SCALABLE AND HIGHLY SECURED IMAGE STEGANOGRAPHY BASED ON HOPFIELD CHAOTIC NEURAL NETWORK AND WAVELET TRANSFORMS

B. Geetha Vani [1], Prof. E. V. Prasad [2]

[1] Research Scholar, Department of CSE, J N T University, Kakinada, AP, India.

[2] Professor, Department of CSE & Rector, J N T University, Kakinada, AP, India.

Abstract

Steganography is the science of communicating in the hidden manner. This paper presents a robust and secured Image Steganography method capable of embedding high volume of text information in digital cover-image without incurring any perceptual distortion. The method is based on compression and encryption. In order to achieve high capacity, dictionary based lossless compression techniques are used. And to achieve high security, encryption mechanism using Hopfield Chaotic Neural network is used. The message to be transmitted is compressed first using Lempel Ziv scheme technique and is encrypted by HCNN and then embedded into the image using Discrete Wavelet transforms. The proposed method is tested with different images and text of various lengths and found to be efficient, secure and has high embedding capacity.

Keywords: *Steganography, Lossless dictionary based compression technique, Lempel Ziv scheme techniques, Hopfield Chaotic Neural network, Discrete Wavelet Transforms.*

1. INTRODUCTION

Steganography is the art and science of communicating in such a way that the presence of a message cannot be detected. Due to availability of Internet throughout the world, content security is playing a major role in multimedia communication. The techniques available to achieve the goal of content security are Cryptography, Encryption and Steganography. Cryptography scrambles the message so that it cannot be understood, while Steganography hides the very existence of the message by carefully embedding it into a cover. An eavesdropper can intercept a Cryptographic message but one may not even know the existence of Steganographic communication. Encryption and Steganography achieves the same goal via different means. Encryption encodes the data so that an unintended recipient cannot determine its intended meaning. Steganography, in contrast attempts to prevent an unintended recipient from suspecting about the hidden information. Combining Encryption with Steganography allows better private communication.

One method of common Steganography technique is to hide the secret message in the least significant bits of pixels of the cover image [2, 3]. The image quality of stego image achieved by applying the LSB technique is very closer to the original one. But the drawback is it cannot survive image processing manipulations [4]. One method of LSB Steganography involves manipulating the LSB plane from direct replacement of the cover image with message bits to some type of logical or arithmetic combination between two. Several examples of LSB techniques are found [6]. This technique achieves both high capacity and low perceptibility. But it is not very sophisticated and subject to extraction by unwanted persons.

The DCT method [7] applies Discrete Cosine Transform to determine the high frequency areas and the message is embedded on these areas of digital image. Here more security can be achieved but the quality of stego image is poor. In DWT (Discrete Wavelet Transform) scheme [8] the digital image is separated into non overlapping blocks and the message is embedded on those blocks. The wavelet coefficients in low frequency sub bands are more important than the high frequency sub bands.

The design issues of Steganography are imperceptibility, robustness, security and high capacity. There is always a trade-off between the three main parameters i.e. capacity, imperceptibility and robustness. If any one of these parameters is changed then the other two gets affected. Though the capacity, robustness, and security issues are driven by the application need and its priorities, one has to optimize all the parameters to get the best results. In the proposed work, main focus is given on high capacity and adding security to the core embedding mechanism to make it difficult for an attacker to detect the existence of evidence of embedding. In this method, based on the length of message, suitable lossless dictionary based compression technique is applied. The compressed text is encrypted by using Hopfield chaotic neural network and then embedded into the cover image by Discrete Wavelet Transform technique.

The rest of the paper is organized as follows. In Section 2, the techniques involved are presented. In Section 3, the proposed system is described. In Section 4, experimental results of proposed approach are shown. Concluding remarks are provided in section 5.

2. TECHNIQUES INVOLVED

2.1 *DATA COMPRESSION:*

Data compression is a process of representing information in the compressed form. It involves encoding of data into fewer bits than the original representation. It helps to communicate more information than the uncompressed form. Data compression may be lossy or lossless based on the compression technique. Mostly for the text files, lossless compression is recommended. In this paper dictionary based compression techniques are considered.

2.1.1 LZW *COMPRESSION*

LZW (Lempel–Ziv–Welch) is a compression algorithm which belongs to the LZ78 family of Lempel Ziv scheme [11]. It works for any type of data. LZW creates a dictionary which is a table of string which occurs commonly in the original plain text and replaces the reoccurring text with the reference of the existing data in the dictionary. This dictionary is formed during compression at the same time at which the data is encoded and during decompression at the same time the data is decoded. LZW technique is a adaptive compression algorithm which decompress the data at the receiver side without the transmission of the dictionary generated during the compression to the receiver.

2.1.2 DEFLATE COMPRESSION

Deflate is a compression technique that combines LZ77 and Huffman together [12]. The dictionary based algorithm similar to LZ77 is used for recurring sequences of the text. The Huffman code is used for entropy encoding. In simple words, it is a compression technique of two stages. In the first stage the dictionary based technique for the reoccurrence of the string is used. In the second stage the commonly used strings is replaced with the shorter representations and the less commonly used strings is replaced with the longer representation.
In the First stage, if the duplicate string is found from the given string then the current occurrence of the string is replaced with the pointer of the previous occurrence in the form of a distance, length pair. Distances are limited to 32K bytes and the length is limited to 256 bytes. Duplicate strings are found in the hash table. The hash table is searched starting from the commonly used strings to less commonly used strings thus taking the advantage of Huffman coding.
In the second stage, Huffman coding method is used to create an unprefixed tree of non-overlapping intervals, where the length of each sequence is inversely proportional to the probability of the symbol that need to be encoded. The issue of the bit sequence of the encoded symbol depends inversely with the commonality of the input string.

2.2 *HOPFIELD CHAOTIC NEURAL NETWORK BASED ENCRYPTION*

The encryption methodology adopted for encrypting text characters plays a vital role in deciding the embedding capacity and the level of robustness and security of the entire Steganographic system.

Hopfield Chaotic Neural network is a suitable environment for cryptography because of some interesting properties like ergodicity, sensitive dependence of initial conditions and control parameters and high speed of information transmission. Yu et al. designed a delayed chaotic neural network based cryptosystem, which makes use of the chaotic trajectories of two neurons to generate basic binary sequences for encrypting plaintext. In Chaotic Neural Network, the weights and biases are determined by a chaotic sequence, a binary random deterministic sequence, and is used to mask or to scramble the original information [9]. The encryption algorithm [1] is used for obtaining the cipher text. The Chaotic neural network consumes less computational power and the sequence generated using this is unpredictable leading to highly secured and efficient in terms of power.

2.3 DISCRETE WAVELET TRANSFORMS

The simplest of DWT is Haar - DWT where the low frequency wavelet coefficients are generated by averaging the two pixel values and high frequency coefficients are generated by taking half of the difference of the same two pixels [8]. For 2D-images, applying DWT will result in the separation of four different bands. LL is the lower resolution approximation of the image. HL is the horizontal, LH is the vertical, HH is the diagonal component.

With the DWT, the significant part (smooth parts) of the spatial domain image exist in the approximation band that consists of low frequency wavelet coefficients and the edge and texture details usually exist in high frequency sub bands, such as HH, HL, and LH. The secret data are embedded to the High Frequency components as it is difficult for the human eye to detect the existence of secret data.

3. PROPOSED METHOD

The proposed method consists of two stages.

Stage 1: Text compression Stage:

The length L of the text to be transmitted is detected and checked with the threshold. The threshold is set to 2500 bytes. Then message format of information is framed with three fields. First field indicates the compression type, second field set to the length of text and third field consists of text itself. The first field in the message is set to '1' if length of text is less than or equal to 2500 bytes and it is set to '2' if length is greater than 2500 bytes. The message format is as shown in the fig.1

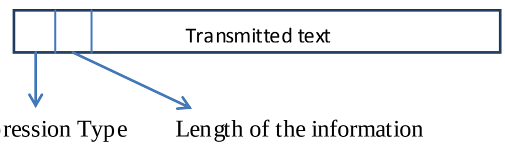

Fig.1

When the compression type is set to '1', the information is compressed using the LZW technique[11] since the LZW technique works efficiently and have considerable processing time with better compression ratio till 2500 bytes. When the size of the information increases, the dictionary size increases and the processing time is gradually increased. When the compression type is set to '2', the information is compressed using the deflate algorithm. Deflate algorithm works well for large volumes of data [12]. The compression ratio keeps improving until some saturation level where the change is undetectable. The compression ratio depends on the length of the text.

At the receiver end the system selects the decompression technique accordingly for the decrypted data based on the first byte of the message format.

Stage 2: Encryption and Embedding Stage:

In this stage the compressed text is encrypted using Hopfield chaotic neural network consisting of two neurons. The encrypted text is then embedded into the cover image by using DWT technique. The flow diagram of encoding and decoding process is shown in Fig.2. The technique is highly secured in many ways. The use of chaotic neural network [1] and embedding the encrypted text using DWT in the transmitter side to get the stego image ensures high security. At the receiver side, the embedded secret text is extracted from stego image and original image is extracted using corresponding decompression technique.

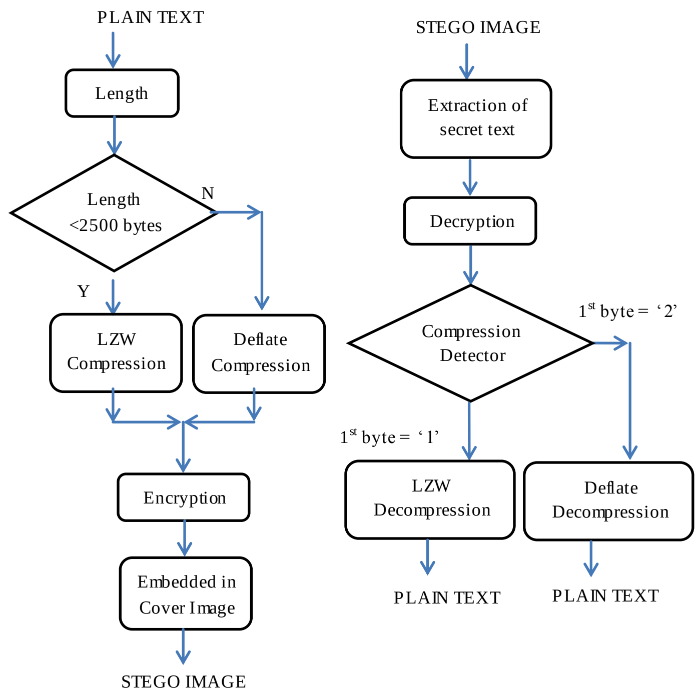

Fig. 2 Process diagram of the proposed Algorithm

The Proposed Steganography algorithm is as follows

Input : cover image and input text

Output : Stego image

Algorithm: The algorithm is performed in five steps

[1] The Detector calculates the length L of input text
[2] The message format to be embedded is formed by setting the first byte to '1' if L < 2500 bytes otherwise to '2'and the second byte to the length of text and then the text is appended.
[3] Based on the length of the text, LZW or Deflate Compression technique is used for compressing the text.
[4] The compressed text is encrypted using Hopfield chaotic neural network.
[5] The encrypted data is then embedded into the cover image by using Discrete wavelet transforms thus forming the stego image.

The inverse of above procedure is applied to stego image to obtain the embedded information correctly.

4. RESULTS AND ANALYSIS

Experiments are performed and the efficiency of the proposed algorithm is recorded and tabled. A GUI was developed using Matlab 7.14.0.739.

The performance of the compression technique is tested with different test files from the Calgary Corpus, which is considered as the set of standard files used for testing the data compression technique. The compression technique is tested in terms of compression ratio, compression size and the time elapsed for compression. Compression size is the size of the compressed file in bits after compression. Compression Ratio is the percentage obtained by dividing the compression size in bits by the original file size in bits. Time elapsed is the time in milliseconds in compressing the file.

The Quantitative performance of the proposed algorithm is evaluated based on Peak signal to noise ratio (PSNR) and Mean Square Error (MSE) as expressed below.

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2}{MSE}\right) \quad \text{----------- Eq.1}$$

$$\text{MSE} = \frac{\sum_i \sum_j (r_{ij} - x_{ij})2}{M \times N} \quad \text{----------- Eq.2}$$

Where *r* refers to Original image, *x* denotes restored image, M x N is the size of Processed image [1].

The qualitative performance of the proposed algorithm is tested on various images such as Lena, Cameraman, Barbara, Boat, Pepper, House (Images are chosen as per the details of the image). The secret text of five different lengths i.e 1000, 1500, 2000, 2500, 3000 bytes is taken for testing. Performance has been evaluated in terms of PSNR and MSE.

TABLE I COMPARISON OF COMPRESSION RATIO, ELAPSED TIME, FILE SIZE AFTER COMPRESSION FOR STANDARD TEST FILES

Text file	Original Size	LZW Elapsed time	LZW compression Ratio (%)	LZW compression size	Deflate Elapsed time	Deflate compression Ratio (%)	Deflate compression size
Bib.txt	111261	11.3101	2.02	35898	0.1404	3.98	70783
Book1.txt	768771	90.0906	2.12	260540	1.1232	4.69	576898
Paper1.txt	53161	6.5520	2.44	20785	0.1092	4.12	35043
Progc.txt	39611	4.3212	2.57	16311	0.0468	3.82	24190
Trans.txt	93695	11.2945	2.23	33433	0.0936	2.99	44767

The parameters like compression ratio, compressed file size and the time required to compress the file, obtained by using LZW and deflate algorithm are shown in Table I. The standard test files are used for the comparison in Table I. From Table I it can be noticed that LZW algorithm consumes more time on compressing but the size of the compressed file is less than the other techniques and it provides a better compression ratio. The processing time of deflate algorithm is much lesser than the LZW technique and found to be very efficient for larger size files of larger sizes.

TABLE II COMPRESSION RATIO, ELAPSED TIME, FILE SIZE AFTER COMPRESSION FOR TEST FILES

Text file	Original Size (Bytes)	LZW Elapsed time	LZW compression Ratio (%)	LZW compression size	Deflate Elapsed time	Deflate compression Ratio (%)	Deflate compression size
Input1.txt	1000	0.1716	3.1844	511	0.0312	4.6484	743
Input2.txt	1500	0.2652	2.8227	679	0.0120	3.4398	825
Input3.txt	2000	0.3276	2.6107	837	0.0312	2.8358	907
Input4.txt	2500	0.3900	2.4285	973	0.0186	2.4385	975
Input5.txt	3000	0.4680	2.2903	1101	0.0232	2.1924	1052

Table II provides results obtained by varying the size of the files. These files are used in our proposed algorithm for embedding in the Image for obtaining the stego image.

TABLE III PERFORMANCE ANALYSIS OF PROPOSED ALGORITHM ON VARIOUS IMAGES WITHOUT COMPRESSION

Input	Size of embedded message	Barbara (512*512)		Boat (512*512)		Lena (512*512)		Camera man (256*256)		House (256*256)	
		MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
Input1.txt	1000 Bytes	1.2607	47.1587	0.82231	49.0144	0.06623	59.9537	0.89994	48.6226	0.06430	60.0827
Input2.txt	1500 Bytes	2.60317	44.0098	0.92089	48.5227	0.13560	56.842	1.29515	47.0416	0.19900	55.1762
Input3.txt	2000 Bytes	3.99932	42.1449	1.15888	47.5244	0.23409	54.4709	2.51324	44.1625	0.67215	49.8901
Input4.txt	2500 Bytes	2.02816	45.0938	1.7524	45.7285	0.32746	53.0131	6.34929	40.1345	2.30499	44.5381
Input5.txt	3000 Bytes	6.5018	40.0345	2.00832	45.1365	0.36917	52.4924	8.49779	38.8717	2.77173	43.7373

Table III provides the details of the performance of the Steganography algorithm proposed in [1] on various test images without any compression technique. Table IV & V contains the result comparing the performance of the proposed algorithm with LZW and Deflate compression technique.

TABLE IV PERFORMANCE ANALYSIS OF PROPOSED ALGORITHM ON VARIOUS IMAGES WITH LZW COMPRESSION

Input	Size of embedded message	Barbara (512*512)		Boat (512*512)		Lena (512*512)		Camera man (256*256)		House (256*256)	
		MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
Input1.txt	1000 Bytes	0.00534	70.8858	0.53026	50.9199	0.01808	65.5906	0.53311	50.8966	0.03213	63.095
Input2.txt	1500 Bytes	0.19651	55.2309	0.64450	50.0726	0.02711	63.8327	0.64930	50.0403	0.04049	62.0609
Input3.txt	2000 Bytes	0.53191	50.9064	0.67856	49.8489	0.03965	62.1815	0.81617	49.047	0.05374	60.8617
Input4.txt	2500Bytes	1.0531	47.9401	0.81822	49.0361	0.04756	61.3915	0.88102	48.7149	0.06362	60.1282
Input5.txt	3000 Bytes	1.45983	46.5218	0.82965	48.9758	0.07260	59.5554	0.94987	48.3881	0.08996	58.624

Table VI discusses the comparison of processing time obtained for embedding the LZW and Deflate compressed information into the image.

TABLE V PERFORMANCE ANALYSIS OF PROPOSED ALGORITHM ON VARIOUS IMAGES WITH DEFLATE COMPRESSION

Input	Size of embedded message	Barbara (512*512)		Boat (512*512)		Lena (512*512)		Camera man (256*256)		House (256*256)	
		MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
Input1.txt	1000 Bytes	0.29945	53.4014	0.67189	49.8918	0.03387	62.8661	0.72175	49.5809	0.04232	61.8985
Input2.txt	1500 Bytes	0.52640	50.9516	0.67786	49.8534	0.03898	62.2557	0.81492	49.0536	0.05282	60.9363
Input3.txt	2000 Bytes	0.64773	50.0508	0.68412	49.8134	0.04238	61.8931	0.87702	48.7347	0.06057	60.3417
Input4.txt	2500 Bytes	1.05519	47.9315	0.81839	49.0352	0.04782	61.3686	0.88429	48.6988	0.06375	60.1199
Input5.txt	3000 Bytes	1.40612	48.6846	0.82586	48.9957	0.06848	59.8086	0.93949	48.4358	0.07009	59.7077

TABLE VI COMPARISON OF PROCESSING TIME FOR EMBEDDING THE LZW AND DEFLATE COMPRESSED MESSAGE INTO IMAGE

Input	Size Of Embedded Message	Barbara (512*512)		Boat (512*512)		Lena (512*512)		Camera man (256*256)		House (256*256)	
		LZW	Deflate	LZW	Deflate	LZW	Deflate	LZW	Deflate	LZW	Deflate
Input1.txt	1000 Bytes	3.6348	3.3228	3.7752	3.4320	3.1200	3.7284	0.8580	1.1388	0.8736	1.0764
Input2.txt	1500 Bytes	4.2120	3.1200	4.2588	3.7596	3.3228	3.7752	0.9516	0.9204	1.0296	1.0140
Input3.txt	2000 Bytes	4.3836	3.2916	3.7128	3.6504	3.5256	3.3852	0.9984	1.0764	1.0340	1.2480
Input4.txt	2500 Bytes	3.5568	3.3696	3.7596	3.2136	4.0092	3.4788	1.0296	1.1076	1.0764	1.2636
Input5.txt	3000 Bytes	4.5240	3.9780	3.3228	3.6660	3.5256	2.7612	1.1076	1.1544	1.0920	1.1076

TABLE VII COMPARISON OF PERFORMANCE OF PROPOSED ALGORITHM ON VARIOUS IMAGES

Input	Size Of Embedded Message	Barbara (512*512)		Boat (512*512)		Lena (512*512)		Camera man (256*256)		House (256*256)	
		MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
Input1.txt	1000 Bytes	0.00534	70.8858	0.53026	50.9199	0.01808	65.5906	0.53311	50.8966	0.03213	63.095
Input2.txt	1500 Bytes	0.19651	55.2309	0.64450	50.0726	0.02711	63.8327	0.64930	50.0403	0.04049	62.0609
Input3.txt	2000 Bytes	0.53191	50.9064	0.67856	49.8489	0.03965	62.1815	0.81617	49.047	0.05374	60.8617
Input4.txt	2500 Bytes	1.05519	47.9315	0.81839	49.0352	0.04782	61.3686	0.88429	48.6988	0.06375	60.1199
Input5.txt	3000 Bytes	1.40612	48.6846	0.82586	48.9957	0.06848	59.8086	0.93949	48.4358	0.07009	59.7077

Table VII discuss the performance of the proposed algorithm and Table VIII discusses the processing time for embedding the compressed message into the cover image and is graphically represented in Figure12.

TABLE VIII PROCESSING TIME FOR EMBEDDING COMPRESSED MESSAGE INTO IMAGE

Input	SIZE OF EMBEDDED MESSAGE	Barbara (512*512)	Boat (512*512)	LENA (512*512)	Camera man(256*256)	House (256*256)
Input1.txt	1000 Bytes	3.6348	3.7752	3.1200	0.8580	0.8736
Input2.txt	1500 Bytes	4.2120	4.2588	3.3228	0.9516	1.0296
Input3.txt	2000 Bytes	4.3836	3.7128	3.5256	0.9984	1.0340
Input4.txt	2500 Bytes	3.3696	3.2136	3.4788	1.1076	1.2636
Input5.txt	3000 Bytes	3.9780	3.6660	2.7612	1.1544	1.1076

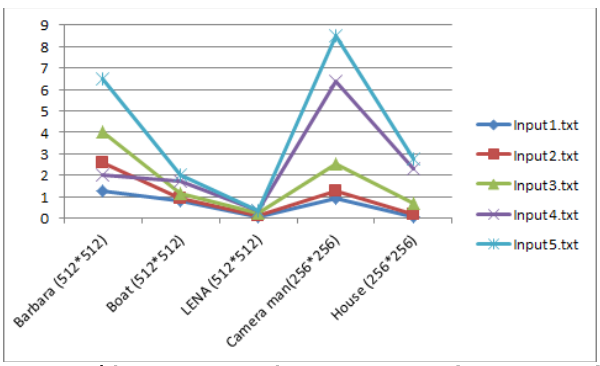

Fig.3 MSE of the stego image without compression technique proposed in [1]

Fig.3 shows the MSE of the stego image with the cover image for the algorithm without compression technique proposed in [1].

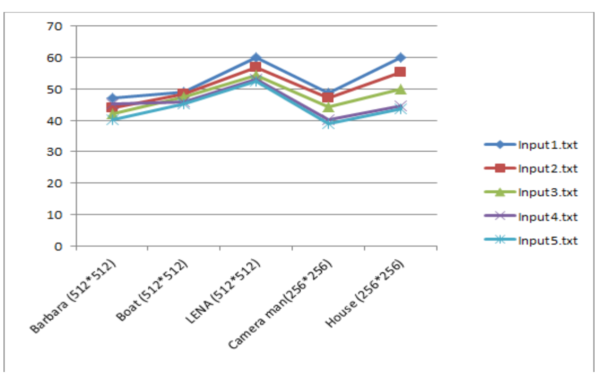

Fig.4 PSNR of the stego image without compression technique proposed in [1]

Fig.4 shows the PSNR of the stego image with the cover image for the algorithm without compression technique proposed in [1].

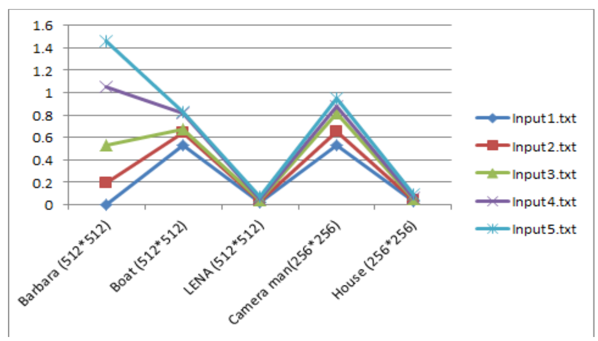

Fig.5 MSE of the stego image of the algorithm with LZW technique

Fig.5 shows the MSE of the stego image with the cover image for the proposed algorithm with LZW as the compression technique.

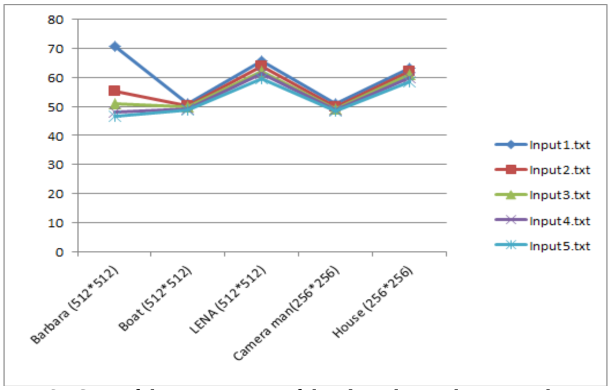

Fig.6 PSNR of the stego image of the algorithm with LZW technique

Fig.6 shows the PSNR of the stego image with the cover image for the proposed algorithm with LZW as the compression technique.

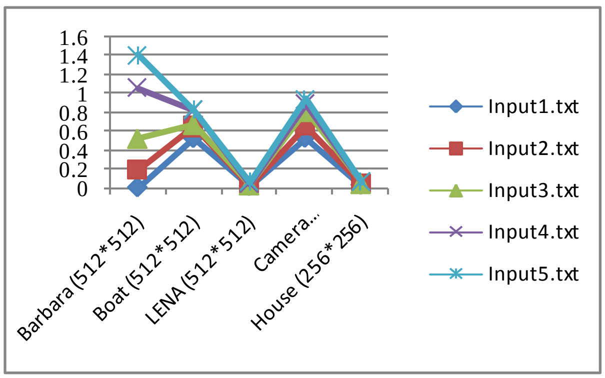

Fig.7 MSE of the stego image of the algorithm with deflate compression technique

Fig.7 shows the MSE of the stego image with the cover image for the proposed algorithm with deflate as the compression technique.

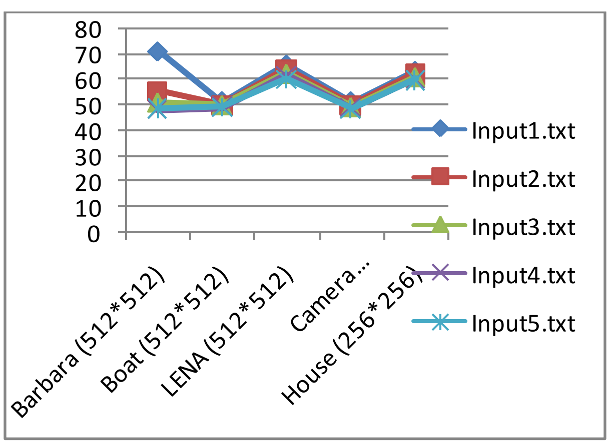

Fig.8 PSNR of the stego image of the algorithm with deflate technique

Fig.8 Shows the PSNR of the stego image with the cover image for the proposed algorithm with deflate as the compression technique.

From the above graphs, it is observed that better PSNR and low MSE is obtained for the proposed algorithm.

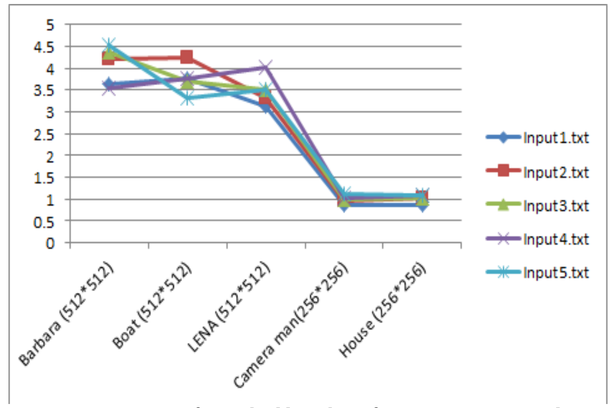

Fig.9 Processing time for embedding the information compressed using LZW compression technique.

Fig.9 shows the processing time for embedding the information compressed using LZW compression technique. Fig.10 shows the processing time for embedding the information compressed using deflate as the compression technique.

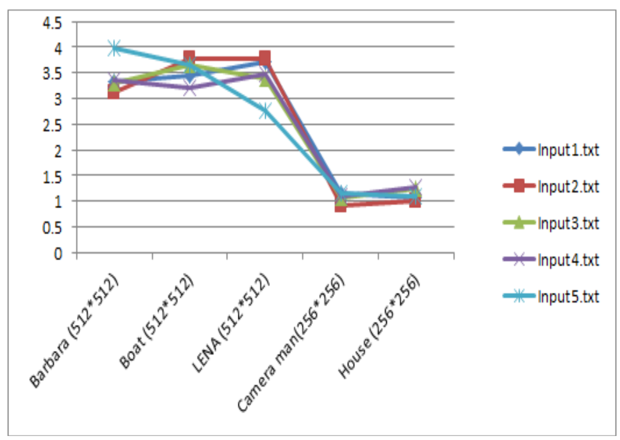

Fig.10 Processing time for embedding the information compressed using deflate compression technique.

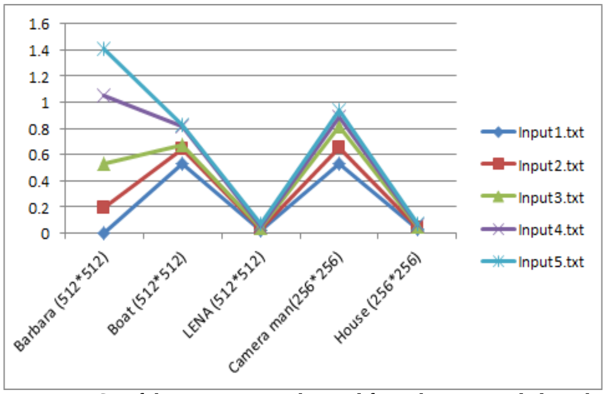

Fig.11 MSE of the stego image obtained from the proposed algorithm

Fig.11 shows the MSE of the stego image with the cover image for the proposed algorithm.

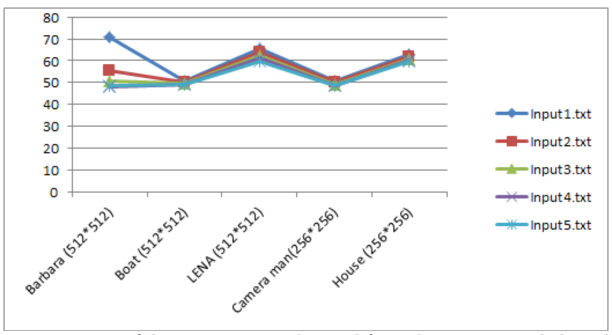

Fig.12 PSNR of the stego image obtained from the proposed algorithm

Fig.12 shows the PSNR of the stego image with the cover image for the proposed algorithm.

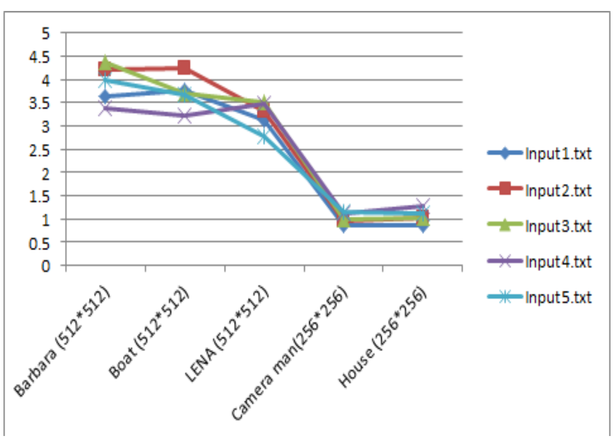

Fig.13 Processing time for the proposed Algorithm.

Fig.13 shows the processing time required for the proposed algorithm.

The Images used for Qualitative Performance is shown below.

(a) Barbara (512*512) (b) Boat (512*512) (c) Lena (512*512) (d) House (256*256) (e)Cameraman (256*256)

Fig.14 Images used for Qualitative Performance

5. CONCLUSION

In this Paper, a novel method of Image Steganography algorithm that uses dictionary based compression techniques is presented. The secret data of various lengths is compressed using LZW technique or Deflate algorithm based on the size of the information and is embedded into the cover image after encryption. The qualitative performance of the proposed system is analyzed. The PSNR, MSE varies depending on the amount of data embedded in the image and the size of the image and better PSNR and low MSE values are obtained with the proposed algorithm. The performance of the system with compression and without compression of text is verified. The results show that the capacity of information embedded is improved by 40%. The Proposed system shows better performance in terms of both capacity and security.

REFERENCES

[1]. B.Geethavani, E.V.Prasad. "*High Secure Image Steganography Based on Chaotic Neural Network*". IJCSNS, Volume13, No.3, pp1-6, March 2013.

[2]. F.A.P.Petitcolas, R.J.Anderson & M.G.Khan, "Information hiding – A survey", IEEE Proceedings, Vol 87, no 7, pp.1062-1078, July 1999.

[3]. Chan.C.K. and Cheng.L.M. "*Hiding data in image by simple LSB substitution*". Pattern Recognition, pp 469 – 474, Aug 2003.

[4]. Pergamon J. Fridrich, M. Goljan, R. Du, Detecting LSB Steganography in color and grayscale images, IEEE Multimedia Special Issue on Security pp22–28,Oct-Nov 2001.

[5]. Fahad Ullah, Mohammad Naveed, Mohammad Inayatullah Babar, Faisal Iqbal, "*Novel use of Steganography for Both Confidentiality and Compression*", IACSIT International Journal of Engineering and Technology, Vol.2, No.4, pp. 361-366, Aug 2010.

[6] H. Arafat Ali. "*Qualitative Spatial Image Data Hiding for Secure Data Transmission*". GVIP Journal, 7(1): pp 35-43, July 2007.

[7] R.J.Anderson & F.A.P.Petitcolas, "On the Limits of Steganography", IEEE Journal on selected Areas in Communication, vol.16, No.4, pp474-481, 2004.

[8] Chin-Chen Chang, Yung-Chen Chou, "Quantization Index Modulation using Vector Quantization with DWT based State-Codebook search", International conference on Intelligent Information hiding and Multimedia, pp. 549-554, 2006.

[9] Yu W, Cao J. "*Cryptography based on delayed neural networks"* Phys Lett A, Vol 8, pp. 333-356, 2006.

[10] Juneja M. and Sandhu P.S., *"Designing of robust image Steganography technique based on LSB insertion and encryption,"* Advances in Recent Technologies in Communication and Computing, pp. 302-305, Oct. 2009.

[11] Haroon Altarawneh, Mohammad Altarawneh *"Data compression techniques on text files: A comparison study"* International Journal of Computer Applications (0975 – 8887) Volume 26, No.5, July 2011.

[12] Senthil Shanmugasundaram, Robert Lourdusamy "*A Comparative Study Of Text Compression Algorithms"* International Journal of Wisdom Based Computing, Vol. 1 (3), pp 68-76, Dec 2011.

B.Geetha Vani has received the B.Tech degree in Computer Science and Engineering from JNTU Hyderabad in 1993 and M. Tech degree in Computer Science and Engineering from JNTU Hyderabad in 2003. Currently pursuing Ph.D from JNTU Kakinada, India. Her Research interests include Theory of Computation, Artificial Neural Networks, Image Processing and Information Security.

Dr.E.V.Prasad has received Ph.D degree in Computer Science and Engineering from IIT, Roorke, India. He is having 34 years of experience in teaching. He joined in JNTU College of Engineering in the year 1979 and served in various positions like Head of the Department, Vice Principal, Principal, Director of IST, Registrar and presently Rector, JNTU Kakinada, India. He has taught over 16 courses in CSE and has guided 6 Ph.D students successfully and presently supervising 9 Ph.D candidates. He is the Co author of six books and published more than six dozen papers in national and International journals and conferences. His research interests include Parallel Computing, Data Mining, and Information Security.