International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 9, Issue 05, ISSN No. 2455-2143, Pages 144-150
Published Online September 2024 in IJEAST (http://www.ijeast.com)

COMPRESSION OF HYPERSPECTRAL IMAGE USING JPEG COMPRESSION ALGORITHM

Sujendra G Bharadwaj, Shruthi B
Department of ECE
SJB Institute of Technology,
Bengaluru - 560060, Karnataka, India

Abstract— **This paper delves into the considerable challenges of working with hyperspectral images, which are notably large and multidimensional, with file sizes often surpassing hundreds of megabytes. Hyperspectral imaging captures light across a continuous range of wavelengths, providing detailed spectral information for each pixel. This rich dataset is invaluable for applications such as environmental monitoring, precision agriculture, mineral exploration, and medical diagnostics, where accurate spectral data aids in identifying materials and detecting subtle variations. However, the immense data volume not only strains storage and transmission resources but also requires efficient processing and analysis techniques to handle the high-dimensional data without compromising quality. Additionally, compression methods are essential to manage storage constraints and improve real-time usability, but they must balance data reduction with the preservation of spectral integrity for effective analysis and application.**

Keywords— **JPEG Compression, Discrete Cosine Transform, Quantization, Image Decompression**

I. INTRODUCTION

Hyperspectral images capture a continuous spectrum for each pixel, enabling detailed spectral analysis through hyperspectral image processing, also known as spectral imaging. Unlike the human eye, which perceives visible light in three bands (red, green, and blue), hyperspectral imaging divides the spectrum into many bands, enhancing data retrieval. This imaging technique is significant in remote sensing for storing detailed information about materials and has applications in target detection, classification, anomaly detection, and spectral unmixing. Sensors collect data across contiguous wavelength bands from 400 to 2500 nm, with fixed spectral resolution and equal pixel counts per band. Each pixel's spatial resolution defines the area it represents, forming a data cube of reflectance values across various wavelengths. Hyperspectral imaging is utilized in military operations to monitor troop movements, in agriculture for quality assessment and disease detection, and in remote sensing for mineral classification and tracking natural disasters. Image compression plays a crucial role in efficiently storing and transmitting image data by minimizing file sizes while preserving essential visual quality. It can be divided into three main categories:

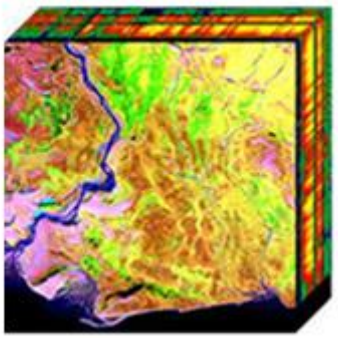


Fig 1.1: Hyperspectral Image

1. Lossless Compression: This method allows for the exact reconstruction of the original image, ensuring that no data is lost during the compression process. While it maintains high image quality, the compression ratios are typically lower compared to lossy methods. Lossless techniques are particularly important in applications requiring precise image fidelity, such as medical imaging and archival storage [20]
2. Lossy Compression: This technique reduces file sizes by removing some image data, resulting in a degree of quality loss. The extent of data loss can vary, and while the reduction in size is significant, it often compromises the image quality compared to lossless methods. Lossy compression is commonly used in multimedia formats, including MP3 for audio and JPEG for images.
3. Near Lossless Compression: This approach offers a middle ground between lossy and lossless compression, allowing for minor distortions that fall within acceptable limits. Near lossless compression achieves better compression performance than lossless techniques while maintaining higher quality than lossy methods, making it suitable for various applications where some level of distortion is permissible.[18]

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 9, Issue 05, ISSN No. 2455-2143, Pages 144-150
Published Online September 2024 in IJEAST (http://www.ijeast.com)

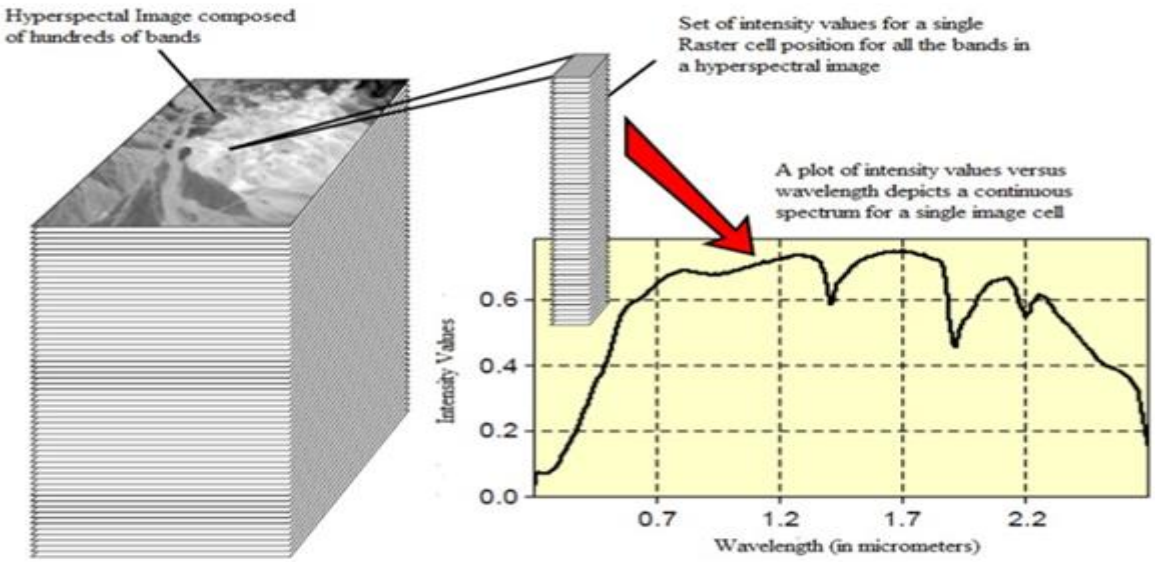


Fig 1.2 Continuous spectrum of the hyperspectral image cell

II. PROPOSED METHOD

The **JPEG algorithm** is a widely adopted method for lossy compression of digital images. This technique significantly reduces file sizes while maintaining acceptable image quality, enabling users to store vast quantities of images that were previously impractical due to storage limitations. By minimizing the size of image files, JPEG compression not only enhances storage efficiency but also reduces the system's processing load, leading to improved overall performance. The JPEG encoder operates through three primary blocks:

1. **Forward Discrete Cosine Transform (FDCT):** This step transforms the spatial representation of the image into the frequency domain, separating important visual information from less significant data.
2. **Quantization**: This process reduces the precision of the DCT coefficients, effectively discarding less critical information while preserving the essential features that the human eye can perceive.
3. **Entropy Encoding:** Utilizing techniques such as Huffman coding, this final block further compresses the quantized coefficients by assigning shorter codes to more frequent values, maximizing storage efficiency. Together, these components allow JPEG compression to achieve a balance between image quality and file size, making it an essential technique in digital imaging applications.

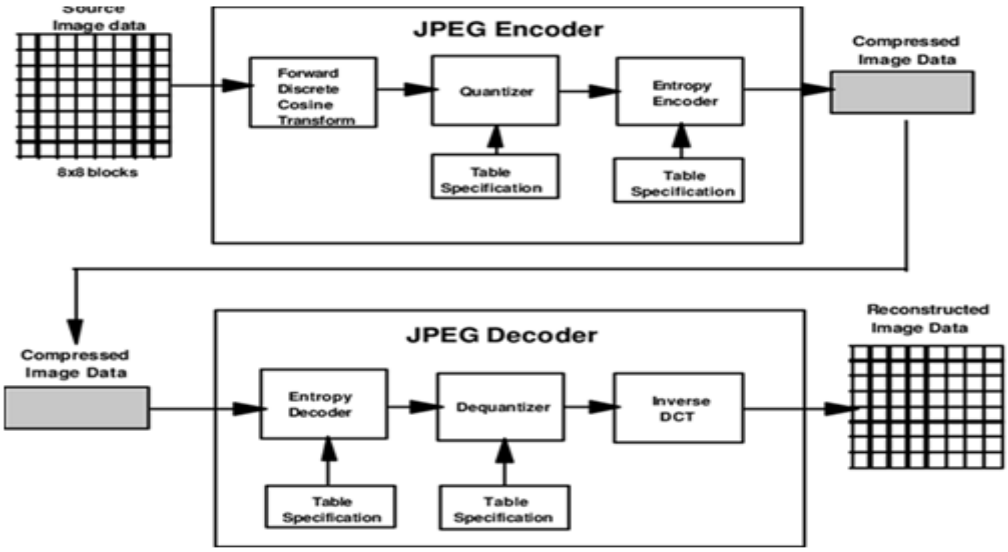


Fig 2.1: block diagram of JPEG compression & decompression

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 9, Issue 05, ISSN No. 2455-2143, Pages 144-150
Published Online September 2024 in IJEAST (http://www.ijeast.com)

A. Forward Discrete Cosine Transform:

The Forward Discrete Cosine Transform (FDCT) is a first step in JPEG compression that enables the efficient representation of images by transforming pixel data into the frequency domain. This mathematical operation processes 8x8 blocks of an image, effectively separating the image into different frequency components. By analyzing these components, FDCT retains low-frequency information that corresponds to the essential visual details, while high frequency components, which capture rapid changes and noise, can be discarded during quantization. This aspect of FDCT not only facilitates significant file size reduction but also optimizes the image quality by focusing on perceptually relevant details. The process is designed to align with human visual perception, ensuring that the most noticeable features are preserved even after compression. By utilizing cosine functions to represent pixel values, FDCT enhances the robustness of image encoding, making it suitable for various applications beyond JPEG, including video processing and other image analysis tasks. Ultimately, the FDCT lays the foundation for the subsequent stages of JPEG compression, highlighting its importance in modern digital imaging and communication systems.

$$F[k,l] = \alpha(k)\alpha(l)\sum_{m=0}^{N-1} f(m,n)\cos\left[\frac{(2m+1)\pi k}{2N}\right] \cos\left[\frac{(2n+1)\pi l}{2N}\right] \quad (1)$$

where,

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}} \text{ if } k = 0 \\ \sqrt{\frac{2}{N}} \text{ if } k \neq 0 \end{cases} \quad (2)$$

Assuming an interleaving block of 8 × 8 (single-component, grayscale images) as inputs for the FDCT, 64 DCT coefficients are produced. These coefficients are then used in the quantization process as shown in the block diagram.

B. Quantization:

Quantization is a second step in image compression, particularly in the JPEG algorithm, where it effectively transforms continuous DCT coefficients into discrete values, significantly enhancing data reduction. This process operates by mapping a range of input values to a smaller set of output values, similar to rounding numbers to their nearest integers. After applying the Forward Discrete Cosine Transform (FDCT) to an 8x8 block of pixels, each DCT coefficient is divided by a corresponding value from a predefined quantization table and then rounded to the nearest integer. These tables are designed based on human visual perception, emphasizing the retention of lower-frequency components that carry more critical image information, while allowing higher-frequency components—often linked to finer details and noise—to be approximated more coarsely or even discarded altogether. Although this introduces a loss of some image quality, the significant reduction in file size makes quantization a beneficial trade-off. Moreover, by increasing the number of zero coefficients in the resultant data, quantization enhances the efficiency of subsequent entropy encoding methods, such as Huffman coding, allowing for even greater compression. This makes quantization a fundamental aspect of modern digital image processing, balancing size reduction with quality preservation.

The figure shows the simple quantization of a signal by choosing the amplitude values closest to the analog amplitude. In the given implementation, quantization is performed in conjunction with a quantization table and the input signal is digitized. This process is fundamentally lossy since this is a many-to-one mapping. This method causes image loss in lossy image compression for DCT-based encoders. Quantization can be represented using equation (9). This is done by rounding of the quotient of dividing each DCT coefficient by its corresponding quantum to the nearest integer.

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 9, Issue 05, ISSN No. 2455-2143, Pages 144-150
Published Online September 2024 in IJEAST (http://www.ijeast.com)

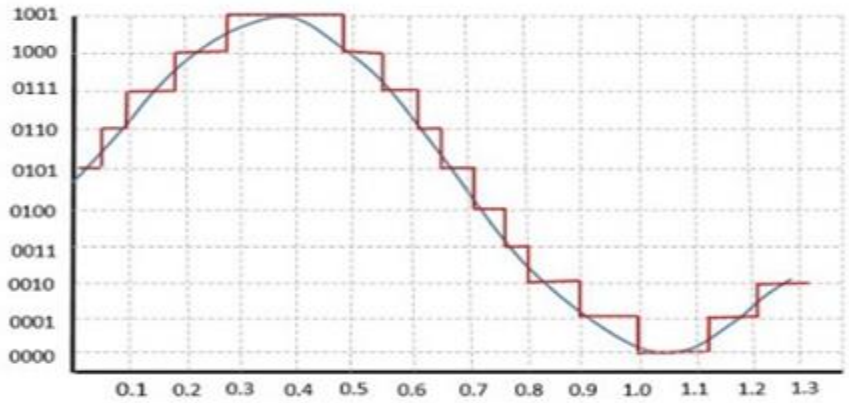


Fig 2.2: Quantization

C. **Entropy Encoding:**

The JPEG algorithm concludes its process by outputting the elements of the DCT block through an entropy encoding mechanism that combines Run-Length Encoding (RLE) and Huffman encoding principles. The entropy encoder generates a sequence of three tokens, which are repeated until the block is fully encoded. These tokens include the **run length**, indicating the count of consecutive zeros that precede the current non-zero DCT coefficient; the **bit count**, representing the number of bits used to encode the amplitude value based on the Huffman encoding scheme; and the value of the DCT coefficient.

As quantization progresses, an increasing number of coefficients become zero, particularly among higher frequency components.

This results in a high likelihood of encountering zeros within the DCT output matrix. To enhance efficiency, the data is reordered to prioritize low-frequency values, with higher frequencies appearing later in the sequence. Consequently, it becomes common to encounter sequences such as "0 0 0 0 0 0 0 0 0 0." Instead of storing these zeros individually, the data can be compressed into a more efficient representation, like "10x 0," significantly reducing the amount of data stored.

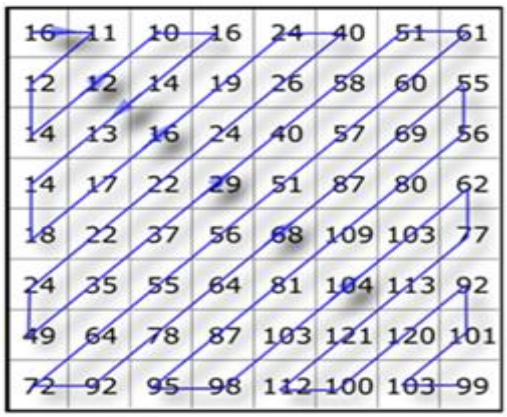


Fig 2.3: Zigzag pixel selection pattern

Huffman Coding

Entropy encoding further optimizes compression through a compact representation of the quantized DCT coefficients, employing techniques like Huffman coding. Specified in the JPEG standard, Huffman coding is utilized in baseline sequential codecs and all operational modes of JPEG compression, effectively encoding source symbols based on their probabilities. Introduced by David Huffman in 1952, this variable-length encoding algorithm minimizes the average number of bits required for symbol representation, given that it adheres to the prefix condition.

The Huffman coding process initiates with a frequency table, detailing the occurrence of each symbol. The algorithm then constructs a Huffman tree from this table, where each node represents a symbol, its frequency, and pointers to its parent and child nodes. The tree grows through successive iterations that identify the two nodes with the lowest frequencies that have not yet become parents. A new node is created, serving as the parent of these two nodes, and its frequency is the sum of the frequencies of its children. This process continues until a single node remains, which becomes the root of the Huffman tree.

During compression, the algorithm traverses the tree starting from the leaf node corresponding to the symbol to be compressed. As it navigates up to the root, it determines whether the current node is a left or right child of its parent, assigning a corresponding binary value of (0) or (1). This efficient approach allows for optimal encoding of the quantized coefficients, ultimately contributing to the overall effectiveness of JPEG compression [2][18][10].

D. Decompression-

The decompression process reverses the compression steps but follows a different order to ensure the original image is accurately reconstructed. Initially, the algorithm retrieves the Huffman tables that were saved with the compressed image. Using these tables, it decodes the Huffman tokens that

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 9, Issue 05, ISSN No. 2455-2143, Pages 144-150
Published Online September 2024 in IJEAST (http://www.ijeast.com)

represent the compressed data. Next, the focus shifts to the Discrete Cosine Transform (DCT) values. The first coefficient of each DCT block is recovered, and the remaining 63 coefficients are reconstructed, with zeros filled in where quantization has led to loss of information. This step is crucial for re-establishing the block's integrity, ensuring that any missing values do not impede the reconstruction process [2][18].

Following the coefficient recovery, the JPEG algorithm decodes the DCT values in a zigzag order, which is instrumental for efficiently retrieving spatial frequency information. This organization aids in reconstructing the pixel values with enhanced visual fidelity. The final stage involves applying the Inverse Discrete Cosine Transform (IDCT) to translate the frequency components back into the spatial domain. During this process, the algorithm evaluates the contribution of each of the 64 frequency coefficients to their corresponding pixels, thus reconstructing the original image as closely as possible. This meticulous approach is fundamental in ensuring that the decompressed image retains its quality while effectively utilizing the information encoded during compression [18]**.**

III. EXPERIMENT AND RESULTS

For this analysis, the Pavia University (PaviaU) dataset was employed. The PaviaU dataset is a widely used benchmark for hyperspectral image processing tasks. It consists of a hyperspectral image captured over the city of Pavia, Italy, by the ROSIS sensor (Reflective Optics System Imaging Spectrometer). The dataset includes 610x340 pixels, each pixel having 103 spectral bands after noisy bands are removed, covering wavelengths in the visible to near-infrared range. The PaviaU dataset is particularly useful for applications in urban analysis, where detailed spectral information is necessary to distinguish between different materials or land cover types, making it ideal for assessing the effectiveness of hyperspectral image compression methods.

Samples of the original band is given:

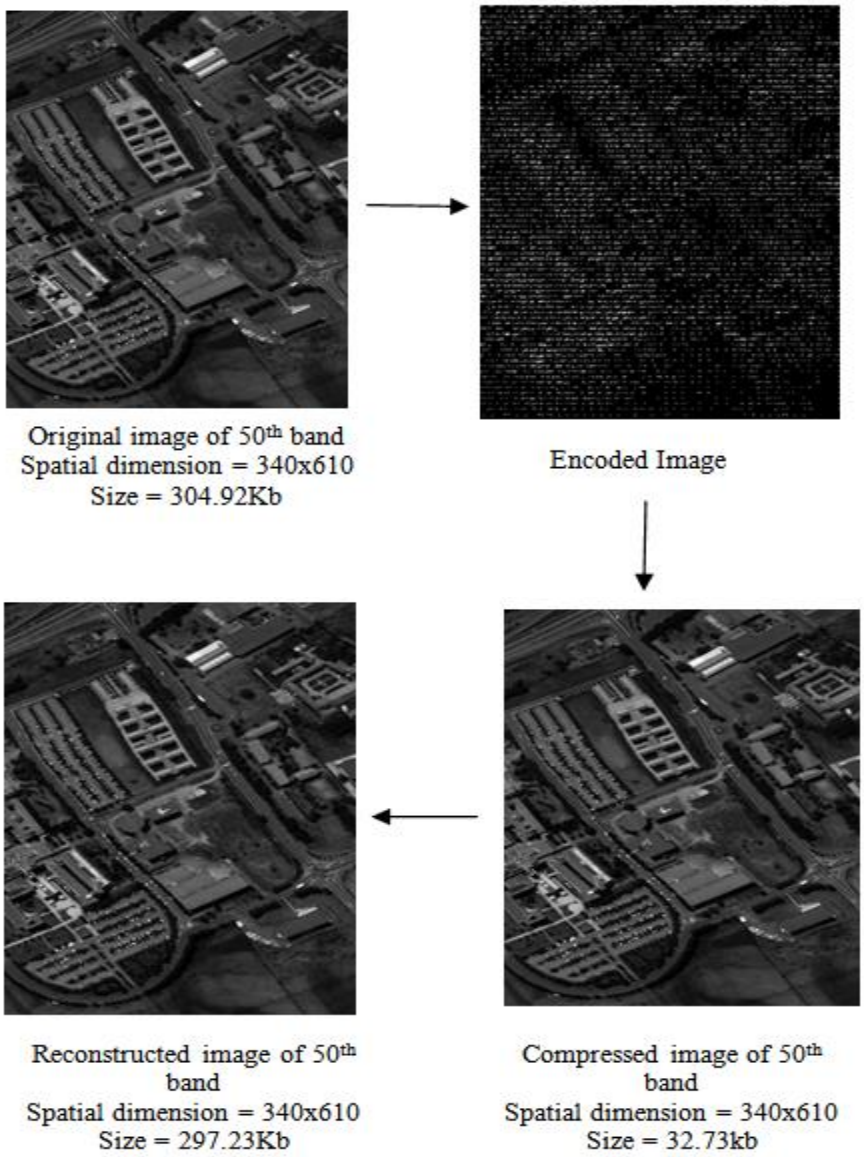
Band 50

Band 70

Band 85

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 9, Issue 05, ISSN No. 2455-2143, Pages 144-150
Published Online September 2024 in IJEAST (http://www.ijeast.com)

Procedure demonstrated for band-50:



A similar compression process is performed across all 103 spectral bands to ensure efficient image reduction and facilitate successful transmission through various channels. Each band, containing unique spectral information, undergoes a standardized compression algorithm that minimizes data size while preserving critical details..By compressing each band individually, the resulting image retains its high-dimensional characteristics, which are crucial for subsequent analyses and applications. This systematic approach not only optimizes the data for storage but also enhances the transmission speed, making it feasible to transfer the compressed images over networks with limited bandwidth. Ultimately, this strategy enables reliable and efficient access to valuable hyperspectral data across multiple applications, ensuring that critical information is preserved while reducing the logistical challenges associated with large datasets.

Result:

Metric	**Band 50**	**Band 70**	**Band 85**	**Band 100**	**Average**
Compression Ratio (CR)	15.45 Kb	11.71 Kb	10.84 Kb	10.29 Kb	12.07 Kb

IV. CONCLUSION

In summary, the JPEG compression algorithm remains a fundamental method in digital image processing, effectively balancing the need for image quality and reduced file sizes. The various stages of JPEG compression—including Forward Discrete Cosine Transform (DCT), quantization, and entropy encoding—collaborate to create a compact representation of visual data. By focusing on lower-frequency components that are more perceptible to the human eye, JPEG maximizes compression while minimizing visual degradation.

Huffman coding plays an essential role within the entropy encoding phase, allowing for efficient data representation by assigning shorter codes to more frequent values. The careful orchestration of these techniques ensures that, during decompression, the original image can be accurately reconstructed. As technology progresses, the JPEG algorithm continues to evolve, adapting to meet contemporary challenges in image processing, such as high-resolution imaging and real-time applications. Future innovations may further enhance compression efficiencies without compromising quality, reinforcing JPEG's relevance in diverse fields like medical imaging, satellite imagery, and digital media. This adaptability highlights JPEG's enduring significance in modern imaging technologies.

V. REFERENCE

[1]. Soni, H., & Gupta, S. (2020). "A Study on Lossy Image Compression Techniques." International Journal of Computer Applications, (pp. 1-15).

[2]. Jain, A. K. (2021). "Image Compression Techniques: A Comprehensive Review." International Journal of Image and Graphics, (pp. 1-20).

[3]. Jia, C., Wu, H., & Wang, L. (2021). "Efficient JPEG Image Compression Method Based on Optimized Huffman Coding." Journal of Information Science, 47(2), 238-250.

[4]. Zhang, L., Liu, Z., & Wang, X. (2022). "Adaptive Huffman Coding for Image Compression." Applied Sciences, 12(4), 1804.

[5]. Feng, L., Zhang, H., & Zhao, J. (2022). "Color Image Processing Using YCbCr Color Space." Journal of Visual Communication and Image Representation, (pp. 1-15).

[6]. Khan, A., & Kumar, P. (2022). "Entropy Coding Techniques in Image Compression." Journal of Computer Science and Technology, (pp. 1-18).

[7]. Zhang, J., Wang, L., & Zhou, Y. (2022). "Efficient DCT Algorithms for Image Compression." Journal of Real-Time Image Processing, 19(3), 605-617.

[8]. Wang, C., Zhou, X., & Liu, Y. (2023). "An Overview of Image Compression Algorithms." Applied Sciences, (pp. 1-20).

[9]. Huang, H., & Yao, X. (2023). "A Fast Huffman Coding Algorithm for Image Compression." Journal of Visual Communication and Image Representation, 104, 103338.

[10]. Lee, S., Kim, J., & Cho, H. (2023). "Improving JPEG Compression Efficiency Using Enhanced Huffman Coding Techniques." IEEE Transactions on Image Processing, 32, 1872-1883.

[11]. Singh, R., Kumar, A., & Gupta, S. (2023). "Adaptive Discrete Cosine Transform for Image Compression." International Journal of Imaging Systems and Technology, 33(1), 234-244.

[12]. Chen, Y., Liu, X., & Wang, H. (2023). "Perceptual Quantization for Image Compression Based on Human Visual Sensitivity." IEEE Transactions on Image Processing, 32(4), 1020-1034.

[13]. Wang, R., Zhang, Y., & Liu, Z. (2023). "Adaptive Huffman Coding for Enhanced JPEG Compression." Journal of Visual Communication and Image Representation, 88, 103271.

[14]. Bhatia, S., Singh, A., & Sahu, S. (2023). "Performance Analysis of Arithmetic Coding in JPEG Compression." Journal of Electronic Imaging, 32(1), 013001.

[15]. Zhao, Y., Liu, D., & Zhang, X. (2023). "Parallel Processing Techniques for Fast JPEG Decompression." Journal of Computer and System Sciences, 131, 105-116.

[16]. Khan, M. A., Ahmad, J., & Alam, M. (2023). "Machine Learning-Based Image Reconstruction for JPEG Decompression." Pattern Recognition Letters, 168, 25-32.

[17]. Jayalakshmi, B., Satish, B. K., & Rajan, V. K. (2023). "A Comprehensive Review on Image Compression Techniques." Journal of Ambient Intelligence and Humanized Computing, (pp. 1-30).

[18]. Li, P., Wu, Y., & Zhang, Y. (2024). "Comparative Analysis of AVIF and JPEG for Image Compression." Journal of Visual Communication and Image Representation, (pp. 1-20).