraction of selections of image features.

The first steganalysis technique presented here is based on extracting varieties of
CGCM. The CGCM takes into account information of both colour correlations and gradients among the pixels in an image.

The second steganaslysis technique works by extracting a number of histogram features. The features are extracted by exploiting the histogram of difference image, which is usually a generalised Gaussian distribution centered at 0. The histogram of difference image and the renormalized histogram are created for clean and stego images, thereby using the peak value and renormalized histogram as features for classification.

Finally, the tested CGCM features and histogram features were merged together to improve the performance of the system. Merging two different types of features allows taking advantages of the beneficial properties of each in order to increase the system ability in terms of detection.

A large image database was created to train and test the system. The database included colour and grey images in various formats using both lossless and lossy compression.

The proposed detection system was trained and tested to distinguish stego images from clean ones using the Discriminant Analysis (DA) classification method and Multilayer Perceptron

neural network (MLP). Stepwise Discriminate Analysis was applied in an attempt to find the best set of predictors. The Multilayer Perceptron (MLP) procedure was used to produce a predictive model for one or more dependent (target) variable based on the values of the predictor variables.

The experimental results prove that the proposed system possesses reliable detection ability and accuracy. The chosen classification methods show dissimilar performance in terms of classifying grey and colour images. The new system is a more generalized detector than previous systems, covering a wider variety of types of stego images, image formats and different hidden file sizes.

Chapter 1

Introduction

1.1. Background

Steganography, the science of embedding secret data in an appropriate cover object, is an important research issue in the computer security field (Anderson and Petitcolas, 1998).

Derived from the Greek words stegos meaning "cover" and grafia meaning "writing", steganography is defined as "covered writing" (Duric et al., 2004; Huayong et al., 2011).

The concept of steganography is very old. In the 5th century BCE, when Histaiacus wanted to send a secret message, he shaved a slave's head and tattooed the secret message on the slave's skull. When the slave's hair grew again, he was dispatched with the message. This is a classic example of a steganographic system that attains secrecy by encoding a secret message.

Images are widespread on the Internet and can be used as carrier objects without raising much suspicion. Therefore, they are the most commonly used cover to hide files.

Steganalytic systems are used to determine whether or not an image contains a hidden file by analysing various features of stego images (images containing hidden files) and of clean images (images containing no hidden files).

Nissar and Mir (2010) highlighted the importance of steganography and steganalysis as a key approach used in law enforcement, computer forensics, and cryptography, and pointed out its interest to the media. However, relatively few articles had appeared before the late 1990s.

Today, cyber-criminals use steganography for the distribution of illegal content such as numbers of a stolen credit cards, fake money orders, user names and passwords for web pages that have been breached and various types of citizen's databases including registry numbers, insurance numbers etc.

In addition, steganograohy is used for a variety of other applications, for example, copyright control of materials, smart IDs and tuning the robustness of image search engines (identification cards) where the data of individuals' are enclosed in their photographs. It is also used for covert communication using images, data integrity, fraud detection, self-correcting images, intelligent browsers, automatic copyright information and for viewing a movie in a given rated version (Cheddad et al. 2010).

Any file that contain hidden information is called a 'stego' file, and files that don't carry any hidden information are called 'clean files' or 'carrier files'.

1.2. Motivation

Recently the concept of 'Image Steganography' has become an important issue in the computer security world. For a number of years, virus infections and network intrusions have been deemed as the most significant problems within the realm of computer security. Nevertheless, steganography, though totally within the law, may be utilised for illicit processes.
Steganography is used for many legall purposes such as: ownership of digital images, authentication, copyright and data integrity.

However, steganography was also used for illegal issues. For example, internet communication with regard to its use by terrorist organisations when communicating information to one another is still under debate, with one party thinking that terrorists use steganography to speak to one another and the other party thinking this is not the case. Wherever the truth may lie in this regard, there is no dispute about the fact that at the present time, steganography is utilised for the allocation of illicit content, including the information gleaned from stolen credit cards, allocation of user's names and passwords, etc. (Ćosić and Bača 2010).
For these reasons many researchers started to pay attention to steganography and steganalysis to introduce different steganography hiding techniques. Others focused on presenting different steganalysis methods in the literature. These methods were developed to combat specific steganography techniques and to detect specific image formats. However, no single steganalysis method or tool can detect all types of steganography and support all available image formats. Therefore, more steganalysis methods need to be developed to cover a wider range of image formats and colours. They are also needed for breaking a greater variety of steganography techniques and for detecting different sizes of hidden files.
In addition, steganalysis is an interesting topic that has became a challenge for steganalysers and forensic examiners.

1.3. Aim and Objectives of the Research

The main aim of the study is to:

- ❖ Develop detection system to distigush between colour clean and stego images based on extracting CGCM and histogram features.

The aim will be achieved through the following objectives:

- A study and review of the literature related to steganography, image steganography, steganalysis and image steganalysis. Trials of some of the image processing techniques to improve the modified method.
- Create an image data-set for validation. Detection of some stego-images or application of some of the image steganography techniques to produce stego-images to be tested using the modified method.
- Use and compare most common steganography methods to embed information into clean images to create varieties of stego-images.
- Evaluate and compare most common steganalysis methods to detect the embedded information.
- Using classification methods to assess and evaluate the developed methods.

1.4. Contributions

This thesis contributes to the area of computer security. Specifically, it introduces novel methods and system to the fields of image steganaography, image steganalysis and experimental systems research in general.

The five contributions from this research are that:

- **Created image data-base: A** large image data-base was created to train and test the introduced detection system. All images used were coloured images (RGB) with 24 bits. The data-base includes grey images as well that were used to test the histogram features in the early stage of the work.

  Two different steganography methods were used to create the stego images. These were Least Significant Bit (LSB) and F5 algorithm.

- **Statistical features techniques:** existing steganalysis techniques have further developed based on extracting varities of statistical features to detect colour stego images. The selected statistical features were extracted from the Colour Gradient Co-Occurrence Matrix (CGCM). Experiments included analysing three different colour channels separately. The techniques were trained and tested on the created images data-base which includes different image foramts.

- **Histogram features techniques:** existing steganalysis techniques have further developed by extracting histogram features from images. The techniques have been extended to detect colour stego images as well and not only the grey images. Experiments included analysing three different colour channels separately. The techniques were trained and tested on the created images data-base which includes different image foramts.

- **Joint features:** All statstical and histogram selected features were merged together. The CGCM features is examined as statistical features, which takes into account the information of both colour correlation and gradient among the pixels in an image while the histogram features exploited the histogram of difference image.
  Merging two different types of features will increase the detection system's ability to analysis a wider range of steganography methods, image formats, colours and properties.

- **Evaluating the effectiveness of different hidden file sizes:** Steganographic capacity is the size of information that can be hidden relative to the size of the cover image. Two different hiding capacities were selected to embed the hidden files, because detection ability relates to hidden file length. Many experiments were conducted using the CGCM and histogram features to evaluate their effectiveness on the results.

1.5. Key Challenges of Steganography and Steganalysis

Developing a stegnalysis system is a real challenge, as many points need to be taken into consideration while designing. These points are the following:

- **Differentiation between cover and stego file**: this is the first step in steganalysis, and the purpose here is to determine if a given file carries a hidden message.

- **Identification of steganographic method**: There are different types of steganography methods and free tools available on the Internet. However, no one tool supports all image formats. Determining the steganography method used will provide a clue which can help the steganalyser develop a revers process to break the use of the particular method. Normally, all steganography methods leave a specific signature after being used.

- **Estimation of the length of a hidden message**: the length of the hidden file affects the detection process. Clearly, the less information embedded into the cover image, the smaller the probability of introducing detectable artefacts through the embedding process.

- **Message extraction:** this technique normally involves extracting and deciphering the hidden message to obtain a meaningful message. However, extracting the hidden message is another issue than distinguishing the stego images from the clean ones break the purpose of steganography.

- **Image Colour:** Analysis of the grey images is different than that the colour ones with 24 bits. Grey images are easier to deal with as only black and white ones are involved. A colour image, however, has three colour channesl and each of them needs to be analysed separately.

- **Image Formats:** Each image format, such as lossless and lossy, has its own properties. Understanding the nature of the test image helps with developing a steganlysis method.

1.6. Organisation of the Thesis

The rest of the thesis is organised into 8 chapters:

Chapter 2 illustrats wide literature review about steganography and steganalysis in general. It alos presents the concept of steganography, the differences between three common computer security techniques, including steganography, cryptography and watermarking. This chapter is also highlight the meaning of 'Steganalysis 'and the misuse of steganography.

Chapter 3 focuses on summarising an intensive literature review about image steganography and image steganalysis in particular. This chapter also introduces different image steganography and steganlysis methods and techniques. In addition, the free image steganography tools available on the internet and how they work is described.

Chapter 4 describes the methodology and stages of the proposed detection system. This includes a description of the features extracted, details of the image data base created, steganography methods used and the classification methods implemented to train and test the system.

Chapter 5 explains in detail the implementation of the CGCM features. In addition, it describes all experiments conducted using the CGCM features and the classified results using stepwise DA and MLP classification methods.

Chapter 6 describes the implementation of the histogram features. Moreover, it shows all experiments conducted using the histogram features and the classified results using the methods of stepwise DA and MLP classification.

Chapter 7 shows all experiments conducted involving merging the CGCM features and the histogram features. Also, the classified results using the methods of stepwise DA and MLP classification.

Chapter 8 presents a summary of the research, including achievements and limitations of the proposed.

Chapter 2

Steganography and Steganalysis

2.1 Introduction

This chapter presents the concept of 'Steganography', the differences between the three most common computer security techniques, including steganography, cryptography and watermarking, as well as reviewing various steganography types, mediums, tools, methods and applications. This chapter will also highlight the meaning of 'Steganalysis 'and its categories, including some brief facts that need to be considered while detecting hidden information.

2.2. Introduction to Steganography

Steganography is an important research issue in the computer security field (Anderson and Petitcolas 1998). There are many definitions of steganography; it can be defined as the science that aims at communicating secret data in an appropriate multimedia carrier, the multimedia carriers being image, audio and video. However, steganography is also defined as 'the art and science of hiding communication; a steganographic system thus embeds hidden content in unremarkable cover media so as not to arouse an eavesdropper's suspicion' (Provos and Honeyman 2003).

A clear idea of steganography can be obtaining from the following example known as 'the prisoner's problem'. Two prisoners, A and B, wish to escape from the jail. They need to contact each other, but their cells are far apart, and the only way to communicate is by sending a message via the warden. This warden, who is free to examine all messages exchanged between A and B, can be passive or active. If the warden is passive, he/she can examine the message and try to determine if it contains a hidden message. If it appears that it does, then the warden either takes an appropriate action or lets the message through without any action. However, an active warden can deliberately alter messages, even though there is no evidence of a hidden message, in order to foil any secret communication that can nevertheless be occurring between A and B (Anderson and Petitcolas 1998; Avcibas et al. 2003 and Raja et al. 2005).

Figure 2.1 shows the idea of steganography, representing how a message is hidden and extracted using steganography algorithm:

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 2.1: The Steganography Embedding and Extracting Process (Thiyagarajan, Aghila, and Venkatesan 2011).

Steganography, in this new era, aims at preventing its presence from being detected. Even if the hidden contents are not unveiled, its existence often is because modifying the cover medium changes its statistical properties. Hence, eavesdroppers may sense the distortion in the stego medium's statistical properties caused by the embedding of a secret message. Finding those distortions is called statistical steganalysis (Provos and Honeyman 2003).

2.3. Steganography: A History

'The word steganography is derived from the Greek words 'steganos' meaning 'cover' and 'graphie' meaning 'writing', defining it as 'covered writing'. For thousands of years, it has existed in various forms '(Johnson and Jajodia 1998: 26). In the 5th century BCE, when Histaiacus wanted to send a secret message, he shaved a slave's head and tattooed the secret message on the slave's skull. When the slave's hair grew again, he was dispatched with the message. This is a classic example of steganographic system that relies on secrecy by encoding a secret message. However, though this system might have worked for a time, once it was known, it would have been quite simple to detect by shaving the heads of people passing by (Provos and Honeyman 2003).

Additionally, in the past, Egyptians used a technique involving illustrations to conceal secret messages.

And not long ago in Saudi Arabia, a project was initiated at the King Abdul-Aziz City of Science and Technology to translate from secret writing into English a large number of ancient

Arabic manuscripts which were believed to have been written over 1200 years ago (Anderson and Petitcolas 1998 and Johnson and Jajodia 1998).

According to some sources, the Nazis invented many steganographic approaches during World War II, including microdots, and they reinvented the use of invisible ink and null ciphers (Anderson and Petitcolas 1998; Hmood et al. 2010).

2.4. Need for Data Hiding

There are many reasons for hiding data in objects, and some of these reasons are:

- Ownership of digital images, authentication and copyright.
- Covert communication using images.
- Traitor-tracing (fingerprinting video-tapes).
- Data integrity, fraud detection and self-correcting images.
- Intelligent browsers, automatic copyright information, viewing a movie in a given rated version.

(Umamaheswari et al. 2010).

2.5. Steganography, Watermarking and Cryptography

Three techniques are interlinked: steganography, cryptography and watermarking.

Steganography is entirely different from cryptography in that cryptography focuses on encrypting the contents of a secret in a message, while steganography concentrates on hiding the existence of a secret in a message. Steganography and cryptography are both methods of withholding information from unwanted parties, but neither technology is so advanced that it cannot be compromised (Anderson and Petitcolas 1998; Provos and Honeyman 2003; Hmood et al. 2010; Zielińska et al. 2014).

Figure 2.2 below shows the different embodiment disciplines of information hiding.

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 2.2: The Different Embodiment Disciplines of Information Hiding (Cheddad et al. 2010: 728).

The strength of steganography can be amplified by combining it with cryptography, as shown in table 2.1.

Table 2.1. Comparison of the Security Levles when using Steganography and Cryptography (Singla 2014).

Base	Steganography without cryptography	Steganography with other cryptography technique
Security	One level security	Two level security
Key Size	No key present	Fixed size of key
Steps involve in enctyption of message	No step	Fixed step
Brute force attack	No need	Can possible

Table 2.2 summrises a comparison of steganography and cryptography in terms of their goals, charactestics and countermeasure.

Table 2.2. Comparison of Characteristics of Steganography and Cryptography (Zielińska 2014; Holla 2014).

		Cryptography	**Steganography**
Defenition		Art of study of hiding information.	Art of hiding messages in a clean (cover) file within another clean file.
Goal		Although the message is encrypted and unreadable, the existence of the message is not hidden.	Hide the fact of communication. No knowledge of the existence of the hidden message.
Characteristics	Secrecy	-Existence of message is visible to the world.	-Only sender and reciver knows the existence of message.
	Security of communication	-Relies on the confidentiality of the key. -It tries to protect content of message.	-Relies on the confidentiality of the method of embedding.
	Warranty of robustness	- Complexity of the ciphering algorithm. - Steganography prevents discovery of the very existence of communication.	-Perceptual invisibility/statistical/invisibility/ compliance with protocol specification. -Encryption prevents an unthorised party from discovering the contents of a communication.
	Attacks	-Detection is easy/extraction is complex. -Strong current algorithms are currently resistant to attack; larger expensive computing power is required for cracking.	-Detection is complex/ extraction is complex.
Countermeasure	Technical	-Reverse engineering. -Cryptography alters the structure of the secret message.	-Constant monitoring and analysis of exchanged data. -Steganography does not alter the structure of the message.
	Legal	-Cryptography export laws	-Rigid device/protocol specification

Another technology related to steganography is watermarking. This technology is mainly concerned with the protection of intellectual property; thus, the algorithms have different requirements than in steganography. With watermarking, all of the instances of an object are 'marked' in the same way. However, in steganography this does not happen. The aim in steganography is to hide information, whilst in watermarking, the aim is to protect the ownership of intellectual property (Provos and Honeyman 2003; Cheddad et al. 2010; Zielińska et al. 2014).

2.6. Types of Steganography

Steganography has three protocol types: pure steganography, secret key steganography and public key steganography (Dunbar 2002; Ashok et al. 2011).

- **Pure Steganography** is the process of enclosing the data into the object without interfering with any private keys. This kind of steganography is exceedingly dependent on secrecy. Personal information to be transmitted uses a cover image in which data is to be enclosed, and encryption/ decryption algorithms are used to embed the message into the image (Ashok et al. 2010 and Dunbar 2002), as shown in Figure 2.3:

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 2.3: Pure Steganography Process (Al-Ani 2010: 159).

- **Secret Key Steganography** is exemplified by a steganographic regime in which the exchange of a secret key (stego-key) is needed in order to communicate. This type of steganography implants a hidden message within a cover message by operating a secret key (stego-key). The secret key can only be known, retrieved and read by the parties who know the message (Dunbar 2002; Ashok et al. 2010), as illustrated in Figure 2.4:

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 2.4: Secret Key Steganography (Al-Ani 2010: 159).

- **Public Key Steganography** is comparable to the notion of 'Public Key Cryptography' insofar as it is a steganographic scheme that employs a pair of keys – namely, a public and a private key – for the purpose of protecting communicative activity for entities who are aiming to initiate secretive communications. As stated by Dunbar (200) and Ashok et al. (2011), the message transmitter uses the public key over the course of the encoding procedure while the recipient employs the private key for the purpose of deciphering, as shown in Figure 2.5:

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 2.5: Public Key Steganography (Al-Ani 2010: 160).

2.7. Steganographic Channels

The aim of steganography is to communicate secret messages as if no communication were taking place. This can be achieved by hiding messages in objects that look ordinary, and then sending these objects in an overt manner through certain communication channels (Fridrich 2010 and Owens 2002). To understand these channels, let's return to the prisoners' story when A & B wanted to secretly communicate. A & B must agree on some basic communication protocol to follow and select the type of cover objects that they will use for sending secrets. Then they need to design the message with hiding and extracting algorithms. Finally, the prisoners will send their messages through a channel that is under the control of the warden, who may or may not interfere with the communication (Owens 2002 and Fridrich 2009). The following are three types of embedding algorithms (Fridrich 2009):

- Steganography by cover selection
- Steganography by cover synthesis
- Steganography by cover modification

2.7.1. Steganography by Cover Selection

With this method, Prisoner A has a fixed database of images from which he selects one that communicates the desired message. For instance, one bit of information could be sent by choosing a picture in a landscape, or a hidden meaning such as 'attack tomorrow' could be shown by the presence of an animal in the picture. Simply, the embedding algorithm can work by pulling images randomly from the database until an image is found that communicates the wanted message. For this stego key, it is important to set the rules that tell A & B know how to interpret the images (Fridrich 2009).

2.7.2. Steganography by Cover Synthesis

With this method, Prisoner A creates the cover so that the desired message is conveyed. The press has speculated that Bin Laden's videos may have contained and communicated hidden messages by using steganography by cover synthesis. These hidden messages could have been in his clothes, the position of his rifle or the choice of words in his speech (Fridrich 2009).

2.7.3. Steganography by Cover Modification

This method is the most studied steganography paradigm today. Here Prisoner A starts with a cover image and makes modifications to it in order to embed secret data. Prisoners A & B then work with a set of all possible covers and sets of keys and messages that may depend on each cover (Fridrich 2009).

A steganographic channel consists of some basic elements: the source of covers, the message source, embedding and extraction algorithms, the source of stego keys, and the communication channel (Fridrich 2009).

2.8. Steganography Applications

A number of applications use steganography, for example, copyright control of materials, smart IDs and identification cards as individuals' data are enclosed in their photographs. Other applications include companies' TV broad casting, video–audio synchronization and TCP/IP packets, which can be a unique ID enclosed into an image to analyse the network traffic of specific users (Cheddad et al. 2010).

The following are examples of these applications:

- Steganography is utilised by intelligence agencies all over the globe to exchange certified data in a covert manner. For example, secret agents may mislead a terrorist campaign by placing a map in a photo using an image steganographic program and posting it on a public blog/discussion board or in a forum (Hayati et al. 2011).

- In the medical image system, a link must be maintained between the image data and the patient's personal information. Therefore, embedding the patient's information in the image could be a useful safety measure and help to guarantee the authenticity of the data (Anderson and Petitcolas 1998; Cheddad et al. 2010).

- Steganography is additionally employed within typical procedures for printing. A Japanese company, Fujitsu, is creating devices that encode information into a printed image, which cannot be seen with the naked eye, but can be decoded using the camera on a mobile phone. This procedure happens quickly (less than one second) as the implanted information is only twelve bytes. Cheddad (2009) and Cheddad (2010) explained that individuals could employ their mobile phones to access the encoded information (Cheddad 2009; Cheddad et al. 2010). Figure 2.6 is a sketch representing these concepts:

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 2.6: Fujitsu Exploitation of Steganography (Cheddad et al. 2010: 730).

- Cheddad et al. (2010) has stated that 'the confidence in the integrity of visual image has been ruined by contemporary digital technology'. Therefore, further research related to digital document forensics has been proposed. For example, Cheddad et al. (2010) proposed a security scheme that uses self-embedding techniques to protect scanned documents from forgery. The method also allows legal or forensics experts to get access to the original document, regardless of whether or not it had been manipulated, as can be seen in Figure 2.7:

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 2.7: (a) digital document forgery detection; (b) stego-image carrying self-duplicate; (c) attacked stego-image (date received changed; the 4th lead inventor's name has been removed); (d)inverse half-toning of the reconstructed hidden data from the attacked version; (e) error signal of b and d; (f) after applying thresholding operation (Cheddad et al. 2010: 731).

2.9. Steganography Mediums

In steganography, data can be hidden in several main carriers, including audio files, videos, images, data transmission and text files. Stemming from this, the main steganographic mediums can be identified as audio, text, and image (Dunbar 2002; Ashok et al. 2011).

2.9.1. Embedding Data in Text

The encryption of a secret message within a text file is very challenging; because, text files possess only a small amount of data deemed to be redundant, when swapping them for secret information. According to Richard (1998), Dunbar (2002) and Ashok et al. (2011), steganography that utilises text can be made more effective or re-termed/altered by other, unwanted, individuals through alterations made in the text itself or by changing the text into a totally different form.

- **Line-shift Encoding:** The line-shift encoding technique changes a document through the vertical shifting of text lines and can be utilized for both file formats and page images. The preassigned code word with regard to a certain document determines the lines of text that are to be used within the document, an example is shown in Figure 2.8 (Richard 1998; Dunbar 2002).

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 2.8: Line Shift Coding: An Example (Richard 1998: 24).

- **Word-shift Encoding**: This merely utilises the horizontal spaces among each word to provide a value for all hidden messages. ‘Shift encoding’ demands this support for the text formatting, and the process is able to change the document through the horizontal

shifting of word placements within lines in order to embed one-of-a-kind marks, an example is shown in Figure 2.9 (Richard 1998; Dunbar 2002; Ashok et al. 2011).

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 2.9: Word Shifting Encoding: An Exaggerated Example (Richard 1998: 25).

- **Feature-Specific Encoding:** As the most challenging of all encoding methods, this approach utilises the encoding of secret messages into formatted text by altering specific text characteristics like the horizontal or vertical length of certain letters like b, d, T (Richard 1998; Dunbar 2002; Ashok et al. 2011).

2.9.2. Embedding Data in Images

The most frequently employed encoding procedure is associated with embedding data in digital images, and the primary reason for this is due to the fact that it can capitalise on the limitations of the HVS. It is possible to use digital images to conceal any form of media which is compatible with bit stream encryption, and this includes plain text, cipher text, and images. The most widespread digital image encryption methods reflect the smallest LSB encrypting, and this are the masking and filtering approaches (Ashok et al., 2011; Maitra et al., 2011).

- **In the LSB technique**: the LSB of the pixels is altered by the message to be sent. The message bits are permuted just before enclosing, effectively distributing the bits evenly, hence on average, only half of the LSBs are going to be modified (Kharrazi et al. 2005).

- **Masking and filtering techniques:** These techniques work by embedding two signals into each other. The embedding process is done in such a manner that only one of the signals is perceptible to the human eye. This is mainly used in watermarking techniques (Dunbar 2002; Raja et al. 2005; Maitra 2011).

2.9.3. Embedding Data in Audio

One of the most challenging techniques to be used while dealing with steganography is encrypting hidden messages within audio files because the HAS has a dynamic range that it can listen to. HAS is the only drawback to this method. HAS transpires when attempting to identify various sounds as louder noises dominant the system. However, the program must be able to identify various sounds to encrypt secret content into sound files. Dunbar (2002) and Maitra (2011) explained the common techniques for encrypting information within a sound file, which are listed below:

- Low-bit encoding
- Phase-coding
- Spread spectrum

- **Low-bit encoding** incorporates concealed data into the lowest level possible LSB sound file. Ashok (2011) and Maitra (2011) explained that these files typically have a conversion ability of 1KB/second/kilohertz.

- **Phase coding** performs a type of encoding for the sound file, according to Ashok (2011) and Maitra (2011), through employing a DFT.

- **Spread spectrum** initially encrypts the sound file throughout its whole system, according to Ashok (2011) and Maitra (2011), and then conveys the sound file through various frequencies.

2.10. Steganography Techniques

Steganographic technologies have a very important role to play in the future of internet security and privacy on open systems such as the internet. Because of the desire to have complete secrecy in an open-system environment and because cryptographic systems are inadequate on their own, research in steganography has been pursued (Ashok et al. 2011).

To limit the strength of cryptosystems, many governments have created restrictive laws, which in some cases prohibit these cryptosystems completely. These actions have been carried out primarily because of law enforcement's fear of not being able to acquire intelligence by wiretaps, etc. (Ashok et al. 2011).

2.10.1. Steganographic Method

Steganography employs various types of methods, and most of them operate in two steps. First, analyse a cover object to decide to what extent it can be modified so that the modifications will not be easily noticeable. Second, create an altered cover object by inserting the message bites into the cover object; this can be done by making changes which are replaced by the message bits (Duric 2004).

2.10.2. Major Steganography Techniques

Steganography techniques can be classified into three main modes: 'Injection', 'Substitution' and 'Generation of New Files'. These techniques use specific bit locations as the covert channel for communications. Mostly, they utilize a stego-key, which provides control for the hiding, restricting detection and recovery processes by those who are not aware of the key (Owens 2002; Ashok et al. 2011).

- **Injection** is the insertion of a message into a cover file. A simple example of this technique is the use of the hidden attribute in Microsoft Word, which allows for hiding text with a special, hidden font (Owens 2002; Ashok et al. 2011).

- **Substitution** information from the initial file with an encrypted image of the initial content. Owens (2002) and Ashok (2011) explained that pixel hues (i.e. micro aspects of digital pictures) are typically represented by the utility of the sum within the 8 bytes of information.

- **Generation of a New File**, in some cases, either the insertion or substitution technique requires a holding file referring to images, and a keeper signal referring to audio signals. Host files get the secret messages, but they also may display signatures after embedding, which can be utilised by steganalysis tools to sense the secret messages (Owens 2002; Ashok et al. 2011).

2.10.3. Embedding Process

In order to get a more technical view of the embedding process, glance over Figure 2.10. It may be helpful to first read the brief explanation of the Figure below:

- ✓ 'First, let C denote the cover of the carrier, Image 'A, and C´ is the stego-image' (Natanj et al. 2011(.
- ✓ 'Second, let K symbolise an optional key (a seed used to encode the message or to create a pseudorandom noise that might to be set to {Ø} for simplicity); and let M be the message wanted/needed to be communicated, Image B. Em is an acronym for enclosing and Ex for Extraction' (Natanj et al. 2011).

Therefore:

$$Em: C \oplus K \oplus M \rightarrow C'$$

$$\therefore Ex(Em(c,k,m)) \approx m, \ \forall c \in C, k \in K, m \in M$$

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 2.10: A Theoretical View of the Embedding Process (Natanj 2011).

However, three components make up the basic structure of steganography: the 'clean (or cover), the message, and the key. The clean (or cover) can be a digital image, an audio file, even a protocol (like TCP/IP packet). The cover will conceal the hidden message (Natanj et al. 2011).

2.11. Introduction to Steganalysis

Steganography, as we have seen, is principally aimed at storing secret information in carrier files. Steganalysis, on the other hand, focuses on detecting hidden files, images and text, and has been defined as 'the art and science of detecting secret messages hidden using steganography' (Johnson and Jajodia 1998; Fridrich, Goljan, and Du 2001 and Nissar and Mir 2010). Steganalysis can also be defined as 'the art and science of stopping or detecting the use of all steganographic techniques' (Huayong et al. 2011). And a steganalyst is defined as 'the one who applies steganslysis in attempting to detect the existence of hidden information' (Johnson and Jajodia 1998). Schaathun (2012) defined the basic steganalyser as 'an algorithm which takes a media file as input, and outputs either "steganogram" or "innocent"'.

Steganalysis is intended to detect or estimate the possibility of hidden information based on observing some data transfer, but it makes no assumptions about the steganography algorithm (Chandramouli 2002; Kessler 2004). The principle aim is to collect enough evidence that there is an embedded message and, where possible, break the security of its carrier. Steganalytic techniques for detecting hidden information in images are increasing in number.
Steganalysis is widely used in computer forensics, cyber warfare and tracking criminal activities through the internet, especially anti-social elements. Steganalysis also helps to improve security of steganographic tools by evaluating and identifying their weaknesses (Johnson and Jajodia 1998; Fridrich, Goljan, and Du 2001).

Attempts to predict the duration of the unknown content. Schaathun (2012) explained that identifying key recovery assaults are clearly beneficial since the key helps recognize and analyse all following discussions that employ an identical key. Additionally, Schaathun (2012) also explained that elongated steganalysis incorporates methods for identifying additional data regarding the unknown content or devices employed; for example, the stego-algorithm that was employed.

2.12. Categories of Steganalysis

There are several types of steganalysis, which consist of detecting, extracting and destroying hidden objects of the stego media. One expert in the field, Neil Johnson, sets out six main categories of attacks:

1) Stego only - The evaluation must be conducted using only the stego item.
2) Known cover - the original cover object and the stego object are available for analysis.
3) Known message - The secret content is accessible when contrasted with the stego-item.
4) Chosen stego - The stego-algorithm and stego-item are both accessible for the evaluation.
5) Chosen message - Uses selected content to create a stego-item for additional evaluations.
6) Known stego - The stego-algorithm, concealing content, and stego-item are known and can be employed for evaluation.

(Johnson and Jajodia 1998; Huayong et al. 2011).

2.13. Facts in Detecting Steganography

Steganography in digital media could be largely classified as operating in the transform or image domain. Image domain tools use bit-by-bit manipulation to hide the message in the carrier. Transform domain tools manipulate the steganography algorithm, such as the discrete cosine transforming coefficients in JPEG images (Chandramouli 2002; Johnson and Jajodia 1998 and Kessler 2004).

The way steganography algorithms operate is followed by steganalysis. One approach is to inspect the carrier and steganography media visually. Another way is to look for structural oddities that suggest manipulation (Chandramouli 2002 and Wayner 2002). While detecting steganography in images the following needs to be addressed:

- It is helpful to first determine the steganalysis category, e.g. stego-only attack, known message attack, etc. (Kessler 2004).
- Hiding the message in the area of brighter colour might be easier, but the program may not pick this up, as a palette-based image can duplicate similar colours which differs only in the least significant bit, causing structural changes, which create a signature of the steganography algorithm that was employed (Chandramouli 2002; Wayner 2002; Kessler 2004).

- Steganographic techniques often change the statistics of the carrier; this will affect longer hidden messages more than shorter ones (Farid 2001; Kessler 2004).
- Statistical analysis is widely used, especially working 'in the blind', by an analyst to detect hidden messages (Wayner 2002; Kessler 2004).
- Detection is harder when messages are encrypted as there is normally a high degree of randomness, e.g. ones and zeros appear with equal likelihood (Farid 2001 and Provos and Honeyman 2001).
- Knowledge of the crypto algorithm, an encryption key, or an estimate of the message length is required to recover the message (Fridrich et al. 2003 and Kessler 2004).

2.14. Digital Forensic, Misuse and Legal Issues of Steganography

For a number of years, virus infections and network intrusions have been deemed as the most significant problems within the realm of computer security. However, steganography, though totally within the law, may be utilised for illicit processes, and it is thus increasingly recognised as a very serious security threat.

2.14.1. Digital Forensics

Digital forensics concentrates on the analysis and preservation of digitised evidence. It may be defined as the 'utilisation of a precisely defined and substantiated method regarding the preservation, validation, collation, assessment, documentation, interpretation, presentation and identification of digital evidence away from digital sources with the intent to propound and facilitate the reconstruction of occurrences deemed either criminal, or alternatively assisting the anticipation of unauthorized actions thought to be either scheduled or disruptive' (Warkentin, Bekkering and Schmidt 2008). Due to the fact that steganography is now more commonly seen and available, and because the volume of data stored in personal and local computers across the internet is growing, detection issues in utilising steganography for digital forensics workers has become more important. Practically speaking, a majority of instances will involve the use and assessment of audio-visual files, as is the case in child pornography.

Nevertheless, in instances of industrial espionage, as well as fraudulent activities according to Warkentin, Bekkering and Schmidt (2008) and Rachel and Adelstein (2009).

- **UK Child Pornography Laws**

  The main concern of legislators and parents in relation to Internet content is child pornography, rather than other forms of pornographic content. This has been the case ever since paedophiles started to use the Internet for circulating pornographic materials related to children. Paedophilia can be seen as a minority sexual group, with its own form of expression explicitly involving fantasies and imaginings about sex with children. But while it is often argued that pornography should not be proscribed on the basis of freedom of speech arguments, there is a general consensus that the line should be drawn with child pornography (Akdeniz, 1997).

- **Software detection**

In a number of instances, steganographic software may be encountered on computerised equipment that is subject to assessment and investigation. At the present time, the Steganography Application Fingerprint Database (SAFDB) is used to identify information across 625 applications relating to steganography, other data-hiding actions and watermarking (Backbone Security, 2008a). In the same manner, the NIST (National Institute of Standards and Technology) retains a list of digital signatures within the National Software Reference Library., A number of these are utilised for steganographic purposes and steganographic software (Warkentin, Bekkering and Schmidt 2008).

- **Detecting Pairs of Carrier Files and Stego Files**

As well as detecting software that is to be employed for the purposes of steganography, experts in the field of digital forensics are able to detect files that have comparable visual characteristics, though they vary with regard to file size, statistical properties and hash values. In the case of all files having been removed or deleted, they can be obtained once more from the trash or recycling bin, and they can even be reconstructed utilising special forensic tools employed for recovery purposes.

- **Using Keywords**

A further means by which files may be detected is to look for keywords, as well as data files, in order to search for the file names and the program's content. This list needs to be precise with regard to steganography; for example, 'steg' as a search term, may be utilised in the identification of steganography, and its use regarding the efficaciousness and efficiency when detecting such files prevents false negatives and false positives, as it is dependent on the keyword quality in the dictionary, according to Merrill, Bekkering and Schmidt (2008).

- **Specialized Steganalysis Software**

Formerly, a majority of steganographic detection tools were targeted at software applications, and these were usually the same applications employed for the purposes of steganography. More contemporaneously, software claims employed in the detection of stego files have been devised using a broad array of programs, among them the Stegdetect 0.6, which utilises a process of linear discriminate analysis in order to find the location of probable images in hidden content by contrasting them with a normal image set (Provos 2008). Stego Suite is another example of these programs (developed by Wetstone Technologies 2008), and it is able to assimilate more and more intense detection levels with content cracking tools.

- **Physical Crime Scene Investigation**

According to Merrill, Bekkering and Schmidt (2008) useful information may be revealed through the use of physical crime scene investigation. Passwords that have been utilised for steganographic tools may be etched on notes or else placed under keyboards, while other features or characteristics of the environment provide further clues as to likely or potential passwords.

2.14.2. Digital Security Issues

Despite the fact that most steganographic tools are utilised for legitimate corporate processes such as the safeguarding of corporate information when transferring it to others, issues have nevertheless emerged regarding the utilisation of forensic investigations, as well as individual worries about criminal use (Merrill, Bekkering, and Schmidt 2008 and Schmidt et al. 2004).

Steganographic tools are becoming more and more useable, and also simpler to use, thus safeguarding against malicious use of such programs requires a higher level of attention. And the disharmony between protection against criminal use and the repercussions of safeguarding on genuine use is a future difficulty to be overcome, according to Kellen (2001) and Merrill, Bekkering and Schmidt (2008).

- **Unauthorized Steganography Software detection and location**

Sadly, not every business adheres to corporate policy, and there are a number of ways restricted user permissions can be bypassed. In a business environment, the measures listed below should be considered:

- Intrusion detection software to detect unusual movement of graphic files. While a majority of businesses do not entail the transference of graphic files, a majority of traffic comes from web browsing. Thus graphic files that are outgoing, whether as attachments in emails or in isolation, need to be scanned.
- Due to the fact that the detection of stego files is impossible unless sought out, automated scanning of computers in the network may be useful. This may be attained through several commercially available software programs.
- Last, all computers that are repaired or maintained on site need to be scanned on a regular basis for steganography, in addition to general malicious software and viruses.

All of these safeguarding processes should add to the efficacious identification and removal of criminal or malicious steganography within a corporation (Merrill, Bekkering and Schmidt 2008).

2.14.3. Steganography and its Misuse

Internet communication with regard to its use by terrorist organisations when communicating information to one another is still under debate, with one party thinking that terrorists use steganography to speak to one another and the other party thinking this is not the case. Uncontested, however, is the fact that steganography is currently utilised for the allocation of illicit content, including information gleaned from stolen credit cards, allocation of user names and passwords, fake monetary orders and the distribution of registry and insurance numbers from public databases. Indeed, it is also clear that steganography is often used in spying,

wherein intruders infiltrate firms, spy and then report information about the firm to those who hired them (Ćosić and Bača 2010).

- **Terrorists**

After the September 11 attack of the Word Trade Centre, there was a heightened awareness of terrorism and terrorist activities at the front of many nations' concerns and interests. Indeed, the presence of terrorism has also raised the issue as to the means by which these individuals communicated with one another, and how orchestrated attacks had been organized. At the time of the attack on the World Trade Centre, *USA Today* published a piece that indicated that the internet was used to relay information to terrorist groups, and bulletin boards, chat rooms and other web sites were indicated in this speculation (Kellen, 2001; Kessler, 2006). Nevertheless, there was no substantiating evidence for this claim, and the readers themselves had to decide on its accuracy. According to Kellen (2001) and Kessler (2004), other news agencies reported that some eBay and Amazon posts contained subliminal information. On July 2009, a terrorist attack on a Mumbai hotel was orchestrated by using coded messaging which had been embedded into ordinary emails within the Yahoo server. For several months, investigators had to look for these messages (Ćosić and Bača 2010; Kessler 2004).

- **Child Pornography**

With the use of steganography, files may be able to carry other images or credit card numbers. Pornographic website images are generally large and take up a considerable spectrum of colours, thus making them perfect files by which to carry illicit information. The bigger the file is, the larger payload it is able to carry with it. Stego tools, which operate with the use of graphic files, generally embed the payload file by altering the LSB within all pixels of the file itself. Bigger files, therefore, have a greater number of pixels and so have with them a greater amount of data, according to Kellen (2001), Rachel and Adelstein (2009) and Kessler (2004).

A number of news reports have expounded on the danger and the threat of steganographic use (SARC), such as the *Times Online*, which made a report to the police force in the UK in October 2008 about a terrorist cell, and claimed to have found that secret messages had been embedded into images of child pornography. Nevertheless, child pornography images are generally hidden and are not used as the carrier information, according to Kerbaj and Kennedy (2008) and Davidson and Jalan (2010). In the instance above, a criminal organisation, on the basis of a deal they had made in the past, allocated in a 'legal' manner, images of children via an eBay

server, and the group itself had been registered on the website, along with correct payment credentials and offers and payments (Ćosić and Bača 2010).

The protection of children Act (1978) was further amended by the Criminal Justice and Immigration Act 2008, which provided that "photograph" includes: "a tracing or other image, whether made by electronic or other means (of whatever nature)— (i) which is not itself a photograph or pseudo-photograph, but (ii) which is derived from the whole or part of a photograph or pseudo-photograph (or a combination of either or both)," and including data stored on a computer disc or by any other form of electronic means that can be converted into such an image (Akdeniz, 1996; Legislation, 1978; Legislation, 2008).

❖ Alternative Scenarios

Alternative possibilities for misusing steganography arise when auctioning websites like eBay present an efficacious means to place and post steganographic files. Rather than using a file of a pornographic nature, the individual may post an image of an object they wish to sell before running the picture through a tool that places hidden information into the file via steganography. The image is then posted online for millions of users to see, though most will not realise the importance of the image as carrying any sensitive information.

The recipient of the image alone is aware of its importance, downloads it, and then is able to see the actual intended information by running it through the same stego as his accomplice, according to Kellen (2001). Terrorist personnel may also have colleagues working in large companies or even providers of websites, thus it is not improbable that such an image featured on the website of a firm may contain terrorist information entirely without the firm knowing it. Steganographic processes are also able to relay other kinds of information through audio files like MP3, AU, MID and WAV files, which present opportune carriers being almost as commonly used as image files. Indeed, there are plenty of steganography tools for audio files, too. Audio files may be embedded within the HTML code of a web page and then played automatically after a number of clicks on them or visits to a page. In reality, the file is downloaded into the 'temporary files' folder and then played from the person's computer. One would not be able to determine if the person is merely hearing the sound or trying to secure illicit material. Therefore, the terrorist would merely have to obtain the audio file from the temporary folder later in the day and decode the file at leisure (Kellen 2001).

- **Related Studies**

Peter Honeyman and Niels Provos (2001) conducted the most thorough of all research articles on the subject. The authors analysed more than two million images from eBay, as well as one million from USENET archives; however, the authors failed to find even one image that contained hidden information. In assessing the information, the authors used their own program, which was intended and devised to identify hidden information created by the most popular steganographic open source programs that existed at the time. As a result of a number of people presenting the argument that the framework for analysis was insufficiently efficacious and that commercial programs were superior, questions arose as to the usefulness of the study itself. A further study demonstrated that the Honeyman and Provos (2002) study was unable to detect any data that had been kept secret with the use of non-open software. This raised the issue that potential terrorists could purchase and use commercial steganographic programs.

Rachel and Adelstein (2009) conducted a further study that assessed a number of popular steganographic programs in the attempt to determine which, if any, artefacts remained once the program had been fully installed, before it was then uninstalled. About half of the programs assessed left some clues or trace information. Nevertheless, the findings of this study indicated that the investigator was able to perform an expedient check for specific directories, registry keys and files within the earliest stages of the investigation itself. Furthermore, it provided evidence that would galvanise the process itself, and these artefacts presented interesting clues. For instance, understanding the identity of the program that was employed would assist the investigator when concentrating on the kind of tool employed to make the information secret.

2.14.4. Challenges and Issues Concerned with the Law

A number of new issues and challenges for digital investigators, enterprise managers and security staff, as well as legal workers and law-making bodies are created by steganography. This will contribute to the multifaceted and growing volume of forensic duties, and the creation of case law. According to Merrill, Bekkering, and Schmidt (2008) and Rachel and Adelstein (2009) and Stahl et al. (2010), prospective research needs to be promoted by and from academics and research authors in the steganographic field. Most developed nations like the

EU nations and the USA are trying hard to supervise and optimise their protection of critical infrastructural frameworks as a result of these issues.

The European Commission (EC), for instance, came to understand that traditional security measures were becoming outdated and that new techniques were needed, one of which is steganographic in nature. The Future Internet initiative was launched by the EC with regard to the future and essential economic growth in Europe (Rachel and Adelstein 2009; Tofan 2010; Dinca 2011). The initiative alludes to the demands that the small- and medium-sized enterprises (SMEs) have concerning the utilisation of their resources, something that poses a significant security danger and threat. It is thought that larger corporations and countries and their governments are able to safeguard their information; this is not the case for SMEs, however. The most significant issues concerning EU SME's internet security can be seen below:

- **Access Control**

There is an increasing call for regulating and monitoring processes (log monitoring), as well as for the most up-to-date firewalls, updates and smart passwords. These should help guarantee that resources are not being abused. A typical means of attaining illegitimate ends is to penetrate the network in question by gaining access and utilising its data after working hours. According to Tofan (2010) and Dinca (2011), an attack may be carried out months before anyone notices, and thus monitoring solutions should take this means, and several others, into account.

- **Steganalysis**

As a result of the growing need for larger and larger bandwidth for SMEs and smaller enterprises, these firms are made more vulnerable with regard to steganography. At the present time, emails contain larger files, such as images and pictures, which can also be sent to numerous individuals at once. It is usual to get amusing messages from colleagues before relaying them to others, though such messages may have within them steganographic information which may be utilised for corporate crime by criminals or terrorists. It is significant that steganalysis tools are used in internal LAN safeguards, according to Tofan (2010) and Dinca (2011).

- **Multi-tenancy**

As a new concept, multi-tendency is something that alludes to the architectural principle wherein a solitary instance of the software runs a SaaS vendor server, thus serving a number

of client organisations. Previously, it was suggested that the safety of the application be assured by installing it within a controlled setting, such as a server in one's own LAN. This process is having an effect on a number of firms, and this is especially true of SMEs who utilise SaaS. Some of these SMEs have given up their own servers and even personal computers as a result (Tofan 2010; Dinca 2011).

Ballard, Hornik, and McKenzie (2002) came to the following conclusions which are pertinent to counterterrorism professionals:

- Despite having been employed in a number of formats over thousands of years, digital steganography presently has a very low visibility in frontline law enforcement agencies.
- The chance that a message will be detected is reduced if it is concealed by steganographic methods.
- An innocuous image is the superlative message carrier, like an image of a car or a house. Terrorists will probably select images that may not be contrasted to the original.
- Steganographic tools are available in great numbers, both as freeware and paid programs.
- These tools are simple to use; most computer users could master them in a short time-frame.

Chapter 3

Image Steganography and Image Steganalysis

3.1. Introduction

This chapter introduces the concepts of image steganography in detail. It discusses the main image steganography techniques that are used to embed the hidden files in the spatial domain and transform domain. In addition, it describes the image steganography and free tools available and presents image steganalysis techniques, types and tools. Finally, previously developed image stegnalysis methods from the literature are reviewed.

3.2. Introduction to Image Steganography

Images are the most frequently employed item for conveying secret information due to numerous causes. Images possess significant capacity for undetectable alterations to the image messages. Furthermore, Duric (2004) explained that since pictures are commonly accessible online, they can be employed to convey items without appearing out of the ordinary. Digital images have numerous types of files, but a majority of images are employed within particular programs with JPEG and GIF being the most common. According to Morkel (2005), various steganographic algorithms prevail that are associated with various picture file types.

3.2.1. Digital Steganography

The application of steganography to messages is founded on two elementary principles.
Firstly, the files have within them a number of images in digitized or audio form that may be altered, to a certain degree, without the loss of functionality. Secondly, it is virtually impossible for human individuals to determine and differentiate small alterations in colour or sound quality. This aspect is particularly simple to exploit within objects when conducting stenography, particularly with regard to those objects that include information deemed redundant, such as the common 8-bit and 24-bit image files as well as 6-bit sound files. For instance, for image files, the alteration of the value of the LSB with regard to the colour of the

pixels does not hold any repercussions for a human observer, as colour changes are indistinguishable to them. The information itself is buried in the "noise" present within or input to the computer file, according to Ballard (2002) and Das et al. (2011).

❖ Image Definition

A digital image can be defined as a group of figures, which incorporate various degrees of light in various regions of the picture (Morkel 2005). Significantly, every computer picture file incorporates series of dots, known as pixels, that are successively arranged in horizontal rows. Every pixel has a different hue that are signified precisely and in isolation within the image data as red, green, and blue. These pixel hues together create the RGB. Within an 8-byte picture, for instance an graphic interchange format (GIF) ten or bitmap (BMP) eleven file, every pixel is represented by a figure between 0 and 255, which represents the precise hue in the colour lookup table or scheme. This system is explained in Table 3.1. Within this system, the figure 0 signifies an absence of hue while the figure 255 signifies an excess of hue.

In other words, the RGB figure of 0,0,0 would mean that the pixel is black while the RGB figure 255,255,255 would mean that the pixel is white. Overall, 16,777,216 (256*256*256) different hues exist within the RGB scheme. On the other hand, a majority of GIF files only employ an 8-byte colour scheme. Therefore, Ballard (2002), Morkel (2005), and Das (2011) explained that out of the total 16,777,216 feasible hues, only 256 RGB hues are incorporated into the picture as the 8 byte binary figures are limited to 256 separate figures.

Table 3.1: Primary Red-Green-Blue Color Pallete (Ballard, 2002).

Color	**Red**	**Green**	**Blue**
Red	255	0	0
Green	0	255	0
Blue	0	0	255
Yellow	255	255	0
White	255	255	255
Black	0	0	0

❖ Color Representation

True Color (24 bit): Every pixel hue is symbolized through employing three bytes of red, green, and blue or RGB hues. Omer and Werman (2004) and Willamette (2013) explained that one byte can signify 256 various hues, which means that approximately sixteen million (256*256*256) hues can be signified.

Figure 3.1: Represenation of the 24 bit Colour (Willamette 2013).

True Color (32bit): The 32 bit is identical to the 24 bit true colour, apart from the additional byte. Omer and Werman (2004) and Willamette (2013) explained that this additional byte is typically known as the alpha element and is particularly employed for lucidity.

Figure 3.2: Represenation of the 32 bit Colour (Willamette 2013).

16 bit Color: Every pixel is depicted by employing sixteen bits or two bytes with five red bits, six green bits, and five blue bits and an overall amount of feasible hues of approximately sixty-five thousand (256*256) (Omer and Werman 2004; Willamette 2013).

Figure 3.3: Represenation of the 16-bit Colour (Willamette 2013).

8 bit Color: There are 256 various twenty-four-bit hues, alternatively sixteen or thirty-two bit) that are chosen from the sixteen million options. These 256 hues are known as the colour scheme. Within the picture, every pixel is signified by one byte, which is not a hue but the position of the hue within the scheme (Omer and Werman 2004; Willamette 2013).

❖ Image Compression

Image compression a method that uses ‘mathematical formulas to analyse and confine image data, resulting in smaller file sizes’. The point of image condensing is to minimise the size of an image (Morkel et al. 2005).
There are two types of compression in images: lossy and lossless. Those two types run different procedures, although both of them save storage space (Ballard 2002; Morkel et al. 2005).

It is possible to produce files of a reduced size via lossy compression, and the process by which file size minimisation is facilitated operates by removing inessential image information from the initial image. In almost every case, the information that is removed is not discernible by the human eye. Ballard (2002) and Morke et al. (2005) note that the Joint Photographic Experts Group (JPEG) image format is a typical case of format that facilitates this form of image compression.

In contrast to a lossy compressed file, a file compressed according to lossless compression denotes information in mathematical formulae and, on this basis, avoids stripping the initial image of data (Ballard, 2002; Morke et al., 2005). According to the aforementioned scholars, two image file formats that utilise the lossless compression approach are the Graphical Interchange Format (GIF) and the 8-bit BMP (namely, a Microsoft Windows bitmap file).

❖ Image Compression and Steganography

The steganographic algorithm is selected partially based on the compression capacity of the image. Within deficit compression methods, the chances that the encoded content might be slightly misrepresented are greater as greater amounts of image information will be eliminated. This method is, however, beneficial for producing smaller picture file dimensions. Morkel (2005) explained that while lossless compression does not create smaller picture dimensions, the initial picture is preserved and the possibility of misplacing or misrepresenting the concealed data is lower.

❖ Image Format

A number of image file formats are suitable for steganography, among them BMP, JPEG, TIFF, GIF and PNG. In terms of hiding information, each of them offers certain advantages and disadvantages.

The two formats used in this research, BMP and JPG, are described in more detail below:

- **Bitmap Images (BMP)**

Microsoft created BMP to be the standard image file format for Windows operating systems. Although it has come to be supported across a number of additional file systems and operating systems, it has lost much of its initial popularity, due primarily to its large file size, a result of poor compression and verbose file format. In principle, losslee format such as bitmap files are organized into two main blocks, the header block and the data block. The header block, consisting of 54 bytes, can be further subdivided into Bitmap header and Bitmap information blocks.

The bitmap format has many advantages, for example, it can be in 1-bit black and white. Also, 8-bit greyscale and 16, 24 or 32-bit RGB color. Alpha channels are supported in new versions of BMP (Elgabar 2013).

- **Joint Photographic Experts Group (JPEG)**

JPEG is a compressed image file format which, unlike the GIF format, does not limit the amount of color in a file. This makes it especially useful for compressing photographic images, and very many of the large, colorful images one sees on the Web are JPEG files. Although JEPG produces colorful, high-resolution images, it is nevertheless a lossy format, and there is some loss of quality associated with the compression (Morkel et al. 2005; Elgabar 2013).
Table 3.2 below highlights the differences between BMP and JPG images.

Table 3.2: Comparison of BMP and JPG images (Elgabar, 2013).

	BMP	**JPG**
File Types	Windows Bitmap Or A map of bits	Joint Photographic Experts Group
Files Suffix	.BMP	.JPG .JPEG
File Size	Larger	Small
Resolution	Medium	High
Support Colour		16 Million Colour
Complexity	Very Simplistic	Quit Complex
Ideal for	Icons and Small Images	Photo
Colour Depth	1-32 bit colour	8-24 bit colour
Compression Algorithms	Lossless	Lossy

3.3. Techniques of Images Steganography

There are two main kinds of hiding techniques used in image steganography:

- Techniques used in the Image Spatial Domain.
- Techniques used in the Transform Domain.

The image domain methods enclose hidden dispatches in the intensity of the pixels directly. In the transform domain (known as frequency) methods, on the other hand, images will first be transformed and later the dispatch will be enclosed in the image (Morkel et al. 2005 and Huayong et al. 2011).

To make the hiding process more robust, those forms hide dispatches in areas of the cover image. Nevertheless, numbers of transform domain forms are independent of the image format and the enclosed dispatches might survive transforming between lossy and lossless condensing. Figure 3.4 shows the categories of image steganography:

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 3.4: Categories of Image Steganography (Morkel et al., 2005).

3.3.1. Image Spatial Domain Embedding

Spatial domain enclosing techniques are the most widely used ones proposed in the literature. In general, these approaches function on the principle of the parameter adjustment of the cover-image. By this means, the difference between the cover-image and the stego-image is tiny and hard to be sensed with the human eye (Ming et al. 2006; Huayong et al. 2011; Ghanbari et al. 2012).

The two most popular embedding methods in this domain are: LSB and BPCS Steganography.

❖ LSB Replacement & Matching

The LSB approach is characterised by the least number of indicators being left, and this is achieved by modifying a cover image's plain LSB, hiding the major significant bits, and leaving the statistical features of the cover image significance undamaged. The LSB-based approach is the hardest bit owing to the fact that it is difficult to distinguish between the cover-object and the stego-object in cases where a relatively small number of the cover object's LSB bits are modified (Raja et al., 2005; Ming et al., 2006).

Two main systems constitute LSB steganography, and these are sequential embedding and scattered embedding. The former embedding process operates on the basis of substituting each pixel's LSBs in sequential fashion with messages. The latter embedding process operates by employing a random sequence to facilitate the scattering of a message across the entire image; in this approach, the random sequence facilitates the control of the embedded locations. Ming et al. (2006) and Ghanbari et al. (2012) highlight the range of tools that incorporate this steganographic approach, examples of which are S-Tools, Hide and Seek, and Hide4PGP.

A stego-image is displayed in figure 3.5 in which the LSBs of the pixels in the upper half of the picture are altered by the presence of a message. Also displayed is the difference between the stego-image and cover image (Ming et al. 2006).

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 3.5: (a) Cover image; (b) LSB plane of cover image; (c) LSB plane of stego-image (Ming et al., 2006).

LSB matching is considered to be a modified version of LSB steganography. If the LSB of the cover pixel matches the secret bit, the pixel value is kept unchanged; otherwise, it is added or subtracted by 1 at random (Ming et al. 2006).

Each tool has its own feature. For example, S-Tool reduces the number of colours in an image to 32. Hide and Seek, however, works by making all the palette entries divisible by 4 and its

versions 4.1 requires that the image be 320*480 pixels and contain 256 colours (Ming et al. 2006).

- **BPCS Steganography**

The BPCS steganography functions through concealing information via substituting blocks as the picture's bit place is divided into identical sections of pixel-blocks. These pixel-blocks' dimensions are commonly 8*8 and can be categorized into descriptive and sound-like blocks with a threshold of α with α commonly equalling. According to Ming (2006), when the block exceeds the threshold it is regarded a sound-like block.

3.3.2. Transform Domain Embedding

The kind of file format attached to this domain ought to be expressed first in order to understand the steganography algorithms which may be utilised while embedding data in the transform domain. The JPEG file composition is a common image file composition on the web, due to the small size of the images (Morkel et al. 2005).

- **JPEG compression**

In order to condense a picture into a JPEG format, the RGB hue depiction is initially transfigured into the YUV format. Within this format, the Y element signifies the degree of light while the U and V aspects represent the hue. Examinations, such as Morkel (2005), have revealed that the human eye is significantly greater affected by alterations in the degree of light in comparison to alterations in hue. The precise configuration of the picture is then performed. The DCT is employed within the JPEG format. While almost identical alterations are feasible, such as the DFT, the DCT's mathematical configuration emphasizes the pixels to enlarge the pixel's locations within the picture. Morkel (2005) explained that the quantization stage of condensing is the concluding stage.
If the Steganography is based on DCT, as mentioned above, JPEG compression goes through different steps, as shown in Figure 3.6:

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 3.6: Data Flow Diagram Showing the General Process of Embedding in the Frequency Domain (Cheddad et al. 2010).

- **Using JPG in Steganography**

Owing to the fact that JPEG images employ lossy compression, it was first considered to be the case that steganography would not be compatible with them. This is because lossy compression impacts the image data elements that are subject to modification. It is critical to note that a key feature of steganography is that data is hidden in the inessential body bits; on this basis, given that inessential bits are left out when employing JPEG, commentators considered that the message to be transmitted would be damaged. Nevertheless, as demonstrated by Morke et al. (2005), compression algorithm characteristics have been formulated so as to create a steganographic algorithm that is compatible with the JPEG format.

3.3.3. Adaptive Steganography

Also referred to "Statistics-aware embedding", "Model-Based", or "Masking", this is an interesting formulation that was created on the basis of the two aforementioned methods. The method uses a statistical global property of the image at the moment prior to attempting to facilitate interaction with its LSB/DCT coefficients. Statistical features determine the places where modifications should be made. Additionally, customisation takes place according to random adaptive entries of pixels, and this is dependent on the cover image and pixel entry in a block with huge domestic standard deviation (STD). The last aforementioned process facilitates the prevention of smooth areas (namely, regions where colour is unchanged).

The concept of 'life in noise' reflects the degree to which information embedded in noise offers a considerable degree of utility. Its robustness has been demonstrated in terms of a range of processes, including cropping adjusting, compression, and image processing. As stated by Cheddad et al. (2010), the model-based method (MB1) generates a stego-image in accordance with a specified allocated model, and this is carried out by employing a generalised Cauchy allocation that takes place in the minimal deformity.

3.4. Algorithms used in Steganography

There are many algorithms that are used in steganography to hide data. Some of them use LSB steganography and some filter the image first. Many of them are going to be explained here: BlinHide, Hide Seek, Filter First and BattleSteg, Jsteg, Ouguess, F3, F4, and F5 (Umamaheswari 2010).

However, JSteg, F3, F4 and F5 algorithms work by embedding secret data in the transform coefficients that meet the requirements of both imperceptivity and robustness. The representative tools of these methods include JStegshella 2.0, JPHS, F5, Outguess, Steganos Security Suite 6.0. Some of these tools will be described in more detail below (Ming et al. 2006; Umamaheswari 2010).

3.4.1. BlinHide

The BlinHide is an uncomplicated method of concealing data within a picture as the data is concealed through being conveyed from the picture's top left corner and subsequently throughout. Ming (2006) and Umamaheswari (2010) explained that the lowest possible amount of bits of pixel colour is then substituted to correlate to the concealed content.

3.4.2. Battle Steg

This algorithm is the best one of them, as it performs 'Battleship Steganography'. It works by filtering the image first then uses the highest filter values as 'ships'. Then it shoots at the image randomly, and when it finds a 'ship' it clusters its shots around that hit in the hope of 'sinking' the 'ship' (Ming et al. 2006; Umamaheswari 2010).

3.4.3. Hide Seek

The Hide Seek algorithm distributes the message all over the image randomly. There is also another steganography tool called 'Hide and Seek', which uses the same technique. It uses a password to produce a random seed, then uses it to pick the first position to hide in. It will continue producing positions until it has finished hiding the message (Ming et al. 2006; Umamaheswari 2010).

3.4.4. Filter First

This is an algorithm which filters the image first before hiding it. It carries out the image filtering by using one of the inbuilt filters. Later it hides changes in the highest filter values first. This algorithm filters mostly the significant bits, and abandons the lower priority bits to be changed (Ming et al. 2006; Umamaheswari 2010).

3.4.5. JSteg

Jsteg works by embedding data in the LSBs of DCT coefficients, by skipping the 0 and 1 coefficient of a JPEG file. So pairs of coefficients are produced: …, -3↔-4, -2↔-1, 2↔3,. An obvious feature of JSteg is that the coefficient's frequency of occurrence trends to equal (Ming et al. 2006).

4.4.6. Outguess

Outguess works by embedding messages in the DCT domain. This algorithm has two separate embedding processes. First it identifies the redundant DCT coefficients that do not have a lot of effect on the cover image. Then it will choose the embedding bits to embed the message into them. This will be done depending on the information obtained in the first steps (Kharrazi et al. 2005).

4.4.7. F3

F3 is another embedding algorithm that decreases its value by 1 when the LSB of nonzero coefficient is not similar to the secret bit. It produces extra 0 that are undistinguishable from

those that are unused for embedding. However, F3 keeps the symmetry of 1 and -1. As a result, this algorithm embeds the affected bit when it produces a zero frequency (Ming et al. 2006).

4.4.8. F4

F4 works by mapping the negative coefficients to the inverted steganographic value. Also, to keep the histogram's symmetry, it repeats embedding to the secret bit of 1 (Ming et al. 2006).

4.4.9. F5

This method utilizes matrix-implanting methods to decrease the amount of required alterations to encode content of a specific extent and functions by encoding content bits into unsystematically selected DCT co-efficient. According to Kharrazi (2005) and Raja (2005), this method has two primary characteristics: matrix implanting and the permutation of DCT co-efficient prior to implanting. Matrix implanting reduces the amount of alterations underwent by the DCT coefficients. Additionally, the DCT coefficients are permuted to cause the altered coefficients to be transmitted throughout the overall picture (Fridrich et al. 2002; Kharrazi et al. 2004).

The description of the F5 algorithm, version 11, shows that the program is able to accept five inputs:

- The stego-image Q's Quality factor
- The input file (GIF, JPEG, TIFF, or BMP)
- The name of the output file
- Secret message contained within the file
- User password, which is to be utilised as a seed for PRNG
- The comment, which is placed beneath the header.

(Fridrich et al. 2002 and Kharrazi et al. 2005).

3.5. Available Image Steganographic Tools

There are a huge number of steganographic tools that are available on the Internet. Some of them are open source and others are commercial products. To evaluate and reviewed most of them, Hayati et al., (2007) have surveyed categories of 14 steganographic tools. Different open source products and 34 commercial products were examined. Those tools are displayed in Table number 3.3. For the open source products, JPEG and BMP are the favourite picks as a cover medium, with 9 products offering the functionality to embed in these image forms. The next most commonly targeted image format is GIF, where F5, GifShuffle and Mandlesteg are useful. The majority of the 14 tools enclose data in the spatial domain, i.e. by altering pixel values, whereas dc-Steganographic, F5 and Outguess enclose in the convert Domain, i.e. by manipulating the convert domain coefficients. These three tools all modify the Discrete Cosine convert (DCT) coefficients to enclose the hidden information.

Table 3.3: Image Steganographic Tools with Open Source Code (Hayati et al. 2007).

Image Steganographic Tools	JPEG	BMP	Others	Embedding Approach	Production
Blindside		Yes		SDS	Yes
Camera Shy	Yes			SDS	Yes
dc-Steganograph			PCX	TDS	
F5	Yes	Yes	GIF	TDS	Yes
Gif Shuffle			GIF	Change the order of the color map	Yes
Hide4PGP		Yes		SDS	Yes
JP Hide and Seek	Yes			SDS	Yes
Jsteg Jpeg	Yes			SDS	Yes
Mandelsteg			GIF	SDS	Yes
OutGuess	Yes		PNG	TDS	Yes
PGM Stealth			PGM		Yes
Steghide		Yes		SDS	Yes
wbStego		Yes		SDS	Yes
WnStorm			PCX		Yes

TDS - Transform Domain Steganography
SDS - Spatial Domain Steganography (LSB Replacement and LSB Matching)

Within freeware or shareware goods, the typically concealed picture is represented by the following sequence: BMP, JPEG, GIF, PNG, PCX, TGA, TIF, PPM, and DIB. Overall, twenty devices are employed to conceal data within the BMP picture, ten devices for the JPEG, and nine devices for the GIF. Of the thirty-four devices, seventeen are being created or in other words, the online like is available such as the Contraband, Contraband Hell, Crypto123, Gif it Up, S-Tools Camouflage, Jpegx, Hide and Seek, InThePicture, Steganos Hermetic, Stego and

Dound, etc. According to Hayati (2007), StegMark is particularly pertinent as it allows information to be encrypted into several formats: JPEG, BMP, PNG, GIF, TIF, and TGA.
Some of the image steganographic tools are described in more detail below:

3.5.1. XSteg

The Xsteg is a graphical front end. It is utilised for the automated detection of JPEG's stego images. Indeed, this was utilised in 2001 by Niels Provos, and the more reliable edition of the tool can be used with the UNIX OS (operating system). Nevertheless, the version that is used for Windows remains in its beta stage and has remained somewhat unreliable with regard to the StegDetect.

Among the precise aspects devised by the program is the means to ascertain the kind of system that has been utilised to cover up the message and hide it, according to Walsh College (2009).

Additionally, it contains these scan options:

- JP Hide and Seek (Unix and Windows).
- J-Steg
- F5 (header analysis).
- Outguess 01.3b.
- Invisible secrets.
- Camouflage, and appendX.

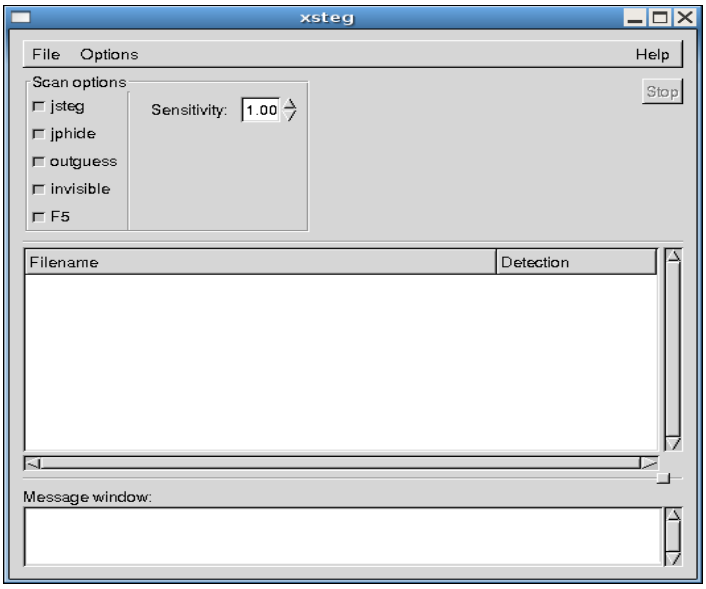


Figure 3.7: The Graphical User Interface (GUI) of XSteg Tool.

Derek Upham devised the **Jsteg**, which is able to hide information within the JPG image composition. Several improvements can be seen in Version 1.0, such as: the potential number of data that the JPG can cover beforehand; 40 bit RC4 encoding; and JPG options, which are user-selectable (i.e. degree of condensing) (Walsh College 2009).

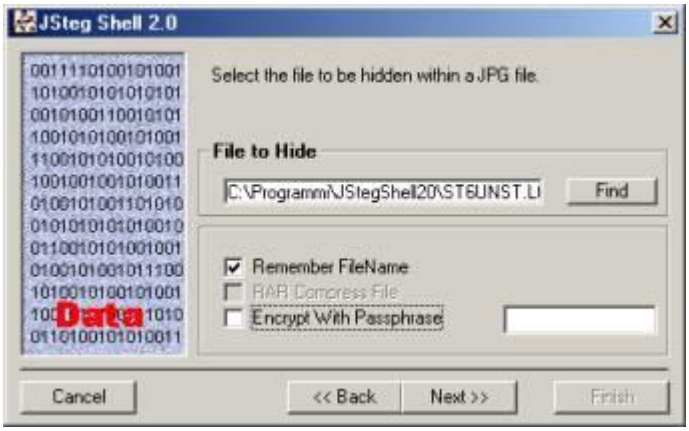


Figure 3.8: The GUI of the JSteg tool.

3.5.2. JPHide

JPHide is an application that can veil an image, and it does so using a visual image in the form of a JPEG file. This application is initiated so that not merely is it able to place a file 'under-cover', but furthermore, it is able to do this in a means by which the file that is hidden exists within the host file itself (Latham 1999; Walsh College 2009).

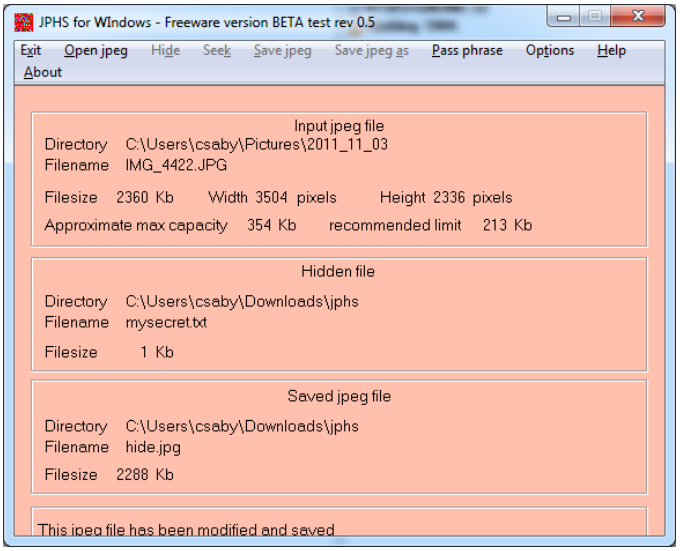


Figure 3.9: The GUI of the JPHide tool.

3.5.3. OutGuess

This steganographic regime exists as UNIX source code. The latest released version contains the ability to preserve statistical characteristics of the masked image (OutGuess 2003).

It works by embedding messages in the DCT domain. This algorithm has two separate embedding processes. First it identifies the redundant DCT coefficients that do not have a lot of effect on the cover image. Then it will choose the embedding bits to embed the message into them (Kharrazi et al., 2004).

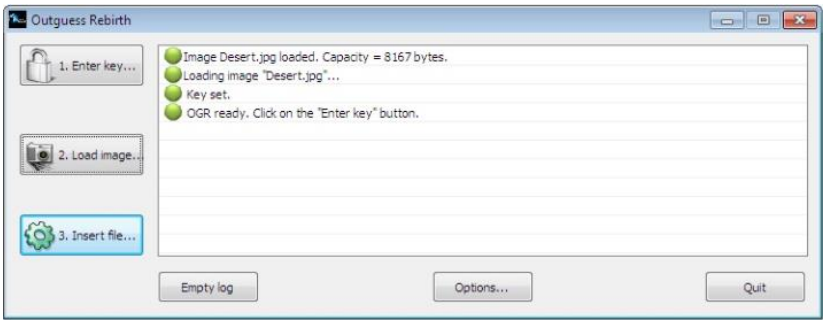


Figure 3.10: The GUI of the OutGuess Tool.

3.5.4. OpenStego

OpenStego is written entirely in Java script and is an open-source program disseminated under the conditions stipulated by the GNU-accepted Public Licence v2.0. It should be able to opperate on every Java-supplied platform, and has been tested on both Linux and MS Windows in the past. The system provides a further safety net in the form of password-based encoding and employs plug architecture wherein an array of plug-ins is created for a number of Steganographic algorithms. At the current time, Ipend Stego is able to support two plug-in Random LSB (a randomised LSB) as well as a LSB (Utilising Least Significant Bit of Image Pixels). According to OpenStego (2010), the program is also able to provide additional forms of images such as TIF and BMP, among others.

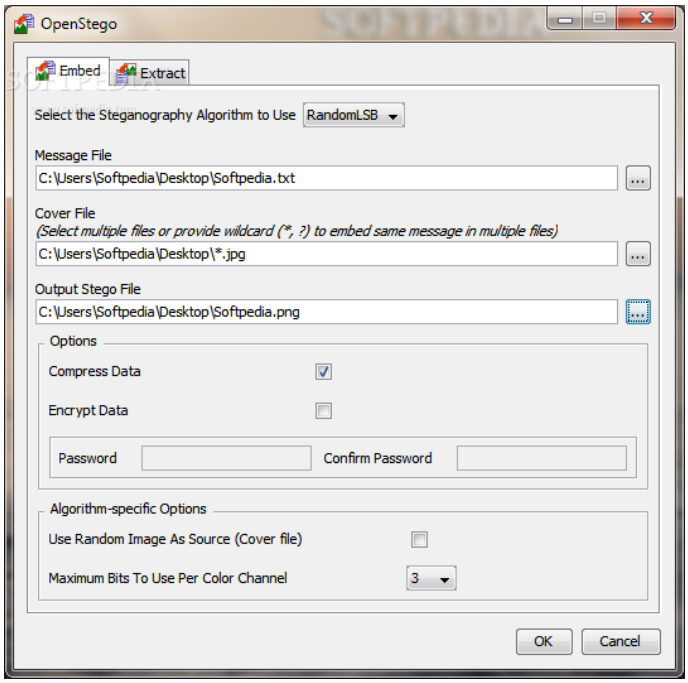


Figure 3.11: The GUI of the OpenStego tool.

3.5.5. S-Tools

S-tools or steganography tools program is a further application that permits both image and audio files to be concealed within other files of audio of image data. The file that contains the other, the base file, needs to be a WAV format file or else a GIF or a BMP file, and certain forms of image or audio files can, indeed, be hidden within the base image. The software program utilises a drag-and-drop technique. According to Walsh College (2009), this results in a manager kind of program that must be opened with the S-tools program.

It may be employed for both revealing and hiding purposes too, and utilises four kinds of encryption algorithm.

- Triple DES
- MDC
- DES
- IDEA

It is unable to deal with JPEC as either a cover or a hidden image. Nor indeed can in work with any 16-bit or 32-bit bitmaps, nor RLE-4 encoding. It is, however, able to utilise GIF images for the cover image, which BMP remains the format for the output image.

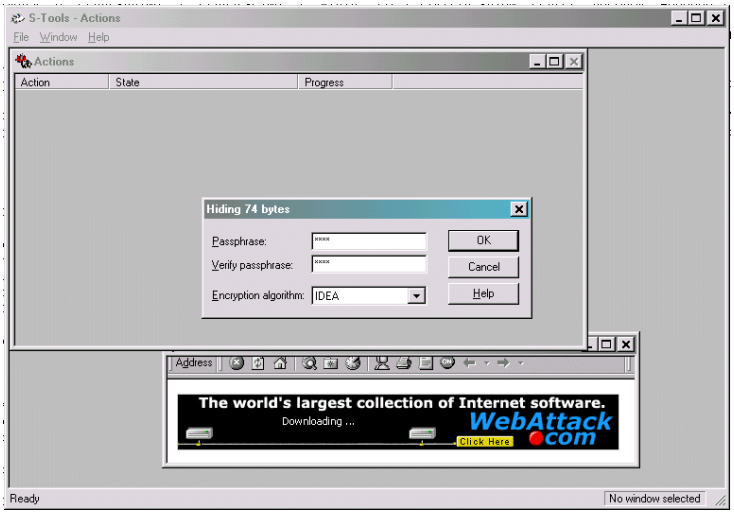


Figure 3.12: The GUI of the S-Tools.

How does the program work?

To comprehend the mechanics behind the revealing process, an understanding of the mechanism that is utilised by the S-tools must first be attained. The LSB algorithm is utilised by the S-Tools in hiding information through the manipulation of the image. The secret message is then distributed within the LSB of the colours within the cover image, though this is dependent on the type of image in question.

El-Emam (2007) has devised an S-Tools algorithm by which the capacity for embedding can be increased by 75%.

3.5.6. Version 2.0 – Hide in Picture (HIP)

A new freeware program was launched in 2001 by Davi de Figueirdo which is able to utilise bitmap images in order to embed all files as a hidden message. According to Rapid Deployment Software (2001), this required passwords for the retrieval of subliminal messages. This is likely a difficult process with regard to acknowledging a subliminal message within the image, unless the password is already known.

The DOS/Linux environment can be utilised for both the retrieval and guiding of information. This 2.0 vision is the fourth edition and remains the only one which has a GUI, therefore making the program compatible with Windows OS. Thus, generally, it has superior documentation and cleaner code, according to Rapid Deployment Software (2001).

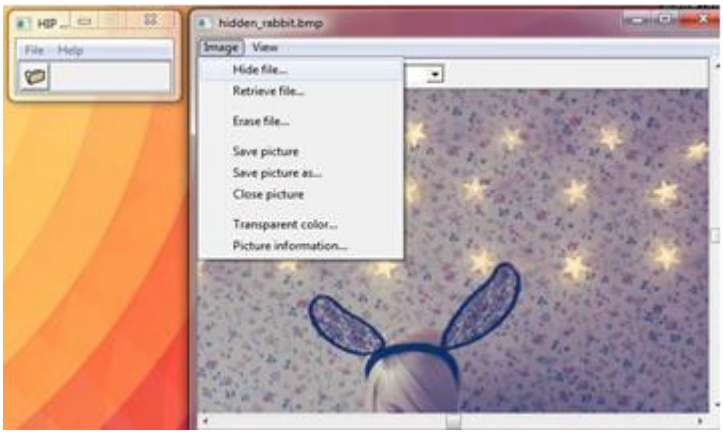


Figure 3.13: The GUI of HIP.

3.5.7. Invisible Secret

After its release in 2002, the program remains a commercial tool owned by NeoByte Solutions. The tool is deemed to be easy to use because of the functions it offers and its simple interface. The most contemporary version of Invisible Secret, version 4.7, can be used with the Windows 7 OS. Indeed, the program itself is a suite (East-tec, 2013) something that includes a series of security functions. Some of these functions are steganography IP to IP password transfer, email encryption and cryptography. Files in JPEG, BMP, HTML, PNG and WAV formats can be made to hide invisible secrets with the use of eight different algorithms.

Figure 3.14: The GUI of Invisible Secret.

3.5.8. White Noise Storm

White Noise Storm is a hugely effective steganography program for DOS'. This tool shuffles the bits inside an image through an encoding scheme. It utilises the LSB method and extracts the LSBs from the mask image and keeps them in a file. The dispatch is encoded and applied to those bits to initiate a new set of LSBs. To initiate the new stego-image, the adjusted bits are then injected into the cover image (Johnson and Jajodia 1998).

3.6. Evaluation of Different Image Steganography Techniques

While image domain approaches utilise the least significant bits in the binary value of image pixels, conversion domain approaches convert image data to the frequency domain and hide the datat in that domain. Image domain approaches are quite simple compared to the other ones. Nevertheless, in general they are more sensitive to tiny alterations in the image such as resizing, filtering and squeezing. The transformation domain approaches, on the other hand, are exceedingly robust when it comes to alterations, as the data hiding capacity is lower than in the image domain approaches (KURTULDU and ARICA, 2008).

All of the previous mentioned algorithms for image steganography have both various weak and strong points and it is important to make sure that one uses the most appropriate algorithm for an application. For instance, F5 and Outguess encoding approaches have been successfully

assaulted, which means they are more vulnerable to being broken easily and effectively than other strong algorithms (Kharrazi et al. 2004).

However, all steganographic algorithms have to comply with a few basic requirements. According to Morkel et al (2005), the most important requirement is that a steganographic algorithm has to be imperceptible. Morkel et al (2005) propose a set of criteria to further define the imperceptibility of an algorithm.

These requirements are as follows:

- **Invisibility** must be met for a successful steganography image. The steganographic algorithm's invisibility is the greatest and most important necessity as the steganography's effectiveness is contingent on its abilities to not be clearly evident. Therefore, the algorithm is flawed if it is obvious that the picture has been altered.

- **Payload capacity** is also important, but dissimilar to watermarking the payload capacity needs only a small section of the copyrighted data to be included. Steganographies emphasize confidential discussions and, therefore, must have efficient encoding abilities.

- **Robustness against statistical attacks:** Statistical steganalysis is the application of sensing secret data by applying statistical tests on an image's information. Some of the steganographic algorithms may mark or make a 'signature' when enclosing data which can be easily sensed through a statistical analysis.

- **Robustness against image manipulation:** In the communication of a stego-image by trusted regimes, the image might undergo alterations by an active warden in an attempt to get rid of the concealed information. In another words, image manipulation.

- **Independence of file format:** with a Varity of different image file compositions utilised on the web, it might appear doubtful that just a single type of file composition is continuously communicated between two parties. The most affective steganographic algorithms hence acquire the ability to enclose data in any sort of file.

- **Unsuspicious files:** This category contains all properties of a steganographic algorithm that might occur in images which are not utilised normally and may lead to suspicion.

Table 3.4 compares LSB insertion in BMP and in GIF files and JPEG compression steganography according to the above requirements:

Table 3.4: Comparison of Image Steganography Algorithms, *Depends on cover image used (Morkel et al. 2005).

	LSB in BMP	LSB in GIF	JPEG Compression
Invisibility	High	Medium	High
Payload capacity	High	Medium	Medium
Robustness against statistical attacks	Low	Low	Mediu
Robustness against image manipulation	Low	Low	Medium
Independent of file format	Low	Low	Low
Unsuspicious files	Low	Low	High

The degree to which the algorithms comply with a given need is rated as high, medium or low. First, high level means the algorithm completely satisfies the need, while a low level shows that the algorithm has a flaw or weakness with respect to this need. A medium level shows that the degree to which the need is met depends on outside impacts (Morkel et al., 2005).
Unfortunately, among the algorithms that are evaluated here, there is not one that satisfies all of the requirements. Thus a trade-off will exist in most cases, depending on which requirements are more important for the specific application (Morkel et al., 2005).

- **LSB in BMP:** In cases where the desired process is to enclose a dispatch "message" in a "raw" image that has not been subject to manipulation with condensing – one example being a BMP – a compromise must be taken with regard to the way in which the message looks and the data quantum that can be enclosed. Morkel et al. (2005) explain that BMPs have the capacity to conceal messages of a relatively large size.

Suggested Applications: The application that is characterised by the greatest degree of suitability for LSB in BMP is one where the focus centres on the quantity of data that is to be sent as opposed to the security of the data.

- **LSB in GIF:** In terms of enclosing information, the quantity of data that can be concealed using this approach is not as great as is the case with the previous approach. This is due to the fact that the bit depth of a GIF image is restricted at 8. As highlighted by Tiwari and Shandilya (2010) and Morke et al. (2005), the outcome of this approach is closely comparable to the outcome of the previous approach.

  Suggested Applications: LSB in GIF is characterised by a considerable degree of efficiency as an algorithm in cases where a user intends to embed a relatively large quantity of information in a greyscale image (Tiwari and Shandilya, 2010).

- **JPEG Compression:** The process of activities associated with enclosing information in the course of JPEG condensing produces a stego-image that is very difficult to detect. This is attributed to the fact that the enclosing reserve is a component of the converted domain. JPEG is a frequently employed image format online and, as a consequence of compression, the file sizes are minute (Morkel et al., 2005).

  Suggested Applications: The JPEG image file composition can be effectively employed in the majority of steganographic applications.

3.7. The Drawback of the Current Image Steganography Techniques

Most of the existing steganographic methods rely on two factors: the secret key and the robustness of the steganographic algorithm. Table 3.5 shows the drawbacks of current steganography methods.

Table 3.5: Drawbacks of Current Steganography Methods (Cheddad et al. 2010).

Method	Descriptions
Spatial Domain Techniques	Large payload but often offset the statistical properties of the image. Not robust against lossy compression and image filters. Not robust against rotation, cropping and translation Not robust against noise. Many work only on the BMP format.
DCT Based Domain Techniques	Less prone to attacks than the former methods at the expense of capacity. Breach of second order statistics. Breach of DCT coefficients distribution. Work only on the JPEG format. Double compression of the file. Not robust against rotation, cropping and translation. Not robust against noise. Modification of quantization on table.

3.7.1. Important Factors

There are three important factors that have to be taken into consideration when choosing a cover image (clean image) and embedding hidden data. These factors are: the length of the hidden message, the image format and the colour of the clean image (Johnson et al. 1998; Fridrich et al. 2001).

A factor of considerable importance is the concealed message's span, primarily because the lower the data content that is embedded into a clean image, the more minute the likelihood of it being detected. Of similar importance is the issue of cover image selection: specific types to avoid are images with limited colour variety, computer art, and images which convey meaning (for example, those including fonts). Certain steganographic specialists argue that grey-scale images are the optimal format for use as cover images, and an additional suggestion is that uncompressed scans of photographs or, alternatively, images taken with a digital camera –

namely, images which contain numerous colours – constitute viable candidates (Fridrich et al., 2001).

It should also be emphasised that image format is a matter of importance. This is due to the fact that the employment of a lossy file format, JPEG being an example, contributes to the loss of bits when reconstructing the embedded file as a result of compression. Johnson et al. (1998) note that the employment of lossless images, including BMP or GIF files, means that the comprehensive reconstruction of the concealed file can take place.

3.8. Performance Specification of Image Steganography

Three characteristics are especially significant when studying steganographic systems: robustness, security and capacity. The relationship between them can be explained with the steganography triangle, shown below in Figure 1.15. To enhance one element, two other elements must be sacrificed. For instance, if capacity is enhanced, security may be at risk of being sacrificed (Provos and Honeyman 2003; Huayong et al. 2011).

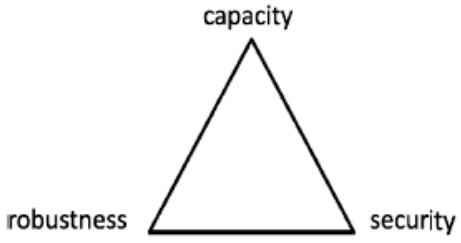


Figure 3.15: The Steganography Triangle (Huayong et al. 2011).

- **Robustness** suggests an encrypted content's capabilities to address either an intentional assault by a dubious other individual or the sound of malfunctioning flaws during another stage of the conversion. According to Provos and Honeyman (2003) and Huayong (2011), the robustness is determined by the concealed content's capability to withstand an attack. For example, if the picture appears only slightly affected, then it can be determined as efficiently robust.

- **Capacity** suggests the highest amount of bits that can be incorporated into an image without compromising the stego-image's invisibility and wholeness. The masking picture employed to generate a stego-image serves as the medium for converting the

encrypted content. Provos and Honeyman (2003) and Huayong (2011) explained that the masking image, similar to alternative data mediums, is essentially defined by its capacity.

- **Security** can be defined as the encrypted converter's capability to be invisible, which is the essential aim of steganography. Dissimilar to alternative methods of discussion, the security is compromised if it is noticed during its transmission. Provos and Honeyman (2003) and Huayong (2011) explained that the primary necessity of any steganographic program is invisibility.

3.9. Popular Image Steganalysis Methods

A number of steganalysis test methods may be utilised when detecting hidden messages, as shown in table 3.6 below. The author intendeds to look at the selected targeted attack in accordance with a specific and predetermined order (Wang and Wang 2004).

Table 3.6: Popular Steganalytic Methods (Wang and Wang 2004).

Steganalytic Methods	Description	Targeted Steganographic Techniques
RS steganalysis	Sensitivity of dual statistics based on spatial correlation of pixels to LSB randomization due to steganographic embedding is used in analysis.	Various LSB modification techniques
PoV-based Chi-square test	A Chi-square test checks whether the occurrence of each pair of values tends to become equal, indicating some data is embedded.	Steganography based on swapping pairs of values of pixel gray levels, colors, or DCT coefficients
Palette checking	Peculiarity in palette ordering is a clear sign of systematic modification.	Steganography in palette images
RQP method	Method based on analyzing the increased number of close-color pairs caused by embedding.	LSB embedding in true-color images
Check JPEG compatibility	Method detects unusual departure from the JPEG signature inherent in images initially stored in JPEG format.	Space-domain steganography using images initially stored in the JPEG format
Histogram analysis	Method reveals discreteness or periodicity in particular coefficients due to quantization-related modification.	QIM or other quantization-related embedding methods
Universal blind detection	Statistical quantities constructed using high-order statistics, and a detection model established with the threshold obtained in a training process.	Various steganographic techniques

3.9.1. Pairs of Values (X2)

Johnson and Jajodia (1998), demonstrated an example of powerful test analysis, an initial (chi-square) approach of steganalysis wherein the assumption of a binary message has been taken into account in general terms, assumed as an i.i.d presumption of either a 0 or 1 probability value.

The presumption of Independent and Identically Distributed (IID) was made on the bases of the subsequent logical phase while enlarging the analysis utilising a Markov chain, aside from contributing the detection-theoretic process with statistically subordinate information.

A number of steganographic systems can be detected by utilising tests like: S-YTools, EZ Stego, JSteg and Steganos.

3.9.2. Measures of Binary Similarity

This technique is, unlike $X^{2,}$ utilised in order to distinguish the cover and stego image from one another with the use of 7 and 8-bit planes within a given image. According to (Avcibaş et al. 2005) this is founded on the concept that systems of steganography have imprints or 'footprints' among the bit plates of LSB, and these may be employed for detection.

3.9.3. Regular and Singular Scheme (RS)

The RS algorithm was, according to Fridrich et al. (2001), addressed as targeting the LSB non-sequential hiding within digital pictures. This was able to ascertain and tell the magnitude of the hidden messages though an assessment of LS's lossless capacity within the LSB and changes in the position of the LSB plane. Nevertheless, the process only operates when used with images and it demands more than 0.005 bits for each pixel.

Equation 3.1 as seen below shows the detection algorithm utilised in RS. Here, the ratio between P (pixels), the number of all couples of neighbouring colours, is shown in equation 3.1. Additionally, U is the number of unique image colours.

$$R= \frac{P}{\binom{U}{2}} \qquad (3.1)$$

3.10. Classification of Image Steganalysis

Image steganalysis are dividied into two classes: targeted steganalysis and blind steganalysis (Nissar and Mir 2010; Suryawanshi and Mali 2015).

The classification is based on whether the signature of the steganography technique or the statistics of image is used to detect the presence of hidden messages in images embedded using steganography.

3.10.1. Targeted Steganalysis

Is able to extract the hidden code or remove the hidden ratio assimilated by understanding the algorithm utilised by stenographic systems. The superior decision algorithms are sample pair analysis (SPA), different image histogram (DIH), LAM and RS within LSB stenography. The algorithm proposed by Pevny and Fridrich (2005) is a good example for detecting those techniques used in steganography which are able to conceal information within the DCR image domain. The benefit of utilising such a targeted steganalysis is its soundness and precision, according to Lou et al. (2008).

Nevertheless, the shortcoming of utilisation of exacting steganalysis is the challenge that comes with utilising it for the detection of a stenography system employed when hiding the message in images.

Fridrich et al. (2001) created the RS steganalysis: a dependable and precise technique for identifying the LSB of randomly encryption within an image. The RS Steganalysis' theoretical outcomes reveal a previously undiscovered prediction of the appropriate dimensions of concealed content that has been encrypted through LSB. For example, high standard pictures taken with scanners or digital cameras have been identified as needing below .005 bits/pixel to be invisible. On the other hand, any figure above .005 bits/pixel would be clearly noticeable.

Another steganalyic technique that can accurately identify concealed content and predict appropriate dimensions for JPEG images employs F5 (a steganographic algorithm) (Jessica, Golijan, and Hogea 2002). This technique primary utilizes a prediction of the masking image histogram that is obtained via the stegoimage through decompressing the stego image, cutting the image into 4*4 pixels to eliminate quantization within the commonly occurring region and then recompressing the image through employing identical standard characteristics as the

stego-image. The amount of comparable alterations that F5 proposes is contingent on the employment of the minimal square fit, which is determined through contrasting the chosen DCT coefficient's predicted histograms to the stego image's histograms. The outcomes of the investigation reveal that comparable alterations of 10% of the employable DCT coefficients can be easily identified. This technique was piloted on various series of images that featured original and encrypted JPEG and BMP images.

Substantial advancement has been made in methods for identifying steganographic algorithms through substituting the LSB scheme. Zhang, Cox, and Doeer (2007) created a specific steganalysis algorithm that manipulated the information that once LSB has been replicated, picture's regional mazima of degrees of grey or hue histogram is reduced while the regional minima is heightened. As a result, the total of the overall variations between regional and nearby exterma within the stego image's heightened histogram is not the same size as the masking picture. The investigation examined two datasets with two thousand pictures and the outcomes revealed that the suggested technique was better than other modern algorithms, specifically if the pictures had substantial noise or had not been compressed like high-resolution scans of pictures or digital recordings.

A specific steganalysis technique was created by Tan and Li (2012) to alter the edges of a steganography picture according to the LSB corresponding reconsidered through the B-spline attachment. Tan and Li (2012) highlighted that re-altering the stage of edge modified picture steganography by reconsidering the LSB corresponding creates contorts the pulse of the histogram's lengthy exponential tail that includes the total variation of pixel matches. This argument can then suggest a specific steganalytic technique founded on the B-spline attachment. The outcomes of the investigation reveal that the suggested technique produces superior outcomes for identifying stego-images that have a small encryption frequency. The primary function of this technique in contrast to modern blind steganalyzers, for example SPAM or SRM, is evident. Additionally, this technique can precisely predict the limits employed within concealed information encryption processes and can isolate the stego-image through separating block units of various dimensions.

3.10.2. Blind Steganalysis

A range of aspects of commonality exist with regard to image blind detection steganography and pattern classification, the latter process focusing on two-class classification. The defining characteristic of blind detection is that it is purposed with the classification of designated images into two categories: original (or cover) and stego images. As described by Luo et al. (2008), a number of the blind image steganalysis techniques that are currently being used initially extract image features and, following this, choose or formulate a classifier. Following this, the classifier is trained by employing the properties extracted from training image sets. Finally, the properties are subject to classification.

Luo et al. (2008) have provided a more rounded framework of blind steganalysis tentatively which consists of the following major parts:

- **Image pre-treatment:** Treating the selected picture prior to removing characteristics; for example, transfiguring the RGB picture to a grey colour scheme, cutting the image, JPEG compression, DCT or DWT transfiguration, etc. These procedures are meant to enhance the categorization process.

- **Feature extraction:** Removing detailed characteristics or primarily identifying characteristics that are vulnerable to encryption or alterations. The characteristics must be identified and built to include vectors of decreased sizes that will reduce mathematical complications of instructing and categorizing.

- **Classifier selection and design:** Identifying or creating suitable categorizes according to the removed characteristics, utilizing a substantial series of pictures with recognized categories to instruct categorizers, and securing essential guidelines for categorizers that will be employed within the categorization process.

- **Classification:** Manipulating the concluded categorizers by providing biased pictures and requesting that these pictures be categorized as either stego or unaltered pictures.

Lie and Lin (2005) proposed a feature classification technique, based on the analysis of two statistical properties in the spatial and DCT domains, to blindly (i.e., without knowledge of the steganographic schemes) to determine the existence of hidden messages in an image. They

have adopted the nonlinear neural classifier to be effective in class separation. Also, they have established a database composed of 2088 plain and stego images for evaluation. Based on this database, extensive experiments were conducted to prove the feasibility and diversity of the proposed system.

Chen et al. (2006) formulated a new steganalysis technique on the basis of a statistical examination of an empirical matrix (EM), thereby facilitating the detection of a concealed message in an image in a different way. In order to extract properties constituted of two parts – one, the moments of PH and, two, the moments of the characteristic function of PH – the technique employs a projection histogram (PH) of EM. It is notable that the approach incorporated properties extracted from prediction-error image – attributed to Shi et al. (2005) – for the purpose of facilitating performance enhancement. Additionally, the designated classified was SVM. A test database was built and, using this a basis, a comprehensive test is conducted in order to examine various property categories and to facilitate a comparison with approaches in prior arts. Empirical tests demonstrated that the advocated properties were characterised by greater effectiveness than prior arts. Furthermore, this technique was shown to have the capacity to effectively facilitate the blind detection of data concealed according to a range of embedding systems.

Pevny and Fridrich (2006) built blind steganalysis systems for JPEG images that have the capacity to assign stego images to existing steganographic programmes. Every JPEG image is characterised through the use of twenty-three calibrated properties that are computed from the JPEG's luminance component. The majority of these properties are computed in a direct manner from the quantised DCT coefficients on the basis of their first order and higher-order statistics. Following this, the properties linked to cover images and stego images embedded with three contrasting relative message spans were employed in the context of administrated training. In order to build a group of binary classifier SVM with Gaussian kernel was employed; after this, the binary classifiers were combined into a multi-class SVM through the use of the Max-Win algorithm. The researchers published outcomes for six widely used JPEG steganographic systems, presented as follows: F5, OutGuess, Model-based steganography, Model-based steganography with deblocking, JP Hide and Seek, and Steghide. One of the most interesting outcomes generated from their technique is the success with which it approached the concept of double compressed images.

Zhou, Feng and Yang (2009) extracted two kinds of features of an image based on the good property of fractional Fourier transform (FRFT) coefficients of image histogram and the histogram of image FRFT coefficients.SVM is used as a classifier.

Yu, Li and Ping (2010) constructed nine statistical models from the DCT and decompressed spatial domain for a JPEG image. By calculating the histogram characteristic function (HCF) and the center of mass (COM), the energy distribution of each model as one part of our feature set is measured. Support vector machines are utilized to construct classifiers.

3.11. Image Steganalysis Classes

Four categorisations of image steganalysis exist, according to Schaathun (2012) as follows:

- **Visual:** The use of a human eye through a manual assessment, as well as assessing the change of the image with the use of the LBS plane, which is then plotted as an instance.

- **Structural:** On the basis of the signs perceived in the media representation/file type like the Hide and Seek tool.

- **Statistical:** Targeted steganalysis is efficacious for a certain system through the creation of a steganogram model. This is then employed to ascertain whether an image is stego or not.

- **Learning:** As a result of the challenge that exists with regard to ascertaining the statistical model through analytic means, the use of brute-force data analysis for the obtaining of an empirical model has been devised in this approach. This is extrapolated from the machine learning principle, as derived from artificial intelligence and can be said to be appropriate for utilisation for multifaceted steganography.

3.12. Steganalysis Available Tools

A number of some steganalysis tools available that can detect the presence of steganography programs are demonstrated in this section. These tools are available on the Internet.

Some research has demonstrated that when the type of steganography that was employed is known, the steganography detection tool works best (Kessler 2004).

In addition, the type of steganography tool found can directly suggest which stegnography software was used. For example, JP Hide-&-Seek and F5 algorithm might direct the analyst to look more closely at JPEG files, while S-Tools might direct attention to GIF, BMP, and WAV files (Kessler 2004).

3.12.1. WetStone

It is possible to utilise WetStone Technologies' Gargoyle (previously known as StegoDetect) for the purpose of detecting the availability of steganography software. Figure 3.16 provides an indication of the Gargoyle interface (WetStone Technologies, 2004; Kessler, 2004). The programme operates by using a registered dataset or, alternatively, a hash set, with respect to the files in the known steganography software. In turn, the programme facilitates a comparison of these files with the hashes of the files that are being searched. It is also notable that Gargoyle datasets can be employed for the purpose of detecting the availability of cryptography, instant messaging, key logging, Trojan horses, and password cracking.

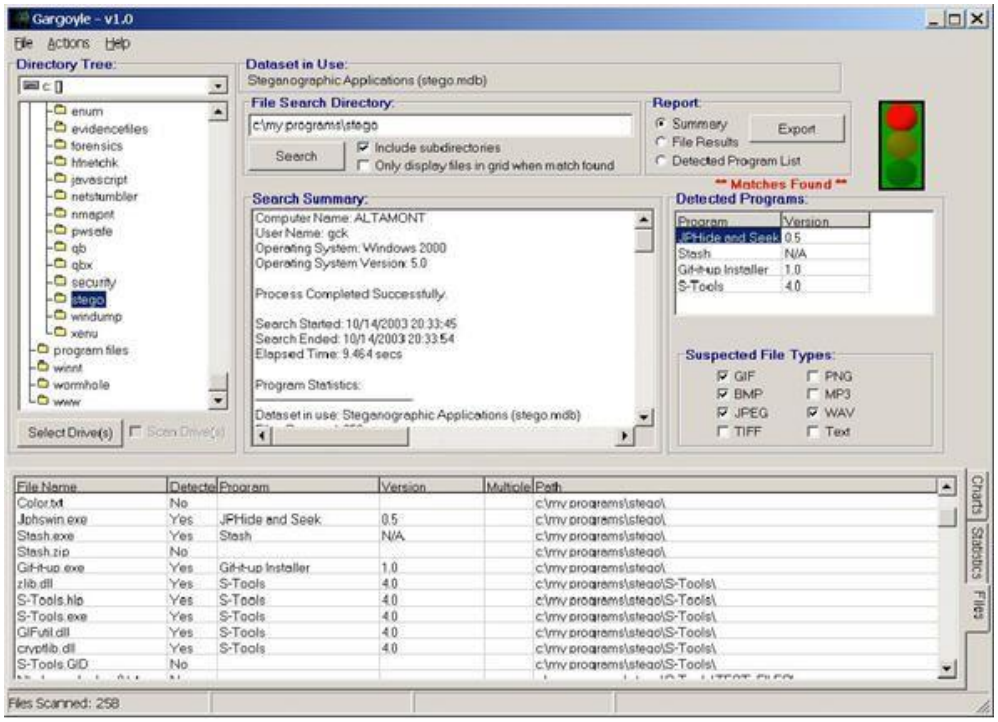


Figure 3.16: The Gargoyle Software Interface (Kessler 2004).

3.12.2. Stegdetect

Stegdetect is another tool used for detection, it was developed by Niel Provos in 2001. The tool has the ability to find hidden information in JPEG images using some steganography schemes, such as: F5, Invisible Secrets, JPHide, and JSteg **(**OutGuess 2003; Kessler 2004).

Stegdetect works by using linear discriminate analysis. The Linear discriminate analysis computes a dividing hyper-plane that separates the clean images from the stego-images. The hyper-plane is characterized as a linear function (Outguess 2003).

A graphical interface for Stegdetect is shown in figure 3.17. This shows the output from Xsteg which is a part from Stegdetect. In this example, the Stegdetect was used to examine two files on a hard drive; a clean JPEG image and its stego version. Note that the software detected the stego images, and suggested the used of the JPHide steganography scheme (Kessler. G 2004).

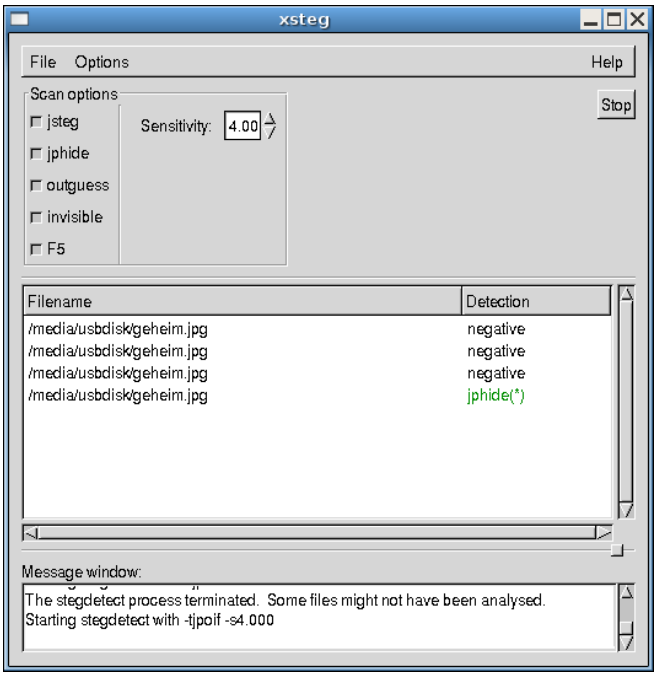


Figure 3.17: The Output from Xsteg When Examining Two Suspect JPEG Files (Kessler 2004).

Another companion tool to Stegdetect is Stegbreak, an automated tool for detecting steganographic content in images. It uses a dictionary attack against JSteg-Shell, JPHide, and OutGuess. It works by finding the password of the hidden data; however, it can only be applied to JPEG files (OutGuess 2003). At this time, the detectable schemes are: Jsteg, Jphide, Invisible Secrets, Outguess 01.3b, F5 (header analysis), AppendX and Camouflage (OutGuess 2003; Kessler 2004).

3.12.3. Steganography Analyser Signature Scanner: StegAlyzerSS v3.91

This tool is a commercial steganalysis tool which is generated with the use of stenographic assessment and research centre SARC, something that permits the scanning of suspicious images by forensic examiners with the utilisation of more than fifty-five signatures or patterns. Additionally, it may be utilised when seeing if there are any data or information hidden within the suspected image as part of a blind detection. It may be utilised for the extraction of clandestine or secret messages in addition to being used as an exclusive feature of StegAlyzerSS. Furthermore, it has, according to SARC (2004), been previously utilised for the efficacious conduct of the Cyber Science Laboratory (CSL) and the Defence Cyber Crime Institute (DCCI).

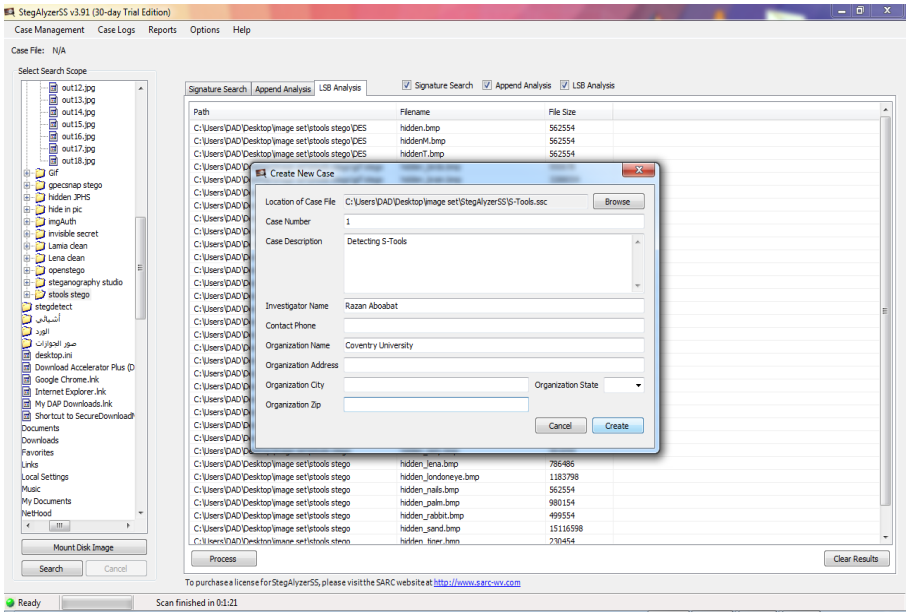


Figure 3.19: Creation of a new case for StegAlyzerSS.

3.13. Related Works

Previous research and related works are summarised in this section. They have been divided into two categories according to the types of features used for developing varieties of steganalysis methods.

Sectin 3.14.1. summarises previous image stegnalysis methods that generally used any image statistical features. On the other hand, section 3.14.2. describes others image steganalysis methods that are used any histogram features for detection.

3.13.1. Steganalysis Methods based on Extracting Image Statistical Features

RS steganalysis, an approach that facilitates the detection of the least significant bit (LSB) non-sequential embedding in a given image in a precise and dependable way, was formulated by Fridrich et al. (2001). The empirical finding gathered by the RS approach to steganalysis affords a novel prediction with regard to the current understanding of what the size of a secure concealed message is when utilising LSB embedding. In the case of high-quality images derived from scanners and digital cameras, the prediction demonstrated that messages needing

anything lower than 0.005 in terms of bits per pixel rate cannot be detected with the RS technique. Bit rates that exceed this limit can be detected using the technique.

A multispectral technique was advocated by Arvis et al. (2004) that considers the relationships between colour bands. Notably, this technique built on an approach previously formulated by Haralick et al. (1973), which operated on the basis of co-occurrence matrices. Despite the fact that it is an old method, the foundational role it has played means that one cannot avoid referring to it. In order to examine the extent to which the technique was efficient, Arvis et al. used it on a classification issue on VisTex and Outex – two image databases available online that are employed by the computer vision community.
In addition to this, the researchers built on the co-occurrence method by utilising two contrasting techniques: the fusion of texture and colour descriptors and, additionally, the quantisation of the colour image. The purpose of this was to facilitate a comparative examination of the three techniques with regard to colour image texture.

A generalized steganalysis technique was created by Lyu and Farid (2006) to identify encrypted content within digital images. This technique reveals through numerous scales or location picture decompositions, i.e. wavelets, the superior degrees and stage facts that are comparatively stable within a vast array of images, but are circulated according to the occurrence of encrypted secret content. The success of this method is evident through Lyu's examination of an extensive portfolio of images, which revealed eight various steganographic encrypted algorithms. Additionally, 360 proportional characteristic vector were identified that are responsive to the information encryption process of multidirectional Markov prototypes within the region of JPEG coefficients.

Forward variations were computed by Sun, Hui, and Guan (2008) into three routes: horizontal, vertical, and diagonal. These routes move in the direction of neighbouring pixels to acquire three-directional variations of pictures different from the original picture. These varying directional images are quantified to pre-determine a threshold for reducing repetitive data. Matrixes' mutual existence of images with various thresholds is employed as steganalysis characteristics. SVMs (support vector machine) that incorporate RBF kernel are utilized to determine categories.

Ghanbari. S *et al.* (2012) presented a steganalysis method using the gray level co-occurrence matrix (GLCM) and neural network. This method worked by extracting features from GLCM which are different between a clean image and a stego-image. The method was used as a combined method of steganography based on both the location and the conversion to hide the information in the clean images.
Features were used for training the neural network and the classification step was accomplished using a four-layer Multi-Layer Perceptron (MLP) neural network. The steganalysis method was tested on 800 standard image databases and it detected 80% of the stego images. Therefore, the efficiency is 80%.

Gong, R and Wang, H. (2012) have proposed a steganalysis method based on colour-gradient co-occurrence matrix (CGCM). The method was designed for GIF images, and based on the method presented by Zhao (2011).
CGCM is constructed with a colour matrix and gradient matrix of the GIF image, and they have extracted 27- dimensional statistical features of CGCM. Those features are sensitive to the colour-correlation between adjacent pixels and the breaking of image texture.
The support vector machine (SVM) technique takes 27-dimensional statistical features to detect hidden messages in GIF images. Experimental results proved that the proposed method is more effective than Zhao's (2011) method for several existing GIF steganographic methods and steganography tools. The method worked well especially for multi bit assignment (MBA) steganography and EzStego. Furthermore, the time efficiency of this method is much higher than the steganalysis method presented by Zhao (2011).

Holub and Fridrich (2013) proposed a different kind of statistical representation. In their scheme no co-occurrence matrix is used. Instead, neighbouring residual samples are projected onto a set of random vectors. From the projections a first-order statistic (histogram) is generated, which is taken as the feature. When a representation of multiple residuals is generated, it is called a projection spatial rich model (PSRM). Holub and Fridrich demonstrated that the PSRM can outperform and more accurately detect certain types of modern steganographic algorithms embedded in spatial, JPEG and side-informed JPEG domains than detection models based on feature sets.

Luo *et al.* (2014) were able to substantially reduce the high dimensionality of statistical features used in steganalysis by developing a feature selection method based on the Fisher criterion used in pattern recognition in order to reduce effectively the high dimensionality of the statistical features used in state-of-the-art steganalysis. First, they have evaluated the separability of each single-dimension feature in the feature space using the Fisher criterion, and these features were reordered in descending order of separability. Then, starting from the first dimension of the reordered features, as the dimension increases, the separability of each feature component was analyzed using the Fisher criterion combined with the Euclidean distance. Lastly, the feature components with the best separability were selected as the final steganalytic features.
Experimental results based on the selection of SPAM (Subtractive Pixel Adjacency Matrix) features in spatial-domain steganalysis and CC-PEV (Cartesian Calibrated feature extracted by PEVný) features in DCT-domain steganalysis showed that the proposed method can reduce the dimensionality of the features efficiently while maintaining the accuracy of the steganalysis.

A unique characteristic series for steganalysis of JPEG pictures was created by Holub and Fridrich (2015). The characteristics were designed as superior figures of quantized noise surplus that were acquired through decompressing the JPEG picture through employing sixty-four kernels of DCT (discrete cosine transformation). This technique can be understood as a estimation prototype within the JPEG industry that is a equivalent of the PSR (projection spatial rich) prototype. This technique's most beneficial feature is the minimal mathematical complications and size in contrast to alternatively rich prototypes. This technique also better results than formerly suggested JPEG region steganalysis characteristics.

A unique DFA (Discrete Firefly Algorithm) was developed by Chhikara and Singh (2015) that was founded on the wrapper method including a complex alpha framework in association with an t-test attachment characteristic chosen algorithm that was employed to identify the most pertinent decreased characteristic sub-series. The purpose of decreasing characteristic dimensions is to enhance the precision of categorizing invisible pictures as masking or stego images and the rate of the instruction program. The suggested method was utilized on pictures created through four steganography devices: nsF5, PQ, Outguess, and JPHS.
The results from popular JPEG steganography algorithms nsF5, Outguess, PQ and JP Hide and Seek show that the proposed method is able to identify sensitive features and reduce the feature set by 67% in the DCT domain and 37% in the DWT domain.

3.13.2. Steganalysis Methods based on Extracting Image Histogram Features

The stego content's histogram was presumed by Harmsen and Pearlman (2003) to be an elaborate representation of the noise PMF (probability mass function) and initial histogram. Within the distribution area, this elaboration was considered to be a HCF (histogram characteristic function) and noise feature function. The techniques for concealing pictures such as LSB, spread spectrum, and DCT were examined within the suggested guidelines and the outcomes revealed that these techniques were identical to a low pass filtering of histograms computed through reducing the HCF COM (centre of mass). These reductions were manipulated through a determined program of identification to categorize original and distributed spectrum pictures through employing a bivariate categorizer. Ultimately, Harmsen and Pearlman (2003) created a blind identification program, which employs exclusively statistics obtained from the original picture. The minimum employed to recognize steganographic pictures can be determined through computing the Mahalanobis length of the COM examination to the centre of instruction.

An easy and successful technique for blind image steganalysis was created by Dong and Tan (2008). This technique is founded on run-length histogram evaluations. Superior statistics of the feature operations of the three kinds of picture run-length histograms were chosen to serve as characteristics. A substantiate vector device was employed as a categorizer. The outcomes of the investigation revealed that the suggested technique is substantially better than the previous techniques for precise identification and universality.

Jia-jun (2009) proposed an image steganalysis scheme based on the differential image histogram in frequency domain. The difference is calculated in three directions, horizontal, vertical and diagonal towards adjacent pixels to obtain three-directional differential images for a natural image. Then the features for steganalysis are extracted from the DFT of the histogram of differential images and divided into low and high frequency bands. Support vector machine (SVM) with RBF kernel is applied as classifier.

Cai. K et al. (2010) formulated a new steganalysis system that can be used when facilitating the detection of LSB matching (also referred to as ""±1 embedding"). Notably, LSB matching is among the widest used steganographic approach. The researchers employed the histogram of difference image (the differences of adjacent pixels), and this is conventionally a generalised

Gaussian distribution with a centre at zero. Following this, it was utilised in the process of facilitating the derivation of statistical properties. The researchers generated a theoretical proof that the histogram's peak-value would fall following LSB matching embedding and, simultaneously, the renormalised histogram (namely, the ratio of the histogram to the peak-value) would rise. In turn, the peak-value and renormalised histogram were taken as properties for classification, and it is notable to acknowledge that the empirical outcomes indicate that the technique is more effective than a number of those that were previously in use.

Identifying the spatial domain LSB of the steganography should correlate with the grey picture (Xia et al. 2011). Zhihua argued that the original image possesses innate characteristics; for example, histogram, association of nearby pixels, and contingence of pixels that are not nearby. These characteristics will probably be assigned based on LSB corresponding; for example, the histogram is more even once LSB corresponding. The encryption of the content will reduce the contingence or association between both types of pixel. As a result, Zhihu argued that the histogram, nearby level of histogram, and run-length histogram should initially be removed. Once these characteristics have been removed, the substantiate vector device should be employed to differentiate and recognize the variations between the original and stego picture. The outcomes of this examination demonstrated that the suggested technique is able to precisely identify stego-images and is better than both former modern techniques.

Lou and Hu (2012) proposed a steganalysis method to resist statistical steganalysis using a reversible histogram transformation function-based LSB steganographic.
They have observed three vulnerabilities of the reversible histogram transformation function (RHTF) steganographic method and thus have proposed pixel grouping and a scheme of dynamical secret keys to improve the security of RHTF. Their experimental results illustrate that the proposed method is efficient in reducing the vulnerabilities of RHTF and maintaining the resistance to x2 detection and RS attack. At various embedding rates, most of the detection results of the stego-imges are as low as those of cover image.

3.14. Conclusion

This chapter illustrats the most common image steganography and image steanalysis algorithms and tools.

Table 3.7 shows a comparison of the reviewd image steganography tools.

Table 3.7: Summary of the Poupular Image Steganography Tools.

Tool	Image Format Support	Descriptions
XSteg	JPG	• UNIX and Windows versions are available. • it contains these scan options: ✓ JP Hide and Seek (Unix and Windows). ✓ J-Steg ✓ F5 (header analysis). ✓ Outguess 01.3b. ✓ Invisible secrets. ✓ Camouflage, and appendX.
JPHide	JPG	• Embedding messages by LSB. • Blowfish encryption. • embedding messages in the SDS domain.
OutGuess	JPG and PNG	• Available as UNIX source code. • embedding messages in the TDS domain.
OpenStego	PNG, BMP and TIFF	• UNIX and Windows versions are available. • Embedding messages using LSB and random LSB.
S-Tools	BMP and GIF	• Utilises four kinds of encryption algorithm. ✓ Triple DES ✓ MDC ✓ DES ✓ IDEA • Embedding messages using LSB. • Works with WAV audio files. • Passwords.
Hide in Picture (HIP)	BMP and GIF	• Blowfish encryption. • Passwords.
Invisible Secret	JPG, BMP and PNG	• Includes security functions such as: IP to IP password transfer and email encryption. • Works with WAV audio files.
White Noise Storm	PCX files	• Utilises the LSB method and extracts the LSBs from the mask image. • It's a DOS tool based. • Limited hiding capacity.

In addition, table 3.8 presents a comparison of the reviewd image steganalysis tools including their advantages and properites.

Table 3.8: Comparison of the Available Image Steganalysis Tools.

Tool	Descriptions
WetStone	• Developed by WetStone Technologies' Gargoyle • Operates by using a registered dataset or, alternatively, a hash set, with respect to the files in the known steganography software. • Previously known as StegoDetect. • Can be employed for the purpose of detecting the availability of cryptography.
Stegdetect	• Developed by Niel Provos. • Detect JPEG images using some steganography schemes, such as: F5, Invisible Secrets, JPHide, and JSteg. • works by using linear discriminate analysis
Stegbreak	• It uses a dictionary attack against JSteg-Shell, JPHide, and OutGuess. • It works by finding the password of the hidden data.
StegAlyzerSS	• Commercial steganalysis tool. • Developed by the research centre SARC. • Scanning of suspicious images by forensic examiners with the utilisation of more than fifty-five signatures or patterns.

Chapter 4

Methodology

4.1. Introduction

This chapter explains the methodology of the proposed detection system and the implementation process of extracting the CGCM and histogram features.

The CGCM features and its equations are described in detail, as well as the histogram features. In addition, the image database that was created is presented, along with the image formats and the steganography methods used to create stego images.

Likewise, the two classification methods used – stepwise discriminant analysis (DA) and multilayer neural network (MLP) – are introduced in this chapter.

4.2. Methodology of the Proposed Detection System

The proposed system has been divided into three stages (see Figure 4.1). An image database that includes clean and stego images was created to train and test the proposed system.

Stage two focusing on extracting statistical and histogram features for the purpose of detection.

Then, in stage three two classification methods were used for training and testing.

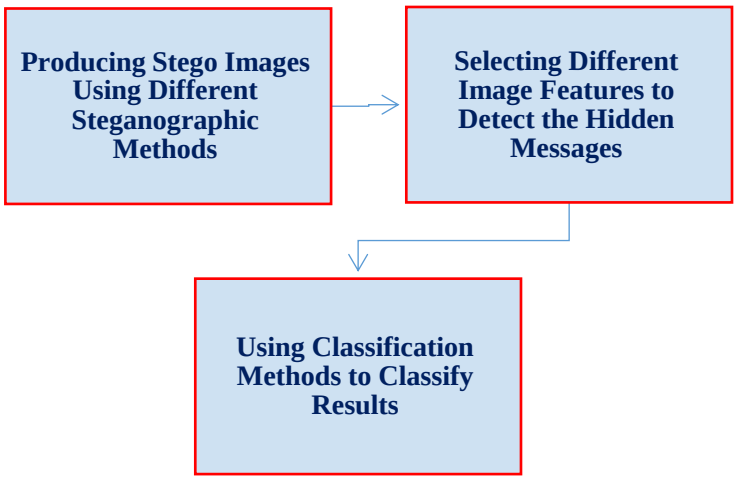


Figure 4.1: Methodology of the Proposed System.

4.2.1. Images Selection and Producing of Stego Images

The choice of clean images is important because it significantly influences the design of the stego system and its security. Images with a low number of colors, computer art, and images with a unique semantic content, such as fonts, should be avoided (Fridrich et. al, 2002).
The clean image should be ordinary, in order not to draw attention to the fact that it could be concealing information. Additionally, the cover image should not contain large blocks of one colour. If one were to use an image with such blocks, a change of one bit could result in an obvious distortion of the image (Kipper 2004).

Selecting the type of image substantially effects the creation of a protected steganographic program. Natural and uncompressed designs like BMP have the greatest dimensions for incorporating protected steganography. However, their apparent repetitiveness causes them to be extremely dubious. Some investigators do not even regard this style suitable for steganography as they argue that transmitting uncompressed images is the same as employing cryptography. However, Fridrich (2002) explained that the majority of steganographic goods accessible online employ uncompressed images styles or styles with only slightly compressed information; for example, BMP, PCX, GIF, PGM, and TIFF. Additionally, the employment of only slightly compressed files like JPEGs causes bits to be misplaced during the compression of the restoration of encrypted files. On the other hand, Johnson (1998) argued that employing only slightly compressed pictures like BMP or GIF permits the concealed file to be totally rebuilt.

All these factors were therefore considered, and the created images data-base includes grey, colour images and different formats to train and test the detection system.
Two steganography methods were used to generate the stego images: LSB and F5 algorithms. The S-Tools was used to implement half of the stego images using LSB steganography. This tool is free and available to use in the internet. However, it supports lossless image format only because LSB needs to be applied to lossless, pixel-based/spatially encoded formats, and doesn't work with lossy or frequency-domain compression algorithms.

Additionally, an LSB developed code was written using Python programming language to generate more sets of stego images. The developed code allows changing and using different hiding capacities in terms of embedding. In contrast, the S-Tools don't support any flexibility

in the hiding capacities because it's a common tool. It has a user interface; therefore, it does not offer any opportunity to change the hiding capacity.

The F5 steganography algorithm was also used to create sets of stego images with the JPG images formats.

The hiding process is shown in figure 4.2.

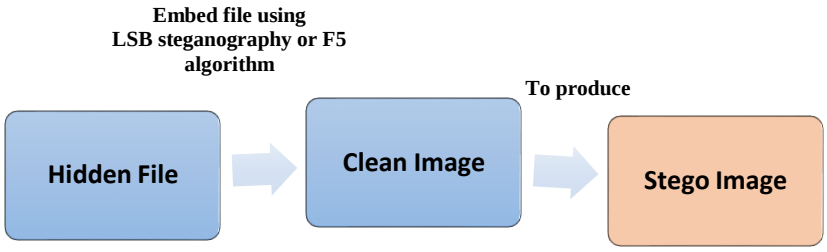


Figure 4.2: Stage One: Creating Stego Images.

❖ Lossless and Lossy Compressions

As depicted earlier, 'Image compression techniques are extensively used in steganography. Among the two types of image compressions are lossy compression and lossless compression' (Miano 1999; Nuruzzaman 2005; Reddy et al. 2011).

'Lossless compression is a class of data compression algorithms that allow the original data to be perfectly reconstructed from the compressed data' (Currie et al. 1996; Miano 1999). As examples of lossless formats, BMP and PNG were used in the present thesis to train and test the system.

Lossy compression is 'the class of data encoding method that uses inexact approximations for representing the content that has been encoded. Such compression techniques are used to reduce the amount of data that would otherwise be needed to store, handle and/or transmit the represented content' (Currie et al., 1996; Miano, 1999).

The JPG image format was used to train and test the system as an example of the lossy format. All images used were grey and coloured images (RGB) with 24 bits. Two different steganography methods were used to create the stego images. LSB were implemented using S-Tools.

Table 4.1 below shows the images' formats, steganography methods and the number of images.

Table 4.1: Images Used and Created in the Data Set.

Image Type	Number of Clean Images	Number of Stego Images	Range of Clean Images	Hiding Capacities	Steganography Method Used	Images Formats
Grey	600	600	599 kb to 1 MB	10% & 25%	LSB	BMP & PNG
Colour	600	600	599 kb to 1 MB	10% & 25%	LSB	BMP & PNG
Grey	600	600	100 kb to 1 MB	10% & 25%	F5 Algorithm	JPG
Colour	600	600	100 kb to 1 MB	10% & 25%	F5 Algorithm	JPG

As shown in table 4.1, the ranges of the clean images are fixed between 599 kb to 1 MB and 100 kb to 1 MB. The fixed range helps to improve the accuracy of the results.
Moreover, two different hiding capacities were selected to embed the hidden files, because detection ability relates to hidden file length. Clearly, the less information embedded into the cover image, the smaller the probability of introducing detectable artefacts through the embedding process (Fridrich. J et al., 2002).

Each steganographic method has an upper bound on the maximal safe hidden file length that tells how many bits can be safely embedded in a given image without introducing any statistically detectable artefacts. Determining this maximal steganographic capacity is a challenging task even for the simplest methods. Therefore, different capacities and hidden files were used to test the differences in the results and to train the system to deal with a variety of stego images (Fridrich. J et al., 2002).

The more diverse the image base is, the larger the training set will have to be in order to be representative (machine learning in steganslysis).

Figure 4.3. shows examples of the images included in the database created.

Figure 4.3: Examples of the Images Included in the Database.

Images included in the database were gathered form free image sources in the internet such as: PublicDomainPictures.net. Also, some of the images were taken by a personal Canon camera.

4.2.2. Core of the Detection System

The primary aim of the system is to distinguish stego images from clean ones.

The architecture of the system contains three core elements: tested images, the decision-making model and the process of determining whether the tested images are clean or stego.

Figure 4.4 illustrates the real implementation of the proposed system within the decision-making model. The model includes two parts: training and testing phases. In the training phase, 33 statistical features were extracted from the colour gradient co-occurrence matrix (CGCM) and 210 colour histogram features were extracted from the images. Then, the system trained using the stepwise DA and MLP classification methods, implemented using SPSS software.

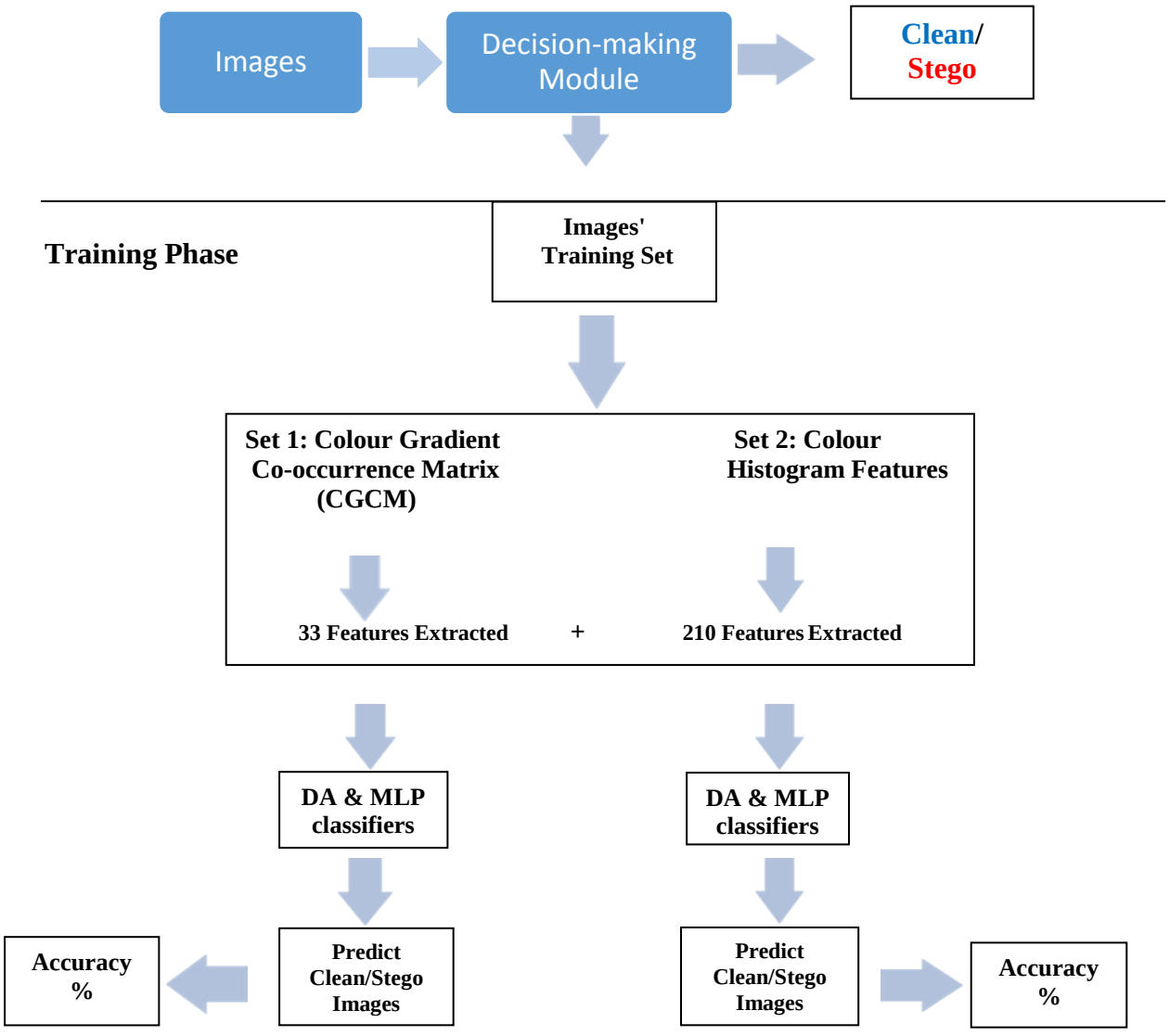


Figure 4.4: (a) Process of the Training Phase.

Testing Phase

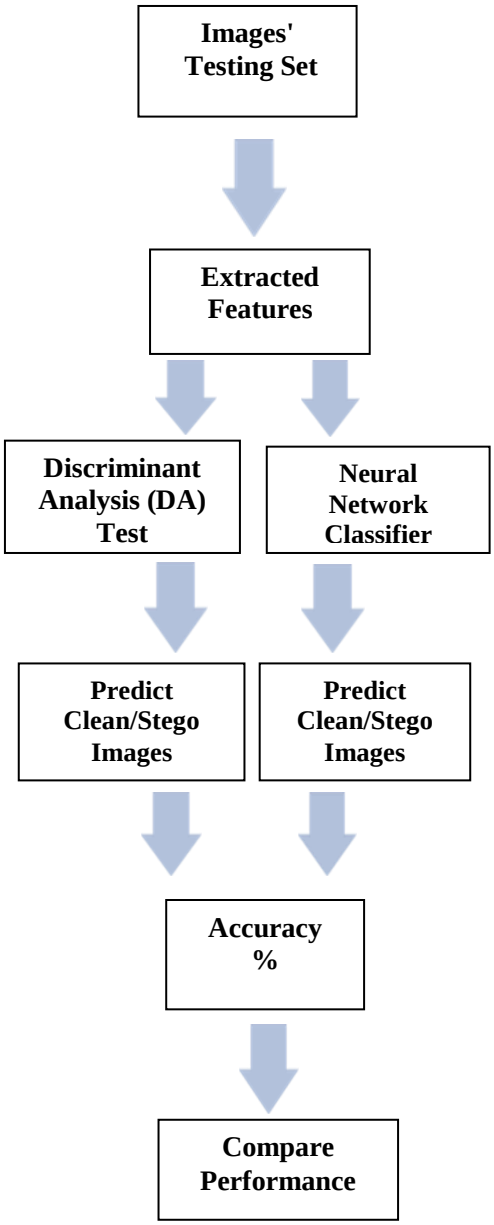


Figure 4.4: (b) Process of the Testing Phase.

Images in the created database were divided between the training and testing phases, 70% of the images were used for training and 30% of the images were used for testing.

4.3. The Tools Used

The steganography tools described below were used to create the stego image sets as shown in table 4.1.

4.3.1. S-Tools

This tool has been mentioned brefiley in section 3.5.5.

The S-Tools employs LSB replacement algorithm to conceal data through exploiting the picture and dispersing the concealed content in LSB regions of hue of the masking picture contingent on the kind of picture. The S-Tools permits audio and picture files to be concealed within alternative audio and picture files. Within this process, the original file must be either a WAV audio or BMP/GIF picture file and alternative audio and picture formats can be concealed within this original file.

This software is a drag and drop-type program, meaning a file manager-type program will need to be opened along with the S-Tools software (see Figur 4.5) (Walsh College 2009).

Figure 4.5: The Graphical User Interface of the S-Tools.

4.3.2. F5 Algorithm

Pfitzmann and Westfeld (2001) created the F5 steganographic algorithm in an attempt to create an idea and feasible technique for encrypting JPEG pictures that would allow substantial steganograpic abilities without having to give up safeguarding characteristics. This algorithm provides a substantial steganographic ability while also employing a straddling device.

According to Crandall (1998) and Westfeld (2001), this algorithm additionally supplies matrix encryption to enhance the implantation success, which decreases the amount of required alterations.

'The straddling mechanism used with F5 shuffles all coefficients using a permutation first. Then, F5 embeds into the permuted sequence' (Westfeld, 2001). The shrinkage only changes the values of coefficients, not their number. F5 delivers the steganographically-changed coefficients in the original sequence to the Huffman coder (Westfeld, 2001). Matrix encoding is considered a new technique, and F5 possibly is the first implementation of matrix encoding. If most of the capacity is unused in a cover image, matrix encoding decreases the necessary number of changes (Crandall, 1998; Westfeld, 2001).

To obtain the greatest outcomes from the matrix coding, F5 should exclusively be employed at the encryption frequency a low amount of figures of coefficients that are not zero. Previous texts typically consider that five per cent as the threshold for content durations. However, within F5 five per cent is closer to the maximum. Figure 4.6 shows an example of implementing the F5 algorithm.

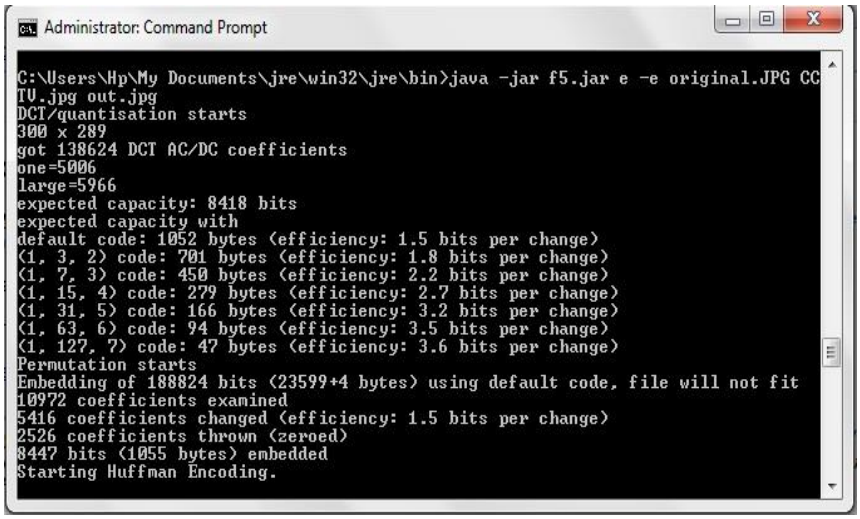


Figure 4.6: Implementing the F5 Algorithm through the Command Line.

4.3.3. Comparison between S-Tools and F5 Algorithm

Each of the two tool has its own features and limitations; table 4.2 shows a comparison between each of them.

Table 4.2: Comparison between the Two Steganographic Tools Used.

Tools	Advantages	Disadvantages
S-Tools	- It hides data using the LSB method. - Encrypts data using many encryption algorithms, such as IDEA (Walsh College 2009). - Compresses data before hiding. - It provides multi-threaded operations, which means multiple files can be hidden in one sound/picture and that data will be compressed before being encrypted, then hidden (Spychecker 2006). - It shows the size of the data that is to be hidden. - It shows the size of the available space for hiding data in the cover object (Walsh College 2009).	- It requires that both sender and receiver have a shared passphrase (Spychecker 2006). - The problem comes into being with how you share a passphrase and at the same time have different ones (Spychecker 2006). - It hides files in BMP and GIF image types only.
F5 Algorithm	- It offers a large steganographic capacity and uses straddling mechanisms (Westfeld 2001). - It also provides matrix encoding to improve the efficiency of embedding. As a result, it reduces the number of changes (Westfeld 2001). - F5 accomplishes a steganographic proportion that exceeds 13% of the JPEG file (Westfeld 2001). - F5 is able to decrease the embedding rate arbitrarily (Westfeld 2001).	- It does not have a user interface; it runs through command lines.

4.3.4. MATLAB

Morris (2007) explained that MATLAB is frequently employed as a medium for difficult sessions or laboratory assignments within the Image Processing programme of study. MATLAB is a device for evaluating information and conceptualization that posses strong imaging abilities and a unique programming language. Additionally, MATLAB was created

with the purpose of decreasing the complexity of matrix exploitation. METLAB's rudimentary dispersion could be further developed through incorporating an array of devices such as those prevalent to picture procedures like IPT (image-processing toolbox) or wavelet toolbox. This rudimentary dispersion and all included toolboxes, according to Morris (2007), will provide a substantial choice in operations, which will be induced through the command line interface.

- **MATLAB's Development Environment**

Windows MacOs and Linux systems are able to access MATLAB. According to Morris (2007), MATLAB's IDE can be categorized into five aspects of Workspace Browser, Command Window, Command History Window, Current Directory Windom, and an additional region for alternative Figure Windows, which depicts graphic images only when they are operative. The Command Window is feature that receives expressions and commands and produces the suitable outcomes. Morris (2007) explained that the series of factors created through a session composed the workspace and is depicted within the Workspace Browser. The methods employed during former sessions and the methods employed during the current session are depicted in the current directory window, but MATLAB can locate files also through the search directory. The search directory, according to Morris (2007), includes the most recent index with every toolbox or alternative routes downloaded by the system administrator.

4.3.5. SPSS

SPSS is a complex program employed by social scientists and alternative experts to conduct statistical evaluations. SPSS has continually been favourable and the SPSS: Analysis without Anguish Version 14.0 for Windows still fulfils SPSS program administrators' requirements. According to Coakes and Steed (2009), Version 14.0 provides a simple explanation to novices and permits individuals wanting to pursue more progressive evaluations to gradually and orderly progress.

4.4. Feature Extraction Process

Two different sets of images features were extracted and used in the detection system. Details of the extracted features and the process of the extraction are described below.

4.4.1. Colour Gradient Co-occurrence Matrix (CGCM)

The 33 selected features have been extracted from the CGCM.

CGCM, the joint statistic of the colour matrix and gradient matrix, takes into account information from both the colour correlation and the gradient among the pixels in an image (Gong and Wang, 2012).

- **Colour Matrix**

For an image I of size M*N, we define matrix C of size M*N. C is the colour matrix. C is defined as:

$$C(i,j) = IN(i,j) \ (i < [0, M-1], j < [0, N-1]) \qquad (4.1)$$

Where

C(I,j) represents the element of C at the ith row and the jth column.

In(i,j) is the colour index corresponding to i,j, the pixel of I [1,3].

- **Gradient Matrix**

F is defined as a gradient matrix, and it is given as:

$$f(i,j) = \sqrt{(I)(i+1j))^2 + (I(ij+1) - I(i,j))^2} \qquad (4.2)$$

The gradient operation can reflect the detailed information of an image clearly.

- **CGCM Matrix**

G is defined as the CGCM, which can be shown as:

$$G(I,j) = INT\llcorner f(I,j) * Ng/ fmax + 5.0 \qquad (4.3)$$

Where

Fmax = max{f(i,j), i<[0,M-2], j<[0,N-2]}.

The INT function returns the value of a number rounded downwards to the nearest integer.

Ng =256, Nc =256.

H is a matrix of size (Nc+1), (Ng+1); it represents the CGCM.

H is given as:

$$H(\mathrm{x},\mathrm{y}) = \sum_{i=0}^{M-2} \quad \sum_{J=0}^{N-2} \partial(c(i,j) = x, G(i,j) = y) \qquad (4.4)$$

Where $\partial(A = x, B = y) = \begin{cases} 1 \; if \; A = x \; and \; B = y \\ 0 \; otherwise \end{cases}$

The CGCM is normalised as:

$$\mathrm{H}' = \frac{\mathrm{H(x,y)}}{\mathrm{(Nc+1)(Ng+1)}} \qquad (4.5)$$

(Gong and Wang, 2012).

- **CGCM Features**

Several statistical features are extracted from the CGCM. The process of extracting the features are explained below.

The stepwise DA and MLP classification methods were used to train and test the system. All features were divided into three vectors as the system deals with RGB images. Therefore, each feature had three values for the three RGB channels: red, green and blue. Thus, the 11 extracted features become became 33 features or predictors.

T1 = Small gradient dominance: (T1_red, T1_green and T1_blue).

T2 = Big gradient dominance: (T2_red, T2_green and T2_blue).

T3 = Colour asymmetry: (T3_red, T3_green and T3_blue).

T4 = Gradient asymmetry: (T4 red, T4_green and T4_blue).

T5 = Energy: (T5_red, T5_green and T5_blue).

T6 = Colours' mean: (T6_red, T6_green and T6_blue).

T7 = Gradient mean: (T7_red, T7_green and T7_blue).

T8 = Colours' variance: (T8_red, T8_green and T8_blue).

T9 = Gradient variance: (T9_red, T9_green and T9_blue).

T10 = Dissimilarity: (T10_red, T10_green and T10_blue).

T11 = Homogeneity: (T11_red, T11_green and T11_blue).

Note: The stepwise DA and MLP considers each single colour channel feature as an individual predictor. For example, it considers T1_red as a single feature on its own.

❖ Statistical Features Equations

- **Small Gradient Dominance (T1):**

$$T1 = \frac{\sum_{Y=0}^{Ng}\sum_{X=0}^{Nc}\frac{H\prime(x,y)}{(1+y)^2}}{\sum_{Y=0}^{Ng}\sum_{X=0}^{Nc}H\prime(x,y)} \quad (4.6)$$

- **Big Gradient Dominance (T2):**

$$T2 = \frac{\sum_{Y=0}^{Ng}\sum_{X=0}^{Nc}H\prime(x,y)*y^2}{\sum_{Y=0}^{Ng}\sum_{X=0}^{Nc}H\prime(x,y)} \quad (4.7)$$

- **Colour Asymmetry (T3):**

$$T3 = \frac{\sum_{x=0}^{Nc}[\sum_{y=0}^{Ng}H\prime(x,y)]^2}{\sum_{Y=0}^{Ng}\sum_{X=0}^{Nc}H\prime(x,y)} \quad (4.8)$$

- **Gradient Asymmetry (T4):**

$$T4 = \frac{\sum_{Y=0}^{Ng}[\sum_{X=0}^{Nc}H\prime(x,y)]^2}{\sum_{Y=0}^{Ng}\sum_{X=0}^{Nc}H\prime(x,y)} \quad (4.9)$$

- **Energy (T5):**

$$T5 = \sum_{y=o}^{Ng}\sum_{x=0}^{Nc}H'(x,y) \quad (4.10)$$

- **Colours' Mean (T6):**

$$T6 = \sum_{x=o}^{Nc}x * [\sum_{y=0}^{Ng}H'(x,y)]^2 \quad (4.11)$$

- **Gradient Mean (T7):**

$$T7 = \sum_{y=o}^{Ng} y * [\sum_{x=0}^{Nc} H'(x,y)]^2 \qquad (4.12)$$

- **Colours' Variance (T8):**

$$T8 = \sqrt{\sum_{x=o}^{Nc}(x-T6)^2 * [\sum_{y=0}^{Ng} H'(x,y)]} \qquad (4.13)$$

- **Gradient Variance (T9):**

$$T9 = \sqrt{\sum_{y=o}^{Ng}(y-T7)^2 * [\sum_{x=0}^{Nc} H'(x,y)]} \qquad (4.14)$$

- **Dissimilarity (T10):**

$$T10 = \sum_{x=0}^{Nc}\sum_{y=0}^{Ng} |x-y| H'(x,y) \qquad (4.15)$$

- **Homogeneity (T11):**

$$T11 = \sum_{x=0}^{Nc} \sum_{y=0}^{Ng} \frac{H\prime(x,y)}{1+|x-y|} \qquad (4.16)$$

(Gong and Wang, 2012; Filippas et al., 2003).

4.4.1.1. Implementation Process for CGCM Phase

Figure 4.7 shows how the implementation processes of the detection system work in more detail.

All tested images are fed into a developed algorithm written in MATLAB to extract the statistical features selected. The 11 features become 33 predictors because of the RGB channels. Results are divided into the red, green and blue channels. In addition, each of them is considered a single predictor (Gonzalez et al. 2004; Mathworks, 2012).

After that, all results are gathered and divided into separate Excel sheets. Those sheets are classified according to the colour channel.

All Excel sheets then are imported into IBM SPSS statistics to implement the stepwise DA and MLP classifiers to train and test the system. Finally, the accuracy of the predicted images is found after implementing the two classifiers.

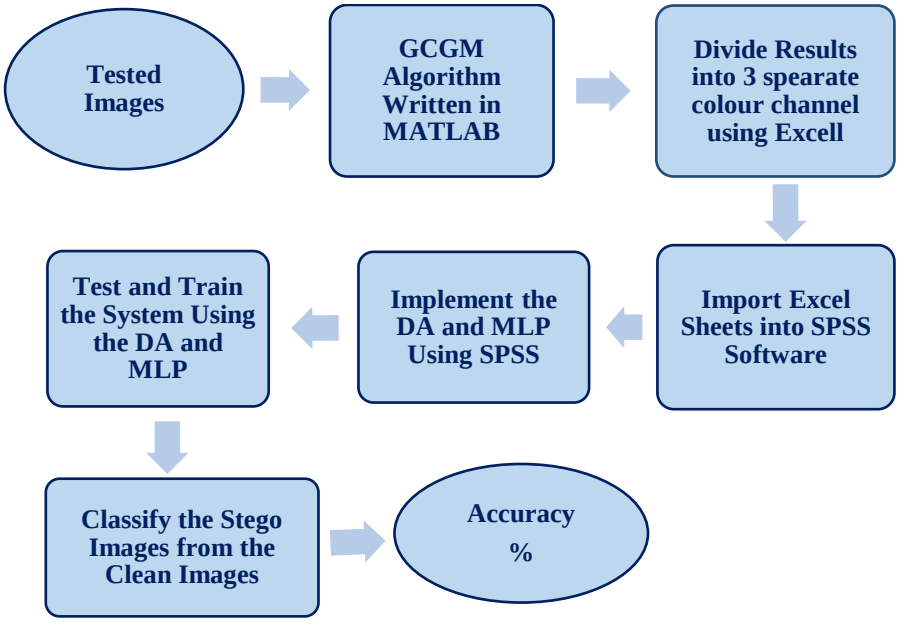


Figure 4.7: The CGCM Implementation Process.

4.4.2. The Extracted Histogram Features

The peak value and the renormalized histogram were used as features for classification (Zhang et al. 2007; Cai et al. 2010). The peak value of the histogram decreased after LSB embedding, while the renormalized histogram (the ratio of the histogram to the peak value) increased.

In addition, the F5 algorithm does modify the histogram of DCT coefficients, preserving some of its crucial characteristics, such as its monotonicity and monotonicity of increments (Fridrich et al. 2002).

The histogram of the cover image can be calculated from the stego-image. Because F5 modifies the histogram in a well-defined manner, the number and the modified coefficients can be calculated by comparing the estimated histogram with the histogram of the stego and clean images (Fridrich et al. 2002).

The histogram features of the colour images were extracted from the three colour channels: red, green, blue (RGB). Each colour channel was treated and analysed separately (Harmesn and Pearlman 2003; Cai et al., 2010; Cai et al. 2011).

The features were extracted by exploiting the histogram of difference image, which is usually a generalised Gaussian distribution centred at 0.

Compared with the histogram of the original image, the histogram of difference image can be more precisely characterised as a Generalised Gaussian Distribution (GGD) centred at 0. In this case, the 'renormalized histogram' is defined as the ratio of the histogram to the peak value; accordingly, the peak value of the renormalized histogram is an exact 1 (Cai et al. 2010).

The histogram of difference image and the renormalized histogram are created for clean and stego images, therefore, the peak value and the renormalized histogram are used as features for classification (Cai et al. 2010).

$$HS(k) = \alpha/4\ hc(k-1) + (1\ \alpha\ /2)\ hc(k) + \alpha/4\ hc(k+1) \quad (4.17)$$

where $\alpha \in (0, 1]$ is the embedding rate, hc and hs are the histograms of cover and stego images. The resulting stego image's histogram is a regularization of the cover image's histogram.

Many notations are taken into consideration; let I be an image, and h be the (normalized) histograms of I:

$$H(k) = \frac{\#\{(i,j): I(i,j) = k\}}{N} \quad (4.18)$$

where the symbol # denotes the cardinal number of a set, and N is the total number of image pixels.

Iv is the difference between neighbouring pixels in the vertical direction:

$$Iv(i,j) = I(i,j) - I(i+1,j) \quad (4.19)$$

The renormalized histogram of Iv is defined as h'v:

$$h'v(k) = \frac{hv(k)}{hv(0)} = \frac{\#\{(i,j):Iv(i,j)=k\}}{\#\{(i,j):Iv(i,j)=0\}} \quad (4.20)$$

where hv is the histogram of Iv.

Ih is the difference between neighbouring pixels in the horizontal direction:

$$Ih(i,j) = I(i,j) - I(i,j+1); \quad (4.21)$$

$$h'h\ (k)=\frac{hh(k)}{hh(0)}=\frac{\#\{(i,j):Ih(i,j)=k\}}{\#\{(i,j):Ih(i,j)=0\}} \quad (4.22)$$

Id is the difference between neighbouring pixels in the diagonal direction:

$$Id(i,j)=I(i,j)-I(i+1,j+1); \quad (4.23)$$

$$h'd\ (k)=\frac{hd(k)}{hd(0)}=\frac{\#\{(i,j):Id(i,j)=k\}}{\#\{(i,j):Id(i,j)=0\}} \quad (4.24)$$

Ia is the difference between neighbouring pixels in the anti-diagonal direction:

$$Ia(i,j)=I(i,j+1)-I(i+1,j); \quad (4.25)$$

$$h'a(k)=\frac{ha(k)}{ha(0)}=\frac{\#\{(i,j):Ia(i,j)=k\}}{\#\{(i,j):Ia(i,j)=0\}} \quad (4.26)$$

The difference between the neighbouring vertical pixel differences is extracted in many notations from the main four equations described above: Ivv, Ivh, Ivd, Iva, Ihh, Ihd, Iha, Idd, Ida, Iaa, as the following:

Ivv represents the difference between neighbouring vertical pixel differences in the vertical direction:

$$Ivv(i,j)=Iv(i,j)-Iv(i,j+1); \quad (4.27)$$

Ivh indicates the difference betwwen neighbouring horizontal pixel differences in the vertical direction:

$$Ivh(i,j)=Iv(i,j)-Iv(i+1,j); \quad (4.28)$$

Ivh represents the difference between neighbouring vertical pixel differences in the diagonal direction:

$$Ivd(i,j)=Iv(i,j)-Iv(i+1,j+1); \quad (4.29)$$

Ivh is the difference between neighbouring vertical pixel differences in the anti-diagonal direction:

$$Iva(i,j) = Iv(i,j+1) - Iv(i+1,j); \quad (4.30)$$

$$Ihh(i,j) = Ih(i,j) - Ih(i+1,j); \quad (4.31)$$

$$Ihd(i,j) = Ih(i,j) - Ih(i+1,j+1); \quad (4.32)$$

$$Iha(i,j) = Ih(i,j+1) - Ih(i+1,j); \quad (4.33)$$

$$Idd(i,j) = Id(i,j) - Id(i+1,j+1); \quad (4.34)$$

$$Ida(i,j) = Id(i,j+1) - Id(i+1,j); \quad (4.35)$$

$$Iaa(i,j) = Ia(i,j+1) - Ia(i+1,j); \quad (4.36)$$

However, the histogram of difference image can be more precisely characterised because neighbouring pixels are often highly correlated. Therefore, the features are derived from the difference images histogram instead of the histogram of the original image.

Let Ic be a gray-scale clean image, Is its stego image embedded by LSB or F5 algorithm with embedding rate α, and let hc and hs be the histograms of Ic and Is.

According to equation (4.17), the following seems true:

$$hs = f\alpha * hc \quad (4.37)$$

where fα is the distribution of embedding noise:

$f\alpha(0) = 1 - \alpha/2$, $f\alpha(1) = f\alpha(-1) = \alpha/4$.

Furthermore, Ivc and Ivs denote the vertical difference images of Ic and Is, and hvc and hvs represent the histograms of Ivc and Ivs. At this point, hvs is the convolution of hvc with a certain kernel function,

$$hvs = g\alpha * hvc \quad (4.38)$$

Several reliable histogram features are obtained from this important property (Ca et al., 2010). In addition, the renormalized histogram is symmetric to 0 and steep in shape. We then simply take

$$\frac{h'v(1) + h'v(-1)}{2}, \ldots, \frac{h'v(n) + h'v(-n)}{2} \quad (4.39)$$

to represent the statistics of the renormalized histogram, wherein > 0 is a pre-selected integer.

In summary, (n+1) reliable histogram features are selected for classification:

$$\mathrm{hv}(0), \frac{\mathrm{hv}(1)+\mathrm{hv}(-1)}{2\mathrm{hv}(0)}, \ldots., \frac{\mathrm{hv}(n)+\mathrm{hv}(-n)}{2\mathrm{hv}(0)} \quad (4.40)$$

where hv is the histogram of difference image.

The following steps summarise the feature extraction process (details of the equations and the written code are shown in the appendix):

- First, compute the difference images for four directions (vertical, horizontal, diagonal and anti-diagonal) to obtain Iv, Ih, Id and Ia.
- Next, again calculate the difference images for each difference image to obtain Is,t, where s, t ∈ {v, h, d, a}. For instance, the image Iv,h here means the horizontal difference image of the vertical difference image Iv.
- Then, exclude identical images (for instance, Iv,h = Ih,v) and obtain 14 totally different images:
  - Iv, Ih, Id, Ia.
  - Ivv, Ivh, Ivd, Iva, Ihh, Ihd, Iha, Idd, Ida, Iaa.

- For each of these 14 images, compute the (n + 1) features.
- Thus, the total of the features is 14(n + 1).
- For n_file=1:size(all_files,1) % for each file in the file list, do the following stat loop (Cai et al., 2010):

```
filename=all_files(n_file).name;
Im = imread(filename);
ncolors = size(Im,3); % number of colour channels (=1 for indexed images and mono, = 3 for RGB images)
if ncolors > 1
  for i=1:ncolors
    [features] = hist_lsb(Im(:,:,i),n);
```

(Cai et al., 2010; Cai et al. 2011).

4.4.2.1. Implementation Process for the Histogram Features Phase

Figure 4.8 represents the implementation processes while extracting the histogram features and using them in the proposed detection system.

All tested images are fed into a developed algorithm written in MATLAB to extract the histogram features. The 70 features become 210 predictors because of the RGB channels. Results are divided into the red, green and blue channels. In addition, each of them is considered a single predictor.

After that, all results are gathered and divided into separate Excel sheets. Those sheets are classified according to the colour channel.

All Excel sheets then are imported into IBM SPSS statistics to implement the sepwise DA and MLP classifiers to train and test the system.

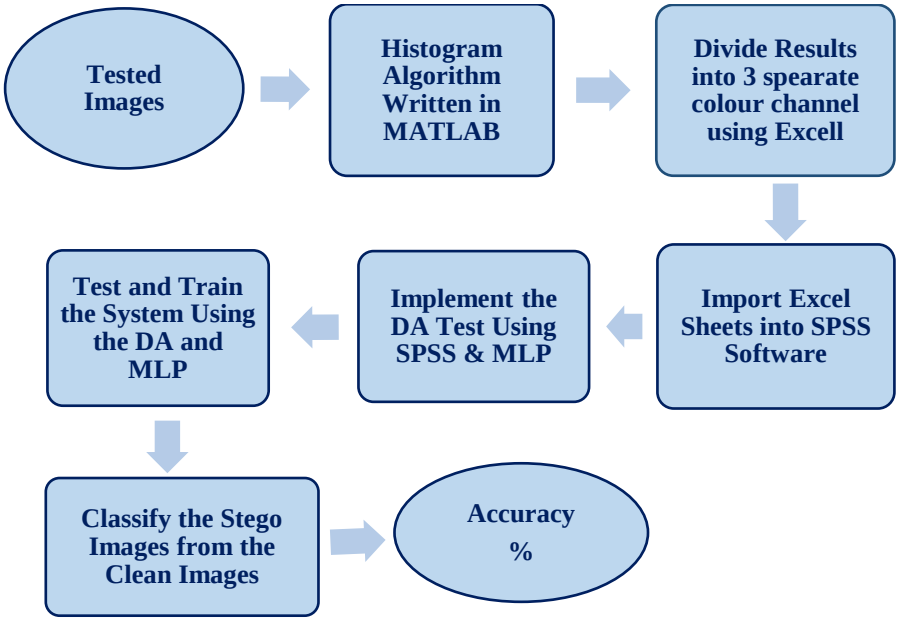


Figure 4.8: The Histogram Implementation Process.

4.5. Methodology of the Experiments

In the both parts of the detection system, three types of experiments were conducted as shown in figures 4.9 and 4.10. Firstly, the effectiveness of the hidden files was tested, because detection ability is related to the hidden file length. Clearly, the less information embedded into

the cover-image, the smaller the probability of introducing detectable artefacts by the embedding process.

Each steganographic method has an upper bound on the maximal safe hidden file length that indicates how many bits can be safely embedded in a given image without introducing any statistically detectable artefacts. Determining this maximal steganographic capacity is a nontrivial task even for the simplest methods. Therefore, various capacities and hidden files were tested to reveal the differences in the results and to train the system to deal with varieties of stego images.

After this initial testing, the second step was to evaluate and classify the results using the stepwise DA classifier. A final step assessed the results and classified by using the MLP classifier.

The steganography methods used were considered as important factors contributing in the evaluation of the results. Each of the methods used showed dissimilar performance in terms of classifying the stego images from the clean ones.

Figure 4.9 represents the categories of the experiments conducted with the CGCM features. In this part, only the colour images were used and the results were classified according to the images compressions, either lossless or lossy. The BMP and PNG formats are used as examples of lossless compression and the JPG format is used as an example of lossy compression.

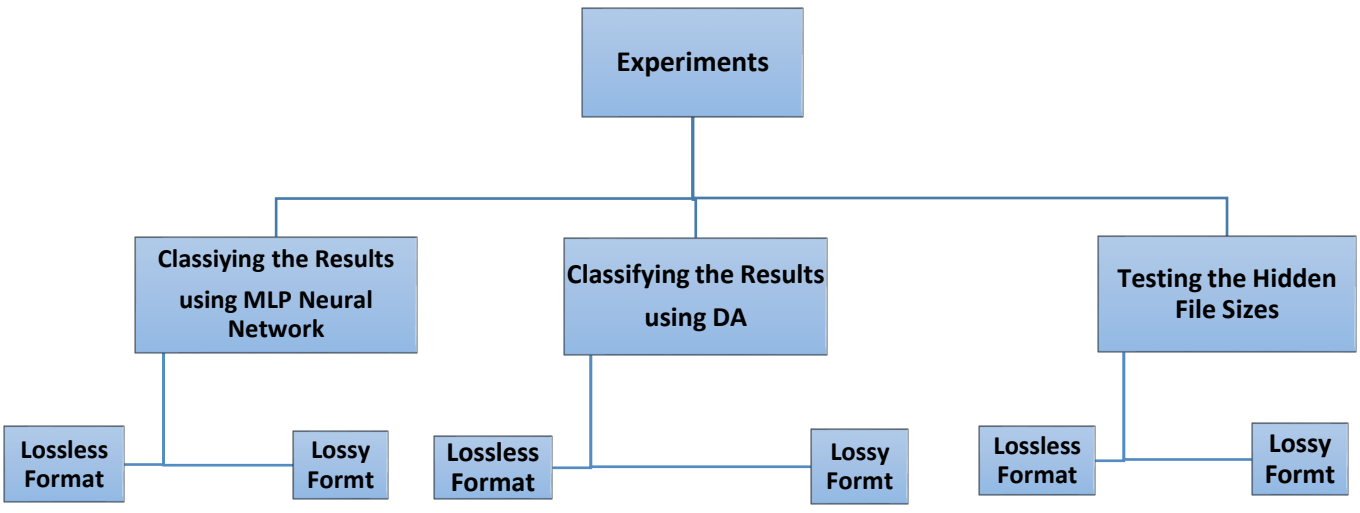


Figure 4.9: Methodology of the Experiments Conducted (CGCM Phase).

The outcomes in this part focused on analysing the differences of the results based on changing the image formats.

Figure 4.10, on the other hand, shows the groups of the experiments conducted with the histogram features. In this part, both grey and colour images were used. The majority of the histogram features were extracted from the grey images first; then the algorithm was developed to extract the features from the colour images as well.
The outcomes in this part focused on analysing the differences of the results based on changing the colour of the images.

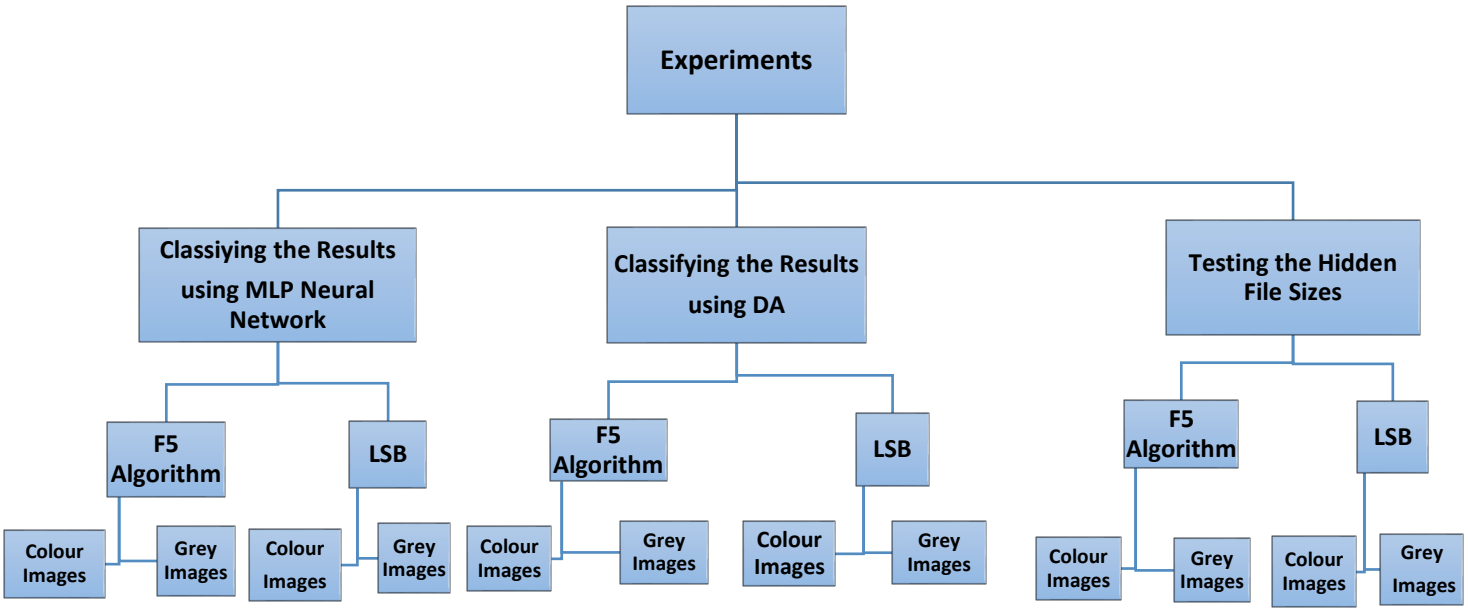


Figure 4.10: Methodology of the Experiments Conducted (Histogram Features Phase).

4.6. Classification Methods Used

Two different classification methods were used to test, train and validate the system. Brief descriptions of the two methods are described below.

4.6.1. Discriminant Analysis (DA)

DA (Discriminant Function Analysis) completes similar functions as the multiple linear regression through estimating results. The multiple linear regression is, however, restricted to situations that feature the dependent variable (or interval variable) on the Y-axis. Only this circumstance creates a prediction of the average group integer Y figures for specific figures of estimated matches with the X figures through a regression equation. Nonetheless, Klecka (1980) and Dunteman (1984) have identified that numerous pertinent variables are unconditional; for example, likely political party votes, immigration status, owning a specific credit card, currently employment status, and so forth. Therefore, these variables should be examined through the DA.

The DA can be simplified into a dual-phase procedure. The first phase is to validate the importance of a series of differentiate operations, which is then followed by categorization. The initial phase is mathematically similar to MANOVA as there is a matrix of overall alterations and co-alterations and a matrix of group alterations and co-alterations. These matrixes are then contrasted through the multivariate F examination to identify if there are substantial variations of the variables within the groups. Klecka (1980) and Dunteman (1984) explained that the multivariate examination is initially conducted. If the results reveal a statistic importance, the investigation then examines the specific variables that differ.

According to Klecka (1980) and Dunteman (1984), DA is used when:

- The dependent variable is categorized with the estimator IV's at interim degrees like age, salary, beliefs, opinions, academic attainment, etc. However, false variable can also be employed to estimate multiple regression.
- Dissimilar to logistic regression, at least three DV classes occur that are restricted to a dichotomous dependent variable.
- To examine variations between two populations to identify possible reasons for phenomenon that significantly cause the division of the population.

- To identify the most beneficial method for differentiating the populations
- To categorize situations into populations the statistical important examination is conducted employing a chi square that allows individuals to examine the success of each population's operation
- To validate the hypothesis of if situations are categorized as estimated.

❖ Discriminant analysis linear equation

DA involves the determination of a linear equation, such as regression, that will predict which group the case belongs to. The form of the equation or function is:

D v X v X v X .....v X a 1 1 2 2 3 3 i i = + + = +

where D = discriminate function

v = the discriminant coefficient or weight for that variable

X = respondent's score for that variable

a = a constant

i = the number of predictor variables.

The aim of the statistical analysis in DA is to combine (weight) the variable scores in some way so that a single new composite variable, the discriminant score, is produced.

One way of thinking about this is in terms of a food recipe, where changing the proportions (weights) of the ingredients will change the characteristics of the finished cakes.

Hopefully the weighted combinations of ingredients will produce two different types of cake. Similarly, at the end of the DA process, it is hoped that each group will have a normal distribution of discriminant scores. The degree of overlap between the discriminant score distributions can then be used as a measure of the success of the technique, so that, like the different types of cake mix, we have two different types of groups (Klecka, 1980; George, 1984).

For example, the top two distributions in Figure 4.11 overlap too much and do not discriminate too well compared to the bottom set. Misclassification will be minimal in the lower pair, whereas many will be misclassified in the top pair (Klecka, 1980; Dunteman, 1984).

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 4.11: Discriminant Distribution (Klecka 1980).

- **Stepwise Discriminant Analysis (DA)**

The stepwise DA is implemented for all experiments conducted in this work.

The stepwise DA is similar to multiple repression as it aims to identify the most beneficial series of estimators and is commonly employed within discovery cases to recognize variables within a great amount of variables that could be employed later within a more challenging and hypothetically influenced examination. Within this method, the greatest associated independent variable is initially inputted to the stepwise software before any other dependent variables that do not substantially contribute to the canonical R squared. The guidelines, according to McLaclan (2004), for further inclusions or deductions are commonly established by the substantial degree of importance of eliminating F. Numerous tables, such as Klecka (1980) and Dunteman (1984), have beneficial data for estimating the results of the DA examination.

Figure 4.12 shows the dialogue of the DA in the SPSS software.

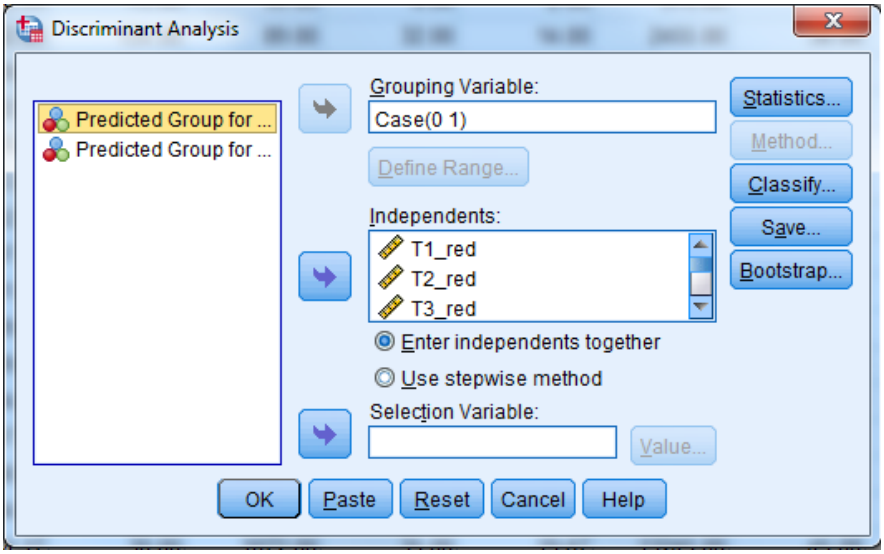


Figure 4.12: User Interface of the DA in SPSS.

❖ **Cross-Validation**

The 'leave-one out classification' is a cross-validation method.

Cross validation is the process of testing a model on more than one sample. This technique is often undertaken to assess the reliability and generalisability of the findings. Cross validation can be executed in the context of factor analyses, discriminant function analyses, multiple regression, and so forth. This process is particularly crucial in discriminant function analysis, because the solutions are often unreliable (Cawley and Talbot 2003).

In this work, the cross-validated outcomes appearing in the classification table are considered in analysing the results for all experiments conducted.

4.6.2. Neural Network

'A computational neural network is a set of non-linear data modelling tools consisting of input and output layers plus one or two hidden layers. The connections between neurons in each layer have associated weights, which are iteratively adjusted with the training algorithm to minimise error and provide accurate predictions' (Dayhoff, 1990; Harb and Jayousi, 2012).

The multilayer perceptron (MLP) and radial basis function (RBF) networks are frequently occurring instances of unbiased network estimative employment. According to Dayhoff (1990), Yen-Jen (2005), and Harb and Jayousi (2012), both networks are regulated by

prototype-estimated outcomes, which can be contrasted to the proven figures of the identified variables. The MLP and RBF algorithms are understood as monitored networks as the prototype-estimated outcomes can be contrasted to the proven figures of the identified variables. A essential benefit of unbiased networks within creating contrasts of traditional statistical methods is that unbiased networks are adaptable and do not include distributional presumptions; for instance, unbiased networks can be employed to estimate unconditional and reoccurring results. Unbiased networks also have drawbacks, as they can typically be hard to understand as they often create very complicated prototypes that include numerous layers. An unbiased networks functions through considering the estimated or inputted figures through an algorithm. The algorithm creates an input later that is employed to conceal another layer of invisible node or units that corresponds to an operation within the input layer. Notably, some networks have additional concealed layers.

The output layer contains the responses or predictions. The network is continually rebuilt or refined so that the synaptic weights in the nodes correctly predict the outcome, see figure 4.13.

Some materials have been removed from this thesis due to Third Party Copyright.
Pages where material has been removed are clearly marked in the electronic version.
The unabridged version of the thesis can be viewed at the Lanchester Library,
Coventry University

Figure 4.13: Graphical Representation of a Neural Network Model (Yen-Jen (2005).

❖ Multilayer Perceptron (MLP)

'The MLP procedure produces a predictive model for one or more dependent (target) variables based on the values of the predictor variables others' (Pallant, 2010).

The MLP neural network algorithm is based on the functional principles of biological neural structures. Researchers of the neural network attempt to achieve similar processing abilities by organising mathematical models similar to the structures and organisation of neurons in biological brains. Moreover, they attempt to incorporate inherent capacities of biological brains, such as learning based on examples, trial and error, and knowledge generalization, among many others (Pallant 2010).

MLP is a monitored instruction algorithm that studies the function (f (.): R^m → R^o) through focusing on the dataset. Within this function, m represents the input size while o represents the output size. Within a series of characteristics X = {x_1, x_2, ..., x_m} with an objective of Y, MLP can study the non-linear operation estimator for categorization or regression. This varies from logistic regression as additional non-linear layers (i.e. concealed layers) can exist between the inputs and output later.

Figure 4.14 by Scikit-learn (2010) depicts a concealed layer within a MLP that has a scalar output. The layer furthest to the left is the input layer, which is composed of a series of neurons (\{x_i | x_1, x_2, ..., x_m\}) that signifies the input characteristics. Every neuron within the concealed layer transfigures the figures of the former layer through the weighted linear equation w_1x_1 + w_2x_2 + ... + w_mx_m and then a non-linear operation of g(.): R → R – similar to the hyperbolic tan operation. The output layer, according to Scikit-learn (2010), obtains figures from the concealed layer and transfigures these figures into the output results. MLPs have two primary benefits as they are able to study non-linear prototypes and current line prototypes through an Internet program employing partial fit.

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 4.14: One Hidden Layer MLP (Scikit-learn 2010).

Additionally, the MLP also needs to be adjusted to numerous hyper guidelines like the amount of concealed neurons, layers, and repetitions. According to Scikit-learn (2010), the MLP is also vulnerable to characteristic calibration.

An example of an MLP Network is shown in figure 4.15; the data feeds forward from the input layer through one or more hidden layers to the output layer.

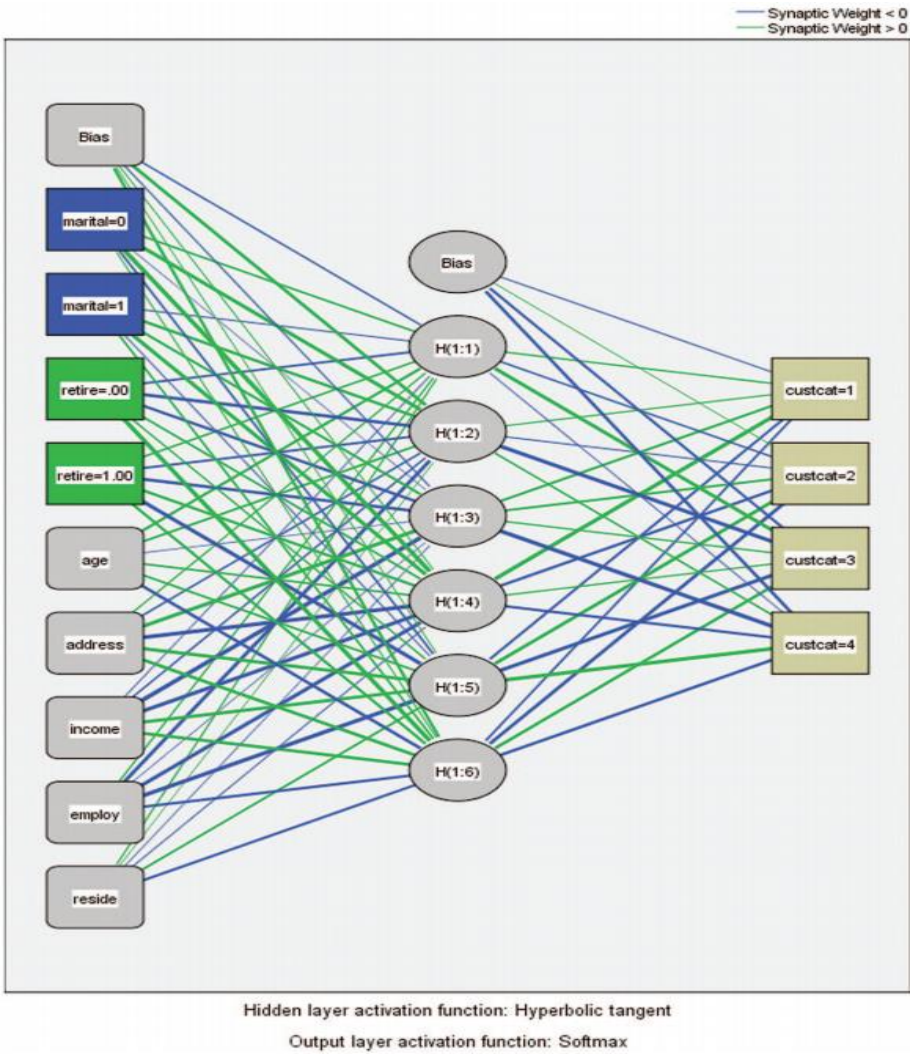


Figure 4.15: An Example of an MLP Network, the data feeds forward from the input layer through one or more hidden layers to the output layer.

4.7. Criteria for Evaluation the Classifier's Performance

In order to reasonably evaluate the performance of various kinds of steganalytic methods, it is necessary to identify some criteria acceptable to the majority.

The confusion matrix and calculating AUC are selected to evaluate the performance of the steganalysis system proposed.

The confusion matrix and the AUC are implemented to evaluate the results and the performance of the stepwise DA and MLP used to train and test the system, as shown in the results presented in chapter 5, 6 and 7.

4.7.1. Confusion Matrix

When applying a steganalytic method on a testing data set, which may consist of cover and stego media, a 2*2 confusion matrix, which is illustrated in Figure 4.16, can be constructed, representing the dispositions of the instances in the set (Li et al. 2011).

There are four possible resultant situations when using any steganalysis method:

- True positive (TP), meaning that a stego medium is correctly classified as stego.
- False negative (FN), meaning that a stego medium is wrongly classified as cover.
- True negative (TN), meaning that a cover medium is correctly classified as cover.
- False positive (FP), meaning that a cover medium is wrongly classified as stego. (Li et al. 2011).

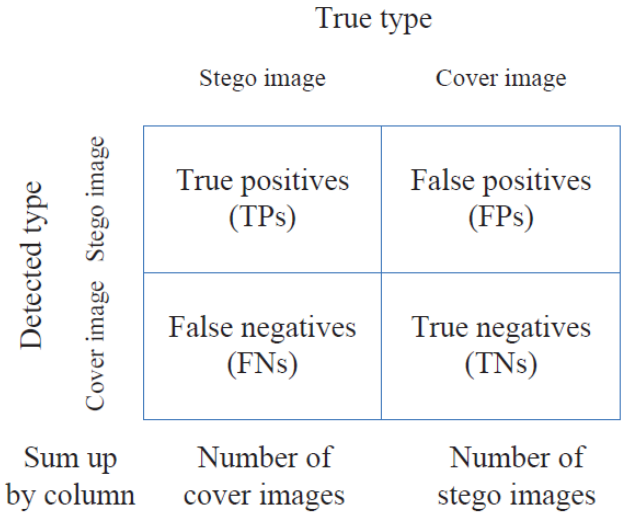


Figure 4.16: Confusion Matrix (Li et al. 2011).

In addition, the sensitivity, specificity, false positive rate, accuracy and precision are calculated for all results to measure the performance of the classification methods used and the detection ability.

Sensitivity (also called the true positive rate, or the recall in some fields) measures the proportion of positives that are correctly identified as such (e.g., the percentage of sick people who are correctly identified as having the condition). In other words, when it's actually yes, how often does it predict yes? However, the false positive rate means that when it's actually no, how often does it predict yes.

Specificity (also called the true negative rate) measures the proportion of negatives that are correctly identified.

Based on this matrix, some evaluation metrics can be defined.

$$\text{True Positive Rate (Sensitivity)} = \frac{\text{TPs}}{\text{TPs} + \text{FNs'}} * 100 \quad (4.41)$$

$$\text{Specificity} = \frac{\text{TNs}}{\text{TNs} + \text{FPs'}} * 100 \quad (4.42)$$

$$\text{False Positive Rate} = \frac{\text{FPs}}{\text{TNs} + \text{FPs'}} * 100 \quad (4.43)$$

$$\text{Accuracy} = \frac{\text{TPs}+\text{TNs}}{\text{TPs} + \text{FNs} + \text{TNs} + \text{FPs'}} * 100 \quad (4.44)$$

$$\text{Precision} = \frac{\text{TPs}}{\text{TPs} + \text{FPs'}} * 100 \quad (4.45)$$

Normally, the performance of a steganalysis method is measured by its detection rate and its error rate. The detection rate shows the likelihood that a stego object is identified correctly, and error rate shows the likelihood that a carrier is incorrectly classified. False *positive* errors occur when the steganalyzer erroneously identifies a clean image as a stego carrier. Such errors diminish the productivity of a steganalysis system by triggering unneeded countermeasures. False *negative* errors, on the other hand, occur when a stego object is classified as a clean image. These errors can be quite serious as they may generate unexpected or undesirable system operations, and losses for an organization could be considerable because potentially harmful messages could be disseminated. Thus, the risks arising from false negative errors are higher than the risks arising from false positive errors. In general, most research on steganalysis employing machine learning techniques have focused on improving the predictive abilities of steganalyzers (Geetha, Sindhu and Kamaraj, 2009).

It should be remembered that accuracy and precision are similar but not the same. Accuracy describes the capability of data, measurements or results to match the actual ‘true’ value,

whereas precision is how close these data, measurements or results are to each other, working as a measure of the spread of data from the average.

Figure 4.17 shows examples of how accuracy and precision are correlated together.

Some materials have been removed from this thesis due to Third Party Copyright. Pages where material has been removed are clearly marked in the electronic version. The unabridged version of the thesis can be viewed at the Lanchester Library, Coventry University

Figure 4.17: Examples of the Relationship between Accuracy and Precision (Climatica 2016).

4.7.2. Area Under Curve (AUC)

Within binary categorization, categorizers' functions are typically quantified through the AUC. Fawcett (2004) explained that the AUC is also computed to quantify the network's function and categorizes the operation F to represent the possibility that an unsystematically chosen beneficial occurrence receives a greater value by F instead of an unsystematically chosen disadvantage occurrence. This method has been demonstrated to be extremely beneficial, according to Fawcett (2004), for assessing categorizers, particularly when assigned categories are excessively uneven. Precision and AUC are beneficial methods for accessing the function of categorisers, according to Fawcett (2004), and the AUC is also not susceptible to category dispersion. The AUC estimation of F can be explained by the following ways (Calders and Jaroszewicz 2007):

$$AUC\ (f) := P\ (f(x) < f(y)\ |C(x) = 0, C(y) = 1) \qquad (4.46)$$

Where P is the probability ranged from 0 to 1.

A larger AUC value means better detection performance. More precisely, an AUC value close to 1.0 indicates excellent discrimination, while a value close to 0.5 indicates poor discrimination (Fawcett 2004).

4.8. Conclusion

This chapter clarified the methodology of the proposed system and presented the created images database.

In addition, it explained in details the steganography tools used to create the stego images used.

Besides, descriptions of the two classifications methods used to train and test the system.

It defined the methodologies of the experiments and the equations extracted from the CGCM features and the histogram features.

Criterias for evaluating the performance of the classificatiers were described at the end of the chapter.

Chapter 5

Implementation of the Colour Gradient Co-occurrence Matrix (CGCM) Features

5.1. Introduction

This chapter explains in detail the implementation of the CGCM part of the research.
The concept of the CGCM and the features extracted have already been introduced in section 4.4.2, along with all equations that were extracted from the matrix.
Section 5.3 shows the experiments conducted to test the effectiveness of changing the hidden file sizes for the lossless images.
In addition, results achieved by the proposed system in terms of classifying the stego images from the clean ones are shown in sections 5.4 and 5.5.
Validation of the results using the stepwise DA and MLP is introduced in sections 5.6 and 5.7.

5.2. GCGM Experiments

Thirty-three selected features were extracted from the CGCM. As shown in section 4.4.1, these features are: small gradient dominance, big gradient dominance, colour asymmetry, gradient asymmetry, energy, colours' mean, gradient mean, colours' variance, gradient variance, dissimilarity and homogeneity.
As explained in section 4.2, the experiments were classified into three main categories: producing stego images, selecting images features and using classification methods to classify the results. In addition, the results were divided and compared according to the image formats types.

The BMP and PNG formats were used as examples of lossless compression and the JPG format was used as an example of lossy compression.
Results were classified using the stepwise DA classification method.

The Cross-validated results shown in all classification tables were considered in examining the results.

5.3. Effectiveness of Different Hidden File Sizes using Stepwise DA

At the beginning of the training processes, the two lossless formats, BMP and PNG, were tested separately. The purpose of separating them was to test the effect of different sizes of hidden files on the results. The size of the hidden files plays an important role in terms of detection (Reddy et al., 2011).

This has been proven by research in the literature which has shown that the length of hidden messages embedded in the least significant bits of signal samples can be estimated with relatively high precision (Reddy et al., 2011).
Experiments in this section have been classified using the stepwise DA classification method only.

A value of 0 was assigned to all clean images, and 1 to all stego images. The stepwise DA grouped the images into two cases: clean and stego. During the training phase, all images were given either a 0 or 1 value according to their case.

5.3.1. Testing Lossless Format (BMP Images)

Three tests were conducted to test the different sizes of the hidden files, as follows:

Test 1:

- 100 bmp clean images were tested.
- 100 stego images were created using LSB steganography.
- The hiding capacity was 10%.
- Sizes of the clean images varied from 599 kb to 1 MB.

Test 2:

- 100 bmp clean images were tested.
- 100 stego images were created using LSB steganography.
- The hiding capacity was 25%.
- The range of the images varied from 599 kb to 1 MB.

Test 3:

Note: In this test, all stego images used in tests 1 and 2 were combined.

- There were 150 bmp clean images tested and 200 stego images, which were created using LSB steganography.
- The hiding capacities were 10% & 25%.

❖ Analysis of Test 1

In Test 1, the same small hidden file was embedded into all the tested stego images for accuracy. As can be seen in table 5.1, the cross-validated percentage is 97% when all features are included in the test.

Table 5.1: The Results Table for Test 1 (BMP Images – Small Hidden File).

Classification Results[a,c]					
			Predicted Group Membership		
		Case	Clean	Stego	Total
Original	Count	Clean	96	4	100
		Stego	0	100	100
	%	Clean	96.0	4.0	100.0
		Stego	.0	100.0	100.0
Cross-validated[b]	**Count**	**Clean**	**95**	**5**	**100**
		Stego	**0**	**100**	**100**
	%	**Clean**	**95.0**	**5.0**	**100.0**
		Stego	**.0**	**100.0**	**100.0**

a. 98.0% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 97.5% of cross-validated grouped cases correctly classified.

The stepwise DA was able to correctly predict the stego images at 100%. Moreover, it performed well (96%) when predicting the clean images. The stepwise DA removed all features that had poor predictive power from the analysis.

An example of the function coefficient of the features used is shown in table 5.2. The stepwise DA orders them according to their predictive power. This table evaluates the valuable contribution of each feature used in the test.

The included features were changed in each test and they had different predictive power every time.

Table 5.2: The Function Coefficient of the Features.

Standardised Canonical Discriminant Function Coefficients

	Function 1
T1_red	.226
T3_red	-.811-
T4_red	.913
T5_red	-.315-
T7_red	6.859
T9_red	-7.879-
T10_red	.983
T11_red	.538
T4_green	.459
T3_blue	.519
T4_blue	-.822-

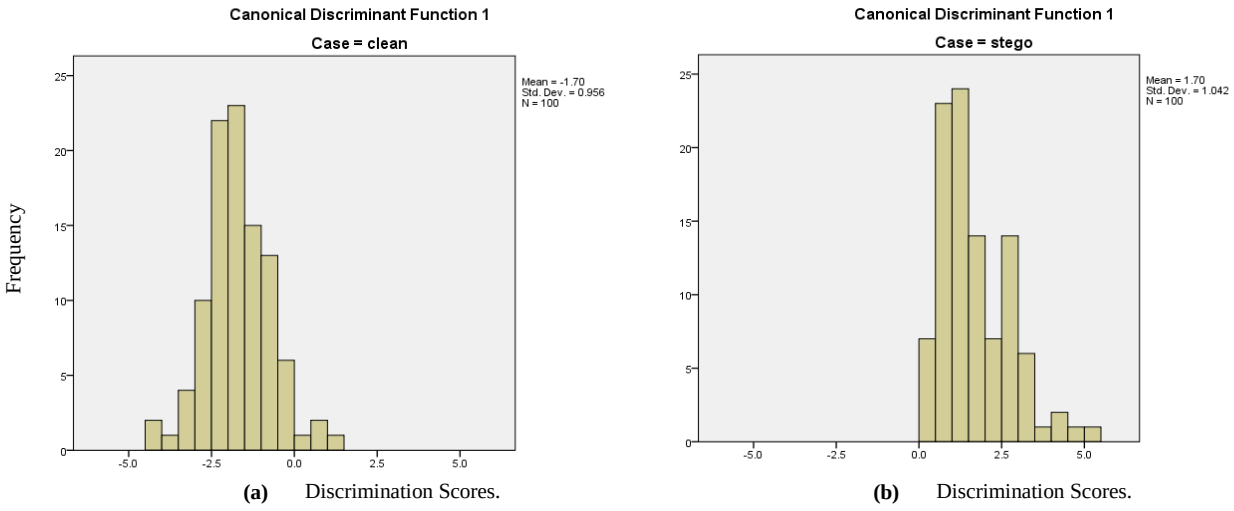
The stepwise DA created the 'structure matrix table', which illustrates all used and excluded features. (Example is shown in appendix. 5.1) Structure matrix tables were used to evaluate the contribution of each feature used in the test.

In addition, the stepwise DA provided the best combination of features by performing many steps (for an example, see appendix 5.2).

In some stepwise analyses, only the first one or two steps might be taken, even though there are more variables, because succeeding additional variables are not adding to the predictive power of the discriminant function. All variables that don't contribute to the analysis are removed by the classifier.

Figure 5.1 shows the effectiveness of the discriminant function. The two histograms illustrate the distribution of the discriminant function scores for each case.

When the distributions do not overlap, this suggests that the function discriminates well.



(a) Discrimination Scores. **(b)** Discrimination Scores.

Figure 5.1: Histograms of the Discriminant Function Distribution for Both Cases (Test 1, BMP Images).

The range of the clean images is from 1 to -4.5, as shown in figure 5.1 (a), and for the stego images is from 0 to 5 as presented in figure 5.1 (b). The range only overlaps from 0 to 1, which is very low overlapping. This very low overlapping means that there is a high degree of discrimination.

❖ Analysis of Test 2

In Test 2, a larger hidden file was embedded into all the stego images created. The same hidden file was used to get accurate results.

Table 5.3 shows that the cross-validated percentage is 96%. However, features that did not contribute well to the analysis were removed by the test. The stepwise DA was able to correctly predict the clean and the stego images with the same percentages.

Table 5.3: The Results Table for Test 2 (BMP Images; Large Hidden File), after implementing the stepwise discriminant analysis test.

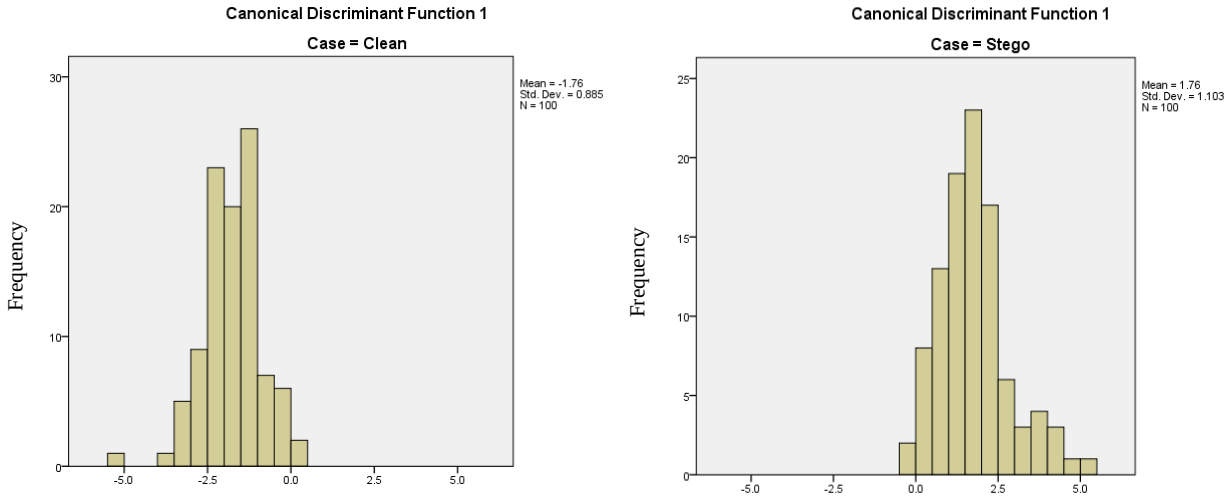
Classification Results[a,c]					
			Predicted Group Membership		
		Case	Clean	Stego	Total
Original	Count	Clean	98	2	100
		Stego	2	98	100
	%	Clean	98.0	2.0	100.0
		Stego	2.0	98.0	100.0
Cross-validated[b]	**Count**	**Clean**	**96**	**4**	**100**
		Stego	**4**	**96**	**100**
	%	**Clean**	**96.0**	**4.0**	**100.0**
		Stego	**4.0**	**96.0**	**100.0**

a. 98.0% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 96.0% of cross-validated grouped cases correctly classified.

Figure 5.2 shows the effectiveness of the discriminant function. The two histograms illustrate the distribution of the discriminant function scores for each case. The histograms prove that the function discriminates well, as the two distributions do not overlap.



(a) Discrimination Scores. **(b)** Discrimination Scores.

Figure 5.2: Histograms of the Discriminant Function Distribution for Both Cases (Test 2, BMP Images).

The range of the clean images is from 0 to -5, as shown in figure 5.2 (a), and for the stego images from 0 to 5, as present in figure 5.2 (b). The range only overlaps from 0 to 1, which is very low overlapping. This shows that the discrimination is very high.

❖ Analysis of Test 3

In Test 3, 200 stego images and 150 clean images were tested. This test examined a combination of all stego images tested in tests 1 and 2. Table 5.4 shows the cross-validated percentage is 84% when all features are supplied to the test.

Table 5.4: The Results Table for Test 3 (150 Clean Images + 200 Stego images, BMP Format) after implementing the Stepwise Discriminant Analysis Test.

Classification Results[a,c]					
		Case	Predicted Group Membership		Total
			Clean	Stego	
Original	Count	Clean	114	36	150
		Stego	20	180	200
	%	Clean	76.0	24.0	100.0
		Stego	10.0	90.0	100.0
Cross-validated[b]	**Count**	**Clean**	**114**	**36**	**150**
		Stego	**20**	**180**	**200**
	%	**Clean**	**76.0**	**24.0**	**100.0**
		Stego	**10.0**	**90.0**	**100.0**

a. 84.0% of original grouped cases correctly classified.
b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.
c. 84.0% of cross-validated grouped cases correctly classified.

The stepwise DA was effectively able to correctly predict the stego images more than the clean images. The error percentage in predicting the stego images was only 10%. However, the error percentage in predicting the clean images was 24%.

Figure 5.3 illustrates the effectiveness of the discriminant function. The two histograms show the distribution of the discriminant function scores from each case. The histograms demonstrate that the function discriminates well, as the two distributions do not overlap.

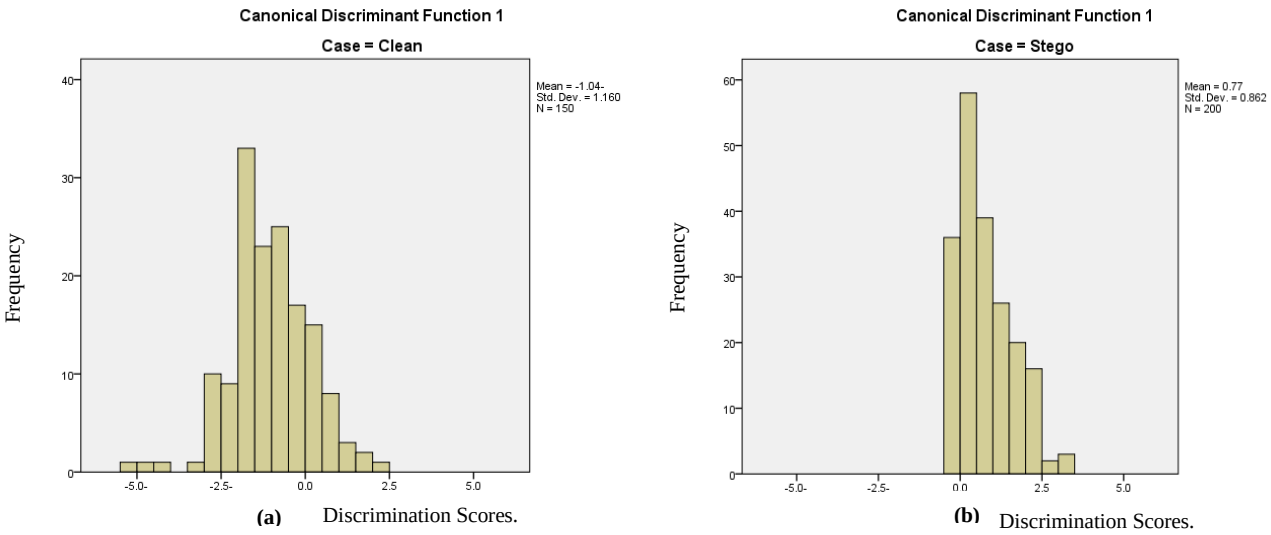


Figure 5.3: Histograms of the Discriminant Function Distribution for Both Cases (Test3, BMP Images).

The range of the clean images is from 2.5 to -5, as shown in figure 5.3 (a), and for the stego images from 0 to 2.8, as presented in figure 5.3 (b). The range of the clean images is clear and doesn't overlap with the stego from 0 to -5.

Although there is some overlapping from 0 to 2.5 only, it is quite minimal, which indicates that the distribution function does good discrimination performance.

❖ Comparison between Test 1, Test 2 and Test 3 (BMP Images)

Table 5.6 illustrates the cross-validated percentages for tests 1, 2 and 3. There are no noticeable differences between the cross-validated percentages of tests 1 and 2. This could mean the larger size of the hidden file used in Test 2 did not have any effect on the results in comparison to Test 1.

However, the correct percentage of predicting the stego images in Test 1 was 100. This percentage decreased slightly in Test 2 to 96%. Moreover, mostly the same features contributed to the analysis of tests 1 and 2. These features are: T4_red, T5_red, T7_red, T9_red, T4_green and T5_green.

Table 5.6: Comparison between Test 1, 2 and 3 (BMP Format).

	Cross-validated%	Correct Prediction % Clean	Error Prediction % Clean	Correct Prediction% Stego	Error Prediction% Stego	Contributed Features	No. clean Images	No. Stego Images
Test 1 (BMP, Small hidden file)	97%	95%	5%	100%	0%	T1_red, T3_red, T4_red, T5_red, T7_red, T9_red, T10_blue, T11_red, T4_green, T3_blue and T4_blue	100 images	100 images
Test 2 (BMP, Large hidden file)	96%	96%	4%	96%	4%	T4_red, T5_red, T7_red, T8_red, T9_red, T2_green, T4_green, T7_green, T9_green, T4_blue and T5_blue.	100 images	100 images
Test 3 (BMP, Small and large hidden files)	84%	76%	24%	90%	10%	T3_red, T4_red, T5_red, T6_red, T9_red, T10_red, T11_red, T3_green	150 images	200 images

The following features contributed to the analysis of the three tests, 1, 2 and 3:
T4_red, T5_red and T9_red. The cross-validated percentage decreased to 84% in Test 3. This percentage is obvious, as more images were used in test 3. The test was successfully able to predict the stego images, at 90%. However, it predicted the clean images at only 76%. Another point to be taken into account is the number of images used in the analysis. In tests 1 and 2, the number of clean and stego images is the same. This means there is balance between the two cases: clean and stego.

In Test 1, the percentage of correctly predicted clean images and correctly predicted stego images was similar: 95% and 100% respectively. But in Test 2, the percentage of correctly predicted clean images and stego images was exactly the same: 96% and 96%. In test 3, the number of stego images was larger than the number of clean images. This was meant to test the effect of increasing the number of images on analysing the results.
The percentage of correctly predicted stego images was much higher than the percentage of correctly predicted clean images, as shown in Table 5.6.

5.3.2. Testing Lossless Format (PNG Images)

Three tests were made to evaluate the length of the hidden files' effectiveness, as follows:

Test 1:

- Using LSB steganography through a developed code written in the Python programming language, 200 clean images and 200 stego images were created.
- The capacity of the hidden file was 10%.
- LSB was used to create the stego images using Python.
- The size of the tested images varied from 14 kb to 162 kb.

Test 2:

- Using LSB steganography through a developed code written in the Python programming language, 200 clean images and 200 stego images were created.
- The capacity of the hidden file was 25%.
- The size of the tested images varied from 14 kb to 162 kb.

Test 3:

Note: In this set, all images from sets 1 and 2 were used to conduct the discriminate analysis test.

- Using LSB steganography through a developed code written in the Python programming language, 200 clean images and 400 stego images were created.
- The capacity of the hidden files were 10% and 25%.
- The size of the tested images varied from 14 kb to 162 kb.

❖ **Analysis of Test 1**

Table 5.8 shows the classification results and the cross-validated percentage for Test 1 when all features are supplied to the system.

The test was able to predict the stego images more efficiently and correctly than the clean images. The percentage of correctly predicted stego images was 99%; the percentage of correctly predicted clean images was 85%.

Table 5.8: Results Table for Test 1 (PNG Images – 10% Hidden File).

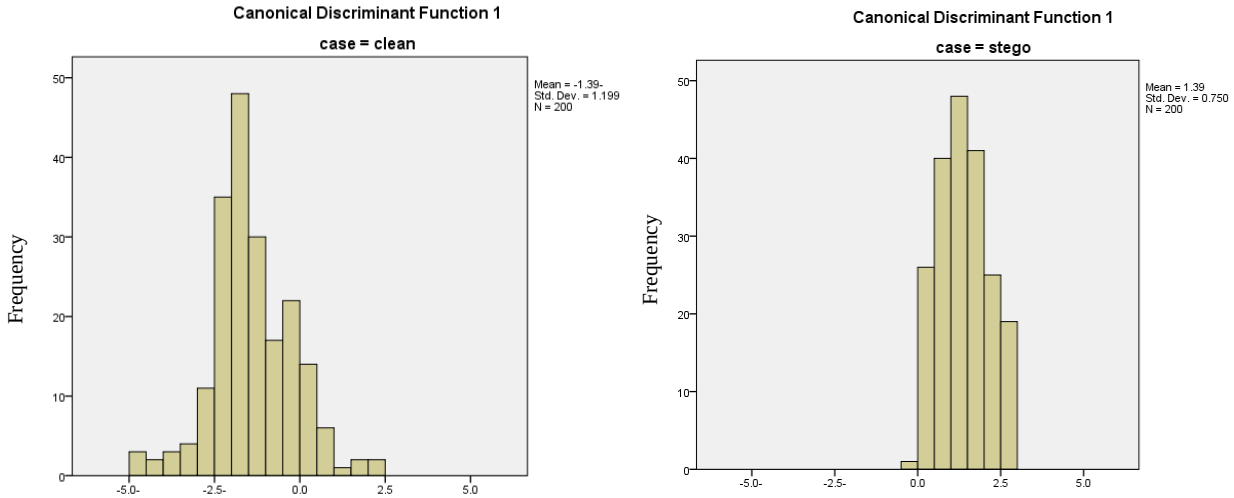
Classification Results[a,c]					
		Case	Predicted Group Membership		Total
			Clean	Stego	
Original	Count	Clean	175	25	200
		Stego	1	199	200
	%	Clean	87.5	12.5	100.0
		Stego	.5	99.5	100.0
Cross-validated[b]	**Count**	**Clean**	**171**	**29**	**200**
		Stego	**2**	**198**	**200**
	%	**Clean**	**85.5**	**14.5**	**100.0**
		Stego	**1.0**	**99.0**	**100.0**

a. 93.5% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 92.2% of cross-validated grouped cases correctly classified.

Figure 5.4 shows the effectiveness of the discriminant function. The two histograms show the distribution of the discriminant function scores for each case. The two distributions only slightly overlap, which means the function discriminates well.



(a) Discrimination Scores. **(b)** Discrimination Scores.

Figure 5.4: Histograms of the Discriminant Function Distribution for Both Cases (Test1, PNG Images).

The range of the clean images is from 2.5 to -5 as shown in figure 5.4 (a), and for the stego images from 0 to 2.8, as presented in figure 5.3 (b). The range of the clean images is clear and doesn't overlap with the stego images from 0 to -5.
However, there is an overlap in the range from 0 to 2.5, the small amount indicating that the distribution function performs well as discriminator.

❖ Analysis of Test 2

In Test 2, the hiding capacity is 25%; Table 5.9 shows the classification results. The cross-validated percentage is 93.5% when all features are tested.

Table 5.9: Results Table for Test 2 (PNG Images – 25% Hidden File).

Classification Results[a,c]					
		Case	Predicted Group Membership		Total
			Clean	Stego	
Original	Count	Clean	180	20	200
		Stego	2	198	200
	%	Clean	90.0	10.0	100.0
		Stego	1.0	99.0	100.0
Cross-validated[b]	**Count**	**Clean**	**176**	**24**	**200**
		Stego	**2**	**198**	**200**
	%	**Clean**	**88.0**	**12.0**	**100.0**
		Stego	**1.0**	**99.0**	**100.0**

a. 94.5% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 93.5% of cross-validated grouped cases correctly classified.

The system was successfully able to predict all stego images better than the clean images. The correct percentage of predicting the stego images was 99%. The error percentage was only 1%. However, it predicted the clean images with a percentage of just 88%. The error percentage of predicting the clean images was 12%. The overall percentage was 93.5%.

Figure 5.5 shows the effectiveness of the discriminant function. The two histograms demonstrate the distribution of the discriminant function scores for each case. The two distributions slightly overlap, which indicates that the function does discriminant well.

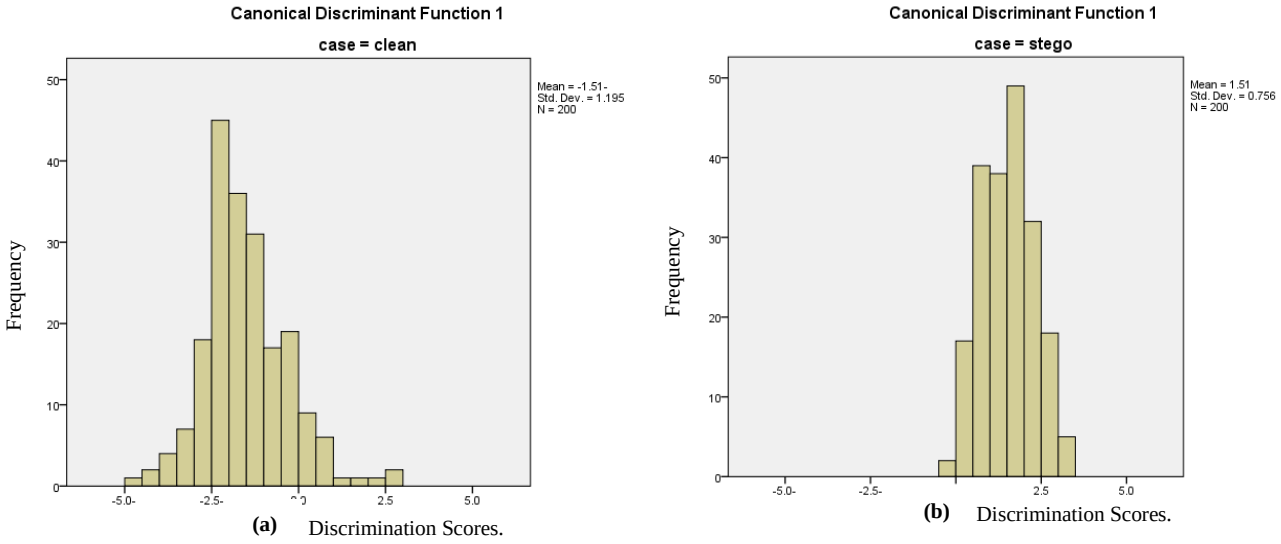


Figure 5.5: Histograms of the Discriminant Function Distribution for Both Cases (Test3, BMP Images).

The range of the clean images is from 2.6 to -5 as shown in figure 5.5 (a), and for the stego images from 0 to 2.8, as present in figure 5.5 (b). The range of the clean images is clear and doesn't overlap with the stego from 0 to -5.

HWhile an overlap is evident from 0 to 2.6, this relatively small amount indicates that the distribution function does a good job at discriminating.

❖ Analysis of Test 3

All stego images used in Test 1 and Test 2 were mixed and tested together in Test 3. Table 5.10 shows the classification results and the cross-validated percentage.

Table 5.10: Results Table for Test 3 (PNG Images – 10%; +25% Stego Images).

Classification Results[a,c]					
		Case	Predicted Group Membership		Total
			clean	stego	
Original	Count	Clean	166	34	200
		Stego	0	400	400
	%	Clean	83.0	17.0	100.0
		Stego	.0	100.0	100.0
Cross-validated[b]	**Count**	**Clean**	**157**	**43**	**200**
		Stego	**0**	**400**	**400**
	%	**Clean**	**78.5**	**21.5**	**100.0**
		Stego	**.0**	**100.0**	**100.0**

a. 94.3% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 92.8% of cross-validated grouped cases correctly classified.

The stepwise DA was successfully able to predict all stego images. It predicted the clean images with a 78% success rate. Adding more clean images to the analysis might help to increase this percentage. The overall percent was 92.8%.

The effectiveness of the discriminant function is shown in Figure 5.6. The two histograms show the distribution of the discriminant function scores for each case.
The two distributions only slightly overlap, which means that the function discriminates well.

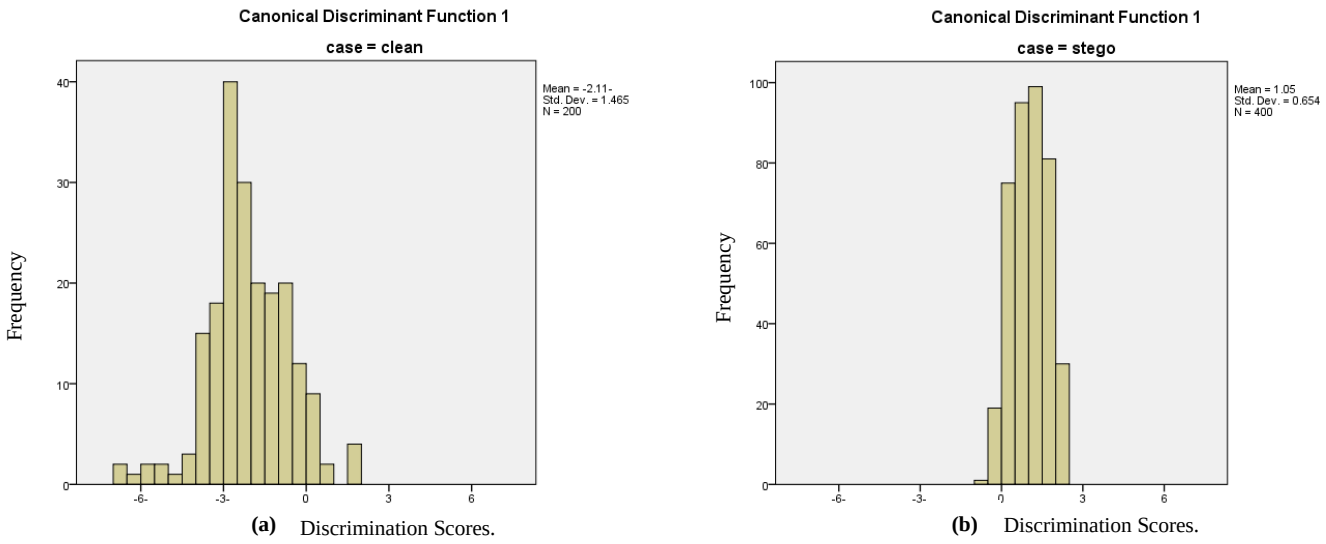


Figure 5.6: Histograms of the Discriminant Function Distribution for Both Cases (Test 3, BMP Images).

The range of the clean images is from 2 to -6.5, as shown in figure 5.6 (a), and for the stego images from -1 to 2.8, as present in figure 5.6 (b). The range of the clean images is clear and doesn't overlap with the stego from 0 to -6.5.

HAlthough there is a range overlap from 0 to 2.8 only, this indicates that the distribution function is a good discriminator.

5.3.3. Comparison between Test 1, Test 2 and Test 3 (PNG Images)

Table 5.11 illustrates that the cross-validated percentages for Test 1, Test 2 and Test 3 are quite similar. The stepwise DA was efficiently able to correctly predict all stego images during the three tests, but for clean images the successful prediction rate fell to just 85% in test 3.

Table 5.11: Comparison between Tests 1, 2 and 3 (PNG Images).

	Cross-Validated %	Correct Percentage of Predicting the Clean Images	Error Percentage of Predicting the Clean Images	Correct Percentage of Predicting the Stego Images	Error Percentage of Predicting the Stego Images	No. of Clean Images	No. of Stego Images	Contributed Features
Test 1	92%	85%	14%	99%	1%	200 Images	200 Images	T1_blue, T2_blue, T3_blue, T4_blue, T6_blue, T9_blue, T3_green, T5_green, T10_green, T11_green, T2_red, T3_red, T7_red, T10_red and T11_red
Test 2	93%	88%	12%	99%	1%	200 Images	200 Images	T1_blue, T2_blue, T3_blue, T4_blue, T6_blue, T9_blue, T1_green,T3_green, T10_green, T11_green, T2_red, T3_red, T7_red, T10_red and 11_red
Test 3	92%	78%	21%	100%	0%	200 Images	400 Images	T1_blue, T2_blue, T3_blue, T6_blue, T7_blue, T10_blue, T11_blue, T2_red, T3_red, T7_red, T10_red and T11_red

Most of the following features contributed to both tests 1 and 2: T1_blue, T2_blue, T3_blue, T4_blue, T6_blue, T9_blue, T3_green, T10_green, T11_green, T2_red, T3_red, T7_red, T10_red and 11_red.

The following features contributed to the analysis of the three tests, 1, 2 and 3: T1_blue, T3_blue, T6_blue, T2_red, T3_red, T7_red, T10_red and T11_red.

Another point to be taken into account is the number of images used in the analysis. In tests 1 and 2, the number of clean images and stego images was the same. This means there was a balance between the two cases: clean and stego. In Test 1, the rate of correctly predicting the clean images was 85%. This was less than the rate of correctly predicting the stego images, which was 99%. From this we can conclude that using an equal number of clean and stego

images in the analysis did not affect the correct percentage for the two cases. Furthermore, the case in Test 2 was similar to Test 1. In Test 3, there were more stego images than clean images. The number of images used in the analysis for the two cases was not the same.

5.4. Classifying the Clean and Stego Images Using Stepwise DA

The aim of this part of the research was to train and test the proposed system to classify all clean and stego images from the database created using the stepwise DA classification method. Experiments presented in this section are different from the ones presented in section 5.3, which aimed at testing the effectiveness of the hidden file sizes only.

All lossless image formats were combined together, and the JPG images were used as an example of lossy format.

The experiments of the lossless and lossy images were conducted separately to allow comparison between the results. Section 5.4.1 presents the results of classifying the lossless images and section 5.4.2 shows the results of classifying the lossy images.

5.4.1. Classifying Lossless Stego Images Created by LSB

In total there were 350 lossless clean images and 600 (lossless) stego-images created. LSB steganography was used to embed the hidden files in the lossless formats.

The images were divided in order to train and test the detection system. In the training phase, 250 clean images and 350 stego were used, and all images were assigned to two categories: 0 (clean) and 1 (stego). The system had count weights based on the values of each of the 33 predictors.

For the testing phase, 100 clean images and 250 stego images were used. Images in the testing phase were assigned as "ungrouped" cases.

The stepwise DA was implemented once as there were no changes in the set of the images tested. Thus, the results would not be changed.

Table 5.12 shows the classification table results and illustrates the grouped and ungrouped cases.

Table 5.12: Results Table of Classifying the Lossless Stego Images Created by LSB.

Classification Results[a,c]

		Case	Predicted Group Membership		Total
			Clean	Stego	
Original	Count	Clean	211	39	250
		Stego	62	288	350
		Ungrouped cases	**127**	**223**	**350**
	%	Clean	84.4	15.6	100.0
		Stego	17.7	82.3	100.0
		Ungrouped cases	**36.3**	**63.7**	**100.0**
Cross-validated[b]	Count	Clean	208	42	250
		Stego	62	288	350
	%	Clean	83.2	16.8	100.0
		Stego	17.7	82.3	100.0

a. 83.2% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 82.7% of cross-validated grouped cases correctly classified.

The system correctly predicted 73 clean images out of 100. In addition, it predicted 196 stego images correctly out of 250. The system achieved a precision rate of 87% and accuracy reached 76%.

Table 5.13 shows the analysis of the ungrouped cases resulted in table 5.12. It represents TP, FP, TN, FN, true positive rate (sensitivity), specificity, precision and accuracy.

Table 5.13: TP, FN, TN, FP, precision and accuracy of the Classification Results of the Lossless Stego Images Created by LSB and Classified using Stepwise DA.

True Positive (TP) & False Positive (FP)	True Negative & False negative (FN)	True Positive Rate (Sensitivity) %	False Positive Rate %	Specificity %	Precision %	Accuracy %
TP: 196 FP: 27	TN: 73 FN: 54	78%	27%	73%	87 %	76%

The system achieved a true positive rate (sensitivity) of 78% with a false positive rate of 27%. The overall performance of the system achieved specificity, precision and accuracy with rates of 73%, 87% and 76% respectively, in terms of predicting stego and clean images.

5.4.2. Classifying Lossy Stego Images Created by the F5 Algorithm

The JPG format was tested as an example of a lossy compression format. The JPG stego images were created using the F5 steganography algorithm.

As in the lossless formats test, the value 0 was assigned to all clean images and 1 to all stego images. The stepwise DA grouped the images into two categories: clean and stego.

During the training phase, the images used for training were given either a 0 or 1 value according to their category. Images used for testing weren't given any category. Thus, in the testing phase, images are known as ungrouped cases.

Table 5.14 shows the classification results and the cross-validated percent.

Table 5.14: Results Table of Classifying the Lossy Stego Images Created by F5 Algorithm.

Classification Results[a,c]

		case	Predicted Group Membership		Total
			clean	stego	
Original	Count	clean	71	29	100
		stego	32	48	80
		Ungrouped cases	**42**	**28**	**70**
	%	clean	71.0	29.0	100.0
		stego	40.0	60.0	100.0
		Ungrouped cases	**60.0**	**40.0**	**100.0**
Cross-validated[b]	Count	clean	70	30	100
		stego	32	48	80
	%	clean	70.0	30.0	100.0
		stego	40.0	60.0	100.0

a. 66.1% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 65.6% of cross-validated grouped cases correctly classified.

The system achieved a more acceptable performance in terms of predicting the clean images than in predicting the stego images. In order to detect JPG stego images, the overall performance of the system needs to be increased as the accuracy reached only 57%, as shown in table 5.15.

The system correctly predicted 31 clean images and 9 stego images.

Table 5.15 shows the analysis of the ungrouped cases resulted in table 5.14. It represents TP, FP, TN, FN, true positive rate (sensitivity), specificity, precision and accuracy.

Table 5.15: TP, FN, TN, FP, precision and accuracy of the Classification Results of the Lossy Stego Images Created by F5 Algorithm and Classified using DA.

True Positive (TP) & False Positive (FP)	True Negative (TN) & False Negative (FN)	True Positive Rate (Sensitivity) %	False Positive Rate %	Specificity %	Precision %	Accuracy %
TP: 9 FP: 19	TN: 31 FN: 11	81%	38%	62%	32%	57%

The system achieved 62% specificity and 32% precision. Also, it achieved 81% true positive (sensitivity) and 38% false positives rate.

5.4.3. Comparisons between Lossless and Lossy Format

Table 5.16 shows a comparison to evaluate the performance of the proposed system. The table illustrates the accuracy of prediction for the lossless and the lossy formats.

Table 5.16: Comparison Table between the Lossless and the lossy Images.

	Cross-Validated Of the Grouped Cases %	Accuracy of Prediction (Ungrouped Cases)	Contributed Features
Lossless Format	82%	76%	T1_red, T5_red, T11_red, T2_green, T3_green, T10_green, T3_blue, T4_blue
Lossy Format	65%	57%	T2_green, T1_blue and T10_blue

The results that 1) the system achieves much better performance with the lossless formats in terms of detection, and 2) the CGCM method is more able to detect LSB steganography.
The performance of detecting the lossy format might improve by increasing the number of the tested images.
Also, using a different method of steganography might affect the results. The F5 algorithm might not leave as obvious a signature as LSB steganogrsphy, which could make it less easy to detect.

As mentioned earlier, more histogram features are going to be added to the system. Adding more types of features will increase the ability of the system to detect different methods of steganography.

5.5. Validating the Results using Stepwise DA

The stepwise DA was run 12 times to validate the accuracy for both the lossless and lossy formats; the images were manually divided into different sets, 500 images for training and 200 for testing. The images selected were changed in each run to avoid repetition.

The validation process was conducted separately for the lossless and lossy formats, as described in the sections below.

5.5.1. Lossless Stego Images Created by LSB

Table 5.17 shows the percentages of validating the results in the training and testing phases for the lossless format. The range of the percentages in the training process is from 80.7 % to 85%, and it's from 68 % to 76 % in the testing process. Percentages in the training process show better performance than the testing process.

In addition, overall accuracy in the training phase is higher than in the testing phase.

Table 5.17. Percent of Validating the Training and Testing Phases for the Lossless Format (12Times Run).

Training Overall Percent	Testing Overall Percent
82.7 %	76 %
80.6%	75%
80.5%	75%
82.8%	74%
84.3%	70%
82.5%	74%
85.7%	68%
84.6%	67%
82.5%	68%
82.7%	70%
82.5%	69%
84.9%	68%
Average: 82%	**Average: 71%**

Figure 5.7 represents all percentages achieved during the validation process in the training and testing phases for the lossless format. It is clear that all percentages in the training phase are much higher than the testing phase.

The blue lines represent the training phase values and the orange lines represent the testing phase values.

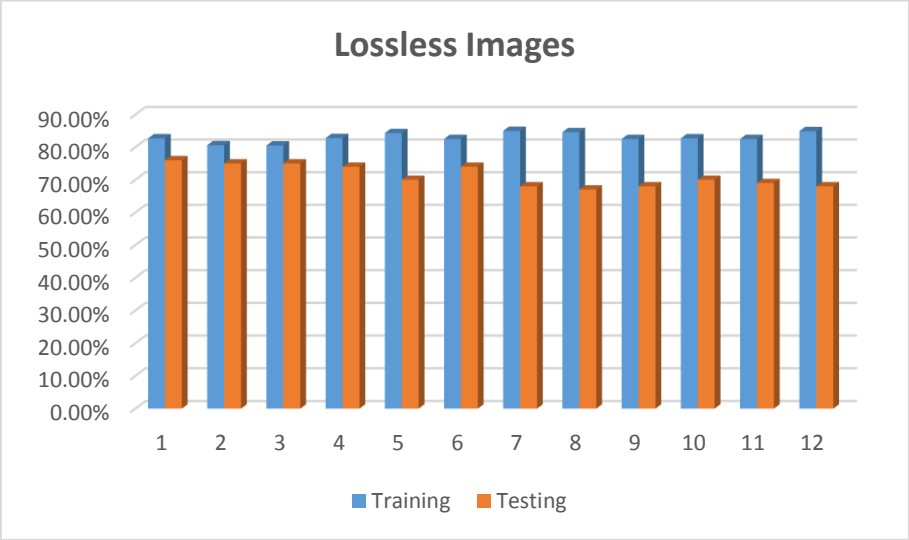


Figure 5.7: Representation of the Percent during the Validation Process for the Lossless Images Stepwise DA.

5.5.2. Lossy Stego Images Created by the F5 Algorithm

Table 5.18 shows the percentages of validating the results in the training and testing phases for the lossless format. The range of the percentages in the training process is from 53% to57 %, and in the testing process it's from 55% to 58%. Percentages in the training process show better performance than in the testing process.

In addition, the overall accuracy in the training phase is higher than in the testing phase.

Table 15.18. Percent of Validating the Training and Testing Phases for the Lossless Images (12 Times Run).

Training Overall Percent	Testing Overall Percent
56.6%	57%
55.7%	57%
56.5%	55%
54.8%	56%
55.5%	56%
55.7%	56%
56.6%	55%
53.4%	56%
53.6%	55%
57.5%	58%
56.4%	57%
56.6%	58%
Overall Accuracy: 55%	**Overall Accuracy: 56%**

All percentages achieved during the validation process in the training and testing phases for the lossy format are shown in Figure 5.8. Most of the percentages achieved in the training phase are greater than the ones in the testing phase. However, some of the percentages are quite close.

The blue lines represent the training phase values and the orange lines represent the testing phase values.

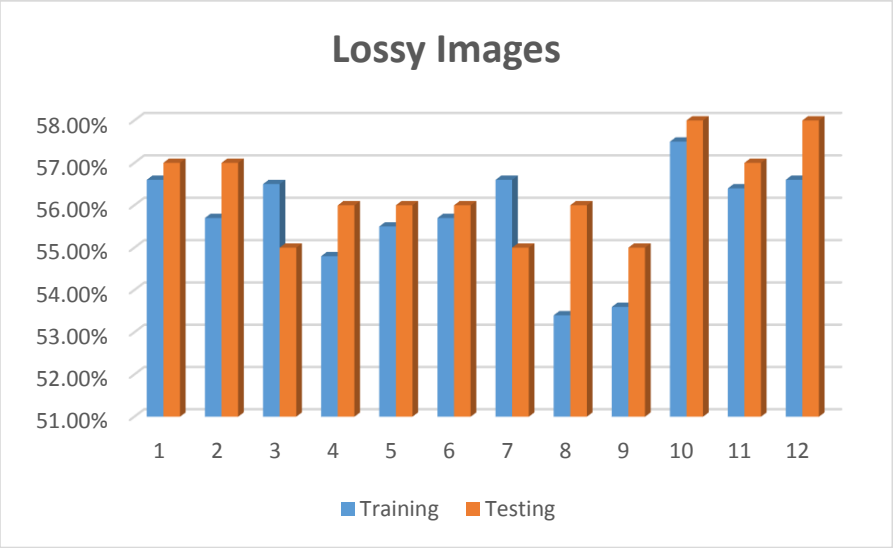


Figure 5.8: Representation of the Percent during the Validation Process for the Lossy Images using Stepwise DA.

5.6. Classifying the Clean and Stego Images Using MLP

In addition to the classifying method described above, MLP was implemented as well, both to independently distinguish the stego images from the clean ones, and – perhaps more importantly – to compare its performance with the stepwise DA classification method.

The following are initial results, as implementing the neural network through the SPSS was found to be not very flexible.

In SPSS, MLP divided the set of images by default. It used 70% of the given images for training and 30% of them for testing. Therefore, it isn't very clear which images exactly were used for training and testing.

5.6.1. Classifying Stego Lossless Images Created by LSB

Table 5.19 shows classification results for the training and the testing phases. MLP implemented the training and the testing processes at the same time.

Table 5.19: The Classification Results after Implementing MLP of the Lossless Stego Images Created by LSB.

Classification

Sample	Observed	Predicted		
		Clean	Stego	Percent Correct
Training	Clean	219	24	90.1%
	Stego	13	402	96.9%
	Overall Percent	**35.3%**	**64.7%**	**94.4%**
Testing	Clean	89	17	84.0%
	Stego	8	177	95.7%
	Overall Percent	**33.3%**	**66.7%**	**91.4%**

Dependent Variable: Case

Focusing on the training phase, 35.3% of the clean images were correctly predicted. In contrast, 64.7% of the stego images were correctly predicted. Thus, the system predicts stego images more accurately than clean images. The overall percentage is 94.4%.

In the testing phase, the system achieved 33.3% in terms of predicting the clean images. However, it predicted the stego images at a higher percentage rate of 66.7%. The overall percentage is 91.4%.

Table 5.20 shows the TP, FN, TN and FP of the results, also, specificity, precision and accuracy of the system.

Table 5.20: TP, FN, TN, FP, precision and accuracy of the Classification Results of the Lossless Stego Images Created by LSB Algorithm and Classified using MLP.

True Positive (TP) & False Positive (FP)	True Negative (TN) & False Negative (FN)	True Positive Rate (Sensitivity) %	False Positive Rate %	Specificity %	Precision %	Accuracy %
TP: 177 FP: 17	TN: 89 FN: 8	95%	16%	83%	95%	91%

The system achieves specificity with percent of 83%, precision with percent of 95% and accuracy that reached 91%.
The system achieves 95% true positive rate (sensitivity) and 16% false positive rate.

Figure 5.9 shows the importance of each single feature (predictor). It is useful for showing the contributing power of each feature (predictor) during the analysis.

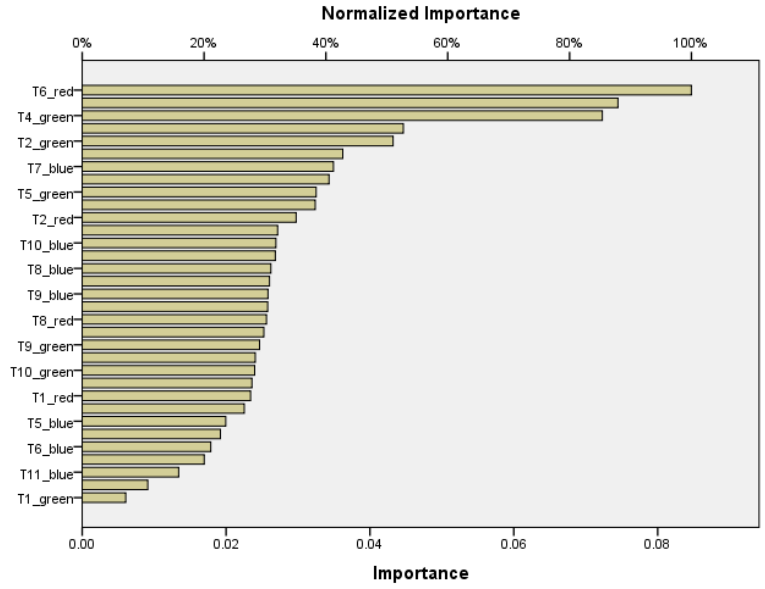


Figure 5.9: The Normalised Importance of the Contributed Features (Lossless Images).

5.6.2. Classifying Stego Lossy Images Created by the F5 Algorithm

Table 5.21 shows classification results for the training and the testing phases.

Table 5.21: The Classification Results after Implementing the MLP of the Lossy Stego Images Created by the F5 Algorithm.

Classification

Sample	Observed	Predicted		
		clean	stego	Percent Correct
Training	clean	99	4	96.1%
	stego	14	52	78.8%
	Overall Percent	**66.9%**	**33.1%**	**89.3%**
Testing	clean	41	6	87.2%
	stego	10	24	70.6%
	Overall Percent	**63.0%**	**37.0%**	**80.2%**

Dependent Variable: case

In the training phase, 66.9% of the clean images were correctly detected. However, the system predicted a lower percentage of the stego images correctly – only 33.1%. The overall percentage is 89.3%.

Focusing on the testing phase, the percentage of clean images that were correctly predicted was 87%. Against that, the percentage of stego images correctly predicted was 70%. This demonstrates that the system achieves much better performance detecting clean images.

For accuracy, all images were analysed after implementing MLP. MLP divided the tested images: 70% for training and 30% for testing. Therefore, the predicted images were checked in general as the MLP doesn't show the exact images used for training and testing.

Table 5.22 shows the TP, FN, TN and FP of the results, also, specificity, precision and accuracy of the CGCM method.

Table 5.22: TP, FN, TN, FP, precision and accuracy of the Classification Results of the Lossy Stego Images Created by F5 Algorithm and Classified using DA.

True Positive (TP) & False Positive (FP)	True Negative (TN) & False Negative (FN)	True Positive Rate (Sensitivity) %	False Positive Rate %	Specificity %	Precision %	Accuracy %
TP: 24 FP: 17	TN: 41 FN: 10	70%	12%	87%	80%	80%

The system achieves specificity with a score of 87%, precision with a score of 80% and accuracy that reached 80%.

The CGCM method achieves a 70% true positive rate (sensitivity) and a 12% false positive rate.

Figure 5.10 shows the importance of each single feature (predictor). They are useful for showing the contributive power of each feature (predictor) during the analysis.

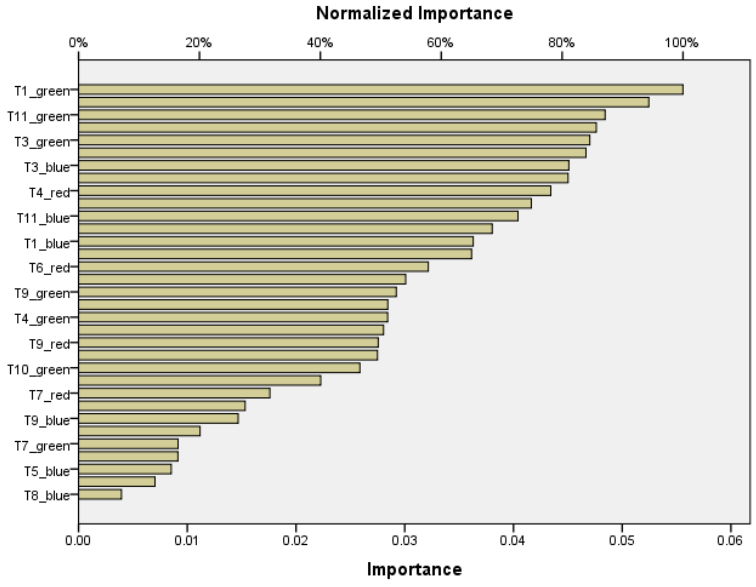


Figure 5.10: The Normalised Importance of the Contributed Features (Lossy Images).

5.7. Validating the Results using MLP

MLP was used to validate the results and the accuracies achieved by the system during the training and the testing phases.

MLP were run 12 times for accuracy; the test automatically divided the images into two sets. It used around 70% of the images for training and 30% for testing.

However, MLP chose different image sets randomly with each run for the training and the testing. Therefore, the results are not the same for each test.

The validation process was implemented for the lossless and lossy formats.

5.7.1. Lossless Stego Images Created using LSB

In this test, all BMP and PNG clean and stego images were used.

AMLP was implemented 12 times and the test achieved dissimilar performance in each of them. Accordingly, the overall accuracy of the training and testing processes were considered as well.

Table 5.23 shows the percentages achieved during the training and the testing phases for each run.

Table 5.23: Representation of the Accuracy Rates during the Validation Process of the Lossless Stego Images Created by LSB.

Training Phase	Testing Phase
99.1%	98.1%
99.6%	98.9%
99.8%	98.1%
99.5%	96.9%
99.8%	96.5%
98.5%	97.2%
99.5%	97.9%
99.3%	95.7%
99.6%	98.9%
99.3%	98.9%
96.4%	95.8%
99.8%	95.6%
Average: 99.18%	**Average: 97.37%**

The overall accuracy for the training phase, reaching 99%, was higher than that attained in the testing phase, which reached 97%.

Figure 5.11 represents all the percentage scores achieved during the validation process in the training and testing phases for the lossless format. It is clear that all percentages in the training phase were much higher than the ones in the testing phase.

The blue lines represent the training phase values and the orange lines represent the testing phase values.

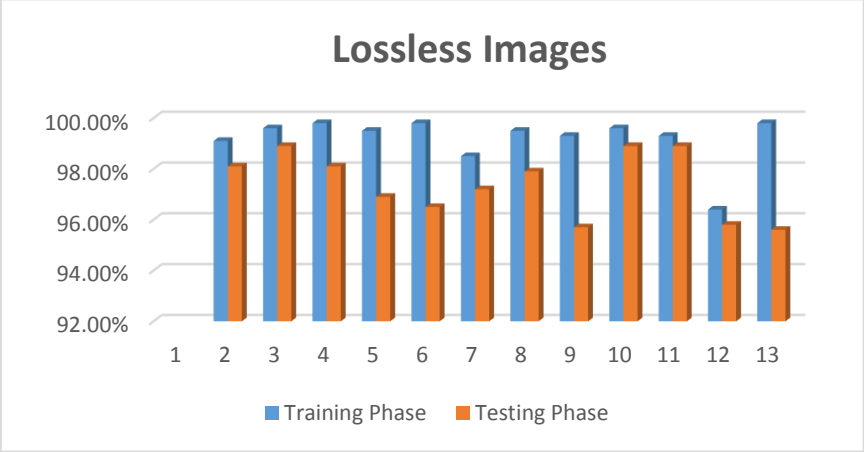


Figure 5.11: Representation of the Percent during the Validation Process for the Lossy Images using MLP.

5.7.2. Lossy Images Created using the F5 Algorithm

In this test, all BMP and PNG clean and stego images were used.

MLP was implemented 12 times and the test achieved dissimilar performance in each of them. Accordingly, the overall accuracy of the training and testing processes were considered as well. As shown in table 5.24, the overall accuracy for the training phase is 89%, and the overall accuracy for the testing phase is 84.4%. This reflects the fact that the system attained higher performance in the training phase than in the testing phase.

Table 5.24: Representation of the Accuracy Rates during the Validation Process of the Lossy Stego Images Created by the F5 Algorithm.

Training Phase	Testing Phase
89.3%	82.8%
88.5%	82.6%
89.9%	85.3%
87.5%	82.6%
89.7%	82.8%
89.6%	85.7%
88.5%	84.6%
89.4%	85.5%
89.7%	85.7%
89.6%	86.6%
88.9%	85.9%
87.6%	82.9%
Average: 89%	**Average: 84.4%**

Figure 5.12 represents all percentages achieved during the validation process in the training and testing phases for the lossy format. It is clear that all percentages in the training phase are much higher than the ones in the testing phase.

The blue lines represent the training phase values and the orange lines represent the testing phase values.

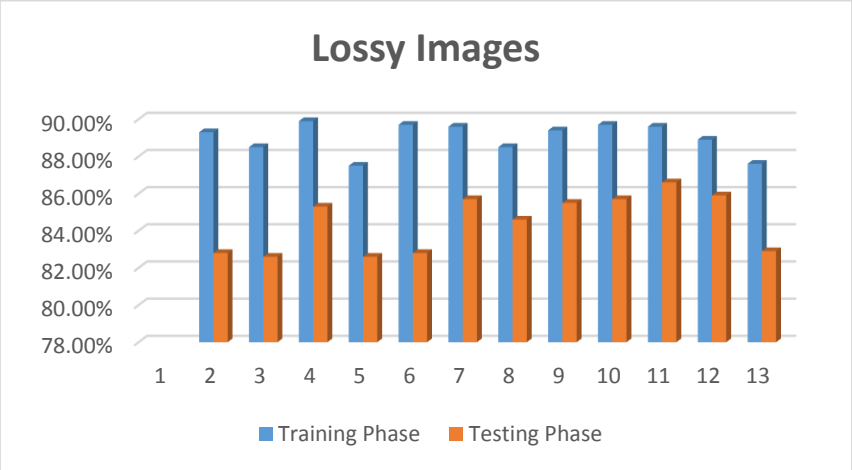


Figure 5.12: Representation of the Accuracy Rates during the Validation Process for the Lossy Images using MLP.

5.7.3. Discussion

Table 5.25 shows the overall percentages achieved by the stepwise DA and MPL classification methods for the lossless and lossy format.
It is clear that MLP performed better than the stepwise DA in both formats during the training and testing phases.

Table 5.25: Overall Accuracies for the Lossless and Lossy Images using Stepwise DA and MLP.

Classification Method	Image Format	Overall Accuracy (Training Phase)	Overall Accuracy (Testing Phase)
DA	Lossless	83%	71%
DA	Lossy	55%	56%
MLP	Losselss	99%	97%
MLP	Lossy	89%	84.4%

The stepwise DA performs better with the lossless format than the lossy format. The performance of the lossy format classified by the stepwise DA is the lowest and needs to be improved.

Comparison of the AUC values for the lossless and lossy formats is shown in table 5.26. All values show outstanding discrimination performance.

Table 5.26: Comparison of the AUC Values for Lossless and Lossy Images with LSB and F5 Steganography.

Steganography Method	Images Types	AUC	Features No.
LSB	Lossless	.999	33 CGCM Features
F5 Algorithm	Lossy	.964	33 CGCM Features

For both cases, the CGCM method achieves excellent detection performance as both AUC values are close to 1.0.

5.8. Area Under Curve

Table 5.27 show the AUC values for the proposed CGCM method and for other pervious steganalysis methods. The proposed CGCM method outperforms all tested steganalyzers when the hiding capacity is 25%. In summary, it can be concluded that the proposed system is better than some of the state-of-the-art algorithms.

Table 5.27: Comparison of the AUC Values of the Proposed System with Previous Methods.

Method	AUC Hiding Capacity: 0.25	Images Type
Our method (33 CGCM features) (LSB Steganography)	**.999**	Colour
Our method (33 CGCM features) (F5 Stegangraphy Algorithm)	**.964**	Colour
Liu's (48 features) (LSB Steganopgraphy)	.8022	Colour
Dong et al. (36 features)	0.618	Grey
Gao et al. (50 features) (LSB Matching)	0.636	Grey
Zhihua et al. (3 features) (LSB Matching Steganography)	0.885	Grey
Cai et al. (70 features) (LSB Matching Steganography)	.844	Grey

The proposed CGCM method achieves higher AUC values than others for grey and colour stego images created by LSB steganography.

Chapter 6

Implementation of the Histogram Features

6.1. Introduction

This chapter describes the experiments conducted by extracting the histogram features, a process which was explained in section 4.4.2.

The first part in the experiments tested the effectiveness of changing the hidden file sizes (see section 6.3). Grey and colour images were tested and the results were classified using the stepwise DA classification method only.

Next, experiments are described in sections 6.4 and 6.5 that were conducted to classify the stego images from the clean ones. In these experiments all images were tested and divided according to their colour and the steganography method used. Booth MLP and stepwise DA classification methods were used to train and test the system.

6.2. Histogram Experiments

This part of the system is based on extracting many histogram features by exploiting the histogram of difference image, which is usually a generalized Gaussian distribution centred at 0. Detection focusing on various types of image features is the most useful technique used for blind steganalysis (Yang et al., 2008). Therefore, the proposed detection system was trained to detect the different types of stego images created by various steganography methods.

6.3. Evaluating the Different Sizes of Hidden Files

A value of 0 was assigned to all clean images, and 1 to all stego images. The stepwise DA classification method grouped the images into two categories: clean and stego. During the training phase, all images were given either a 0 or 1 value according to their category.

Only the stepwise DA classification method was used in the experiments in this section.

Section 6.3.1 explains the results achieved by the system in testing the effectiveness of changing the hidden files sizes for the grey stego images created using LSB steganography. Section 6.3.2 shows the results achieved by the system in testing the effectiveness of changing the hidden files sizes for the colour stego images created using LSB steganography

Finally, sections 6.3.3 and 6.3.4 illustrate the results achieved by the system in testing the effectiveness of changing the hidden files sizes for, respectively, the grey and colour stego images created using the F5 steganography algorithm.

6.3.1. Classification of Grey Stego Images Created by LSB

Two tests were carried out to test the different sizes of the hidden files, as follows:

Test 1

- ✓ 300 clean images were tested.
- ✓ 290 stego images were tested.
- ✓ The hiding capacity was 10%.
- ✓ The size of the clean images ranged from 599 kb to 1 MB.
- ✓ All images were grey.

Test 2

- ✓ 300 clean images were tested.
- ✓ 300 stego images were tested.
- ✓ The hiding capacity was 25%.
- ✓ The size of the images ranged from from 599 kb to 1 MB.
- ✓ All images were grey.

❖ Analysis of Test 1 and Test 2

Table 6.1 shows the classification results of test 1, and table 6.2 illustrates the classification results of test 2. The same small hidden file was embedded into all the tested stego images to ensure accuracy.

Table 6.1: Classification Results of Test 1 using Histogram Features (Small Hidden File; Grey Images).

Classification Results[a,c]

			Predicted Group Membership		
		case	**clean**	**stego**	**Total**
Original	Count	clean	262	38	300
		stego	50	240	290
	%	clean	87.3	12.7	100.0
		stego	17.2	82.8	100.0
Cross-validated[b]	**Count**	**clean**	**259**	**41**	**300**
		stego	**51**	**239**	**290**
	%	**clean**	**86.3**	**13.7**	**100.0**
		stego	**17.6**	**82.4**	**100.0**

a. 85.1% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 84.4% of cross-validated grouped cases correctly classified.

As can be seen in table 6.1, the cross-validated percentage is 84.4% when all histogram features are included in the test.

The stepwise DA was able to correctly predict the stego images at 82.8%. Moreover, it performed at higher percentages when predicting the clean images, i.e. 86.3%. The stepwise DA removed all histogram features that had poor predictive power from the analysis.

In addition, a larger hidden file was embedded into all the tested stego images to ensure accuracy. As can be seen in table 6.2, the cross-validated percentage was 83.2% when all features were incorporated into the test.

Table 6.2: Classification Results of Test 2 using Histogram Features (Large Hidden File, Grey Images).

Classification Results[a,c]

			Predicted Group Membership		
		Case	**clean**	**stego**	**Total**
Original	Count	clean	262	38	300
		stego	60	240	300
	%	clean	87.3	12.7	100.0
		stego	20.0	80.0	100.0
Cross-validated[b]	**Count**	**clean**	**260**	**40**	**300**
		stego	**61**	**239**	**300**
	%	**clean**	**86.7**	**13.3**	**100.0**
		stego	**20.3**	**79.7**	**100.0**

a. 83.7% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 83.2% of cross-validated grouped cases correctly classified.

The stepwise DA was able to correctly predict the stego images at a rate of 79.7%. Moreover, it performed at higher percentages when predicting the clean images (86.7%). The stepwise DA removed all histogram features that have poor predictive power from the analysis.

There is a very small difference between the cross-validation percentage of tests 1 and 2. The percentage of correctly predicting the clean images for both groups is the same.
However, there was a very small difference between the rates at which the stego images were correctly predicted for both groups, namely 82.4% and 79.7% respectively.

6.3.2. Classification of Colour Stego Images Created by LSB

All images included in these tests were colour images, each image having three colour channels: red, green and blue RGB; each of the channels was considered as a single feature.
Three tests were conducted to test the different sizes of the hidden files, as follows:

Test 1

- ✓ 300 clean images were tested.
- ✓ 290 stego images were tested.
- ✓ The hiding capacity was 10%.
- ✓ Sizes of the clean images ranged from 599 kb to 1 MB.

Test 2

- ✓ 300 clean images were tested.
- ✓ 300 stego images were tested.
- ✓ The size of the hidden files was 25% in total.
- ✓ The size of the images ranged from 599 kb to 1 MB.

❖ Analysis of Tests 1 and Test 2

Table 6.3 shows the classification results of test 1, and table 6.4 summarises the classification results of test 2.

The same small hidden file was embedded into all the tested stego images so that accuracy could be assured. As can be seen in table 6.4, the cross-validated percentage was 85.5% when all histogram features were included in the test.

The stepwise DA was able to correctly predict the stego images 85.5% of the time. Moreover, it performed at higher percentages when predicting the clean images as well, i.e. 86%. The stepwise DA removed all histogram features that had poor predictive power from the analysis.

Table 6.3: Classification Results of Test 1 using Histogram Features (Small Hidden File; Colour Images).

Classification Results[a,c]

			Predicted Group Membership		
		Case	**clean**	**stego**	**Total**
Original	Count	clean	259	41	300
		stego	37	253	290
	%	clean	86.3	13.7	100.0
		stego	12.8	87.2	100.0
Cross-validated[b]	**Count**	**clean**	**258**	**42**	**300**
		stego	**41**	**249**	**290**
	%	**clean**	**86.0**	**14.0**	**100.0**
		stego	**14.1**	**85.9**	**100.0**

a. 86.8% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 85.9% of cross-validated grouped cases correctly classified.

Moreover, a larger hidden file was embedded into all the tested stego images to ensure accuracy. As can be seen in table 6.4, the cross-validated percentage was 84.7% when all features were incorporated into the test.

Table 6.4: Classification Results of Test 2 using Histogram Features (Large Hidden File; Colour Images).

Classification Results[a,c]

			Predicted Group Membership		
		Case	clean	stego	**Total**
Original	Count	clean	239	61	300
		stego	28	272	300
	%	clean	79.7	20.3	100.0
		stego	9.3	90.7	100.0
Cross-validated[b]	**Count**	**clean**	**237**	**63**	**300**
		stego	**29**	**271**	**300**
	%	**clean**	**79.0**	**21.0**	**100.0**
		stego	**9.7**	**90.3**	**100.0**

a. 85.2% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 84.7% of cross-validated grouped cases correctly classified.

The stepwise DA was able to correctly predict the clean images at 79%. It performed at higher percentages (90.3%) when predicting the stego images. The stepwise DA removed all histogram features that had poor predictive power from the analysis.

There was a small difference in the cross-validation percentage between test 1 and 2. However, there were noticeable differences between correctly predicting the clean and the stego images for both groups.

The rate of accurately predicting the clean images was 86% for test 1 and 79% for test 2. Also, the percentage of accurately predicting the stego images was 85.9% for test 1 and 90.3% for test 2.

6.3.3. Classification of Grey Stego Images Created by F5 Algorithm

Test 1

- ✓ 100 clean images were tested.
- ✓ 100 stego images.
- ✓ The hiding capacity was 25%.
- ✓ Sizes of the clean images ranged from 300 kb to 1 MB.

Test 2

- ✓ 100 clean images were tested.
- ✓ 100 stego images.
- ✓ The hiding capacity was 25%.
- ✓ The size of the images ranged from 300 kb to 1 MB.

❖ Analysis of Test 1 and Test 2

Table 6.5 shows the classification results of test 1, and table 6.6 illustrates the classification results of test 2.

Table 6.5: Classification Results of Test 2 using Histogram Features (Small Hidden File; Grey Images).

Classification Results[a,c]

			Predicted Group Membership		
		case	**clean**	**stego**	**Total**
Original	Count	clean	72	28	100
		stego	20	80	100
	%	clean	72.0	28.0	100.0
		stego	20.0	80.0	100.0
Cross-validated[b]	**Count**	**clean**	**66**	**34**	**100**
		stego	**25**	**75**	**100**
	%	**clean**	**66.0**	**34.0**	**100.0**
		stego	**25.0**	**75.0**	**100.0**

a. 76.0% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 70.5% of cross-validated grouped cases correctly classified.

The same small hidden file was embedded into all the tested stego images to ensure accuracy. As shown in table 6.5, the cross-validated percentage is 70.5% when all histogram features are incorporated into the test.

The system was able to correctly predict the stego images at 75%. It predicted the clean images with a rate of 66%.

Table 6.6, below, shows the classification results of test 2, where a larger file was hidden in all the stego images used in the test. The cross-validated percent of the classified results is 78%. The system was able to predict the stego images with a high percentage of 81%. Also, it predicted the clean images with a percentage of 75%.

Table 6.6: Classification Results of Test 2 using Histogram Features (Large Hidden File; Grey Images).

Classification Results[a,c]

			Predicted Group Membership		
		case	**clean**	**stego**	**Total**
Original	Count	clean	76	24	100
		stego	17	83	100
	%	clean	76.0	24.0	100.0
		stego	17.0	83.0	100.0
Cross-validated[b]	**Count**	**clean**	**75**	**25**	**100**
		stego	**19**	**81**	**100**
	%	**clean**	**75.0**	**25.0**	**100.0**
		stego	**19.0**	**81.0**	**100.0**

a. 79.5% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 78.0% of cross-validated grouped cases correctly classified.

6.3.4. Classification of Colour Stego Images Created by the F5 Algorithm

Two tests were conducted to test the different sizes of the hidden files, as follows:

Test 1

- ✓ 100 clean images were tested.
- ✓ 100 stego images.
- ✓ The hiding capacity was 10%.
- ✓ Sizes of the clean images ranged from 300 kb to 1 MB.

Test 2

- ✓ 100 clean images were tested.
- ✓ 100 stego images.
- ✓ The hiding capacity was 25%.
- ✓ The images ranged in size from 300 kb to 1 MB.

❖ Analysis of Test 1 and Test 2

Table 6.7 presents the classification results of test 1, and table 6.7 illustrates the classification results of test 2.

The same small hidden file was embedded into all the tested stego images to ensure accuracy. As shown in table 6.8, the cross-validated percentage is 60.5% when all histogram features are included in the analysis.

The stepwise DA was able to correctly predict the stego images at 67%. It predicted the clean images with a percentage of 54%.

Table 6.7: Classification Results of Test 1 using Histogram Features (Small Hidden File; Colour Images).

Classification Results[a,c]

			Predicted Group Membership		
		Case	**clean**	**stego**	**Total**
Original	Count	clean	54	46	100
		stego	33	67	100
	%	clean	54.0	46.0	100.0
		stego	33.0	67.0	100.0
Cross-validated[b]	**Count**	**clean**	**54**	**46**	**100**
		stego	**33**	**67**	**100**
	%	**clean**	**54.0**	**46.0**	**100.0**
		stego	**33.0**	**67.0**	**100.0**

a. 60.5% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 60.5% of cross-validated grouped cases correctly classified.

Table 6.8 shows the classification results of test 2, where a larger file was hidden in all the stego images used in the test. The cross-validated percentage of the classified results is 86.5%. The DA test was able to predict the stego images with a high percentage of 89%. Also, it performed well at predicting the clean images, with a percentage of 84%.

Table 6.8: Classification Results of Group 2 using Histogram Features (Large Hidden File; Colour Images).

Classification Results[a,c]

			Predicted Group Membership		
		case	**clean**	**stego**	**Total**
Original	Count	clean	86	14	100
		stego	9	91	100
	%	clean	86.0	14.0	100.0
		stego	9.0	91.0	100.0
Cross-validated[b]	**Count**	**clean**	**84**	**16**	**100**
		stego	**11**	**89**	**100**
	%	**clean**	**84.0**	**16.0**	**100.0**
		stego	**11.0**	**89.0**	**100.0**

a. 88.5% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 86.5% of cross-validated grouped cases correctly classified.

❖ Analysis of the Results

Table 6.9 shows a summary of the experiments conducted for test 1 in sections 6.3.1 and 6.3.2. The percentages represent the effects of the small hidden files on the results.

There is a noticeable difference between the percentages showing correct prediction of the stego images in grey images as opposed to colour images. The prediction percentage increases from 82% to 85%. The percentages showing correct prediction of the clean images, however, are the same for grey and colour images.

In addition, there is a slight difference in the cross validated percentages, an increase from 84% to 85% when moving from grey to colour images.

Table 6.9: Classification Results of Grey and Colour Images (Small Hidden File).

Steganography Method	Cross-Validated %	Correct Percentage of Predicting the Clean Images	Error Percentage of Predicting the Clean Images	Correct Percentage of Predicting the Stego Images	Error Percentage of Predicting the Stego Images
LSB with Grey Images	**84 %**	86 %	13 %	**82 %**	17 %
LSB with Colour Images	**85 %**	86 %	14 %	**85 %**	14 %

Table 6.10 summarizes the experiments conducted for test 2 in sections 6.3.1 and 6.3.2. The percentages represent the effect of the large hidden files on the results.

There is a huge difference between the percentages showing correct prediction of the clean images. The prediction percentage decreases from 86% with grey images to 79% with the colour images, showing that performance is better in terms of predicting the grey clean images. Another clear difference emerges with regard to stego images, where the prediction percentage clearly increases from 79% with grey images to 90% with the colour images. The performance is higher in terms of predicting the colour stego images.

There is a slight difference in the cross validated percentages for both cases; the percentage increases from 83% to 84%.

Table 6.10: Classification Results of the Grey and Colour Images (Large Hidden File).

Steganography Method	Cross-Validated %	Correct Percentage of Predicting the Clean Images	Error Percentage of Predicting the Clean Images	Correct Percentage of Predicting the Stego Images	Error Percentage of Predicting the Stego Images
LSB with Grey Images	**83 %**	**86 %**	13 %	**79 %**	20 %
LSB with Colour Images	**84 %**	**79 %**	21 %	**90 %**	9 %

Table 6.11 summaries the results of the experiments conducted for test 1 in sections 6.3.3 and 6.3.4. There are many noticeable differences between the results for the grey and the colour images. The cross-validated percent for the colour images is 60.5 %; however, it is higher for the grey images, with a percentage of 70.5 %. In addition, the percentages for correct prediction of clean and the stego grey images (66% and 75% respectively) are both higher than the corresponding percentages for colour images (54% and 67% respectively).

Likewise, a lower percentage of errors were made in predicting both the clean and stego images in the grey images than in the colour images.

Table 6.11: Classification Results of the Grey and Colour Images (Small Hidden File).

Steganography Method	Cross-Validated %	Correct Percentage of Predicting the Clean Images	Error Percentage of Predicting the Clean Images	Correct Percentage of Predicting the Stego Images	Error Percentage of Predicting the Stego Images
F5 Algorithm with Grey Images	**70.5 %**	**66 %**	34 %	**75 %**	25 %
F5 Algorithm with Colour Images	60.5 %	54 %	46 %	67 %	33 %

With regard to the tests involving large hidden files, table 6.12 shows some obvious differences between the results for grey and for colour images.
The cross-validated percentage for the colour images (86.5%) is higher than the percentage for the grey images (78%).

Table 6.12: Classification Results of the Grey and Colour Images (Large Hidden File).

Steganography Method	Cross-Validated %	Correct Percentage of Predicting the Clean Images	Error Percentage of Predicting the Clean Images	Correct Percentage of Predicting the Stego Images	Error Percentage of Predicting the Stego Images
F5 Algorithm with Grey Images	78 %	75 %	25 %	81 %	19 %
F5 Algorithm with Colour Images	**86.5 %**	**84 %**	16 %	**89 %**	11 %

Additionally, the percentages for correct predictions of clean and stego images in colour images (84 and 89% respectively) are both higher than the corresponding percentages in grey images (75 and 81% respectively). Also, a lower percentage of errors were made predicting both clean and the stego images in colour images than in grey images.

Furthermore, a look back at tables 6.10 and 6.11 makes clear that cross-validated percentages increase dramatically when the hidden file becomes larger. This is true for both clean and stego images, grey and colour.

6.4. Classifying the Clean and Stego Images using Stepwise DA

(Classified According to the Steganography Methods)

In this part, all grey and colour images were used in the experiments explained in this section. The tested images were divided into two groups depending on the steganography method used to generate the stego images.

All experiments conducted to train and test the system are shown in detail below.

These experiments were designed to evaluate the performance of the proposed system in terms of its detection abilityand to examine the differences in the results when using different steganography methods to create the stego images.

In the training phase, clean images were given "0" value and stego images were given "1" value. This means that the clean and stego files were identified as such during the stepwise DA. However, in the testing phase, all images were given "null" values, which means the stepwise DA had to classify the images as either stego or clean on the basis of the training. Results of the testing phase appear as "ungrouped cases" in all the tables presenting classification results below.

6.4.1. Classifying Grey Stego Images created by LSB

In this test, all clean and stego images created using LSB steganography were used to train and test the system. The details of the images used in this experiment are the following:

- ✓ The range of the images was from 599 kb to 1 MB.
- ✓ 500 clean images and 500 stego images were used for training.
- ✓ The hiding capacities were 10% and 25%.
- ✓ 200 images were used for testing.

Table 6.13 shows the classification results for the grey images. In the training phase, the system was given 500 clean images and 500 stego images.

The stepwise DA was efficiently able to correctly predict the stego images more often than the clean images. The percentage of correctly predicted stego images was 85.9%; the percentage of correctly predicted clean images was 71%.

In the testing phase, the system was given 200 images for testing, this includes 100 clean and 100 stego images.

Table 6.13: Classification Results of the Clean and Grey Stego Images created by LSB Using Histogram Features.

Classification Results[a,c]

			Predicted Group Membership		
		case	**clean**	**stego**	**Total**
Original	Count	clean	359	141	500
		stego	68	422	500
		Ungrouped cases	**82**	**118**	**200**
	%	clean	71.8	28.2	100.0
		stego	13.9	86.1	100.0
		Ungrouped cases	**41.0**	**59.0**	**100.0**
Cross-validated[b]	Count	clean	355	145	500
		stego	69	421	490
	%	clean	71.0	29.0	100.0
		stego	14.1	85.9	100.0

a. 78.9% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 78.4% of cross-validated grouped cases correctly classified.

The system predicted 82 clean images; however, the number of accurately predicted clean images was 67.

Moreover, the system predicted 118 stego images, while the number of accurately predicted stego images was 85.

Table 6.14 shows the TP, FN, TN and FP of the results, as well as precision, specificity and accuracy of the system.

Table 6.14: TP, FN, TN, FP, precision and accuracy of the Classification Results for the Grey Stego Images Created by LSB and Classified using DA.

True Positive & False Positive (TP & FP)	True Negative & False Negative (TN & FN)	Precision (%)	TP Rate (Sensitivity) %	FP Rate %	Specificity %	Accuracy %
TP: 85 FP: 15	TN: 67 FN: 33	85%	75%	18%	81%	76%

The system achieved 75% true positive (sensitivity) and 18% false positive rates. The system achieved 81% specificity, 85% precision and 76% accuracy.

The stepwise DA provides the best combination of features by performing many steps.
In some stepwise analyses, only the first one or two steps might be taken, even though there are more variables, because it becomes clear that succeeding additional variables do no add to the predictive power of the discriminant function. Figure 6.1 shows the effectiveness of the discriminant function. The two histogram diagrams illustrate the distribution of the discriminant function scores from each case.
When the distributions do not overlap, this suggests that the function discriminates well.

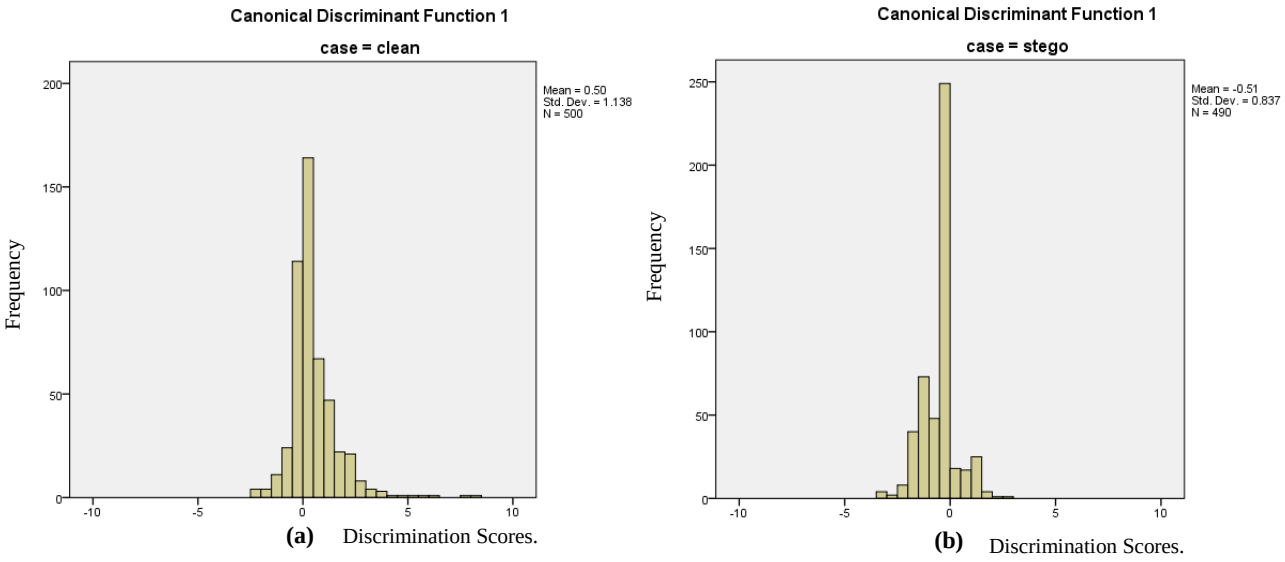


Figure 6.1: Histograms of the Discriminant Function Distribution for Clean Images.

The range for the clean images is from -1.5 to 4, as shown in figure 6.1 (a). For the stego images, it is from -3 to 2, as presented in figure 6.1 (b).
The range of the clean images is clear and it overlaps with the range of the stego images from -1.5 to 1.5 only. From 12.5 to 4, however, it does not overlap with the stego range, which indicates good performance of the discriminate function.

6.4.2. Classifying the Grey Stego Images created by the F5 Algorithm

The details of the images used in this experiment are as follows:

- ✓ The images ranged in size from 599 kb to 1 MB.
- ✓ 500 clean images and 500 stego images were used for training.
- ✓ The hiding capacities of the stego images were 10% and 25%.200 images were used for testing.

Table 6.15 shows the classification results for the grey images. In the testing phase, the system was given 200 images and they were assigned as ungrouped cases.
The test was efficiently able to correctly predict the stego images at a higher percentage level than the clean images.

Table 6.15: Classification Results of the Clean and Stego Images created using F5 Algorith (Grey Images).

Classification Results[a,c]

			Predicted Group Membership		
		case	**clean**	**stego**	**Total**
Original	Count	clean	349	151	500
		stego	43	457	500
		Ungrouped cases	**70**	**130**	**200**
	%	clean	69.8	30.2	100.0
		stego	8.6	91.4	100.0
		Ungrouped cases	**35.0**	**65.0**	**100.0**
Cross-validated[b]	Count	clean	347	153	500
		stego	46	454	500
	%	clean	69.4	30.6	100.0
		stego	9.2	90.8	100.0

a. 80.6% of original grouped cases correctly classified.
b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.
c. 80.1% of cross-validated grouped cases correctly classified.

In the training phase, the percentage of correctly predicted stego images was 90.8%; the percentage of correctly predicted clean images was 69.4%. In the testing phase, the system was able to correctly classify the clean images with parentage of 35% and classifying the stego images correctly with percentage of 65%.
The number of the stego images that were accurately predicted by the system was 92. The accuracy of the system in terms of predicting the clean and stego images was 77%.

Figure 6.2 displays the effectiveness of the discriminant function. The two histograms explain the distribution of the discriminant function scores for each case.

When the distributions do not overlap, this suggests that the function discriminates well.

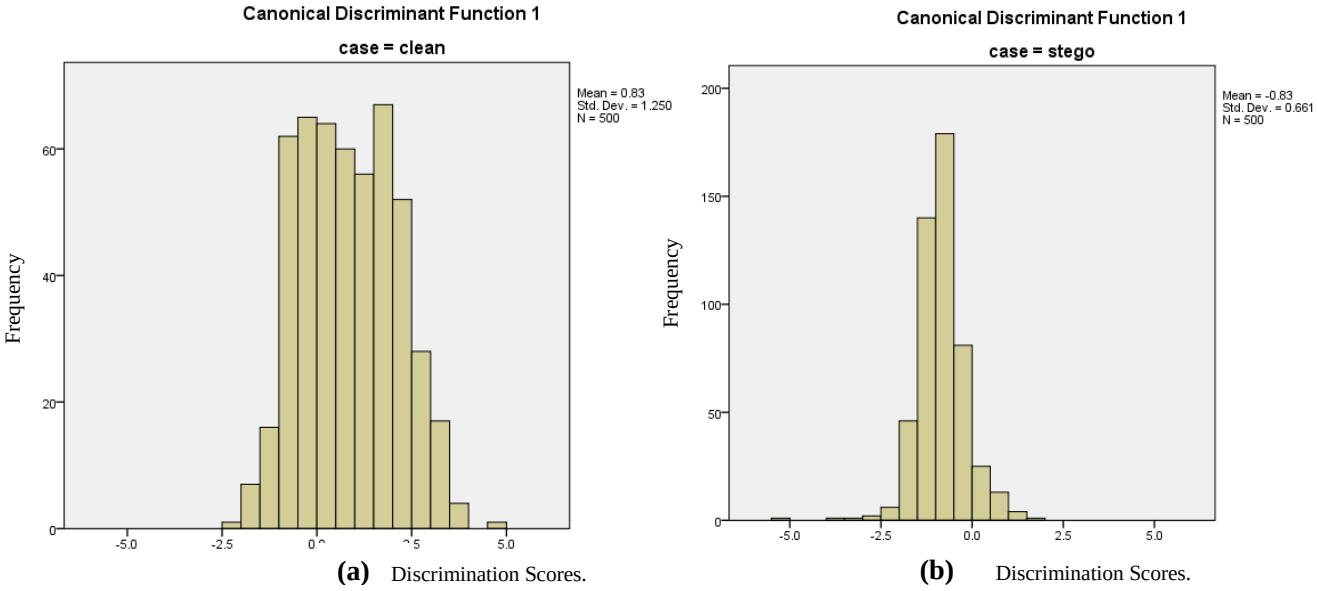


Figure 6.2: Histograms of the Discriminant Function Distribution for Clean and Stego Images

The range of the clean images is from -2.5 to 3.5, as shown in figure 6.2 (a). For stego images, it is from -2.5 to 2, as presented in figure 6.2 (b).

The range of the clean images is clear, and it overlaps with the images range from -2.5 to 1.5. However, it does not overlap with the stego range from 2.5 to 5, which indicates acceptable performance of the discriminate function.

6.4.3. Classifying Colour Stego Images created by LSB

The details of the images used in this experiment are as follows:

- ✓ The images ranged in size from 599 kb to 1 MB.
- ✓ 500 clean images and 490 stego images were used for training.
- ✓ The hiding capacities of the stego images were 10% and 25%.
- ✓ 200 images were used for testing.

Table 6.16 shows the classification results for the colour images. In the training phase, the system was given 500 clean images and 490 stego images. The test was efficiently able to correctly predict the stego images at a higher percentage rate than the clean images. In the testing phase, the system was given 200 images for testing. There were exactly 100 clean images and 100 stego images.

Table 6.16: Classification Results for the Colour Images using Histogram Features.

Classification Results[a,c]

			Predicted Group Membership		
		Case	**clean**	**stego**	**Total**
Original	Count	clean	354	146	500
		stego	44	446	490
		Ungrouped cases	**104**	**96**	**200**
	%	clean	70.8	29.2	100.0
		stego	9.0	91.0	100.0
		Ungrouped cases	**52.0**	**48.0**	**100.0**
Cross-validated[b]	Count	clean	350	150	500
		stego	44	446	490
	%	clean	70.0	30.0	100.0
		stego	9.0	91.0	100.0

a. 80.8% of original grouped cases correctly classified.
b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.
c. 80.4% of cross-validated grouped cases correctly classified.

The percentage of correctly predicted stego images was 91%; the percentage of correctly predicted clean images was 68%.
The system predicted 104 clean images, however, only 76 of them were accurate.
Moreover, the system predicted 96 stego images; however, the number of accurately predicted stego images was 73.

Table 6.17 shows the TP, FN, TN and FP of the results, as well as precision and accuracy of the system.

Table 6.17: TP, FN, TN, FP, precision and accuracy of the Classification Results for the Colour Stego Images Created by LSB and Classified using DA.

True Positive & False Positive (TP & FP)	True Negative & False Negative (TN & FN)	Precision %	TP Rate (Sensitivity) %	FP Rate %	Specificity	Accuracy %
TP: 73 FP: 28	TN: 76 FN: 23	72%	76%	26%	73%	74%

The system achieved a 76% true positive rate (sensitivity) and a 26% false positive rate.
The overall performance of the system achieved 73% specificity, 72% precision and 74% accuracy.

The two histogram diagrams shown in figure 6.3 show the distribution of the discriminant function scores for each case. Keeping in mind that when the distributions do not overlap, this suggests that the function discriminates well, the two diagrams confirm the effectiveness of the discriminant function.

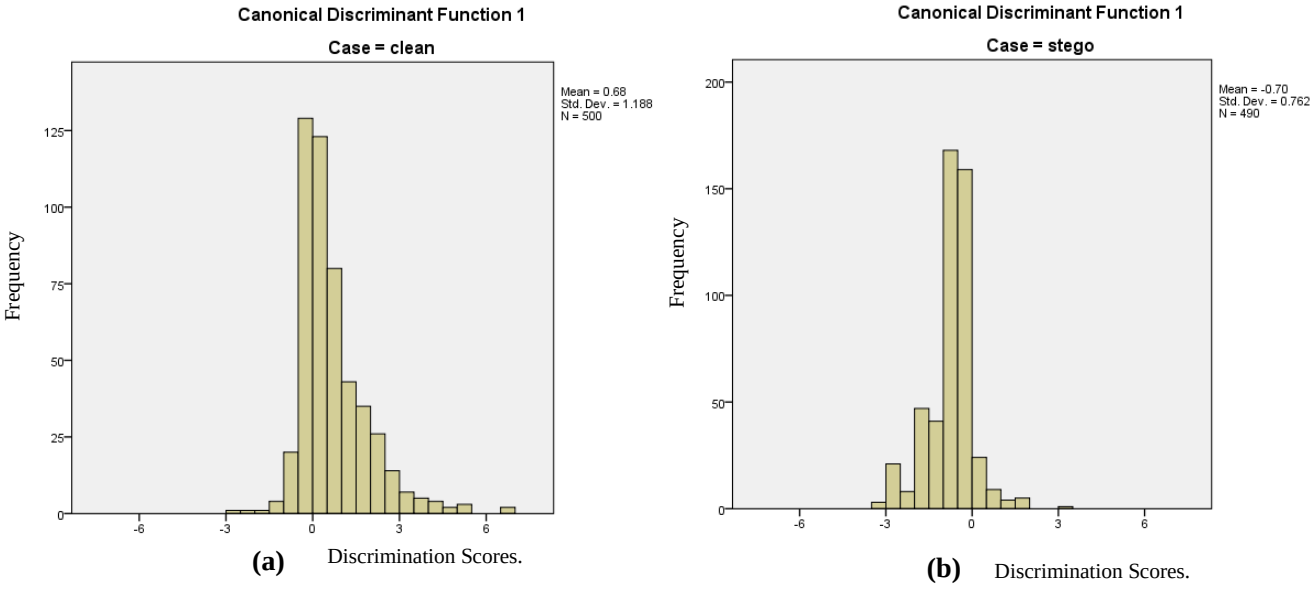


Figure 6.3: Histogram of the Discriminant Function Distribution for Colour Clean and Stego Images created by LSB Steganography.

The range of the clean images is from -2 to 5.0, as shown in figure 6.3 (a), and for the stego images, it is from -3 to 2, as presented in figure 6.3 (b).
The range of the clean images is clear and doesn't overlap with the stego range from 2 to 5.5. Although the range overlaps from -1 to 2, the overall pattern indicates that the distribution function performs well at discrimination.

6.4.4. Classifying Colour Stego Images Created by the F5 Algorithm

The details of the images used are as follows:

- ✓ The images ranged in size was from 100 kb to 1 MB.
- ✓ 500 clean images and 500 stego images were used for training.
- ✓ The hiding capacities of the stego images were 10% and 25%.
- ✓ 200 images were used for testing.

Table 6.18 represents the classification results for the colour images. In the testing phase, the system was given 200 images and they were assigned as ungrouped cases. One hundred of them were clean images and one hundred were stego images.

Table 6.18: Classification Results for the Colour Images (F5 Algorithm).

Classification Results[a,c]

			Predicted Group Membership		
		case	**clean**	**stego**	**Total**
Original	Count	clean	394	106	500
		stego	46	454	500
		Ungrouped cases	**73**	**127**	**200**
	%	clean	78.8	21.2	100.0
		stego	9.2	90.8	100.0
		Ungrouped cases	**36.5**	**63.5**	**100.0**
Cross-validated[b]	Count	clean	387	113	500
		stego	51	449	500
	%	clean	77.4	22.6	100.0
		stego	10.2	89.8	100.0

a. 84.8% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 83.6% of cross-validated grouped cases correctly classified.

The system was able to correctly predict the stego images better than the clean images. The percentage of correctly predicted clean images is 36.5% was not very sufficient. On the other hand, the percentage of correctly predicted clean images was 63.5%.

The system predicted 73 clean images; 64 of which were accurate.

Moreover, the system predicted 127 stego images. The number of accurately predicted stego images was 89.

Table 6.19 shows the TP, FN, TN and FP of the results, also, specificity, precision and accuracy of the system.

Table 6.19: TP, FN, TN, FP, precision and accuracy of the Classification Results for the Grey Stego Images Created by F5 algorithm and Classified using DA.

True Positive (TP) & False Positive (FP)	Ture Negative (TN) False negative (FN)	True Positive Rate (Sensitifity) %	False Positive Rate %	Specificity %	Precision %	Accuracy %
TP: 89 FP: 9	TN: 64 FN: 38	70%	12%	87%	93%	76%

The system achieved a 70% true positive rate (sensitivity) and a 12% false positive rate.

The system achieved 87% specificity, 93% precision and 76% accuracy.

The two histograms shown in figure 6.4 present the distribution of the discriminant function scores for each case. They confirm the effectiveness of the discriminant function.

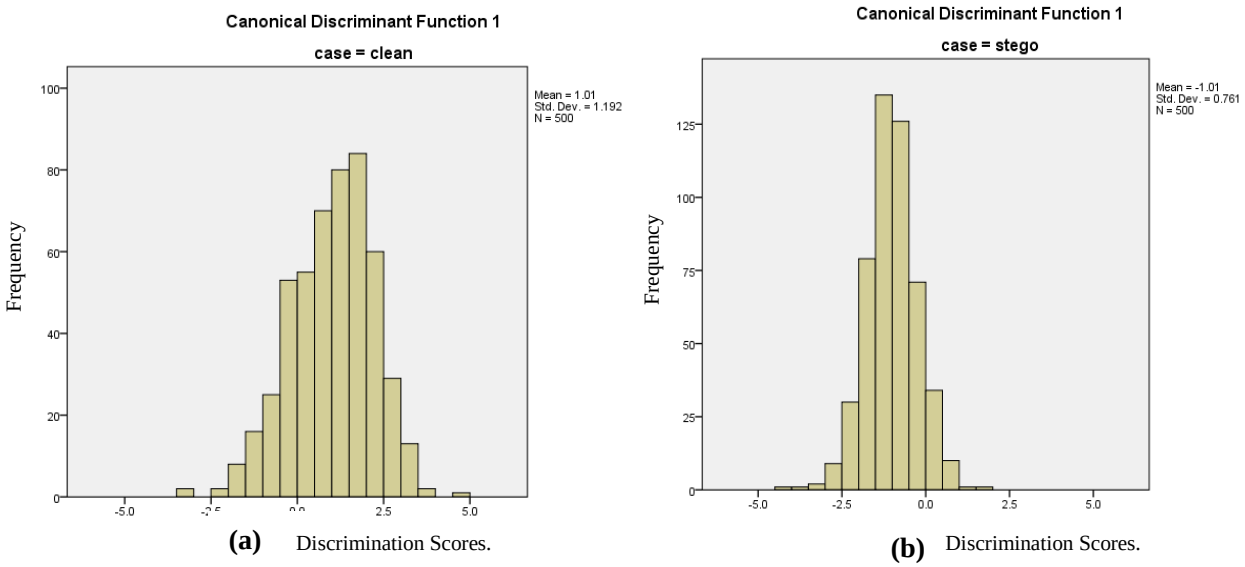


Figure 6.4: Histogram of the Discriminant Function Distribution for Colour Clean Images and Stego Images created by F5 Algorithm.

The range of the clean images is from -2.5 to 5.0 as shown in figure 6.4 (a), and for the stego images it is from -4 to 1.5, as presented in figure 6.4 (b).

The range of the clean images is clear and doesn't overlap with the stego range from 1.5 to 5.0. Even though there is overlap from 1 to -3, the non-overlapping segment is enough to indicate that the distribution function discriminates well.

6.4.5. Analysis of the Results

Table 6.20 summarises the performance of the system for grey images, colour images and the two steganography methods used.

The cross-validated percentages for the colour images are higher than for the grey ones when using LSB and the F5 steganography algorithms.

In addition, the highest accuracy is achieved when the F5 algorithm used with the grey images.

Table 6.20: Comparison Results between stego images created by LSB Steganography and stego images created by the F5 Algorithm.

	Cross-Validated Of the Grouped Cases (Training) %	Accuracy of Prediction (Ungrouped Cases) (Testing) %
LSB Steganography / Grey Images	78 %	75 %
LSB Steganography / Colour Images	**80 %**	74 %
F5 Algorithm / Grey Images	77%	**80%**
F5 Algorithm / Colour Images	**83.6%**	76 %

6.5. Classifying the Clean and Stego Images using Stepwise DA

(Grey and Colour Images)

In this section, all images in the database are used together to train and test the system. The images were divided into two groups only – grey and colour images – to check the differences in the performance. All experiments are illustrated and analysed in sections 6.5.1 and 6.5.2.

6.5.1. Grey Images

Table 6.23 shows the classification results of all grey images that were in the created database. In the training phase, the system was given 1800 images and they were assigned as grouped cases.

The cross-validated section in table 6.21 represents training results.

The system was able to predict the stego images with a percentage rate of 67.9 % and the clean images with a percentage rate of 63.9%.

In the testing phase, the system was given 445 images and they were assigned as ungrouped cases. The actual number of the clean images was 221, and the number of the stego images was 224.

Table 6. 21: Classification Results of the Grey Images Classified by DA using Histogram Features.

Classification Results[a,c]

			Predicted Group Membership		
		case	clean	stego	Total
Original	Count	clean	571	329	900
		stego	260	640	900
		Ungrouped cases	**249**	**196**	**445**
	%	clean	63.4	36.6	100.0
		stego	28.9	71.1	100.0
		Ungrouped cases	**56.0**	**44.0**	**100.0**
Cross-validated[b]	Count	clean	575	325	900
		stego	289	611	900
	%	clean	63.9	36.1	100.0
		stego	32.1	67.9	100.0

a. 67.3% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 65.9% of cross-validated grouped cases correctly classified.

The actual number of the clean images that were accurately predicted by the system was 160 out of 249. Moreover, the number of the stego images that were actually predicted by the system was 152 out of 196.

Table 6.22 shows the TP, FN, TN and FP of the results; also, specificity, precision and accuracy of the system.

Table 6.22: TP, FN, TN, FP, specificity, precision and accuracy of the Classification Results for the Grey Stego Images Classified by DA.

True Positive (TP) & False Positive (FP)	Ture Negative (TN) & False Negative (FN)	True Positive Rate (Sensitivity) %	False Positive Rate %	Specificity %	Precision %	Accuracy %
TP: 152 FP: 89	TN: 160 FN: 44	77%	35%	64%	63%	70%

The system achieved 77% true positive rate (sensitivity) and 35% false positive rate.

The system achieved 64% specificity, 63% precision and 70% accuracy.

6.5.2. Colour Images

Table 6.23 displays the classification results for the colour images. The clean and stego images were divided into training and testing groups: 950 clean and 950 stego images were used in the training phase; 250 clean images and 250 stego were used in the testing phase.

In the training phase, the system achieved a cross-validated percentage of 92.3 %. Additionally, it performed very well in terms of predicting the clean and the stego images, with a percentage of 89% and 94% respectively.

In the testing phase, the system predicted 263 clean images out of 500 images. Of the 263, 212 were accurate predictions.

With respect to the stego images, the system predicted 237 images out of 500, with 199 of the 237 predictions being accurate.

The total accuracy of the system in terms of correctly predicting the clean and stego images was 82%.

Table 6.23: Classification Results of the Colour Images Classified by DA using Histogram Features.

Classification Results[a,c]

			Predicted Group Membership		
		case	clean	stego	Total
Original	Count	clean	258	692	950
		stego	45	905	950
		Ungrouped cases	**263**	**237**	**500**
	%	clean	27.2	72.8	100.0
		stego	4.7	95.3	100.0
		Ungrouped cases	**52.6**	**47.4**	**100.0**
Cross-validated[b]	Count	clean	853	97	950
		stego	50	900	950
	%	clean	89.8	10.2	100.0
		stego	5.3	94.7	100.0

a.61.2% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 92.3% of cross-validated grouped cases correctly classified.

In the training phase, the system achieves 92.3 % cross-validated percentage in the training phase. Additionally, it performes better at predicting the clean images than the stego images, with percentages of 89.8% and 94.7% respectively.

Table 6.24 shows the TP, FN, TN and FP of the results, also, specificity, precision and accuracy of the system.

Table 6.24: TP, FN, TN, FP, specificity, precision and accuracy of the Classification Results for the Colour Stego Images Classified by DA.

True Positive (TP) & False Positive (FP)	Tue Negative (TN) & False negative (FN)	True Positive Rate (Sensitivity) %	False Positive Rate %	Specificity %	Precision %	Accuracy
TP: 199 FP: 51	TN: 212 FN: 38	83%	20%	80%	79%	82.2%

The system achieves 83% true positive rate (sensitivity) and 20% false positive rate.
The system achieves 80% specificity, 79% precision and 82.2% accuracy.

6.6. Classifying Clean and Stego Images Using MLP

The MLP automatically divided the images into two sets. It used around 70% of the images for training and 30% for testing.

6.6.1. Classifying Grey and Colour Stego Images Created by LSB Steganography

- **Grey Stego Images**

In this test, 1200 grey images (600 clean and 600 stego) were used. The stego images that were used were created using LSB steganography. The MLP used 70% of the images for training and 30% of the images for testing.
Table 6.25 shows the classification results for the training and the testing phases of the grey images.

Table 6.25: Classification Results of the Grey Stego Images Created by LSB Classified using MLP.

Classification

Sample	Observed	Predicted		
		clean	stego	Percent Correct
Training	clean	416	72	85.2%
	stego	82	240	74.5%
	Overall Percent	**61.5%**	**38.5%**	**81.0%**
Testing	clean	170	29	85.4%
	stego	62	92	59.7%
	Overall Percent	**65.7%**	**34.3%**	**74.2%**

Dependent Variable: case

Table 6.26 shows the overall percentages achieved in the training and testing phases.

The system achieved 81% in the training phase and 74.2% percent in the testing phase.

Table 6.26: Overall Percentages for the Training and Testing Phases of the Grey Stego Images Created by LSB and Classified using MLP.

Training Overall Percent	Testing Overall Percent
81%	74.2%

❖ Colour Stego Images

Table 6.27 illustrates the percentages for the training and testing phases of the colour images.

In this experiment, 1200 colour images were used: 600 clean images and 600 stego images.

Table 6.27: Classification Results for the Stego Colour Images Created by LSB Classified using MLP.

Classification

Sample	Observed	Predicted		
		clean	stego	Percent Correct
Training	clean	352	56	86.3%
	stego	7	409	98.3%
	Overall Percent	**43.6%**	**56.4%**	**92.4%**
Testing	clean	163	30	84.5%
	stego	6	177	96.7%
	Overall Percent	**44.9%**	**55.1%**	**90.4%**

The overall percentages for the results of the training and testing phases are shown in table 6.28.

The system achieved 92.4 % in the training phase and 90.4% in the testing phase.

Table 6.28: Overall Percentages for the Training and Testing Phases of the Colour Stego Images Created by LSB and Classified using MLP.

Training Overall Percent	Testing Overall Percent
92.4%	90.4%

6.6.2. Classifying the Grey and Colour Stego Images Created by the F5 Algorithm

In this test, 1200 grey images were used: 600 clean images and 600 stego images.

Table 6.29 demonstrates the percentage results for the training and the testing phases.

The system achieved a high accuracy level that reached 83.8% in the training phase and an acceptable accuracy rate of 78.5% in the testing phase.

Table 6.29: Percentages for Validating the Training and Testing Phases for the Grey Stego Images Created by F5 Steganography.

Training Overall Percent	Testing Overall Percent
83.8%	78.5%

Table 6.30, on the other hand**,** shows the percentages for the training and the testing phases of 1200 colour images. In this test, the stego images used were created using the F5 steganography algorithm.

Table 6.30: Percentages for Validating the Training and Testing Phases for the Colour Stego Images Created by F5 Steganography Algorithm.

Training Overall Percent	Testing Overall Percent
88.9%	84%

❖ **Analysis of the Results**

Table 6.31 summarises all accuracy figures for the two steganography methods used for both the grey and colour images.

As one can clearly see, MLP achieved much higher performance rates in terms of classifying the colour stego images created by LSB steganography than it did in classifying the ones created by the F5 algorithm during the training and the testing phases.

The overall percentages for classifying the colour stego images created by the F5 algorithm came in second place, with 88.9% in the training phase and 84% in the testing phase.

Table 6.31: Accuracies for the Two Steganography Methods Used Classififed by MLP.

Steganography Method	Images Types	Overall Accuracy Training	Overall Accuracy Testing
LSB	Grey	81%	74.2%
LSB	Colour	**92.4%**	**90.4%**
F5 Algorithm	Grey	83.8%	78.5%
F5 Algorithm	Colour	88.9%	84%

6.7. Analysis of the Effectiveness of the Histogram Features

The purpose of these experiments were to prove that all extracted histogram features should be included when analysing the results.

In order to test the validity of the 70 histogram features for grey images and 210 histogram features for the colour images used in the analysis, the features were divided into 6 groups and tested separately. Tables 6.32, 6.33 6.34 and 6.35 show the performance of the 6 divided groups and to what extend the number of the features used affected the detection accuracy.

As shown in table 6.32, the highest percentages were achieved while using either 60 or 70 histogram features at the same time to classify the grey images created using LSB.

Table 6.32: Testing the Validation of the Histogram Features used in the Analysis when LSB Steganography was used with the Grey Images.

	Cross-Validated Of the Grouped Cases % (Training)		**Accuracy of Prediction (Ungrouped Cases) % (Testing)**
LSB Grey Images	20 Features	61%	64%
	30 Features	72%	74%
	40 Features	72%	74%
	50 Features	75%	74%
	60 Features	**78%**	**76%**
	70 Features	**78%**	**75 %**

As table 3.33 shows, the highest percentages were achieved while using either 150, 180 and 210 histogram features at the same time to classify the colour stego images created using LSB.

Table 3.33: Testing the Validation of the Histogram Features used in the Analysis when LSB was used with the Colour Images.

	Cross-Validated Of the Grouped Cases % (Training)		**Accuracy of Prediction (Ungrouped Cases) % (Testing)**
LSB Colour Images	30 Features	64%	53%
	60 Features	78%	71%
	90 Features	78%	71%
	120 Features	78%	71%
	150 Features	**80%**	**74%**
	180 Features	**80%**	**74%**
	210 Features	**80%**	**74%**

Likewise, the highest percentages were achieved while using 60 or all 70 histogram features at the same time to classify the grey stego images created using the F5 algorithm, as shown in table 3.34.

Table 6.34: Testing the Validation of the Histogram Features used in the Analysis when F5 Steganography Algorithm was used with the Grey Images.

	Cross-Validated Of the Grouped Cases % (Training)		**Accuracy of Prediction (Ungrouped Cases) % (Testing)**
F5 Algorithm Grey Images	**20 Features**	65.8 %	10%
	30 Features	70.4 %	35 %
	40 Features	76.7 %	75 %
	50 Features	76.7 %	75 %
	60 Features	**80 %**	**78 %**
	70 Features	**80. 1 %**	**77%**

However, with respect to colour stego images created using the F5 algorithm, one can see in table 6.35 that the highest percentages were achieved when using only 60 histogram features.

Table 6.35: Testing the Validation of the Histogram Features used in the Analysis when the F5 Steganography Algorithm was used with the Colour Images.

	Cross-Validated Of the Grouped Cases % (Training)		**Accuracy of Prediction (Ungrouped Cases) % (Testing)**
F5 Algorithm Colour Images	30 Features	75 %	58%
	60 Features	**84%**	**77%**
	90 Features	83.6%	76%
	120 Features	83.6%	76%
	150 Features	83.6%	76%
	180 Features	83.6 %	76%
	210 Features	83.6 %	76%

However, the percentage then became stable when more features contributed to the analysis.

6.8. Validating the Results using Stepwise DA

Stepwise DA was run 12 times to provide accuracy and validation; the images were manually divided into different sets, 900 images for testing and 445 for training. The images selected were changed in each run to avoid repetition.

Below experiments involving grey images are described separately from those involving colour images.

❖ Grey Images

Table 6.36 shows performance percentages for the training and testing processes of the grey images. For the training process, percentages range from 60 % to 74.60 %, and for the testing process, from 55 % to 71 %. Percentages in the training process are higher than in the testing process.

Table 6.36: Percentages for Validating the Training and Testing Phases in the Case of the Grey Images (12 Times Run).

	Training Overall Percent	**Testing Overall Percent**
1	67.30%	70%
2	60%	68%
3	74.60%	66%
4	69.60%	58%
5	62.70%	55%
6	69.30%	63%
7	71.80%	71%
8	62.00%	70%
9	67.40%	56%
10	70%	55%
11	73%	66%
12	69.50%	68%
	Average: 68.1%	**Average: 64%**

The average of all percentages in the training phase is 68%, which is higher than the average of the percentages achieved in the testing phase, which is 64%. All in all, both phases achieved acceptable performance percentages.

Figure 6.6 reconfigures the information presented above in table 6.38, representing in bar-graph format all the percentages achieved during the validation process in the training and testing phases for grey images. This makes the relationship between the two images clearer.

It is clear that most percentages in the training phase are much higher than the corresponding ones in the testing phase.

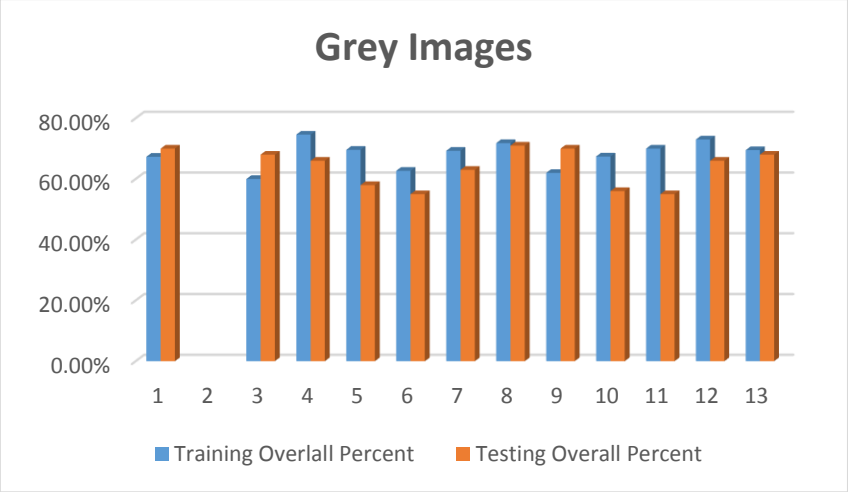


Figure 6.6: Representation of the Percent during the Validation Process for the Grey Stego Images using Stepwise DA.

❖ Colour Images

Performance percentages for the training and testing processes of the colour images are presented in table 6.37. For the training process, percentages range from 72 % to 92.3 %, and for the testing process, from 50% to 72.1%. Percentages in the training process illustrate better performance than those in the testing process.

The average of all percentages in the training phase is 75.54%, which is higher than the average in the testing phase, which is 64.24%.

Table 6.37 Percentages for Validating the Training and Testing Phases in the Case of the Colour Images (Run 12 Times).

	Training Overall Percent	Testing Overall Percent
1	92.3 %	82%
2	75.90%	68.34%
3	77%	62.98%
4	74.00%	62.30%
5	71.70%	60.40%
6	75.6 %	61.2%
7	72.7 %	60.53%
8	72.0 %	72.1%
9	72.5 %	50 %
10	76.20 %	61.2 %
11	73.50 %	62.40 %
12	73.1 %	67.5 %
	Average: 75.54 %	**Average: 64.24 %**

Figure 6.7 reconfigures the information presented above in table 6.37, representing in bar-graph format all percentages achieved during the validation process in the training and testing phases for the colour images. This makes the relationship between the two images clearer.

It is clear that all percentages in the training phase are much higher than the corresponding ones in the testing phase.

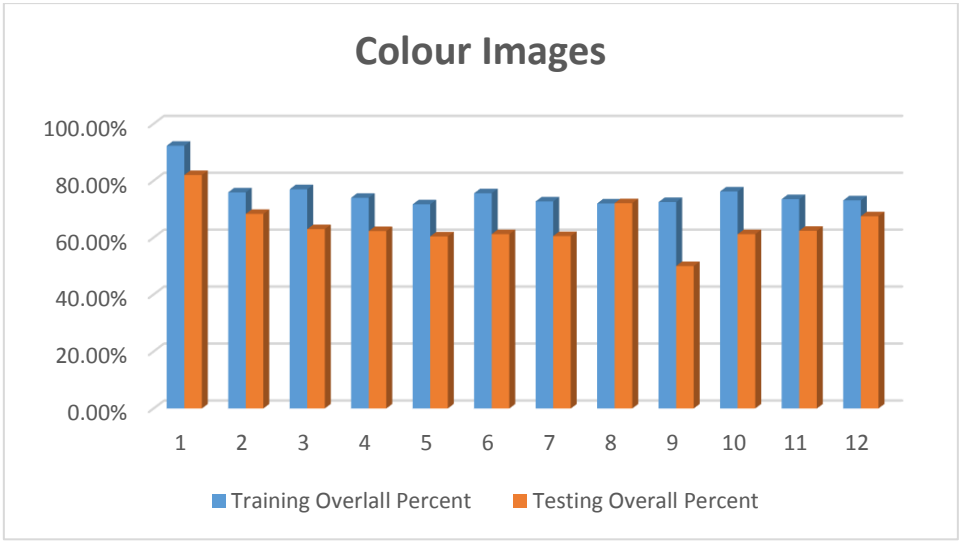


Figure 6.7: Representation of the Percent during the Validation Process for the Colour Images using DA.

6.9. Validating the Results using MLP

The MLP was used to validate the results and the accuracy levels achieved by the system during the training and the testing phases.

The MLP was run 12 times to enhance accuracy; the test automatically divided the images into two sets. It used around 70% of the images for training and 30% for testing.

However, the MLP chose different images with each run for the training and the testing. Therefore, the results are not the same for each test.

6.9.1. Grey Stego Images Created by LSB

In this test, 1200 grey images were used: 600 clean images and 600 stego images.

The MLP was implemented 12 times and the results showed dissimilar performance in each of them. Accordingly, the overall accuracy of the training and testing processes were considered as well.

As shown in table 6.38, the average of all percentages in the training phase was 91.88%, which is higher than the average of the percentages achieved in the testing phase, which was 89.53%.

Table 6.38: Percent of Validating the Training and Testing Phases for the Grey Stego Images Created by LSB (12 Times Run).

	Training Overall Percent	**Testing Overall Percent**
1	96.10%	88.60%
2	93.70%	92.50%
3	94.80%	91.80%
4	89.10%	94.00%
5	91.00%	93.80%
6	84.30%	82.20%
7	89.50%	83.70%
8	90.50%	88.60%
9	90.30%	91.70%
10	91.90%	84.90%
11	94.60%	89.00%
12	96.70%	93.50%
	Average: 91.88%	**Average: 89.53%**

Figure 6.8 represents all percent achieved during the validation process in the training and testing phases for the colour clean and colour stego images created by LSB, which were presented in table 6.38. It is clear that most percentages in the training phase are much higher than corresponding ones in the testing phase.

The blue bars represent the training phase values and the orange bars represent the testing phase values.

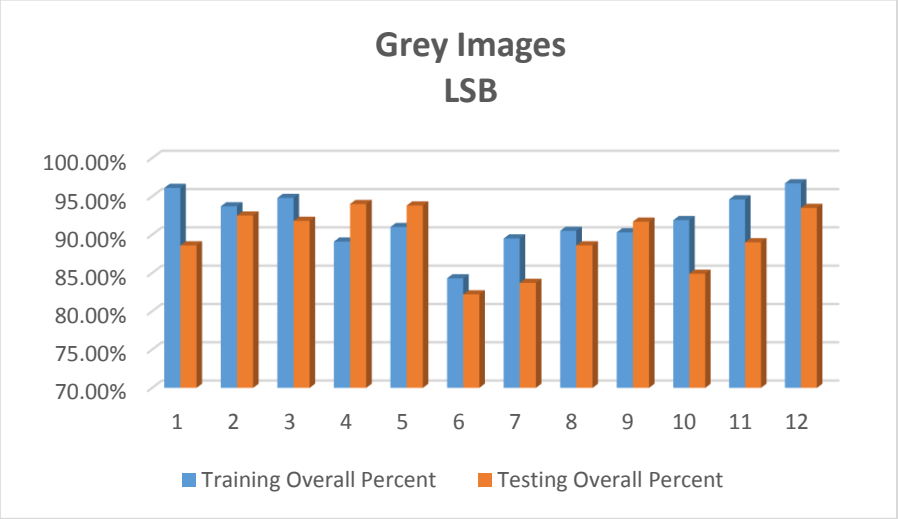


Figure 6.8: Representation of the Percent during the Validation Process for the Grey Stego Created by LSB using MLP.

6.9.2. Colour Stego Images Created by LSB

Table 6.39 illustrates the overall percentage of the training and testing phases of the colour images.

In this experiment, 1200 colour images were examined: 600 clean images and 600 stego images.

MLP was implemented 12 times and the results showed dissimilar performance in each of them. Accordingly, the overall accuracy of the training and testing processes were considered. The average of all percentages in the training phase was 92.18%, and the average of the percentages achieved in testing phase was 96.68%.

This reflects the fact that the system performed unexpectedly well in terms of classifying the clean and stego images.

Table 6.39: Percentages of Validating the Training and Testing Phases for the Colour Stego Images Created by LSB using MLP (12 Times Run).

	Training Overall Percent	Testing Overall Percent
1	99.50%	93.60%
2	100%	89.80%
3	98.50%	89.40%
4	97.80%	91.80%
5	97.90%	94.50%
6	97.40%	98.70%
7	89.70%	89.20%
8	92.50%	95.80%
9	98.90%	90.40%
10	92.20%	92.00%
11	97.80%	83.50%
12	97.90%	97.40%
	Average: 96.68%	**Average: 92.18%**

Figure 6.9 represents all percent achieved during the validation process in the training and testing phases for the colour clean and colour stego images created by LSB, which were presented in table 6.39. This makes the relationship between the two images clearer.

It is clear that all percentages in the training phase are much higher than the corresponding ones in the testing phase.

The blue bars represent the training phase values and the orange bars represent the testing phase values.

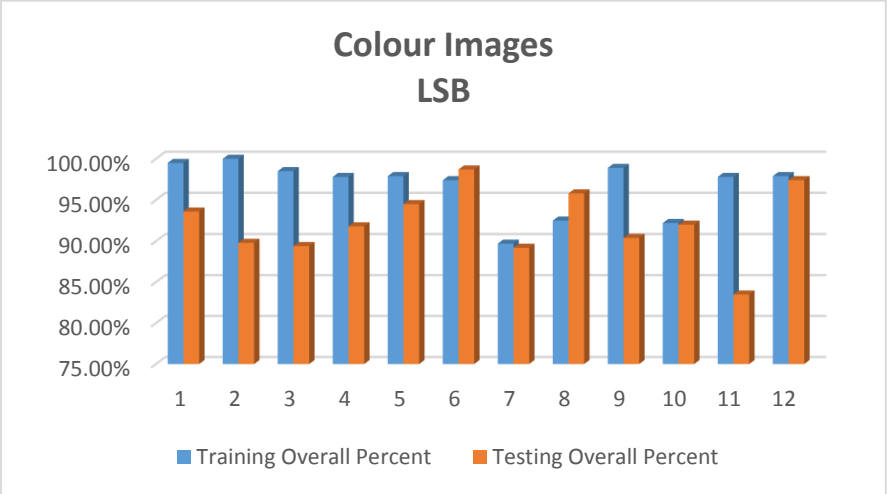


Figure 6.9: Representation of the Percent during the Validation Process for the Colour Stego Images Created by LSB using MLP.

6.9.3. Grey Stego Images Created by the F5 Algorithm

In this test, 1200 grey images were used: 600 clean images and 600 stego images.
Table 6.40 demonstrates the overall percentage for the training and the testing processes.
The MLP was implemented 12 times and the test results showed dissimilar performance in each of them.

Table 6.40: Percentages of Validating the Training and Testing Phases for the Grey Stego Images Created by the F5 Algorithm (12 Times Run).

	Training Overall Percent	**Testing Overall Percent**
1	83.8%	78.5%
2	81.1%	82.5%
3	83.5%	81.5%
4	81.8%	83.2%
5	83.0%	81.4%
6	86.6%	82.2%
7	80.6%	80.7%
8	84.4%	78.3%
9	84.4%	80.6%
10	83.4%	80.7%
11	82.7%	81.5%
12	81.6%	81.1%
	Average: 83.08%	**Average: 81.02%**

The average of all percentages in the training phase was 83.02%, which is higher than the average of the percentages achieved in the testing phase, which was 81.8%.

Figure 6.10 represents all percentages achieved during the validation process in the training and testing phases for the grey clean and grey stego images created by the F5 algorithm, which were presented in table 6.40. It is clear that most percent in the training phase are much higher one in the testing phase.
The blue bars represent the training phase values and the orange bars represent the testing phase values.

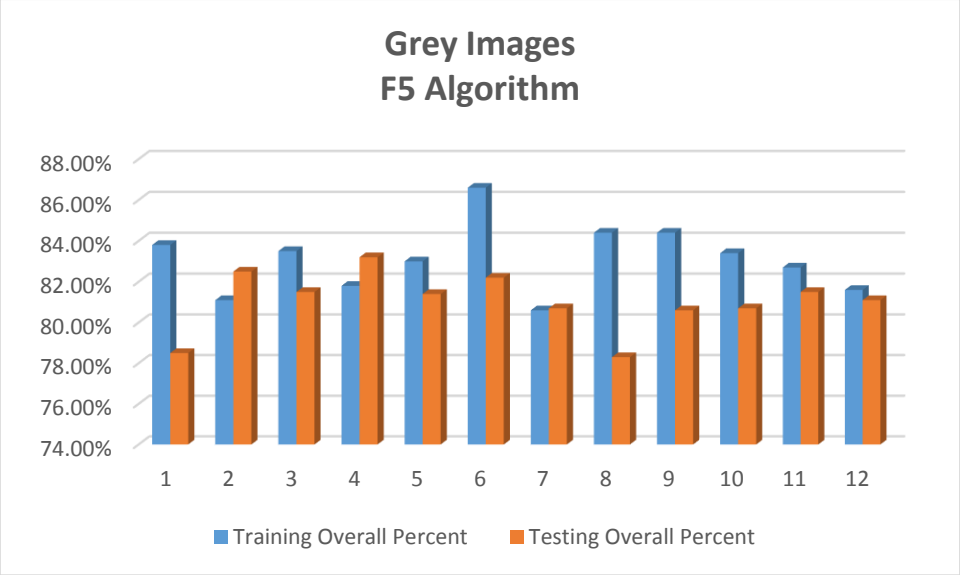


Figure 6.10: Representation of the Percent during the Validation Process for the Grey Stego Images Created by F5 Algorithm using MLP.

Stego colour images are taken up next, with table 6.41 showing the overall percentages for the training and the testing processes for 1200 colour images. In this test, the stego images used were created using the F5 steganography algorithms.

As shown in the table, the average of all percentages in the training phase was 85.14%, which is higher than the average of the percentages achieved in the testing phase, which was 83.73%.

Table 6.41: Percentages of Validating the Training and Testing Phases for the Colour Stego Images Created by the F5 Algorithm (12 Run).

	Training Overall Percent	**Testing Overall Percent**
1	88.9%	84%
2	88.9%	84.1%
3	83.6%	82.7%
4	84.3%	83.9%
5	88.0%	84.3%
6	84.8%	85.3%
7	83.8%	86.7%
8	85.0%	81.7%
9	86.4%	81.9%
10	85.7%	81.0%
11	83.0%	83.6%
12	79.3%	85.6%
	Average: 85.14%	**Average: 83.73%**

As before, the information in table 6.41 is reconfigured in figure 6.11, which represents in a bargraph format all percentages achieved during the validation process in the training and testing phases for the colour clean and colour stego images created by F5 algorithm. As before, it is clear that most percentages in the training phase are much higher than the ones in the testing phase.

The blue bars represent the training phase values and the orange bars represent the testing phase values.

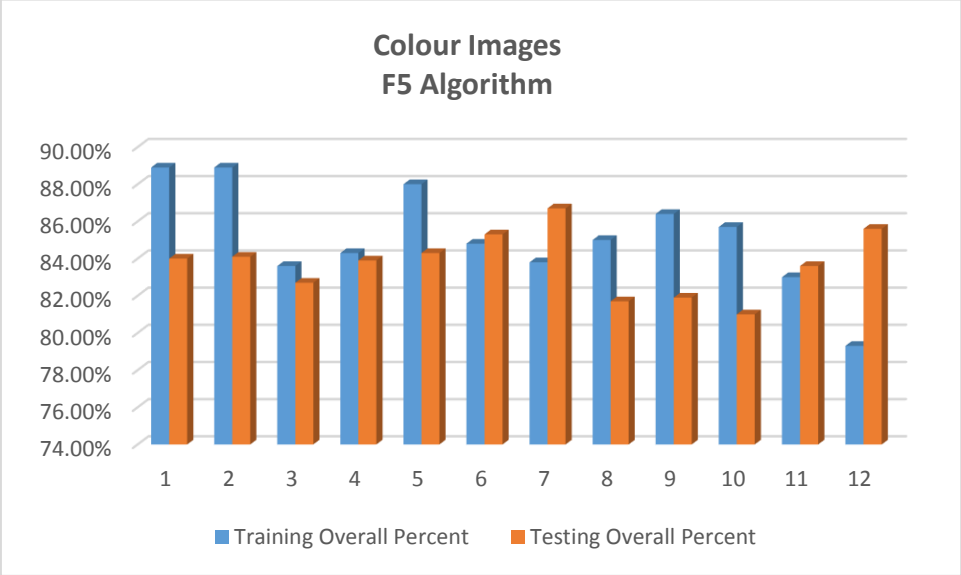


Figure 6.11: Representation of the Percent during the Validation Process for the Colour Stego Images Created by F5 Algorithm using MLP.

❖ Analysis of the Results

Table 6.42 summarises the overall accuracy for the two steganography methods used with the grey and colour images.

As can be seen, during the training and the testing phases the MLP performed much better when classifying LSB steganography than when classifying steganography produced by the F5 algorithm.

Table 6.42: The Percentages of the Overall Accuracies for the Two Steganography Methods Used.

Steganography Method	Images Types	Overall Accuracy Training	Overall Accuracy Testing
LSB	Grey	91%	89%
LSB	Colour	**100 %**	**99%**
F5 Algorithm	Grey	83%	81%
F5 Algorithm	Colour	85%	83%

6.9.4. All Grey Images

Table 6.43 illustrates the overall percentagees for the training and the testing processes. The MLP was implemented 12 times and the test results were dissimilar in each of them.
The average of all percentages in the training phase was 80.64%, which was higher than the average of the percentages achieved in testing phase, which was 76.97%.

Table 6.43: Percent of the Training and Testing Phases for the Grey Images using MLP (12 Times Run).

	Training Overall Percent	**Testing Overall Percent**
1	82.1%	79.1%
2	82.7%	76.2%
3	78.0%	73.5%
4	80.7%	80.4%
5	76.0%	76.5%
6	80.9%	79.8%
7	81.2%	77.8%
8	84.5%	74.8%
9	80.0%	77.3%
10	81.9%	79.1%
11	83.0%	77.1%
12	76.7%	72.0%
	Average: 80.64%	**Average: 76.97%**

Figure 6.12 represents all percent achieved during the validation process in the training and testing phases for the grey images, which were presented in table 6.43. It is clear that all percent in the training phase are much higher one in the testing phase.
The blue bars represent the training phase values and the orange bars represent the testing phase values.

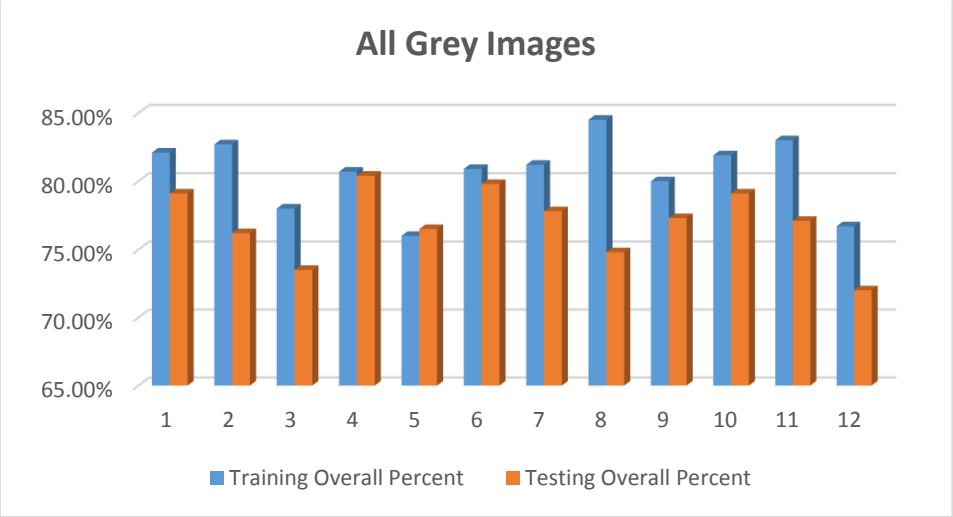


Figure 6.12: Representation of the Percent during the Validation Process for the Gre Images using MLP.

6.9.5. All Colour Images

Table 6.44 illustrates the overall percentages for the training and the testing processes. MLP was implemented 12 times and the test results showed dissimilar performance in each of them. Accordingly, the overall accuracy of the training and testing processes were considered as well. MLP achieved high overall accuracy that reached 89.42% in the training pahse and 86.38% in the testing phase. This demonstrates that the system performed very well when classifying the clean and stego images.

Table 6.44: Percentages of the Training and Testing Phases for the Colour Images using MLP (12 Runs).

	Training Overall Percent	**Testing Overall Percent**
1	89.9%	85.1%
2	89.6%	86.6%
3	92.4%	90.4%
4	90.2%	87.3%
5	89.1%	85.2%
6	85.6%	86.1%
7	85.7%	82.2%
8	88.5%	87.2%
9	91.4%	85.6%
10	88.3%	85.0%
11	89.6 %	87.9 %
12	92.7 %	88.0 %
	Average: 89.42%	**Average: 86.38%**

Figure 6.13 represents in bar graph format the information in table 6.44, including all percentages achieved during the validation process in the training and testing phases for the colour images. It is clear that all percentages in the training phase are much higher than the corresponding ones in the testing phase

The blue bars represent the training phase values and the orange bars represent the testing phase values.

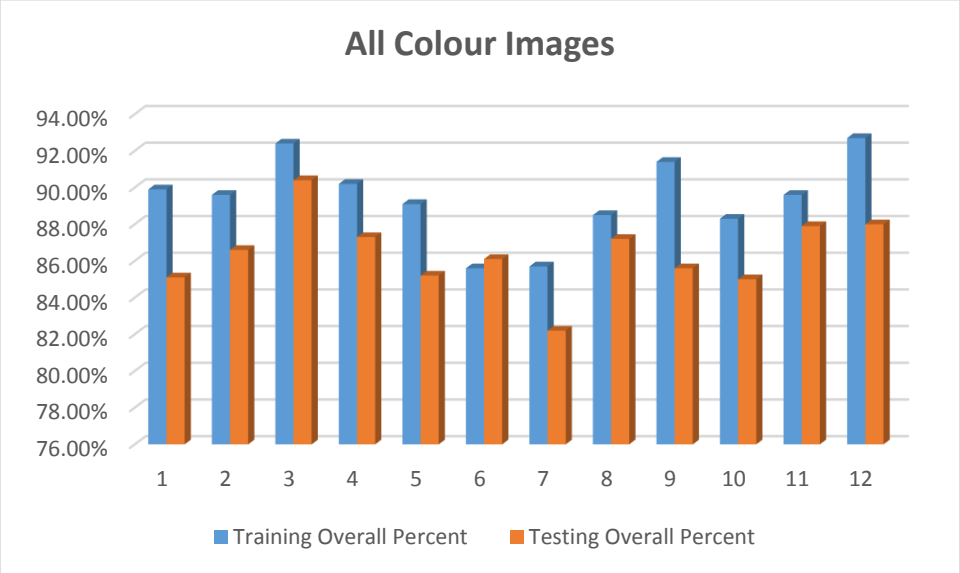


Figure 6.13: Representation of the Percent during the Validation Process for the Colour Images Using MLP.

❖ Analysis of the Results

As shown in table 6.45, the MLP achieved higher accuracy during the training and the testing phases with the colour images than with the grey images.

Table 6.45: Overall Accuracies for the Grey and the Colour Images.

Images Types	Overall Accuracy Training	Overall Accuracy Testing
Grey	80%	76%
Colour	86%	89%

6.10. Testing the System with Larger Images

Two additional expermients conducted to test the ability of the system to deal with huge images's sizses.

The results of the first experiment are shown in table 6.46. Five clean and five stego images are included in this test.

The images tested's sizes is larger than 4 MB and all images are coloured with 24 bit. The S-Tools used to create the stego images and the hidden file zise is 604 kb. The histogram features only extracted from the images and classified by stepwise DA.

Table 6.46: Classification Rsults of the Extra-Large Images.

Classification Results[a,c]					
			Predicted Group Membership		
		70	clean	stego	Total
Original	Count	clean	5	0	5
		stego	0	5	5
	%	clean	100.0	.0	100.0
		stego	.0	100.0	100.0
Cross-validated[b]	Count	clean	5	0	5
		stego	0	5	5
	%	clean	100.0	.0	100.0
		stego	.0	100.0	100.0
a. 100.0% of original grouped cases correctly classified.					
b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.					
c. 100.0% of cross-validated grouped cases correctly classified.					

The results prove that the system correctly predict all clean and stego image during the training and the testing phases.

In addition, these ten images were mix with twenty lossless images from the created images database.

Table 6.47: Classification Rsults of the Extra-Large Images Mixed with Previous Images.

Classification Results[a,c]					
			Predicted Group Membership		
		case	clean	stego	Total
Original	Count	clean	16	4	20
		stego	2	18	20
	%	clean	80.0	20.0	100.0
		stego	10.0	90.0	100.0
Cross-validated[b]	Count	clean	14	6	20
		stego	2	18	20
	%	clean	70.0	30.0	100.0
		stego	10.0	90.0	100.0
a. 85.0% of original grouped cases correctly classified.					
b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.					
c. 80.0% of cross-validated grouped cases correctly classified.					

The system correctly predicted the clean images with 70%. In addition, it predicted the stego images with percent of 90%. The cross-validation percent is 80%.

The experiments prove that the system can support all type of images regardless of their sizes.

6.11. Area under Curve (AUC)

A comparison of the AUC values for the LSB and F5 steganography algorithm is shown in table 6.47; all values show outstanding discrimination performance.
A larger AUC value means better detection performance. An AUC value close to 1.0 indicates excellent discrimination, while a value close to 0.5 indicates poor discrimination.

Table 6.47: Comparison of the AUC Values for Grey and Colour Images using MLP with LSB and F5 Steganography.

Steganography Method	Images Types	AUC	Features No.
LSB	Grey	.963	70 Features (n = 4)
LSB	Colour	.998	70 Features (n = 4)
F5 Algorithm	Grey	.916	70 Features (n = 4)
F5 Algorithm	Colour	.938	70 Features (n = 4)
LSB & F5 Algorithm	Grey	.890	70 Features (n = 4)
LSB & F5 Algorithm	Colour	.988	70 Features (n = 4)

Table 6.48 show the AUC values for the proposed system and for other pervious steganalyzers. For the two different hiding capacities, the proposed system outperforms all tested steganalyzers. In summary, it can be concluded that the proposed system is better than some of the state-of-the-art algorithms for these two hiding capacities.

Table 6.48: Comparison of the AUC Values of the Proposed System with Previous Methods.

Method	AUC Hiding Capacity: 0.1	AUC Hiding Capacity: 0.25	Images Type
Our method (n=4) (70 features) (LSB Steganography)	**.916**	**.963**	Grey
Our method (n=4) (70 features) (F5 Steganography Algorithm)	**.938**	**.916**	Colour
Cai et al. (n=4) (70 features) (LSB Matching Steganography)	0.669	.762	Grey
Gao et al. (50 features) (LSB Matching Steganography)	0.950	0.639	Grey
Dong et al. (36 features)	0.567	0.618	Grey

Chapter 7

Implementation of the Joint Features

7.1. Introduction

This chapter summarizes the implementation of merging all CGCM and histogram features. In addition, it describes the experiments conducted and compares the accuracy attained with the ones achieved by implementing the CGCM and histogram features separately.

7.2. Experiments

All CGCM and histogram features explained in section 4.4.1 and 4.4.2 were joined together in one method to improve accuracy and performance. They became 243 features in total after merging.

7.2.1. Classifying Colour Images using Stepwise DA

Table 7.1 shows the classification results for 1,209 colour images that were randomly chosen from the created database. Images were divided between the training and testing phase.

Table 7.1: Classification Results of the Clean and Stego Images Classified by Merged CGCM and Histogram features.

Classification Results[a,c]

			Predicted Group Membership		
		Case	Clean	Stego	Total
Original	Count	Clean	406	119	525
		Stego	5	392	397
		Ungrouped cases	**126**	**161**	**287**
	%	Clean	77.3	22.7	100.0
		Stego	1.3	98.7	100.0
		Ungrouped cases	**43.9**	**56.1**	**100.0**
Cross-validated[b]	Count	Clean	432	93	525
		Stego	8	389	397
	%	Clean	82.3	17.7	100.0
		Stego	2.0	98.0	100.0

a. 86.6% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 89.0% of cross-validated grouped cases correctly classified.

In the training phase, the system was given 525 clean images and 397 stego images. The test was able to correctly predict both types of images efficiently. However, while the correct prediction rate for clean images was 82%, it was an impressive percentage of 98% for stego images.

The overall cross-validated percentage achieved by the system in the training phase was 89%.

In the testing phase, the system was given 287 images, consisting of 100 clean images and 187 stego images.

Even though there were only 100 clean images, the system predicted 126, misclassifying 26 images with percentage of 43%.

Moreover, the system was able to predict 161 stego images correctly, yielding a percentage of 56%.

Table 7.2 illustrate the TP, FN, TN, FP, true positive rate (sensitivity) and false positive rate of the results; also, specificity, precision and accuracy of the system.

Table 7.2: TP, FN, TN, FP, Specificity, Precision and Accuracy of the Results (Testing Phase).

True Positive (TP) & True Negative (TN)	False Positive (FP) & False Negative (FN)	True Positive Rate (Sensitivity) %	False Positive Rate %	Specificity %	Precision %	Accuracy %
TP: 161 TN: 100	FP: 26 FN: 0	100%	20%	98%	86%	91%

The overall performance of the system achieved 98% specificity, 86% precision and 91% accuracy.

In addition, the system achieved an extraordinary true positive rate (sensitivity) of 100%.

Against this, its false positive rate was 20%.

7.2.2. Classifying Colour Images using MLP

The MLP classification method automatically divided the images into two groups, 70% for the training phase and 30% for the testing phase, see table 7.3.

Table 7.3: Case Processing Summary.

		N	Percent
Sample	Training	297	70.0%
	Testing	127	30.0%
Valid		424	100.0%
Excluded		2002	
Total		2426	

Table 7.4 shows the classification results after implementing the MLP.

The overall percentage achieved by the system for predicting the clean images was 47.1% during the training phase. The overall rate predicting the stego images was 52.9 %.

In the testing phase, the system was able to predict the clean images with an overall percentage of 54.3 %. It was able to predict the stego images with an overall percentage of 45.7 %.

Table 7.4: The Classification Results of the Colour Images using Merged Featured and classified by MLP.

Classification

		Predicted		
Sample	**Observed**	**Clean**	**Stego**	**Percent Correct**
Training	Clean	140	0	100.0%
	Stego	0	157	100.0%
	Overall Percent	**47.1%**	**52.9%**	**100.0%**
Testing	Clean	68	1	98.6%
	Stego	1	57	98.3%
	Overall Percent	**54.3%**	**45.7%**	**98.4%**

Dependent Variable: Case

Table 7.5 illustrates the TP, FN, TN, FP, true positive rate and false positive rate of the results, also, specificity, precision and accuracy of the system.

Table 7.5: TP, FN, TN, FP of the Results (Testing Phase).

True Positive (TP) & True Negative (TN)	False Positive (FP) & False Negative (FN)	True Positive Rate (Sensitivity) %	False Positive Rate %	Specificity %	Precision %	Accuracy %
TP: 57 TN: 68	FP: 1 FN: 1	98%	1%	98.5%	98.2%	98.4%

The system performed excellently, reaching 98% specificity, 98% precision and 98% accuracy in terms of predicting stego and clean images.

In addition, the system achieved a high true positive rate (sensitivity) that reached 98%, while at the same time maintaining the very low false positive rate of only 1%.

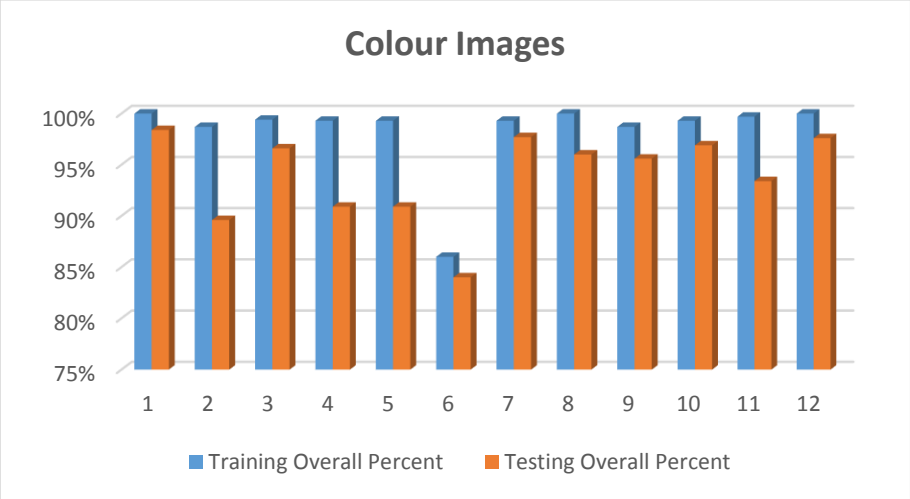
7.2.3. Validating the Results using MLP

Table 7.6 illustrates the overall percentages for the training and the testing processes. The MLP was run 12 times and the results showed dissimilar performance in each of them.
The average of all percentages in the training phase was 98%, which is much higher than the average achieved in the testing phase, which was 93%.

Table 7.6: Percentages for the Training and Testing Phases for the Colour Images classified by MLP (12 Times Run).

	Training Overall Percent	Testing Overall Percent
1	100 %	98.4%
2	98.7%	89.6%
3	99.4%	96.6%
4	99.3%	90.9%
5	99.3%	90.9%
6	86%	84%
7	99.3%	97.7%
8	100%	96%
9	98.7%	95.6%
10	99.3%	96.9%
11	99.7%	93.4%
12	100%	97.6%
	Average: 98%	**Average : 93%**

Figure 7.1 illustrates the percentages presented in table 7.6. The blue bars represent the training phase percentages and the orange bars represent the testing phase percentages.
All percentages in the training phase were higher than the percentages in the testing phase.



F Figure 7.1: Representation of the Percent during the Validation Process for Colour Images Classified by MLP for the 12 runs.

Figure 7.2 shows the importance of each single feature (predictor). It is helpful to show the contribution power of each feature (predictor) during the analysis.

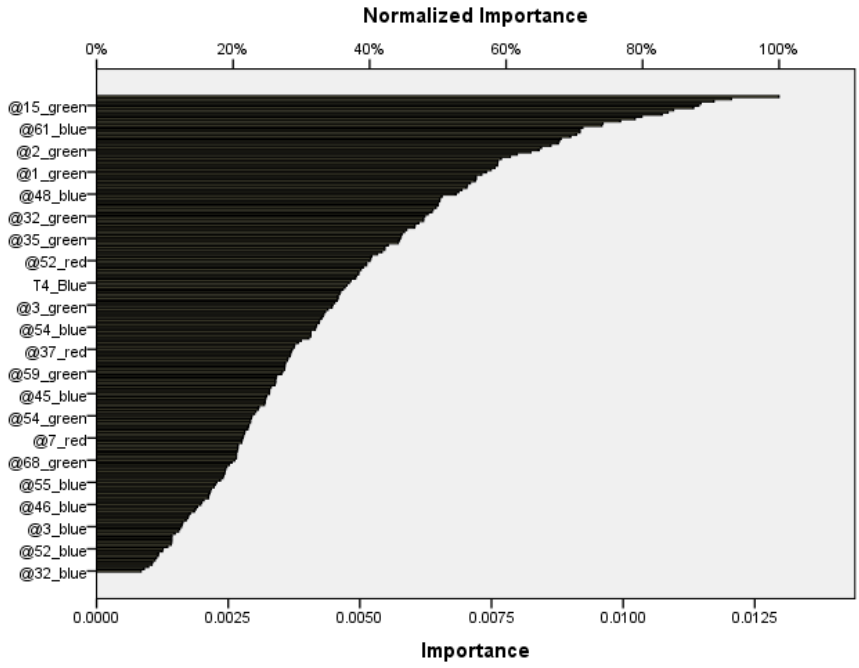


Figure 7.2: The Normalised Importance of the Contributing Features.

7.3. Comparing All Extracted Features

Table 7.7 shows a comparison of the results when using CGCM features, histogram features and joint features. Percentages shown in this table are classified by stepwise DA.
The results are compared according to precision, accuracy and AUC value.

Table 7.7: Comparison between the Extracted Features for Colour Images Classified by Stepwise DA.

Features	Number of Images	Images Types	Accuracy (Testing)	Precision	AUC
CGCM		Lossless Colour	76%	**87%**	**.999**
Histogram		Colour	82.2%	79%	.988
Joint		Colour	**91%**	86%	.997

As shown in table 7.7, the joint features achieved the highest accuracy with a percentage of 91%. The CGCM features achieved the highest precision with a percentage of 87% and an AUC value of .999.
However, the joint features also achieved a high precision score of 86%, which is very close to the percentage attained by the CGCM features.
The histogram features achieved high accuracy after the joint features and high AUC after the CGCM features.
All AUC values are close to 1.0, which indicates excellent discrimination and show outstanding detection performance.
In evaluation, it can be noted that joint features beat the other features by a considerable margin in terms of accuracy.

Table 7.8 shows a comparison of the results when using CGCM features, histogram features and joint features. Percentages shown in this table are classified by MLP.
The results are compared according to precision, accuracy and AUC value.

Table 7.8: Comparison between the Extracted Features for Colour Images Classified by MLP.

Features	Images Types	Accuracy	Precision	AUC
CGCM	Colour	95%	91%	.999
Histogram	Colour	85%	79%	.988
Joint	Colour	**98.4%**	**98.2%**	.997

The joint features achieved the highest accuracy and precision. The CGCM features achieved the highest accuracy and precision after the joint features.

The histogram features achieved high accuracy and acceptable precision.

It is clear that all AUC values are close to 1.0, which indicates excellent discrimination and show outstanding detection performance.

All in all, it can be noted that in both tables (7.7 and 7.8) the joint features achieved the top accuracy in terms of classifying the clean and stego images.

Chapter 8

Conclusion and Limitation of the Research

8.1. Summary

This thesis has introduced a novel detection system to distinguish between clean and stego images. The proposed detection system was designed to work as a blind steganalyser and was based on extracting CGCM and histogram features.

The proposed detection system was trained to classify grey or colour clean images and grey or colour stego images, which were created using LSB and the F5 steganography algorithm.

The results were classified by two different classifiers, stepwise DA and MLP, in order to compare their performances and improve the accuracy of detecting the stego images.

The work presented in this thesis was done based on revisiting a large number of previous existing detection methods introduced in the literature. The proposed detection system has combined several types of features and applied feature selection techniques for the analysis, which improves the detection accuracy and increases the classifires's sensitivity with respect to the differences caused by multi steganography artefacts.

The most significant features and achievements of the proposed detection system are summarised below:

- ***Novelity of the work:*** A novel image detection system for grey and colour stego images is introduced in this thesis. The detection system is working by extracting large number of CGCM features and histogramn features.

- ***Scope of the System***: The system has been tested on grey and colour images, on both lossless and lossy image formats, and on stego images created by the two steganography methods LSB and F5 algorithm. More than 3000 clean and stego images have been studied and analysed.

- ***Blind Steganalysis:*** Based on the analysis, effective features extraction techniques were developed to extract a set of sensitive and discriminating features. Using

stepwise DA and MLP as classifiers, a blind steganalysis system was constructed to detect the presence of hidden files in the lossless and lossy images.

- ***Extracting Different Features:*** The proposed detection system is based on extracting two types of images features: 33 CGCM and 210 histogram features. Each type of features was implemented and tested separately on the images dataset. The tests were evaluated and compared with each other.

- ***Joint Features:*** The selected CGCM features and histogram features are joined to improve the accuracy of the detection system. The joint features achieved higher accuracy than the CGCM and histogram features that reached 91% for the colour stego images classified by stepwise DA, as presented in section 7.5.
  In addition, the joint features achieved higher accuracy than the CGCM and histogram features that reached 98% for the colour stego images classified by MLP, as shown in section 7.5.

- ***Different Sizes of Hidden Files:*** Stego images having different sizes of hidden files were studied and analysed.

- ***Outperforms Previous Methods:*** Results proved that the developed system achieved better performance than five previous detection methods introduced in the literature. In addition, it outperforms their outcomes as shown in sections 5.8 and 6.9.

8.2. Limitations and Future Research Directions

Although the proposed detection system performs extremely well and has the ability to deal with different image formats and steganography methods, it still needs to be improved in some parts. The following points summarise them:

- The CGCM features perform very well in terms of differentiating between the clean and stego images. However, the extraction process from the tested images takes too much time, which results in slow progress when analysing large numbers of images.

For this reason, more tests were conducted by using the histogram features than by using CGCM in this work.

- The overall performance of the system was much higher in terms of classifying and analysing the lossless format rather than the lossy format. The cross-validated percents achieved in the training phase was 82% for classifying the lossless images and 65% for classifying the lossy images. In addition, the accuracy achieved during the testing phase was 76% for the losslee images and 57% for the lossy images as presented in section 5.4.3. The performance percentage for classifying the lossy images needs to be improve to achieve higher accuaracy.

- Although the proposed detection system is able to differentiate between clean and stego images, it cannot retrieve the hidden files; thus, additional techniques are needed to enhance the ability of the system.

- The performance of the system could be further improved by analysing stego images created by more steganography methods.

- Train and test the system using other classification methods such as: Support Vector Machine (SVM) and Deep Neural Network (DNN).

- The proposed detection system can be further developed to be used as a commercial tool. A userinterface is also needs to be created to make it easier for users to use.

List of Referencs

Akdeniz, Y., 1996. Computer pornography: A comparative study of the US and UK obscenity laws and child pornography laws in relation to the internet. *International Review of Law, Computers & Technology*, 10(2), pp.235-261.

Akdeniz, Y., 1997. Governance of pornography and child pornography on the global Internet: a multi-layered approach. *Law and the Internet: regulating Cyberspace*, pp.223-241.

Al-Ani, Z.K., Zaidan, A.A., Zaidan, B.B. and Alanazi, H. (2010) 'Overview: Main fundamentals for steganography.' *Journal of Computing* 2, (3) 158-165.

Anderson, R.J. and Petitcolas, F.A. (1998) 'On the limits of steganography'. *IEEE Journal on selected areas in communications* 16, (4) 474-481.

Ashok. J.,Raju. Y., Curran Kevin., and Srinivas, K. (2010) 'STEGANOGRAPHY: AN OVERVIEW.' *International Journal of Engineering Science and Technology* 2, (10) 5985-5992.

Avcibas, I., Memon, N. and Sankur, B. (2003) 'Steganalysis using image quality metrics.' *IEEE transactions on Image Processing* 12, (2) 221-229.

Backbone Security (2008a). 'Steganography Application Fingerprint Database', http://www.sarc-wv.com/docs/safdb.pdf, June 20.

Ballard, J.D., Hornik, J.G. and McKenzie, D. (2002) 'Technological Facilitation of Terrorism Definitional, Legal, and Policy Issues.' *American Behavioral Scientist* 45, (6) 989-1016.

C.D. Tofan (2010) 'Protection of Information Technology Critical Infrastructures in UE' *Simpoziului international al tinerilor cercetatori (Editia a VIII-a)*

Cai, K., Li, X., Zeng, T., Yang, B. and Lu, X. (2010) 'Reliable histogram features for detecting LSB matching.' *IEEE International Conference on Image Processing* 1761-1764.

Calders, T. and Jaroszewicz, S. (2007) 'Efficient AUC optimization for classification.' *In European Conference on Principles of Data Mining and Knowledge Discover*y, Springer Berlin Heidelberg, 42-53.

Calders, T. and Jaroszewicz, S., 2007, September. Efficient AUC optimization for classification. In *European Conference on Principles of Data Mining and Knowledge Discovery* (pp. 42-53). Springer Berlin Heidelberg.

Cawley, G.C. and Talbot, N.L. (2003) 'Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers.' *Pattern Recognition* 36 (11), 2585-2592.

Chandramouli, R. (2002) 'Mathematical approach to steganalysis.' *Electronic Imaging International Society for Optics and Photonics* 14-25.

Cheddad, A., (2009) Digital Image Steganography: Concept, Algorithm, and Application. VDM.

Cheddad, A., Condell, J., Curran, K. and Mc Kevitt, P. (2010) 'Digital image steganography: Survey and analysis of current methods.' *Signal processing* 90, (3) 727-752.

Chen, M., Zhang, R., Niu, X. and Yang, Y. (2006) 'Analysis of current steganography tools: classifications & features.' *In 2006 International Conference on Intelligent Information Hiding and Multimedia,* 384-387.

Chen, X., Wang, Y., Tan, T. and Guo, L. (2006) 'August. Blind image steganalysis based on statistical analysis of empirical matrix.' *In IEEE 18th International Conference on Pattern Recognition (ICPR'06)* 3, 1107-1110.

Chhikara, R.R. and Singh, L. (2015) 'February. An improved discrete firefly and t-test based algorithm for blind image steganalysis.' *In 2015 IEEE 6th International Conference on Intelligent Systems, Modelling and Simulation* 58-63.

Climatica (2016) *Uncertainty, Precision and Accuracy* [online] available from http://climatica.org.uk/climate-science-information/uncertainty> [05 May 2016].

Coakes, S.J. and Steed, L.(2009) *SPSS: Analysis without anguish using SPSS version 14.0 for Windows*. John Wiley & Sons, Inc.

Ćosić, J. and Bača, M., (2010) 'Steganography and steganalysis-does local web sites contain "Stego" contents*?.' In Proceedings ELMAR-2010*.

Crandall, R., (1998) *Some notes on steganography*. Posted on steganography mailing list.

Currie III, D. L., & Irvine, C. E. (1996). Surmounting the effects of lossy compression on Steganography. Naval Postgraduate School Monterey Ca Dept Of Computer Science

Das, S., Das, S., Bandyopadhyay, B. and Sanyal, S. (2011) 'Steganography and Steganalysis: different approaches.' *International Journal of Computers, Information Technology and Engineering (IJCITAE)*, 2, (1)

Davidson, J.L. and Jalan, J., (2010) 'January. Canvass-A Steganalysis forensic tool for JPEG images.' *In Proceedings of the Conference on Digital Forensics, Security and Law* (p. 99). Association of Digital Forensics, Security and Law.

Dayhoff, J. E. (1990). *Neural network architectures: an introduction*. Van Nostrand Reinhold Co.

Deakin, E.B., (1972). 'A discriminant analysis of predictors of business failure.' *Journal of accounting research*, 167-179.

Deng, Q.L. and Lin, J.J. (2009) 'A Universal Steganalysis Using Features Derived from the Differential Image Histogram in Frequency Domain.' *2nd International Congress on Image and Signal Processing, CISP'09*, 1-4. IEEE.

Dinca, L. (2011) 'Survey of the Use of Steganography over the Internet'. *Informatica Economica* 15 (2), 153

Dong, J. and Tan, T. (2008) 'Blind image steganalysis based on run-length histogram analysis.' *In ICIP*, 2064-2067.

Dunbar, B. (2002). A detailed look at Steganographic Techniques and their use in an Open-Systems Environment. *Sans Institute*, 1-9.

Dunteman, G.H. (1984) *Introduction to multivariate analysis*. Sage Publications, Inc.

Duric.Z, Jacobs. M andJajodia, S (2004) 'Information Hiding: Steganography and Steganalysis'. *Preprint Submitted to Elsevier Science*

East-tec (2013) *InvisibleSecret*s [online] available from <http://www.east-tec.com/invisiblesecrets/>[01 Jun 2015].

EL-Emam, N. N. (2007). 'Hiding a large amount of data with high security using steganography algorithm'. Journal of Computer Science, 3(4), 223.

Elgabar, E.E.A. and Mohammed, F.A., 2013. JPEG versus GIF Images in forms of LSB Steganography. International Journal of Computer Science and Network (IJCSN), ISSN, pp.2277-5420.

Eloff, J.H.P. T, Mrkel. and MS, Olivier (2005) 'An overview of image steganography.' *In Proceedings of the fifth annual Information Security South Africa Conference*.

Farid, H. (2001) *Detecting steganographic messages in digital images* (Vol. 2, p. 12). Technical Report TR2001-412, Deparment of Computer Science, Dartmouth College.

Fawcett, T., (2004) ROC graphs: Notes and practical considerations for researchers. *Machine learning*, 31 (1), 1-38.

Filippas. J, Amin. S, Naguib. R, & Bennett. M, (2003) 'A Parallel Implementation Of A Genetic Algorithm For Colonic Tissue Image Classification.' *Proc of the 4th Annual IEEE Conf on Information Technology Applications in Biomedicine*

Fridrich, J. (2009) *Steganography in digital media: principles, algorithms, and applications*. New York: Cambridge University Press.

Fridrich, J. and Goljan, M., (2002) 'Practical steganalysis of digital images: state of the art.' *In Electronic Imaging 2002.* International Society for Optics and Photonics. 1-13.

Fridrich, J. and Long, M. (2000) 'Steganalysis of LSB encoding in color images.' *IEEE International Conference on Multimedia and Expo, ICME 2000.* vol.3, 1279-1282. IEEE.

Fridrich, J., 2009. Steganography in digital media: principles, algorithms, and applications. Cambridge University Press.

Fridrich, J., Goljan, M. and Hogea, D. (2002) 'Steganalysis of JPEG images: Breaking the F5 algorithm.' *In International Workshop on Information Hiding*, Springer Berlin Heidelberg. 310-323.

Fridrich, J., Goljan, M., and Du, R. (2001) 'Reliable Detection of LSB Steganography in Colour and Grayscale Images'. *Proceedings of the 2001 Workshop on Multimedia and Security*: *New Challenges*. 27-30. ACM.

Fridrich, Jessica, Miroslav Goljan, and Dorin Hogea. (2003) 'Steganalysis of JPEG images: Breaking the F5 algorithm.' *Information Hiding*. Springer Berlin

Gao, Y., Li, X., Yang, B. and Lu, Y. (2009) 'Detecting LSB matching by characterizing the amplitude of histogram'. *IEEE International Conference on Acoustics, Speech and Signal Processing*,1505-1508

Geetha, S., Sindhu, S.S.S. and Kamaraj, N. (2009) 'Blind image steganalysis based on content independent statistical measures maximizing the specificity and sensitivity of the system.' *computers & security*, 28 (7), 683-697.

Ghanbari. S, Keshtegarym. M and Ghanbari, N. (2012) 'New Steganalysis Method using Clcm and Neural Network'. International Journal of Computer Applications 42(7).

Gong, R. and Wang, H. (2012) 'Steganalysis for GIF images based on colors-gradient co-occurrence matrix.' *Optics Communications* 285 (24), 4961-4965.

Gonzalez, R. C., Woods, R. E., & Eddins, S. L. (2004). Digital image processing using MATLAB. Upper Saddle River, N. J: Pearson Prentice Hall.

Hanley, J.A. and McNeil, B.J. (1982) 'The meaning and use of the area under a receiver operating characteristic (ROC) curve'. *Radiology* 143 (1), 29-36.

Harb, A.M.J.A.D. and Jayousi, R.A.S.H.I.D. (2012) 'Comparing neural network algorithm performance using SPSS and Neurosolutions.' *In proceedings of the International Arab Conference on Information Technology (ACIT'2012).*

Harmsen, J.J. and Pearlman, W.A. (2003) 'Steganalysis of additive-noise modelable information hiding.' *In Electronic Imaging 2003*. International Society for Optics and Photonics. 131-142.

Hayati, P., Potdar, V. and Chang, E. (2007) 'A survey of steganographic and steganalytic tools for the digital forensic investigator.' *In Workshop of Information Hiding and Digital Watermarking.*

Haykin, S. (1998) *Neural Networks: A Comprehensive Foundation*, 2nd ed. New York: Macmillan College Publishing.

Hmood. A, Zaidan. B, and et al. (2010) 'An Overview on Hiding Information Technique in Images'. Journal of Applied Sciences 10 (18)

Holla. K (2014) *Steganography* [online] avialble from <http://www.slideshare.net/ksholla/steganography-36747999>[09 January 2017]

Holub, V. and Fridrich, J. (2013) 'Random projections of residuals for digital image steganalysis.' *IEEE Transactions on Information Forensics and Security* 8 (12), 1996-2006.

Holub, V. and Fridrich, J. (2015) 'Low-complexity features for JPEG steganalysis using undecimated DCT.' *IEEE Transactions on Information Forensics and Security* 10 (2), 219-228.

Honeyman, N.P.P. (2002) 'Detecting Steganographic Content on the Internet.' *In Proc 2002 Network and Distributed System Security Symposium*. Internet Society.

Huayong, G. Mingshenga, H and Qiana, W. (2011) 'Steganography and Steganalysis Based on Digital Image.' *4th International Congress on Image and Signal Processing*, 252-255

Johnson, N. F., and Jajodia, S. (1998) 'Exploring steganography: Seeing the unseen.' *Computer* 31 (2), 26-34.

Johnson, N. F., and Jajodia, S. (1998) 'Steganalysis of images created using current steganography software.' *In Information Hiding*. Springer Berlin Heidelberg, Lecture Notes in Computer Science, vol. 15252, 73-289.

Johnson, N. F., and Katzenbeisser, S. (2000) 'A survey of steganographic techniques.' *In Information hiding*. Norwood, MA: Artech House. 43-78.

Johnson. N, and Jajodia. S. (1998) 'Steganalysis: The Investigation of Hidden Information.' *IEEE Information Technology Conference*, 113-116.

Kellen, Tom. (2001) 'Hiding in plain view: Could steganography be a terrorist tool'. *SANS Institute InfoSec Reading Room*

Kerbaj, R. and Kennedy, D., (2008) *Link between Child Porn and Muslim Terrorists Discovered in Police Raids* [online]. The Times, 17, available from <http://www.thetimes.co.uk/tto/news/uk/crime/article1875115.ece> [05 March 2016].

Kessler, G. C. (2004). 'An overview of steganography for the computer forensics examiner.' *Forensic Science Communication*s 6 (3), 1-27.

Kharrazi, M., Sencar, H.T. and Memon, N. (2005). 'Benchmarking steganographic and steganalysis techniques. *In Electronic Imaging 2005*, International Society for Optics and Photonics, 252-263.

Kipper, G. (2004) *Investigator's Guide to Steganography*. London: Boca Raton

Klecka, William R. (1980) *Discriminant Analysis*. California: Sage Publications, Incorporated.

Kurtuldu, O. and Arica, N. (2008) 'A new steganography method using image layers.' *23rd International Symposium on Computer and Information Sciences*. ISCIS'08, 1-4. IEEE.

Latham, A (1999) JJTC Johnson and Johnson Technology Consultant [Online] available from http://linux01.gwdg.de/~alatham/stego.html [06 Aug 2014].

Legislation (1978) *Protection of Children Act 1978* [online] available from <http://www.legislation.gov.uk/ukpga/1978/37 [15 December 2016].

Legislation (2004) *Children Act 2004* [online] available from <http://www.legislation.gov.uk/ukpga/2004/31/contents> [15 December 2016].

Legislation (2008) *Criminal Justice and Immigrations 2008* [online] available from 9 <http://www.legislation.gov.uk/ukpga/2008/4/contentss> [19 December 2016].

Li, B., He, J., Huang, J. and Shi, Y.Q. (2011) 'A survey on image steganography and steganalysis.' *Journal of Information Hiding and Multimedia Signal Processing* 2 (2), 142-172.

Lie, W.N. and Lin, G.S. (2005) 'A feature-based classification technique for blind image steganalysis.' *IEEE Transactions on multimedia* 7 (6), 1007-1020.

Liu, Q., Sung, A.H., Ribeiro, B., Wei, M., Chen, Z. and Xu, J., (2008) 'Image complexity and feature mining for steganalysis of least significant bit matching steganography.' *Information Sciences* 178 (1), 21-36.

Lou, D.C. and Hu, C.H., (2012) 'LSB steganographic method based on reversible histogram transformation function for resisting statistical steganalysis.' *Information Sciences* 188, 346-358.

Lu, J.C., Liu, F.L. and Luo, X.Y., (2014) 'Selection of image features for steganalysis based on the Fisher criterion.' *Digital Investigation* 11(1), 57-66.

Luo, X. Y., Wang, D. S., Wang, P., and Liu, F. L. (2008) 'A review on blind detection for image steganography.' *Signal Processing* 88 (9), 2138-2157.

Lyu, S. and Farid, H., (2006) 'Steganalysis using higher-order image statistics.' *IEEE transactions on Information Forensics and Security* 1 (1), 111-119.

Maitra, I.K., (2011) 'Digital steganalysis: Review on recent approaches.' *Journal of Global Research in Computer Science 2* (1).

MathWorks (2001) *Image Processing Toolbox* [online] available from<http://www.mathworks.com/products/image/description2.html?s_cid=ip_import_a>[1 August 2015]

MathWorks (2012) MATLAB the Language of Technical Computing [online] available from < http://uk.mathworks.com/products/matlab/> [02-10-2014]

McLachlan, Geoffrey (2004) *Discriminant analysis and statistical pattern recognition*. Vol. 544. John Wiley & Sons.

Miano, J. (1999). *Compressed image file formats: Jpeg, png, gif, xbm, bmp*. Addison-Wesley Professional.

Ming. C, Ru. Z, and et al. (2006) 'Analysis of Current Steganography Tools: Classifications & Features'. I*nternational Conference on Intelligent Information Hiding and Multimedia Signal Processing*

Morkel. T, Eloff. J, and et al. (2005) 'An Overview of Image Steganography'.*The Fifth Annual Information Security South Africa Conference*

Morris, T.(n.d.) *Image Processing with MatLab* [lecture] module COMP20072, Manchester: Manchester University

Natanj, S. and Taghizadeh, S.R., (2011) 'Current Steganography Approaches: A survey.' *International Journal of Advanced Research in Computer Science and Software Engineering*, 1.

Nissar, A and Mir, A. (2010) 'Classification of Steganalysis Techniques: A Study.' *Digital Signal Processing*, 20.

Norusis, Marija. (2008) *SPSS 16.0 statistical procedures companion*. Prentice Hall Press, 2008.
Nuruzzaman, M. (2005) *Digital Image Fundamentals in MATLAB*. AuthorHouse.

Omer, I. and Werman, M. (2004) 'Color lines: Image specific color representation.' *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* 2, II-946

OpenStego (2010) Overview [online] available from <http://openstego.sourceforge.net/index.html> [30-08-2012].

OutGuess (2003) *Steganography Detection with Stegdetect* [Online] available from <http://www.outguess.org/detection.php>[29 December 2013].

Owens, M., (2002) 'A discussion of covert channels and stegangraphy.' *SANS institute* 1, 1-18.

Pallant, J. (2010). SPSS survival manual: A step by step guide to data analysis using SPSS. McGraw-Hill International.

Pevný, T. and Fridrich, J. (2005) 'Towards multi-class blind steganalyzer for JPEG images.' *In International Workshop on Digital Watermarking*, Springer Berlin Heidelberg, 39-53.

Pevný, T. and Fridrich, J., (2006) 'Multi-class blind steganalysis for JPEG images.' *Proceedings of the SPIE on Security and Watermarking of Multimedia Contents VIII*, 6072 (1), 257-269.

Popa, R., 1998. An analysis of steganography techniques. Master's thesis, The Polytechnic University of Timisoara, Timisoara, Romênia.

Provos, N. (2008) *Steganography Detection with Stegdetect* [online] available from <http://www.outguess.org/detection.php> [20 June 2014].

Provos, N. and Honeyman, P., (2003) 'Hide and seek: An introduction to steganography.' *IEEE Security & Privacy* 1 (3), 32-44.

Provos, Niels, and Peter Honeyman. (2001) 'Detecting steganographic content on the internet.' *Center for Information Technology Integration*

Raja, K.B., Chowdary, C.R., Venugopal, K.R. and Patnaik, L.M. (2005) 'A secure image steganography using LSB, DCT and compression techniques on raw images.' *In 2005 3rd International Conference on Intelligent Sensing and Information Processing*, 170-176. IEEE.

Rapid Deployment Software (2001) Euphoria [Online] available from http://www.rapideuphoria.com/ [06 Aug 2014].

Raúl Ramos-Pollán and Naimy González de Posada. (2011) 'Optimizing the Area Under the ROC Curve in Multilayer Perceptron-based Classifiers'. FUTURE COMPUTING: The Third International Conference on Future Computational Technologies and Application.

Reddy, V. L., Subramanyam, A., & Reddy, P. C. (2011). Implementation of LSB steganography and its evaluation for various file formats. Int. J. Advanced Networking and Applications 2 (05), 868-872.

Sarc-tech (2004) *StegAlyzerAS* [online] available from <http://www.sarc-wv.com/products/stegalyzeras/> [10 Aug 2015]

Schaathun, H.G., (2012) Steganography and Steganalysis. *Machine Learning in Image Steganalysis*, Wiley Online Library, 7-24.

Schmidt, M. B., Bekkering, E., and Warkentin, M. (2004). ,On the Illicit Use of Steganography and Its Detection'. ISOneWorld International Conference. April 14-16. Las Vegas, NV.

ScikitLearn (2010) *Neural network models (supervised)* [online] available from <http://scikit-learn.org/dev/modules/neural_networks_supervised.html> [15 June 2015]

Shi, Y.Q., Xuan, G., Zou, D., Gao, J., Yang, C., Zhang, Z., Chai, P., Chen, W. and Chen, C. (2005) 'Image steganalysis based on moments of characteristic functions using wavelet decomposition, prediction-error image, and neural network.' *In ICME* (Vol. 2005, pp. 269-272).

Singla. P (2014) *RSA Cryptography and Image Steganography* [online] available from < http://www.slideshare.net/pd2791/rsa-cryptography-steganography> [09 January 2017].

SPYCHECKER (2006) *S-Tools* [online] available from < http://www.spychecker.com/program/stools.html> [27 September 2012]

Stahl, B., Elizondo, D., Carroll-Mayer, M., Zheng, Y. and Wakunuma, K. (2010). Ethical and legal issues of the use of computational intelligence techniques in computer security and computer forensics. *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 1-8.

Steganalysis, H.C.D.B. and Westfeld, A. (2001) ‘F5—A Steganographic Algorithm.’ *In Information Hiding: 4th International Workshop*, IH. Springer Science & Business Media, (Vol.2137, p. 289).

Sun, Z., Hui, M. and Guan, C. (2008) ‘Steganalysis based on co-occurrence matrix of differential image.’ *IIHMSP'08 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 1097-1100. IEEE.

Suryawanshi, G.R. and Mali, S.N. (2015) ‘Study of Effect of DCT Domain Steganography Techniques in Spatial Domain for JPEG Images Steganalysis.’ *International Journal of Computer Applications* 127 (6), 16-20.

Symblogogy (2007) Hiding Messages in Plain Sight [online] available from <http://symblogogy.blogspot.co.uk/2007_02_01_archive.html> [21 May 2013].

Tan, S. and Li, B. (2012) ‘Targeted steganalysis of edge adaptive image steganography based on LSB matching revisited using B-spline fitting.’ *IEEE Signal Processing Letters* 19 (6), 336-339.

Thiyagarajan, P., Aghila, G. and Venkatesan, V.P. (2011) ‘Stego-Image Generator (SIG)-Building Steganography Image Database.’ *In Advances in Digital Image Processing and Information Technology*, Springer Berlin Heidelberg, 257-267.

Tiwari, N. and Shandilya, D.M. (2010) ‘Evaluation of Various LSB based Methods of Image Steganography on GIF File Format.’ *International Journal of Computer Applications* (0975–8887) Volume.

Umamaheswari. M, Sivasubramanian. S and Pandiarajan, S. (2010) ‘Analysis of Different Steganographic Algorithms for Secured Data Hiding’. *LJCSNS International Journal of Computer Science and Network Security* 10 (8).

Umbaugh, S. (1998) *Computer Vision and Image Processing: a practical approach using CVIP tools.* UK: Prentice Hall PTR

V. Arvis, C. Debain et al. (2004) ‘Generalization of the concurrencematrixforcolour images: Applicationto Colour Texture Classification.’ *Image Anal Stereo.*

V. Kecman (2001) *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. Cambridge, MA: MIT Press.

Walsh College (2009) *Detect* [Online] available from <http://bit599.netai.net/detect_stego.htm> [31 Jul 2013]

Walsh College (2009) *Steganography Tools* [online] available from < http://bit599.netai.net/s_tools.htm > [26 September 2012]

Wang, H., and Wang, S. (2004) 'Cyber warfare: steganography vs. steganalysis.' *Communications of the ACM* 47 (10), 76-82.

Warkentin, M., Bekkering, E. and Schmidt, M.B. (2008) 'Steganography: Forensic, Security, and Legal Issues.' *The Journal of Digital Forensics, Security and Law: JDFSL* 3 (2), 17-34.

Wayner. P (2002) *Disappearing Cryptography: Information Hiding: Steganography & Watermarking'*. San Francisco: 2nd. ed. Morgan Kaufmann.

Westfield, A. (2001) 'F5—A Steganographic Algorithm High Capacity Despite Better Steganalysis'.*4th International Workshop on Information Hiding*.

Wetstone Technologies (2008) *Stego Suite* [online] available from < https://www.wetstonetech.com/product/stego-suite/ > [25 June 2014].
Willamette University (T.D) *Image File Format* [online] available from <http://www.willamette.edu/~gorr/classes/GeneralGraphics/imageFormats/> [06 Feburary 2016].

Xia, Z., Yang, L., Sun, X., Liang, W., Sun, D. and Ruan, Z. (2011) 'A learning-based steganalytic method against LSB matching steganography.' *Radioengineering* 20 (1), 102-109.

Xuan, G., Cui, X., Shi, Y.Q., Chen, W., Tong, X. and Huang, C. (2007) 'Jpeg steganalysis based on classwise non-principal components analysis and multi-directional markov model.' *In 2007 IEEE International Conference on Multimedia and Expo*, 903-906.

Yen-Jen. O, Shien-Ching, and et al. (2005) 'Data Classification With Radial Basis Function Networks Based on a Novel Kernel Density Estimation Algorithm'. *IEEE Transaction On Neural Networks* 16 (1).

Yu, W., Li, Z. and Ping, L. (2010) 'Blind detection for JPEG steganography.' *In 2010 International Conference on Networking and Information Technology*, 128-132. IEEE.

Zhang, Jun., Cox, Ingemar. J., & Doërr, G. (2007) 'Steganalysis for LSB matching in images with high-frequency noise. *IEEE 9th Workshop on Multimedia Signal Processing. MMSP 2007*, 385-388.

Zhao, H., Wang, H. and Khan, M.K. (2011) 'Steganalysis for palette-based images using generalized difference image and color correlogram.' *Signal Processing* 91 (11), 2595-2605.

Zhou, C.E., Feng, J.C. and Yang, Y.X. (2009) 'Blind steganalysis based on features in fractional Fourier transform domain.' *International Conference on Communications, Circuits and Systems ICCCAS*, 301-303. IEEE.

Zhou, L., Burgoon, J.K., Twitchell, D.P., Qin, T. and Nunamaker Jr, J.F. (2004) 'A comparison of classification methods for predicting deception in computer-mediated communication.' *Journal of Management Information Systems* 20 (4), 139-166.

Zielińska, E., Mazurczyk, W., and Szczypiorski, K. (2014) 'Trends in Steganography'. *Communications of the ACM* 57 (3), 86-95.

Appendix

5.1. The Structure Matrix Table for Test 1 (BMP Images)

Structure Matrix

	Function
	1
T10_red	.333
T4_red	.332
T5_red	-.304-
T4_green[a]	.240
T7_green	.234
T2_red[a]	-.218-
T1_blue[a]	.205
T3_green[a]	.204
T5_green	.188
T4_blue	.182
T11_red	.158
T11_blue	.143
T3_blue[a]	.121
T2_green[a]	-.119-
T1_green[a]	.115
T11_green	.108
T5_blue[a]	.083
T6_green[a]	.069
T10_blue[a]	.065
T10_green[a]	.059
T1_red[a]	.054
T6_red[a]	.047
T9_blue[a]	-.039-
T9_red	-.039-
T9_green[a]	-.038-
T8_blue[a]	-.037-
T7_blue[a]	-.031-
T2_blue[a]	-.026-
T8_red[a]	-.018-
T3_red	-.013-
T8_green[a]	-.013-
T6_blue[a]	-.011-
T7_red	.001

Pooled within-groups correlations between discriminating variables and standardized canonical discriminant functions
Variables ordered by absolute size of correlation within function.
a. This variable not used in the analysis.

5.2. Variables Used in Test 1 (BMP Images).

Variables in the Analysis

Step		Tolerance	F to Remove	Wilks' Lambda
1	T10_red	1.000	68.984	
2	T10_red	.962	76.593	.775
	T5_red	.962	64.837	.742
3	T10_red	.862	101.997	.730
	T5_red	.958	60.594	.629
	T11_red	.896	31.632	.558
4	T10_red	.648	127.234	.729
	T5_red	.946	44.337	.542
	T11_red	.803	44.228	.541
	T3_red	.708	17.301	.480
5	T10_red	.648	99.928	.574
	T5_red	.929	26.166	.430
	T11_red	.775	22.349	.423
	T3_red	.532	40.741	.459
	T4_red	.653	31.863	.441
6	T10_red	.613	114.910	.484
	T5_red	.924	24.769	.342
	T11_red	.692	40.831	.368
	T3_red	.478	62.875	.402
	T4_red	.505	67.174	.409
	T9_red	.656	48.085	.379
7	T10_red	.612	108.360	.444
	T5_red	.924	22.314	.317
	T11_red	.691	40.016	.343
	T3_red	.474	64.352	.379
	T4_red	.504	64.871	.380
	T9_red	.554	62.921	.377

	T4_blue	.800	13.135	.303
	T10_red	.604	88.787	.400
	T5_red	.922	19.670	.302
	T11_red	.690	39.334	.330
8	T3_red	.419	35.658	.324
	T4_red	.385	24.492	.308
	T9_red	.005	11.616	.290
	T4_blue	.785	9.659	.287
	T7_red	.005	7.434	.284
	T10_red	.601	77.793	.372
	T5_red	.919	20.188	.292
	T11_red	.689	35.509	.313
	T3_red	.419	34.812	.312
9	T4_red	.380	19.879	.292
	T9_red	.005	13.282	.282
	T4_blue	.776	7.403	.274
	T7_red	.005	9.001	.276
	T7_green	.926	6.773	.273
	T10_red	.598	78.661	.361
	T5_red	.917	18.058	.280
	T11_red	.685	36.444	.304
	T3_red	.417	35.652	.303
10	T4_red	.379	20.381	.283
	T9_red	.005	13.901	.274
	T4_blue	.776	7.006	.265
	T7_red	.005	9.512	.268
	T7_green	.926	6.608	.264
	T11_green	.979	6.495	.264
	T10_red	.594	79.974	.352
	T5_red	.909	15.225	.267
	T11_red	.676	38.730	.298
	T3_red	.417	33.533	.291
	T4_red	.378	20.552	.274
11	T9_red	.005	13.316	.265
	T4_blue	.774	7.330	.257
	T7_red	.005	8.994	.259
	T7_green	.923	5.576	.255
	T11_green	.968	7.563	.257
	T11_blue	.949	6.048	.255
	T10_red	.590	71.922	.335
12	T5_red	.909	14.326	.260
	T11_red	.670	40.263	.294
	T3_red	.416	33.737	.286

T4_red	.378	20.460	.268
T9_red	.005	12.026	.257
T4_blue	.767	8.209	.253
T7_red	.005	7.721	.252
T7_green	.921	5.051	.248
T11_green	.931	5.027	.248
T11_blue	.934	7.126	.251
T5_green	.849	4.106	.247

Extracting Features form the CGCM Algorithms

CGC Code 1

```
function TT=cgc(C)
%C is the color matrix, read the indexed image, read image from graphics file
[M,N]= size(C);
Nc = 256;
C = double(C);  %Convert C to double (real numbers), to do numerical manipulations on it
f=zeros(M,N);   %Zeros to create the matrix of size M*N filled with zeros
for i=1:M-1  % The for loop is used to replace all the zeros by the values calculated by the formula (equation)
   for j=1:N-1
      f(i,j)= sqrt((C(i+1,j)- C(i,j))^2 +(C(i,j+1)-C(i,j))^2); %f is the gradient matrix, equation number (1)
   end
end

Ng= 256; %Maximum value of the elments in C

fmax = max(f);
fmax = max(fmax);

%Colors-gradient co-occurence matrix

G=zeros(M,N); %G is the scaled matrix f, G is the Colors-gradient co-occurence matrix (CGCM)
for i=1:M-1
   for j=1:N-1
      G(i,j)= floor(f(i,j)*Ng/fmax+0.5); %Equation number (2), G is the CGCM matrix
   end
end;
tic;
% Equation (3)
H=zeros(Nc,Ng); % H is a matrix
c=0;
for k=1:Nc
   for l=1:Ng
      H(k,l)=hsuma(k,l,C,G);
   end
end
toc;

Ch=zeros(M-1,N-1);  %Ch Horizontal direction

% Equation (4)
```

```
for i=1:M-1
   for j=1:N-1
      Ch(i,j)=abs(C(i,j)-C(i,j+1));  %Ch Horizontal direction, equation (4)
   end
end

% Equation (11)
Hp=1/(Nc*Ng).*H;

den=sum(Hp(:));  %Denominator common to equations 12,13,14,15

%Small gradient dominance, equation (12)

T1=0;  %Initial value of T1 is zero
for j=1:Nc
   for i=1:Ng
      T1=T1+Hp(i,j)/(1+j)^2;
   end
end
T1=T1/den;
fprintf('Small gradient dominance: %0.5g.\n',T1);

%Equation (13), Big gradient dominance
T2=0;
for j=1:Nc
   for i=1:Ng
      T2=T2+Hp(i,j)*(j)^2;
   end
end
T2=T2/den;
fprintf('Big gradient dominance: %0.5g.\n',T2);

%Equation (14), Colors asymmetry
T3=0;
for i=1:Nc
   T3a=0;
   for j=1:Ng
      T3a=T3a+Hp(i,j);
   end
   T3=T3+T3a^2;
end
T3=T3/den;
fprintf('Colors asymmetry: %0.5g.\n',T3);

%Equation (15), Gradient asymmetry
T4=0;
for j=1:Nc
   T4a=0;
   for i=1:Ng
      T4a=T4a+Hp(i,j);
   end
   T4=T4+T4a^2;
end
T4=T4/den;
fprintf('Gradient asymmetry: %0.5g.\n',T4);


%16 Energy
T5=0;
for i=1:Nc
```

```
    for j=1:Ng
        T5=T5+Hp(i,j)^2;
    end
end
fprintf('Energy: %0.5g.\n',T5);

%17 Colors mean

T6=0;
for i=1:Nc
    T6a=0;
    for j=1:Ng
        T6a=T6a+Hp(i,j);
    end
    T6=T6+i*T6a^2;
end
fprintf('Colors mean: %0.5g.\n',T6);


%18 Gradient mean

T7=0;
for j=1:Nc
    T7a=0;
    for i=1:Ng
        T7a=T7a+Hp(i,j);
    end
    T7=T7+j*T7a^2;
end
fprintf('Gradient mean: %0.5g.\n',T7);

%19 Colors variance
T8=0;
for i=1:Nc
    T8a=0;
    for j=1:Ng
        T8a=T8a+Hp(i,j);
    end
    T8=T8+(i-T6)^2*T8a^2;
end
fprintf('Colors variance: %0.5g.\n',T8);

T8=sqrt(T8);

%20 Gradient variance
T9=0;
for j=1:Nc
    T9a=0;
    for i=1:Ng
        T9a=T9a+Hp(i,j);
    end
    T9=T9+(j-T7)^2*T9a^2;
end

T9=sqrt(T9);
fprintf('Gradient variance: %0.5g.\n',T9);

%21 Correlation
T10=0;
for i=1:Nc
```

```
    for j=1:Ng
        T10=T10+(i-T6)*(j-T7)*Hp(i,j);
    end
end
fprintf('Correlation: %0.5g.\n',T10);

%22 Colors entropy
T11=0;
p1= log2(sum(Hp(:))+1);
for i=1:Nc
    for j=1:Ng
        T11=T11+Hp(i,j);
    end
end
T11=-p1*T11;
fprintf('Colors entropy: %0.5g.\n',T11);

%23 Gradient entropy
T12=0;
for j=1:Nc
    for i=1:Ng
        T12=T12+Hp(i,j);
    end
end
T12=-p1*T12;
fprintf('Gradient entropy: %0.5g.\n',T12);

%24 Entropy
Hprod=Hp.*log2(Hp+1);

T13=sum(Hprod(:));
fprintf('Entropy: %0.5g.\n',T13);

%25 Inertia
T14=0;
for i=1:Nc
    for j=1:Ng
        T14=T14+(i-j)^2*Hp(i,j);
    end
end
fprintf('Inertia: %0.5g.\n',T14);

%26 Inverse difference moment
T15=0;
for i=1:Nc
    for j=1:Ng
        T15=T15+Hp(i,j)/(1+(i-j)^2);
    end
end
fprintf('Inverse difference moment: %0.5g.\n',T15);

TT=[T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15];


%figure(2);
%CC=(C);
%imshow(CC,map);
```

CGC Code 2

```
clear all;
clc;
pwd % Current directory
%cd 'C:\Users\Hp\Desktop\Colour_Images' %Very important!!
colors={'red_';'green_';'blue_'};
files = [dir('*.png') ; dir('*.jpg'),dir('*.jpeg');dir('*.bmp');dir('*.tif');dir('*.tiff')]; % list image files in the current
directory
all_files = [files]; % list of image files in the current directory

if size(all_files,1) == 0  % if the length of the list of files is zero then no image files found
   disp(strcat('No image files found in the directory :',32,pwd)); % print the current directory on the console
window
   f1 = errordlg(strcat('No image files found in the directory :',32,pwd), 'Error Dialog'); % print the current
directory on a dialog box
else
%
 filecount=0; %initialise counting the number of image files to be processed
 result_table(1,:)= {'ZFile', 'T1', 'T2','T3','T4','T5','T6','T7','T8','T9','T10','T11','T12','T13','T14','T15'}; % create a
header for the reult table
 for n_file=1:size(all_files,1)     %for each file in the file list do the cgc loop
 filename=all_files(n_file).name;
 [C, map] = imread(filename);
  ncolors = size(C,3); % number of color channels (=1 for indexed images and mono, = 3 for RGB images)
  for k=1:ncolors
    info = imfinfo(filename); % image info
%   if strcmp(info.ColorType,  'indexed')   %if image  is indexed do the stat analysis
     s = cgc(C(:,:,k));  % call cgc function to calculate the image stat
     filecount = filecount+1; % increment the file count
     if ncolors > 1
      ff=strcat(colors(k),filename);
      result_table(filecount+1,:)={ff{1},s(1),
s(2),s(3),s(4),s(5),s(6),s(7),s(8),s(9),s(10),s(11),s(12),s(13),s(14),s(15)}; % put results in the table
     else
     result_table(filecount+1,:)={filename,s(1),
s(2),s(3),s(4),s(5),s(6),s(7),s(8),s(9),s(10),s(11),s(12),s(13),s(14),s(15)}; % put results in the table
     end
 %  else
 %      disp(strcat('Image is not an indexed format',32,filename)); % if
 %      image is not indexed skip
  end
 end
 end % end for loop of each file
 clearvars -except ans result_table; % tidy up the work space. keep the result table and the working directory
variables.
 sorted_table=sortrows(result_table,1);

 KEY={'T1 = Small gradient dominance.',
'T2 = Big gradient dominance.',
'T3 = Colors asymmetry.',
'T4 = Gradient asymmetry.',
'T5 = Energy.',
'T6 = Colors mean.',
'T7 = Gradient mean.',
'T8 = Colors variance.',
'T9 = Gradient variance.',
'T10 = Correlation.',
```

```
'T11 = Colors entropy.',
'T12 = Gradient entropy.',
'T13 = Entropy.',
'T14 = Inertia.',
'T15 = Inverse difference moment.'}
```

CGC Code 3

```
function ret=hsuma(x,y,C,G,M,N)

ret=sum(sum(C==x & G==y));
```

Extracting Histogram Features Algorithms

Hisotgram Code 1

```
clear all;
clc;
pwd % Current directory
%cd 'C:\Users\Hp\Desktop\Colour_Images' %Very important!!
%
files = [dir('*.jpg'),dir('*.jpeg');dir('*.bmp')]; % list image files in the current directory
all_files = [files]; % list of image files in the current directory

if size(all_files,1) == 0  % if the length of the list of files is zero then no image files found
   disp(strcat('No image files found in the directory :',32,pwd)); % print the current directory on the console
window
   f1 = errordlg(strcat('No image files found in the directory :',32,pwd), 'Error Dialog'); % print the current
directory on a dialog box
else
%
%
 filecount=0; %initialise counting the number of image files to be processed
 n = 0; % determines the number of features (=14*(n+1))i,e. for n=4, number of features extracted = 70.
 no_of_features=14*(n+1);
 feature_id = [1:no_of_features];
 header=cellfun(@num2str, num2cell(feature_id), 'UniformOutput', false);
 table_size=no_of_features+1;
 result_table=cell(1,table_size);
 result_table(1,:)={'file'};
 for i=2:no_of_features+1
    result_table(i)= header(i-1);
 end

  for n_file=1:size(all_files,1)     %for each file in the file list do the stat loop
    filename=all_files(n_file).name;
    Im = imread(filename);
    ncolors = size(Im,3); % number of color channels (=1 for indexed images and mono, = 3 for RGB images)
    if ncolors > 1
       for i=1:ncolors
         [features] = hist_lsb(Im(:,:,i),n);
         filecount = filecount+1;
         result_table(filecount+1,:)=[strcat(filename,'_',num2str(i)),num2cell(features)]; % put results in the tab
       end
    end
```

```
    end
end
```

Histogram Code 2

```
function [features] = hist_lsb(IM,n)
%extracts (n+1)*14 features from image IM

edge=-255:1:255; % edge is the x-axis of pixel differences

IM = double(IM); %convert pixel values to double
[M,N] = size(IM); %size of the image IM
Ih =zeros(M-1,N);  % Initialise Ih, Ih is difference of neighbouring pixels in the horizontal direction
for i=1:M-1
   for j=1:N
      Ih(i,j) = IM(i,j) - IM(i+1,j);
   end
end
hh=histc(Ih(:),edge); % hh is the histogram of Ih
hh=hh/256;  %normalise Ih
feah=hh(256:1:256+n); %feah is a vector of n features from Ih around from edge = 0 to edge = n
%
Iv =zeros(M,N-1); % Initialise Iv, Iv is difference of neighbouring pixels in the vertical direction
for i=1:M
   for j=1:N-1
      Iv(i,j) = IM(i,j) - IM(i,j+1);
   end
end
hv=histc(Iv(:),edge);
hv=hv/256;
feav=hv(256:1:256+n);
%
%
Id =zeros(M-1,N-1); % Initialise Id, Id is difference of neighbouring pixels in the diagonal direction
for i=1:M-1
   for j=1:N-1
      Id(i,j) = IM(i,j) - IM(i+1,j+1);
   end
end
hd=histc(Id(:),edge);
hd=hd/256;
fead=hd(256:1:256+n);

%
%
Ia =zeros(M-1,N-1); % Initialise Ia, Ia is difference of neighbouring pixels in the anti-diagonal direction
for i=1:M-1
   for j=1:N-1
      Ia(i,j) = IM(i,j+1) - IM(i+1,j);
   end
end
ha=histc(Ia(:),edge);
ha=ha/256;
feaa=ha(256:1:256+n);


%difference of difference images
[K,L]= size(Iv);
Ivv =zeros(K,L-1); % Initialise Ivv, Ivv is difference of neighbouring vertical pixel differences in the vertical
direction
for i=1:K
```

```
    for j=1:L-1
        Ivv(i,j) = Iv(i,j) - Iv(i,j+1);
    end
end
hvv=histc(Ivv(:),edge);
hvv=hvv/256;
feavv=hvv(256:1:256+n);
%
Ivh =zeros(K-1,L);% Initialise Ivh, Ivh is difference of neighbouring vertical pixel differences in the horizontal
direction
for i=1:K-1
    for j=1:L
        Ivh(i,j) = Iv(i,j) - Iv(i+1,j);
    end
end
hvh=histc(Ivh(:),edge);
hvh=hvh/256;
feavh=hvh(256:1:256+n);
%
Ivd =zeros(K-1,L-1);
for i=1:K-1
    for j=1:L-1
        Ivd(i,j) = Iv(i,j) - Iv(i+1,j+1);
    end
end
hvd=histc(Ivd(:),edge);
hvd=hvd/256;
feavd=hvd(256:1:256+n);
%
%
Iva =zeros(K-1,L-1);
for i=1:K-1
    for j=1:L-1
        Iva(i,j) = Iv(i,j+1) - Iv(i+1,j);
    end
end
hva=histc(Iva(:),edge);
hva=hva/256;
feava=hva(256:1:256+n);
%%%%%%%%%%%%%%%%%
[O,P] = size(Ih);

Ihh =zeros(O-1,P);
for i=1:O-1
    for j=1:P
        Ihh(i,j) = Ih(i,j) - Ih(i+1,j);
    end
end
hhh=histc(Ihh(:),edge);
hhh=hhh/256;
feahh=hhh(256:1:256+n);
%
Ihd =zeros(O-1,P-1);
for i=1:O-1
    for j=1:P-1
        Ihd(i,j) = Ih(i,j) - Ih(i+1,j+1);
    end
end
hhd=histc(Ihd(:),edge);
hhd=hhd/256;
```

```
feahd=hhd(256:1:256+n);
%
Iha =zeros(O-1,P-1);
for i=1:O-1
   for j=1:P-1
      Iha(i,j) = Ih(i,j+1) - Ih(i+1,j);
   end
end
hha=histc(Iha(:),edge);
hha=hha/256;
feaha=hha(256:1:256+n);
%%%%%%%%%%%%%%%%%%
[Q,R] =size(Id);

Idd =zeros(Q-1,R-1);
for i=1:Q-1
   for j=1:R-1
      Idd(i,j) = Id(i,j) - Id(i+1,j+1);
   end
end
hdd=histc(Idd(:),edge);
hdd=hdd/256;
feadd=hdd(256:1:256+n);
%
Ida =zeros(Q-1,R-1);
for i=1:Q-1
   for j=1:R-1
      Ida(i,j) = Id(i,j+1) - Id(i+1,j);
   end
end
hda=histc(Ida(:),edge);
hda=hda/256;
feada=hda(256:1:256+n);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[S,T] =size(Ia);

Iaa =zeros(S-1,T-1);
for i=1:S-1
   for j=1:T-1
      Iaa(i,j) = Ia(i,j+1) - Ia(i+1,j);
   end
end
haa=histc(Iaa(:),edge);
haa=haa/256;
feaaa=haa(256:1:256+n);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

features=[feah',feav',fead',feaa',feavv',feavh',feavd',feava',feahh',feahd',feaha',feadd',feada',feaaa']; %concatenate
the 14 feature vectors


% the h(0) values in features are calculated differently from the the other feature values (see equation 10).
% we need to divide/scale all the features that are not h(0) by h(0)
h0_ind=1:n+1:14*(n+1); %indices of h(0) values
all_ind = 1:14*(n+1); %indices of all values
none_h0_ind =setdiff(all_ind,h0_ind); % indices of all values excluding h(0) values

for i=none_h0_ind
   features(i) = features(i)/features(floor(i/(n+1))+1); %divide none h(0) values by h(0)
end
```

Ethics Form

Project ref	**P45604**
Full name	Ahd AlJarf
Faculty	[EEC] Faculty of Engineering, Environment and Computing
School/FRC	[CM] Computing & The Digital Environment
Supervisor	Saad Amin
Module Code	ECCOM
Project Title	Develop a Detection System for Colour Images using the Colour Gradient Co-occurrence Matrix (CGCM) and Histogram of Difference Image Features
Project Dates	23/01/2012 - 31/01/2017
Date Created	19/08/2016 16:33
Project Summary	*The work of this thesis focuses on image steganlysis. A detection system is presented that has three different steganalysis techniques. All technique are addressing blind image steganalysis and established by extracting selections of image features.* *The first steganalysis technique developed based on extracting varieties of colour gradient co-occurrence matrix (CGCM).* *The second steganaslysis technique developed by extracting large number of histogram features. The features are extracted by exploiting the histogram of difference image, which is usually a generalised Gaussian distribution centered at 0.* *Finally the tested CGCM features and histogram features were merged together to improve the performance of the system. In addition, merging two different types of features increases taking advantages of their properties to increase the system ability in terms of detection.* *Large image-data base was created to train and test the system.* *The proposed detection system was trained and tested to distinguish stego images from clean ones using the Discriminant Analysis (DA) classification method and Multilayer Perceptron neural network (MLP).*

Is the project self-funded? Yes

Are you required to use a Professional Code of Ethical Practice appropriate to your discipline? No

Project Details

1	What is the purpose of the project? *The main aims of the study are to:* *1. Use and compare several existing steganography methods to embed information into clean images to create varieties of stego-images.* *2. Evaluate and compare several existing steganalysis methods to detect the embedded information.* *3. Create images data-set for validation.* *4. Develop methods based on expanding existing algorithms to detect embedded information in images.* *5. Use classification methods, to assess and evaluate the developed methods.*
2	What are the planned or desired outcomes? *Proposing a detection system based on extracting statistical features from the CGC matrix and histogram features. The detection system was designed to work as a blind form of steganalysis. In this type of steganalysis, the system does not target specific steganography methods or specific image formats. The proposed system is able to to differentiate between clean and stego images. The proposed system can deal with different image formats and detect many steganography methods.*
3	Explain your research design *A framework are designed to develop the proposed detection system. The implementation includes extracting features, create stego images using steganography tools and analysing the results using classifiers.*
3	

4	Outline the principal methods you will use *I have developed algorithms based on existing steganalysis methods. The developed algorithm are written in MATLAB and the results are analysed by different classifiers using SPSS.*	
5	Are you proposing to use an external research instrument, validated scale or research method? (e.g. a measurement scale, questionnaire, interview schedule, observation protocol for ethnographic work or in the case of unstructured data collection, or a topic list)	No
6	Are you intending to undertake research which will investigate activist, religious or political groups directly or indirectly involved in armed struggles, terrorism or a form of extremism that lies outside the commonly-accepted norms of British Society?	No
7	Are you dealing with Secondary Data? (e.g. sourcing info from websites, historical documents)	No
8	Are you dealing with Primary Data involving people? (e.g. interviews, questionnaires, observations)	No
9	Are you dealing with Personal or Sensitive data?	No
10	Is the project solely desk based? (e.g. involving no laboratory, workshop or off-campus work or other activities which pose significant risks to researchers or participants)	No
11	Are there any other ethical issues or risks of harm raised by the study that have not been covered by previous questions?	No

1. Laboratory/Workshops		
1	Does any part of the project involve work in a laboratory or workshop which could pose risks to you, researchers or others?	No

2. Research with non-human vertebrates		
1	Will any part of the project involve animal habitats or tissues or non-human vertebrates?	No

3. Blood Sampling / Human Tissue Analysis		
1	Does your study involve collecting or use of human tissues or fluids? (e.g. collecting urine, saliva, blood or use of cell lines, 'dead' blood)	No

4. Travel		
1	Does any part of the project require data collection off campus? (e.g. work in the field or community)	No