Journal of Robotics and Control (JRC)
Volume 5, Issue 5, 2024
ISSN: 2715-5072, DOI: 10.18196/jrc.v5i5.21834

LICA-CS: Efficient Lossless Image Compression Algorithm via Column Subtraction Model

Mahmoud Al Qerom [1*], Mohammad Otair [2], Farid Meziane [3], Sawsan AbdulRahman [4], Maen Alzubi [5]
[1] Department of Information Communication Technology, British University of Bahrain, Bahrain
[2] Department of Computer Science, Khawarizmi University Technical College, Jordan
[3] Data Science Research Centre, University of Derby, UK
[4] Department of Computer Studies, Arab Open University, Bahrain IT
[4] Cyber Security Systems and Applied AI Research Center, Department of CSM, Lebanese American University, Lebanon
[5] Department of Robotics and Artificial Intelligence, Jadara University, Jordan
Email: [1] m.a.qerom@bub.bh
*Corresponding Author

***Abstract*—Driven by the unprecedented amount of data generated in the last few decades, data storage and communication are becoming more challenging. Although many approaches in data compression have been developed to alleviate these challenges, more efforts are still needed, especially for lossless image compression, which is a promising technique when critical information loss is not allowed. In this paper, we propose a new algorithm called the Lossless Image Compression Algorithm using a Column Subtraction model (LICA-CS). LICA-CS is efficient, low in complexity, decreases the image bit-depth, and enhances state-of-the-art performance. LICA-CS first implements a color transformation method as a pre-processing phase, which strategically minimizes inter-channel correlations to optimize compression outcomes. After that, a novel subtraction method was developed to compress the image data column-wise. We tackle the similarity and proximity of pixel values within adjacent columns, which offers a distinct advantage in reducing image size observing a significant size reduction of 71%. This is achieved through the subtraction of neighboring columns. The conducted experiments on colored images show that LICA-CS outperforms existing algorithms in terms of compression rate. Moreover, our method exhibited remarkable enhancements in execution time, with compression and decompression processes averaging 1.93 seconds. LICA-CS advances the state-of-the-art in lossless image compression, promising enhanced efficiency and effectiveness in image compression technologies.**

Keywords—Lossless Compression; Reversible Color Transformation; Column Subtraction Compression; Data Compression; Color Transformation Method.

I. INTRODUCTION

Most of our everyday activities are now dominated by social media, e-commerce, and other digital resources, allowing Big Data to pave its way to further growth in the future. When it comes to images and videos, data gets more challenging to handle due to its large size and the number of pixels it contains, leading to exponential growth in the data generated. With the proliferation of smartphones equipped with high-resolution digital cameras, extra metadata such as location, date, and time has to be saved. These smartphones can capture motion images with a 16-bit depth and a size of 50.3 megabytes. It has been reported [1] that 2500 petabytes of data are generated daily. Moreover, with the advances in network technology, a large number of images are generated and shared over the network [2], [31]-[35]. However, this image explosion is exacerbating its storage and transmission due to the limited storage capacity and network bandwidth [3], [4], [36], [37]. Thus, it is crucial to have efficient image compression techniques.

The process of image compression involves reducing the size of the image data, while retaining the essential information and preserving its visual quality. Images are commonly encoded by lossless or lossy compression methods. Firstly, lossy methods eliminate redundant and irrelevant data by allowing it to be decrypted with a particular amount of degradation using different techniques (i.e., Vector Quantization, Block truncation coding, Fractal Coding, Sub band coding, and Transformation Coding). Vector Quantization is effective in reducing data redundancy, but it suffers from high computational complexity and memory requirements, making it impractical for real-time applications. Block truncation coding tends to produce blocky artifacts, especially in regions with sharp transitions, limiting its ability to preserve fine details and textures in compressed images. Fractal coding, while it is theoretically powerful, it is often requiring extensive computational resources and parameter tuning, rendering it less practical for real-world applications. Subband coding is capable of achieving high compression ratios, but it may struggle with preserving image quality, particularly in terms of preserving fine details or textures.

Transformation coding is widely used, but may introduce artifacts such as blurring or ringing, especially around sharp edges, impacting the visual quality of compressed images. These limitations underscore the need for innovative and efficient compression techniques to address the challenges posed by diverse image content and achieve optimal compression performance. Secondly, lossless methods fully reconstruct the original image using other techniques (i.e., Run-length encoding, LZW coding, Huffman encoding, and

Area coding). Each compression technique has its own set of drawbacks. Run-length encoding is effective for images with long sequences of identical pixels, but it may struggle to compress images with irregular patterns or complex textures efficiently. LZW coding can achieve high compression ratios, often involves significant computational complexity in managing the codebook, which can potentially hinder performance. Huffman encoding, though efficient for images with uniform pixel distributions, may not be as effective for images with uneven frequency distributions, leading to suboptimal compression results.

Additionally, Area coding's reliance on dividing images into fixed-size regions limits its adaptability to images with diverse content, potentially resulting in reduced compression performance when dealing with varied textures or complex structures. These limitations highlight the necessity for innovative approaches in image compression to effectively tackle the challenges posed by diverse image content and achieve optimal compression efficiency. [5], [38]-[41]. When high-quality images are desired and data loss is not an option as in some important applications, loss-less compression is required [6], [7], where the decompressed image should be a bit-for-bit perfect match with the source one. Such applications include medical, satellite, scientific, military, aerospace imaging, and many more domains. In addition, lossless compression is also essential for preserving the integrity of digital images during transmission and storage.

Traditional lossless-based methods apply prediction or integer transforms [7], [44]. In a prediction technique [8], [9], a raster scan order is used to scan and encode an image. Then, the next pixel to be compressed is predicted from the previously compressed ones. As for the integer transform technique [10], [42], [43], integer-to-integer transforms are used to convert the image to a transform domain using an invertible feature. In recent years, the authors in [1], [11], [45], [46] have introduced a flow-based method, which learns rich transformations on high-dimensional data, while the authors in [12] have proposed Bit-Swap method for Asymmetric Numeral Systems [47], [48]. On the other hand, deep learning-based image compression techniques have also gained attention, leveraging the power of neural networks (NN) to learn efficient image representations and compression schemes. Some works [13], [14], [49] predict the coefficients of image transformation using NN, whereas others [15], [16] rely on self-organizing map networks.

Existing compression techniques such as Vector Quantization, Block Truncation Coding, Fractal Coding, Subband Coding, and Transformation Coding always lack concerns or become impractical due to their high computational complexity and the ever-increasing burden on large images storage. In other words, such techniques have shown promising results by presenting good compression performance, at the expense of speed. Furthermore, the high correlation between neighboring pixels in natural images makes it challenging to achieve a high compression ratio.

To address the mentioned challenges, this paper proposes a novel method called LICA-CS for Lossless Image Compression Algorithm using Column Subtraction. LICA-CS takes us a step further in reducing the compression complexity and execution time by combining a simple color transformation method with a reversible compression technique. The color transformation method designed in this paper takes advantage of the high correlation between the three main colors in the RGB color space in natural images (e.g., images from digital cameras).

By producing a new color space that is less correlated, the proposed method produces matrices with smaller pixel values and is invertible with integer arithmetic. Particularly, some integer subtractions are performed in the forward phase with some other addition and subtraction operations performed at the inverse phase. Next, a reversible compression technique is formulated to decrease the size of the transformed image. The compression technique, called Column Subtraction Compression (CSC), takes advantage of the remaining correlation between neighboring pixels, and decreases the image intensities by subtracting each column from the nearest column. The resulting values are saved in the first column starting from the left side of each matrix, reducing the size of the transformed image.

The main contribution of this paper is to present a novel approach to lossless image compression, offering several unique features and innovations. Firstly, we introduce a simple and efficient reversible color transformation method that decreases the correlation between image components in the RGB color space by generating a new color space that is less correlated and has smaller pixel values, this method enhances the compression efficiency without sacrificing image quality. Secondly, we propose a a novel lossless compression model based on subtraction, leveraging the inherent correlation between neighboring pixels to achieve higher compression ratios while reducing the execution time compared to the standard and state-of-the-art works. Lastly, extensive experimental evaluation demonstrates the superiority of our proposed method, LICA-CS, over existing works by significantly reducing storage and transmission loads, particularly in the context of large-scale image data generated daily across various domains.

These distinctive features position LICA-CS as a promising solution to the challenges posed by the exponential growth of image data and the limitations of existing compression techniques. Through comprehensive experimentation and evaluation, our paper aims to demonstrate the effectiveness and superiority of LICA-CS in addressing the identified challenges and advancing the state-of-the-art in image compression technology.

The remaining paper is organized to address the identified issues in the introduction as follows. In section 2, we go over some relevant existing works and discuss their limitations. In section 3, we describe the new proposed model by presenting its encoding and decoding methods and illustrating them through an example. Next, section 4 shows the conducted experiments with their results and substantiates claims of improved performance with quantitative comparisons from experiments or benchmarks, followed by a conclusion in section 5.

II. Related Work

Various lossless image compression techniques have been proposed in the literature, including predictive coding, entropy coding, and transform coding. These techniques achieve high compression ratios by exploiting the statistical redundancies present in the image data. However, each technique has its limitations and trade-offs, and selecting the most appropriate technique depends on the application requirements. In this section, we review the relevant efforts targeting Color Transformation along with Compression methods.

A. Color Transformation

The authors in [17] reviewed the most common color transform, which converts RGB to YCrCb space. They also showed YCoCg transform with its reversible form and how it can enhance the coding gain against YCrCb. Additionally, a new 4-channel transform, called CMYK, was proposed in the work and it was shown to achieve a significant enhancement in coding gain. The author in [18] proposed new simple transformations called RDgDb and LDgEb. The former performs two integer subtractions for each pixel, whereas the latter is based on analog transformations in human vision system. Both transformations showed good results compared to the traditional RCT, YCoCg-R, and KLT [50]. The limitations of RGB to YCrCb and YCoCg transform lie in their susceptibility to introduce color artifacts and inefficiency in handling extreme color variations, while RDgDb may face challenges in accurately representing image details due to its less efficient decorrelation of color channels. Traditional RCT, YCoCg-R, and KLT may suffer from increased computational complexity and reduced adaptability to varying image characteristics, limiting their effectiveness in achieving optimal compression performance.

The work in [19], [51], [52] focused on a new transform method, which is based on lifting technique, showing better decorrelation performance. Particularly, the lifting process relies on two integers representing the scalar gains and the update steps in the lifting butterfly. This is followed by a compression method that is based on hierarchical coding of chrominance channel pixels. In [20], the authors targeted whole-slide images and proposed a transformation method based on mosaic optimization method. First, the image is divided into representative blocks using an efficient heuristic technique. Then, a mosaic of the extracted blocks undergoes an optimization process for further enhancement [53], [54] Such color transformations may suffer from limited effectiveness in reducing computational complexity due to constraints in accurately preserving color information and handling complex image structures.

The authors in [21] proposed a new method for secure image transmission by transforming a secret image into a mosaic one. The process starts by fitting tile images into target blocks, followed by color characteristic transformation. Next, the tile image is rotated into directions with minimum RMSE value, and finally relevant information are embedded for future recovery. One of its limitations in handling complex image structures or colors, and challenges in achieving high compression ratios without significant loss of visual quality.

B. Compression Methods

Lucas et al. [22] proposed a new method, called 3D-MRP, for lossless compression that targets medical images. 3D-MRP is based on the of minimum rate predictors algorithm and uses: (1) 3D-shaped predictors, which leverage the spatial and interframe dependencies of the image, (2) Volume-based optimization, where the predictors depend on a set of frames rather than an individual one, (3) Hybrid 3D-block classification, where consecutive frames can have multiple neighboring blocks assigned to the same class, and (4) Extension to 16 bit-depth images rather than the 8 bit-depth originally designed by MRP [55], [56]. Ton et al. [23] encodes images using Set Partitioning in Hierarchical Trees, SPIHT. The latter is based on integer wavelet transform, divided into Split, Predict, and Update steps. some of the 3D-MRP limitations are the potential difficulty in effectively managing extensive datasets due to computational complexity. Additionally, it may struggle to attain significant compression ratios for specific data types without sacrificing compression speed or memory efficiency.

The interpolating wavelet transform with the form CDF is the one used in this work. The authors in [24] proposed a compression method in spatial domain. The proposed algorithm partitions the image into blocks and employs variable bit allocation to store the pixel values of each block. The determination of the variable bit allocation is based on the pixel values within each block, taking advantage of the correlation between pixels. The authors in [25] leverage both the multi-scale and the pixel-wise approaches and proposed a multi-scale progressive statistical model. The limitation of multi-scale progressive statistical models lies in their susceptibility to increased computational complexity, especially when encoding or decoding high-resolution images, potentially leading to slower processing speeds or higher resource requirements.

The main contribution of the work is to easily adjust the compression rate and compression speed. The authors in [26] employed soft compression and introduced a compressible indicator function for images. This function determines a threshold for the average number of bits needed to represent a location and serves as an illustrative tool for understanding the underlying principle [58]. The proposed work explored and analyzed the application of soft compression in binary images, grayscale images, and multi-component images using defined algorithms, while evaluating different compressible indicator values. The limitation of the highly reliable and low-complexity image compression scheme using the neighborhood correlation sequence algorithm could involve constraints in effectively handling diverse image content and achieving optimal compression ratios, particularly in scenarios with complex image structures or variations in data patterns.

III. The Proposed LICA-CS for Natural Image Compression

In this section, we first describe the overall proposed LICA-CS scheme. Then, we show its reversible encoding and decoding methods, which are composed of a color transformation and a Column Subtraction Compression.

A. *Overview of the Proposed Scheme*

The proposed LICA-CS scheme is a versatile and effective method for lossless image compression that can be used in a wide range of applications. The algorithm is designed to work with any image format and can handle images of any resolution, from low-quality images to high-definition photographs. LICA-CS can be used as a standalone compression technique or as a pre-processing step for other lossless or lossy compression methods. As illustrated in Fig. 1, the LICA-CS image compression scheme consists of two main methods; the first method is the color transformation, and the second method is the CSC. Firstly, the color transformation method converts the input image into a new color space that is better suited for compression.

Using color transform as a preprocessing phase with the proposed image compression algorithms offers several advantages; it enables efficient decorrelation of color channels, separating luminance from chrominance information, which often results in improved compression performance. This decorrelation reduces redundancy in the image data, allowing for more effective compression. Additionally, color transforms can help in adapting the image representation to better suit the characteristics of the compression algorithm, potentially enhancing compression efficiency. Moreover, employing color transforms can facilitate compatibility with various compression techniques and standards, enabling interoperability and versatility in compression workflows. Overall, integrating color transforms as a preprocessing step enhances the effectiveness and adaptability of image compression algorithms, leading to better compression ratios and preserved image quality.

Secondly, The CSC method then compresses the transformed image by removing redundant information in the image's columns. Indeed, one of the significant advantages of CSC compression algorithms is its ability to decrease computational complexity. CSC is relatively simple to implement and requires fewer resources compared to some other image compression algorithms. This reduction in computational demands translates into faster execution times, making CSC an attractive option for applications where speed is crucial, such as real-time image processing or systems with limited computing resources. By streamlining the computational workload, CSC contributes to quicker compression and decompression processes without compromising compression efficiency or image quality.

The combination of these two lossless methods results in a highly compressed image that can be easily decompressed without any loss of quality. The LICA-CS decompression process reverses the compression steps to reconstruct the original image. The color transformation and CSC methods are applied in reverse order to restore the compressed image to its original form.

B. *LICA-CS Encoding & Decoding*

Let NI be a Natural Image, which is colored and represented by a three-color matrices (R, G, B). The three matrices have the same resolution (m×n), where m and n represent the number of rows and columns respectively. Each of the two encoding methods are described as follows:

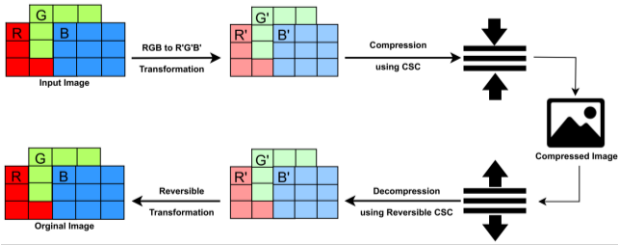

Fig. 1. LICA-CS image compression scheme

Reversible Color Transform: In natural images, the three main colors in the RGB color space typically have high correlation, meaning that the information carried by at least two of the three main components is similar for the same pixel's addresses. For instance, if a particular area of an image is bright in the red channel, then it is likely to be bright in the green and blue channels as well. However, high correlation between image components can make the compression of an image more challenging. Therefore, the aim of color transformation is to reduce the correlation between the image components by producing a new color space that is less correlated. The proposed Reversible Color Transform method achieves this by converting the input natural image NI(R,G,B) from an RGB color space into an RG`B` format. The latter format includes smaller pixel values than the original one, and the overall transformation process outputs an image with less correlated matrices. We denote by G` and B` the transformed Green and Blue matrices respectively, while the red matrix remains unchanged. The Transformed Image is denoted by TI and has the same dimension (m×n) as the input matrices.

Inspired by a popular existing transformation method presented by Starosolski [18], our proposed forward and inverse transformations are defined as follows:

$$\begin{aligned} R &= R \\ G` &= R - G \\ B` &= B - G \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} R &= R \\ G &= R - G` \\ B &= G + B` \end{aligned} \tag{1}$$

Fig. 2 illustrates an input sample of 8×8 block for the RGB matrices, where the transformation equations are applied. Overall, the Reversible Color Transform method reduces the correlation between the image components, making it easier to compress the image without losing important information. The method is reversible, meaning that the original RGB image can be reconstructed from the transformed image without loss of information.

CSC: Image compression aims to reduce the amount of data required to represent an image, which makes it easier to store, transmit, and process. One way to achieve image compression is by taking advantage of the high correlation between neighboring pixels in an image. In other words, each pixel value is similar or very close to the value of its adjacent pixels [18]. To achieve high compression ratio using the similarity of nearby pixels, we developed CSC to decrease the image intensities by subtracting each column from the nearest column and saving the resulting value in the first column starting from the first column from the left side of each matrix. Thus, CSC decreases the size of the transformed image TI by converting it into a Compressed Image

CI(RcG`cB`c) Where RcG`cB`c represent the compressed Red, Green, and Blue channels respectively. Formally, the compression equations are defined as follows:

$$Rc(i,j) = R(i,j) - R(i,j+1) \quad \forall 1 \le i \le m \,\&\, 1 \le j \le n-1 \tag{2}$$

$$G`c(i,j) = G`(i,j) - G`(i,j+1) \quad \forall 1 \le i \le m \,\&\, 1 \le j \le n-1 \tag{3}$$

$$B`c(i,j) = B`(i,j) - B`(i,j+1) \quad \forall 1 \le i \le m \,\&\, 1 \le j \le n-1 \tag{4}$$

Sample 8x8 for the Red Matrix (R)									R = R							
99	99	99	99	99	99	99	99	=	99	99	99	99	99	99	99	99
99	99	99	99	100	100	100	100		99	99	99	99	100	100	100	100
99	99	99	102	98	95	96	96		99	99	99	102	98	95	96	96
99	99	99	105	79	59	69	72		99	99	99	105	79	59	69	72
99	99	99	105	85	72	87	90		99	99	99	105	85	72	87	90
99	99	99	103	94	88	103	95		99	99	99	103	94	88	103	95
99	99	99	102	98	97	96	101		99	99	99	102	98	97	96	101
99	99	99	100	101	97	100	105		99	99	99	100	101	97	100	105
Sample 8x8 for the Green Matrix (G)									**G′ = R - G**							
99	99	99	99	99	99	99	99	→	0	0	0	0	0	0	0	0
99	99	99	99	98	99	99	99		0	0	0	0	2	1	1	1
99	99	99	100	96	93	95	95		0	0	0	2	2	2	1	1
99	99	99	103	75	55	63	66		0	0	0	2	4	4	6	6
99	99	99	103	81	66	79	82		0	0	0	2	4	6	8	8
99	99	99	102	90	82	95	88		0	0	0	1	4	6	8	7
99	99	99	100	95	92	91	93		0	0	0	2	3	5	5	8
99	99	99	99	98	92	96	101		0	0	0	1	3	5	4	4
Sample 8x8 for the Blue Matrix (B)									**B′ = B - G**							
99	99	99	99	99	99	99	99	→	0	0	0	0	0	0	0	0
99	99	99	99	99	96	96	96		0	0	0	0	1	-3	-3	-3
99	99	99	100	94	91	90	90		0	0	0	0	-2	-2	-5	-5
99	99	99	101	71	49	54	55		0	0	0	-2	-4	-6	-9	-11
99	99	99	101	74	55	64	67		0	0	0	-2	-7	-11	-15	-15
99	99	99	97	84	71	80	70		0	0	0	-5	-6	-11	-15	-18
99	99	99	96	86	79	73	75		0	0	0	-4	-9	-13	-18	-18
99	99	99	94	88	79	77	80		0	0	0	-5	-10	-13	-19	-21

Fig. 2. Sample of 8×8 block before and after transformation phase

To reconstruct the compressed image CI, the three matrices RcG`cB`c are loaded to the decompression algorithm, which applies the reversible CSC equations for each of the three matrices as follows:

$$R(i,j-1) = Rc(i,j) + Rc(i,j-1) \quad \forall 1 \le i \le m \,\&\, n-1 \le j \le 2 \tag{5}$$

$$G`(i,j-1) = G`c(i,j) - G`c(i,j-1) \quad \forall 1 \le i \le m \,\&\, n-1 \le j \le 2 \tag{6}$$

$$B`(i,j-1) = B`c(i,j) - B`c(i,j-1) \quad \forall 1 \le i \le m \,\&\, n-1 \le j \le 2 \tag{7}$$

Fig. 3 shows the matrices obtained using the proposed CSC method. LICA-CS, which consists of the Reversible Color Transform and CSC, is simple and fast, with a time complexity of O(n2) for the encoding and decoding phases. In addition to its application for color images, the proposed LICA-CS method can also be effectively utilized for greyscale images. When compressing greyscale images, the compression phase involves the use of Eq. (2), which performs the necessary size reduction. Similarly, during the decompression phase, Eq. (5) is applied to recover the original image.

R									R_c							
99	99	99	99	99	99	99	99	=	0	0	0	0	0	0	0	99
99	99	99	99	100	100	100	100		0	0	0	-1	0	0	0	100
99	99	99	102	98	95	96	96		0	0	-3	4	3	-1	0	96
99	99	99	105	79	59	69	72		0	0	-6	26	20	-10	-3	72
99	99	99	105	85	72	87	90		0	0	-6	20	13	-15	-3	90
99	99	99	103	94	88	103	95		0	0	-4	9	6	-15	8	95
99	99	99	102	98	97	96	101		0	0	-3	4	1	1	-5	101
99	99	99	100	101	97	100	105		0	0	-1	-1	4	-3	-5	105
G′									G'_c							
0	0	0	0	0	0	0	0	→	0	0	0	0	0	0	0	0
0	0	0	0	2	1	1	1		0	0	0	-2	1	0	0	1
0	0	0	2	2	2	1	1		0	0	-2	0	0	1	0	1
0	0	0	2	4	4	6	6		0	0	-2	-2	0	-2	0	6
0	0	0	2	4	6	8	8		0	0	-2	-2	-2	-2	0	8
0	0	0	1	4	6	8	7		0	0	-1	-3	-2	-2	1	7
0	0	0	2	3	5	5	8		0	0	-2	-1	-2	0	-3	8
0	0	0	1	3	5	4	4		0	0	-1	-2	-2	1	0	4
B′									B'_c							
0	0	0	0	0	0	0	0	→	0	0	0	0	0	0	0	0
0	0	0	0	1	-3	-3	-3		0	0	0	-1	4	0	0	-3
0	0	0	0	-2	-2	-5	-5		0	0	0	2	0	3	0	-5
0	0	0	-2	-4	-6	-9	-11		0	0	2	2	2	3	2	-11
0	0	0	-2	-7	-11	-15	-15		0	0	2	5	4	4	0	-15
0	0	0	-5	-6	-11	-15	-18		0	0	5	1	5	4	3	-18
0	0	0	-4	-9	-13	-18	-18		0	0	4	5	4	5	0	-18
0	0	0	-5	-10	-13	-19	-21		0	0	5	5	3	6	2	-21

Fig. 3. Sample of 8×8 block before and after compression phase

The LICA-CS is designed as a simple and high-speed algorithm, thanks to its low complexity and scalability, which allow it to effectively manage large-scale image datasets. Notably, LICA-CS seamlessly accommodates images of any size. Its method of subtracting values for image columns makes it straightforward and adaptable for all image types, as images can be uniformly converted into integer matrix values. This inherent flexibility highlights its capability for diverse image compression tasks.

IV. NUMERICAL RESULTS

Thorough experiments were performed to evaluate the performance of our proposed LICA-CS. In this section, we briefly describe the used datasets, the baseline approaches that we compared our work to, then we discuss the obtained results.

A. About the Datasets

The following datasets of 8-bit RGB images were used for the evaluation:

- Waterloo [61], which is sourced from the Fractal Coding and Analysis Group - University of Waterloo, is a well-known dataset and widely used by researchers to evaluate the performance of various algorithms across different imaging tasks. It includes three sets: Greyscale Set 1, Greyscale Set 2, and Color Set, consisting of 12 small, 12 medium, and 8 large images respectively.

- Kodak [62], which is sourced from Kodak company, is used as a benchmark dataset in the field of image processing and compression. It consists of 24 photographic images that cover various subjects and scenes.

- EPFL [63], which is sourced from Ecole Polytechnique Féderale de Lausanne, is also a widely recognized set of images used in various research studies, particularly in the field of image processing and computer vision. It consists of 10 high-resolution colored images.

B. Experiments Setup

The experiments were conducted on a computer system with the following specifications: Intel(R) Core(TM) i7-10510U CPU running at 1.80-2.30 GHz, 8 GB of RAM, and 1 GB of virtual memory. The algorithms were implemented using the MATLAB programming environment.

C. Experimental Results

a) Compression ratios compared with standard algorithms: We start by studying the compression ratios for transformed images, which are expressed in bits per pixel (bpp). The ratio is measured using:

$$\frac{8 \times s}{n} \tag{8}$$

where n denotes the number of pixels in the studied image, and s represents the combined size, measured in Bytes, of the separately compressed three components of the transformed image. The results obtained from this formula align with those obtained when dividing the compressed image size by the number of pixels. Therefore, smaller ratios indicate more effective compression.

We compare our proposed transformation (RG`B`) and CSC compression (RcG`cB`c) methods to a combination of different standards. The latter include: RCT and YCoCg transformations along with JPEG2000, JPEG-LS, JPEG XR, and Huffman algorithms. All the mentioned methods are applied on the three datasets. The obtained compression ratio results are quantified in Table I, where the best three ratio values are indicated in bold. We can notice that our methods outperform in most cases the standard approaches, which are very popular and well-recognized. The highest performance is achieved when combining YCoCg with our CSC compression (method #J), followed by RCT + CSC (method #E), and then RG`B` + CSC (method #N).

The findings indicate that employing the CSC algorithm as a postprocessing phase following transformation produces the highest performance. The most favorable outcomes emerge when combining YCoCg with the proposed CSC, showcasing an average compression ratio of 6.53. This integration highlights the efficacy of utilizing YCoCg alongside the proposed CSC for optimal compression results. This is attained by the YCoCg color space, which separates luminance from chrominance data, thereby diminishing redundancy and amplifying compression effectiveness. Such results show how robust our *CSC* method is. Specifically, in method *#G*, when comparing *YCoCg* with the best-performing standard approach, namely JPEG-LS, the results show that *CSC* achieves an average improvement of *33.03%*. Similar interpretation applies when studying the performance of *RCT* (method *#B*), leading to *29.99%* improvement for *CSC*. Furthermore, it is noteworthy that the *CSC* method, without any accompanying transformation, surpasses all the studied algorithms that apply *RCT* or YCoCg transformations. On the other hand, the proposed transformation by itself demonstrates favorable performance, surpassing *RCT* by an average of *2.11 bpp* when applied to Huffman.

b) Running time compared with standard algorithms: For the best three reported methods in terms of compression ratio (methods J, E, and N), we studied their running time for all the datasets. Table II, Table III, and Table IV show the compression and decompression time for Waterloo, Kodak, and EPFL respectively. Although methods J and E have better compression ratios, method N achieves faster running times for every image in the compression and decompression phases. On average, RG`B` + CSC (method N) is 1.79 and 1.5 seconds faster in compressing EPFL images than methods J and E respectively. Faster results for method N are also reported for Waterloo and Kodak images, according to the values presented in Table II, Table III and Table IV.

This is accomplished through employing RG`B` transformations as a preprocessing step. This decorrelation minimizes redundancy within the image data, enabling more efficient compression. Additionally, RG`B` transformations are computationally simpler compared to certain other color spaces, leading to quicker compression and decompression processes. Another reason for achieving fast execution time is the simplicity of implementing the CSC, which requires fewer resources, further enhancing the overall speed of the compression algorithm.

TABLE I. AVERAGE COMPRESSION RATIOS (BPP) FOR OUR PROPOSED RG`B` AND CSC METHODS COMPARED WITH STANDARD ALGORITHMS

Method		Comp	Datasets			Average
No	TF		Waterloo	Kodak	EPFL	
A	RCT	JPEG2000	11.21	9.51	10.84	10.52
B	RCT	JPEG LS	8.96	9.57	10.47	9.67
C	RCT	JPEG XR	13.32	10.92	11.76	12.00
D	RCT	Huffman	16.55	15.06	16.9	16.17
E	RCT	CSC	6.81	6.33	7.17	6.77
F	YC_oC_g	JPEG2000	11.21	9.49	10.98	10.56
G	YC_oC_g	JPEG-LS	9.02	9.60	10.61	9.75
H	YC_oC_g	JPEG-XR	13.27	10.86	11.9	12.01
I	YC_oC_g	Huffman	17.59	16.04	17.74	17.12
J	YC_oC_g	CSC	6.6	6.16	6.84	6.53
K	-	Huffman	15.25	16.49	16.91	16.22
L	RG`B`	Huffman	13.65	13.52	15.01	14.06
M	-	CSC	7.94	9.08	9.12	8.71
N	RG`B`	CSC	7	6.66	7.48	7.05

Fig. 4 presents a concise visual representation of the data provided in Table I, facilitating easier comprehension. It outlines the Average Compression Ratios (BPP) attained by the Proposed LICA-CS Algorithm in comparison with Standard Algorithms.

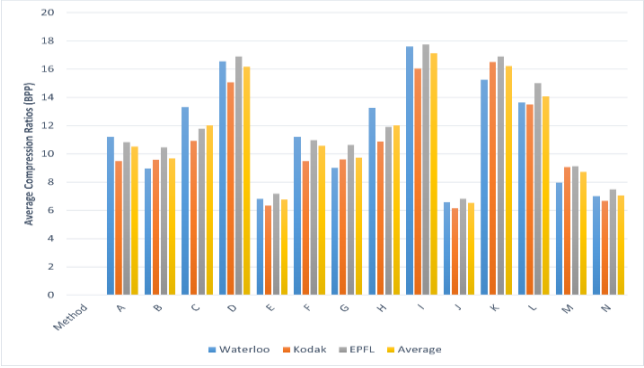

Fig. 4. Average compression ratios (BPP) for the proposed LICA-CS algorithm compared with standard algorithms

TABLE II. COMPRESSION AND DECOMPRESSION TIME (IN SECS) FOR OUR PROPOSED RG`B` AND CSC METHODS USING WATERLOO IMAGES

Image	Compression Time (s)			Decompression Time (s)		
	Method #J	Method #E	Method #N	Method #J	Method #E	Method #N
Clegg	1.40	1.26	0.78	0.03	0.04	0.03
Frymire	3.40	2.74	1.55	0.06	0.05	0.04
Lena3	0.26	0.23	0.12	0.02	0.02	0.01
Monarch	0.31	0.30	0.13	0.03	0.03	0.01
Peppers3	0.23	0.22	0.12	0.02	0.02	0.01
Sail	0.32	0.33	0.15	0.03	0.02	0.01
Serrano	0.66	0.60	0.35	0.03	0.03	0.02
Tulips	0.32	0.30	0.15	0.03	0.03	0.02
Average	0.86	0.75	0.42	0.03	0.03	0.02

Illustrating Table II allows us to observe in Fig. 5 the cumulative compression and decompression time in seconds for the proposed LICA-CS algorithms when applied to Waterloo Images.

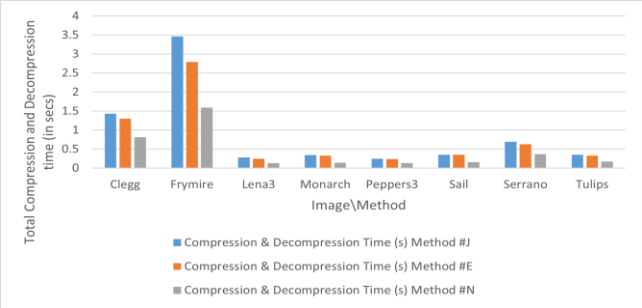

Fig. 5. Total compression and decompression time (S) for the proposed LICA-CS algorithms using waterloo images

TABLE III. COMPRESSION AND DECOMPRESSION TIME (IN SECS) FOR OUR PROPOSED RG`B` AND CSC METHOD USING KODAK IMAGES

Image	Compression Time (s)			Decompression Time (s)		
	Method #J	Method #E	Method #N	Method #J	Method #E	Method #N
kodim01	0.34	0.35	0.15	0.02	0.03	0.01
kodim02	0.28	0.26	0.13	0.03	0.03	0.02
kodim03	0.27	0.25	0.14	0.03	0.03	0.01
kodim04	0.26	0.25	0.13	0.03	0.03	0.02
kodim05	0.37	0.37	0.17	0.03	0.03	0.01
kodim06	0.34	0.31	0.15	0.03	0.03	0.01
kodim07	0.26	0.26	0.13	0.03	0.03	0.02
kodim08	0.42	0.41	0.15	0.03	0.03	0.01
kodim09	0.32	0.34	0.14	0.03	0.03	0.01
Kodim10	0.34	0.35	0.17	0.03	0.03	0.01
Kodim11	0.34	0.31	0.14	0.03	0.03	0.01
Kodim12	0.33	0.30	0.14	0.03	0.03	0.02
Kodim13	0.40	0.38	0.16	0.03	0.03	0.01
Kodim14	0.35	0.35	0.17	0.03	0.03	0.02
Kodim15	0.33	0.31	0.14	0.03	0.03	0.01
Kodim16	0.22	0.21	0.09	0.01	0.03	0.01
Kodim17	0.34	0.33	0.16	0.03	0.03	0.02
Kodim18	0.44	0.40	0.18	0.02	0.04	0.02
Kodim19	0.35	0.34	0.16	0.03	0.03	0.02
Kodim20	0.37	0.40	0.17	0.03	0.03	0.01
Kodim21	0.34	0.33	0.14	0.03	0.03	0.01
Kodim22	0.39	0.37	0.17	0.03	0.03	0.01
Kodim23	0.29	0.30	0.14	0.03	0.03	0.02
Kodim24	0.41	0.37	0.18	0.03	0.03	0.01
Average	0.34	0.33	0.15	0.03	0.03	0.02

Fig. 6 elucidates the total compression and decompression time is seconds that extracted from Table III, contrasting the performance of the Proposed LICA-CS Algorithm when applied to Kodak Images.

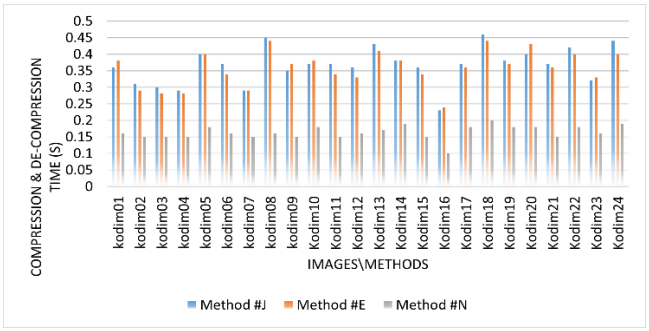

Fig. 6. Total compression and decompression time (S) for the proposed LICA-CS algorithm compared with standard algorithms using kodak images

TABLE IV. COMPRESSION AND DECOMPRESSION TIME (IN SECS) FOR OUR PROPOSED RG`B` AND CSC METHODS USING EPFL IMAGES

Image	Compression Time (s)			Decompression Time (s)		
	Method #J	Method #E	Method #N	Method #J	Method #E	Method #N
Bike	4.72	4.36	2.17	0.09	0.08	0.08
Cafe	5.88	5.37	2.83	0.08	0.08	0.07
P01	3.31	3.12	1.81	0.1	0.1	0.07
P04	3.79	3.45	2.46	0.09	0.08	0.08
P06	4.44	4.11	2.49	0.1	0.09	0.09
P10	3.01	2.86	1.47	0.11	0.08	0.09
P14	3.05	2.87	1.71	0.09	0.09	0.08
P22	2.85	2.58	1.57	0.09	0.09	0.08
P30	3.91	3.47	2.4	0.1	0.09	0.08
Women	3.5	3.36	1.65	0.1	0.09	0.08
Average	3.84	3.55	2.05	0.1	0.09	0.08

Fig. 7 illustrates the total compression and decompression time in seconds for the Proposed LICA-CS algorithm when applied to EPFL images.

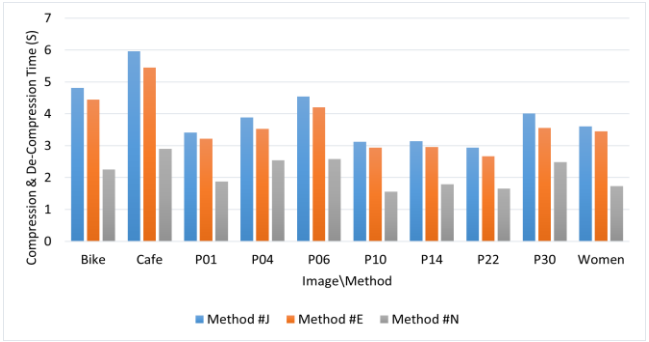

Fig. 7. Total compression and decompression time (in secs) for the proposed LICA-CS algorithm compared with standard algorithm using EPFL images

c) Compression ratios compared with state-of-the-art schemes: We also evaluated the performance of our proposed methods with other state-of-the-art schemes, including BWCA [27], [57], GST (based on KMTF-BWCA) [28], [58], and BBWCA [29], [59], [60]. Table V shows their images sizes and their compression ratios using Eq. (8) for selected Kodak images. On average, our method outperforms the others with an improvement of 59.8, 57.87, and 5.2% in the

compression ratio compared to BWCA, GST, and BBWCA respectively.

TABLE V. IMAGE SIZES (KB) AND COMPRESSION RATIOS (BPP) FOR OUR PROPOSED RG`B` AND CSC METHODS COMPARED WITH STATE-OF-THE-ART SCHEMES USING KODAK IMAGES

	BWCA		GST		BBWCA		LICA-CS	
Image	Size	Ratio	Size	Ratio	Size	Ratio	Size	Ratio
kodim02	765	15.89	750	15.58	381	7.95	320	6.67
kodim05	991	20.69	947	19.67	244	5.08	376	7.82
kodim07	780	16.22	743	15.48	350	7.29	282	5.88
kodim08	1008	21.05	981	20.51	463	9.64	395	10.48
kodim11	837	17.39	800	16.67	297	6.19	307	6.40
kodim13	1020	21.24	965	20.17	352	7.34	396	8.25
kodim20	591	12.31	560	11.65	332	6.92	270	5.63
kodim21	827	17.27	783	16.33	480	10.00	311	6.49
kodim22	891	18.60	839	17.52	358	7.45	355	7.41
kodim23	791	16.44	739	15.38	346	7.21	297	6.20
Average	850	17.71	811	16.9	360	7.51	330.9	7.12

d) Running time compared with state-of-the-art schemes: To accurately assess the computational efficiency of our proposed method and provide a fair and meaningful comparison, we conducted experiments using identical experimental setups and settings as the existing methods presented in the previous section. For this specific set of experiments only, we utilized a different machine with the following specifications: Intel Core 2 Quad CPU running at 2.4 GHz, 2 GB of RAM, and 1 GB of virtual memory. By adopting this approach, we compare the running time of our method with the previously reported results in Table VI. Significant improvements were observed in the compression and decompression processes of RG`B`-CSC when applied to the presented images. This demonstrates the robustness and efficiency of the proposed transformation and compression methods.

TABLE VI. COMPRESSION AND DECOMPRESSION TIME (IN SECS) FOR OUR PROPOSED **RG`B`** AND CSC METHODS COMPARED WITH STATE-OF-THE-ART SCHEMES USING SELECTED KODAK IMAGES

	Compression Time (s)				Decompression Time (s)			
Image	BWCA	GST	BBWCA	LICA-CS	BWCA	GST	BBWCA	LICA-CS
kodim02	16.46	17.21	17.70	1.43	23.76	25.39	28.06	0.1
kodim05	16.07	16.82	19.14	2.03	23.85	28.09	30.50	0.1
kodim07	17.58	18.33	18.41	1.48	22.46	25.95	30.60	0.1
kodim08	16.32	17.07	19.77	1.79	21.32	24.82	27.80	0.1
kodim11	18.02	18.77	19.49	1.82	23.34	23.24	28.55	0.1
kodim13	17.37	18.12	18.18	1.99	22.77	25.55	28.66	0.1
kodim20	18.82	19.57	20.98	2.07	21.17	24.40	27.46	0.1
kodim21	19.17	19.92	19.48	1.85	21.46	22.63	28.95	0.1
kodim22	16.91	17.66	19.60	2.21	21.97	22.19	22.87	0.1
kodim23	16.78	17.53	19.95	1.69	22.90	25.98	29.23	0.1
Average	17.35	18.10	19.27	1.83	22.50	24.82	28.27	0.1

Fig. 8 displays a chart describing the compressed image size in kilobytes (KB) for the Proposed LICA-CS algorithm in contrast to State-of-the-Art schemes when applied to KODAK images.

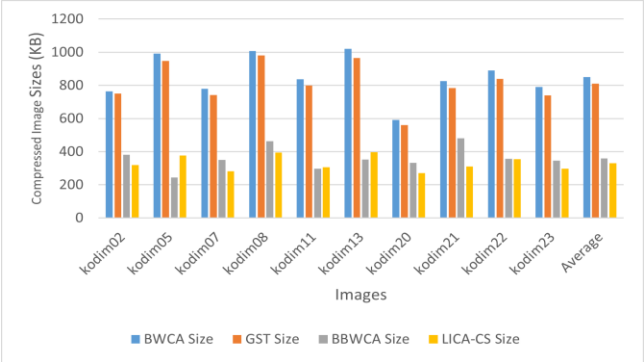

Fig. 8. Compressed image size (KB) for the proposed LICA-CS algorithm compared with state-of-the-art schemes using KODAK images

Fig. 9 summarizes the data extracted from Table VI, presenting the total compression and decompression time (in seconds) for the Proposed LICA-CS algorithm in comparison to State-of-the-Art schemes when applied to KODAK images.

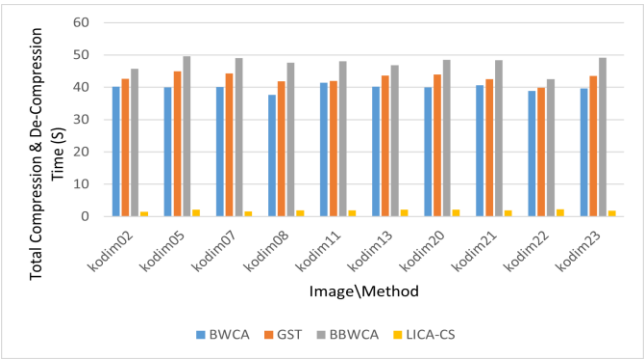

Fig. 9. Total compression and decompression time (in secs) for the proposed LICA-CS algorithm compared with state-of-the-art schemes using KODAK images

Comparison with Benchmark Schemes: Besides the reported results, we compared our LICA-CS method with several benchmark software and compression algorithms that are commonly used in the field of image compression. The performance of each method [29] is evaluated in terms of compression efficiency using Kodak dataset. Table VII presents the obtained compressed file sizes in KBs after applying the respective compression methods. We can observe significant improvements in terms of compression efficiency for LICA-CS. Specifically, our method compresses the 11520 KBs of data to 3310 KBs, which is a size reduction of 71.27%, outperforming all benchmark software/algorithm approaches. The interesting observation that was made from Table I when analyzing the results of YCoCg-CSC is better shown in this comparison in Table VII, where the images are compressed to 3093 KBs. Additionally, BBWCA demonstrated competitive compression ratios, with a total file size of 3553 KBs, a reduction of 69.16%. HEVC and LZ4X, on the other hand, achieved impressive compression results with an average reduction of 63.37% and 62.6% respectively.

TABLE VII. COMPRESSED IMAGE SIZES (KB) FOR OUR METHODS ALONG WITH SEVERAL BENCHMARK SOFTWARE AND COMPRESSION ALGORITHMS

KODAK TEST IMAGE											
Scheme	Kodim02	Kodim08	Kodim13	Kodim21	Kodim22	Kodim21	Kodim07	Kodim20	Kodim05	Kodim11	Total
ADVANCE COMP	608	422	495	493	477	666	460	468	614	451	5154
ALLUME	386	576	487	503	699	407	788	370	410	372	4998
BBWCA	381	463	352	480	308	346	350	332	244	297	3553
BCM	457	785	628	490	532	737	375	381	507	390	5282
BULK ZIP	450	710	372	441	753	613	469	751	482	603	5644
CAESIUM	475	422	496	518	797	500	706	407	736	411	5468
C-MIX	453	510	457	497	625	520	761	666	657	456	5602
COMPRESSOR.IO	413	765	425	597	380	415	392	391	603	668	5049
CRUSH	409	692	556	536	544	795	646	612	583	376	5749
FILE MINIMIZER	560	758	471	521	740	681	691	596	573	495	6086
FILE OPTIMIZER	409	432	721	665	559	502	401	447	618	518	5272
HEVC (x265)	403	493	343	487	397	538	326	418	462	353	4220
LICA-CS	320	395	396	311	355	297	282	270	376	348	3310
LZ4X	370	420	412	373	402	537	465	456	444	429	4308
MRP	497	760	550	511	513	791	534	474	628	409	5667
NANOZIP	475	556	472	519	446	543	490	431	551	715	5198
PAQ8PXD_V4	450	490	598	607	489	655	733	596	372	569	5559
UPACK 0.25	661	710	379	675	503	529	371	572	587	568	5555
WINRK 3.1.2	598	515	398	783	674	457	374	611	456	593	5459
YCoCg-CSC	296	380	380	287	334	265	254	259	355	283	3093
ZCM 0.92	495	631	772	542	408	714	565	597	416	450	5590

Fig. 10 illustrates the data from Table VII through a chart representing the compressed image size (KB) for the Proposed LICA-CS algorithm, alongside various benchmark software and compression algorithms.

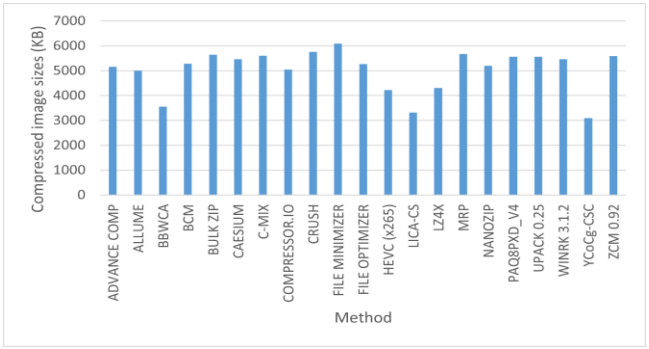

Fig. 10. Compressed image size (KB) for the proposed LICA-CS algorithm along with several benchmark software and compression algorithms

e) Comparison of Compression ratios for grayscale images: In the comparison of 12 selected images from the Waterloo grayscale image sets 1 and 2, we evaluated the compression ratios achieved by applying the algorithms from [30]. The results are presented in Table VIII, indicating the extent to which each algorithm compressed the image. On average, our method demonstrated superior performance compared to all other algorithms, achieving a compression ratio that was 13% better than the IWT-HF.

TABLE VIII. COMPRESSION RATIOS (BPP) FOR OUR PROPOSED RG`B` - CSC METHODS COMPARED WITH STATE-OF-THE-ART SCHEMES FROM [30] USING 12 SELECTED WATERLOO IMAGES

Image	Huffman	Arithmetic	J2K	JLS	7-Zip	IWT-HF	LICA-CS
bird.TIF	6.8016	6.7747	3.1390	3.4712	4.2333	2.8630	2.45
bridge.TIF	7.6937	7.6689	5.9086	5.7904	6.3164	4.9043	4.00
circles.TIF	1.8484	1.7811	1.2627	0.1526	0.1136	1.3375	1.11
crosses.TIF	1.0001	0.1879	1.4285	0.3855	0.1786	2.0281	1.24
slope.TIF	7.5411	7.5177	1.0643	1.5713	1.6929	1.6449	1.77
squares.TIF	1.3517	1.0776	0.2505	0.0771	0.0500	0.6989	1.05
boat.TIF	7.1468	7.1238	4.1005	4.2498	5.2881	3.5184	3.13
goldhill2.TIF	7.4970	7.4779	4.6544	4.7116	5.5985	3.8035	3.29
lena2.TIF	7.4683	7.4456	4.0166	4.2437	5.5190	3.2545	3.12
library.TIF	5.8704	5.8490	5.8350	5.1011	4.2544	5.1664	3.72
mandrill.TIF	7.3804	7.3580	6.0232	6.0365	6.3869	5.0441	4.24
peppers.TIF	7.5951	7.5716	4.4042	4.4887	5.5480	3.5105	3.16
Average	5.7662	5.6528	3.5073	3.3566	3.7650	3.1478	2.69

Fig. 11 depicts the average Compression Ratios (BPP) for the proposed LICA-CS algorithm compared to State-of-the-Art schemes, utilizing selected Waterloo images as described in Table VIII.

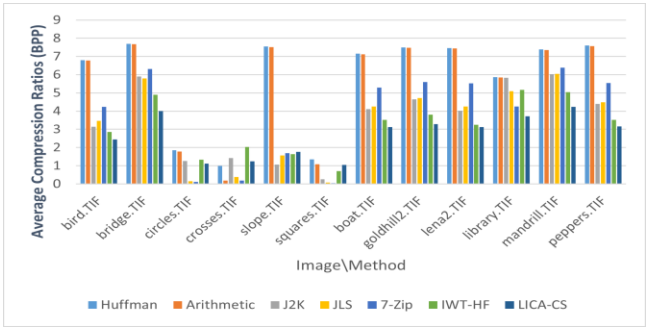

Fig. 11. Average compression ratios (BPP) for the proposed LICA-CS algorithm with state-of-the-art schemes using selected waterloo images

V. CONCLUSIONS

In this paper, we proposed a lossless image compression algorithm (LICA-CS) that uses a transformation method to reduce the correlation between image components to prepare the image for the next compression stage that requires less correlation, followed by a compression method called Columnar Subtraction Model to reduce image intensity. by taking advantage of the similarity and proximity of pixel values within adjacent columns, providing a clear gain in reducing image size, this is accomplished by subtracting neighboring columns from each other in each of the three-color matrices within the targeted image. Thorough experiments were conducted to compare LICA-CA with cutting-edge algorithm and various performance benchmarks demonstrated substantial enhancements for LICA-CS, particularly in compression ratio and execution time.

Improvement for LICA-CS in term of compression ratio, our method compresses the 11520 KBs of data to 3310 KBs, which is a size reduction of 71.27%, outperforming all benchmark software/algorithm approaches. The interesting observation that was made from Table I when analyzing the

results of YCoCg-CSC is better shown in this comparison in Table VII, where the images are compressed to 3093 KBs. Additionally, BBWCA demonstrated competitive compression ratios, with a total file size of 3553 KBs, a reduction of 69.16%. HEVC and LZ4X, on the other hand, achieved impressive compression results with an average reduction of 63.37% and 62.6%, respectively.

Improvement for LICA-CS in terms of execution time, we compare the running time of our method with the previously reported results in Table VI. Significant improvements were observed in the compression and decompression processes of LICA-CS when applied to the presented images. The proposed method achieved an average of 1.93 seconds, compared with 40.35 seconds needed by BWCA.

Improvement for LICA-CS in terms of implementation complexity, while the fundamental concepts of BBWCA are relatively simple, implementing it can become complex, particularly when handling large images. The efficient encoding and decoding of the BBWCA imposes detailed management of indices and suffix arrays, which can impose significant computational demands. In contrast, LICA-CS offers a simpler implementation and is a high-performance algorithm capable of effectively managing large image sizes or complex data.

As a future direction, our aim is to expand the scope of this work to include the compression of low-resolution images, specifically focusing on raster map images which often have less redundancy compared to higher-resolution images, making it more challenging for compression algorithms to identify and exploit redundancy for efficient compression. By extending our methods to address the unique challenges posed by this type of imagery, we anticipate further advancements in efficient and effective compression algorithms.

ACKNOWLEDGMENT

I would like to acknowledge the initial support received from British University of Bahrain. This support played a vital role in facilitating this research.

REFERENCES

[1] E. Hoogeboom, J. Peters, R. V. D. Berg, and M. Welling, "Integer discrete flows and lossless compression," *Advances in Neural Information Processing Systems*, vol. 32, 2019, doi: 10.48550/arXiv.1905.07376.

[2] M. Zhang, X. Tong, Z. Wang, and P. Chen, "Joint lossless image compression and encryption scheme based on calic and hyperchaotic system," *Entropy*, vol. 23, no. 8, p. 1096, 2021, doi: 10.3390/e23081096.

[3] H. Zhang, X.-Q. Wang, Y.-J. Sun, and X.-Y. Wang, "A novel method for lossless image compression and encryption based on lwt, spiht and cellular automata," *Signal Processing: Image Communication*, vol. 84, p. 115829, 2020, doi: 10.1016/j.image.2020.115829.

[4] M. A. Rahman and M. Hamada, "Lossless image compression techniques: A state-of- the-art survey," *Symmetry*, vol. 11, no. 10, p. 1274, 2019, doi: 10.3390/sym11101274.

[5] A. P. Singh, A. Potnis, and A. Kumar, "A review on latest techniques of image compression," *International Research Journal of Engineering and Technology (IRJET)*, vol. 3, no. 7, pp. 2395–0056, 2016.

[6] H. Rhee, Y. I. Jang, S. Kim, and N. I. Cho, "Lossless image compression by joint prediction of pixel and context using duplex neural networks," *IEEE Access*, vol. 9, pp. 86632–86645, 2021, doi: 10.1109/ACCESS.2021.3088936.

[7] S. Gumus and F. Kamisli, "A learned pixel-by-pixel lossless image compression method with 59k parameters and parallel decoding," *arXiv preprint arXiv:2212.01185*, 2022, doi: 10.48550/ arXiv.2212.01185.

[8] J. Sneyers and P. Wuille, Flif: Free lossless image format based on maniac compression," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 66–70, 2016, doi: 10.1109/ICIP.2016.7532320.

[9] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The loco-i lossless image compression algorithm: Principles and standardization into jpeg-ls," *IEEE Transactions on Image processing*, vol. 9, no. 8, pp. 1309–1324, 2000, doi: 10.1109/83.855427.

[10] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The jpeg2000 still image coding system: an overview," *IEEE transactions on consumer electronics*, vol. 46, no. 4, pp. 1103–1127, 2000, doi: 10.1109/30.920468.

[11] J. Ho, E. Lohn, and P. Abbeel, "Compression with flows via local bits-back coding," *Advances in Neural Information Processing Systems*, vol. 32, 2019, doi: 10.48550/arXiv.1905.08500.

[12] F. Kingma, P. Abbeel, and J. Ho, "Bit-swap: Recursive bits-back coding for lossless compression with hierarchical latent variables," in *International Conference on Machine Learning*, pp. 3408–3417, 2019, doi: 10.48550/arXiv.1905.06845.

[13] A. J. Hussain, D. Al-Jumeily, N. Radi, and P. Lisboa, "Hybrid neural network predictive-wavelet image compression system," *Neurocomputing*, vol. 151, pp. 975–984, 2015, doi: 10.1016/j.neucom.2014.02.078.

[14] B. Perumal and M. P. Rajasekaran, "A hybrid discrete wavelet transform with neural network back propagation approach for efficient medical image compression," in *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, pp. 1–5, 2016, doi: 10.1109/ICETETS.2016.7603060.

[15] D. R. Hidalgo, B. B. Cortés, and E. C. Bravo, "Dimensionality reduction of hyperspec- tral images of vegetation and crops based on self-organized maps," *Information Processing in Agriculture*, vol. 8, no. 2, pp. 310–327, 2021, doi: 10.1016/j.inpa.2020.07.002.

[16] J. Han, "Texture image compression algorithm based on self-organizing neural network," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, p. 4865808, 2022, doi: 10.1155/2022/4865808.

[17] H. S. Malvar, G. J. Sullivan, and S. Srinivasan, "Lifting-based reversible color transformations for image compression," *Applications of digital image processing XXXI*, vol. 7073, pp. 44–53, 2008, doi: 10.1117/12.797091.

[18] R. Starosolski, "New simple and efficient color space transformations for lossless image compression," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 1056–1063, 2014, doi: 10.1016/j.jvcir.2014.03.003.

[19] S. Kim and N. I. Cho, "A lossless color image compression method based on a new reversible color transform," in *2012 Visual Communications and Image Processing*, pp. 1–4, 2012, doi: 10.1109/VCIP.2012.6410808.

[20] M. Hernàndez-Cabronero, V. Sanchez, I. Blanes, F. Auli-Llinas, M. W. Marcellin, and J. Serra-Sagristà, "Mosaic-based color-transform optimization for lossy and lossy-to-lossless compression of pathology whole-slide images," *IEEE transactions on medical imaging*, vol. 38, no. 1, pp. 21–32, 2018, doi: 10.1109/TMI.2018.2852685.

[21] Y.-L. Lee and W.-H. Tsai, "A new secure image transmission technique via secret- fragment-visible mosaic images by nearly reversible color transformations," *IEEE Transactions on circuits and systems for video technology*, vol. 24, no. 4, pp. 695–703, 2013, doi: 10.1109/TCSVT.2013.2283431.

[22] L. F. Lucas, N. M. Rodrigues, L. A. Silva Cruz, and S. M. Faria, "Lossless compression of medical images using 3-d predictors," *IEEE transactions on medical imaging*, vol. 36, no. 11, pp. 2250–2260, 2017, doi: 10.1109/TMI.2017.2714640.

[23] X.-J. Tong, P. Chen, and M. Zhang, "A joint image lossless compression and encryp- tion method based on chaotic map," *Multimedia Tools and Applications*, vol. 76, pp. 13995–14020, 2017, doi: 10.1007/s11042-016-3775-6.

[24] S. A. Hassan and M. Hussain, "Spatial domain lossless image data compression method," *2011 International Conference on Information and Communication Technologies*, pp. 1–4, 2011, doi: 10.1109/ICICT.2011.5983563.

[25] H. Zhang, F. Cricri, H. R. Tavakoli, N. Zou, E. Aksu, and M. M. Hannuksela, "Lossless image compression using a multi-scale progressive statistical model," in *Proceedings of the Asian Conference on Computer Vision*, 2020, doi: 10.48550/arXiv.2108.10551.

[26] G. Xin and P. Fan, "Soft compression for lossless image coding based on shape recognition," *Entropy*, vol. 23, no. 12, p. 1680, 2021, doi: 10.3390/e23121680.

[27] J. Abel, "Improvements to the burrows-wheeler compression algorithm: After bwt stages," *ACM Trans. Computer Systems*, 2003.

[28] M. A. Ali, A. Khan, M. Y. Javed, and A. Khanum, "Lossless image compression using kernel based global structure transform (gst)," in *2010 6th International Conference on Emerging Technologies (ICET)*, pp. 170–174, 2010, doi: 10.1109/ICET.2010.5638494.

[29] A. Khan, A. Khan, M. Khan, and M. Uzair, "Lossless image compression: application of bi-level burrows wheeler compression algorithm (bbwca) to 2-d data," *Multimedia Tools and Applications*, vol. 76, pp. 12391–12416, 2017, doi: 10.1007/s11042-016-3629-2.

[30] X. Liu, P. An, Y. Chen, and X. Huang, "An improved lossless image compression algorithm based on huffman coding," *Multimedia Tools and Applications*, vol. 81, no. 4, pp. 4781–4795, 2022, doi: 10.1007/s11042-021-11017-5.

[31] N. Tumanyan, M. Geyer, S. Bagon, and T. Dekel, "Plug-and-play diffusion features for text-driven image-to-image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1921-1930, 2023.

[32] T. Zhou, Q. Li, H. Lu, Q. Cheng, and X. Zhang, "GAN review: Models and medical image fusion applications," *Information Fusion*, vol. 91, pp. 134-148, 2023.

[33] R. Kumar *et al*., "An integration of blockchain and AI for secure data sharing and detection of CT images for the hospitals," *Computerized Medical Imaging and Graphics*, vol. 87, p. 101812, 2021.

[34] R. Mokady, A. Hertz, and A. H. Bermano, "Clipcap: Clip prefix for image captioning," *arXiv preprint arXiv:2111.09734*, 2021.

[35] M. Zhu *et al*., "Genimage: A million-scale benchmark for detecting ai-generated image," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[36] L. Guo *et al*., "Hydrogen safety: An obstacle that must be overcome on the road towards future hydrogen economy," *International Journal of Hydrogen Energy*, vol. 51, pp. 1055-1078, 2024.

[37] F. Xu, L. Huang, X. Gao, T. Yu, and L. Zhang, "Research on yolov3 model compression strategy for uav deployment," *Cognitive Robotics*, vol. 4, pp. 8-18, 2024.

[38] Z. R. Mohammed, Z. A. Yakoob, and M. Rajih, "Hybrid color image compression based on FMM and Huffman encoding techniques," in *AIP Conference Proceedings*, vol. 3009, no. 1, 2024.

[39] P. Samarathunga, V. Gowrisetty, T. Fernando, Y. Ganearachchi, and A. Fernando, "Region of interest scalable image compression using semantic communications," in *2024 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1-5, 2024.

[40] B. Chidirala and B. Acharya, "Run length encoding based reversible data hiding scheme in encrypted images," *Journal of Electronic Imaging*, vol. 33, no. 1, pp. 013018-013018, 2024.

[41] V. I. Ungureanu, P. Negirla, and A. Korodi, "Image-Compression Techniques: Classical and "Region-of-Interest-Based" Approaches Presented in Recent Papers," *Sensors*, vol. 24, no. 3, p. 791, 2024.

[42] A. Mefoued, N. Kouadria, S. Harize, and N. Doghmane, "Improved discrete Tchebichef transform approximations for efficient image compression," *Journal of Real-Time Image Processing*, vol. 21, no. 1, p. 12, 2024.

[43] S. Subramani, G. Thirugnanam, N. Prabakaran, and J. B. Fernandes, "Robust medical image watermarking technique using integer wavelet transform and shearlet transform with BSVD," *International Journal of Advanced Intelligence Paradigms*, vol. 27, no. 1, pp. 72-81, 2024.

[44] Q. Sima, H. Feng, and B. Hu, "Latitude-Adaptive Integer Bit Allocation for Quantization of Omnidirectional Images," *Applied Sciences*, vol. 14, no. 5, p. 1861, 2024.

[45] Y. Xia *et al*., "Multi-scale architectures matter: Examining the adversarial robustness of flow-based lossless compression," *Pattern Recognition*, vol. 149, p. 110242, 2024.

[46] Y. Tian *et al*., "Intelligent reconstruction algorithm of hydrogen-fueled scramjet combustor flow based on knowledge distillation model compression," *International Journal of Hydrogen Energy*, vol. 49, pp. 1278-1291, 2024.

[47] P. A. Hsieh and J. L. Wu, "A review of the asymmetric numeral system and its applications to digital images," *Entropy*, vol. 24, no. 3, p. 375, 2022.

[48] C. Su *et al*., "Analysis of pre-processing methods for lossless compression of multi component medical images based on latent variable models," in *Laser Science*, pp. JW5B-53, 2022.

[49] C. B. Pande, S. A. Kadam, R. Jayaraman, S. Gorantiwar, and M. Shinde, "Prediction of soil chemical properties using multispectral satellite images and wavelet transforms methods," *Journal of the Saudi Society of Agricultural Sciences*, vol. 21, no. 1, pp. 21-28, 2022.

[50] A. Rajput, J. Li, F. Akhtar, Z. Hussain Khand, J. C. Hung, Y. Pei, and A. Börner, "A content awareness module for predictive lossless image compression to achieve high throughput data sharing over the network storage," *International Journal of Distributed Sensor Networks*, vol. 18, no. 3, 2022.

[51] C. Gnanavel, A. Johny Renoald, S. Saravanan, K. Vanchinathan, and P. Sathishkhanna, "An Experimental Investigation of Fuzzy-Based Voltage-Lift Multilevel Inverter Using Solar Photovoltaic Application," *Smart Grids and Green Energy Systems*, pp. 59-74, 2022.

[52] L. Zhang, Y. Wang, and D. Zhang, "Research on multiple-image encryption mechanism based on Radon transform and ghost imaging," *Optics communications*, vol. 504, p. 127494, 2022.

[53] R. Li *et al*., "A Real-Time Incremental Video Mosaic Framework for UAV Remote Sensing," *Remote Sensing*, vol. 15, no. 8, p. 2127, 2023.

[54] T. Wang, H. Wang, K. Wang, and Z. Yang, "Research on Image Mosaic Method Based on Fracture Edge Contour of Bone Tag," *Applied Sciences*, vol. 13, no. 2, p. 756, 2023.

[55] R. K. Kanna, A. Ambikapathy, M. R. Al-Hameed, V. V. Reddy, and N. Singh, "Modern 3d Compression Application in Medical Imaging Approach," in *2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, vol. 10, pp. 1492-1496, 2023.

[56] T. H. Park and S. D'Amico, "Adaptive Neural-Network-Based Unscented Kalman Filter for Robust Pose Tracking of Noncooperative Spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 9, pp. 1671-1688, 2023.

[57] M. Waleed, T. W. Um, A. Khan, and A. Khan, "On the Utilization of Reversible Colour Transforms for Lossless 2-D Data Compression," *Applied Sciences*, vol. 10, no. 3, p. 937, 2020.

[58] J. Uthayakumar, M. Elhoseny, and K. Shankar, "Highly reliable and low-complexity image compression scheme using neighborhood correlation sequence algorithm in WSN," *IEEE Transactions on Reliability*, vol. 69, no. 4, pp. 1398-1423, 2020.

[59] M. Alqerom. *An intelligent system for the classification and selection of novel and efficient lossless image compression algorithms*. University of Salford (United Kingdom), 2020.

[60] A. Shalayiding, Z. Arnavut, B. Koc, and H. Kocak, "Burrows-Wheeler Transformation for Medical Image Compression," in *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0723-0727, 2020.

[61] https://links.uwaterloo.ca/Repository.html.

[62] https://www.kaggle.com/datasets/sherylmehta/kodak-dataset.

[63] http://documents.epfl.ch/groups/g/gr/gr-eb-unit/www/IQA/Original.zip.