A Performance Study of Different Approaches of Digital Image Compression Techniques

Wahida Ali Mansouri
Department of Computer Science and Information Technology, Faculty of Sciences and Arts, Turaif, Northern Border University, Saudi Arabia | LETI laboratory, University of Sfax, Tunisia
wahida.smari@nbu.edu.sa (corresponding author)

Salwa Hamda Othman
Department of Computer Science and Information Technology, Faculty of Sciences and Arts, Turaif, Northern Border University, Arar 91431, Kingdom of Saudi Arabia | LETI laboratory, University of Sfax, Tunisia
salwa.othmen@nbu.edu.sa

Somia Asklany
Department of Computer Science and Information Technology, Faculty of Sciences and Arts, Northern Border University, Saudi Arabia | Modern Academy for Science and Technology, Egypt
somia.asklany@nbu.edu.sa

Doaa Mohamed Elmorsi
Department of Computer Science and Information Technology, Faculty of Sciences and Arts, Northern Border University, Saudi Arabia
doaa.elmourssi@nbu.edu.sa

Received: 17 May 2024 | Revised: 30 May 2024 | Accepted: 7 June 2024

 DOI: https://doi.org/10.48084/etasr.7862

ABSTRACT

Today, managing a large amount of information becomes increasingly crucial. Efficient storage and retrieval of digital data are essential for their effective utilization. This study investigates the efficacy of Spatial Domain Image Compression Techniques, which directly manipulate the original image to reduce its size by leveraging pixel spatial relationships. These techniques segment the image into blocks and process each block independently. Evaluation entails measuring perceptual quality through metrics, such as PSNR, WPSNR, NMSE, and SSIM applied to the compressed image. Experimental results provide a comparative analysis of the performance of these techniques.

Keywords-image compression; WPSNR; EPTC; MPBTC; AMPTC

I. INTRODUCTION

Today, a vast volume of digital data, much of which comprise graphical or pictorial content, is stored, processed, and transmitted,. Consequently, the demands on storage and communication systems are substantial. The storage and transmission of uncompressed multimedia data require significant storage capacity and transmission bandwidth. Although there have been rapid advances in mass-storage density, processor speeds, and digital communication system performance, the need for data storage capacity and transmission bandwidth consistently exceeds the capabilities of current technologies [1-17]. Data compression provides methods for representing data more compactly, providing better storage utilization and faster transmission. This approach swaps computations for storage or bandwidth, as the benefits of data compression come at the expense of numerical calculations. Data are processed so that their representation requires fewer bits before they are stored or sent [2]. Visual quality is achieved through various pre-processing and post-processing techniques, which increase complexity and cost [18]. This study offers insights into improving the efficiency and effectiveness of image compression methods by evaluating spatial domain compression techniques using metrics, such as NMSE, PSNR, WPSNR, and SSIM. Furthermore, a comparison of these metrics offers a comprehensive understanding of the performance and potential enhancements in image compression, thus contributing to the further development of image processing technologies [13].

II. IMAGE COMPRESSION APPROACHES

In [7], Block Truncation Coding (BTC) was examined to compress images, comparing both the original BTC and the Absolute Moment BTC (AMBTC) [9]. BTC involves dividing the original image into *n*×*n* sub-blocks, thus reducing the number of grey levels within each block. In [10], an Efficient BTC approach was introduced.

An image consists of both essential information and repetitive elements. Essential information is the part of the data that is crucial for understanding their meaning or purpose, while redundancy refers to the parts that can be temporarily removed. Data compression is a method used to minimize redundancy. During decompression, the removed redundancy is reintroduced to restore the data to their original form. Most photos share the trait of having connected nearby pixels that contain redundant information. In general, there are two kinds of redundancy [3]:

- Spatial redundancy: This refers to the correlation between neighboring pixels, which represents the only form of redundancy present in grayscale images.
- Spectral redundancy: This pertains to the correlation between distinct color planes or spectral bands. This type of redundancy is observed in color images.

Image compression research aims to minimize the number of bits required to represent a picture, and there are two main types of compression methods [4]:

- Lossless compression: information is compressed but not lost when the data are reconfigured (uncompressed).
- Lossy compression: this type of compression loses detail pixel by pixel and approximates the image to make it look identical to the original.

Specifically, image compression aims to:

- Require less memory and disk space, lowering storage requirements.
- Make possible faster transmission. Compared to uncompressed images, compressed images can be transferred across networks faster.
- Use less bandwidth. When transferring images or videos, image compression uses less bandwidth and eases the strain on networks.
- Perform faster processing: The smaller data size of compressed images allows for faster processing and rendering.

III. IMAGE COMPRESSION USING SPATIAL DOMAIN TECHNIQUES

Spatial domain techniques work directly on the original image without any transformation. They try to compress the image by taking advantage of spatial correlations between the pixels in the image. This study introduces three lossy image compression techniques: Moment Preserving Truncation Coding (MPBTC), Absolute Moment Block Truncation Coding (AMBTC), and Efficient Block Truncation Coding (EBTC).

A. Image Compression Using Momment Preserving Truncation Coding (MPBTC)

MPBTC is a well-known lossy compression technique that was suggested for grayscale images [5]. This approach is also known as BTC because it keeps the first and second minutes of every picture block. The fundamental MPBTC algorithm quantizes a local area of the image using a Q-level quantizer, making it a lossy fixed-length compression technique [6-7]. The quantizer levels are selected to preserve some moments of a particular region in the image in the quantized output. To retain a grayscale image sample, the mean and sample standard deviation are the basic goals of MPBTC. The steps in the MPBTC algorithm are:

- Divide the provided image into non-overlapping rectangular sections. To keep things simple, the blocks are *n*×*n* square regions.
- Compute the mean ($\bar{x}$) using (1) and the standard deviation (σ) using (2) for each block.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (1)$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}_i)^2} \quad (2)$$

where x_i denotes the pixel value of the i^{th} pixel within the image block and n signifies the total number of pixels within this block.

- A two-level bit plane is produced by comparing each pixel value (x_i) with the threshold, denoted by $\bar{x}$. The pixels are also represented by a binary block, denoted as B. If a pixel's gray level is greater than or equal to $\bar{x}$, it is represented by 1, while if it's less than $\bar{x}$, it is represented by 0, as shown in (3).

$$B = \begin{cases} 1 & x_i \geq \bar{x} \\ 0 & x_i < \bar{x} \end{cases} \quad (3)$$

- For every block, the block matrix (B), mean ($\bar{x}$), and σ are sent.
- Reconstruct the image blocks in the bit plane with H and the zeros with L, which are given by (4) and (5):

$$H = \bar{x} + \sigma\sqrt{\frac{p}{q}} \quad (4)$$

$$L = \bar{x} - \sigma\sqrt{\frac{p}{q}} \quad (5)$$

where p and q are the numbers of zeros and ones in the compressed bit plane. Eight bits per pixel of the image are compressed to two bits per pixel (bpp). This is because the bit map in MPBTC requires 16 bits, the block mean requires 8 bits, and the block standard deviation requires 8 bits. Consequently, 32 bits are needed for the complete 4×4 block, meaning the data rate is 2 bpp. The benefit of MPBTC is that it can produce images with excellent quality while requiring minimum processing time.

B. Image Compression using Absolute Moment Block Truncation Coding (AMBTC)

In [8], a simplified version of MPBTC, called Absolute Moment BTC (AMBTC) and designed to efficiently maintain both the higher and lower means within a block, was introduced. AMBTC essentially follows the same framework as MPBTC, with the key difference lying in how it computes these two quantization levels. This study will not delve into the intricate specifics of the encoding and decoding processes [9]. The steps in the AMBTC algorithm are:

- An image is split into distinct, non-overlapping blocks, each with dimensions such as 4×4 or 8×8 among others.
- Calculate the mean gray level within the 4×4 block using:

$$\bar{x} = \frac{1}{16}\sum_{i=1}^{n16} x_i \quad (6)$$

where x_i represents the i^{th} pixel in the block.

- The pixels within the image block are categorized into two groups based on their gray levels. The upper range consists of gray levels that exceed the block's average gray level ($\bar{x}$), while the lower range includes the remaining values. The means of the upper range (X_H) are calculated using (7), and the lower range X_L means are calculated utilizing (8):

$$X_H = \frac{1}{K} \sum_{x_i \geq \bar{x}}^{n} x_i \quad (7)$$

$$X_L = \frac{1}{16-K} \sum_{x_i \geq \bar{x}}^{n} x_i \quad (8)$$

where K is the number of pixels whose gray level is greater than $\bar{x}$.

- A binary block, labeled B, is employed to represent pixels, where 1 signifies a pixel with a gray level greater than or equal to $\bar{x}$, and 0 indicates a pixel with a gray level lower than $\bar{x}$. The encoder records X_L and X_H. Allocating 8 bits to represent each X_H and X_L, the total bit count for a block becomes 8+8+16 = 32 bits. Consequently, the bit rate for the AMBTC algorithm is 2 bpp.
- In the decoder, an image block is reconstructed by substituting ones with X_H and zeros with X_L in the AMBTC scheme. Although both AMBTC and MPBTC require 16 bits to encode the bit plane, AMBTC requires less computational effort than MPBTC.

C. Image Compression using Efficient Block Truncation Coding (EBTC)

Using the mean value to classify the pixels in each block is the simplest way but may not be optimal. Between the maximal and minimal pixel values in one block, there is one optimal threshold that optimizes Mean Squared Error (MSE) for the compressed image block. To obtain the optimal threshold value, all possible values should be examined to find the best. Obviously, Optimal grouping BTC (OBTC) [11] is both inefficient and time-consuming. In [10], Efficient BTC (EBTC) was proposed to reduce the computational cost of OBTC, achieving a substantial decrease in execution time. In the EBTC method, the feature of inter-pixel correlation is exploited to further reduce the requirement of bits to store a block. This technique utilizes inter-pixel redundancy to further reduce the bit rate required for encoding images. It takes advantage of the commonality that nearby pixels often share similar intensity values. Initially, the image is segmented into small blocks, typically of size $n \times n$ pixels. These blocks are then divided into two groups: low- and high-detail blocks [12]. A block is classified as a high-detail block when there are noticeable variations in intensity values among its neighboring pixels. Conversely, if the intensity value differences are minimal, the block is labeled as a low-detail block. This classification process optimizes encoding efficiency by preserving crucial details in high-detail blocks while conserving bit-rate in less intricate regions of the image [13]. The steps in the EBTC algorithm are:

- An image is divided into non-overlapping blocks. Blocks can have dimensions such as 4×4 or 8×8 among others.
- Calculate the average gray level of the block using:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (9)$$

where x_i denotes the pixel value of the i^{th} pixel within the image block and n signifies the total number of pixels within that block.

- Categorize blocks using (10) and (11):

$$S = \sum_{i=1}^{k} abs\,(x_i - \bar{x}) \quad (10)$$

where S is the sum of the absolute difference between the individual components of the block, and the mean of the block is calculated using:

$$X = \begin{cases} B_{th} & S \geq T \\ B_l & S < T \end{cases} \quad (11)$$

where B_{th} is the high-detail block, B_l is the low-detail block, and $\bar{x}$ is the mean of the block. T is the threshold value.

- For low-detail blocks, only the mean value is stored instead of the two quantizing levels L and H. Therefore, for a low-detail block, it requires only 8 bits.
- For a high-detail block, generate the bit plane using (12) and store it.

$$X = \begin{cases} 1 & x_i \geq \bar{x} \\ 0 & x_i < \bar{x} \end{cases} \quad (12)$$

- Compute the two quantizing levels L and H utilizing (4) and (5).
- A high-detail block necessitates 32 bits for storage. The proposed method offers a notable reduction in the bit rate.

IV. PERFORMANCE EVALUATION

The effectiveness and performance of compression methods are evaluated through various performance criteria [16].

A. Normalized Mean-Square Error (NMSE)

The Mean-Square Error (MSE) describes the total squared difference between the compressed and the original images and provides insight into the disparity between them. However, the NMSE normalizes the MSE by dividing it by the squared range

of the original image. A lower NMSE value indicates a more efficient compression algorithm. NMSE is defined by:

$$NMSE = \sum_{i=1}^{M}\sum_{j=1}^{N}[y(i,j) - x(i,j)]^2) / \sum_{i=1}^{M}\sum_{j=1}^{N}(x^2(i,j) \quad (13)$$

where *M* and *N* represent the size of the image, *y* are the actual values, and *x* are the expected values.

B. Peak Signal to Noise Ratio (PSNR)

PSNR [14] is defined as the ratio of the maximum pixel intensity to the MSE, typically measured in dB. It serves as an indicator of image quality, where a higher PSNR value indicates better image quality. PSNR is an essential tool for evaluating the effectiveness of compression algorithms, as it quantifies the level of information loss in the reconstructed image compared to the original uncompressed image using:

$$PSNR = 10\, log\left(\frac{L^2}{MSE}\right) \quad (14)$$

where *L*, equaling 255, denotes the dynamic range of the pixel values.

C. Weighted PSNR (WPSNR)

WPSNR extends the traditional PSNR by incorporating a local activity factor, which is associated with the local variance. This metric considers the sensitivity of the human visual system by weighting each term of the PSNR calculation. It evaluates image quality while taking into account the surrounding pixels, thereby providing a more comprehensive measure that aligns with human perception.

$$WPSNR = 10\ log\left(\frac{L^2}{\|(y-x).NVF\|^2}\right) \quad (15)$$

where:

$$NVF = {}^{1}/_{(1 + \theta\ \sigma_x^2(i,j))}$$

D. Structure Similarity Index(SSIM)

SSIM measures the degree of similarity between the original and compressed images. SSIM is good at predicting the perceived quality of the compressed image and focuses on observable structures in the image. Its output has a value between -1 and 1, where 1 denotes identical photos. Equations (16) and (17) are used to calculate SSIM for two windows, labeled as *x* and *y*, both of size *N*×*N*:

$$SSIM(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \quad (16)$$

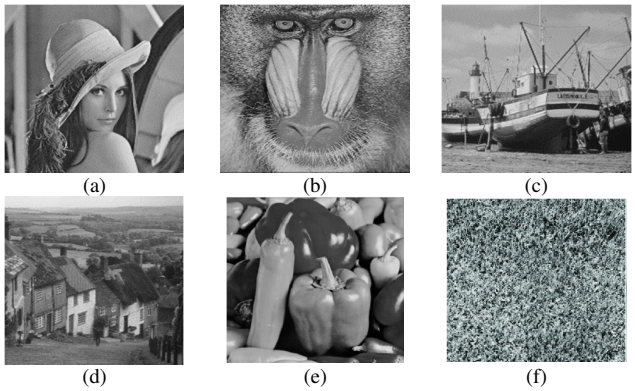
This leads to a particular form of the SSIM index:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (17)$$

where $C_1 = (K_1\, L)^2$, $K_1 \ll 1$ and $C_2 = (K_2\, L)^2$, $K_2 \ll 1$.

V. DATASET

This study deployed a set of six 8-bit grayscale digital images of various types from [19], each with dimensions of 512×512 pixels, as evidenced in Figure 1. The selected images are Lena, Baboon, Gledhill, Boat, Peppers, and Grass. Additionally, a texture image representing grass was incorporated. These grayscale images encompass a range of texture complexities, including low, medium, and high textures, distributed across different areas within each image. These grayscale images contain low, medium, and high textures in different areas. The characteristics of the test images were evaluated in the spatial domain using Spatial Frequency Measures (SFM) [15]. Table I portrays the SFM values calculated for this set of images.



(a) (b) (c) (d) (e) (f)

Fig. 1. Image dataset (512×512 px): (a) Lena, (b) Baboon, (c) Boat, (d) Goldhill, (e) Peppers, and (f) Grass.

TABLE I. SPATIAL FREQUENCY MEASURES OF IMAGES

MPBTC				
	NMSE	**PSNR**	**WPSNR**	**SSIM**
2:1	0.001328138	35.59698	43.29572	0.97532
4:1	0.00396	30.38676	37.65284	0.92268
6:1	0.0068	27.69868	34.00476	0.8601
8:1	0.01696	23.54918	28.02504	0.70518
AMBTC				
	NMSE	**PSNR**	**WPSNR**	**SSIM**
2:1	0.000920872	37.15896	44.75648	0.98094
4:1	0.0035	30.87636	38.17234	0.9261
6:1	0.00622	28.09028	34.49294	0.86624
8:1	0.01452	24.38188	29.14146	0.73958
EBTC				
	NMSE	**PSNR**	**WPSNR**	**SSIM**
2:1	0.001414534	35.3292	42.24032	0.96506
4:1	0.00626	28.94376	36.5859	0.85896
6:1	0.0068	27.6981	34.00302	0.85998
8:1	0.00696	27.61934	33.74296	0.83794
10:1	0.00736	27.44822	33.35884	0.80952
15:1	0.00906	26.71696	32.31852	0.73752
20:1	0.01108	25.91434	31.45912	0.68372

The Baboon test image contains abundant details, leading to a large SFM. A large SFM indicates that the image contains components in the high-frequency area. This suggests that such images have low redundancy, making them difficult to compress. The Lena test image has fewer details, resulting in a small SFM. A small SFM indicates that the image contains components in the low-frequency area. This implies that such images exhibit high redundancy, making them easier to compress.

VI. RESULTS AND DISCUSSION

The quality assessment of compressed images using spatial domain techniques involves calculating and recording metrics, such as NMSE, PSNR, WPSNR, and SSIM. Table II displays the results for analysis and comparison. The test images were compressed with different spatial domain techniques at different compression ratios. The values obtained from compressing different images using each technique were then averaged. The resulting NMSE, PSNR, WPSNR, and SSIM values are plotted against the different compression ratios as observed in Figures 2 to 5.

TABLE II. AVERAGE RESULTS

	Lena	Baboon	Goldhill	Boat	Peppers	Grass
SFM	14.0421	36.5146	16.1666	17.8565	15.8446	20.547

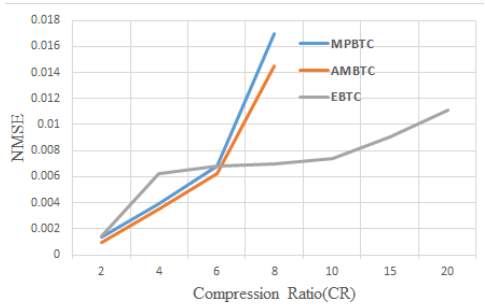


Fig. 2. NMSE against compression ratio.

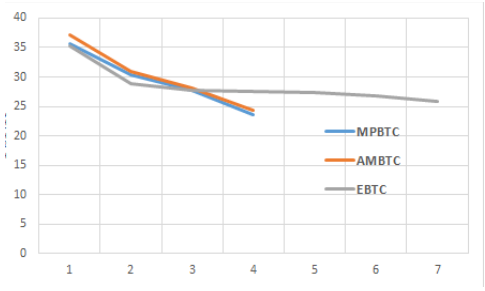


Fig. 3. PSRN against compression ratio.

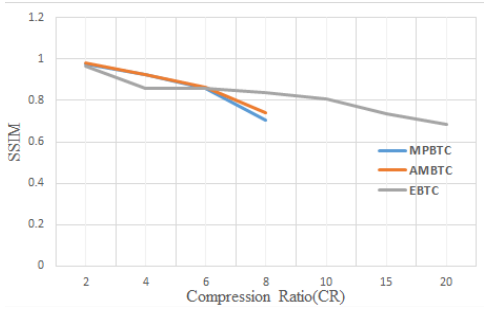


Fig. 4. SSIM against compression ratio.

From the tables and the plot curves, it can be concluded that the EBTC algorithm has the best performance only at high compression ratios, as it gives lower NMSE and higher PSNR, WPSNR, and SSIM values than the other techniques.

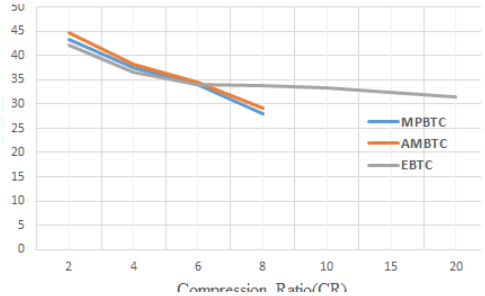


Fig. 5. WPSNR against compression ratio.

Figure 6 depicts the Lena image before and after applying three different spatial domain techniques for CR =8. It should be noted that MPBTC and AMBTC cannot be used to obtain compression ratios higher than 8 because they rely on the choice of block size. EBTC can be applied utilizing different threshold values. As the threshold value increases, performance is improved and data loss is reduced.

(a) (b) (c) (d)

Fig. 6. (a) Original image, (b) Compressed image using MPBTC at CR=8, (c) Compressed image using AMBTC at CR=8, (d) Compressed image using EBTC at CR=8.

VII. CONCLUSION

This study investigated the performance of spatial domain image compression techniques to select the one that gives the highest compression ratio and the minimum data loss. Spatial domain techniques, such as MPBTC, AMBTC, and EBTC are the most amenable for use on low-power computer systems due to their low computational complexity in image decoding. The performance of these techniques was evaluated deploying NMSE, PSNR, WPSNR, and SSIM. The experimental results showed that the EBTC algorithm had the best performance only at high compression ratios since it achieved lower values of NMSE and higher values of PSNR, WPSNR, and SSIM than the others. Some limitations of the present study may include constraints in dataset diversity, which can be addressed in future research endeavors. Future research in image compression could also involve exploring more advanced compression algorithms, investigating the integration of artificial intelligence techniques into compression processes, exploring the impact of compression on different image types (e.g., medical images) and novel approaches to improve compression efficiency while preserving image quality.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the approval and support of this research study by grant no. CSAT-2023-12-

2267 from the Deanship of Scientific Research at Northern Border University, Arar, K.S.A.

REFERENCES

[1] T. Acharya and A. K. Ray, *Image Processing: Principles and Applications*. Hoboken, NJ, USA: John Wiley & Sons, 2005.

[2] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Pearson Education, 2018.

[3] D. Salomon, *Data Compression*, 2nd ed. Heidelberg, Germany: Springer-Verlag, 2005.

[4] R. Ghodhbani, T. Saidani, L. Horrigue, A. M. Algarni, and M. Alshammari, "An FPGA Accelerator for Real Time Hyperspectral Images Compression based on JPEG2000 Standard," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13118–13123, Apr. 2024, https://doi.org/10.48084/etasr.6853.

[5] P. Fränti, O. Nevalainen, and T. Kaukoranta, "Compression of Digital Images by Block Truncation Coding: A Survey," *The Computer Journal*, vol. 37, no. 4, pp. 308–332, Jan. 1994, https://doi.org/10.1093/comjnl/37.4.308.

[6] D. Nayak, K. B. Ray, T. Kar, and C. Kwan, "A novel saliency based image compression algorithm using low complexity block truncation coding," *Multimedia Tools and Applications*, vol. 82, no. 30, pp. 47367–47385, Dec. 2023, https://doi.org/10.1007/s11042-023-15694-2.

[7] D. Mohammed and F. Abou-Chadi, "Image Compression Using Block Truncation Coding," *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT)*, Feb. 2011.

[8] M. Lema and O. Mitchell, "Absolute Moment Block Truncation Coding and Its Application to Color Images," *IEEE Transactions on Communications*, vol. 32, no. 10, pp. 1148–1157, Oct. 1984, https://doi.org/10.1109/TCOM.1984.1095973.

[9] S. S. R. Bhagya, G. N. Chandan, M. Girish, and K. Painthamil, "Image Compression using AMBTC," *International Journal of Electronics, Electrical and Computational System IJEECS*, vol. 5, no. 10, Oct. 2016.

[10] S. Karuppanagounder and V. Alagumalai, "Efficient Block Truncation Coding," *International Journal on Computer Science and Engineering*, vol. 2, no. 6, pp. 2163–2166, Sep. 2010.

[11] M. H. El Ayadi, M. M. Syiam, and A. A. Gamgoum, "A Comparative Study of Various Lossy Image Compression Techniques," *International Journal of Intelligent Computing and Information Sciences IJICIS*, vol. 6, no. 1, Jul. 2016.

[12] D. Nayak, K. B. Ray, T. Kar, and C. Kwan, "A novel saliency based image compression algorithm using low complexity block truncation coding," *Multimedia Tools and Applications*, vol. 82, no. 30, pp. 47367–47385, Dec. 2023, https://doi.org/10.1007/s11042-023-15694-2.

[13] G. Garg and R. Kumar, "Analysis of Different Image Compression Techniques: A Review," in *International Conference on Innovative Computing & Communication (ICICC) 2022*, Delhi, India, Feb. 2022, https://doi.org/10.2139/ssrn.4031725.

[14] N. Yamsang and S. Udomhunsakul, "Image Quality Scale (IQS) for Compressed Images Quality Measurement," in *Proceedings of the International MultiConference of Engineers and Computer Scientists IMECS 2009*, Hong Kong,China, Mar. 2009.

[15] D. Elmourssi, W. A. Mansouri, W. A. Elyass, S. H. Othman, and S. Asklany, "A Performance Study Of Two Jpeg Compression Approaches," *Journal of Intelligent Systems and Applied Data Science (JISADS)*, vol. 2, no. 1, pp. 20–28, Apr. 2024.

[16] M. Testolina and T. Ebrahimi, "Review of subjective quality assessment methodologies and standards for compressed images evaluation," in *Applications of Digital Image Processing XLIV*, San Diego, CA, USA, Aug. 2021, vol. 11842, pp. 302–315, https://doi.org/10.1117/12.2597813.

[17] L. Horrigue *et al.*, "Efficient Hardware Accelerator and Implementation of JPEG 2000 MQ Decoder Architecture," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13463–13469, Apr. 2024, https://doi.org/10.48084/etasr.7065.

[18] M. V. Daithankar and S. D. Ruikar, "Analysis of the Wavelet Domain Filtering Approach for Video Super-Resolution," *Engineering, Technology & Applied Science Research*, vol. 11, no. 4, pp. 7477–7482, Aug. 2021, https://doi.org/10.48084/etasr.4262.

[19] *The Waterloo fractal coding and analysis group dataset*. [Online]. Available: https://links.uwaterloo.ca/Repository.html.