1

Visual Perception-Inspired Image Transmission for Intelligent Mobile Devices Using FMQM

Liu. Zhihan, Student, Universitat Autònoma de Barcelona

***Abstract*—With the rapid popularization of intelligent mobile devices such as drones, smartphones, and dashcams, efficient image transmission has become a critical challenge. High-resolution images generated by these devices often face constraints of limited network bandwidth, storage capacity, and real-time processing requirements. This paper proposes an image compression algorithm that uses perceptual coding and imports a Flexible Modulus Quantization Method (FMQM)[1], inspired by the Five Modulus Method (FMM) [1]. The FMQM method simplifies the data by grouping values based on the modulus and it can flexibly configure the module, making the image data more suitable for Run-Length Encoding (RLE) while minimizing visual impact. Following FMQM quantity, Run-Length Encoding (RLE) is applied to exploit redundancies in low-frequency regions, to achieve visually lossless but compression with efficient computing. Finally, from the perspective of human vision, SSIM and other indicators are used to measure Image Quality Assessment (IQA) to ensure that the compressed image quality meets expectations [2,3].**

***Keywords*—Flexible Modulus Quantization Method (FMQM), Image compression, Perceptual coding, Real-time transmission, Run-Length Encoding (RLE).**

I. INTRODUCTION

The proliferation of intelligent mobile devices such as drones, smartphones, and dashcams has significantly increased the demand for real-time image transmission. These devices often get high-resolution images, which provide more visual detail, pose challenges related to storage capacity and network bandwidth. An efficient image compression method to meet constraints on transmission speed, storage, and computational resources while ensuring high-quality image reconstruction is urgently needed.

This paper introduces a flexible image compression algorithm tailored for such resource-constrained environments. The proposed base on perceptual coding, which leverages the characteristics of human visual perception to achieve visually lossless compression. By exploiting the human eye's varying sensitivity to luminance and chrominance, the algorithm selectively preserves critical visual details while compressing less perceptible channels. This ensures that essential image information remains intact while reducing redundancy in less sensitive channels.

At the core of this algorithm is the Flexible Modulus Quantization Method (FMQM) cooperating with Run-Length Encoding (RLE), which extends the traditional Five Modulus Method (FMM) by import dynamic modulus selection. Inspired by prior work highlighting the advantages of the YCbCr color space for concentrating energy into the luminance channel [4]. Unlike FMM, which applies a fixed modulus across all channels, FMQM assigns smaller modulus values to the luminance (Y) channel to retain crucial details and larger modulus values to the chrominance (Cb and Cr) channels to reduce data size. This flexibility allows FMQM to balance compression efficiency and visual quality, adapting to the specific requirements of different application scenarios.

Using Run-Length Encoding (RLE) after FMQM quantization. RLE effectively compresses the redundancy inherent in chrominance channels that regions of identical values are common. Finally, image quality is evaluated using image quality assessment methods such as SSIM and PSNR. This combination of FMQM and RLE not only achieves high compression ratios but also maintains low computational complexity, making it ideal for real-time transmission in intelligent mobile devices like drones or other mobile devices.

The data in this paper are from Rawzor - Lossless compression software for camera raw images (https://imagecompression.info/test_images/).

II. PREVIOUS WORK

In 2019, Nagendra Kumar Gupta and M. P. Parsai proposed an Improvised Five Modulus Method (IFFM) for image compression, which integrates FMM into the JPEG compression pipeline [1]. The algorithm converts each pixel value in an 8×8 block into a multiple of 5, reduces them to 6-bit values, and applies YCbCr transformation, DCT, quantization, zigzag scanning, and Huffman encoding. Experimental results show that IFFM achieves higher compression ratios compared to FJPEG while keeping acceptable visual quality.

In 2021, researchers proposed a learned image compression method using a gained variational auto-encoder (gained-VAE) in the YCrCb domain, which enables image-adaptive luma-chroma bit allocation during inference [4]. By optimizing Y PSNR at the expense of chroma PSNR, the method produces sharper images without introducing color artifacts, solving the typical balance between fidelity and sharpness seen in RGB-based models. Experimental results showed improved VMAF and Y PSNR compared to state-of-the-art methods,

[1]The term Flexible Modulus Quantization Method (FMQM) is an original term of this paper, developed to extend the Five Modulus Method (FMM) for the more flexible image compression. FMQM import dynamic modulus selection tailored to luminance and chrominance channels, enabling enhanced flexibility and efficiency in compression algorithms.

demonstrating the effectiveness of this approach in achieving high-fidelity compression.

In 2023, Xutan Peng et al. proposed an enhanced Run-Length Encoding (RLE) algorithm to address the issue of size inflation when sequences lack consecutive elements [5]. By leveraging combinatorics, their method quantifies RLE's compression potential for a given input distribution and selectively applies RLE only to suitable symbols. This method maintains the efficiency of traditional RLE while reducing size inflation, as validated through experiments on diverse real-world datasets.

A. Run-length encoding, RLE

Run-Length Encoding (RLE) is a simple compression method that works by replacing consecutive repeated values with a single value and its count. For example, the sequence AAABBBBCCDEEEE can be encoded as 3A4B2C1D4E [6]. This method is most effective when data has many repeated values in a row.

The main advantage of RLE is its low computational complexity. It processes the data in one pass, with a time complexity of O(n), where n is the length of the data. This makes it fast and suitable for systems with limited resources. However, if the data has few repetitions, the compression ratio will be low or might even increase the data size. Therefore, the FMQM method proposed in this paper makes up for the shortcomings of RLE.

B. YCbCr Transform

YCbCr is a color space often used in image compression to separate brightness (luminance) and color (chrominance) components. The Y, Cb, and Cr elements of color image are defined in YUV color coordinate, where Y is luminance and Cb, Cr are generally called the chrominance [1]. It makes images easier to compress because the human eye is more sensitive to brightness than color.

RGB to YCbCr conversion equation is given by:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

YCbCr to RGB conversion equation is given by:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.34413 & -0.71414 \\ 1 & 1.772 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix}$$

These transformations allow efficient compression by reducing data in the Cb and Cr channels (less noticeable to the eye), while keeping the Y channel detailed.

C. Image Quality Assessment, IQA

Quality is a very important parameter for all objects and their functionalities. In image-based object recognition, image quality is a prime criterion [7]. This paper mainly uses SSIM and PSNR methods.

The SSIM index method, a quality measurement metric, is calculated based on the computation of three major aspects termed as luminance, contrast and structural or correlation term. This index is a combination of multiplication of these three aspects [8].

Structural Similarity Index Method can be expressed through these three terms:

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

Here, l is the luminance (used to compare the brightness between two images), c is the contrast (used to differ the ranges between the brightest and darkest region of two images) and s is the structure (used to compare the local luminance pattern between two images to find the similarity and dissimilarity of the images) and α, β and γ are the positive constants [9].

PSNR is used to calculate the ratio between the maximum possible signal power and the power of the distorting noise which affects the quality of its representation [8].

PSNR is expressed as:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\mathrm{peakval}^2}{\mathrm{MSE}}\right)$$

Here, peakval (Peak Value) is the maximal in the image data. If it is an 8-bit unsigned integer data type, the peakval is 255 [10].

The effectiveness of the FMQM+RLE algorithm was evaluated using two standard metrics: (i)PSNR (Peak Signal-to-Noise Ratio): A PSNR value greater than 30 dB indicates good image quality, suitable for storage and transmission, with minimal perceptual loss. (ii)SSIM (Structural Similarity Index): The SSIM value ranges between 0 and 1, with higher values representing closer similarity to the original image. For the tested images, SSIM values are consistently between 0.95 and 0.99, indicating that the images are nearly lossless.

III. Proposed Method

The proposed method consists of two primary stages: encoding and decoding. Each stage involves several key steps to efficiently and flexible compress and decompress image data while maintaining visual quality.

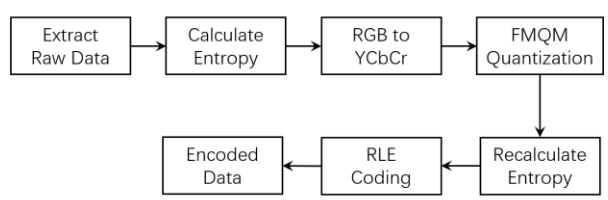

Fig. 1. Encoding pipeline

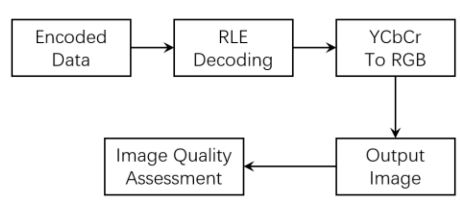

Fig. 2. Decoding pipeline

To demonstrate the practicality and efficiency of the proposed image compression algorithm, we consider a real-world scenario of a wildlife protection monitoring system. In this system, multiple drones are deployed to capture high-resolution images of wildlife and transmit them to a central monitoring station in real-time. This application faces several constraints and requirements:

(i) Network Bandwidth: In city, network bandwidth ranges from 10 Mbps to 50 Mbps, we assume a conservative bandwidth of 10 Mbps for this scenario.

(ii) Transmission Delay: Real-time monitoring requires that each image be transmitted within 2 seconds to avoid delays in observation.

(iii) Storage Space: The drones are equipped with limited local storage, necessitating a high compression rate to maximize the amount of data stored and extend operational time without deleting data.

(iv) Image Quality: To ensure the usability of the images for research purposes, the reconstructed image must maintain a Peak Signal-to-Noise Ratio (PSNR) of at least 30 dB and a Structural Similarity Index (SSIM) of 0.95 or higher.

To meet the above constraints, the proposed method must achieve the following:

(i) Use high-resolution images with a resolution of at least 1920×1080.

(ii) Achieve a compression ratio of at least 50%, reducing a 3 MB image to 1.5 MB or smaller.

(iii) Maintain an image transmission bitrate of no more than 2 Mbps.

(iv) Ensure that the transmission time does not exceed 2 seconds at a bandwidth of 10 Mbps.

(v) Preserve high image quality with a PSNR of at least 30 dB and an SSIM of 0.95 or greater.

A. Compression

Step 1. Extract Raw Data. Extracting raw pixel data from images in the PPM format. The PPM format stores image data along with a header containing metadata. The extraction process involves the following steps: (i)The image header is read to extract essential metadata, such as width, height and max color value of the image, etc. (ii)Records the offset where binary pixel data begins. (iii)Extracting starting from the binary offset and saving as RAW format for further processing.

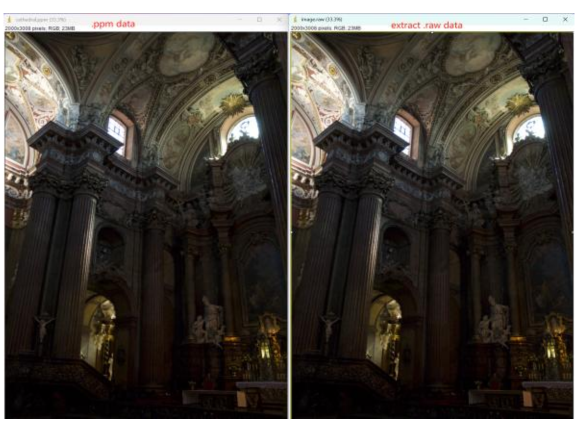

Fig. 3. Compare the extracted image with the original image

Step 2. Calculate entropy. Shannon entropy is computed to quantify the average information or uncertainty in the raw image data. This metric provides insight into the redundancy present in the data. The formula for entropy is:

$$H = -\sum_{i=1}^{n} P(x_i)\log_2 P(x_i)$$

Lower entropy means more redundancy, making the data easier to compress. This helps to check how well compression reduces data size.

Step 3. RGB to YCbCr. To align with human visual perception and enhance compression efficiency, the input image is transformed from the RGB color space to the YCbCr color space. The YCbCr model separates luminance (Y) from chrominance (Cb and Cr), allowing the algorithm to apply different compression levels to each component based on their perceptual importance. The transformation process involves the following steps: (i)The raw binary data is read and reshaped into an array of dimensions (height, width, 3), representing the RGB channels of the image. (ii)Using the PIL library, the RGB array is converted to the YCbCr color space. (iii)The transformed data is split into three independent channels: Y (Luminance) represents brightness and is crucial for preserving visual detail; Cb and Cr (Chrominance) represent color differences and are less sensitive to the human eye.

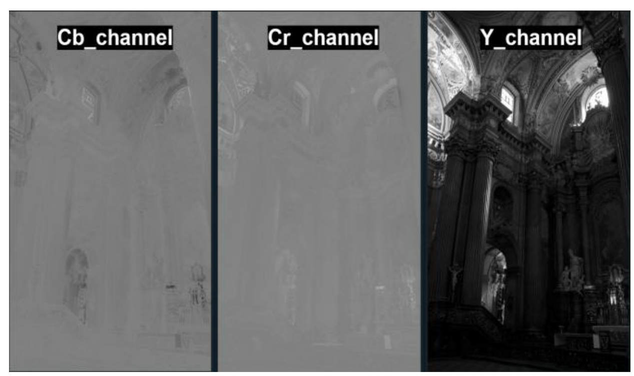

Fig. 4. Visualization of three channels after conversion to YCbCr

Step 4. FMQM Quantization. Flexible Modulus Quantization Method (FMQM) is used to reduce the data size of each channel while maintaining visual quality. This method leverages modulus-based quantization to group pixel values into discrete levels. By applying different modulus values to the luminance (Y) and chrominance (Cb and Cr) channels, FMQM balances compression efficiency with perceptual fidelity. The quantization process involves the following steps: (i)Each pixel value in a channel is divided by the modulus, rounded to the nearest integer, and then multiplied by the modulus to approximate the original value. Mathematically:

$$Q(x) = round(\frac{x}{m}) \cdot m$$

(ii)Set the modulus for each channel. Apply a smaller modulus (e.g., m=1) to the Luminance (Y) to retain critical visual details. Also applies a larger modulus (e.g., m=8) to the Chrominance (Cb, Cr) to compress less sensitive color information.

TABLE I. Comparison of 100 values before and after quantization

Data before quantification	Quantized data (modulus=10)
[107 109 111 112 112 113 111 110 104	[110 110 110 110 110 110 110 110 100
99 102 106 108 109 111 110 110 110	100 100 110 110 110 110 110 110 110
109 109 105 102 103 103 103 108 111	110 110 100 100 100 100 100 110 110
113 116 117 113 110 106 101 101 99	110 120 120 110 110 110 100 100 100
91 84 86 86 88 87 86 90 96 101	90 80 90 90 90 90 90 90 100 100
107 111 115 118 117 116 115 116 117	110 110 120 120 120 120 120 120 120
119 120 122 120 118 117 113 113 113	120 120 120 120 120 120 110 110 110
115 118 115 112 114 119 122 125 124	120 120 120 110 110 120 120 120 120
124 124 126 125 123 122 119 117 114	120 120 130 120 120 120 120 120 110
114 112 115 119 122 127 127 126 125	110 110 120 120 120 130 130 130 120
124 124 125 127 126 126 126 126 126]	120 120 120 130 130 130 130 130 130]

Step 5. Recalculate Entropy. After quantization, the entropy of each channel is recalculated to evaluate the compression potential.

Step 6. Run-Length Encoding. The quantized channels are compressed using RLE, which encodes sequences of repeated values as a pair: the value and its count. The RLE encoding process involves the following steps: (i)Flattens each channel into a 1D array. (ii)Iterates through the array, counting consecutive identical values.

Step 7. Encoded Data. Stores the compressed result as a binary file.

B. Decompression

Step 1. Encoded Data. Read the binary file saved after image compression.

Step 2. Run-Length Decoding. Restore RLE-encoded binary data to original data. The decompression process involves the following steps: (i)Input a list of pairs (value, count), where each pair represents the value and its frequency in the original data and the dimensions of the original data array. (ii)An empty array of the same shape as the original image is initialized. (iii)Fill the corresponding positions in the array to restore the original values.

Step 3. YCbCr to RGB. The image is reconstructed by converting the modified YCbCr data back to the RGB color space. The transformation process involves the following steps: (i)The quantized Y, Cb, and Cr channels are stacked along the third dimension to form a single array representing the YCbCr image. (ii)Using the PIL library, the stacked YCbCr array is converted into an RGB image. The PIL library handles the mathematical operations required to map YCbCr values to their corresponding RGB values based on standard transformation matrices.

Step 4. Output Image. Output the restored RGB image to a common image format, such as PNG.

Step 5. Image Quality Assessment. To evaluate the quality of the compressed images, two metrics were used: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). These metrics provide quantitative measures of the visual fidelity of the decompressed image compared to the original.

IV. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed image compression algorithm, experiments were conducted on high-resolution images using different modulus values for luminance (Y) and chrominance (Cb and Cr) channels. The results demonstrate the trade-off between compression rate and image quality, measured using PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index).

The two tests use the RGB-8 images provided by Rawzor. Two sets of modulus values were applied to compress and restore the image.

TABLE 2. Comparison of different moduli of two sets of images

cathedral.ppm provided by Rawzor		
Y=1 Cb=10 Cr=10	PSNR: 31.03 dB SSIM: 0.975	RAW: 17625 KB Recover: 8300 KB
Y=4 Cb=8 Cr=8	PSNR: 31.39 dB SSIM: 0.955	RAW: 17625 KB Recover: 6099 KB
flower_foveon.ppm provided by Rawzor		
Y=1 Cb=10 Cr=10	PSNR: 29.98 dB SSIM: 0.985	RAW: 10047 KB Recover: 3138 KB
Y=4 Cb=8 Cr=8	PSNR: 31.01 dB SSIM: 0.971	RAW: 10047 KB Recover: 1561 KB

From the data in the table, we can see that the flexible compression quantization method of FMQM+RLE is feasible, and both the compression rate and image quality evaluation have reached the expected standards.

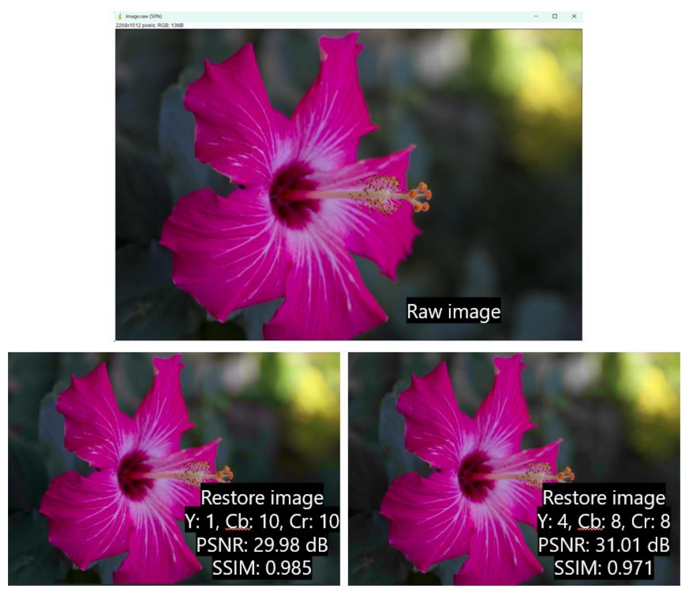

Fig. 5. Compression results of the picture flower_foveon

From the results of the image flower_foveon, we can see that for brighter images, humans do not easily notice the loss in the luminance layer. It is possible to use higher modulus values for all three channels without affecting visual quality, which helps to improve the compression rate.

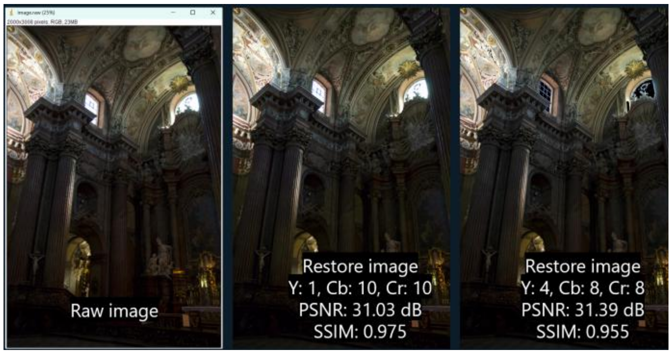

Fig. 6. Compression results of the picture cathedral

From the results of the image cathedral, we can see that for images with more dark areas and strong contrast, the modulus for the luminance channel should be as small as possible. A high modulus will have a noticeable visual impact on the quality of the brightness layer.

For the two test results, the proposed FMQM+RLE method achieved both the expected compression rate and image quality. It also met the real-time image transmission limits: for a 10 Mbps bandwidth and transmission time under 2 seconds. For example, using the flower_foveon image: Compressed size is 1481 KB; Estimated transmission time at 10 Mbps is 1.28 seconds. The formula is as follows:

$$Transmission\ Time = \frac{Compressed\ Size \times 8}{Bandwidth} = \frac{1481 \times 8 \times 1024}{10 \times 10^6} \approx 1.28 seconds$$

This shows that the method is suitable for real-time applications, such as drone image transmission, while meeting all constraints.

V. Conclusion

This paper presents a novel image compression method combining the Flexible Modulus Quantization Method (FMQM) and Run-Length Encoding (RLE). Through a series of experiments, the results show that the proposed FMQM+RLE method is effective in achieving high compression rates while maintaining visual quality, meeting the expectations for image transmission in resource-constrained environments.

The method uses a perceptual coding approach that aligns with human visual sensitivity. It applies lower modulus values to the luminance (Y) channel to preserve critical details and higher modulus values to the chrominance (Cb and Cr) channels to achieve efficient compression. The experimental results validate the feasibility of this approach. Using RGB-8 images provided by Rawzor, the compression rate exceeded 50% in all cases, and both PSNR and SSIM metrics confirmed high-quality reconstruction. For example: (i)With the flower_foveon image, the compressed size was 1481 KB, achieving an 85% compression rate with a PSNR of 31.16 dB and an SSIM of 0.985. The estimated transmission time at a 10 Mbps bandwidth was approximately 1.28 seconds. (ii)For the cathedral image, a compression rate of 55% was achieved without affecting visual perception with a PSNR of 30.03 dB and an SSIM of 0.975.

The study further highlights that for brighter images (e.g., flower_foveon), higher modulus values can be used across all channels without significant visual quality loss. Conversely, for images with dark areas and strong contrast (e.g., cathedral), a smaller modulus for the luminance channel is necessary to maintain visual fidelity.

In addition, the proposed method met the real-time transmission constraints of a 10 Mbps bandwidth and a maximum transmission time of 2 seconds. This demonstrates its suitability for real-time applications, such as drone-based image transmission, while satisfying all quality and efficiency requirements. These results confirm that the FMQM+RLE method is a flexible and practical solution for image compression in resource-limited scenarios.

VI. Future Work

In the future, this work can be extended in several ways to further improve the algorithm and explore its applications. For example:

(i)Dynamic modulus optimization: Currently, the modulus values for luminance and chrominance channels are fixed. Future work could develop an adaptive system that dynamically adjusts values based on the image content to achieve a better balance between compression and quality.

(ii)Extension to video compression: This method is currently applied to images, but it could be extended to video compression in the future. Using FMQM and RLE on video frames may achieve high efficiency while keeping video quality.

(iii)Comparison with other standards: Additional experiments could compare the performance of FMQM with industry standards such as JPEG or HEVC to identify specific strengths and weaknesses in different use cases.

(iv)Deep Learning Integration: Future work could join deep learning to improve image quality assessment, such as using neural networks to predict perceptual quality metrics more accurately.

Acknowledgment

Thanks to Rawzor - Lossless compression software for camera raw images for providing the dataset used in this research (https://imagecompression.info/test_images/). Their high-quality test images were crucial for evaluating the proposed algorithm.

Thanks also to the contributions of previous studies in the fields of image compression, perceptual coding, and quantization, which laid the foundation for this work. These prior achievements inspired the development and innovation of the proposed method.

References

[1] *N. K. Gupta and M. P. Parsai, "Improvised method of five modulus method embedded JPEG image compression with algebraic operation," 2019 2nd International Conference on Intelligent Computing,*

Instrumentation and Control Technologies (ICICICT), Kannur, India, 2019, pp. 1338-1342, doi: 10.1109/ICICICT46008.2019.8993154.

[2] *Sara, U. , Akter, M. and Uddin, M. (2019) Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. Journal of Computer and Communications, 7, 8-18. doi: 10.4236/jcc.2019.73002.*

[3] *J. C. Mier, E. Huang, H. Talebi, F. Yang and P. Milanfar, "Deep Perceptual Image Quality Assessment for Compression," 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, 2021, pp. 1484-1488, doi: 10.1109/ICIP42928.2021.9506217.*

[4] *O. U. Ulas and A. M. Tekalp, "Flexible luma-chroma bit allocation in learned image compression for high-fidelity sharper images," 2022 Picture Coding Symposium (PCS), San Jose, CA, USA, 2022, pp. 31-35, doi: 10.1109/PCS56426.2022.10017994.*

[5] *X. Peng, "A Hybrid Run-Length Encoding Approach to Mitigate Size Inflation," arXiv preprint, arXiv:2312.17024, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2312.17024.*

[6] *Solomon W. Golomb, "Run-length encodings (Corresp.)," IEEE Transactions on Information Theory, vol. 12, pp. 399–401, 1966.*

[7] *U. Sara, M. Akter, and M. S. Uddin, "Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study," Journal of Computer and Communications, vol. 7, no. 3, pp. 8-18, 2019. DOI: 10.4236/jcc.2019.73002.*

[8] *Brooks, A.C., et al. (2008) Structural Similarity Quality Metrics in a Coding Context: Exploring the Space of Realistic Distortions. IEEE Transactions on Image Processing, 17, 1261-1273. https://doi.org/10.1109/TIP.2008.926161*

[9] *Kumar, R. and Moyal, V. (2013) Visual Image Quality Assessment Technique Using FSIM. International Journal of Computer Applications Technology and Research, 2, 250-254. https://doi.org/10.7753/IJCATR0203.1008*

[10] *Deshpande, R.G., Ragha, L.L. and Sharma, S.K. (2018) Video Quality Assessment through PSNR Estimation for Different Compression Standards. Indonesian Journal of Electrical Engineering and Computer Science, 11, 918-924.*