

**The Hiding of Multimedia Secret Files in Dual RGB Cover  
Images Using LSB Steganography Techniques**

الإخفاء للملفات السرية المتعددة الوسائط في صور مزدوجة ثلاثية الألوان باستخدام  
تقنيات الإخفاء في البتات الأقل وزنا

By

**Marwah Tareq Ahmed Al-Bayati**

**Supervisor**

**Dr. Mudhafar Al-Jarrah**

**A Thesis Submitted in Partial Fulfillment of the Requirements for the  
Master Degree in Computer Science**

**Department of Computer Science**

**Faculty of Information Technology**

**Middle East University**

**Amman, Jordan**

**May, 2016**

### Authorization

I Marwah Tareq Ahmed Al-Bayati, authorize Middle East University (MEU) to provide copies of my thesis to the concerned libraries, establishments, and institutions upon request.

Name: Marwah Tareq Ahmed Al-Bayati

Date: 21/5/2016

Signature:



## اقرار تفويض

انا مروة طارق احمد البياتي افوض جامعة الشرق الاوسط بتزويد نسخ من رسالتي  
للمكتبات المعنية، المؤسسات، الهيئات عند طلبها.

الاسم: مروة طارق احمد البياتي

التاريخ: 2016/5/21

التوقيع:



## Thesis Committee Decision

This is certifying that the thesis entitled “**The Hiding of Multimedia Secret Files in Dual RGB Cover Images Using LSB Steganography Techniques**” was successfully defended and approved on 21/5/2016.

### Examination Committee Members Signature

Supervisor

**Dr. Mudhafar Munir Al-Jarrah**

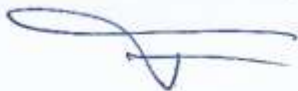
Middle East University



(Head of the Committee and Internal Committee Member)

**Prof. Dr. Ahmad Kayed**

Middle East University



(External Committee Member)

**Prof. Dr. Bilal Abu-huda**

Yarmouk University



## Acknowledgement

Prior to acknowledgments, I must glorify Allah the Almighty for His blessings who gave me courage and patience to carry out this work successfully. Then I would like to express my deepest gratitude to my Advisor: Dr. Mudhafar AL-Jarrah for his persistent support and his guidance in answering all my questions about my research, I also wish to express my deepest gratitude to the members of the committee for spending their precious time on reading my thesis and giving me encouragement and constructive comments. I would like to thank all Information Technology Faculty members at Middle East University and Thank to my Grandmother; I could not do anything without you. Last but not least a big thank to my loving family, my father, my mother, my brother and my sister I love you all.

# *Dedication*

TO the big heart, my father

TO the fountain of patience and optimism and hope,  
my mother

TO my happiness in life, my grandmother

TO the wonderful girl, my sister

TO the best friend, my brother

*Marwah*

Table of Contents	
Title.....	I
Authorization.....	II
إقرار التفويض.....	III
Thesis Committee Decision.....	IV
Acknowledgments.....	V
Dedication.....	VI
Table of contents.....	VII
List of Abbreviations.....	X
List of Figures.....	XII
List of Tables.....	XIV
Abstract in English.....	XV
Abstract in Arabic.....	XVII
<b>Chapter One: Introduction.....</b>	<b>1</b>
1.1 Topic .....	1
1.2 Problem Statement.....	3
1.3 Research Questions.....	3
1.4 Objectives .....	4
1.5 Motivation.....	5
1.6 Methodology.....	6
1.7 Limitations of the Present Work.....	6
1.8 Thesis Organization.....	7

<b>Chapter Two: Background and Related Work.....</b>	<b>8</b>
2.1 Background.....	8
2.2 Definition and Concept of Steganography.....	8
2.3 BMP Image Format.....	10
2.4 Quality Evaluation Metrics.....	11
2.5 Image Steganography Characteristics.....	12
2.6 Steganography Models .....	14
2.7 Steganography Categories.....	15
2.8 Steganography Techniques.....	16
2.8.1 Spatial Domain.....	16
2.8.2 Transform Domain.....	18
2.9 Types of Steganalysis Attacks.....	18
2.10 High Capacity Hiding.....	19
2.11 Security of the Hidden Data.....	20
2.12 Steganography Using Gray-Scale Images.....	20
2.13 Related Work.....	21
.....	
<b>Chapter Three: The Proposed Model.....</b>	<b>28</b>
3.1 Overview .....	28
3.2 The Proposed Model's Required Features .....	28
3.3 Design Considerations.....	29
3.4 Data Layout of the Cover Image and the Secret File.....	31
3.5 Data Structures.....	32
3.6 Processing Steps of the DuoHide Model.....	33
3.6.1 Embedding Steps.....	33
A- The Main Embedding Algorithm.....	33
B- Embed Half-Bytes Sub-Algorithm.....	34
3.6.2 Extraction Steps.....	35



A- The Main Extraction Algorithm .....	35
B- Extract the Hidden Data Sub-Algorithm.....	36
3.7 Illustration of the Embedding and Extracting Procedure.....	37
<b>Chapter Four: Experimental Work and Discussion of Results.....</b>	<b>38</b>
4.1 Overview.....	38
4.2 Embedding / Extracting Results of Multimedia Files in Large Images.....	39
4.3 Embedding / Extracting Results of Image Files in Standard Test Images.....	42
4.4 PSNR Imperceptibility Results.....	45
4.5 Image Histogram Analysis.....	48
4.6 Comparison with Other Models.....	55
4.6.1 Comparison with a 2-3-3 LSB Model Using JPG Covers.....	55
4.6.2 Comparison Results of One Cover and Two Covers 4-bit LSB Embedding.....	56
4.7 Comparison of PSNR Results Using Heterogeneous Cover Pairs.....	58
4.8 Verification of Results.....	61
<b>Chapter Five: Conclusion and Future Work.....</b>	<b>64</b>
5.1 Conclusion.....	64
5.2 Future Work.....	65
References.....	67
<b>Appendix A DuoHide Dataset Sources.....</b>	<b>73</b>
A.1 Cover images.....	74
A.2 Secret multimedia files embedding in Labelle.bmp and Poppies.bmp.....	76
A.3 Secret Multimedia Files Embedded in Lena.bmp.....	79

## List of Abbreviations

<b>LSB</b>	Least Significant Bit
<b>MSB</b>	Most Significant Bit
<b>RGB</b>	Red-Green-Blue
<b>PSNR</b>	Peak Signal -to- Noise Ratio
<b>dB</b>	Decibel
<b>JPEG</b>	Joint Photographic Experts Group
<b>BMP</b>	Bitmap Image File
<b>HVS</b>	Human Vision System
<b>BPP</b>	Bit Per Pixel
<b>GIF</b>	Graphics Interchange Format
<b>PNG</b>	Portable Network Graphics
<b>HB</b>	Half-Byte
<b>TIFF</b>	Tagged Image File Format
<b>RSC</b>	Ratio of Secret to Cover
<b>HC</b>	Hiding Capacity
<b>LH</b>	Left Half
<b>RH</b>	Right Half
<b>BC</b>	Byte Count
<b>HBC</b>	Half Byte Count
<b>SB</b>	Secret Byte

<b>MP3</b>	MPEG Audio Layer 3
<b>MP4</b>	MPEG-4 Part 14
<b>WMV</b>	Windows Media Video
<b>RGBA</b>	Red, Green, Blue, Alpha

<b>List of Figures</b>	
<b>Figure 2.1</b>	Tradeoff between Image Steganography Features.....13
<b>Figure 2.2</b>	Different Models of Steganography.....14
<b>Figure 3.1</b>	Reading Multimedia Files as a Stream of Bytes.....31
<b>Figure 3.2</b>	Images Cover in RGB Format.....31
<b>Figure 3.3</b>	The main Embedding Algorithm.....33
<b>Figure 3.4</b>	Embed Half-Bytes Algorithm.....34
<b>Figure 3.5</b>	The Main Extracting Algorithm.....35
<b>Figure 3.6</b>	Extract the Hidden Data Sub Algorithm.....36
<b>Figure 3.7</b>	Example the of Embedding and Extracting Procedure.....37
<b>Figure 4.1</b>	Labelle Cover + Krokussen.Png with Stego1 and Stego2.....39
<b>Figure 4.2</b>	Labelle Cover + Xynthia_animated.Gif with Stego1 and Stego2.....40
<b>Figure 4.3</b>	Labelle Cover + Renoir4-128.Jpg with Stego1 and Stego2.....40
<b>Figure 4.4</b>	Poppies Cover + Beethovenno9.Mp4 with Stego1 and Stego2.....41
<b>Figure 4.5</b>	Poppies Cover + Xynthia_animated.Gif with Stego1 and Stego2.....41
<b>Figure 4.6</b>	Poppies Cover + Renoir4-128.Jpg with Stego1 and Stego2.....42
<b>Figure 4.7</b>	Lena Cover + Vase1024.Jpg with Stego1 and Stego2.....43
<b>Figure 4.8</b>	Lena Cover + Vase 512.Jpg with Stego1 and Stego2.....43
<b>Figure 4.9</b>	Lena Cover + Vase 256.Jpg with Stego1 and Stego2.....44
<b>Figure 4.10</b>	Lena Cover + Vase128.Jpg with Stego1 and Stego2.....44
<b>Figure 4.11</b>	Lena Cover + Renoir4-128.Jpg with Stego1 and Stego2.....45
<b>Figure 4.12</b>	Histogram of Lena Cover .....49
<b>Figure 4.13</b>	Histogram of Stego1 Lena Cover + Kodim24.Png.....50
<b>Figure 4.14</b>	Histogram of Stego2 Lena Cover + Kodim24.Png.....50

<b>Figure 4.15</b>	Histogram of Lena Cover.....	51
<b>Figure 4.16</b>	Histogram of Stego1 Lena Ccover + Roses.Jpg.....	52 XIII
<b>Figure 4.17</b>	Histogram of Stego2 Lena .Bmp + Roses.Jpg .....	52
<b>Figure 4.18</b>	Histogram of Lena Cover.....	54
<b>Figure 4.19</b>	Histogram of Stego1 Lena Cover + Renoir4-128.Jpg.....	54
<b>Figure 4.20</b>	Histogram of Stego2 Lena Cover + Renoir4-128.Jpg.....	54
<b>Figure 4.21</b>	PSNR Values Between 2-3-3 LSB and DuoHide Models .....	56
<b>Figure 4.22</b>	PSNR Values Between One-Cover 4-bit LSB and DuoHide Embedding.....	57
<b>Figure 4.23</b>	Peppers Cover with Stego2 + Kodim24.PNG.....	60
<b>Figure 4.24</b>	Peppers Cover with Stego2+ Roese.JPG.....	60
<b>Figure 4.25</b>	Peppers Cover with Stego2+ Renoir4-128.JPG.....	61
<b>Figure 4.26</b>	Verification of the Integrity of Extracted Secret File Extime- lapse8.....	62
<b>Figure 4.27</b>	PSNR Results of ImageMajick for Verification of DuoHide PSNR Results.....	63

<b>List of Tables</b>	
<b>Table 2.1</b>	Summary of Related Work Features and Drawback.....26
<b>Table 4.1</b>	PSNR Results for Labelle Cover in BMP.....46
<b>Table 4.2</b>	PSNR Results for Poppies Cover in BMP.....47
<b>Table 4.3</b>	PSNR Results for Cover Image Lena BMP.....48
<b>Table 4.4</b>	Comparison of PSNR Values Between 2-3-3 LSB and DuoHide Models Using Secret Image Lena128.Jpg(26.1 KB)..... 55
<b>Table 4.5</b>	Comparison of PSNR Values Between One-Cover 4-bit LSB and DuoHide Two Covers Embedding.....57
<b>Table 4.6</b>	PSNR Results for Cover Images Lena. BMP vs. Peppers.BMP with Different Sizes of Secret File.....59

# **The Hiding of Multimedia Secret Files in Dual RGB Cover Images Using LSB Steganography Techniques**

**By: Marwah Tareq Ahmed Al-Bayati**

**Supervisor: Dr. Mudhafar Al-Jarrah**

## **Abstract**

Steganography, the technology of protecting a secret message by embedding it inside a cover image, continues to be investigated and enhanced as an alternative data protection method. The main advantages of steganography over cryptography are: it does not provoke attacks as ideally, the existence of a secret message cannot be observed, and there is no key management overhead.

This thesis deals with hiding multimedia files in true color RGB cover images with an emphasis on high hiding capacity and secret data protection. A proposed model (DuoHide) is presented in which a secret multimedia file, regardless of its type, is read as a stream of bytes and split vertically into two parts, one part contains the LSB half-bytes and the other part contains the MSB half-bytes. The two parts are hidden inside two uncompressed RGB images using 4-bit LSB replacement technique. Extraction of the secret file is achieved through merging the two hidden parts. The model is implemented in the MATLAB environment.

The proposed model is evaluated using a set of public multimedia files; images, audios, and videos, of various sizes. The secret file sizes ranged from 5% to about 100% of the cover image's size. The purpose of the evaluation is to measure imperceptibility based

on two criteria: Visual, by comparison between stego and cover images, and Statistical, using the Peak Signal-to-Noise Ratio (PSNR) value between the stego and the cover images. The experimental results showed that even at the highest embedding ratio, which is based on the secret to cover sizes, there are no perceptible visual differences between cover and stego images. The PSNR value is calculated as PSNR1, for cover and stego1, and PSNR2 for cover and stego2. There is a small difference between PSNR1 and PSNR2. The lowest PSNR value is around 31 dB for the highest embedding ratio, which is considered acceptable concerning statistical imperceptibility. The PSNR value increased as the embedding ratio decreased, reaching around 65 Decibel (dB) for the case of 5% embedding ratio. Additional comparisons are performed, using the standard image Lena as cover, and a set of secret images. The first comparison looks at PSNR values using the DuoHide model against a 3-bit LSB method. The PSNR value for the DuoHide is higher than that of the 3-bit model. This difference can be due to using a compressed JPG cover in the 3-bit model results, where the compression can add distortion. The second comparison is between DuoHide and a single cover 4-bit LSB method. The PSNR value of the dual cover method is around 3 dB higher than the single cover method, despite the fact that the dual cover method had double the hiding capacity.

Concerning enhancing the security of the secret file, in case an attacker manages to recover it from the stego file, the attacker will only get an incomprehensible set of bits.

The thesis ends with conclusions and suggestions for future work based on observations on the present research.

**Keywords:** Steganography, Dual Hiding, Secret File, Cover File, Embedding, Extracting.



**الإخفاء للملفات السرية المتعددة الوسائط في صور مزدوجة ثلاثية الألوان باستخدام تقنيات الإخفاء**

في البتات الأقل وزنا

إعداد

مرودة طارق أحمد البياتي

إشراف

الدكتور مظفر الجراح

الملخص

حقل إخفاء المعلومات ، التقنية الخاصة بحماية الرسائل السرية بتضمينها ضمن رسائل غطاء ، يتواصل البحث فيه وتطويره كأسلوب بديل لحماية البيانات . الميزتين الرئيسيتين لإخفاء المعلومات بالمقارنة مع أسلوب التشفير هما : انه لا يثير الهجمات ويتمثل بوجود رسالة سرية لا يمكن ملاحظتها ، كما لا يتحمل المرسل والمتلقي جهد إدارة كلمات السر.

يتعامل البحث في هذه الاطروحة مع إخفاء الملفات المتعددة الوسائط في صور غطاء غير مضغوطة ذات الحزم اللونية الثلاث (أحمر ، أخضر ، أزرق) ، مع التركيز على زيادة سعة التخزين وحماية سرية البيانات. الموديل المقترح (دوهايد : الاخفاء المزدوج) يتم فيه قراءة الملفات المتعددة الوسائط ، بغض النظر عن نوعها ، كسلسلة من البايتات ، ويجري تقسيم الملف عموديا الى جزئين ، يحتوي الجزء الاول على أنصاف البايتات الاقل أهمية عدديا (LSB) ويحتوي الجزء الثاني على أنصاف البايتات الاكثر أهمية (MSB). يجري إخفاء الجزئين في ملفي غطاء منفصلين وذلك من خلال تبديل أنصاف البايتات السرية مع اربعة بتات ذات الاقل أهمية في ملفي الغطاء . عملية

إسترجاع الملف السري تتم من خلال إستخراج أنصاف البايتات من ملفي التغطية ودمجها لإعادة تشكيل الملف الاصيلي نفذ تطبيق الموديل المقترح بأستخدام بيئة MATLAB .

تقييم الموديل المقترح تم باستخدام ملفات عامة متعددة الوسائط (صورة ، صوت ، فيديو) ذات احجام مختلفة ، حيث تراوح حجم الملف السري بين 5% و حوالي 100% من حجم ملف الغطاء. يهدف التقييم الى قياس فعالية الاخفاء على أساس معيارين : التقييم البصري : من خلال المقارنة بين صورة الغطاء الاصلية وصورة الاخفاء (stego)، والإحصائي: وذلك باستخدام مقياس نسبة ذروة الإشارة إلى الضوضاء (PSNR) بين صورة الغطاء وصورة الاخفاء.

أظهرت النتائج التجريبية إلى أن إستخدام أعلى نسبة تضمين حسب الموديل المقترح ، والمستندة الى نسبة حجم الملف السري الى حجم ملف الغطاء ، لم تحصل فروق بصرية محسوسة بين صورة التغطية وصورة stego . تم إحتساب مقياس PSNR في قيمتين : PSNR1 بين صورة الغطاء وصورة stego1 و PSNR2 بين صورة الغطاء وصورة stego2. أدنى قيمة PSNR كانت 31 ديسبل وتمثل حالة أعلى نسبة تضمين ، والتي تعتبر مقبولة فيما يتعلق بحساب فعالية الاخفاء الاحصائي. كما بينت النتائج إزدياد قيمة PSNR مع إنخفاض نسبة التضمين ، لتصل الى 65 ديسبل مقابل نسبة إخفاء 5% . كذلك شمل البحث على إجراء مقارنات إضافية بإستخدام الصورة المعيارية "لينا" كصورة غطاء لتضمين مجموعة من الصور التي أستخدمت كصور سرية . المقارنة الاولى أجريت بين قيمة PSNR بإستخدام الموديل المقترح دوهايد مع موديل يخزن في 3-بت من المواقع الاقل اهمية (3-bit LSB) ، وكانت قيمة PSNR لموديل دوهايد أعلى من موديل 3-بت . يمكن أن يكون الفرق قد نتج عن أن موديل 3-بت إستخدم ملف غطاء مضغوط ونتج عن الضغط زيادة في التشويش . أجريت المقارنة الثانية بين نتائج PSNR لموديل دوهايد الذي يستخدم ملفي غطاء

وطريقة إستخدام ملف غطاء أحادي والخرن في 4-بت . كانت قيم PSNR عند إستخدام غطائين أعلى بحوالي 3 ديسبل عن نتائج إستخدام غطاء واحد مع أن إستخدام الغطائين أدى الى مضاعفة نسبة التضمين . أما فيما يخص حماية محتوى الملف السري في حالة تمكن المهاجم من استرجاعه من ملف الاخفاء ، فإن ما سيحصل عليه المهاجم لن يكون سوى بنات غير قابلة للفهم .

تنتهي الاطروحة بتقديم إستنتاجات وتوصيات لاجتاه مستقبليه بالاستناد على نتائج البحث الحالي .

**الكلمات المفتاحية:** علم الاخفاء، الاخفاء المزدوج، الملف السري، ملف الغطاء، التضمين،

الاستخراج

# Chapter one

## Introduction

### 1.1 Topic

Multimedia file transfer over the Internet is becoming an important part of information technology usage, due to the vast increase in document exchange for business and personal communications, in particular among social networks users, through mobile devices and computers. Sending unprotected sensitive documents over the Internet is risk-prone, in our imperfect world where criminals and hackers are doing their best to get hold of documents flying around the Internet (Al-Ani, Zaidan, Zaidan, & Alanazi, 2010).

Therefore, the demand for better security measures and procedures is increasing, to prevent unauthorized access to private documents, during transmission over local networks and the Internet. Higher interest in information security is leading to the development of technologies and methods for secret data protection against any attack and threat by adversaries (Chedded, Condell, Curran, & Mc Kevitt, 2010) .

Securing multimedia data requires preventing unauthorized users from access, distortion, destruction, detection or modification of the data during its transfer, and any system that transmits such data through communication channels should provide the necessary mechanisms to protect its data. Also, the level of security and sensitivity of the exchanged documents influence the need for different types of data protection methods; for

example, diplomatic, military and banking documents require far more sophisticated protection methods (Ghosal, 2011).

There are two primary methods of data security protection, encryption, and steganography. The encryption method protects data by converting it to an unclear form that cannot be perceived by attackers. The weak point of cryptography techniques is that even though the message is encrypted, knowledge of its existence would lead to attempts by adversaries to break the encryption and decipher the encoded data. The second method of data security is steganography, which tries to conceal the presence of the secret data by hiding it inside documents of various types such as text, image, audio, and video. The documents that are used to cover up the secret data are called cover, carrier, or clean documents. The hiding of secret data is carried out by mixing bits of the secret information with bits of the cover document in such a way that an observer or attacker will not notice a perceivable change in the cover document. Steganography involves concealing the secret data so that it is imperceptible to the observer (Doshi, Jain, & Gupta, 2012). Each data hiding scheme consists of the embedding algorithm and the extracting algorithm. The embedding algorithm is used to hide the secret message inside a cover, and the extracting algorithm is used to recover the secret message (Thanikaiselvan, Arulmozhivarman, Subashanthini, & Amirtharajan, 2013).

Some applications of data protection combine both the encryption and steganography methods, to take advantage of the benefits of both approaches in strengthening information security (Juneja & Sandhu, 2013).

The work in this thesis focuses on the steganography techniques for data protection, but it is possible that encryption can be added later to enhance the proposed model, within the context of an implemented product.

In this research, a model is proposed to hide a secret multimedia file within two uncompressed RGB cover images, where the secret file is vertically split into two parts; each part is stored in one of the two covers.

## 1.2 Problem Statement

The exchange of confidential and private information has changed lately from sending just short text messages, to sending multimedia files of various formats and sizes. The problem addressed in this research is the hiding of secret multimedia files within true color images in a way that prevents an adversary from the acquisition of the hidden file, in case she/he detects the presence of embedded data within the carrier images.

## 1.3 Research Questions

The research in this thesis attempts to provide answers to the following questions:

1. What is the possible solution for handling the hiding of a large secret multimedia file inside cover images, without uncompressing the secret file and without the need for compression if the file is uncompressed?

2. Can an attacker be prevented from accessing the hidden secret data if she/he detects the existence of an embedded message inside a cover?
3. What are the metrics of imperceptibility that will be used with the proposed solution?
4. Using the proposed solution, what will be the result of distortion comparison between a clean cover and the corresponding cover that is loaded with a secret file?
5. What is the effect of secret file size increase on the value of the distortion metric?
6. How will the integrity of the hidden file be verified after it is extracted?
7. How will the results of the distortion metric be verified?

## 1.4 Objectives

This research aims to enhance the hiding capacity and security of steganography in uncompressed RGB cover images; to allow for the hiding of multimedia files, and to strengthen the protection of the embedded data against an attack by an adversary, in case presence of the hidden data is detected through an analytical tool. The following objectives are set for this research:

1. Visual imperceptibility of cover images containing embedded data.
2. Acceptable level of distortion based on standard metrics such as PSNR.
3. The integrity of the hidden data should be maintained; there should be no loss or change of the hidden data and no change in the size of the extracted secret file as a result of the embedding / extracting process; the original secret file should be identical to the extracted file.

4. Provide high capacity payload to accommodate multimedia files.
5. Compressed multimedia files should remain compressed during the processes of embedding and extraction.

## 1.5 Motivation

The increasing need and demand for the privacy and protection of multimedia documents exchanged over the Internet, or inside enterprises and organizations, and the rising trend in criminal attacks on the transferred data, with malicious intents to access or damage the data or both, are motivating further research to enhance the security of data.

The steganography approach is one of the essential techniques in protecting the security of data. Steganography has become an attractive alternative to cryptography, due to the simplicity of implementation, as well as the fact that cryptography can be broken once discovered. On the other hand, steganography does not invite attacks because ideally there is nothing that can be seen to be attacked.

In addition to the security issue, limitations of the hiding capacity of the cover image, to accommodate multimedia files, can be a problem that needs to be tackled, to allow for higher capacity embedding.



## 1.6 Methodology

The adopted methodology approach in this research is experimental, which involved the following main steps:

1. Identifying possible technical solutions to achieve the defined objectives of high capacity hiding of multimedia data files in RGB images, and the enhancement of security of the embedded data.
2. Implementation of the selected technical solutions using the Matlab software.
3. Evaluation of the results of embedding various types of multimedia files (image, audio, and video) in standard public RGB true color images of the BMP format, of various sizes, using hiding capacity measures and distortion evaluation metrics.
4. Comparison with results of previous research using standard images.

## 1.7 Limitations of the Present Work

The proposed work has some limitations that are outside the scope of the research, these are:

1. Not using compressed images for cover.
2. Not using more than two covers.

A practical limitation that the research faced is the unavailability of both secret and cover image details (size and dimensions) in many publications, which would have provided a source of data for comparison between different models.

## 1.8 Thesis Organization

This thesis consists of five chapters:

- Chapter one presents an introduction to the domain of steganography, the research topic of the thesis, problem statement, objectives, research questions from our perspectives, and the adopted methodology.
- Chapter two presents the general concepts and definitions of steganography, and gives a summary of literature work associated with this thesis.
- Chapter three presents the methodology and implementation of the proposed steganography scheme, represented through algorithms and flow charts.
- Chapter four presents the experimental work and discussion of results.
- Chapter five presents conclusions about the proposed methodology, results, and findings of the experimental work, and provide suggestions for future work along the lines of the current research.

## Chapter 2

### Background and Literature Review

#### 2.1 Background

A considerable amount of research has been carried out over the past several years to strengthen the effectiveness of the steganography approach as a secure data protection method. Steganography has been suggested as an alternative method to cryptography that does not suffer from key management overhead. Recently, the research effort is tackling several problems areas related to steganography, such as data hiding techniques that are robust against steganalysis, the format of media files that can serve as cover for storing secret data, possible secret data formats, hiding capacity of the cover media, security of the hidden data, and steganalysis techniques.

#### 2.2 Definition and Concept of Steganography

Steganography is one of the techniques of secret data protection that involves concealing the secret data within media files such as images, audio files, and video files. The concept of steganography is derived from the Greek word “Steganos”, which means covered writing. The steganography technique includes three elements: the cover object, which is the media where the secret message is hidden, the secret message, and the stego object, which combines the cover object and the secret message in a discreet way, sometimes referred to as the steganogram (Gupta, 2013).

The origin of steganography dates back to around 440 BC, during the Greek civilization, and there are many stories over the years about the utilization of steganography to hide secret messages from the enemy, for example by writing the secret message on the wood back of a wax tablet before applying the bee-wax surface. Wax tablets were used then as reusable writing surfaces (Kumar & Pooja, 2010).

In modern times, the aim of steganography is sending a message over the network to the intended receiver and preventing anyone else to know that it was sent. It is meant to make communication safer by avoiding to draw attention to the existence of the transmission of a hidden data (Cheddar, et.al., 2010).

The secret messages can be concealed in different data formats so that it will be undetectable by the Human Visual System (HVS), to avoid raising suspicion of an observer to the transmission of secret data. Steganographic technologies have become very important in the Internet field for protecting the exchanged data. However, it is possible to combine steganography with encryption so that if a hidden message is exposed, encryption will provide a second line of defense (Mandal, 2012).

One of the limitations of using image-in-image steganography is that the size of the secret image can be close to or larger than cover image. Compression can be a solution for hiding large secret data, but most effective compression techniques are lossy. The same problem applies if other media files need to be hidden inside cover images (Yugala, 2013).

Image steganography techniques can be divided into two major categories: spatial domain and frequency domain. Spatial domain techniques include the Least-Significant-Bit (LSB) method, which replaces least significant bits of the cover's pixels with bits of the

messages. The advantages of the spatial domain method are in less possibility for degradation of the original image, and more information can be stored in the image. In the frequency domain technique, images are first transformed into spatial domain and then the message is embedded in the cover image (Hussain, 2013).

This thesis focuses on using the LSB technique for hiding multimedia files in uncompressed 24-bit RGB images. The main disadvantage of the LSB technique is that it has considerably low robustness against modification. Initially, LSB steganography schemes used one bit of each pixel, the least significant bit, to store bits of the secret message, but multiple least significant bits have also been used, up to 4 bits per byte. The main advantages of the LSB technique are that it allows for higher hiding capacity, and it is fundamentally uncomplicated and simple to research (Sandilya, 2014).

## 2.3 BMP Image Format

The Bitmap Image File (BMP) format is used to store uncompressed bitmap digital images. Bitmap file format supports several pixel formats, which are used to store images with a color depth of 1, 4, 8, 16, 24 or 32 bits per pixel. The BMP format allows an optional alpha channel for variable transparency, so a 32-bit BMP image consists of 24-bit RGB color channels in addition to the 8-bit alpha channel (Sarayreh, 2014).

BMP image format structure contains four parts. The first part is the file header of 14 bytes, which includes information about the type, size, and arrangement of the bitmap file, to indicate that the file is a BMP file or not. The second part consists of 40 bytes, which contains the width and height of the image and the number of bits that are used to

represent the color intensities of pixels. The third part is the optional palette, which consists of a block of bytes for color indexing, and the last part is the image data (Koppola, 2009).

The BMP image format is widely used in steganography as a cover due to its uncompressed feature, which simplifies comparison between different methods (Juneja & Sandhu).

In earlier studies in steganography and steganalysis, the gray-scale 8-bit BMP was the most popular format for the cover image, and more recently, the RGB BMP format has been used due to its larger hiding capacity.

## 2.4 Quality Evaluation Metrics

The Peak Signal-to-Noise Ratio (PSNR) is an engineering term for the ratio between the maximum possible power of a signal and the power of distorting noise that affects the quality of the signal. As many signals have a very wide dynamic range, PSNR is expressed on a logarithmic decibel scale. In image processing, the PSNR metric is used as a measure of image quality, whose value is influenced by the distortion in a modified image. The PSNR value refers to the quality approximation between the original cover image and the distorted stego image, which contains embedded data. The Mean Square Error (MSE) is calculated first, as shown in the first equation below, which is the sum of the squared error between the cover and the stego images, where the error is the numeric difference between pixels of the two images. The PSNR value is computed as shown in the second equation below (Efimushkina, 2013):

$$MSE = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|O(i, j) - D(i, j)\|^2 \quad (1)$$

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

The symbols O and D are the original image and distorted image, and m x n is the image pixel resolution (height times width), while Max is the peak value of the pixel in the image, which is represented by 255 in 8-bit format. For 24-bit RGB format, the MSE is calculated for each color channel separately, then the sum of the three channels' MSE is divided by three (Yalman, 2013).

## 2.5 Image Steganography Characteristics

The performance of different steganographic methods can be estimated by three main properties, capacity, robustness, and imperceptibility, as follows:

1. Capacity: it refers to the amount of data that can be stored inside the cover image. It is represented as a bit per pixel (bpp).
2. Imperceptibility: it is the quality of the stego image in concealing the secret data without any noticeable distortion.
3. Robustness: it is the capability of the stego image to steadfastness for manipulation, such as filtering, cropping, rotation, compression.

A tradeoff between those features is shown in Figure 2.1 (Swain & Lenka, 2014)

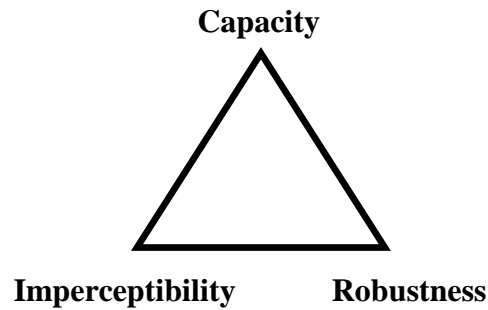


Figure 2.1 Tradeoff Between Image Steganography Features

To illustrate the tradeoff between capacity and imperceptibility, we take an RGB image of 512 x 512 pixels, in which we either change a 1-LSB bit or 2-LSB bits, for all pixels. In the 1-LSB case, the hiding capacity is  $512 \times 512 \times 3 / 8 = 98,304$  bytes. In the 2-LSB case, the hiding capacity is 196,608 bytes. Therefore, doubling the hiding capacity for the same cover by using 2-LSB instead of 1-LSB will result in more distortion in the 2-LSB image. Regarding tradeoff between capacity and robustness for the same cover, higher hiding capacity means that when more bits are changed due to embedding, the new image will be less robust if it is cropped or manipulated.



## 2.6 Steganography Models

There are five models that are used in hiding data in the steganography techniques, as shown in Figure 2.2 (Hussain & Hussain, 2013).

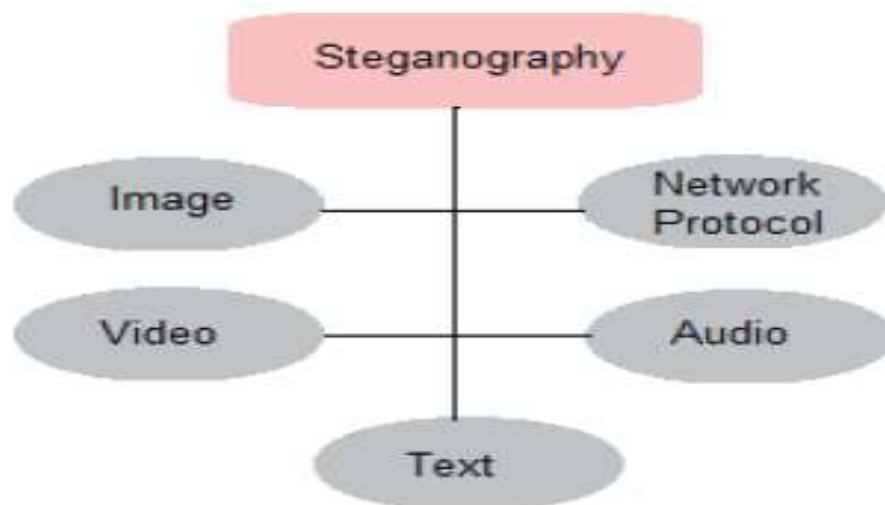


Figure 2.2 Different Models of Steganography

1. Text steganography: In this method, the secret data, which is also in text format, is hidden in the  $n$ th letter of every word. This method is not used a lot because the text files have very limited amount of redundant bits (Sarayreh, 2014).
2. Image steganography: This method uses common and attractive cover object for hiding secret data. The attraction of this approach is the availability of a large amount of redundancy. The cover image can store various types of media files such as text, image, audio and video. This method relies on the limitation of human vision where many shades of color cannot be seen (Qasem, 2014).

3. Audio steganography: This method relies on the properties of the human hearing system, which cannot detect all frequencies of sound. It can be used to hide any type of media (Morkel, 2012).
4. Video steganography: For the most part, a video file is a collection of pictures and sounds, so the vast majority of the introduced schemes on pictures and audio can be applied to video records as well. The main advantage of video files are the large amount of information that can be hidden within the video file. The disadvantage is that video clips tend to be large, which makes it less popular for steganography (Morkel, 2012).
5. Network Protocol steganography: Embedding the data within messages and network control such as Transmission Control Protocol TCP, User Datagram Protocol UDP, Internet Protocol IP (AL Haj, 2015).

## 2.7 Steganography Categories

The steganography approach can also be categorized in its association with other data protection methods (Mishra, Mishra & Adhikary, 2014) as follows:

1. Pure Steganography (or No Key Steganography - NKS): This is the easiest form of steganography, in which the secret message is hidden in a cover image without the use of a key. The success of this hidden communication depends upon the assumption that adversaries are not aware of the presence of the secret message.
2. Secret Key Steganography (SKS): In this type of steganography the secret message is inserted into and removed out of the stego picture utilizing secret keys, both the receiver and transmitter have agreed upon these keys. The keys

can be independently shared between both sides by using some private channel before the genuine transmission begins. The quality of this framework is higher security. Parties other than the planned beneficiary cannot recover the mystery message or will require high computational time and energy to recover. The limitation of this type is the burden and effort of managing secret keys.

3. **Public Key Steganography (PKS):** This approach is similar to the SKS category, in which an encryption key is used to encrypt the secret message before the embedding. It is based on the public key method where a pair of private and public keys are utilized. The public key is used for encrypting a message while a private is used for decrypting.

## **2.8 Steganography Techniques**

There are several steganographic techniques for embedding data within cover images, which can be divided into two categories: spatial domain and transform domain.

### **2.8.1 Spatial Domain**

In this technique, the secret messages are hidden directly in the cover file without modification. The simplest method in the spatial domain is the least significant bits (LSB) method (Singla & Syal, 2012). It can be used in gray-scale and RGB images by replacing the least significant bits of a pixel with bits of the secret data. This approach relies on limitations of the human vision system, where a small change in color intensities is not easy to be noticeable.

The LSB method can change one or more of the least significant bits (bits on the right-hand side of the byte). The one-bit LSB method refers to the right-most bit of a binary sequence. A binary sequence consists 0 or 1.

The following 8-bit binary number demonstrates the one-bit LSB method:

1 0 1 1 0 0 1 **1**

By summing up all the values equal to 1 gives a result of 179. The right most bit, shown in bold text is the LSB of this particular sequence. Changing the LSB value from 1 to 0 does not have a large effect on the decimal sum, which will be 178.

LSB substitution is also suitable for Graphics Interchange Format (GIF) images, Joint Photographic Experts Group (JPEG) images, and Portable Network Graphics (PNG) images. The main advantages of the LSB technique are that the distortion of the original image is less than other techniques, and it provides large redundancy area for embedding in the cover medium.

Disadvantages of LSB technique is that during image manipulation, hidden data can be lost, and it is venerable to statistical steganalysis attacks.

There are several varieties of the LSB method regarding of the number of least significant bits per byte that will be replaced, such as 1-bit, 2-bit, 3-bit, and 4-bit LSB. Also, the cover image can consist of one byte per pixel, as in gray-scale images, or of three bytes as in RGB images. In RGB images, some stego techniques change only the LSB of the right most byte (the blue channel), while others change the LSB of every color channel (Schaathun, 2012).

## 2.8.2 Transform Domain

This technique is divided into two types: the Discrete Cosine Transform (DCT) and the Discrete Wavelet Transform (DWT) (Goel, Rana & Kaur, 2013), as follows:

1. **Discrete Cosine Transform Technique (DCT):** This type is a more complex way of hiding the secret message, and it is used for Joint Photographic Experts Group (JPEG) compression. DCT is a mathematical equation that transforms the image from spatial domain to frequency domain. It separates the image into parts (high, medium and low-frequency components) and the secret message is hidden in the least significant bit of the medium-frequency components.
2. **Discrete Wavelet Transform Technique (DWT):** It is a mathematical function that transforms an image from spatial domain to frequency domain, it includes t

## 2.9 Types of Steganalysis Attacks

Steganalysis is the science of detecting the use of steganography in stego images. The goal of steganalysis is to know if the stego image contains an embedded data or not. There are many types of steganalysis attacks (Aljarf, Amin, Filippas & Shuttelworth, 2013).

1. Visual attacks: are considered the simplest form of steganalysis. It includes examining the stego files with the naked eye to identify any noticeable distortion or by comparing the cover with the stego image to see the difference (Qasem, 2014).
2. Statistical attacks: In this type of attacks, a statistical analysis is applied to the images, using a mathematical equation, to detect the existence of hidden data. The statistical attack is similar to visual attack, but it has more detection power. The

statistical tests can find out that an image has been modified, by comparing statistical properties of the cover and stego images. Statistical attacks are classified as passive or active. Passive attacks deal with identifying the existence or absence of a secret message. Active attacks are used to look for the secret message length, hidden message location or the secret key (Devi, 2013).

3. Structure attack: In this type of attacks, the attacker may detect the existence of a hidden message by examining the structural profile of the image. These changes to the structure of the image can be detected through comparison between cover and stego images' structures, for example adding an alpha channel to an RGB BMP stego image will change the structure to 32 bits without changing the format (Devi, 2013).

## 2.10 High Capacity Hiding

The content, format and size of hidden secret messages have changed a lot recently due to the increase in using computers and smartphones in exchanging multimedia messages through electronic mails and social networks. To hide multimedia messages in cover images requires higher hiding capacity. One approach for reducing the hiding capacity requirements is to compress the data, as in the work in (Koppola, 2009). However, most multimedia files, such as audio and video, are already compressed; therefore, further compression can result in obvious sound or video distortion. Also, efficient compression techniques that are employed for images are of the lossy compression type, which might not be appropriate for many applications, such as medical or precision engineering drawings, that require a complete retention of the original message data (Morkel, 2012).

## 2.11 Security of the Hidden Data

Despite the fact that secret data hidden within steganograms are assumed to be safe from access by adversaries, however, additional measures of security protection of the hidden data have been considered, to deal with the case where the hidden data are discovered by a steganalyst (Wu & Hwang, 2007).

Several papers have proposed to combine encryption with steganography (Juneja & Sandhu, 2013). Adding an encryption key, whether using public key or private key encryptions, results in key management overhead. On the other hand, some authors considered the pure steganography approach a safe protection method because it hides the existence of a secret message; therefore, it does not encourage attacks (Rodrigues, Rios & Puech, 2004). However, since steganalysis is becoming more sophisticated in detecting hidden messages, additional security is required to protect the hidden data if its presence is detected.

## 2.12 Steganography Using Gray-Scale Images

A considerable number of research publications in steganography and steganalysis have focused on gray-scale 8-bit depth images (one channel), using standard images from the Gonzales dataset (Gonzales, 2015), such as Lena, Mandrill, and Peppers. The continued use of gray-scale 8-bit images can be due to research tradition, availability of previous work to compare with, and technical simplicity. In a recent paper (Kamaldeep & Yadav, 2015) a model called "LSB-S" is presented that hides 2-4 bits in LSB part of gray-scale images. Although the experimental model has shown good PSNR values, however hiding capacity limitation of the gray-scale image undermines its practical application.

Lee and Chen (2000) presented an image steganographic model that aimed to maximize the embedding capacity of an image using a variable-sized LSB scheme and to maintain image quality. In the proposed model, estimation of the embedding capacity of each pixel is calculated by taking into consideration contrast and luminance characteristics of the image. The shown experimental work demonstrated an effectiveness of the proposed model in providing high hiding capacity (4 bpp in one channel) while maintaining visual image quality. The main disadvantage of using gray-scale images in steganography is that it will be more likely to be suspected of being a stego, as nowadays color images are more often exchanged in private and business communications rather than black and white or gray images.

## 2.13 Related Work

Manjula and AjitDanti (2015) presented a method to conceal a secret image into a color RGB cover image by using hash based LSB technique in which 2-3-3 bits are replaced in the red, green and blue channels. The cover image is in the JPG format, and the experimental work has utilized the Lena color images of resolutions 400x400 and 580x580, as covers. The results show better PSNR results in comparison with a 3-3-2 LSB method.

The thesis by Qasem (2014) presented two models based on the spatial domain, by extending the LSB method to store 4 LSB bits in each color byte of the RGB channels, thereby crossing the traditional limit of 3 bits that is considered as the limit of unnoticeable change to a color channel. It has presented two algorithms, the first algorithm called (Embed-All) which stores the hidden image in the RGB channels of successive pixels (odd and even pixels); it gives a hiding capacity of 50% of the available pixel capacity. The



second algorithm is called (Embed-Odd) which stores the hidden image in the RGB channels of the odd pixels, while changing the RGB channels of even pixels by adding or subtracting the difference between the secret image's half-byte, and the LSB half-bytes of the odd pixels. The purpose of this change is twofold, to neutralize the color change in the odd pixel, and to add noise to the even pixel to confuse the attacker. These algorithms were implemented in Matlab 2012b, and used standard images such as Lena as cover, as well as other cover images that were used in a previous study but were not of the standard type. For secret images, the choice was for JPG images of various sizes, up to the maximum hiding capacity of the cover images. The reported result does not show any noticeable difference to the human visual system, even for the successive pixels method (Embed-All). This thesis used the image comparison metric PSNR, which has shown acceptable distortion values even when hiding in up to 50% of pixel capacity of an image.

Por et al (2013) proposed a new algorithm using the sequential color cycle to optimize the current LSB mechanism by utilizing and integrating stego 1-LSB to stego 4-LSB using a color cycle LSB model. The proposed scheme can encode up to four LSBs in the each of the RGB pixels according to the contents of the secret data without visually degrading the stego-image. The study presents a multi-embedding feature that involves hiding the secret data into several layers of covers, thereby creating a stealth camouflage to avoid an intruder's unwanted attention. The scheme implemented bit substitution using sequential color cycle algorithm to ensure the capacity of stego images remain unchanged despite having multiple layers of encoding and decoding.

The paper by Sajedi (2012) proposed a technique based on batch steganography, which is hiding secret data in more than one cover image. This paper proposed a new adaptive batch steganography approach for hiding a large secret data in multiple cover images by defining and using the steganography capacity of images. Images have various properties due to their different contents, Therefore, for a certain size secret data they could result in stego images with an unequal degree of imperceptibility. This paper proposed a novel approach to estimate the steganography capacity of images based on a signature of clean images, which is achieved by analyzing the similarity between features of cover images. In this regard, fuzzy evolutionary algorithm is employed to formulate fuzzy if-then rules based on features and signatures of clean mages. After discovering the signature of clean images, the steganographer can choose the proper cover images from the database. A proper cover image is the one in which effective features do not deviate from the signature of clean images after embedding. According to the obtained results, the proposed approach reduces the detection rate of steganalyzers compared to the traditional use of steganography methods. The advantage of the proposed scheme is its adaptability when new steganalyzers are introduced. The fuzzy rule-base can be upgraded and thus the signature of clean images can become more trustable.

Ghosal (2011) proposed a steganographic technique to hide information within the spatial domain of the 24-bit color image. The proposed steganographic technique embeds secret bits in the green and blue channels of an RGB image, while using the red channel as a guide to help determine where to embed in the other channels. The number of zeroes and ones in each pixel's red channel are calculated, and the absolute difference between the number of zeros and ones is divided by two, which is the number of embedding channels.

The resultant number is used as the number of bits to be hidden in the LSB bits of the green and blue channels' bytes (in bit positions b0-b3) of each pixel of the cover image. In the extraction phase, the hidden data from the green and blue channels are extracted by using the red channel to determine the location of the secret bits to be recovered. Experimental results are presented which show that the proposed technique has improved the hiding capacity of data (text as well as image) and at the same time retained good visual clarity of the stego-image. The average data hiding capacity is 2 bits in each of the green and blue channels, which gives 4 bits per pixel. The paper presented results of embedding a secret image of size 31.7 kb in a set of BMP images of 1024x1024 dimensions, which showed a good visual similarity between cover and stego images. The PSNR value was about 47-56 dB, which is quite high but it is understandable considering the small secret image size.

Gutub (2010) presented an LSB-based hiding technique for RGB images, by storing in the green and blue channels. The red channel was excluded from being used for embedding, as the red color has a higher frequency, hence changes in it can be more noticeable than in the green and blue channels. However, the red channel is used as an indicator for selecting which channel to embed in, and the number of bits to be embedded in the green and blue channels. The indicator value represents the 2 LSB bits of the red channel, so that it can be one of four values (00, 01, 10, 11). The average hiding capacity per pixel is 2 bpp, and it varies depending on the color distribution of the cover image. The disadvantage of this scheme is the limited hiding capacity. However, the proposed system has an added security feature by having random embedding which depends on the value of the red channel, a random value by itself.

The thesis by Koppola (2009) proposed a technique for hiding a color image (secret object) in another color image (cover object), where both images might be of the same size, therefore achieving up to 100% payload. It is based on one of the popular and simple LSB substitution techniques. It was extended to take into account the alpha channel in RGBA images, which are used as cover images. Also, a transformation function based on the conversion from RGB color space into YIQ color space was used to reduce the size of the secret image before embedding. Combining those techniques allowed achieving the initial objectives of providing a way to embed a large amount of secret data while maintaining imperceptibility. The main disadvantage of this work is that the adopted process of embedding involved changing the color model, hence resulting in lossy compression, which is a problem if the original embedded image is needed to be retrieved without modification. This thesis performed four types of comparison; the first one was used to compare the present algorithm with S-Tools algorithm through the amount of data that can be hidden. The second and third comparisons were made by using the statistical attack; it shows that it is hard to distinguish between the cover object and the stego object, when calculating the Euclidian distance and the brightness information. Finally, the last comparison used the PSNR value, which indicated that changes in the stego-image using this model produced acceptable PSNR values.

Table 2.1 summarizes the main features of the and drawbacks of the related work.

**Table 2.1: Summary of the Related Work's Features and Drawbacks**

<b>Papers</b>	<b>Features and Benefits</b>	<b>Drawbacks</b>
Manjula and AjitDanti (2015)	<ul style="list-style-type: none"> <li>- Stores secret data in all RGB channels.</li> <li>- Stores in 2-3-3 LSB (better than 1 LSB).</li> </ul>	<ul style="list-style-type: none"> <li>- The cover is compressed (JPG), which results in lower PSNR due to more distortion.</li> </ul>
Qasem (2014)	<ul style="list-style-type: none"> <li>- Stores in all channels using a 4-bit LSB technique.</li> <li>- Color adjustment of even pixels based on change to odd pixels, to neutralize change in the odd pixels, and to be a decoy against hacking.</li> </ul>	<ul style="list-style-type: none"> <li>- The color adjustment of even pixels resulted in lower PSNR than embedding in all pixels.</li> </ul>
Por (2013)	<ul style="list-style-type: none"> <li>- Stores in 1, 2, 3, 4 LSBs depending on secret size.</li> </ul>	<ul style="list-style-type: none"> <li>- Secret data size is not available in the published paper.</li> </ul>
Ghosal (2011)	<ul style="list-style-type: none"> <li>- Stores in green and blue channels using random selection.</li> <li>- The selector is based on the difference between the number "1" and "0" bits in the red channel.</li> <li>- The number of bits stored in both channels is either 0, 1 or 2.</li> <li>- Avoid storing in the red channel, as it is more sensitive to change, due to its higher frequency.</li> </ul>	<ul style="list-style-type: none"> <li>- Hides images only.</li> <li>- Low hiding capacity, on average two bits per RGB pixel.</li> </ul>
Gutub (2010)	<ul style="list-style-type: none"> <li>- Stores in the green and blue channel using random selection.</li> <li>- The 2-bit LSB of the red channel is used as an indicator to determine the number of bits to be stored in the green and blue channels, either 0, 1 or 2 bits per channel.</li> <li>- Avoid storing in the red channel, as it is more sensitive to change due to</li> </ul>	<ul style="list-style-type: none"> <li>- Hides secret images only.</li> <li>- Low hiding capacity, average two bits per RGB pixel.</li> </ul>

	higher frequency.	
Koppola (2009)	<ul style="list-style-type: none"> <li>- Uses RGBA (32 bit) images as covers.</li> <li>- Converts the 24-bit color channels of the secret file into 13 bits using the YIQ color model.</li> <li>- Stores the compressed 13-bit color data in the four channels of the RGBA cover.</li> <li>- Achieves 100% hiding capacity.</li> </ul>	<ul style="list-style-type: none"> <li>- Hides images only</li> <li>- Uses lossy compression, which is unacceptable in applications that require complete recovery of the secret data.</li> </ul>
Sajedi (2012)	<ul style="list-style-type: none"> <li>- Stores a large image in multiple covers.</li> <li>- Select cover images that can avoid detection based on features of clean images extracted through training</li> </ul>	<ul style="list-style-type: none"> <li>- Hides images only</li> <li>- Uses gray-scale images only.</li> <li>- Partial secret data can be extracted if its existence is detected.</li> </ul>

## Chapter Three

### The Proposed Model

#### 3.1 Methodology Approach

The methodology approach adopted in this thesis is experimental. This chapter presents the design of the DuoHide data hiding model which is aimed to embed and retrieve multimedia files within pairs of RGB images. The proposed model is implemented in the Matlab environment, to provide a working model for evaluating the proposed functionalities. Implementation of the proposed model is divided into two main modules: Embed, to store a secret file in two images, and Extract, to retrieve the hidden image without alteration.

#### 3.2 The Proposed Model's Required Features

The aim of the proposed model is to provide a data hiding scheme for secret multimedia files that meets the following criteria:

1. Allows high capacity hiding by splitting the secret file into two parts and storing the parts inside two RGB cover images such that the resulting stego images are sent separately over different communication channels, to avoid capture of the whole document by an adversary.
2. Splitting of the secret file should result in parts that are incomprehensible if a stego image that carries part of the secret file is captured and the hidden part of the secret file is uncovered.

3. The resulting stego images should meet un-detectability evaluation criteria such as the image quality metric of PSNR, and visual imperceptibility.
4. The secret file should be recovered unchanged, neither compressed nor altered in any way, and that a comparison between the secret and recovered data should show zero differences.
5. If the secret multimedia file is a compressed file, it should not be uncompressed during the process of embedding and extracting. Also, if the secret file is uncompressed, it should not be compressed during the embedding and extracting process.

### 3.3 Design Considerations

The proposed model aims to meet the required features mentioned in section 3.2, and to provide the essential functions for hiding a multimedia file, extracting the hidden file, and producing the necessary imperceptibility evaluation details.

The following design factors have been taken into consideration:

1. Dual RGB cover images are used to store the hidden secret multimedia files, such that each stego image will carry 50% of the hidden data. It should be possible to hide the two halves of a secret file in two similar cover images, using either the same cover image twice, or using two different images of equal dimensions.
2. The secret multimedia file is read as a stream of bytes, regardless of its format, and it should not be uncompressed if it was a compressed file.



3. The secret multimedia file will be split vertically into two parts, i.e. each byte is divided into two half-bytes and, each half-byte is hidden in a different cover image, the LSB half-byte can be stored in stego1 and the MSB half-byte in stego2.
4. The secret data half-bytes are embedded in the red, green and, blue channels of the stego images using a 4-bits LSB technique, whereby the secret half-byte replaces the LSB half-byte of the color channel.
5. Extraction of the hidden file from the two stego images should result in a file that is identical to the original secret file.
6. The PSNR results of comparing the cover image(s) with the two stego images should be identical to results produced by an acceptable standard image comparison software such as ImageMajick (available: [www.imagemajick.org](http://www.imagemajick.org)).
7. The RGB BMP cover image should be near equal in size to the secret file, where the cover size should exceed the secret file size by 54 bytes only, which is the BMP file header size. The Hiding Capacity (HC) within a cover with equal size to the secret file will be sufficient to store half of the secret file, using the 4-LSB technique. The hiding capacity of each cover is calculated as:

$$HC = \text{Width} \times \text{Height} \times 3 / 2.$$

For example, a cover image of 1024 x 1024 dimensions will have the following hiding capacity:

Hiding Capacity = number of pixels x byte per pixel / 2 = 1024 x 1024 x 3 / 2  
 = 1,572,864 bytes.

### 3.4 Data Layout of the Cover Image and the Secret File

1. The secret multimedia file is processed as a stream of bytes; each byte consists of MSB half-byte (Left Half-Byte or LH), and the LSB half-byte (Right Half-Byte or RH) as shown in Figure 3.1

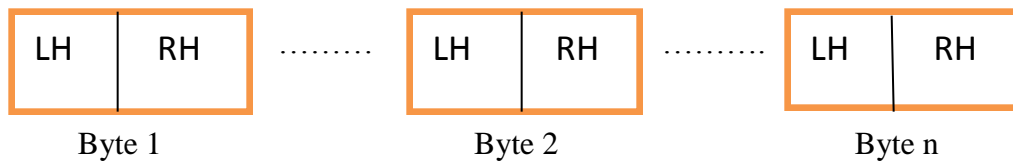


Figure 3.1 Reading Multimedia Files as a Stream of Bytes

2. The dual cover images are in RGB BMP format where each pixel is stored in 24 bits, 8-bits per color channel. The right half of each color channel (the LSB half-byte) is replaced with a half-byte from the secret file as shown in Figure 3.2

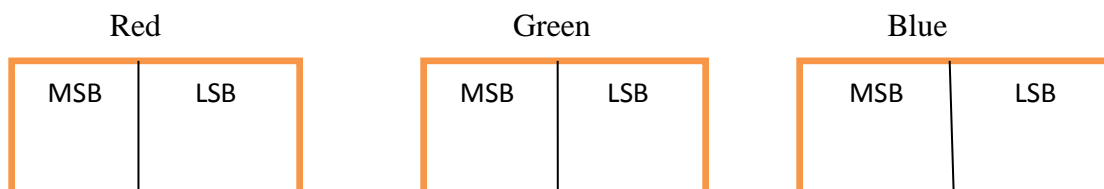


Figure 3.2 Cover Images in RGB Format

### 3.5 Data Structures

The following data structures are used in the Extract and Embed modules:

1. FullBytes: a one- dimensional array of bytes to store the secret file's byte-stream.
2. HalfBytes: a one-dimensional array of bytes to store half-bytes (LSB and MSB) of the secret file.
3. Cover1 and Cover2: a two-dimensional array (Width x Height) whose elements are pixels that consist of three bytes for the R, G, B channels.
4. Stego1 and Stego2: same as cover1 and cover2. Initially, stego1 and stego2 contain a copy of cover1 and cover2 data. During the embedding process, LSB half-bytes of stego1 and stego2 are replaced with half-bytes from the secret file.
5. ExtractedBytes: a one-dimensional array of bytes that receives recovered secret bytes during the extraction process.

## 3.6 Processing Steps of the DuoHide Model

### 3.6.1 Embedding Steps:

#### A- The Main Embedding Algorithm

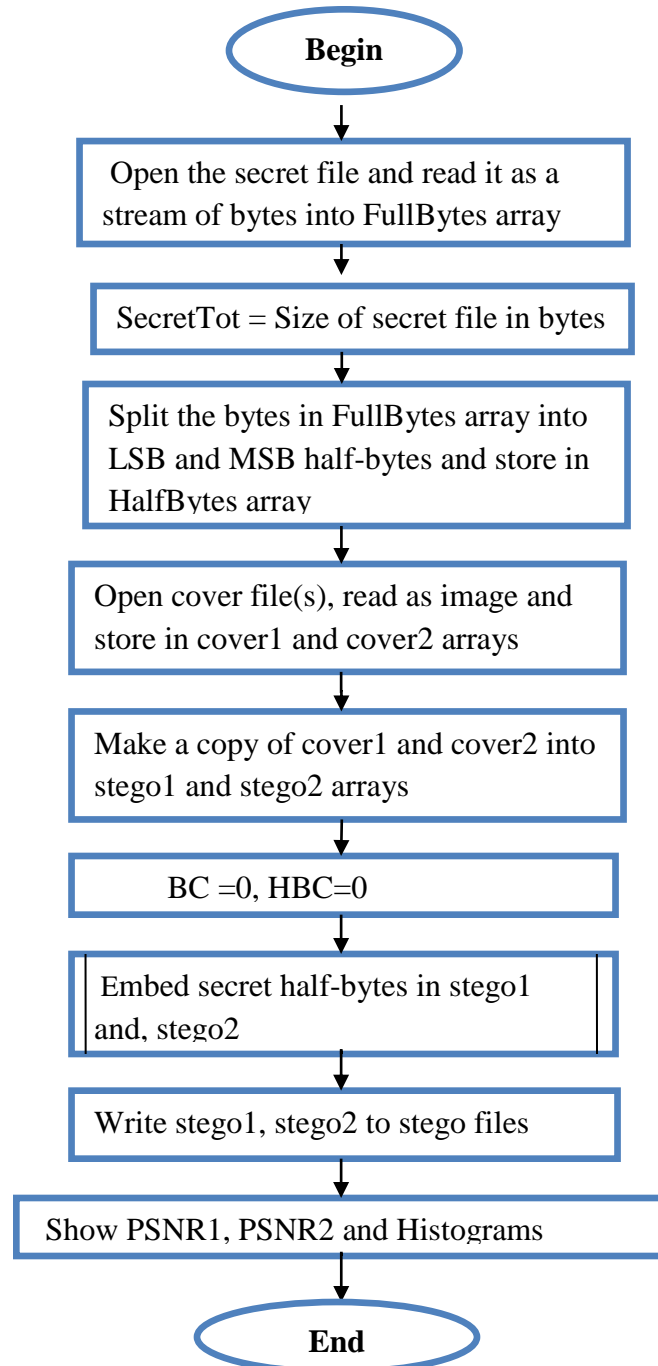


Figure 3.3 The Main Embedding Algorithm

## B- Embed Half-Bytes Sub-Algorithm

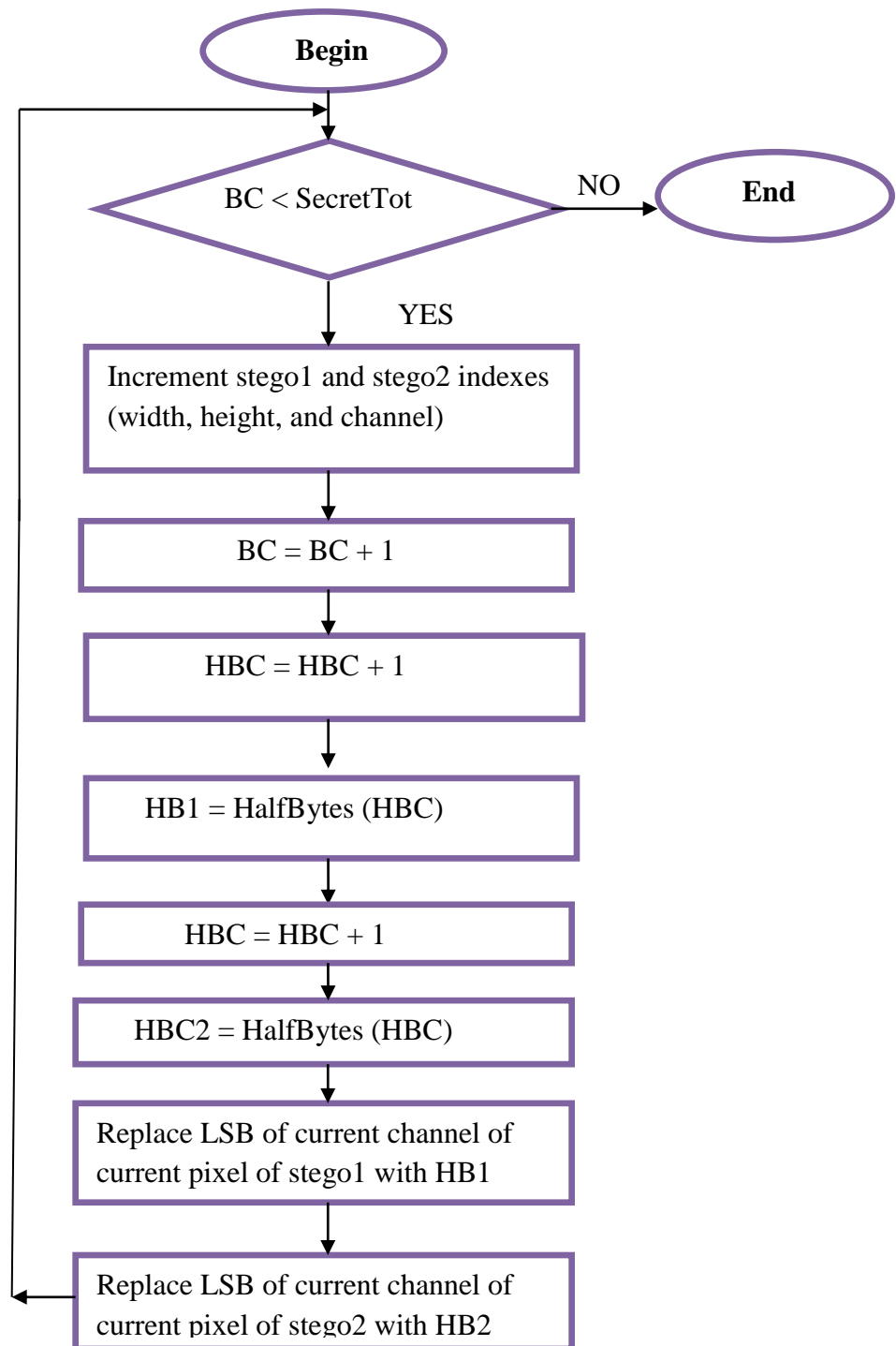


Figure 3.4 Embed Half-Bytes Algorithm

### 3.6.2 Extraction Steps

#### A-The Main Extraction Algorithm

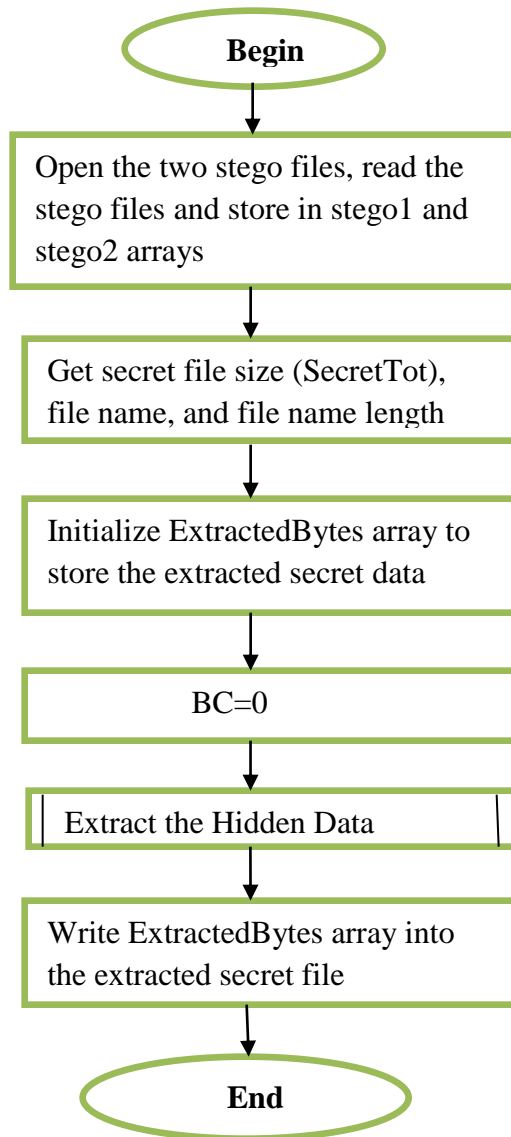


Figure 3.5 The Main Extraction Algorithm

## B-Extract the Hidden Data Sub-Algorithm

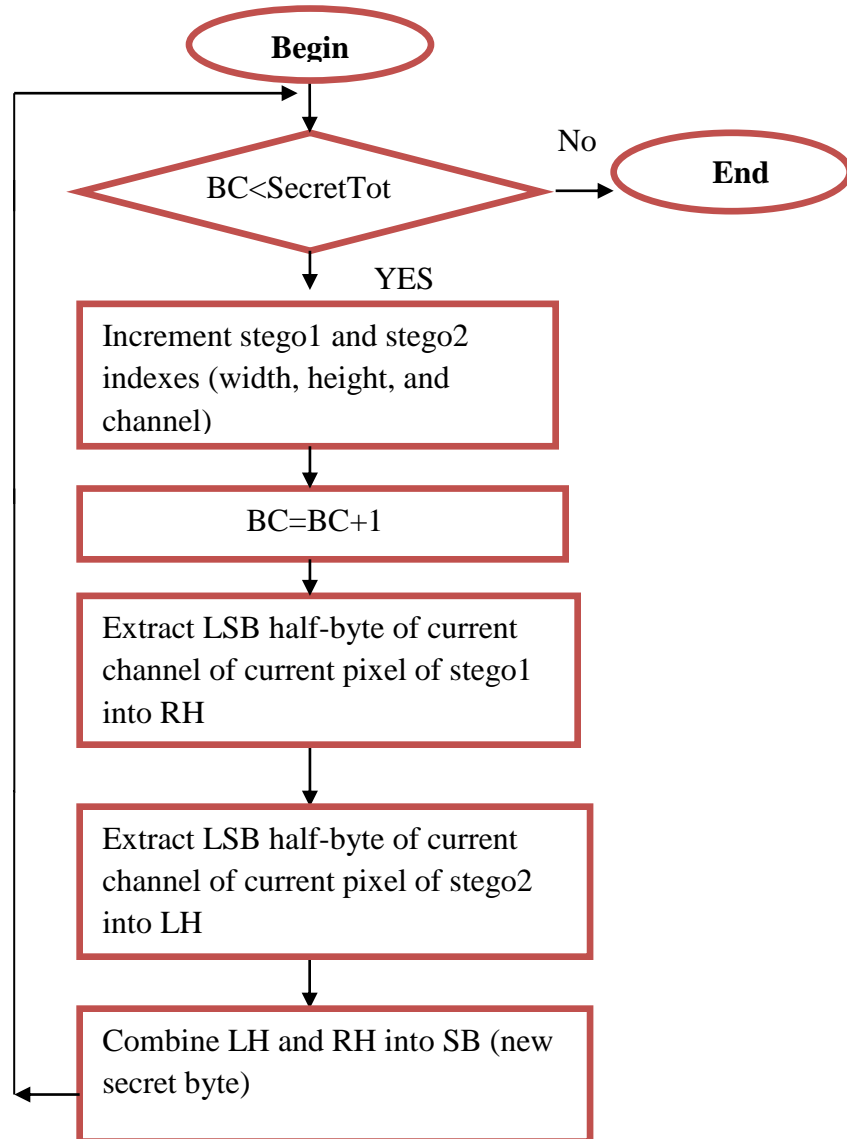


Figure 3.6 Extract the Hidden Data Sub-Algorithm

### 3.7 Illustration of the Embedding and Extracting Procedure



Figure 3.7 Example of the Embedding and Extracting Procedure



## Chapter Four

### Experimental Work and Discussion of Results

#### 4.1 Overview

This chapter presents a discussion of the experimental work results and comparison with previous work. The results have been obtained using the DuoHide steganography model and its implementation in the Matlab environment, as discussed in chapter three. The experimental work used uncompressed RGB cover images of the BMP format, in various sizes, according to requirements of each embedding case. The secret multimedia files that were embedded in the cover images include JPG, PNG, BMP, TIF and GIF images, as well as audio MP3 and video MP4 and WMV files. The set of cover images and secret multimedia files, and their sources are listed in Appendix A (DuoHide Dataset Sources).

The proposed model relies on the least significant bits (LSB) method by hiding secret data in 4-LSB bits of each of the RGB channels, in two cover images of equal size and type, with the aim of achieving high hiding capacity as well as to enhance security. The secret file is split vertically, i.e. each byte is separated into two halves, where the two halves are hidden in different covers, to protect the secrecy of the hidden data, in case it is recovered by an attacker.

Discussion of the various experimental results is presented in the following sections.

## 4.2 Embedding / Extracting Results of Multimedia Files in Large Images

In this experiment, we have used a collection (dataset) of secret multimedia files ranging in size from 4.8 KB to 9.01 MB, of various types. Sources and description of the secret multimedia files and the cover images are detailed in Appendix A. The cover images are large BMP images to accommodate the largest secret multimedia file of our selected dataset.

The visual imperceptibility comparison is demonstrated using a set of cover and stego images, for selected secret files of large, medium and small sizes, as shown in Figures 4.1 to 4.6.

Figure 4.1 shows the cover image Labelle.BMP (9.11 MB, dimensions 1500 x 2123) and the two stego images stego1, and stego2. The PNG photo secret file, Krokussen.PNG (9.01MB), was embedded in the two stego images. The resulting PSNR values are 31.9861 (cover with stego1) and 31.9863 (cover with stego2).



Figure 4.1 Labelle Cover + Krokussen.PNG with Stego1 and Stego2

Figure 4.2 shows the cover image Labelle.BMP (9.11MB) and the two stego images stego1, and stego2. The secret image file, Xynthia\_animated.GIF (3.48MB), was embedded in the two stego images. The resulting PSNR values are 36.3457 (cover with stego1) and 36.3321 (cover with stego2).



Figure 4.2 Labelle Cover + Xynthia\_animated.GIF with Stego1 and Stego2

Figure 4.3 shows the cover image Labelle.BMP (9.11MB) and the two stego images stego1, and stego2. The secret image file Renoir4-128.JPG (4.8 KB), was embedded in the two stego images. The resulting PSNR values are 65.1278 (cover with stego1) and 64.9055 (cover with stego2)



Figure 4.3 Labelle Cover + Renoir4-128.JPG with Stego1 and Stego2

Figure 4.4 shows the cover image Poppies.BMP (8.28 MB) and the two stego images stego1, and stego2. The secret image file BeethovenNo9.MP4 (8.01 MB), was embedded in the two stego images. The resulting PSNR values are 31.4487 (cover with stego1) and 31.4468 (cover with stego2).



Figure 4.4 Poppies Cover + BeethovenNo9.MP4 with Stego1 and Stego2

Figure 4.5 shows the cover image Poppies.BMP (8.28 MB) and the two stego images stego1, and stego2. The secret image file Xynthia\_animated.GIF (3.48MB), was embedded in the two stego images. The resulting PSNR values are 35.5632 (cover with stego1) and 35.5587 (cover with stego2).



Figure 4.5 Poppies Cover + Xynthia\_animated.GIF with Stego1 and Stego2

Figure 4.6 shows the cover image Poppies.BMP (8.28MB) and the two stego images stego1, and stego2. The secret image file Renoir4-128.JPG (4.8 KB), was embedded in the two stego image. The resulting PSNR values are 64.0469 (cover with stego1) and 64.0583 (cover with stego2).



Figure 4.6 Poppies Cover + Renoir4-128.JPG with Stego1 and Stego2

### 4.3 Embedding / Extracting Results of Image Files in Standard Test Images

In this section, we present results of embedding in standard test images from the Gonzales (2015) standard test images dataset. The embedded images vary in size from very small (4.8 KB) to almost the same size as the cover image. The standard image Lena.BMP with size 768 KB and 512x512 dimensions is used as a cover.

Figure 4.7 shows the cover image Lena.BMP and the two stego images stego1, and stego2. The secret image file Vase1024.JPG (490 KB), was embedded in the two stego images. The resulting PSNR values are 33.1448 (cover with stego1) and 33.1442 (cover with stego2).

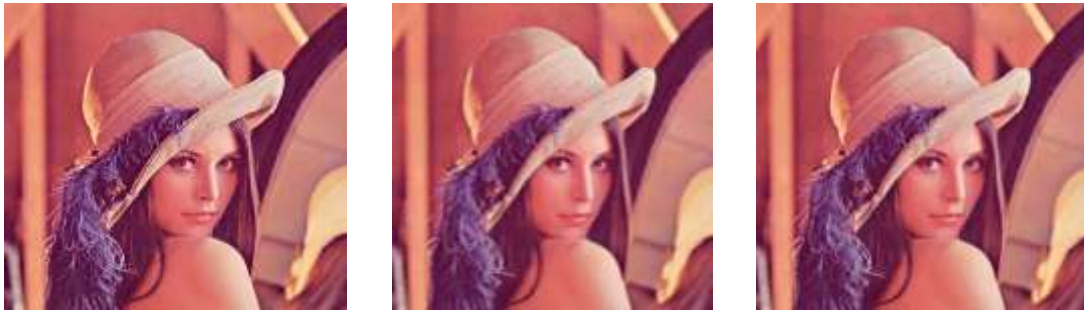


Figure 4.7 Lena Cover + Vase1024.JPG with Stego1 and Stego2

Figure 4.8 shows the cover image Lena.BMP and the two stego images stego1, and stego2. The secret image file Vase512.JPG (107 KB), was embedded in the two stego images. The resulting PSNR values are 39.8728 (cover with stego1) and 39.8741(cover with stego2).



Figure 4.8 Lena Cover + Vase512.JPG with Stego1 and Stego2

Figure 4.9 shows the cover image Lena.BMP and the two stego images stego1, and stego2. The secret image file Vase256.JPG (32.9 KB), was embedded in the two stego images. The resulting PSNR values are 44.8757 (cover and stego1) and 44.8258 (cover and stego2).



Figure 4.9 Lena Cover + Vase256.JPG with Stego1 and Stego2

Figure 4.10 shows the cover image Lena.BMP and the two stego images stego1, and stego2. The secret image file Vase128.JPG (12.5 KB), was embedded in the two stego images. The resulting PSNR values are 49.2190 (cover and stego1) and 49.0233 (cover and stego2).



Figure 4.10 Lena Cover + Vase128.JPG with Stego1 and Stego2

Figure 4.11 shows the cover image Lena.BMP and the two stego images stego1, and stego2. The secret image file Renoir4-128.JPG (4.8 KB), was embedded in the two stego images. The resulting PSNR values are 53.6699 (cover and stego1) and 53.5124 (cover and stego2).



Figure 4.11 Lena Cover + Renoir4-128.JPG with Stego1 and Stego2

#### 4.4 PSNR Imperceptibility Results

In this section, we present the PSNR values and the ratio of secret to cover size (RSC), in descending order of secret file size. The RSC metric is proposed in this work as a useful indicator of embedding capacity utilization, to be used in conjunction with the PSNR metric.

Table 4.1 shows the PSNR and RSC metrics values of embedding a group of multimedia files of various sizes, inside a 9.11 MB BMP image. There are two PSNR values for each case, PSNR1, and PSNR2, and the results show that they are very close. The minor differences between PSNR1 and PSNR2 are because the two halves of a byte have different values; hence, they have different accumulative effects on the stego images.

The results show that even when the secret to cover size is about 100%, the PSNR value is above 31 dB, which is considered acceptable regarding imperceptibility. The highest PSNR value is 65.1278 dB, for a very small secret file with a secret to cover ratio of 0.05%, which suggests that PSNR value alone, without consideration to embedding ratio, is not a sufficient factor in evaluating the effectiveness of a steganography scheme.



**Table 4.1: PSNR Results for Labelle Cover in BMP (9.11MB, dimensions 1500 x 2123, max. capacity = 9,553,500 byte) with Different Sizes of Secret Files**

Secret File	Secret File Size	Ratio of Secret to Cover	PSNR1 (cover with stego1)	PSNR2 (cover with stego2)
Krokussen.png	9.01 MB	98.90%	31.9861	31.9863
Mount-of-olives.mp4	8.51 MB	93.41%	32.0680	32.0660
BeethovenNo9.mp4	8.01 MB	87.93%	32.3222	32.3268
Time-Lapse.mp4	7.30 MB	80.13%	32.8569	32.8561
MilkyWay.wmv	6.14 MB	67.40%	33.2885	33.6805
Flower.tif	5.5 MB	60.37%	34.1284	35.3978
Xynthia_animated.gif	3.48 MB	38.20%	36.3457	36.3321
Saut.mp4	2.68 MB	29.42%	37.1968	37.1925
Elisa.mp3	1.46MB	16.05%	39.9877	40.0290
Poppies.jpg	995KB	10.67%	41.7511	41.7513
Renoir2.bmp	958 KB	10.27%	41.8279	42.6137
Vase1024.jpg	490 KB	5.25%	43.9881	43.9877
Renoir4-2048.jpg	487 KB	5.22%	44.7682	44.7764
First-day-of- spring.gif	136 KB	1.46%	50.2369	50.1603
Vase128.jpg	12.5 KB	0.13%	60.4890	60.3719
Renoir4-128.jpg	4.8 KB	0.05%	65.1278	64.9055

Table 4.2 shows values of the PSNR metric, and the ratio of secret to cover (RSC), of embedding a group of multimedia files of various sizes, inside Poppies.BMP image (8.28 MB) image. The results have similar pattern of increase in PSNR, as the ratio of secret to cover decreases.

**Table 4.2: PSNR Results for Poppies.BMP Cover (8.28 MB, dimensions 1920x1508, max. capacity 8,686,080 bytes) with Different Sizes of Secret File.**

Secret File	Secret File Size	Ratio of Secret to Cover	PSNR1 (cover with stego1)	PSNR2 (cover with stego2)
BeethovenNo9.mp4	8.01 MB	96.85%	31.4487	31.4468
Time-Lapse.mp4	7.30 MB	88.27%	31.9825	31.9845
MilkyWay.wmv	6.14 MB	74.24%	32.5450	32.9276
Flower.tif	5.5 MB	66.50%	33.3234	34.4992
Kynthia_animated.gif	3.48 MB	42.07%	35.5632	35.5587
Saut.mp4	2.68 MB	32.40%	36.4795	36.4755
elisa.mp3	1.46 MB	17.65%	39.2366	39.2605
Renoir2.bmp	958 KB	0.113%	41.1037	41.4819
Vase1024.jpg	490 KB	0.05%	43.4586	43.4580
Vase 128.jpg	12.5 KB	0.001%	59.7831	59.6101
Renoir4-128.jpg	4.8 KB	0.005%	64.0469	64.0583

Table 4.3 shows values of the PSNR metric, and the ratio of secret to cover (RSC), of embedding a group of images of various sizes, inside the standard test image Lena.BMP (768 KB) image. The results have similar pattern of increase in PSNR as the embedding ratio decreases, as noted earlier with larger covers in BMP format.

**Table 4.3: PSNR Results for Cover Image Lena.BMP (768 KB, dimensions 512x512, max. capacity = 786,432 bytes) with Different Sizes of Secret File**

Secret File	File Size	Ratio of Secret to Cover	PSNR (cover1with stego1)	PSNR (cover2 with stego2)
Kodim24.png	689 KB	89.7%	32.2337	32.2366
Kodim23.png	544KB	70.8%	33.3050	33.3104
Vase1024.jpg	490 KB	63.8%	33.1448	33.1442
Roses.jpg	349 KB	45.4%	35.1572	35.1438
Vase512 .jpg	107 KB	13.9%	39.8728	39.8741
Vase256.jpg	32.9 KB	4.2%	44.8757	44.8258
Vase 128.jpg	12.5KB	1.6%	49.2190	49.0233
Renoir4-128.jpg	4.8 KB	0.6%	53.6699	53.5124

## 4.5 Image Histogram Analysis

The histogram is a graph that represents the number of pixels in an image at each intensity value found in the image. In gray-scale images, every pixel is described by a single gray-level intensity value, which represents its shade of gray. The maximum value in the range can be up to 255, so the histogram will show 255 numbers or bins. Also, the histogram can be taken of color images that include individual histograms for the three

color channels (red, green, and blue) and brightness of each pixel in the case of RGBA images (AL Haj, 2015).

The histogram can be used to detect distortion in a stego image, by comparing it with the histogram of the original image to distinguish any change in shape. In case the original histogram is unavailable, distortion cannot be detected unless a certain uniform histogram is expected by the observer (Qasem, 2014).

Figure 4.12 shows the clean cover image of Lena.BMP (512 x 512) and its histogram, while Figures 4.13 and 4.14 show histograms of stego1 and stego2 images of Lena.BMP, each embedded with half of Kodim24.BMP. The distortion in the stego histograms is obvious because we have embedded a secret file whose size is very close to the full hiding capacity of the cover, i.e. there is a lot of distortion for the available hiding space despite the fact that the payload is shared between the two stego images (50% each).

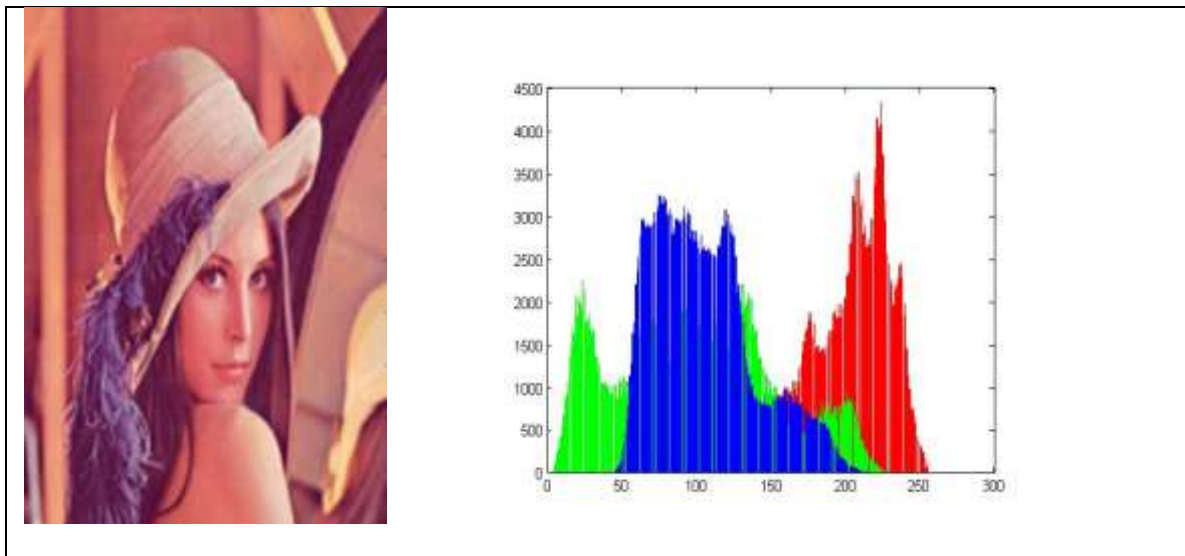


Figure 4.12 Histogram of Lena Cover

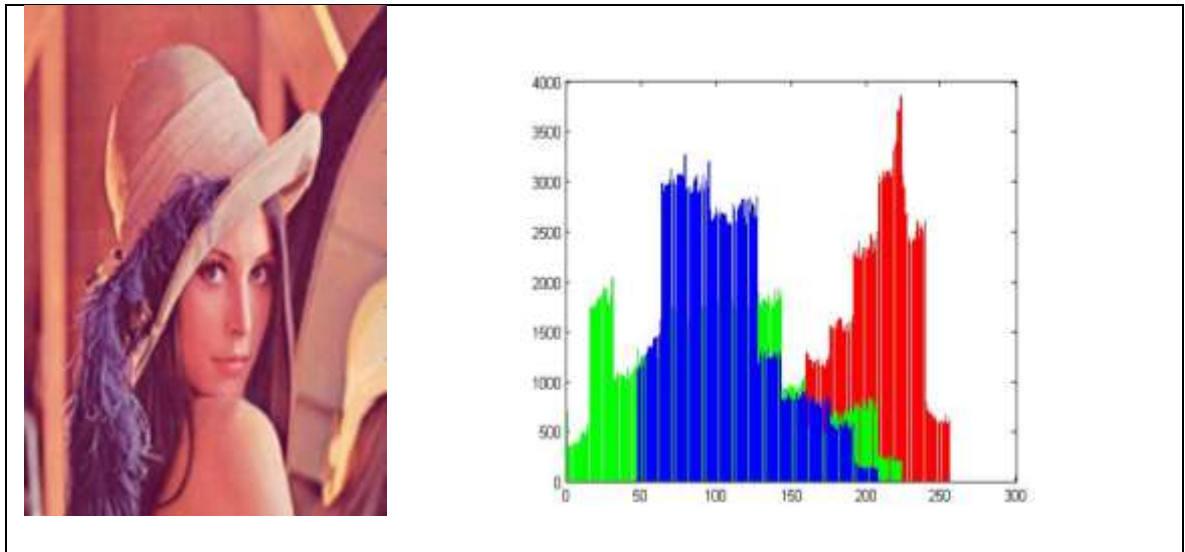


Figure 4.13 Histogram of Stego1 Lena Cover + Kodim24.PNG

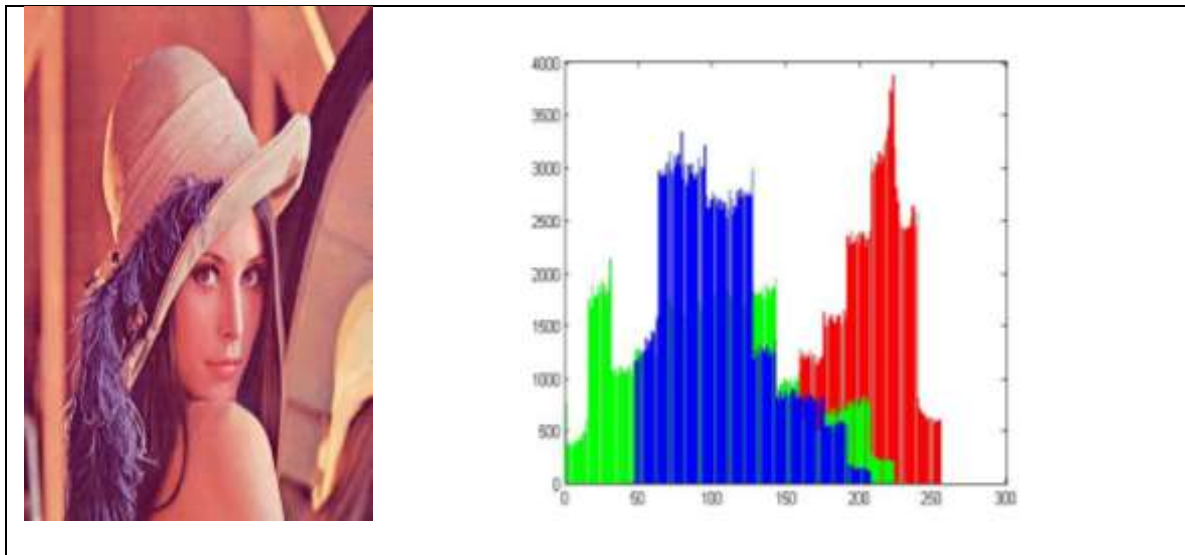


Figure 4.14 Histogram of Stego2 Lena Cover + Kodim24.PNG

Figures (4.15 to 4.17) show the histograms of cover, stego1 and stego2 for embedding a secret file of about half the cover size, and it can be seen that the change in the histograms is much less obvious than the previous case due to the lower embedding ratio as a result of the decrease in secret image size.

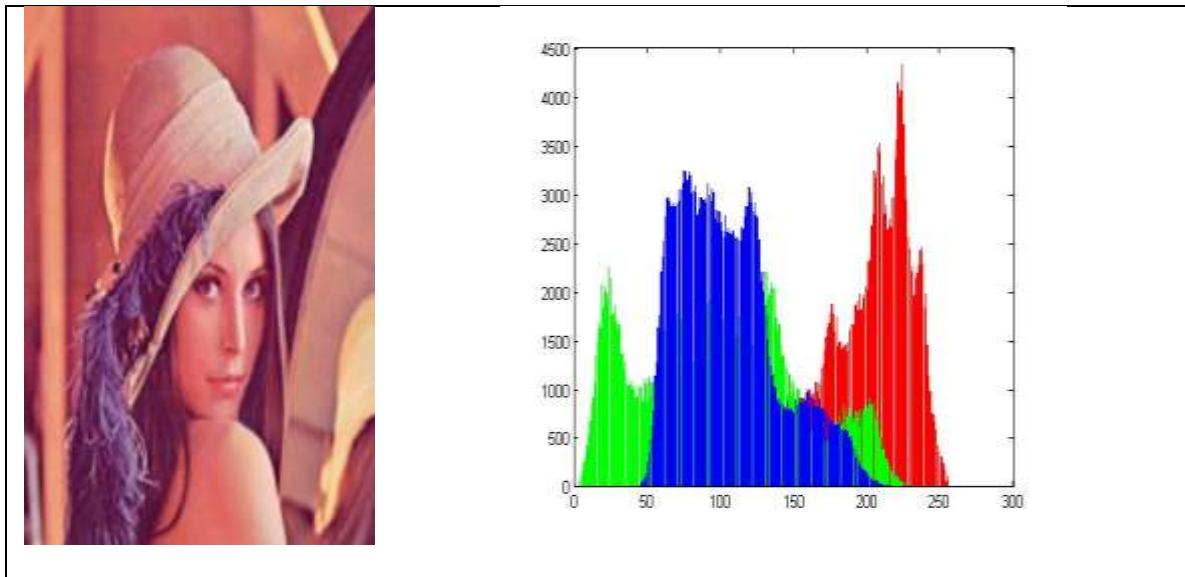


Figure 4.15 Histogram of Lena Cover

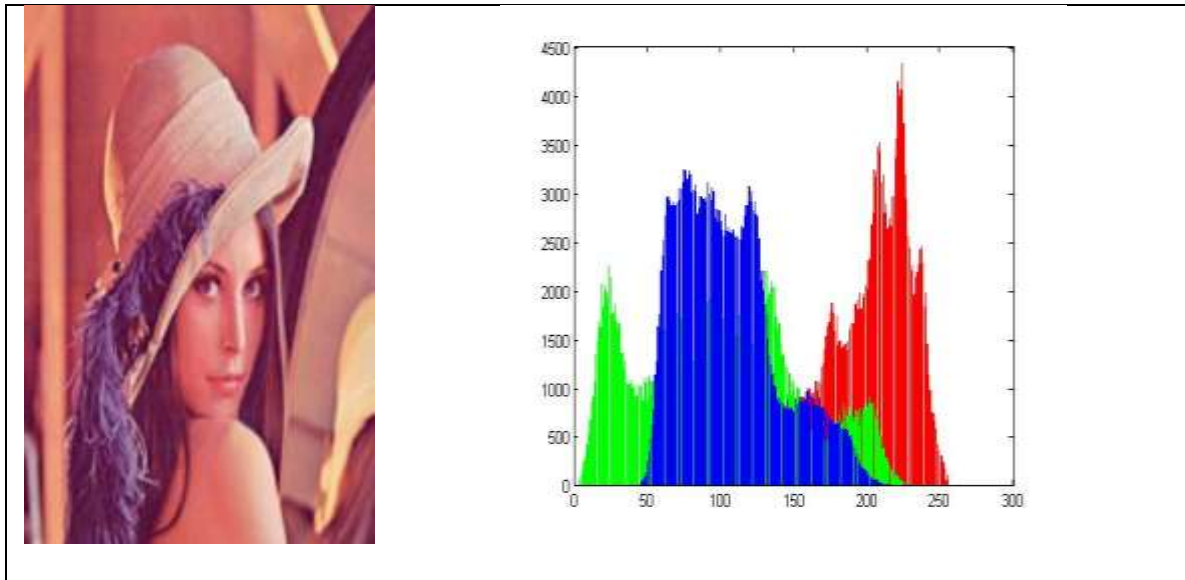


Figure 4.16 Histogram of Stego1 Lena.BMP +Roses.JPG

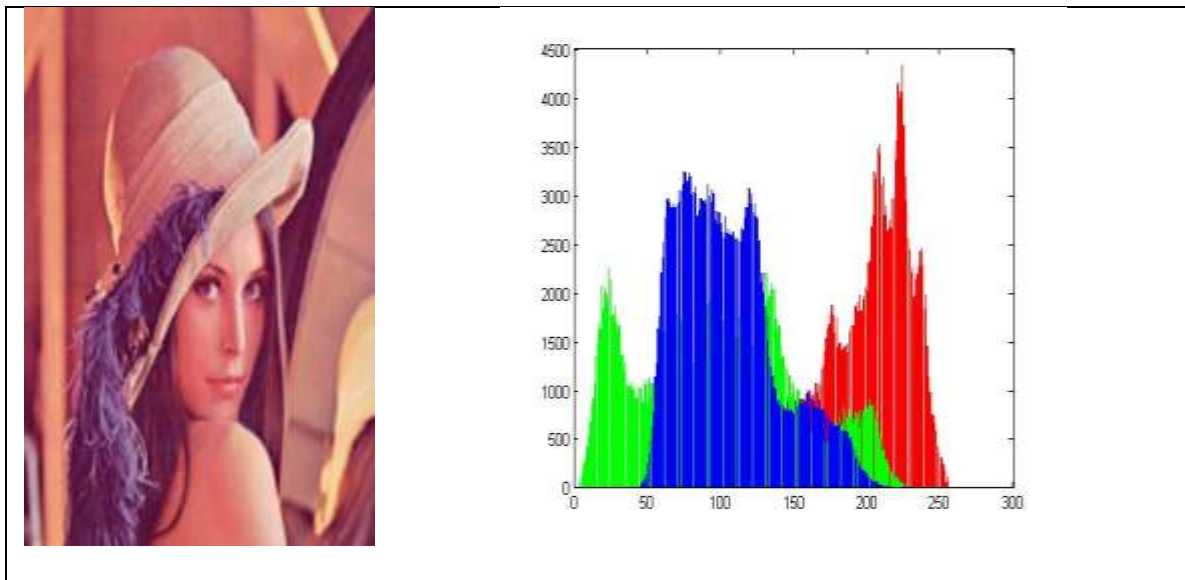


Figure 4.17 Histogram of Stego2 Lena.BMP + Roses.JPG

The last case of histogram comparison is shown in Figures 4.18 to 4.20, which represents embedding of the smallest secret file Renoir4-128.JPG (4.8 KB) in the same Lena cover image, where the secret to cover ratio is 0.6% as shown in Table 4.3 . It is clear that there is no noticeable difference between the cover and stegos histograms, because of the small secret image size. Therefore, histogram difference as a measure of distortion between cover and stego image are only noticeable when we have a high embedding ratio.

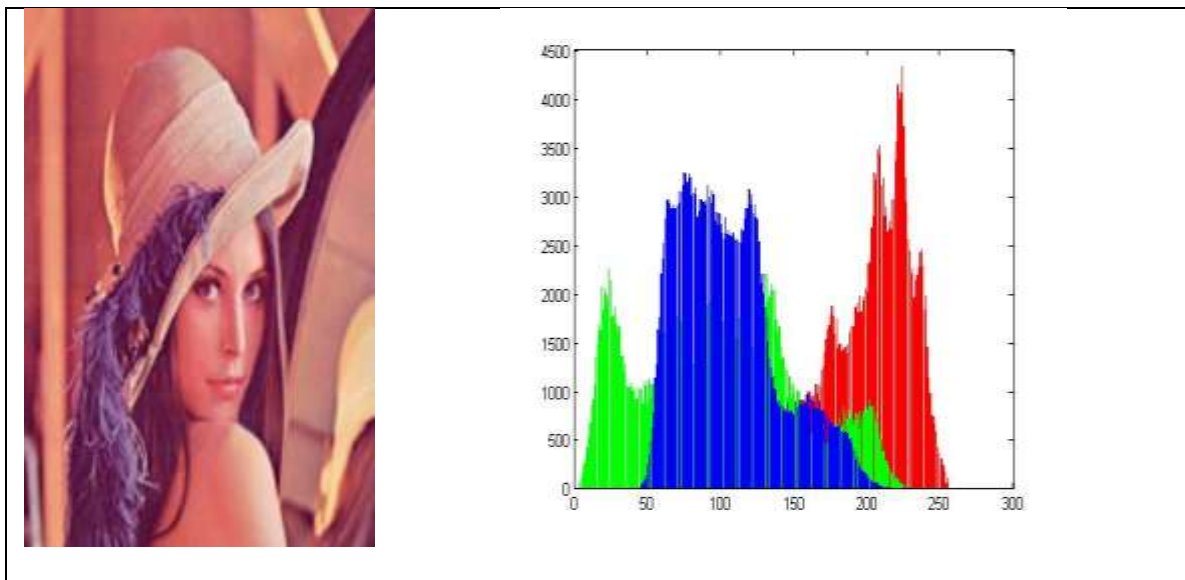


Figure 4.18 Histogram of Lena Cover



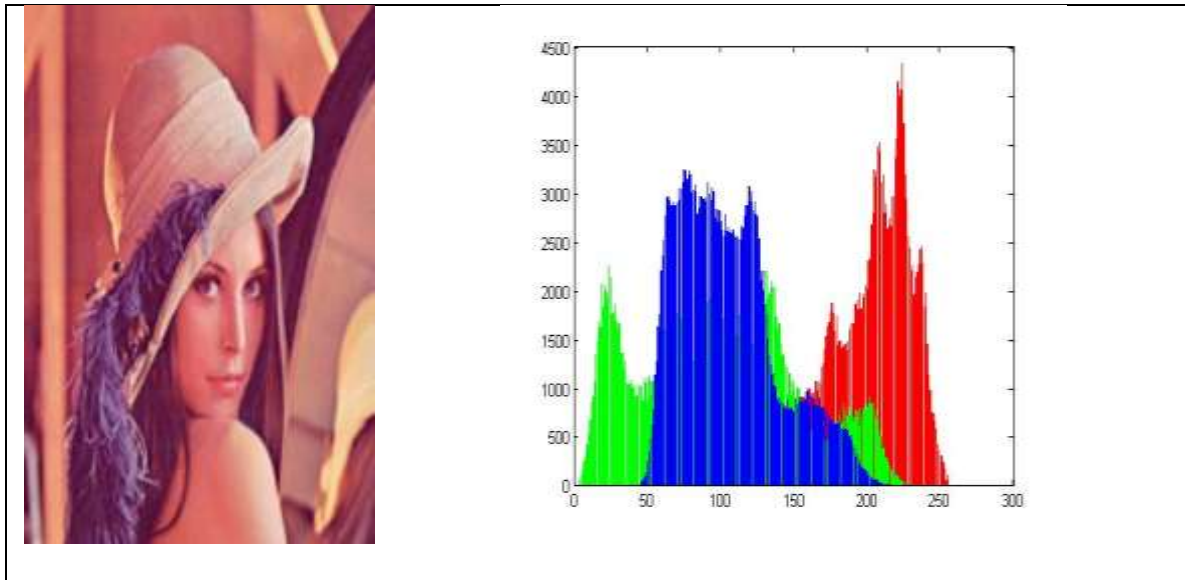


Figure 4.19 Histogram of Stego1 Lena + Renoir4-128.JPG

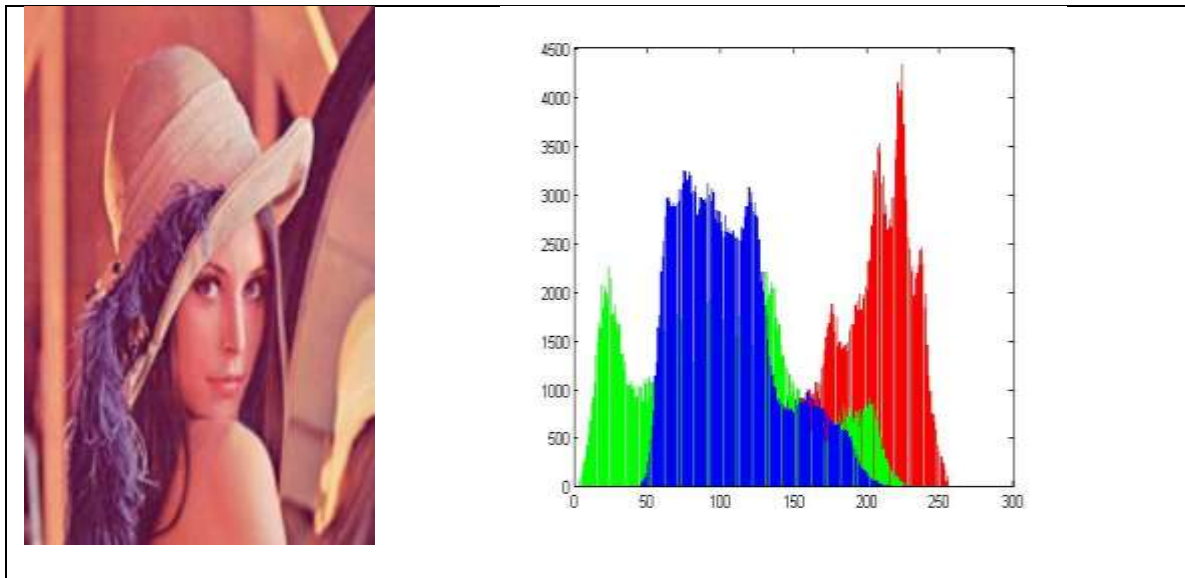


Figure 4.20 Histogram of Stego2 Lena + Renoir4-128.JPG

## 4.6 Comparison with Other Models

### 4.6.1 Comparison with a 2-3-3 LSB Model Using JPG Covers

The work in Manjula and AjitDanti (2015) presented PSNR results of hiding JPG secret images of various sizes in JPG Lena images of two resolutions; 400x400 and 580x580. For comparison with this work, we converted Lena512.BMP to 400x400 and 580x580 resolutions. The secret image used in this comparison is Lena128.JPG, which is a standard image with known size in KB. Table 4.4 shows PSNR comparison between the DuoHide model and the 2-3-3 model. The PSNR of the DuoHide is higher in both covers with a significant difference. The difference can be attributed partly to splitting of the secret image in two steps. The second reason is that the JPG cover and stego images are compressed, which can result in more differences between the two images.

**Table 4.4 Comparison of PSNR Values Between 2-3-3 LSB and DuoHide Models Using Secret Image Lena128.JPG (26.1 KB)**

Model	Cover Image	Dimensions	PSNR1	PSNR2
2-3-3 LSB	Pic400.jpg	400x400	37.6828	----
DuoHide	Lena400.bmp	400x400	44.0323	43.9993
2-3-3 LSB	Pic580.jpg	580x580	42.7804	----
DuoHide	Lena580.bmp	580x580	47.2953	47.2378

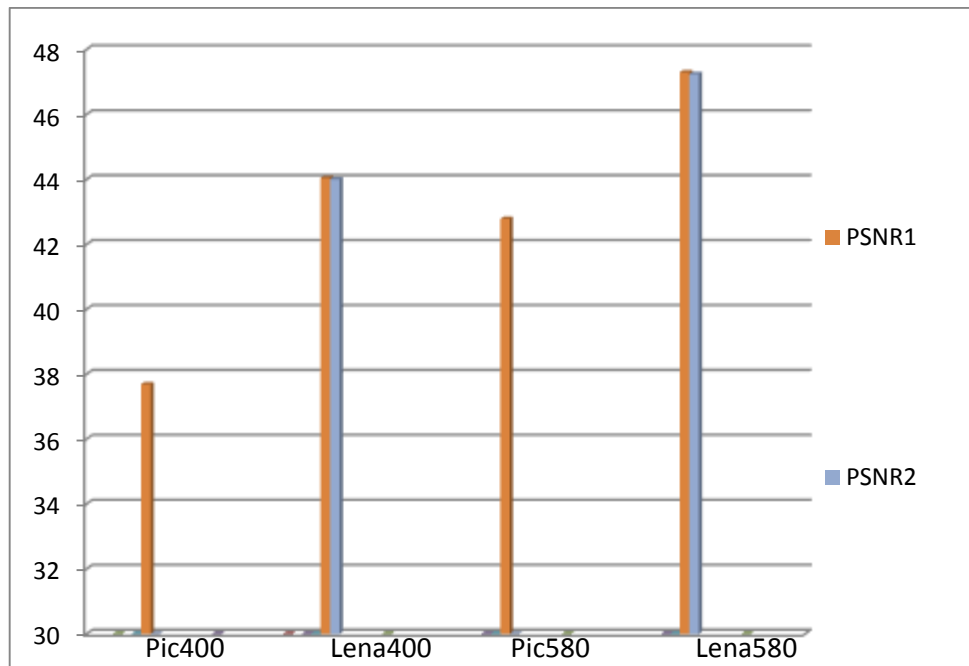


Figure 4.21 PSNR Values Between 2-3-3 LSB and DuoHide Models

## 4.6.2 Comparison Results of One Cover and Two Covers 4-bit LSB

### Embedding

To compare between PSNR results of using one cover 4-bit LSB embedding and two covers, the standard Lena512.BMP image is used as a cover for both models. A set of secret images of various sizes and types is used. Table 4.5 shows the PSNR results for both models. As the results show, the difference in PSNR values between the two models is about 3 dB, despite the fact that the stego images of the DuoHide model are hiding half of the secret file compared with the one cover 4-bit model. This confirms that the increase in PSNR value is much slower than the increase in hidden secret size. Also, the PSNR difference in this case is less than the JPG vs. covers difference that was noted in section 4.6.1

**Table 4.5 Comparison of PSNR Values Between One-Cover 4-bit LSB and DuoHide**

**Two-Covers Embedding**

Secret File	Secret File Size	One-Cover 4-LSB	DuoHide Model		PNSR Difference
		PSNR	PSNR1	PSNR2	dB
Roses.jpg	349 KB	32.1569	35.1572	35.1438	2.9936
Pump.gif	282 KB	32.8875	35.8905	35.884	2.99975
Livingroom.jpg	256 KB	34.2521	36.527	38.1065	3.06465
Apples.jpg	197 KB	34.7017	37.6925	37.6972	2.99315
Spring.gif	136 KB	35.9821	38.9713	38.9013	2.9542
kodim09-256.tif	134 KB	36.2505	39.1995	39.225	2.96175
Lena_Gray_256.jpg	64.2 KB	40.1065	42.3535	43.7724	2.95645
Lena128.jpg	26.1 KB	43.2759	46.2583	46.2396	2.97305

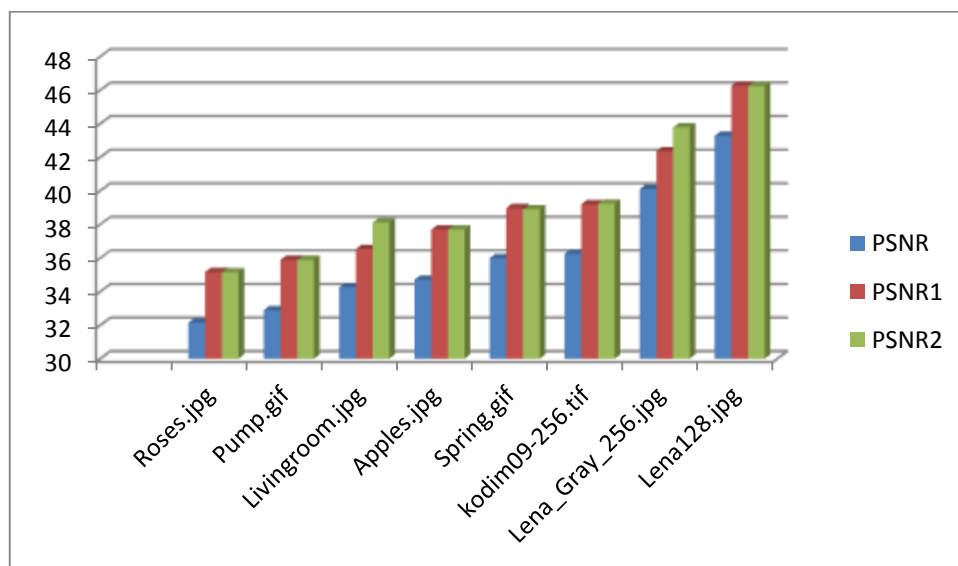


Figure 4.22 PSNR Values Between One-Cover 4-bit LSB and DuoHide Embedding

## 4.7 Comparison of PSNR Results Using Heterogeneous Cover Pairs

In the previous analyses in this chapter, the cover pairs were homogeneous, i.e. one cover is used twice, as cover1 and cover2. The shown PSNR1 and PSNR2 were very close in values, with less than 1% dB difference. In this section, we present PSNR results using heterogeneous covers, where the two covers are different images (Lena.BMP and Peppers.BMP) but they share the same size, format and dimensions (512 x 512). Both covers are standard images from the Gonzales dataset (2015). The first cover (Lena.BMP) is chosen because it was used in the results shown in Table 4.3, for comparison purpose with results in this section. The second cover (Peppers.BMP) is chosen as it has the most color variation among the Gonzales dataset, and so it has different color variation than Lena.BMP.

Table 4.6 shows the PSNR results of embedding the same secret files of Table 4.3, using two different images (Lena.BMP and Peppers.BMP), both have the size of 768 KB and 512 x 512 dimensions. Once more, PSNR1 and PSNR2 results are very close, with difference of less than 1% between them. This confirms that using the same cover image twice as dual covers, or using different cover images of equal size and dimensions, have similar results as far as the difference between PSNR1 and PSNR2.

**Table 4.6: PSNR Results for Cover Images Lena.BMP vs. Peppers.BMP (768 KB, dimensions 512\*512, max. capacity = 786,432 bytes) with Different Sizes of Secret File**

Secret File	Secret File Size	Ratio of Secret to Cover	PSNR (cover1with stego1)	PSNR (cover2 with stego2)
Kodim24.png	689 KB	89.7%	32.2337	32.1864
Kodim23.png	544 KB	70.8%	33.3050	33.2789
Vase1024.jpg	490 KB	63.8%	33.1448	33.0513
Roses.jpg	349 KB	45.4%	35.1572	35.1740
Vase512 .jpg	107 KB	13.9%	39.8728	39.9523
Vase256.jpg	32.9 KB	4.2%	44.8757	45.1716
Vase 128.jpg	12.5KB	1.6%	49.2190	49.3410
Renoir4-128.jpg	4.8 KB	0.6%	53.6699	53.5905

Figure 4.23 shows the second cover image (Peppers.BMP) and stego2 image, that is the result of embedding half of the largest secret file of the image set in Table 4.3, which is Kodim24.PNG (689 KB). As can be seen, there is no noticeable difference between the cover and stego2 images despite the 50% embedding ratio. Also, Figure 4.24 and 4.25 shows the cover and stego2 images using Roses.JPG (349) KB and Renoir.JPG (4.8 KB) secret images.



Figure 4.23 Peppers Cover with Stego2 + Kodim24.PNG



Figure 4.24 Peppers Cover with Stego2+ Roses.JPG



Figure 4.25 Peppers Cover with Stego2+ Renoir4-128.JPG

## 4.8 Verification of Results

Publically available software tools are used in support of the of the verification of the experimental outputs and results obtained in this research, using the DuoHide system, as follows:

### 1. Verification of the Extracted Secret Files Integrity

The extracted secret file is required to be identical 100% to the original secret file. The Microsoft Windows command for file comparison, FC, is used, which compares two files to verify that they are identical or not, and if they are not identical, the differences are displayed. If there are no differences the output is “No differences encountered” as shown in the sample output in Figure 4.26 below:



```
C:\DU>FC time-lapse8.22.mp4 extime-lapse8.22.mp4
Comparing files Time-Lapse8.22.mp4 and EXTIME-LAPSE8.22.MP4
FC: no differences encountered
```

Figure 4.26: Verification of the Integrity of Extracted Secret File Extime- lapse8.22.MP4

The above integrity verification step was applied to all extracted files of this research.

## 2. Verification of PSNR Results

The PSNR results of all the experiments reported in this chapter have been verified using the publically available ImageMajick software. The PSNR result of the DuoHide system for a pair of cover and stego images is compared with the PSNR result of ImageMajick. Figure 4.27 shows the PSNR comparison output of ImageMajick's COMPARE command, using the cover image Poppies.BMP and the two stego images st1poppies.BMP and St2poppies.BMP.

```
C:\DU>compare -verbose -metric psnr poppies.bmp st1poppies.bmp :null 2>&1
poppies.bmp BMP3 1920x1508 1920x1508+0+0 8-bit sRGB 8.686MB 0.031u 0:00.032
st1poppies.bmp BMP3 1920x1508 1920x1508+0+0 8-bit sRGB 8.686MB 0.016u 0:00.019
Image: poppies.bmp
Channel distortion: PSNR
red: 31.3578
green: 31.7378
blue: 31.0552
all: 31.3746
poppies.bmp=>null BMP3 1920x1508 1920x1508+0+0 8-bit sRGB 8.686MB 0.469u
0:00.298
C:\DU>compare -verbose -metric psnr poppies.bmp st2poppies.bmp :null 2>&1
poppies.bmp BMP3 1920x1508 1920x1508+0+0 8-bit sRGB 8.686MB 0.016u 0:00.015
```

```
st2poppies.bmp BMP3 1920x1508 1920x1508+0+0 8-bit sRGB 8.686MB 0.016u 0:00.015
Image: poppies.bmp
Channel distortion: PSNR
red: 31.3534
green: 31.7427
blue: 31.0582
all: 31.3758
poppies.bmp=>null BMP3 1920x1508 1920x1508+0+0 8-bit sRGB 8.686MB
0.563u 0:00.291
```

Figure 4.27: PSNR Results of ImageMajick for Verification of DuoHide PSNR Results

## Chapter Five

### Conclusion and Future Work

#### 5.1 Conclusion

The work in this thesis investigated the enhancement of securely sending multimedia files, over communication channels, through embedding them within dual RGB images. The advantages of using two RGB cover images are two-fold; higher hiding capacity, which is needed for multimedia files, and better protection of the hidden data. The proposed model (DuoHide) and its implementation have presented a solution in which the secret multimedia file is split vertically into two halves, where each half is stored in a separate cover. Using this method, one vertical half of the secret message will provide no clue to the hacker about the contents of the secret message, if she/he succeeds in capturing a stego image and extracting its contents.

The experimental results obtained from hiding a variety of multimedia files into uncompressed RGB BMP images can be summarized as follows:

1. It is possible to hide a large secret file within two uncompressed RGB covers, where the ratio of secret file to cover file sizes is  $\leq 100\%$ , and the embedding ratio in the two covers is  $\leq 50\%$  of the combined hiding capacity. To achieve maximum hiding capacity embedding, the cover file size should be chosen to be near equal to the secret file size (the difference between the secret and cover file sizes are the header size of the cover file).

2. The PSNR value is inversely proportional to the ratio of secret-to-cover sizes, therefore, if imperceptibility has higher priority over hiding capacity, a larger cover file should be used.
3. In spite of the full capacity hiding in the two covers, the resulting PSNR value was above 30 dB, which is acceptable with regard to imperceptibility.
4. Reading a secret multimedia file as a stream of bytes made it possible to read and hide all types of compressed multimedia files, without the need for uncompression, and without any change to the hidden data or its compression.
5. The secret file integrity was maintained, as the recovered file was identical to the original secret file in contents, size, and file name. The PSNR value for secret file vs. recovered secret file was infinity, which verifies that the two files are identical.

## 5.2 Future Work

Results of the experimental research work in this thesis is a small step towards secure multimedia file exchange. Several enhancements and extensions are thought to be possible and can be the subject of future research, in areas below:

1. Investigating the effect of splitting a secret file over multiple ( $> 2$ ) covers, with regard to the tradeoff between stego quality and user overhead to deal with many stegos.
2. Using less bits per byte in embedding, to improve imperceptibility.

3. Investigating the effect of using the alpha channel for embedding within a 32-bit RGBA image, where the LSB bit replacement is 3 bits per channel (12 bits per pixel), or 4 bits per channel (16 bits per channel).
4. Investigating the effect of alternating the embedding of the LSB and MSB half-bytes of the secret data between the two stegos.
5. Studying the use audio and video files as cover media, using the dual or multiple covers approach, which would provide higher hiding capacity, provided that audio or video distortion are not noticeable.

## References

- Al Haj, H.M. (2015). **Image steganography technique based on predetermined pattern and histogram analysis**, (Unpublished master dissertation), Middle East University, Amman, Jordan.
- Al-Ani, Z. K., Zaidan, A. A., Zaidan, B. B., & Alanazi, H. (2010). Overview: Main fundamentals for steganography. *arXiv preprint arXiv:1003.4086*.
- Aljarf, A., Amin, S., Filippas, J., & Shuttelworth, J. (2013). Develop a detection system for grey and colour stego images. *International Journal of Modeling and Optimization*, 3(5), 458.
- Bateman, P., & Schaathun, H. G. (2008). **Image steganography and steganalysis**, (Unpublished master dissertation), Department Of Computing, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, Surrey, United Kingdom, 4th August.
- Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P. (2010). Digital image steganography: survey and analysis of current methods. *Signal Processing*, 90(3), 727-752.
- Devi, K. J. (2013). **A sesure image steganography using LSB technique and pseudo random encoding technique**, (Unpublished master dissertation), National Institute of Technology, Rourkela.
- Doshi, R., Jain, P., & Gupta, L. (2012). Steganography and its applications in security. *International Journal of Modern Engineering Research (IJMER)*, 2(6), 4634-4638.

Efimushkina, T. (2013). **Reversible data hiding in digital images**, (Unpublished master dissertation), Tampere University of Technology.

Ghosal, S. K. (2011). A new pair wise bit based data hiding approach on 24 bit color image using steganographic technique. *Proceedings of IEMCON*, 123-129.

Goel, S., Rana, A., & Kaur, M. (2013). Comparison of image steganography techniques. *International Journal of Computers and Distributed Systems*, 3(1), 20-30.

Gonzales

Gonzales, R. C. (2008). Digital Image Processing, 3rd edition, Prentice-Hall,

Image databases (On-Line), available:

[http://www.imageprocessingplace.com/root\\_files\\_V3/image\\_databases.htm](http://www.imageprocessingplace.com/root_files_V3/image_databases.htm).

Gupta, H., Kumar, R., & Changlani, S. (2013). Enhanced data hiding capacity using LSB-based image steganography method. *International Journal of Emerging Technology and Advanced Engineering*, ISSN, 2250-2459.

Gutub, A. A. A. (2010). Pixel indicator technique for RGB image steganography. *Journal of Emerging Technologies in Web Intelligence*, 2(1), 56-64.

Hayati, P., Potdar, V., & Chang, E. (2007, July). A survey of steganographic and steganalytic tools for the digital forensic investigator. *In Workshop of Information Hiding and Digital Watermarking*, 1-12.

Hussain, M., & Hussain, M. (2013). A survey of image steganography techniques. *International Journal of Advanced Science and Technology*, 54, 113-124.

Joshi, K., & Yadav, R. (2015, December). A new LSB-S image steganography method blend with Cryptography for secret communication. *In 2015 Third International Conference on Image Information Processing (ICIIP) (pp. 86-90). IEEE.*

Juneja, M., & Sandhu, P. S. (2013). A new approach for information hiding in color images using adaptive steganography and hybrid feature detection with improved PSNR and capacity. *International Journal of Engineering and Technology (IJET) Vol 5 No 2*

Koppola, R. R. (2009). **A high capacity data-hiding scheme in LSB-based image steganography**, (Unpublished doctoral dissertation), University of Akron.

Kumar, A., & Pooja, K. (2010). Steganography-a data hiding technique. *International Journal of Computer Applications, 9(7), 19-23.*

Laskar, S. A., & Hemachandran, K. (2012). High capacity data hiding using LSB steganography and encryption. *International Journal of Database Management Systems (IJDMS), 4(6), 57-68.*

Lee, C. W., & Tsai, W. H. (2013). A data hiding method based on information sharing via PNG images for applications of color image authentication and metadata embedding. *Signal Processing, 93(7), 2010-2025.*

Lee, Y. K., & Chen, L. H. (2000, June). High capacity image steganographic model. *In Vision, Image and Signal Processing, IEE Proceedings (Vol. 147, No. 3, pp. 288-294). IET.*

Magut, S. J. (2010). **An overview of digital steganography**, (Unpublished Master dissertation), University of Colorado.



Mandal, P. C. (2012). Modern steganographic technique: A survey. *International Journal of Computer Science & Engineering Technology (IJCSET)*,3(9), 444-448.

Manjula, G. R., & Danti, A. (2015). A novel hash based least significant bit (2-3-3) image steganography in spatial domain. *arXiv preprint arXiv:1503.03674*.

Mishra, M., Mishra, P., & Adhikary, M. C. (2014). Digital image data hiding techniques: A Comparative Study. *arXiv preprint arXiv:1408.3564*.

Morkel,T.(2012). **Image steganography applications for secure communication**, (Unpublished doctoral dissertation), University of Pretoria.

Por, L. Y., Yin, D. B., Ang, T. F., & Ong, S. Y. (2013). An enhanced mechanism for image steganography using sequential colour cycle algorithm.Int. *Arab J. Inf. Technol.*, 10(1), 51-60.

Qasem, M. (2014). **Hiding secret image within RGB images using an enhanced LSB method**, (Unpublished master dissertation), Middle East University, Amman, Jordan.

Rodrigues, J. M., Rios, J. R., & Puech, W. (2004). SSB-4 System of Steganography using bit 4. *In 5th International Workshop on Image Analysis for Multimedia Interactive Services*.

Sajedi, H. (2012). RABS: Rule-Based Adaptive Batch Steganography. *Recent Advances in Steganography*, 35.

- Salih, m. (2015). **A new audio steganography method using Bi-LSB embedding and secret message integrity validation**. (Unpublished master dissertation), Middle of East University, Amman, Jordan.
- Sandilya, M., & Chawla, M. (2014). Spatial Domain Image Steganography based on Security and Randomization. *Editorial Preface, 5(1)*.
- Sarayreh, G. S. (2014). **Text Hiding in RGBA Images Using the Alpha Channel and the Indicator Method**, (Unpublished master dissertation), Middle East University, Amman, Jordan.
- Schaathun, H.G. (2012). *Machine learning in image steganalysis*, A lesund University College: Norway.
- Singla, D., & Syal, R. (2012). Data security using LSB & DCT steganography in images. *International Journal Of Computational Engineering Research, 2*, 359-364.
- Thanikaiselvan, V., Arulmozhivarman, P., Subashanthini, S., & Amirtharajan, R. (2013). A graph theory practice on transformed image: A random image steganography. *The Scientific World Journal, 2013*.
- Wiley, j., & Ltd, s.(2012). *Machine learning in image steganalysis*, (1<sup>st</sup> ed.). A° lesund University College: Norway.
- Wu, N. I., & Hwang, M. S. (2007). Data hiding: current status and key issues. *IJ Network Security, 4(1), 1-9*.

Yugala, k. (2013). Steganography. *International Journal of Engineering Trends and Technology (IJETT) Volume4Issue5*.

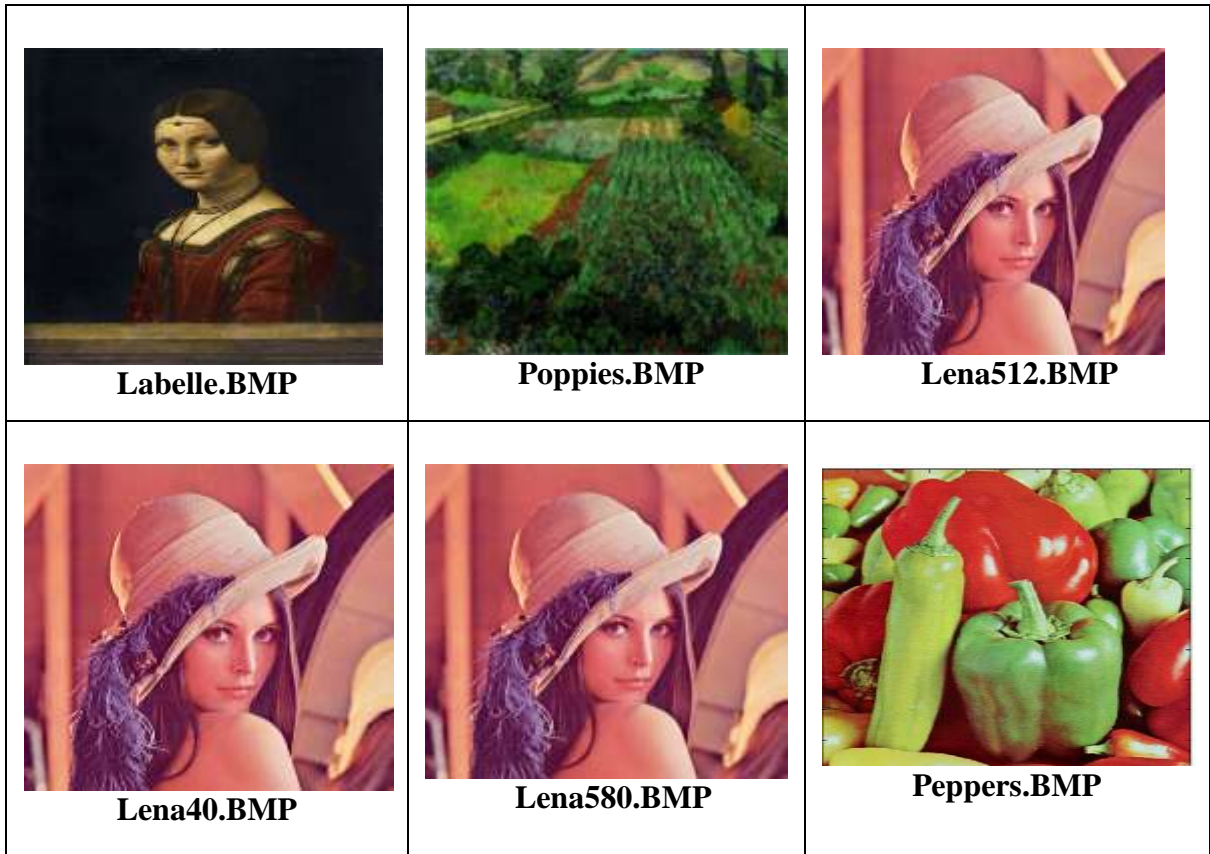
Yalman, Y., & ERTÜRK, İ. (2013). A new color image quality measure based on YUV transformation and PSNR for human vision system. *Turkish Journal of Electrical Engineering & Computer Sciences, 21(2), 603-612*.

# Appendix A

## (DuoHide Dataset Sources)

## A.1 Cover Images

File Name and Specs	Source and Details
<b>Labelle.BMP</b> 9.11 MB 24-bit 1500 x 2130	La Belle Ferroniere, Circa 1495-1499 Painting attributed to Leonardo Da Vinci Location: La Louvre Museum Downloaded in JPG format from: <a href="https://commons.wikimedia.org/wiki/File:Leonardo_da_Vinci_(attrib)-_la_Belle_Ferroniere.jpg">https://commons.wikimedia.org/wiki/File:Leonardo da Vinci (attrib)- la Belle Ferroniere.jpg</a> , Converted to BMP format using <a href="http://www.online-convert.com">online-convert.com</a>
<b>Poppies.BMP</b> 8.27 MB 24-bit 1920 x 1508	Field with Poppies, 1889 Painting by Vincent Van Gokh Location: Kunsthalle Bremen Museum Downloaded in JPG format from: <a href="https://commons.wikimedia.org/wiki/File%3AVincent_van_Gogh_-_Field_with_Poppies_(1889).jpg">https://commons.wikimedia.org/wiki/File%3AVincent van Gogh - Field with Poppies (1889).jpg</a> Converted to PNG format using <a href="http://www.online-convert.com">www.online-convert.com</a>
<b>Lena512.BMP</b> 768 KB 24-bit 512 x 512	Lena Soderberg (aka Lenna), 1972 Photograph Downloaded in TIF format from Gonzles book website: <a href="http://www.imageprocessingplace.com/root_files_V3/image_databases.htm">http://www.imageprocessingplace.com/root_files_V3/image_databases.htm</a> Converted to BMP format using <a href="http://www.online-convert.com">www.online-convert.com</a>
<b>Lena400.BMP</b> 468 KB 24-bit 400 x 400	Same as above
<b>Lena580.BMP</b> 985 KB 24-bit 580 x 580	Same as above
<b>Peppers.BMP</b> 768 KB 24-bit 512 x 512	Photograph Downloaded in TIF format from Gonzles book website: <a href="http://www.imageprocessingplace.com/root_files_V3/image_databases.htm">http://www.imageprocessingplace.com/root_files_V3/image_databases.htm</a> Converted to BMP format using <a href="http://www.online-convert.com">www.online-convert.com</a>



## A.2 Secret Multimedia Files Embedded in Labelle.BMP and

### Poppies.BMP

Secret File Name and Specs	Source and Details
<b>Krokussen.PNG</b> 9.01 MB 24-bit 2222 x 1590	Krokussen (aka Crocus tommasinianus), 2013 Photograph by Duminica Johannes Bergas Downloaded from in JPG format from: <a href="https://commons.wikimedia.org/wiki/File%3AKrokussen_(Crocus)%2C_Locatie%2C_Tuinreservaat_Jonkervallei.jpg">https://commons.wikimedia.org/wiki/File%3AKrokussen_(Crocus)%2C_Locatie%2C_Tuinreservaat_Jonkervallei.jpg</a> Converted to PNG format using <a href="http://www.online-convert.com">www.online-convert.com</a>
<b>Mount-of-Olives.MP4</b> 8.51 MB 34 Seconds 1280 x 720 30 Frames / Sec.	View from the Mount of Olives, 2011 Time-Lapse video in webm format, by Marcus Cyrun <a href="https://commons.wikimedia.org/wiki/File:Mount_of_Olives_in_Jerusalem_3.webm">https://commons.wikimedia.org/wiki/File:Mount_of_Olives_in_Jerusalem_3.webm</a> Converted to MP4 format using <a href="http://www.online-convesrt.com">www.online-convesrt.com</a>
<b>BethovenNo9.MP4</b> 8.01 MB 90 Seconds 640 x 360 25 Frames / Sec.	Symphony No.9 (10000 Japanese) in D Minor, 2012 Composer: Ludwig Van Beethoven Location: Tokyo University Downloaded in MP4 format from youtube.com: <a href="https://www.youtube.com/watch?v=X6s6YKITpfw">https://www.youtube.com/watch?v=X6s6YKITpfw</a> Clip cutting using <a href="http://www.online-convert.com">www.online-convert.com</a>
<b>Time-Lapse.MP4</b> 8.83 MB 83 seconds 854 x 480 29 frame / second	Spring is Creeping, 2015 Time lapse video in webm format, downloaded from: <a href="https://commons.wikimedia.org/wiki/File:Spring_is_creeping_in.webm">https://commons.wikimedia.org/wiki/File:Spring_is_creeping_in.webm</a> Converted to MP4 format and cropped using <a href="http://www.online-convert.com">www.online-convert.com</a>
<b>MilkyWay.WMV</b> 6.14 MB 92 seconds 640 x 360 30 frames / second	The Milky Way over Yumi Lake, 2014 Time lapse in webm format, downloaded from: <a href="https://commons.wikimedia.org/wiki/File:Milky_way_-_route_292_shiga_kusatsu_road-1920x1080.webm">https://commons.wikimedia.org/wiki/File:Milky_way_-_route_292_shiga_kusatsu_road-1920x1080.webm</a> Converted to WMV format and cropped using <a href="http://www.online-convert.com">www.online-convert.com</a>

<b>Saut.MP4</b> 2.68 MB 480 x 640 25 frames / second	Video of the <i>saut de l'Ognon, France</i> Downloaded in ogv format from: <a href="https://commons.wikimedia.org/wiki/File%3ASaut_de_l'Ognon.ogv">https://commons.wikimedia.org/wiki/File%3ASaut_de_l'Ognon.ogv</a> Converted to MP4 format using <a href="http://www.online-convert.com">www.online-convert.com</a>
<b>Elisa.MP3</b> 1.48 MB 96 seconds	Fur Elisa, Bagatelle in A minor, 1810 Composer: Ludwig Van Beethoven Downloaded in MP3 format from: <a href="http://www.forelise.com/media/fur_elise_valentina_lisitsa">http://www.forelise.com/media/fur_elise_valentina_lisitsa</a> . Cropped using <a href="http://www.online-convert.com">www.online-convert.com</a>
<b>Renoir2.BMP</b> 958 KB 24-bit 512 x 639	Steps in Algiers, 1882 Painting by Pierre-August Renoir Location: Private collection Downloaded in JPG format from: <a href="https://commons.wikimedia.org/wiki/File%3APierre-Auguste_Renoir_149.jpg">https://commons.wikimedia.org/wiki/File%3APierre-Auguste_Renoir_149.jpg</a> Converted to BMP format using <a href="http://www.online-convert.com">www.online-convert.com</a>
<b>Vase1024.JPG</b> 490 KB 24-bit 1024 x 1298	Sonnenblumen, 1888 Painting by Vincent Van Gogh Location: Amsterdam Museum Downloaded in JPG format from: <a href="https://commons.wikimedia.org/wiki/File%3AVincent_Van_Gogh_0010.jpg">https://commons.wikimedia.org/wiki/File%3AVincent_Van_Gogh_0010.jpg</a>
<b>Renoir4-2048.JPG</b> 487 KB 24-bit 2048 x 1567	Still life with fruit, 1881 Painting by Pierre-August Renoir Location: Art Institute of Chicago Downloaded in JPG format from: <a href="https://commons.wikimedia.org/wiki/File%3APierre-Auguste_Renoir_141.jpg">https://commons.wikimedia.org/wiki/File%3APierre-Auguste_Renoir_141.jpg</a>
<b>Renoir4-128.JPG</b> 4.87 KB 24-bit 128 x 98	Same as in Renoir4-2048.JPG

\* Embedded in Labelle.BMP only








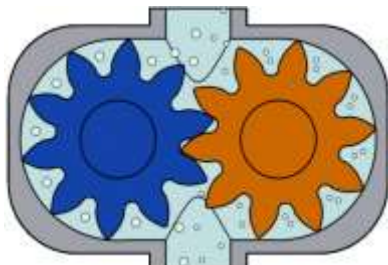






 <p><b>Krokussen.PNG</b></p>	 <p>Mount-of-Olives.mp4</p>	 <p>BeethovenNo9.mp4</p>
 <p><b>Renoir2.BMP</b></p>	 <p>Time-Lapse.mp4</p>	 <p>Saut.mp4</p>
 <p><b>Vase1024.JPG</b></p>	 <p>MilkyWay.wmv</p>	 <p>elisa.mp3</p>
 <p><b>Renoir4-2048.JPG</b></p>	 <p><b>Renoir4-128.JPG</b></p>	

### A.3 Secret Multimedia Files Embedded in Lena.BMP

Secret File Name and Specs.	Source and Details
<b>Kodim24.PNG</b> 689 KB 24-bit 768 x 512	Country Home (Little Red Riding Hood), 1999 Photographer: Alfons Rudolph Downloaded from KODAK dataset: <a href="http://www.r0k.us/graphics/kodak/kodim24.html">http://www.r0k.us/graphics/kodak/kodim24.html</a>
<b>Kodim23.PNG</b> 544 KB 24-bit 768 x 512	Two Macaws, 1999 Photographer: Steve Kelly Downloaded from KODAK dataset: <a href="http://www.r0k.us/graphics/kodak/kodim23.html">http://www.r0k.us/graphics/kodak/kodim23.html</a>
<b>Vase1024.JPG</b>	As in Table A.2
<b>Roses.JPG</b> 349 KB 24-bit 2024 x 1724	Stilleben, Rosen vor Blauem Vorhang, 1908 Painting by Pierre-August Renoir Downloaded in JPG format from: <a href="https://commons.wikimedia.org/wiki/File%3APierre-Auguste_Renoir_144.jpg">https://commons.wikimedia.org/wiki/File%3APierre-Auguste_Renoir_144.jpg</a>
<b>Vase512.JPG</b> 107 KB 24-bit 512 x 645	As in Table A.2
<b>Vase256.JPG</b> 32.9 KB 24-bit	As in Table A.2
<b>Vase 128.JPG</b>	As in Table A.2
<b>Pump.GIF</b> 282 KB 8-bit 600 x 400	Animation of gear pump, 2011 Downloaded in GIF format from: <a href="https://commons.wikimedia.org/wiki/File%3AGear_pump_animation.gif">https://commons.wikimedia.org/wiki/File%3AGear_pump_animation.gif</a>
<b>Apples.JPG</b> 197 KB 24-bit 280 x 1130	Apples from Heaven, 2016 Painting by Shereen Al-Jarrah Location: Cambridge High School

<b>First-day-of-Spring.GIF</b> 136 KB 8-bit 546 x 215	Google first day of spring, 2016 Downloaded in animated gif from: <a href="https://www.google.com/doodles/first-day-of-spring-2016-northern-hemisphere">https://www.google.com/doodles/first-day-of-spring-2016-northern-hemisphere</a>
<b>kodim09-256.PNG</b> 134 KB 24-bit 256 x 256	Sailboats, 1999 Photographer: John Menihan Downloaded in PNG format from KODAK dataset: <a href="http://www.r0k.us/graphics/kodak/kodim09.html">http://www.r0k.us/graphics/kodak/kodim09.html</a> Cropped to 256 dimension using <a href="http://www.online-convert.com">www.online-convert.com</a>
<b>Lena_Gray_256.TIF</b> 64.2 KB 8-bit 256 x 256	Lena Soderberg (aka Lenna), 1972 Photograph Downloaded in tif gray scale format from Gonzles book website: <a href="http://www.imageprocessingplace.com/root_files_V3/image_databases.htm">http://www.imageprocessingplace.com/root_files_V3/image_databases.htm</a>

 <p><b>Kodim24.PNG</b></p>	 <p><b>Kodim23.PNG</b></p>	 <p><b>Vase1024.JPG</b></p>
 <p><b>Roses.JPG</b></p>	 <p><b>Vase512.JPG</b></p>	 <p><b>Vase256.JPG</b></p>
 <p><b>Vase128.JPG</b></p>	 <p><b>Pump.GIF</b></p>	 <p><b>Apples.JPG</b></p>
 <p><b>First-day-of-spring.GIF</b></p>	 <p><b>Kodim09-256.PNG</b></p>	 <p><b>Lena_Gray_256.TIF</b></p>