Hardware-Accelerated Least Significant Bit Framework: A Low-Cost Approach to Securing Clinical Data

A.M. Adeshina, Siti Fatimah Abdul Razak, Sumendra Yogarayan, Md Shohel Sayeed
Faculty of Information Science and Technology, Multimedia University, Malaysia
E-mail: am.adeshina@mmu.edu.my, codedengineer@yahoo.com, fatimah.razak@mmu.edu.my, sumendra@mmu.edu.my, shohel.sayeed@mmu.edu.my

Keywords*:* steganography, least Significant Bit, CUDA

Received: December 24, 2023

Information Technology is continuously making communication much easier thereby drawing the interest of millions of users, including medics, to data communications. However, security of the information communicated still remains issues of concern, especially when it comes to patients' confidential information. Apparently, various concepts have being proposed to solving most of the problems of information security, such as the cryptography and the steganography. Image steganography approach focuses on concealing the existence of information from unintended users. Least Significant Bit (LSB) approach of hiding data is one of the most popular in this regard but with weakness of low embedding capacity and therefore low security. Consequently, there have been several attempts by researchers to improving the embedding capacity and low security in LSB. This study therefore proposes hardware-accelerated steganography data hiding framework towards addressing the weaknesses of low security and embedding capacity exhibited in LSB data hiding approach. Within Compute Unified Device Architecture (CUDA), the proposed framework was able to conceal secret data using the LSB approach where the pixels of the cover image and the secret data were converted into their equivalent binary bits, and the binary bits grouped into bytes of 8-bits each. The LSB of the cover image was replaced with the bits of the secret data starting with the most significant bit of each byte of 8-bit of the secret data, until the whole data is fully embedded in the cover image to generate the stego-image. The proposed framework was further evaluated for possible security of the embedded secret data by encrypting the stego-image with password. The experimental results show that the images in PNG, JPG and JPEG format tend to embed and hide the secret data successfully but with increase in the size of the generated stego-image, while images in TIFF and BMP format tend to embed and hide the secret data successfully and retain their original sizes. Interestingly, the entire data hiding stages were achieved within 3s for up to 110KB of data and not more than 4s for 192KB.

Povzetek: Raziskava uvaja okvir za steganografijo s pospeševanjem strojne opreme na osnovi metode najmanj pomembnega bita (LSB), ki izboljšuje zmogljivost skrivanja podatkov v slikah. Omogoča povečanje varnosti in zmogljivosti vgradnje podatkov pri nizkih stroških.

1 Introduction

The ever-increasing development in the field of technology has made data communication much easier and therefore drawn the interest of millions of users. Hospitals and clinics are greatly benefiting from the revolutions in communication. Data communication deals with transferring data from one location to another through a certain medium such as cable or wireless medium inform of radio frequency. Therefore, securing information over the network is of great importance.

Over the years, different methods such as Cryptography and Steganography have been developed to secure transmission of data. In cryptographic method, information are encoded into scrambled format which are unreadable using different encryption algorithms (Kesa, 2018), but these scrambled information, when spotted by intruders may raise a flag and intruders may pre-empt that they are encrypted information and as a result perform decryption to obtain the hidden data. However, steganography methods on the other hand, conceal the existence of information by hiding it in any cover media such as videos, audios, texts and images (Hussain et al., 2018; Sahu & Swain, 2019).

The use of cryptography in information security helps to achieve confidentiality, authenticity, data integrity, and access control. There are various methods developed for encrypting data into scrambled format which will make no meaning to intruders or unauthorized persons. However, encrypted data will create an impression to unauthorized person that a secret data is being transmitted and be curious to reveal the data. Thus, cryptography is not enough for keeping the information as a secret. As it is important to secure secret data as well as to hide its existence, steganography is the technique which is used to implement such. Steganography is different from cryptography because cryptography focuses on keeping information secret whereas steganography focuses on making the existence of the information secret (Kesa, 2018).

Image steganography technique (i.e., using images to conceal the existence of information) is the most popular among the steganography, this is as a result of consistent availability of images and the simplicity of the method. Image steganography techniques can be classified into two major categories; the spatial domain techniques and the frequency domain techniques. In spatial domain techniques, image pixels are manipulated to store the secret message, while in frequency domain techniques, the image is first transformed and then embedded (Silambarasan & Abenaya, 2016).

This study aims to develop a hardware-accelerated steganography data hiding technique towards addressing the weaknesses of low security and embedding capacity exhibited in the Least Significant Bit (LSB) approach. A framework with high processing capabilities for data hiding using least significant bit is proposed, designed and implemented. The study was evaluated using cover image in different format and comparing the generated stego-image image with the original image based on size and Human Visual System (HVS). The focus of the study is to improve the processing capabilities in LSB in terms of the usual constrains in computational cost, and also the low embedding capacity therefore attributed to its low security which are the main shortcomings discouraging the frequent use LSB for sensitive and reliable applications.

2 Related works

Cryptography is a data security technique used for data protection and confidentiality. Cryptography techniques do not only help users in sending the information in a safe and secured way but also in the process of authentication before accessing the files or the data (Kesa, 2018). The confidentiality of the information which is communicated over the internet is the most crucial worry and issue to the users and organizations respectively. Leakage of internal confidential documents of an organization may put the entire organization at risk (Kumari, 2017), patients are mostly not comfortable with such.

Different steganography methods have been proposed by many researchers, the most relevant to this study are in the image steganography, where the secret data is encoded in an image. Research conducted by Chitradevi et al. (2017) makes use of Least Significant Bit (LSB) algorithms for hiding the data into an image. The technique converted the secret data from decimal to binary and also read the cover image and converted it from decimal to binary. The converted binary for the cover image is then petitioned into bytes of 8bits each, where the least significant bit of each byte is replaced with the already converted secret data. This technique successfully hides the secrete data from Human Visual System (VHS) though it causes higher distortion to the cover file in many cases.

Sahu & Swain (2018) proposed an improved image stenographic technique based on the principle of modified least significant bit (LSB) substitution and LSB matching which was able to improve on the distortion encountered in Chitradevi et al. (2017) approach. The technique proposed by Sahu and Swain (2018) was divided into 3 variants. A data consisting of 2 pixels was embedded in all the three variants. In the proposed work, the first variant initially uses the 6th and 7th bits to hide 2 bits of secret data in the first pixel of a block and with more modification to the pixel by ±1 or 0, it hides 2 more bits in a block. The second variant hides 3 bits while the third variant hides 2 bits respectively in a block. The experimental results prove that the first variant offers better capacity whereas third variant offers better peak signal to noise ratio (PSNR).

In another approach proposed by Khan & Bianchi (2018), Ant Colony Optimization (ACO) was used to detect complex region of the cover image where the Least Significant Bit were substituted with the secret data in the complex region's pixel to hide the secret data. This technique provides efficient and secure data hiding method with high quality stego-image, good Peak Signal to Noise Ratio (PSNR) and reasonable data hiding capacity.

Nie et al. (2019) proposed an approach which uses Least Significant Bit (LSB) and knight tour algorithm for image steganography. Basically, the proposed approach was divided into two parts, the sender and the receiver side. At sender's side, the secret messages were converted into binary from its original ASCII value and were embedded in the image pixels by replacing the Least Significant Bit of the image pixel values with the secret message bits in the order determined using the Knight Tour Algorithm. While at the receiver's side, the stego key (i.e., the order in which the cover image is embedded with the secret message) and the extracting algorithms to decode the image were used with the LSB decoder to separate the secret message from the image pixel values. The extracting algorithm was used to decrypt the data to reveal the hidden message. The approach proved resourceful in hiding the secret data when tested with different types of attacks on stenography algorithms. However, the approach has a fatal flaw of only allowing the used images that can be divided by 4 without any remainder as cover which enables the algorithm to be able to walk through the whole image pixels for encoding as well as the capacity of embedding that can be done in the cover image. This was improved further on by an approach proposed by Sahu & Swain (2019) which uses two Reversible Data Hiding (RDH) approaches to promote the embedding capacity and image quality. The first approach extends the LSB matching in dual images while the second approach utilizes four identical images of the cover image to embed the secret data using n-rightmost bit replacement (n-RBR) phase in the first two identical images, and modified pixel value differencing (MPVD) phase in the last two identical images. Though this approach improves on the embedding capacity but the image quality is not at its best and it takes more time to process.

Aditya & Gandharba (2019) studies proposed two techniques. The techniques used original image *O* with pixels *{O1,O2,O3,O4,…..,On}* and its two mirrored images are *M* and *G* with pixels *{m1,m2,m3,m4,…..,mn}* and *{n1,n2,n3,n4,…..,nn}* respectively where the mirrored images are the replica of the original image. The first technique initially considers a pair of two consecutive pixels *(01, 02)* from the original image. Then, using the LSB matching, two separate pairs *(m1, m2)* and *(g1, g2)* for the mirrored images were modified. Each pair of the two mirrored image hides 2 bits. Later, the pixels were readjusted to ensure it could be restored at the receiver's side with the exact data recovery. Applying modified LSB matching, two distinct stego-pixels were obtained for each original pixel following the second technique. Later, with these two separate sets of stego-pixels, two stego-images were obtained. Both the techniques ensured complete reversibility of the original image and extraction of secret data at the recipient end. The first technique offered higher PSNR than the second technique while both techniques had equal hidden capacity.

Sakshi et al. (2022) considered spatial domain approaches for image and text hiding. One of the eight bits of the first components of the pixels in the carrier image were replaced with the most significant bits of the secret data in the implementation of Least Significant Bit steganography for the text and image hiding. Similarly, Faheem et al. (2022) also applied Least Significant Bit through an image gradient and chaotic map. The gradient of the image expresses the rapid changes in an image while a chaotic substitution box was used to scramble the watermark in accordance to the guided piecewise linear chaotic map.

Chhabra et al. (2022) used image steganalysis with image decoder using Least Significant Bit and Multiple Significant Bit techniques with the plans of combining the data compression and cryptography technologies to fulfill the need for privacy in internet. More efforts were observed by Sahu & Swain (2022) when the proposed improved Reversible Data Hiding-based approaches such as the improved dual image-based Least Significant Bit matching with reversibility, and also like n-rightmost bit replacement and modified pixel value differencing. The study of Setiadi (2022) was a novel study that proposed dilated hybrid edge detection of the three Most Significant Bits pixels of cover images for image steganography. Embedded messages were achieved using the number of Least Significant Bits replaced in the edge area and the number of bits replaced in the non-edge area.

One of the most flexible approaches in data hiding is Least Significant Bit, however, its shortcomings discouraged its frequent use for sensitive and reliable applications such as clinical data, and when such has to be augmented, it is usually with high cost. Similarly, to the point of this study, there have been several attempts to improving the performances of Least Significant Bit approach. Undoubtedly, more efforts to improving the embedding capacity, the low security in LSB within a reasonably cheaper cost would not only make the Least Significant Bit approach more resourceful and reliable in securing clinical data but generally in data hiding procedures and applications. Table 1 summarizes previously related contributions to this study.

Table 1: Summary of literature

Authors	Approach	Contribution
Chitradevi et al. (2017)	Use of Least Significant Bit (LSB) algorithms for hiding the data into an image.	The technique converted the secret data from decimal to binary and also read the cover image and converted it from decimal to binary.
Sahu & Swain (2018)	The technique proposed was divided into three variants for usage with a data consisting of 2 pixels embedded in all the three variants.	Proposed an improved image stenographic technique based on the principle of modified least significant bit (LSB) substitution and LSB matching which was able to improve on the distortion encountered in Chitradevi et al. (2017) approach.
Khan & Bianchi (2018)	Ant Colony Optimization (ACO) was used to detect complex region of the cover image where the Least Significant Bit was substituted with the secret data in the complex region's pixel to hide the secret data.	This technique provides efficient and secure data hiding method with high quality stego-image, good Peak Signal to Noise Ratio (PSNR) and reasonable data hiding capacity.
Nie et al. (2019)	Proposed approach which used Least Significant Bit (LSB) and knight tour algorithm for image steganography.	The approach proven resourceful to hiding the secret data when tested with different types of attacks on stenography algorithms such

		assteganalysis .
Sahu and Swain (2019)	Used two Reversible Data Hiding (RDH) approaches to promote the embedding capacity and image quality.	The approach improved on the embedding capacity recorded from previously related studies.
Aditya and Gandharba (2019)	The technique uses original image *O* with pixels $\{o_1,o_2,o_3,o_4,\ldots..,o_n\}$ and its two mirrored images are *M* and *G* with pixels$\{m_1,m_2,m_3,m_4,\ldots..,m_n\}$ and $\{n_1,n_2,n_3,n_4,\ldots..,n_n\}$ respectively where the mirrored images were the replica of the original image.	Both the techniques ensure complete reversibility of the original image and extraction of secret data at the recipient end.
Sakshi et al. (2022)	The study considered spatial domain approaches for image and text hiding.	One of the eight bits of the first components of the pixels in the carrier image were replaced with the most significant bits of the secret data in the implementation of Least Significant Bit steganography for the text and image hiding.
Faheem et al. (2022)	This study applied Least Significant Bit through a image gradient and chaotic map.	The gradient of the image expresses the rapid changes in an image while a chaotic substitution box was used to scramble the watermark in accordance to the guided piecewise linear chaotic map.
Chhabra et al. (2022)	The study used image steganalysis with image decoder using Least Significant Bit and Multiple Significant Bit techniques.	It combines the data compression and cryptography technologies to fulfill the need for privacy in internet.
Sahu & Swain (2022)	Improved Chhabra et al. (2022) approach with the proposed Reversible Data Hiding-based approach.	Improved dual image-based Least Significant Bit matching with reversibility, and also like n-rightmost bit replacement and modified pixel value differencing.
Setiadi (2022)	A novel study that proposed dilated hybrid edge detection of the three Most Significant Bits pixels of cover images for image steganography.	Embedded messages were achieved using the number of Least Significant Bits replaced in the edge area and the number of bits replaced in the non-edge area.

3 Methodology

The proposed framework for the study makes use of the Least Significant Bit (LSB) to hide secret data from unintended users and it consist two phases: Embedding Phase and Extraction Phase. The embedding phase, consist of series of processes such as obtaining the pixel of the cover image and embedding the secret data in the least significant of the pixel in a while the secret data is extracted from the Least Significant Bit of the cover image pixel in the extraction phase. The entire implementation was hardware-accelerated using Compute Unified Device Architecture (CUDA). The proposed framework is presented in Figure 1.

3.1. Cover image

The cover image is required in embedding the secret data. The Cover Image Pixels made up the cover image and are extracted from the cover image. The Cover Image Pixels are converted into binary so as to embed the secret data in the Least Significant Bit of each pixel of the cover image.

3.2. Secret data

The secret data is the intended data to be sent over a channel using the cover media (image). The data’s existence is to be concealed from unintended users who may tamper with the data or use it for malicious activities. The intended secret data is converted to binary which are imbedded in the Least Significant Bit of the cover image pixels by replacing the Least Significant Bit of the cover image pixel with each bit of the secret data.

3.3. Encryption system

At the embedding phase, the Encryption System is used to provide password to further enhance the security of the data being embedded in the pixels of the cover image. While during extraction, it is used to authenticate the password is provided to extract the secret data.

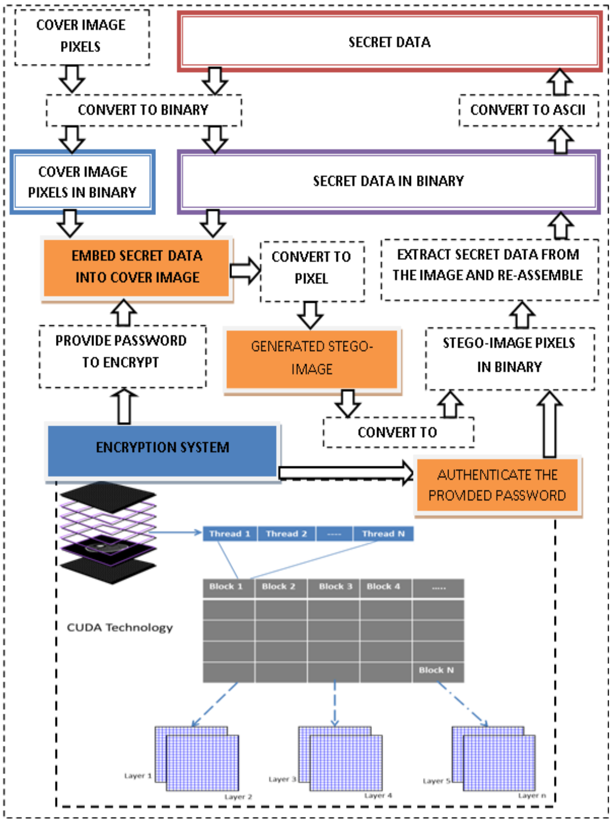


Figure 1: The Proposed framework

3.4. Stego-Image

The stego-image is the image generated when the secret data has been embedded in the cover image. The Human Visual System (HVS) is unable to identify any difference between the original image and the stego-image.

3.5. The proposed embedding process

The algorithm for the proposed embedding process is as follow:

1. **Begin**
2. Convert the secret data from ASCII form to binary.
3. Read the pixels of the cover image and convert the pixels to binary.
4. Apply the Least Significant Bit (LSB) technique to replace the eighth bit of every chosen pixel with the secret data bit as indicated

4.1. Identify the first pixel from the cover image.

4.2. Identify the first bit of the secret data.

4.3. Replace the Least Significant Bit of the first pixel with the first bit of the secret data.

5. Repeat “Step 4” for all other pixels of the cover image until the data in completely embedded.
6. Convert the generated binary bits back to pixel to generate the stego-image.
7. Use the Encryption System to provide password in the embedding stage to further enhance the security of the secret data.
8. **End**

The flow of the proposed embedding process is presented in Figure 2.

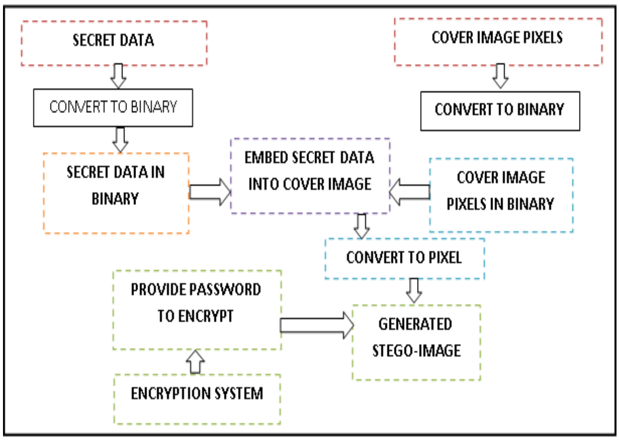


Figure 2: The proposed embedding process

3.6. The proposed extraction process

The algorithm for the proposed extraction process is as follow:

1. **Begin**
2. Use the Encryption System to authenticate the password provided at the embedding stage.
3. Read the pixels of the stego-image and convert the pixels to binary.
4. Extract the 8th bit of every pixel identified.
5. Group the collected bits into bytes of 8 bits each
6. Convert each byte to its equivalent ASCII value to reveal the secret message.
7. **End**

The diagrammatic flow of the entire steps of the proposed hardware-accelerated steganography data hiding technique is presented in Figure 3, Figure 4 and Figure 5.

3.7. Compute unified device architecture (CUDA)

Partitioning of images or data entering the CUDA phase of the framework is significant to achieving prompt processing. Each data is partitioned and processes following the layers, blocks, and thread subroutines integrated in the workflow thereby facilitating greater reduction in the processing time of the entire data hiding procedures. The CUDA framework integrated in the proposed framework was adopted from Adeshina & Hashim (2016).

Hardware-accelerated utilizes Nvidia CUDA Compiler (nvcc) and creates pointers and device arrays while setting bands for each array. Arrays are allocated to each CUDA host and process threads accordingly. For efficient computations, all blocks copy to shared memory with the block size of 1KB, and later fragment blocks into orthogonal slices for parallel processing of datasets to record efficient processing time within the entire framework.

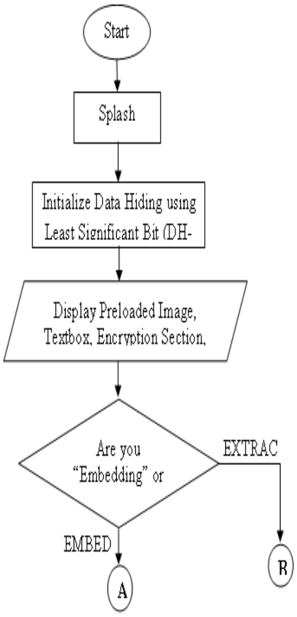

Start
Splash
Initialize Data Hiding using Least Significant Bit (DH-
Display Preloaded Image, Textbox, Encryption Section,
Are you "Embedding" or
EXTRAC
EMBED
A
B

Figure 3: Hardware-accelerated data hiding steps

4 Implementation

The hardware-accelerated data hiding using Least Significant Bit was implemented as a standalone application with backend coding, and user interfaces that are visible to the end user of the application. These backend coding and interfaces were developed using Microsoft Visual C-Sharp (C#) within .NET framework integrated with NETcore library packages, an event driven program in such a way that action is carried out when an event is performed.

C# has a graphical user interface (GUI) which must be designed to communicate with the user in other to make the usage of the software easier. The user interface is used to accept input from the user and the computer will run the backend codes that are in the program for output. The application ran conveniently on desktop computer of minimum capacity of Pentium II, 350 MHz processor, 128 MB of RAM.

4.1. Study datasets

To evaluate the proposed framework, study dataset (images) ware obtained from the database of Signal and Image Processing Institute (SIPI) of the University of Southern California (USC), a database designed to support research in image processing, image analysis, and machine vision.

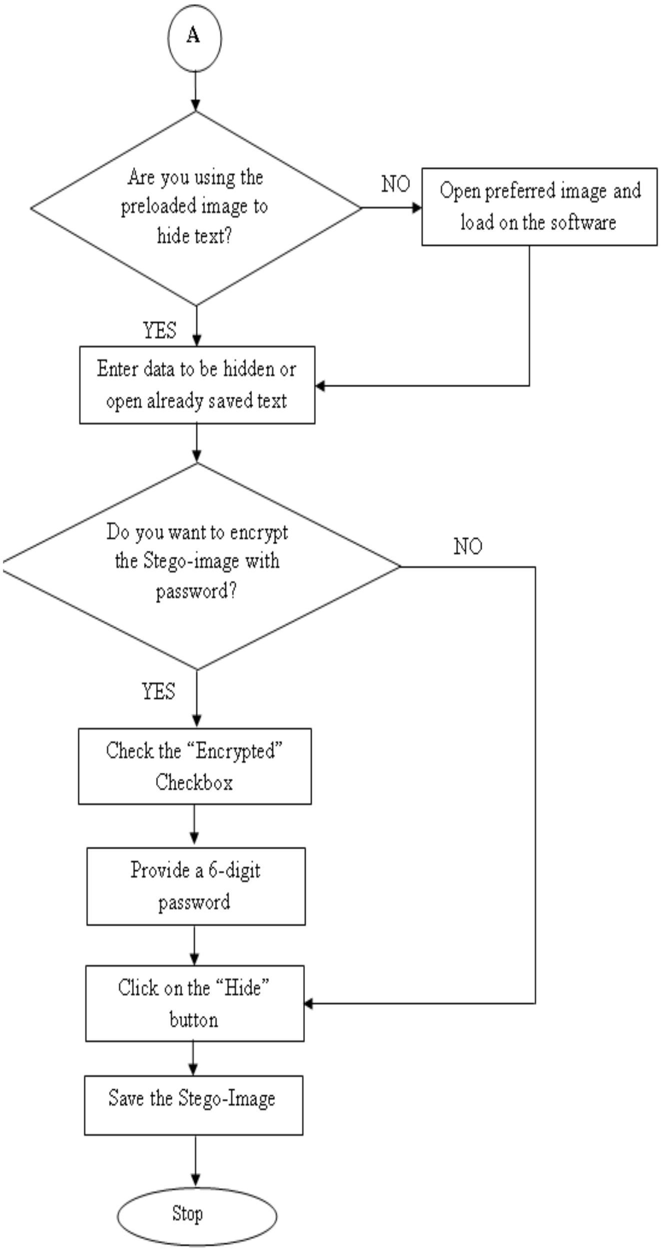

A
Are you using the preloaded image to hide text?
NO
Open preferred image and load on the software
YES
Enter data to be hidden or open already saved text
Do you want to encrypt the Stego-image with password?
NO
YES
Check the "Encrypted" Checkbox
Provide a 6-digit password
Click on the "Hide" button
Save the Stego-Image
Stop

Figure 4: Hardware-accelerated data hiding steps

The data were of 2D images of different sizes of dimension 256 X 256 pixels, 512 X 512 pixels and 1024 X 1024 pixels on 8 bit/pixel black and white. This enabled us to substantiate the embedding capacity and effectiveness of the Random Walk model of the proposed framework to embed in every available pixel of the image.

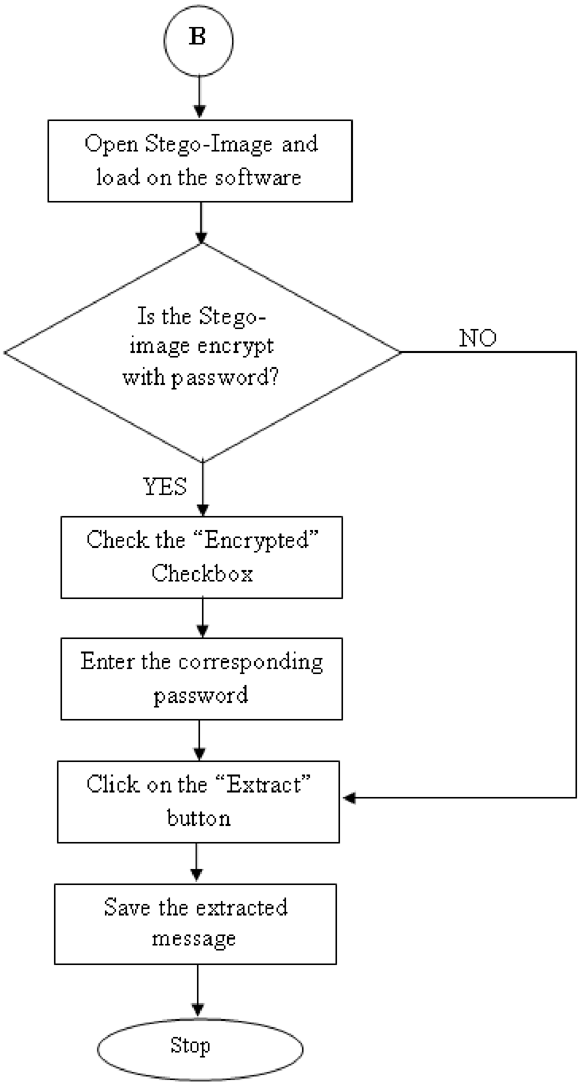

B
Open Stego-Image and load on the software
Is the Stego-image encrypt with password?
NO
YES
Check the "Encrypted" Checkbox
Enter the corresponding password
Click on the "Extract" button
Save the extracted message
Stop

Figure 5: Hardware-accelerated data hiding steps

4.2. Results

In order to analyze and determine the best image format suitable for hiding data in the proposed framework, cover images of different format were used and the obtained results were documented. Processing time required to hide data from image (in Seconds) and time required to extract data from image (in Seconds) are presented using CPU and GPU as the experimental testbeds, are presented in Table 2 and Table 3 respectively. Histograms of cover images are presented in Fig. 6 while the analyses of Signal-to-Noise Ratio (SNR) and the Mean Absolute Error (MAE) are documented in Table 4(a-b). The SNR values obtained were considerably higher, indicating the signal levels are greater than the noise level, they would have been

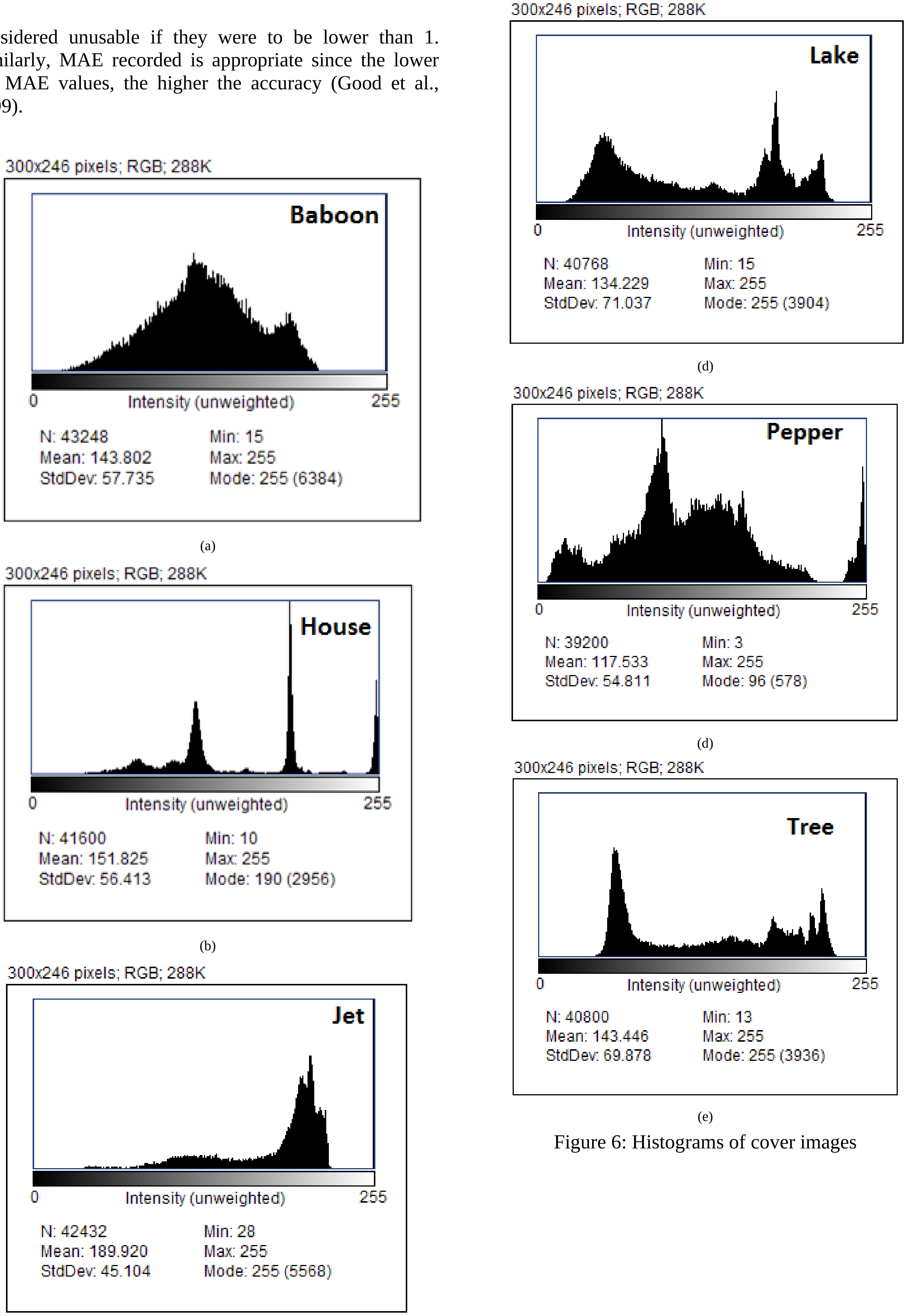
Table 2: Analysis of the Sample Images (GPU)

Image Name	**Baboon**	**House**	**Jet**	**Lake**	**Pepper**	**Tree**
Cover Image Size	833KB	22.60KB	768KB	768KB	110KB	192KB
Cover Image Dimension	512 X 512	256 X 256	512 X 512	512 X 512	512 X 512	256 X 256
Cover Image Format	PNG	JPG	TIFF	BMP	JPEG	TIFF
Secret Message Size	55.9KB	55.9KB	55.9KB	55.9KB	55.9KB	55.9KB
Secret Message Format	TXT	TXT	TXT	TXT	TXT	TXT
Stego-Image Size	1.00 MB	192KB	768KB	768KB	768KB	192KB
Stegoimage Dimension	512 X 512	256 X 256	512 X 512	512 X 512	512 X 512	256 X 256
Stego-Image Format	PNG	PNG	PNG	PNG	PNG	PNG
Hidden Time	3s	4s	3s	3s	3s	4s
Extraction Time	3s	4s	3s	3s	3s	4s

Table 3: Analysis of the Sample Images (CPU)

Image Name	**Baboon**	**House**	**Jet**	**Lake**	**Pepper**	**Tree**
Cover Image Size	833KB	22.60KB	768 KB	768KB	110KB	192KB
Cover Image Dimension	512 X 512	256 X 256	512 X 512	512 X 512	512 X 512	256 X 256
Cover Image Format	PNG	JPG	TIFF	BMP	JPEG	TIFF
Secret Message Size	55.9KB	55.9KB	55.9 KB	55.9KB	55.9KB	55.9 KB
Secret Message Format	TXT	TXT	TXT	TXT	TXT	TXT
Stego- Image Size	1.00MB	192KB	768 KB	768KB	768KB	192KB
Stegoimage Dimension	512 X 512	256 X 256	512 X 512	512 X 512	512 X 512	256 X 256
Stego- Image Format	PNG	PNG	PNG	PNG	PNG	PNG
Hidden Time	7s	8s	7s	7s	7s	8s
Extraction Time	7s	8s	7s	7s	7s	8s

considered unusable if they were to be lower than 1. Similarly, MAE recorded is appropriate since the lower the MAE values, the higher the accuracy (Good et al., 1999).



N: 43248 Min: 15
Mean: 143.802 Max: 255
StdDev: 57.735 Mode: 255 (6384)

(a)

N: 41600 Min: 10
Mean: 151.825 Max: 255
StdDev: 56.413 Mode: 190 (2956)

(b)

N: 42432 Min: 28
Mean: 189.920 Max: 255
StdDev: 45.104 Mode: 255 (5568)

(c)

N: 40768 Min: 15
Mean: 134.229 Max: 255
StdDev: 71.037 Mode: 255 (3904)

(d)

N: 39200 Min: 3
Mean: 117.533 Max: 255
StdDev: 54.811 Mode: 96 (578)

(d)

N: 40800 Min: 13
Mean: 143.446 Max: 255
StdDev: 69.878 Mode: 255 (3936)

(e)

Figure 6: Histograms of cover images

Table 4(a-b): Cover images analysis

(a) Signal-to-Noise Ratio (SNR) Analysis of Cover Images

(Nº) Reference Image	**(Nº) Test Image**	**SNR (dB)**
(1) Baboon.png	(1) Baboon.png	Infinite
(1) Baboon.png	(1) Baboon - Additive Noise.png	15.54953151
(1) Baboon.png	(1) Baboon – Salt and Pepper.png	13.80275480
(1) House.jpg	(1) House.jpg	Infinite
(1) House.jpg	(1) House-Additive Noise.jpg	10.98615457
(1) House.jpg	(1) House-Salt and Pepper.jpg	14.14271022
(1) Jet.tif	(1) Jet.tif	Infinite
(1) Jet.tif	(1) Jet-Additive Noise.tif	17.78811704
(1) Jet.tif	(1) Jet-Salt and Pepper.tif	15.39355299
(1) Lake.bmp	(1) Lake.bmp	Infinite
(1) Lake.bmp	(1) Lake-Additive Noise.bmp	10.96267612
(1) Lake.bmp	(1) Lake-Salt and Pepper.bmp	13.17486803
(1) Pepper.jpg	(1) Pepper.jpg	Infinite
(1) Pepper.jpg	(1) Pepper-Additive Noise.jpg	10.12535991
(1) Pepper.jpg	(1) Pepper-Salt and Pepper.jpg	12.30067664
(1) Tree.tif	(1) Tree.tif	Infinite
(1) Tree.tif	(1) Tree-Additive Noise.tif	16.01842830
(1) Tree.tif	(1) Tree-Salt and Pepper.tif	13.72942467

(b) Mean Absolute Error (MAE) Analysisof Cover Images

(Nº) Reference Image	**(Nº) Test Image**	**MAE**
(1) Baboon.png	(1) Baboon.png	0
(1) Baboon.png	(1) Baboon - Additive Noise.png	19.95595444
(1) Baboon.png	(1) Baboon – Salt and Pepper.png	6.16296617
(1) House.jpg	(1) House.jpg	0
(1) House.jpg	(1) House-Additive Noise.jpg	38.47705000
(1) House.jpg	(1) House-Salt and Pepper.jpg	10.10357500
(1) Jet.tif	(1) Jet.tif	0
(1) Jet.tif	(1) Jet-Additive Noise.tif	19.98088874
(1) Jet.tif	(1) Jet-Salt and Pepper.tif	6.47951686
(1) Lake.bmp	(1) Lake.bmp	0
(1) Lake.bmp	(1) Lake-Additive Noise.bmp	34.00478967
(1) Lake.bmp	(1) Lake-Salt and Pepper.bmp	6.14926072
(1) Pepper.jpg	(1) Pepper.jpg	0
(1) Pepper.jpg	(1) Pepper-Additive Noise.jpg	33.09069138
(1) Pepper.jpg	(1) Pepper-Salt and Pepper.jpg	10.19653269
(1) Tree.tif	(1) Tree.tif	0
(1) Tree.tif	(1) Tree-Additive Noise.tif	19.84471662
(1) Tree.tif	(1) Tree-Salt and Pepper.tif	6.46784866

5 Discussion

One of the most flexible approaches in data hiding is Least Significant Bit, however, its shortcomings of low embedding capacity attributing to its low security coupled with its bottleneck of processing capabilities in terms of the usual constrains in computational cost discouraged its frequent use for sensitive and reliable applications. To the point of this study, there have been several attempts to improving such constrains, however when such has to be augmented, it is usually with high computational cost. In the actual sense, in all the efforts so far as summarized in Table 1, though great contributions, all were identified as complementary in one way or the other but the weaknesses of low embedding capacity and therefore low security in LSB was not completely resolved.

Undoubtedly, more efforts to improving the embedding capacity, the low security in LSB within a reasonably cheaper cost would not only make the Least Significant Bit approach more resourceful and reliable in securing clinical data but generally in data hiding procedures and applications. This study addressed the shortcomings on improving the low embedding capacity in LSB through the proposed hardware-accelerated steganography data hiding framework approach.

6 Conclusion

The proposed framework is made up of three stages; the preparations stage, the embedding stage and the extraction stage. The preparation stage is where the cover image and the secret data is being converted into their equivalent binary bits, which enhance the embedding process. In this stage, the cover image and the secret data are converted into bytes of 8-bits each. At the embedding stage, the Encryption System is used to provide password which enhances the security of the secret data that will be embedded in the cover image. This is where the Least Significant Bit of the cover image is replaced with the bit of the secret data starting with most significant bit of each byte of 8-bit of the secret data, until the whole data is fully embedded. During the extraction stage, the Encryption System is also used to authenticate the password provided at the embedding stage to reveal the secret data.

The extracted bit is then arranged in 8-bits and converted to it equivalent ASCII value to reveal the secret data. However, the study is limited to spatial domain approach of image steganography in the data hiding technique using Least Significant Bit.

Interestingly, the experimental results with PNG, JPG and JPEG formats show that the images tend to embed and hide the secret data successfully which was with increase in the size of the generated stego-image. However, images in TIFF and BMP formats tend to embed and hide the secret data successfully and even retain their original sizes. Significantly, with the proposed Hardware-Accelerated Least Significant Bit approach, the entire data hiding stages were achieved within 3s for up to 110KB of data and not more than 4s for 192KB without compromising those secure qualities in Least Significant Bit approach.

Acknowledgement

This study is supported by Multimedia University, Malaysia. Authors would like to thank all the anonymous reviewers for their constructive comments.

References

[1] Hussain, M., Wahab, A.W.A., Idris, Y.I. Bin, Ho, A.T.S. & Jung, K.-H., (2018). Image Steganography in Spatial Domain: A Survey. Signal Process. Image Communication. 65, 46–66.

[2] Sahu, A.K. & Swain, G, (2019). High fidelity based reversible data hiding using modified LSB matching and pixel difference, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.07.004

[3] Kesa, N.R.K. (2018). Steganography A Data Hiding Technique. Culminating Projects in Information Assurance.

[4] Silambarasan, D., Abenaya, S.N. (2016). Secure Data Hiding Using Multilevel Steganography.

International Journal of Engineering & Technology Research. 4(3). May-June.

[5] Sahu, A.K. & Swain, G., (2018). An Improved Data Hiding Technique using Bit Differencing and LSB Matching. Internetworking Indonesian Journal. Vol.10/No.1 (2018), pp17-21.

[6] Kumari, S. (2017). A research paper on cryptography encryption and compression techniques. International Journal of Engineering and Computer Science, 6(4), 20915-20919.

[7] Chitradevi, B., Thinaharan, N., Vasanthi, M. (2017). Data Hiding using Least Bit Steganography in Digital Images. Statistical Approaches on Multidisciplinary Research 1, 144-150.

[8] Khan S., & Bianchi T., (2018). Ant Colony Optimization (ACO) based Data Hiding in Image Complex Region, International Journal of Electrical and Computer Engineering (IJECE) Vol. 8, No. 1, February 2018, pp. 379~389 ISSN: 2088-8708, DOI: 10.11591/ijece. v8i1. pp 379-389.

[9] Nie S.A., Sulong G., Ali R. & Abel A. (2019). The use of least significant bit (LSB) and knight tour algorithm for image steganography of cover image. International Journal of Electrical and Computer Engineering (IJECE) Vol. 9, No. 6, December 2019, pp. 5218~5226 ISSN: 2088-8708, DOI: 10.11591/ijece. v9i6.pp5218-5226

[10] Aditya K.S. & Gandharba Swain (2019). Dual Stego-imaging Based Reversible Data Hiding Using Improved LSB Matching. International Journal of Intelligent Engineering and Systems.

[11] Sakshi, S., Verma, S., Chaturvedi, P. & Yadav, S.A. (2022). International Conference on Intelligent Engineering and Management (ICIEM). 415-421.

[12] Faheem, Z.B., Ali, M., Raza, M.A., Arslan, F. & Mehedi, J.A. (2022). Image Watermarking Scheme using LSB and Image Gradient. Applied Sciences. 12(9), 4202.

[13] Chhabra, A., Woeden, T., Singh, D., Rakhra, M., Dahiya, O. and Grupta A. (2022). Image Steganalysis with Image Decoder using LSB and MSB Technique. International Conference on Intelligent Engineering and Management (ICIEM). 900 -902.

[14] Sahu, A.K. & Swain, G. (2022). High-Fidelity Based Reversible Data Hiding using Modified LSB Matching and Pixel Difference. Journal of King Saud University-Computer and Information Sciences. 34 (4), 1395-1409.

[15] Setiadi D.R.I.M. (2022). Improved Payload Capacity in LSM Image Steganography Using Dilated Hybrid Edge Detection. Journal of King Saud University – Computer and Information Sciences. 34(2), pp. 104-114. Saud University – Computer and Information Sciences. 34(2), pp. 104-114.

[16] Adeshina, A.M. Hashim, R. (2016). ConnectViz: Accelerated Approach for Brain Structural Connectivity Using Delaunay Triangulation. Interdiscip Sc Comput Life Sci. 8(1).

[17] Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J., & Riedl, J. (1999). Combining collaborative filtering with personal agents for better recommendations. Aaai/iaai, 439(10.5555), 315149-315352.