ON THE IMAGE QUALITY AND ENCODING TIMES OF LSB, MSB AND COMBINED LSB-MSB STEGANOGRAPHY ALGORITHMS USING DIGITAL IMAGES

Solomon O.Akinola and Adebanke A.Olatidoye

Department of Computer Science, University of Ibadan, Nigeria

ABSTRACT

The Least Significant Bit (LSB) algorithm and the Most Significant Bit (MSB) algorithm are steganography algorithms with each one having its demerits. This work therefore proposed a Hybrid approach and compared its efficiency with LSB and MSB algorithms. The Least Significant Bit (LSB) and Most Significant Bit (MSB) techniques were combined in the proposed algorithm. Two bits (the least significant bit and the most significant bit) of the cover images were replaced with a secret message. Comparisons were made based on Mean-Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR) and the encoding time between the proposed algorithm, LSB and MSB after embedding in digital images. The combined technique produced a stego-image with minimal distortion in image quality than MSB technique independent of the nature of data that was hidden. However, LSB algorithm produced the best stego-image quality. Large cover images however made the combined algorithm's quality better improved. The combined algorithm had lesser time of image and text encoding. Therefore, a trade-off exists between the encoding time and the quality of stego-image as demonstrated in this work.

KEYWORDS

Steganography, Cover image, Most Significant Bit (LSB), Least Significant Bit (LSB).

1. INTRODUCTION

Steganography is the art and science of hiding sensitive information in ways that prevent detection. The purpose of steganography is to convey a message in such a way that nobody apart from the sender and intended recipient suspects the existence of the message. These messages are transferred through cover carriers such as text, audio, images and protocols [1, 2]. The secret message could be a plaintext, cipher text or images. The embedding of the message into a cover object results in the production of a stego-image. Images are mostly used as cover objects in steganography.

Different image Steganography technique exists which are classified into spatial domain and transform domain steganography. In spatial domain scheme, the secret information is directly embedded. Its high capability of hiding and easy retrieval makes it to be used frequently. An example is the least significant bit algorithm which is key to the embedding algorithm proposed in this paper.

Transform domain scheme is used for hiding a large amount of data. It hides information in frequency domain by altering magnitude of all transforms of cover image. Discrete Cosineransform (DCT), Discrete Fourier Transform, and Wavelet Transform are the main types

DOI:10.5121/ijcsit.2015.7407

of transforms used in steganography. These transforms all have coefficients associated with them. The secret data is hidden within these coefficients which also defines how the image or file should be transformed [3]. Examples include JPEG Steganography and Spread Spectrum.

The performance of a steganography technique can be measured using several parameters, among which are imperceptibility, robustness and capacity. Imperceptibility is defined as the ability to avoid detection, i.e. the inability to determine the existence of a hidden message. This makes it an important requirement in steganography. Robustness refers to how well a steganography technique can resist the extraction of hidden data. It measures the ability of the steganography technique to survive the attempts of removing the hidden information. Such attempts include, image manipulation (like cropping or rotating), data compression and image filtering [4]. Payload Capacity represents the maximum amount of information that can be safely embedded and retrieved in a work without being statistically detectable. When compared with watermarking that requires embedding only a small amount of copyright information, Steganography requires sufficient embedding capacity [5].

The Least Significant Bit (LSB) is one of most common embedding techniques. The least significant bit is the least value in a binary number. In LSB algorithm, data is hidden in the least significant bits of the cover image wfqhhich is not noticeable when viewed with the human eye [4]. The most significant bit (also called the high-order bit) is the bit position in a binary number having the greatest value.

The aim of this work is to compare the image quality and the encoding times of LSB, MSB and the proposed Combined-LSB-MSB Steganography algorithms using digital images. The objectives are to:

- Combine the LSB and MSB techniques into a Hybrid algorithm that embeds secret message bits into the least significant bit and most significant bit of the cover image.
- Compare the LSB, MSB and the proposed algorithms (named Combined-LSB-MSB and hence called Hybrid) in terms of encoding time, MSE (Mean Squared Error) and PSNR (Peak Signal to Noise Ratio).
- Test the algorithms using different image formats (JPG and PNG) and the quality of image with increase in file size.

The rest of the paper is organized as follows. Section II reviews existing image steganography methods and section III presents the proposed image embedding method. The experimental results & discussion are shown in section IV and conclusions are drawn in section V.

2. RELATED WORK

There has been several researches in hiding data inside an image using steganography technique. In Warkentin *et al.* [6] proposed algorithm, the idea was to hide data inside the audiovisual files. El-Emam's [7] proposed steganography algorithm is based on hiding a large amount of data file inside a coloured bitmap image. In his work, he filtered and segmented the image by using bits replacement on the appropriate pixels. A concept defined by main cases with their sub cases for each byte in one pixel was used to select these pixels randomly rather than sequentially. This concept was both visual and statistical. The result of this concept was that 16 main cases with their sub cases covered all aspects of the input data into color bitmap image. Three layers provided high security which made it difficult to break through the encryption of the input data and also undectable when steganalysis is applied. it was concluded that a large amount of data that occupies 75% of the image size can be embedded efficiently and the output will be of high quality.

Chen *et al.* [8] modified a method proposed by Chang *et al.* [9] using the side match method. In this method, data was hidden in the edge portions of the image. The image quality was improved while maintaining the same embedding capacity because the human eyes could rarely see differences in the edge portion. The embedding capacity can also be adjusted based on the demands of individual users. In addition to the improvement on image quality, the proposed approach provided respectable security as well. Wu and Tsai [10] proposed an algorithm using pixel-value differencing which partioned the original image into non-overlapping blocks of two consecutive pixels. A different value was calculated from the values of the two pixels in each block. All possible different values were classified into a number of ranges. The human vision sensitivity to gray value variations from smoothness to contrast was used in selecting the range intervals. A new value which replaced the different value was used to embed the value of a sub-stream of the secret message. The width of the range that the different value belongs to determines the number of bits that can be embedded in a pixel pair. However, in this method the modification is never out of the range interval. The result produced by this method is more imperceptible than those yielded by simple least significant bit replacement method. The secret message that was embedded can be extracted from the resulting stego-image without making reference to the method of the original cover image. The security of the method was shown using dual statistics attack.

Scott [11] work on steganographic techniques using digital images used several iterations of replacement strategies during the construction of the application. The aim was to implement a replacement and extraction steganography scheme using cover images. To extract the embedded textual information from the image, the image created by the application must be processed. This processing outputs the original message and some extra erroneous information. Comparism between LSB replacement scheme with MSB replacement scheme asserted that MSB produced noticeable differences to the cover during the most significant bit replacement. Rohit and Tarun [12] compared LSB and MSB based steganography in gray-scale images. It was concluded that the resulting stego-image using LSB shows no distortion when compared with the original image. The performance of LSB was better than that of MSB. Kanzariya and Nimavat [4] compared various image steganography techniques. The objectives were to identify the requirements of a good steganography algorithm and to determine steganography techniques that are suitable for different applications. In this work, some criteria for imperceptibility of an algorithm were proposed.

3. THE PROPOSED METHOD

The proposed hybrid algorithm combines the LSB and MSB Steganography techniques. Two bits (the least significant and the most significant bits) of the cover images were replaced with a secret message. Figure 1 shows the framework of the proposed algorithm.

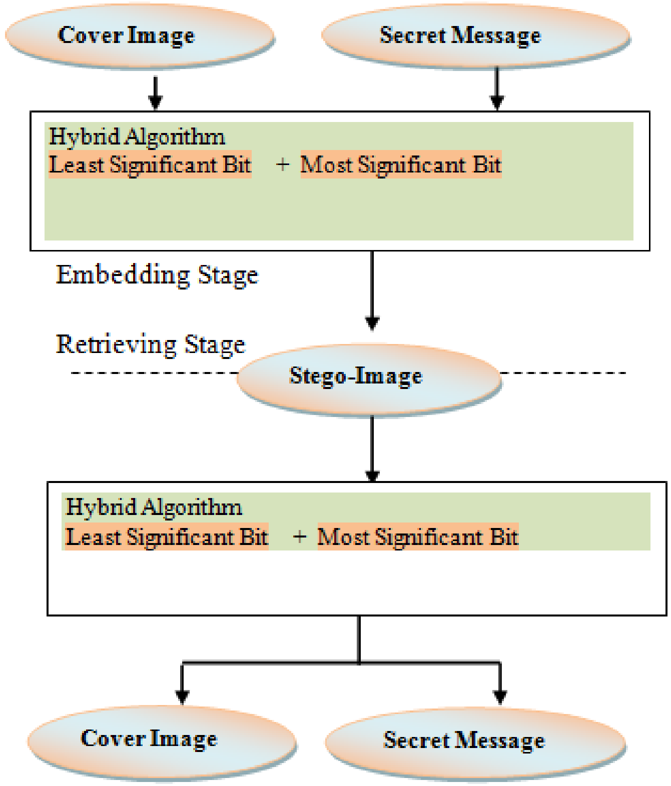


Figure 1: Framework for the Proposed Hybrid-LSB-MSB Algorithm

From figure 1, once the cover image and the secret message (image or text) has been selected, the embedding stage of the combined algorithm takes two bits of the secret message and embeds the first message bit in the least significant bit of the cover image byte and the second message bit in the most significant bit of the cover image byte. The output of this process is a stego-image. The retrieving stage is just the inverse of the embedding stage.

A. Embedding Algorithm

```
Begin
      Load the cover image
      Convert image to byte array
      Convert message data to byte array
      If message cannot be contained in   cover image
          Exit with error message
     Else
        For each bit in the message byte
        Begin
          If LSB
             Hide message bit in the lsb of the corresponding cover image byte
         If MSB
             Hide message bit in the msb of the corresponding cover image byte
        If HYBRID
           Get two message bits
```

```
                Hide the first message bit in the lsb of the corresponding cover image byte
                Hide the second message bit in the msb of the corresponding cover image byte
            End
        End
```

B. Decoding Algorithm

```
    Begin
        Load stego image
        Convert stego image into byte array
        If decoding type is LSB
          Begin
            For the first 32 byte
                Copy the lsb into an array of length 32
                Convert the array into integer value
                Create an array of length of the integer value
                  Starting from length 32+1 of the stego-
                  image array
              Begin
                Copy the lsb of the equivalent stego array into an array of length 8
                Convert the array into a byte value and save in the corresponding index of the
                created array
        Convert the array value into string or image
        End
End

The same approach goes for MSB and HYBRID
End
```

C. File Format

Any image file format can be used as both the cover image and the secret image. However, the image was first converted into PNG format before anything can be done on it. After the whole process, the image was converted back to its original format. PNG format is preferred because it is supported by the Java image IO library; it applies lossless file compression method and allows for easy interchange and viewing of image data stored on local or remote computer systems [13]. Also, it seems to maintain a high degree of image quality after the message has been embedded [11].

D. Comparison Procedures

To compare the image quality of the three algorithms i.e. the LSB, MSB and the proposed Hybrid algorithm, three metrics were used, which are the Mean-Squared Error (MSE), the Peak Signal-to-Noise Ratio (PSNR) and the encoding time.

- *Mean-Squared Error (MSE)*

The MSE represents the cumulative squared error between the cover image and the stego-image. To calculate the mean-squared error (MSE) between two images I_1 (M, N) and I_2 (M, N) the equation is as follows:

$$MSE = \frac{\sum M,N\ [I1\,(M,N)\ -\ I2\,(M,N)]^2}{M * N}$$

M and N are the number of rows and columns in the input images respectively [14].

- *Peak Signal-to-Noise Ratio [PSNR]*

The PSNR measures the statistical difference between the cover and stego image [15]. The mean-squared error value is needed to compute the PSNR. The equation is as follows:

$$PSNR = 10\log_{10}\frac{R^2}{MSE}$$

The value of R is 255.

However, the lower the MSE value and the higher the PSNR value then the better the quality of the image.

4. RESULTS AND DISCUSSION

A simple system was developed to implement the LSB, MSB and the proposed combined algorithms using JAVA programming language. There are two sides to the system, the encoding interface and the decoding interface for hiding and retrieving purposes respectively.

We tested the system using two different image formats (roses.jpg, giraffe.png) as cover images. A blue-footed booby bird (Figure 2) with dimension 160 x 120 pixels and file size of 4 kilo byte was used as the message image for each cover image respectively. A 30.4 kilo byte document was also used as message text.

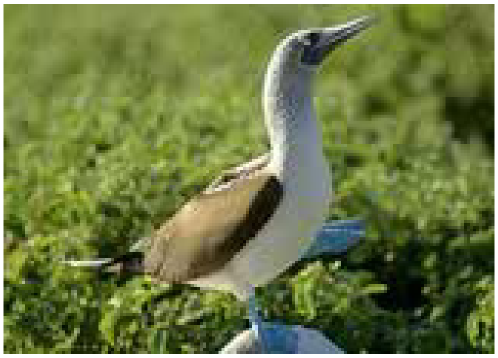


Figure 2: Secret Image

In order to evaluate the performance of the proposed method, stego-images from the LSB, MSB and the proposed Hybrid method were compared using MSE, PSNR and encoding time metrics The methods were also tested with increased sizes of the images. Figures 3a, 3b, 4a and 4b show the differences between the Least Significant Bit (LSB), Most Significant Bit (MSB) and the combined algorithm after embedding messages in them.

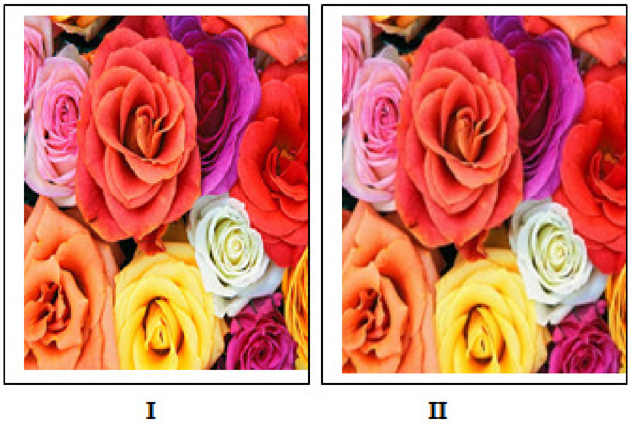


Figure 3a (560 x 448 pixels rose.jpg hiding an image): (I) Original image (II) Stego-image using LSB

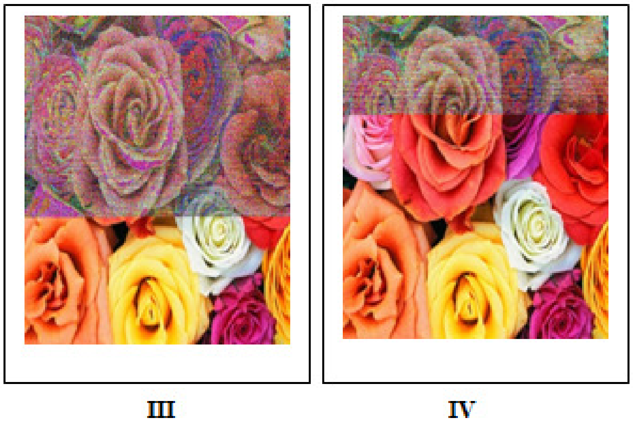


Figure 3a (560 x 448 pixels rose.jpg hiding an image): (III) Stego-image using MSB (IV) Stego-image using Combined LSB-MSB

Using roses.jpg with dimension 560 x 448 pixels as cover image and the blue-footed booby bird image as message, it can be seen from image II of figure 3a that there are no noticeable differences between the original cover image and the resultant image after hiding in the Least Significant Bit. Images 3a (III) and 3a (IV) show noticeable differences when compared to the original cover image using Most Significant Bit and combined LSB-MSB algorithms respectively. However, MSB (3a III) shows much difference.

Increasing the dimension of roses.jpg to 5040 x 4032 pixels, the payload capacity increases for MSB and proposed algorithm (figure 3b (III & IV). Therefore, the larger the cover image the more data that can be stored.

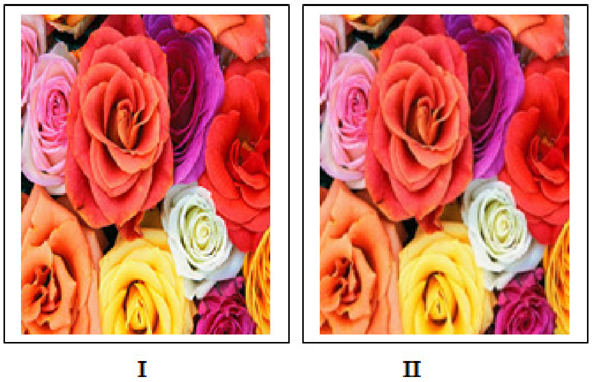


Figure 3b (5040 x 4032 pixels rose.jpg hiding an image): (I) Original image (II) Stego-image using LSB Combined LSB-MSB

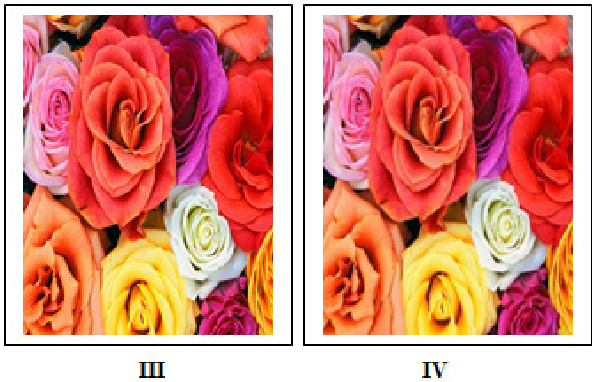


Figure 3b (5040 x 4032 pixels rose.jpg hiding an image): (III) Stego-image using MSB (IV) Stego-image using Combined LSB-MSB

Figure 4a shows the output of the newly created stego-images after hiding text with a file size of 30.4kb (31,160 bytes) in an image in PNG format. The dimension of the cover image, giraffe.png is 750 x 1125 pixels. Image 4a (II) showed no noticeable difference when compared to the original cover image after embedding text using LSB algorithm. The differences are noticeable at the top sections of Figures 4a III and IV for MSB and the combined algorithms respectively.

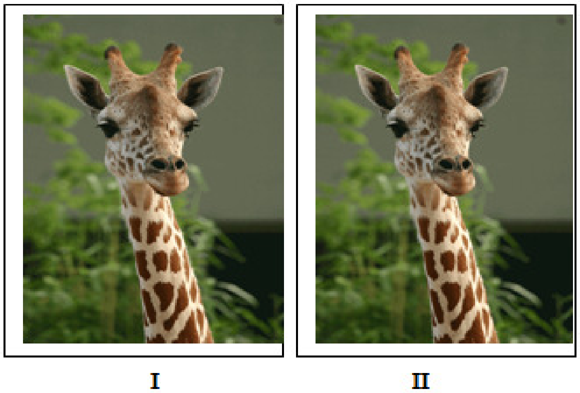


Figure 4a (750 x 1125 pixels giraffe.png hiding text): (I) Original image (II) Stego-image using LSB

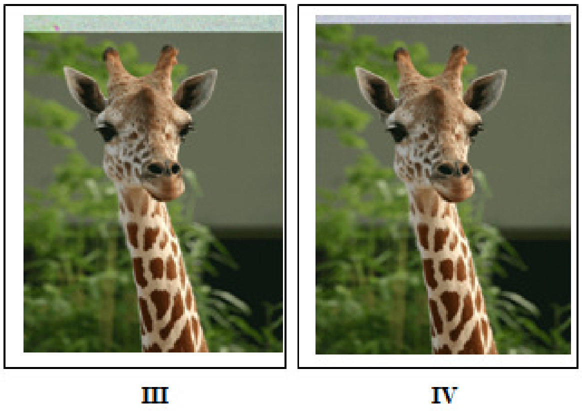


Figure 4a (750 x 1125 pixels giraffe.png hiding text): (III) Stego-image using MSB (IV) Stego-image using Combined LSB-MSB

Increasing the dimension of the PNG file to 6750 x 10125 pixels produced a stego-image indistinguishable from the original cover image when viewed with the human eyes for the three algorithms (Figure 4b).

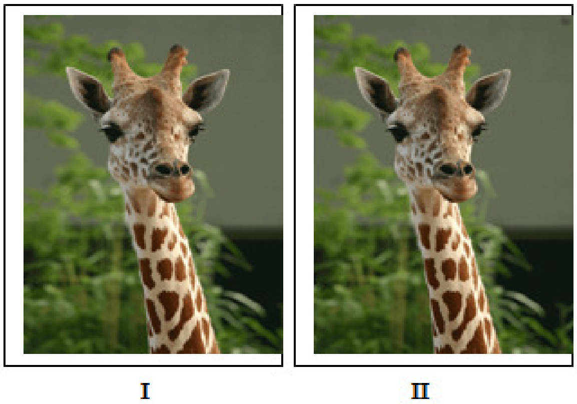


Figure 4b (6750 x 10125 pixels giraffe.png hiding text): (I) Original image (II) Stego-image using LSB

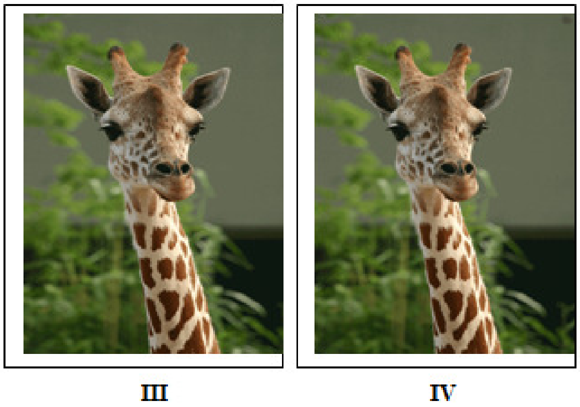


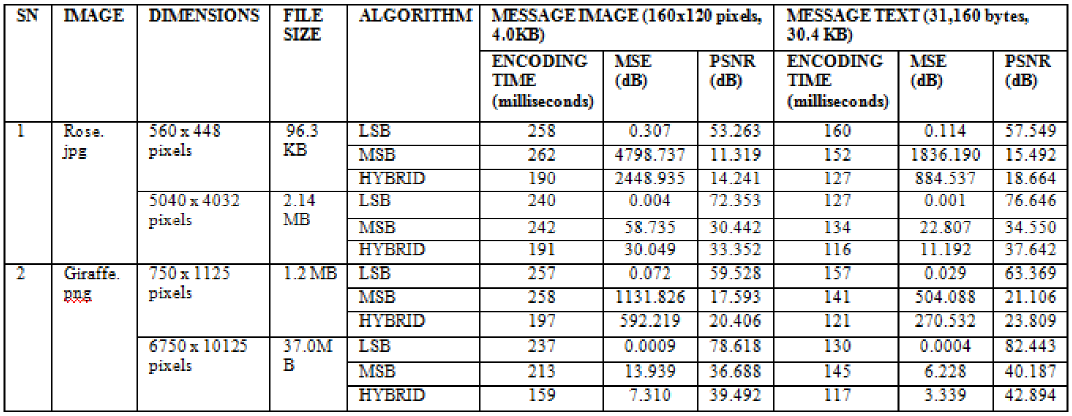
Figure 4b (6750 x 10125 pixels giraffe.png hiding text): (III) Stego-image using MSB (IV) Stego-image using Combined LSB-MSB

5. HELPFUL HINTS

Table 1 shows the MSE, PSNR and encoding times of the cover images for image and text embedding. It can be seen that a lower MSE value and a higher PSNR value for LSB algorithm as compared to the MSB and proposed Hybrid algorithms for both image and text were obtained. This results into a better image quality since the lower the MSE value and the higher the PSNR value, the better the quality of the image and hence imperceptibility is improved.

Although the embedding capacity of the proposed method (Hybrid) is low compared to LSB, the proposed method gives better performance in all the parameters than MSB. The stego-image generated after embedding the secret message in the cover image is almost identical to the original image. However, when the sizes of the cover images were increased, the image quality of the proposed algorithm increased, which means that the larger the cover image, the better the hiding capacity. Also, the encoding times of the proposed Hybrid algorithm for various sizes of the different images were lesser compared to other methods.

Table 1: Values of Encoding Time, MSEs and PSNRs of stego-images in which image and text is embedded respectively.



SN	IMAGE	DIMENSIONS	FILE SIZE	ALGORITHM	MESSAGE IMAGE (160x120 pixels, 4.0KB)			MESSAGE TEXT (31,160 bytes, 30.4 KB)		
					ENCODING TIME (milliseconds)	MSE (dB)	PSNR (dB)	ENCODING TIME (milliseconds)	MSE (dB)	PSNR (dB)
1	Rose. jpg	560 x 448 pixels	96.3 KB	LSB	258	0.307	53.263	160	0.114	57.549
				MSB	262	4798.737	11.319	152	1836.190	15.492
				HYBRID	190	2448.935	14.241	127	884.537	18.664
		5040 x 4032 pixels	2.14 MB	LSB	240	0.004	72.353	127	0.001	76.646
				MSB	242	58.735	30.442	134	22.807	34.550
				HYBRID	191	30.049	33.352	116	11.192	37.642
2	Giraffe. png	750 x 1125 pixels	1.2 MB	LSB	257	0.072	59.528	157	0.029	63.369
				MSB	258	1131.826	17.593	141	504.088	21.106
				HYBRID	197	592.219	20.406	121	270.532	23.809
		6750 x 10125 pixels	37.0M B	LSB	237	0.0009	78.618	130	0.0004	82.443
				MSB	213	13.939	36.688	145	6.228	40.187
				HYBRID	159	7.310	39.492	117	3.339	42.894

Figures 5a and 5b shows the bar chart of results obtained with rose.jpg of sizes 560 x 448 and 5040 x 4032 pixels after embedding an image using LSB, MSB and Hybrid algorithms respectively. The LSB algorithm had the lowest MSE and highest PSNR values while the proposed combined algorithm had the lowest encoding time.

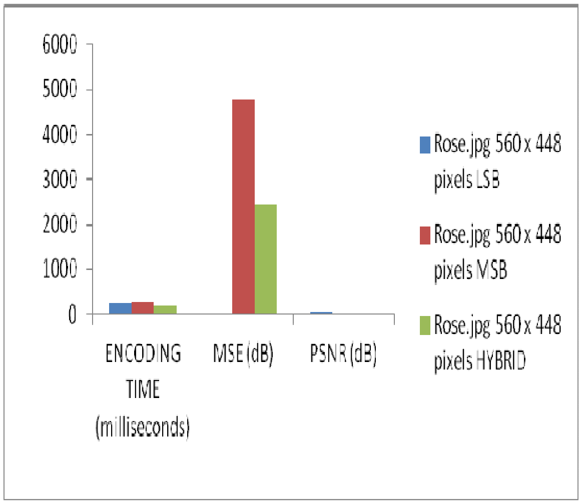


Figure 5a: The Encoding Time (ms), MSE (dB) and PSNR (dB) of 560 x 448 pixels rose.jpg

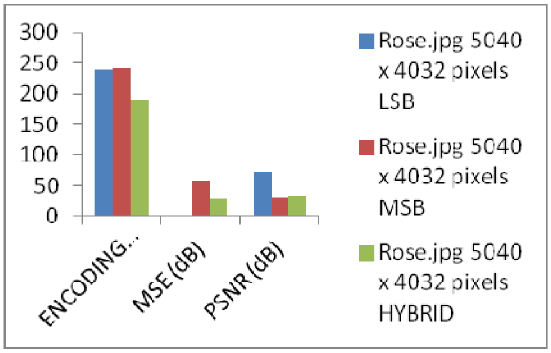


Figure 5b: The Encoding Time (ms), MSE (dB) and PSNR (dB) of 5040 x 4032 pixels rose.jpg

Figures 6a and 6b shows the bar chart of results obtained with giraffe.png of sizes 750 x 1125 and 6750 x 10125 pixels after embedding text using LSB, MSB and com algorithms respectively. The LSB algorithm also had the lowest MSE value and the highest PSNR value while the proposed Hybrid algorithm had the lowest encoding time value.

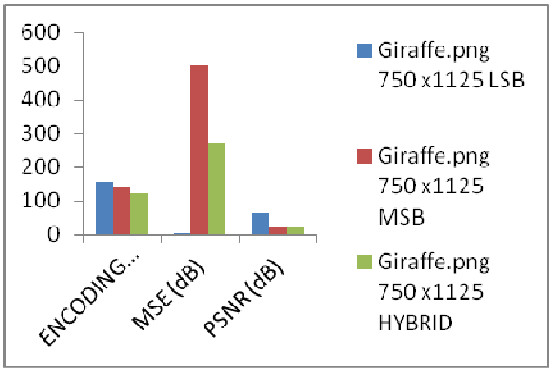


Figure 6a: The Encoding Time (ms), MSE (dB) and PSNR (dB) of 750 x 1125 pixels giraffe.png

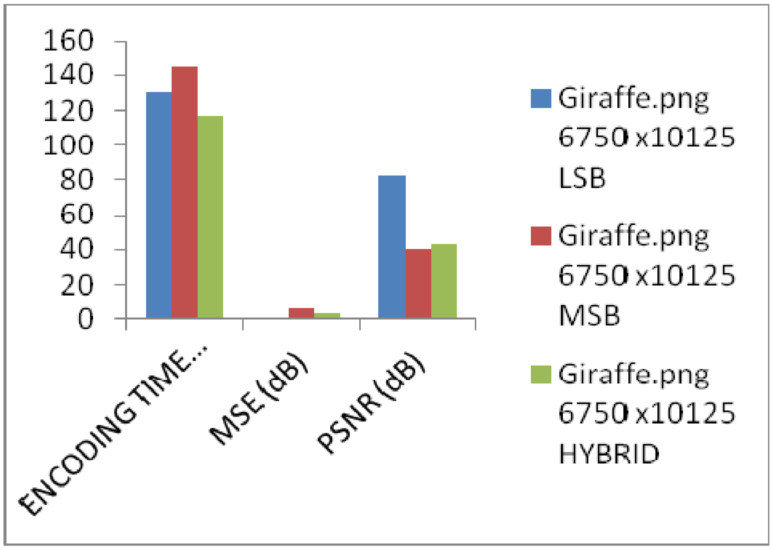


Figure 6b: The Encoding Time (ms), MSE (dB) and PSNR (dB) of 6750 x 10125 pixels giraffe.png

Overall, LSB gives a best performance in terms of MSE and PSNR than the MSB and the proposed Hybrid algorithm while the proposed Hybrid algorithm gives the best performance in terms of the encoding time and better than MSB. The result obtained in this work confirms the submission of Rohit and Tarun [12] that LSB steganography is much better than MSB steganography for hiding messages. Scott [11] paper also compared LSB replacement scheme with MSB replacement scheme and asserted that MSB produced noticeable differences to the cover during the most significant bit replacement.

6. CONCLUSION

In this work, a Hybrid (LSB-MSB) algorithm was developed for embedding images and text into images. The Hybrid algorithm suggests the embedding of secret message bits into the least significant bit and the most significant bit of the cover image. The performance measure of image quality due to embedding for LSB, MSB and the proposed Hybrid technique was evaluated based on three error metrics; Mean-Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR) and the time it takes each algorithm to embed a message in a cover image. The results presented and analyzed show that the stego-images of LSB have the highest PSNR and the lowest MSE values, making it very efficient to hide the data inside the image. However, based on the encoding time, the combined LSB-MSB algorithm takes lesser time in embedding than LSB and MSB.

Conclusively, LSB algorithm gives a better performance than the combined LSB-MSB algorithm but a larger file size of the cover image makes the combined algorithm produces images with good quality. Nevertheless, the proposed Hybrid algorithm produced better image quality than MSB algorithm.

In the future work, the security of using the hybrid algorithm could be improved by working on the compression ratio for stronger embedding procedures and also the use of the proposed combined LSB-MSB algorithm on large sized gray-scale images.

REFERENCES

[1] Currie, D.L. & Irvine, C.E. (1996). Surmounting the Effects of Lossy Compression on Steganography, 19th National Information Systems Security Conference, Oct 1996, Baltimore, Md., USA, pp. 194 - 201

[2] Jonathan Blackledge and Oleksandr Iakovenko (2014). Resilient Digital Image Watermarking for Document Authentication, IAENG Internation Journal of Computer Science, 41 (1).

[3] Jessica Codr (2009). Unseen: An Overview of Steganography and Presentation of Associated Java Application C-Hide, Washington University in St.Louis, Department of Computer Science & Engineering, Bryan Hall, Campus Box 1045, One Brookings Drive, St. Louis, Missouri 63130, http://www.cse.wustl.edu/~jain/cse571-09/ftp/stegano/index.html

[4] Kanzariya Nitin K. and Nimavat Ashish V. (2013). Comparison of Various Images Steganography Techniques, International Journal of Computer Science and Management Research, Vol. 2, Issue 1

[5] Nagham Hamid, Abid Yahya, Badlishah Ahmad R. and Osamah Al-Qershi M. (2012). Image Steganography Techniques: An Overview, International Journal of Computer Science and Security (IJCSS), Vol. 6, Issue 3

[6] Warkentin M., M.B. Schmidt and Bekkering E. (2008). Steganography and Steganalysis, Intellectual Property Protection for Multimedia Information Technology, Chapter XIX, pp. 374 - 380

[7] El-Emam N. N (2007). Hiding a Large Amount of Data with High Security Using Steganography Algorithm, Journal of Computer Science, Vol. 3, pp. 223 - 232

[8] Chen P.Y., Wu W.E. (2009). A Modified Side Match Scheme for Image Steganography, International Journal of Applied Science & Engineering, Vol. 7 pp. 53 – 60

[9] Chang C.C. and Tseng H.W. (2004). A Steganographic Method for Digital Images using Side Match Pattern Recognition Letters, 1431-1437

[10] Wu P.C, Tsai W.H. (2003). A Steganographic Method for Images by Pixel-value Differencing, Pattern Recognition Letters, pp. 1613 - 1626

[11] Scott Bishop (2004). Steganographic Techniques using Digital Images, California State University, Hayward

[12] Rohit Garg and Tarun Gulati (2012). Comparison of Lsb & Msb Based Steganography in Gray-Scale Images, International Journal of Engineering Research & Technology (IJERT), Vol. 1, Issue 8

[13] Lokeswara Reddy V., Dr. Subramanyam A., Dr. Chenna Reddy P. (2011). Implementation of LSB Steganography and its Evaluation for Various File Formats, International Journal Advanced Networking and Applications, Vol. 02, Issue 05, pp. 868 – 872

[14] Stuti Goel, Arun Rana & Manpreet Kaur (2013). A Review of Comparison Techniques of Image Steganography, Global Journal of Computer Science and Technology Graphics & Vision, Vol.13, Issue 4, Version 1.0

[15] Kumar Shiva K.B., Raja K.B., Chhotaray R.K and Sabyasachi Pattnaik (2011). Performance Comparison of Robust Steganography Based on Multiple Transformation Techniques, International Journal Computer Technology Application, Vol. 2 (4), pp. 1035 – 1047.

Authors

Akinola Olalekan is a Senior lecturer of Computer Science at the University of Ibadan, Nigeria. He had his PhD Degree in Software Engineering from the same University in Nigeria. His research focus is on software quality assurance techniques.

Adebanke Olatidoye had her Masters Degree in Computer Science from University of Ibadan, Nigeria. She also had her first Degree in Computer Science from Ladoke Akintola University of Technology, Ogbomoso, Nigeria.