Journal of Intelligent & Fuzzy Systems 45 (2023) 11135–11147
DOI:10.3233/JIFS-234314
IOS Press

Efficient octagonal compression of multimedia data using LZW-OMCA compressor for secured data transmission

Tammineni Sreelatha[a,*], M. Maheswari[b], G. Ravi[c], N. Manikanda Devarajan[d] and M. Arun[e]

[a]*Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur District, Andhra Pradesh, India*

[b]*Department of Computational Intelligence, SRM Institute of Science and Technology, Kattankulathur, Chennai, India*

[c]*Department of ECE, Sona College of Technology, Salem, Tamilnadu, India*

[d]*Department of Electronics and Communication Engineering, Malla Reddy Engineering College, Medchal - Malkajgiri District, Telangana, India*

[e]*Department of ECE, Panimalar Engineering College, Tamilnadu, Chennai, India*

Abstract. Data compression is the ancestor of image compression, which uses fewer bits to represent the same picture. It is categorised as lossy or lossless depending on the quality required. In a lossless compression situation, no information is lost during the decompression process. Data loss is possible with the lossy technique since it is not reversible. In an effort to boost compression efficiency while maintaining a high xiv reconstruction quality of picture, near lossless approaches have evolved. The medical pictures consist of a large number of items, each of which may be described in detail and utilised for a variety of purposes. The clinically relevant item in 2D medical pictures is referred to as the Region of Interest (ROI), whereas in 3D images, it is referred to as the Volume of Interest (VOI). Saving energy is crucial since it is one of the most limited resources in these networks. However, DTN has an additional difficulty since communication between nodes is maintained so long as they are in physical proximity to one another. However, because to the nodes' mobility, this may not be long enough to provide the necessary multimedia data transmission. Wireless networks are susceptible to security assaults, and traditional security solutions are computationally demanding, making them unsuitable for networks that constantly need to recharge their batteries. All of these are reasons for tackling the problems of multimedia data processing and transmission via wireless networks in this dissertation. With this in mind, it has been attempted to investigate low-overhead and safe multimedia data compression as a solution to the issue that energy-constrained nodes in these networks limit complex multimedia processing while keeping at least basic security features. LZW-OMCA compression using the Octagonal Multimedia Compression Algorithm is part of the suggested method. The purpose of this is to improve the compression ratio. The proposed approach uses a little bit of crypt to compress data, which makes the data unreadable to anybody except the intended receiver, hence providing network security. The previous proposed works analysed the performance of several compression algorithms applied to multimedia material. Performance assessment utilising MSE, SSIM, and other metrics are used to show the pros and cons of each segment.

Keywords: Octagonal multimedia compression algorithm, data compression, LZW-OMCA compression, MSE, SSIM

[*]Corresponding author. Tammineni Sreelatha, Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur District, Andhra Pradesh, 522502, India. E-mail: sreelatha457@gmail.com. Orchid: 0000-0002-0951-2796.

ISSN 1064-1246/$35.00 © 2023 – IOS Press. All rights reserved.

1. Introduction

The practise of concealing information is not academically distinct. It has such deep ties to several other disciplines. Steganography is a subfield that studies this phenomenon. It explains the idea behind using a seemingly innocuous medium to transmit a secret multimedia message (such as text, audio, video, etc.) [1]. It is widely used as a multimedia security information-hiding method. Equally feasible is the transmission of pictures using this medium. Unfortunately, it is still feasible for hackers to decipher the encrypted information. The ways in which we pass the time, interact with one another, and absorb and share information are all being altered by the rapid development of networks, computers, digital storage, and wireless technology. The explosion in interest in and use of digital multimedia has been a major factor in these shifts. However, this has prompted legitimate worries about the safety, validity, and ownership of multimedia information. Some of the academics have devised security algorithms with the goal of making steganography more secure and hence less vulnerable to such attacks [2].

From Fig. 1, The practise of multimedia data-hiding—the covert incorporation of data into a multimedia host—offers promising answers to several issues, although in the face of unknown obstacles. The research community continues to devote a lot of time and energy to data hiding because of its potential uses in the protection of multimedia material. In order to overcome the difficulties presented by multimedia data concealment, a number of different fields, including image processing, computer vision, information theory, signal compression, error correction coding, and communication theory, must be brought together [3]. The researcher in this dissertation tackles a number of important problems in the area, laying the groundwork for the development of deployable, real-world methods. The researcher offers answers to various challenges of significance to the research community by combining experimental and analytical methods.

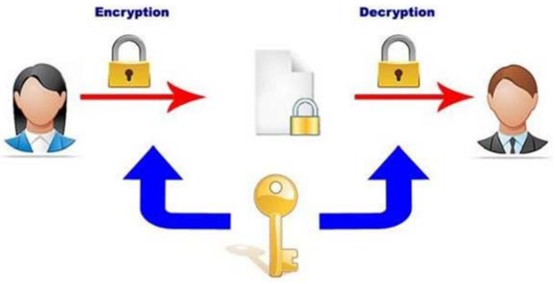


Fig. 1. Encryption and Decryption.

The primary goal of this study is to propose a novel technique called OMCA (Octagonal Multimedia Compression Algorithm) for safely compressing various types of data content. Using a system of information concealment, we can send multi-media files over the Internet while keeping them safe from prying eyes. To provide attention to multimedia data compression in data mining and the means of transmitting data safely. As a means of decreasing file sizes, processing times, storage needs, etc., OMCA is a novel hybrid data compression technique. Steganography is used by the majority of multimedia mining techniques to compress images and hide information about their categories. method of encoding multimedia data that minimises the possibility of data corruption during transmission and storage [4].

By using a data compression approach to lessen the amount of data that must be sent, we may save time and space on the hard drive. (Both “lossless” and “lossy” compression [5] methods exist for reducing the size of digital files. When it is necessary to restore the original form of the data after compression, lossless compression is used. (In contrast, the premise of lossy compression is that imperfect data restoration is acceptable.) Both types of compression are included into the suggested strategy. To get a greater compression ratio, this is done. The proposed approach uses a little bit of crypt to compress data, which makes the data unreadable to anybody except the intended receiver, hence providing network security. The goal of multimedia data compression is to reduce storage requirements or increase data density while maintaining original file sizes. Character and byte data files, for example, may be compressed using various compression algorithms. Like Shannon’s examination of the basic limitations of communication systems [6], one may wish to assume an intelligent opponent, as is done in certain game-theoretic assessments of jamming systems, in order to deduce the fundamental limits of watermarking and data concealing systems. The next proposed work will be a thorough analysis of the various literature surveys.

2. Literature survey

In order to secure images, [7] recommended using a 2D-Sine Logistic Modulation Map. The sine and logistic maps were used to create a 2D chaotic map. Using a chaotic and permutation-substitution network, [8] developed a cryptosystem. The straightforward picture was muddled and diffused with the use of a reworked logistic map. The updated logistic map was used to build s-boxes for the substitution process. In 2016, Wu et al. suggested an encryption approach based on a 6D-hyperchotic system, with encryption taking place in both the pixel and wavelet domains.

An encryption technique based on a linked chaotic system was presented by [9]. A new and enhanced chaotic map was created by coupling the Chen chaotic system with a 3D chaotic map. The picture was encrypted by using the coupled chaotic system to permute the 3D bit matrix of the image. A technique for encrypting images at the bit level, based on piecewise linear chaotic maps, was proposed by [10]. A grayscale picture was split into two binary sequences of equal length. Chaos-based bit-level permutation of the binary sequences was used to decode the picture.

An image encryption system based on hyperchaos and non-uniform cellular automata has been developed by [11]. Experimental and theoretical findings demonstrated the suggested method's robustness against noise assaults and huge key space. New chaotic map based on Beta function was suggested by [11] Pseudorandom sequences were created with the help of the new map. Permutations and substitutions were performed on the pseudo-random sequence to produce the cypher. The double image encryption system presented by [12] makes use of a modified logistic map and cellular automata. The encryption keys for a picture are generated by convolution of the logistic maps. Using cellular automata and a combination of the least significant bits of two 7-bit pictures and diffusion, it is possible to encrypt the images twice. As a result, chaos is crucial to the security of symmetric key encryption, which relies on a very secret key. Chaotic encryption's benefits include extensive design freedom, a large pool of available chaotic systems, and a vast number of complicated and varied key options. Table 1 shows the existing methodology comparison.

Data compression and encryption both aid in lowering the time and space required for data transmission and storage. In the past, data was compressed after being sampled, with just the minimum amount of information being kept for later use. It is helpful, nevertheless, to take just the most crucial measurements immediately. Compressive sensing is a method that may be used to accomplish this. When a signal is compressible or sparse, compressive sensing aids in its flawless reconstruction using fewer samples. By using a random sensing matrix, Compressive Sensing helps to both compress and encrypt a sparse signal. While the conventional approach of sampling followed by compression and encryption has its uses, the coupled compression-encryption that may be done while sampling gives significant benefit.

However, CS-based encryption becomes vulnerable with time. If many signals are encrypted with the same secret key, the secret sensing matrix may be deduced using cryptanalysis. The design of the random sensing matrix is crucial for a practical CS-based data encryption method. The design should simplify computing while simultaneously enhancing communication security and throughput. A sensing matrix based on chaos theory may help with this. The field of cryptography [17] has made extensive use of chaos. These chaotic systems are very sensitive to their beginning circumstances, and their seemingly random states of disorder and irregularity are governed by underlying patterns and deterministic principles. Therefore, this paper suggests frameworks based on Chaotic Compressive Sensing, which allows for both the compression and encryption of visual data at the same time.

3. Architecture

Terry Welch's LZW-OMCA algorithm, a variant on the original LZ78 technique, was widely used. The compression method known as Lempel-Ziv-Welch (LZW-OMCA) converts sequences of characters into individual codes. It does not process the incoming text in any way. Instead, it simply appends any new character strings it encounters to a master list. The output is compressed if it is a single code rather than a string of characters [18]. Code generated by the LZW-OMCA method may be of any length, but it must include more than just a single character's worth of bits. When employing eight-bit characters, the first 256 codes are designated as the default character set. As the process continues, it assigns the remaining codes to strings.

Figure 2 shows the architecture of proposed work. The wavelet transform is a mathematical technique for performing hierarchical decompositions of functions, which may be further characterised by their rough overall shape and finer details [19]. Using the wavelet's time-frequency localization characteristic, crucial details may be gleaned from both the temporal and spectral dimensions. Wavelet's multiresolution

Table 1
Existing Methodology Comparison

Reference	Methodology	Disadvantage
[13]	Tensor based compression	To achieve a low-overhead bit budget, the factor matrices were compressed. This technique fared better than both tensor and wavelet-based approaches previously used.
[14]	Patch-Based Low-Rank Tensor Decomposition	It achieves remarkable improvements over standard approaches, but suffers from the lack of automatic patch-size selection.
[15]	CANDECOMP/PARAFAC Tensor-Based Compression (CPTBC)	The condensed outcomes were achieved through engineering sparsity and a uniform distribution of coefficients. PSNR measurements showed that the visual quality of this approach was quite similar to that of six conventional compression methods, including MPEG4, band-wise JPEG2000, TD, 3D-SPECK, 3D-TCE, and 3DTARP, for the same compression performance.
[15]	Tucker Decomposition and thresholding	While core tensor's compaction accuracy was commendable and its results were superior to those of the generally used TD 32 techniques in terms of compression performance, its reconstruction quality was marginally hampered by the coefficient thresholding.
[16]	Multilinear Singular Value Decomposition (MLSVD)	When compared to more conventional compression techniques like symmetric 3D-SPIHT and asymmetric 3D-SPIHT schemes, it resulted in higher quality reconstruction.

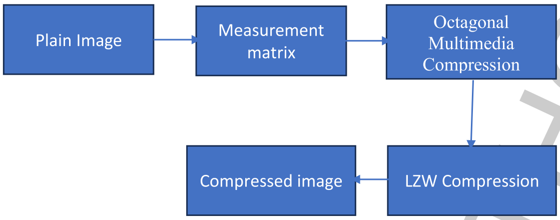


Fig. 2. Architecture of LZW-OMCA Compression with Octagonal Multimedia Compression Algorithm.

analysis is a key characteristic that mimics the human visual system's (HVS) processing of pictures. High-pass kernels, or wavelets, are basis functions that have a variety of special characteristics [20]. When seen from a purely theoretical perspective, wavelets are just the functions that may be created from a single function via dilations and translations.

$$\Psi_{a,b}(t) = |a|^{-1/2}\Psi\left(\frac{t-b}{a}\right) \qquad (1)$$

From Equation 1, where $\psi(x)$ A wavelet with parameters a and b denoting a dilation (scale) and a translation, respectively, is called a mother wavelet or an analysis wavelet. Either a compactly-supported or a biorthogonal basis function may be used. The wavelet transform may be used to express any function as the superposition of wavelet basis functions.

To get coefficients for every time step in a Continuous Wavelet Transform (CWT), the mother wavelet is continuously rotated along the time scale. The CWT is defined for the x(t) signal as:

$$CWT_{(a,b)} = \int_{-\infty}^{\infty} x(t)\cdot\Psi_{a,b}(t)\,dt \qquad (2)$$

In Equation 2, Since the OMCA's discretization lowers computing costs by discretizing the scaling and shifting factor as a = ,2-j. and b = ,2-j.k, it has largely replaced the CWT in popular usage. Because of this, we may define OMCA as

$$\mathrm{OMCA}(t) = \left|2^j\right|^{-1/2}\int_{-\infty}^{\infty} x(t)\cdot\psi\left(\frac{t-2^j*k}{2^j}\right)dt \qquad (3)$$

where j and $k \in Z$.

By using High-pass and Low-pass filters on the OMCA output, we are able to create a set of coefficients that is generalizable to high dimensional data.

Algorithm

Algorithm: This is a simplified version of the LZW-OMCA compression method. A superficial analysis of the algorithm reveals that LZW-OMCA prioritises the production of codes for known strings. In addition, the string table is updated whenever a new output code is generated. The following is an example string that may be used to illustrate the technique. A small set of English words, separated by a slash character (/), serves as the input string. A first run through the loop checks to verify whether the string "/W" is in the table, as can be shown by stepping through the beginning of the algorithm for this string. Since it isn't, the '/' code is printed and the text "/W" is appended to the list. The initial string description may be assigned to code 256 since we already have 256 characters declared for codes 0–255. The second-string code, "WE," is appended to the table once the third letter, "E," has been read in, and the code for the letter "W" is shown. It keeps on until the string number 256 is found in the second word, which consists of the letters '/' and 'W.

Algorithm* 1 *Compression using LZW − OMCA

```
Routine LZW-OMCA COMPRESS
STRING = get input character
WHILE there are still input characters DO
CHARACTER = get input character
IF STRING+CHARACTER is in the string table then STRING
= STRING + character
ELSE
output the code for STRING
add STRING+CHARACTER to the string table
STRING = CHARACTER
END of IF
END of WHILE
output the code for STRING
```

The value 256 is printed and a new string of three characters is added to the existing string table. This operation will keep going until no more codes can be extracted from the string. When using 12-bit code, the demo application functions properly. Therefore, bytes are represented by the 0–255 range of codes, and substrings by the 256–4095 range.

The table of strings that resulted from the example output is seen above. Because a new string is added to the database every time a code is produced, it becomes full very quickly. Five code replacements and seven characters were generated from this very repetitive input. The 19-character input string would be compressed to 13.5 bytes if we were utilising 9-bit codes for output. This is a contrived example meant to illustrate code replacement, of course. Compression often doesn't kick in until a big table has been formed, after at least a hundred bytes have been read in.

Codes generated by the compression method must be used by the decompression algorithm to faithfully reproduce the input stream. The LZW-OMCA approach is fast in part because it does not transmit the string table to the decompression programme. The input stream may be used to reconstruct the 50 table in the exact same way it existed before compression. The compression method always sends out the STRING and CHARACTER parts of the code before using it in the output stream, therefore this is doable.

Algorithm* 2 *Decompress using LZW − OCMA

```
Routine LZW-OMCA DECOMPRESS
Read OLD_CODE
output OLD_CODE
CHARACTER = OLD_CODE
WHILE there are still input characters DO
Read NEW_CODE
IF NEW_CODE is not in the translation table THEN
STRING = get translation of OLD_CODE
STRING = STRING + CHARACTER
ELSE
STRING = get translation of NEW_CODE
END of IF
output STRING
CHARACTER = first character in STRING
add OLD_CODE + CHARACTER to the translation table
OLD_CODE = NEW_CODE
END of WHILE
```

***Explanation* :** What's crucial to remember is that after compression, the string table looks precisely the same as the table that was generated. When using a compression technique, the output string will be the same as the input string. Like the compression code, the first 256 codes have been specified to map directly to strings of a single character [21].

If your data stream contains any strings that are repeated, LZW-OMCA compression will shine. This means it is very effective in reducing the size of English text. We anticipate compression ratios of at least 50%. Similarly, when compressing previously recorded screens and displays, the results are often flawless. Compressing binary data files involves a higher degree of danger. The effectiveness of data compression varies widely. Data files may sometimes be compressed even more than text files. You can typically get a sense of whether or not your data will compress effectively with some trial and error. Files with a lot of duplicate information benefit most from LZW-OMCA compression. Especially with text

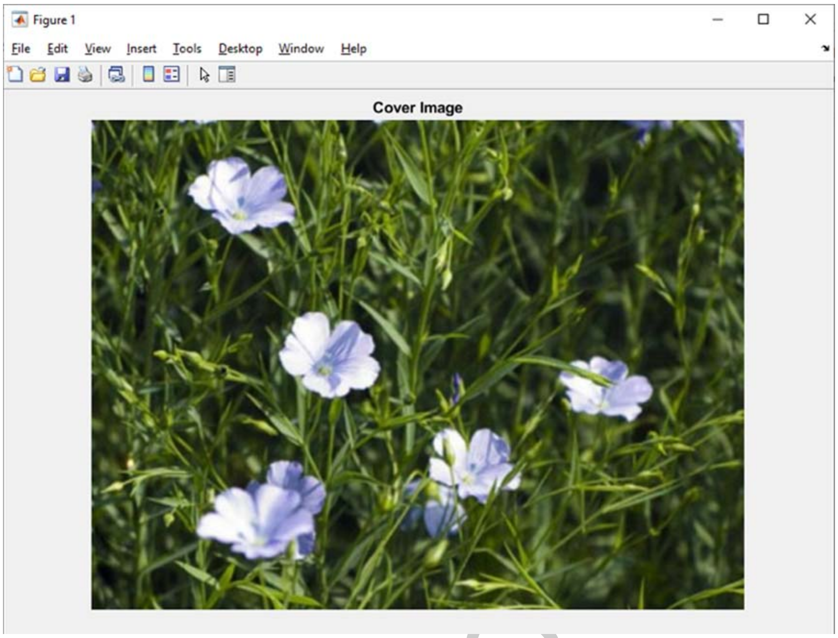


Fig. 3. Cover Image.

and black-and-white graphics. Compressed files that don't include any repetition may actually expand in size. The speed of LZW-OMCA compression.

4. Experimental outcomes

The C programming language is used to implement the Multimedia Data Compression Techniques incorporate in Information concealing component. Data compression methods have been devised to solve this problem, and their computational efficiency has been tested (compression ratio, compression time). In common parlance, "coding" refers to the process of compressing data; nevertheless, "coding" is a fairly broad phrase that may apply to any kind of unique representation of data that serves a specific purpose. By definition, information theory is the study of efficient coding and its effects on transmission speed and error probability. One aspect of information theory, data compression seeks to reduce the volume of data that must be transferred. This paper's goal is to introduce and evaluate many different data compression techniques. Data compression may be defined as the process of reducing the size of a string of characters by converting it to another string (of bits, for example) that includes the same information but has the shortest feasible length. Compressing data is useful for both sending and storing digital information. As the use of computers spreads to new fields, there is a growing need to store massive amounts of data for a variety of data processing applications. Meanwhile, an abundance of digital communication networks causes a flood of information to flow across wires. Compressing data before storage or transmission helps cut down on expenses shown in Fig. 3.

The capacity of the communication channel is effectively increased when the quantity of data to be transferred is decreased. As an example, halving a file's size via compression effectively doubles its storage capacity. This might relieve pressure on the computer's I/O ports, making it possible to move the data to a more efficient storage tier. Below, the structural similarity index and PSNR, which indicate the most relevant sensitivity outcomes after making minute tweaks at several critical levels, are shown (SSIM)

The following metrics are used to evaluate the effectiveness of the suggested compression algorithms:

4.1. Compression ratio (CR)

The compression ratio measures the decrease in file size relative to the original. To put it another way, the compression ratio is defined as the ratio of the number of bits required to represent the original picture to the number of bits required to represent the compressed image. Here is the formula used to determine the CR:

$$\mathrm{CR} = \frac{n_o}{n_c} \tag{4}$$

n_o – number of bits to represent the original image
n_c- number of bits to represent the compressed image

4.2. Bit rate (BR)

Bit Rate refers to the average number of bits used for each picture sample (BR). Number of bits required to represent compressed picture divided by total number of image unit samples.

were,

$$\mathrm{BR} = \frac{n_c}{n_s} \tag{5}$$

n_c – number of bits to represent the compressed image
n_s – Total number of unit samples

Different picture types may need a different sample size. Bits Per Pixel refers to the number of bits used to store a two-dimensional picture (BPP). Where ns is the total number of pixels in the picture, BPP is the number of bits needed for each individual pixel. Bits Per Voxel refers to the standard measurement of data storage for three-dimensional pictures, where each voxel serves as the fundamental unit (BPV). In the context of 3D images, ns represents the total number of voxels and BPV represents the number of bits needed per voxel. Bit Rate may refer to either Bit Per Second or Bit Per Vector (BR).

4.3. Peak signal to noise ratio (PSNR)

The Peak Signal-to-Noise Ratio (PSNR) is one extensively used measure for evaluating the quality of compressed images. It's a reference-based assessment measure that looks at the differences in the two pictures' intensities. Simply said, PSNR is the ratio of the greatest intensity value (peak) in a picture to the standard deviation of the intensity values within the image. The PSNR values are expressed on a logarithmic decibel (dB) scale. Here is the formula for calculating the PSNR:

$$PSNR = 10 \log_{10} \left(max \frac{i}{MSE} \right) \tag{6}$$

where,
maxi– maximum intensity / high peak
MSE– Mean Square Error

The two photos being compared must also be of same size. That is to say, each image should have the equal number of rows and columns. Calculating the PSNR requires determining the MSE. Mean Square Error (MSE), a distance metric based on Euclidean geometry, is used to calculate the intensity difference between the two pictures. A high PSNR value in decimal form suggests a greater degree of similarity between pictures, and this range covers the values from 0 to 1. In the absence of picture deterioration, with an MSE of 0, the PSNR value approaches to infinity.

4.4. Structural similarity index (SSIM)

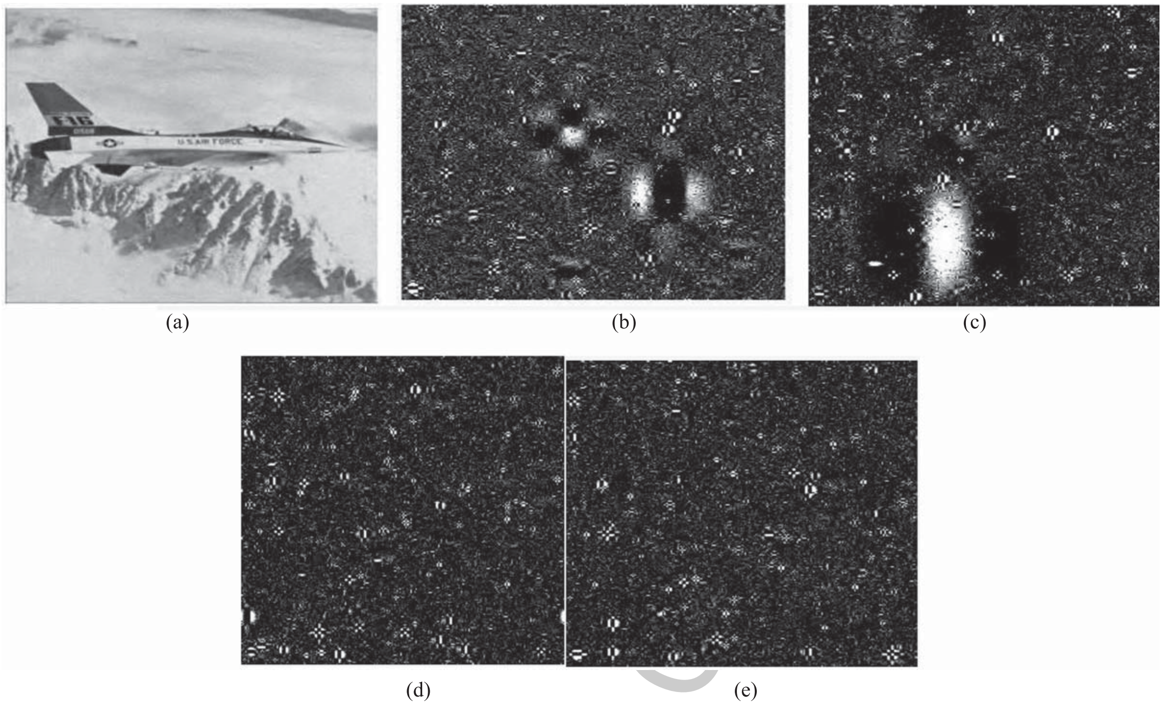
SSIM is a quality metric that is based on the HVS and is used to quantify the visual difference between two otherwise identical pictures. Multiplying an image's brightness, contrast, and structural term yields the SSIM, therefore its calculation is straightforward. The reference and processed pictures must have the same dimensions, since this measure relies only on their similarity. SSIM compares the original and modified photos and can also be used for 3D volumes. Three terms, including brightness, contrast, and structural term, are used in the SSIM calculation, as shown below.

$$\mathrm{SSIM}\,(\mathrm{x, y}) = \left[l\,(x, y)\right]^{\alpha} \cdot \left[\,c\,(x, y)\right]^{\beta} \cdot \left[s\,(x, y)\right]^{\gamma} \tag{7}$$

where, μx and μy are the local means, σx and σy are the standard deviations and σxy is cross covariance for images x and y. l, c and s are luminance, contrast and structural term respectively.

The security key required for an efficient encryption scheme should be highly sensitive. A small change in the security key should result in a totally different decrypted image. The cipher is decrypted using the same key as the one used in encryption and the reconstructed image is shown in Fig. 4(a). The encrypted picture may be retrieved by using the right keys and the tiny value = 1015. Figure 4 displays the erroneously rebuilt pictures with different keys, demonstrating the proposed scheme's extreme sensitivity to even little variations in the key. Table 2 displays the PSNR of a deciphered picture with a little off key.

Decrypted image PSNR is particularly sensitive to changes in the key parameters, as seen in Table 2. The attacker cannot even decipher a simple picture with a slight change on the order of 1015.



(a) (b) (c)

(d) (e)

Fig. 4. Reconstructed image with (a) Correct Keys ($X0$, $\mu 1$, $n0$, d) (b) Incorrect Key ($X0 + 10^{-15}$) (c) Incorrect Key ($\mu 1 + 10^{-15}$) (d) Incorrect Key ($n0 + 2$) (e) Incorrect Key ($d + 2$).

Table 2
Key sensitivity analysis of CCS and Masking scheme

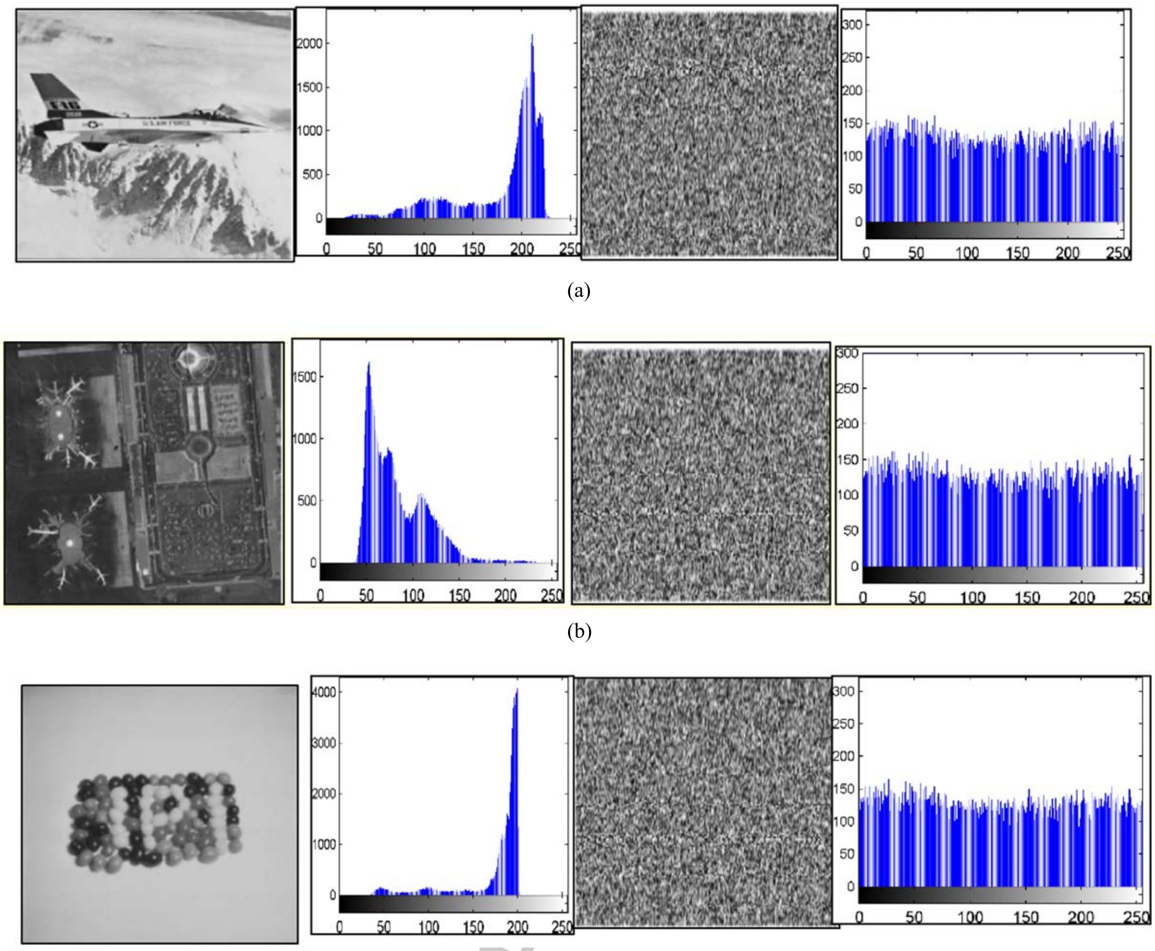
Parameter	PSNR (dB)	SSIM
($x0$, $\mu 1$, $n0$, d)	32.8271	0.8767
($x0$+ 10^{-15}, $\mu 1$, $n0$, d)	4.4524	0.0124
($x0$, $\mu 1 + 10^{-15}$, $n0$, d)	3.7489	0.0111
($x0$, $\mu 1$, $n0 + 2$, d)	3.7023	0.0104
($x0$, $\mu 1$, $n0$, $d + 2$)	3.7335	0.0127

4.5. Statistical attack analysis

The histogram displays the distribution of pixel intensities of a digital picture in a visual format. The histogram of the unprocessed picture is also unique for each digital image since the pixel intensities are different. Because of its one-of-a-kind nature, it is susceptible to statistical assaults. Therefore, encrypted photos should have a distinct histogram from the original, plain image, and cypher images should have a comparable histogram. The test pictures and their matching encrypted versions are shown in a histogram in Fig. 5. The cyphers' histogram (shown in Fig. 5) is distinct from that of the plain picture (shown in Fig. 5) but otherwise displays a similar distribution. As a result, the suggested approach is secure against statistical assaults. No statistical information about the original picture can be gleaned from the cypher image's histogram.

The Chi-square test provides a mathematical justification for the histogram's consistency. Table 3 displays the Chi-square test results for the encrypted pictures. The greater the homogeneity of the histogram, the smaller the results of the Chi-square test. The table shows that the chi-square value for each of the test photos is lower than 293.2478. This means that an attacker cannot use the histogram of the encrypted photos to reveal any statistical information.

The ability to decorrelate neighboring pixels is crucial for a secure encryption scheme that can withstand statistical attacks. Table 4 displays the calculated horizontal, vertical, and diagonal correlation coefficients between contiguous pixels in the test picture Aircraft and its cypher. In the unprocessed picture, the correlation between neighboring pixels is quite strong, coming close to 1. Comparing the encrypted picture to the plain image, the table reveals very little association between nearby pixels. The suggested method decorates the pixel in the x, y, and z axes. Test image correlation coefficients for several horizontal and diagonal methods are broken down in Table 5. Table 5 suggests that, in comparison to other schemes,



(a)

(b)

(c)

Fig. 5. Histogram Analysis: (a) Plain Image (b) Histogram of Plain Image (c) Ciphered Image (d) Histogram of Ciphered Image.

Table 3
Chi-square value of ciphered images

Test Images	Chi-square value
Airplane	297.4531
Barbara	279.2344
Bee	287.2188
Bobcat	296.5000
Butterfly	284.5625
Fishing Boat	286.7500
Gold hill	288.0781
Jelly Beans	280.4219
Peppers	264.4688
Splash	292.9531

the horizontal correlation between neighboring pixels in the cypher picture is lower.

Figure 6 shows the pictorial representation of the distribution of the pixels in the plain and ciphered images in the horizontal, vertical and diagonal direction for the test image Aircraft.

The entropy of the encrypted picture may be calculated to reveal how random it is. Calculating the global entropy. With 256 levels of grey, an image's global entropy ceiling is set at 8 bits per pixel. For this reason, an ideal entropy for a cypher picture is 8. Table 6 displays the results of the calculation of the global entropy for each of the test pictures. The entropy of the cypher is quite near to the theoretical value of 8 in the table, indicating that the encrypted picture has a high degree of unpredictability. When compared to previous schemes, the suggested system has a higher global entropy.

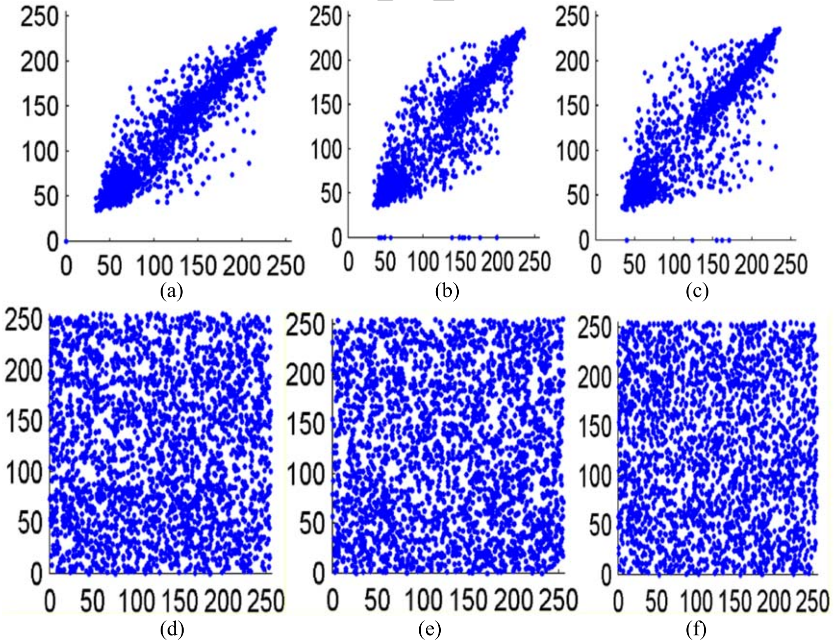
In Table 6, we see the average local entropy for 15 randomly chosen blocks from the encrypted picture. The local Shannon entropy obtained using the

Table 4
Correlation Coefficient between Adjacent Pixels for test image Aircraft

Test Images	Plain Image Correlation			Ciphered Image Correlation		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Airplane	0.9438	0.9319	0.8824	0.0318	0.0223	–0.0384
Barbara	0.9458	0.9666	0.9170	–0.0112	0.0052	–0.0187
Bee	0.9791	0.9807	0.9653	–0.0251	0.0063	–0.0103
Bobcat	0.9731	0.9741	0.9608	0.0242	0.0484	–0.0106
Butterfly	0.9404	0.9338	0.8846	–0.0024	–0.0294	–0.0087
Fishing Boat	0.9304	0.9431	0.8707	0.0177	0.0382	–0.0218
Goldhill	0.9644	0.9672	0.9399	–0.0182	0.0434	0.0032
Jelly Beans	0.9754	0.9847	0.9599	–0.0252	0.0122	–0.0188
Peppers	0.9636	0.9697	0.9442	0.0015	0.0001	–0.0371
Splash	0.9777	0.9827	0.9529	0.0116	0.0303	–0.0060

Table 5
Comparison of Correlation Coefficient for different algorithms

Test Images	Horizontal Correlation			Diagonal Correlation		
	[17]	[18]	Proposed	[17]	[18]	Proposed
Airplane	0.0120	–0.0015	0.0318	–0.0128	–0.0094	–0.0384
Barbara	0.0231	0.0157	–0.0112	0.0075	–0.0113	–0.0187
Bee	0.0135	0.0093	–0.0251	0.0233	–0.0363	–0.0103
Bobcat	–0.0451	0.0188	0.0242	0.0013	0.0264	–0.0106
Butterfly	–0.0020	0.0381	–0.0024	–0.0174	–0.0133	–0.0087
Fishing Boat	0.0099	0.0200	0.0177	0.0270	0.0102	–0.0218
Goldhill	–0.0054	0.0050	–0.0182	0.0388	–0.0132	0.0032
Jelly Beans	–0.0368	0.0468	–0.0252	0.0442	–0.0086	–0.0188
Peppers	0.0150	0.0115	0.0015	–0.0004	–0.0072	–0.0371
Splash	0.0106	0.0286	0.0116	0.0173	0.0198	–0.0060



(a) (b) (c)

(d) (e) (f)

Fig. 6. Correlation Analysis of test image aircraft: (a) Horizontal correlation of Plain Image (b) Vertical correlation of Plain Image (c) Diagonal correlation of Plain Image. (d) Horizontal correlation of Ciphered Image (e) Vertical correlation of Ciphered Image (f) Diagonal correlation of Ciphered Image.

Table 6
Comparison of Global Entropy for different algorithms

Test Images	Global Entropy			
	Plain Image	[17]	[18]	Proposed
Airplane	6.7294	7.9921	3.3311	7.9940
Barbara	7.5838	7.9915	4.1461	7.9945
Bee	6.6953	7.9931	4.3726	7.9943
Bobcat	5.8494	7.9941	4.3263	7.9942
Butterfly	7.3075	7.9934	4.2088	7.9943
Fishing Boat	7.1583	7.9919	4.0745	7.9943
Goldhill	7.4450	7.9924	4.3420	7.9942
Jelly Beans	5.7286	7.9948	4.2615	7.9944
Peppers	7.5770	7.9936	3.4408	7.9948
Splash	7.2372	7.9933	3.1728	7.9941

Table 7
Comparison of Local Shannon Entropy for different algorithms

Test Images	Local Shannon Entropy		
	[17]	[18]	Proposed
Airplane	6.9971	2.7740	7.0597
Barbara	7.0079	3.4445	7.0962
Bee	7.0933	3.8062	7.1539
Bobcat	7.0526	3.7310	7.0963
Butterfly	7.1329	3.5427	7.1466
Fishing Boat	7.0538	3.4076	7.8907
Goldhill	7.0578	3.6561	7.0647
Jelly Beans	7.0250	3.7172	7.1278
Peppers	6.9629	2.9210	7.1366
Splash	7.1247	2.7164	7.1149

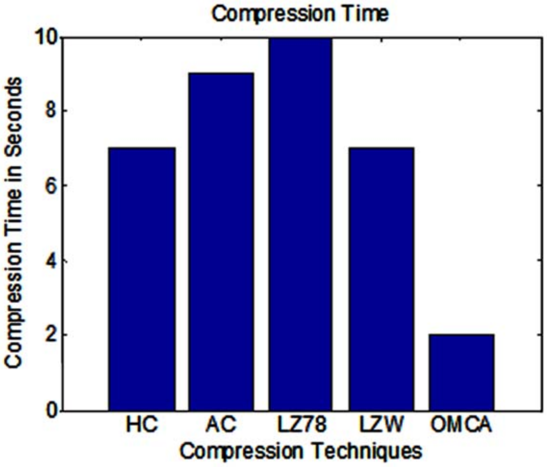


Fig. 7. Compression techniques.

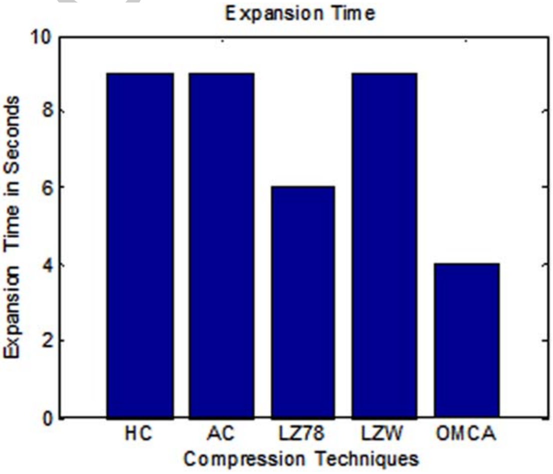


Fig. 8. Decompression techniques.

suggested technique is quite near to the theoretical value, as shown in the table. For a completely random grayscale picture with a block size of 16 by 16, the theoretical mean of the local Shannon entropy is 7.174966353. Unlike global entropy, which suffers from inaccuracy, inconsistency, and inefficiency, local Shannon entropy is more precise, consistent, and efficient.

Table 8 compares the LZW-OMCA-coded contents of 10 distinct sets of text data. The input and output file sizes, compression ratios, and compression times are all variable for the provided files. The total compression/decompression time is 0.14 seconds, and the average savings ratio is 52.94 percent. Table 9 shows a comparison of ten distinct picture files compressed using LZW-OMCA. The input and output file sizes, compression ratios, and compression times are all variable for the provided files. The total compression and decompression time is 0.14 seconds, and the average savings ratio is 52.53 percent. Using MATLAB, we evaluated the provided data. The compression ratios of many common compression methods are shown in Fig. 7.

The given data analysed by using MATLAB. In Fig. 8 – mentioned the Compression time of different compression techniques. The suggested LZW-OMCA-OMCA-encryption techniques are shown to be very secure and to drastically cut down on both storage and transmission bandwidth via a study of performance measures.

5. Conclusion

This study developed an image-encrypting cryptosystem using chaotic compressive sensing-based LZW-OMCA-OMCA-. For the sake of data safety, a framework was presented that combines CS with chaos. The CS-based encryption techniques included chaotic aspects such sensitivity to beginning val-

Table 8
Text File compression - LZW-OMCA Coding

Name of the File	Size of Input file (KB)	Size of Compressed file (KB)	Percentage Saving Ratio	Compression time + Decompression time
20ng-test-all-terms.txt	10556	5027	52.38 %	0.06 + 0.08 sec.
20ng-test-no-short.txt	9336	4363	53.27 %	0.07 + 0.08 sec.
20ng-test-no-stop.txt	7028	3331	52.61 %	0.05 + 0.07 sec.
20ng-test-stemmed.txt	6276	2989	52.38 %	0.07 + 0.08 sec.
20ng-train-all-terms.txt	16292	7613	53.27 %	0.07 + 0.08 sec.
20ng-train-no-short.txt	14396	6711	53.38 %	0.06 + 0.08 sec.
20ng-train-no-stop.txt	10847	5080	53.17 %	0.06 + 0.08 sec.
20ng-train-stemmed.txt	9684	4570	52.81 %	0.06 + 0.07 sec.
cade-test-stemmed	11933	5635	52.78 %	0.07 + 0.08 sec.
cade-train-stemmed	25086	11708	53.33 %	0.07 + 0.08 sec.
Average			**52.94 %**	**0.06 + 0.08 sec.**

Table 9
Image File compression - LZW-OMCA Coding

Name of the File	Size of Input file (KB)	Size of Compressed file (KB)	Percentage Saving Ratio	Compression time + Decompression time
000001.jpg	14	7	52.45 %	0.08 + 0.09 sec.
000002.jpg	16	8	52.61 %	0.06 + 0.08 sec.
000003.jpg	17	8	53.05 %	0.05 + 0.07 sec.
000004.jpg	23	11	52.38 %	0.08 + 0.09 sec.
000005.jpg	11	5	52.15 %	0.06 + 0.09 sec.
000006.jpg	14	7	53.01 %	0.07 + 0.09 sec.
000007.jpg	20	10	52.22 %	0.07 + 0.09 sec.
000008.jpg	16	8	52.66 %	0.05 + 0.07 sec.
000009.jpg	19	9	52.48 %	0.06 + 0.08 sec.
000010.jpg	20	10	52.25 %	0.06 + 0.08 sec.
Average			**52.53 %**	**0.06 + 0.08 sec.**

ues, pseudo-randomness, and ergodicity as a result of the intimate relationship between these cryptographic elements. In order to implement chaotic compressive sensing, new one-dimensional chaotic maps were created, analysed quantitatively, and put into practise. By using a confusion-diffusion architecture and a measurement matrix structure that is reliant on the plaintext, traditional assaults like chosen-plaintext/known plain-text attacks may be prevented. In addition, we developed and analysed a visually safe picture encryption system based on chaotic compressive sensing. The suggested cryptosystem has been shown to be secure after undergoing extensive security examination. Metrics like as PSNR, SSIM, Entropy, Correlation coefficient, NPCR, and UACI are used to assess the effectiveness of the suggested systems. In this proposed work, simulations were used to produce the experimental findings. Hardware implementation for real-time applications will be possible in the future. For instance, LZW-OMCA-OMCA-encryption may be used for the safe storing and transmission of electronic pictures and clinically-relevant data in picture archiving and communication systems (PACS), which are largely utilised by healthcare institutions.

References

[1] S. Zheng, D. Li, D. Hu, D. Ye, L. Wang and J. Wang, Lossless data hiding algorithm for encrypted images with high capacity, *Multimedia Tools and Applications* **75**(21) (2016), 13765–13778.

[2] S. Singh and R. Devgon, Analysis of encryption and lossless compression techniques for secure data transmission. In *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, (2019), pp. 1–5. IEEE.

[3] K.N. Singh and A. Kumar Singh, Towards Integrating Image Encryption with Compression: A Survey, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* **18**(3) (2022), 1–21.

[4] J. Samuel Manoharan and A. Sathesh, Super-resolution reconstruction model using Compressive Sensing and Deep Learning, *International Journal for research and development in Technology* **7**(4) (2017), 884–889.

[5] C. Priya, C. Ramya, R.V. Agashthiya, R. Hema, G. Mythily and V.P. Preethi, An Efficient Method for Secure Image Compression, *Int. J. Innov. Technol. Explor. Eng* **8**(6) (2019), 266–270.

[6] Y. Pourasad and F. Cavallaro, A novel image processing approach to enhancement and compression of X-ray images, *International Journal of Environmental Research and Public Health* **18**(13) (2021), 6724.

[7] P. Mathur, A. Yadav, V. Kumar Verma and R. Purohit, Paradigms of image compression and encryption: A review. In *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, (2019), pp. 313–317. IEEE.

[8] S. Maqbool, N. Ahmad, A. Muhammad and A.M. Martinez Enriquez, Simultaneous Encryption and Compression of Digital Images Based on Secure-JPEG Encoding. In *Mexican Conference on Pattern Recognition*, (2016), pp. 145–154. Springer, Cham.

[9] X.-Y. Li, X.-B. Zhou, Q.-L. Zhou, S.-J. Han and Z. Liu, High-capacity reversible data hiding in encrypted images by information preprocessing, *Complexity* **2020** (2020).

[10] Q. Li, Y. Fu, Z. Zhang, A. Joseph Fofanah and T. Gao, Medical images lossless recovery based on POB number system and image compression, *Multimedia Tools and Applications* **81**(8) (2022), 11415–11440.

[11] A. Anil Kumar and A. Makur, Distributed source coding-based encryption and lossless compression of gray scale and color images. In *2008 IEEE 10th Workshop on Multimedia Signal Processing*, (2008), pp. 760–764. IEEE.

[12] A. Kingston, S. Colosimo, P. Campisi and F. Autrusseau, Lossless image compression and selective encryption using a discrete radon transform. In *2007 IEEE International Conference on Image Processing* **4** (2007), pp. IV–465. IEEE.

[13] K.S. Kasmeera, Shine P. James and K. Sreekumar, Efficient compression of secured images using subservient data and Huffman coding, *Procedia Technology* **25** (2016), 60–67.

[14] X. Kang, A. Peng, X. Xu and X. Cao, Performing scalable lossy compression on pixel encrypted images, *EURASIP Journal on Image and Video Processing* **2013**(1) (2013), 1–6.

[15] U. Jayasankar, V. Thirumal and D. Ponnurangam, A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications, *Journal of King Saud University-Computer and Information Sciences* **33**(2) (2021), 119–140.

[16] A.Y. Horita, R. Bonna, Denis S. Loubach, I. Sander and I. Söderquist, Lempel-Ziv-Markov Chain Algorithm Modeling using Models of Computation and ForSyDe. In *FT2019. Proceedings of the 10th Aerospace Technology Congress, October 8-9, 2019, Stockholm, Sweden*, no. 162, pp. 152–155. Linköping University Electronic Press, 2019.

[17] Q. Gao, Secure Reversible Information Hiding in Image Based on Loseless Compression. In *Proceedings of the 2020 5th International Conference on Multimedia Systems and Signal Processing*, (2020), pp. 50–53.

[18] H. Gao and T. Gao, A secure lossless recovery for medical images based on image encoding and data self-embedding, *Cluster Computing* **25**(1) (2022), 707–725.

[19] P. Chaudhary, R. Gupta, A. Singh, P. Majumder and A. Pandey, LZW-OMCA-OMCA and encryption using a novel column-wise scanning and optimization algorithm, *Procedia Computer Science* **167** (2020), 244–253.

[20] R. Arnold and T. Bell, A corpus for the evaluation of lossless compression algorithms. In *Proceedings DCC'97. Data Compression Conference*, (1997), pp. 201–210. IEEE.

[21] O.F. Abdel Wahab, A.I. Hussein, Hesham F.A. Hamed, Hamdy M. Kelash and Ashraf A.M. Khalaf, Efficient Combination of RSA Cryptography, Lossy, and Lossless Compression Steganography Techniques to Hide Data, *Procedia Computer Science* **182** (2021), 5–12.