International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Optimizing Image Transmission in Low-Bandwidth Wireless Sensor Networks for Real-Time Applications

Pratibha[1*], Shikha Singh[2], Sandip Vijay[3], Sandeep Kumar Dubey[4]

Submitted: 26/01/2024 **Revised**: 04/03/2024 **Accepted**: 12/03/2024

Abstract- In wireless image sensor networks, a multimedia sensor node equipped with a miniature camera captures a series of images and transmits them to a sink through multiple wireless hops. This process entails stringent requirements, including reliable data transmission, minimal energy consumption of nodes and low end-to-end delay in image transmission. This work proposes a data compression algorithm with low complexity tailored for WSNs in structural health monitoring and security applications. The algorithm employs a pruning approach to the H.265 compression standard optimized for continuous image frames extracted from real-time videos (exceeding 25 fps). It achieves superior compression ratios compared to original images, facilitating efficient data storage. By reducing the number of data bits, the algorithm extends the network lifetime by mitigating node failures due to resource scarcity. Implementation involves real-time image capture of natural scenes using an advanced Raspberry Pi4B+WSN gateway equipped with camera modules. Evaluation includes various parameters such as Peak Signal to Noise Ratio, Mean Square Error, Bit Rate, Compression Ratio, and Structural Similarity Index. This technique optimally balances energy consumption, image quality and data transmission accuracy in each cycle, demonstrating its effectiveness for practical deployment.

Keywords- *High efficiency video coding, Image compression, Mean Square Error, Peak signal to noise ratio, Wireless multimedia sensor network,*

1. INTRODUCTION

In today's world, the demand for multimedia applications in Wireless Sensor Networks (WSNs) is soaring as they have become integral part to day to day life. However, WSNs encounter limitations such as battery issues, security and slow communication [1]. To counter these limitations, researchers focus on multimedia sensors that capture video, audio, images and transmitting such data across the network. WSNs consist of many nodes which are used to detect any physical parameters such as sound, motion and temperature etc. Target tracking, medical applications, security and habitat detection are the major applications of WSNs [2].

WSNs face higher data loss compared to wired networks. While congestion causes loss in wired setups, WSNs suffer from node failures, interference, noise and unreliable links. Communication over WSNs is not reliable due to wireless links and high error rates. Requirement of multimedia transmission add challenges of time constraint, high bandwidth and reliability, impacting communication protocol for effective transmission [3].

[1]*Electronics & Communication Engineering, Shri Ramswaroop Memorial University, Barabanki, Uttar Pradesh, India*
[2] *Electrical and Electronics Engineering, Shri Ramswaroop Memorial University, Barabanki , Uttar Pradesh, India*
[3]*Electronics & Communication Engineering, Tula's Institute, Dehradun ,Uttarakhand, India*
[4]*Computer Science & Engineering, Shri Ramswaroop Memorial College of Engineering and Management, Lucknow, Uttar Pradesh, India*
** Corresponding Author Email: pratieic@gmail.com*

Fig. 1 illustrates a typical WSN setup having a base station situated amidst users and network. Network features sensor nodes that uses radio signal to communicate. These nodes have computing devices, sensors, power components and radios. When a sensor detects a significant event, it sends a message to the base station. Message priority affects delivery ratio and delay [1]. Critical messages will be

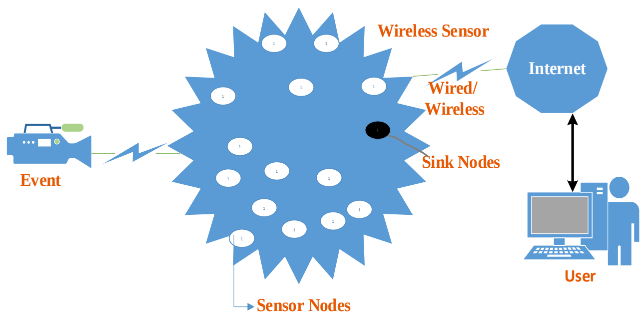


Figure 1-Traditional wireless sensors network

given priority, while irrelevant ones take longer time. Delayed data at the receiver end is discarded. Real-time wireless multimedia sensor networks (WMSNs) finds in applications such as surveillance, healthcare and efficient multimedia transfer, which faces the challenge of high bandwidth demand [2].

To tackle the bandwidth issue, a solution is sought by combining data compression and minimum loss transmission. In a scenario where a WMSN is deployed for real-time video monitoring, compression reduces frame size. Receivers reconstruct the original frame from compressed measurements using recovery algorithms. Instead of transmitting entire data, these compressed measurements are sent as packets through the network using motes [3, 4, 5].

This paper explores a pruned approach to the conventional H.265 compression standard, aiming to reduce computational complexity and conserve energy. Subsequently, the image data undergoes local compression at the monitoring nodes, achieving a reduction of over 85%. This ensures that each node's image fulfills the requirements for real-time data transmission while significantly diminishing the overall power consumption of the system [6]. Both existing and proposed algorithms are evaluated using a finite set of test image frames and their performance is compared based on standard metrics such as Compression Ratio (CR), Peak Signal to Noise Ratio (PSNR), Bit Rate, Mean Square Error (MSE) and Structural Similarity Index (SSIM). The processing time and implementation are assessed using a Raspberry Pi4B+WSN gateway equipped with camera modules and Python programming language, with graphical analysis conducted using Python libraries. Recorded results indicate improved values across the aforementioned metrics, demonstrating the effectiveness of the proposed approach

The subsequent sections of the paper are structured as follows: Section II furnishes an outline of pertinent literature. Section III presents an overview of the methodology. Section IV delves into comparative evaluation metrics. Section V includes overview of H.265 compression techniques. Section VI deals with the experimental setup. Section VII discusses the simulation results and analysis. Lastly, in Section VIII summary and suggestions for future work is discussed.

2. RELATED WORKS

WMSN are gaining attention for their ability to transmit both scalar sensor data and multimedia content. However, unlike scalar WSNs, WMSNs tend to consume more power due to handling multimedia data like pictures, audio, and video. Therefore, efficient energy-aware scheduling algorithms are crucial to extend the network lifetime. Given the remote deployment of multimedia sensors, ensuring a significant network lifetime is essential to avoid frequent replacements, which can be economically impractical. Thus, a thorough examination is necessary before implementing a framework aimed at prolonging the lifespan of WMSNs.

Hui Huang, Zhe Li and their colleagues (2018) developed a real-time image transmission technique for low-bandwidth Wireless Sensor Networks (WSNs). Localised DCT compression at monitoring nodes compresses picture data in their system. The TX1 board processes local picture data and wirelessly sends encapsulated data packets to the server [7]. Another study by R. Pitchai, G. Reshma, J. Raja, and colleagues (2020) developed a priority strategy to detect picture intruders. The technique sends only intruder-containing blocks to the sink node, conserving half the bandwidth. This method saves bandwidth by sending only the necessary data [8]. Caroline Conti et al. created a light-field image codec utilizing High-Efficiency Video Coding techniques. Self-similarity estimation and correction enhanced coding performance compared to prior techniques [9]. Hao et al. (2015) created an adaptive image processing and transmission strategy that balanced energy and computing during transmission using real-time state record matrices, multi-objective optimisation, and cooperative mechanisms [10].

Implementing real-time communication in WMSNs poses challenges as nodes need to broadcast multimedia content and communicate with other nodes in a multi-hop environment. This increases the demand for bandwidth significantly. However, there is hope in Ultra Wide Band communication (UWB) for providing the high-order bandwidth required for multimedia communication [11]. Achieving efficient data compression poses a challenge for sensor nodes due to the demanding computing requirements involved. To enhance energy efficiency in image communication, minimizing the computational complexity of compression processes is crucial. By reducing computational overhead, we can significantly prolong the network's lifespan [12]. Cyclops consists of several key components: an image sensor, a microcontroller unit (MCU), a complex programmable logic device (CPLD), along with external Static Random Access Memory (SRAM) and external Flash Memory, as detailed in references [13][14]. Its firmware, developed in the nesC programming language, operates on the TinyOS operating system. This firmware enables resource transparency, extended calculations, and synchronized access by both the MCU and the CPLD. The software architecture of Cyclops is structured into three main components: libraries, drivers, and sensor applications. Drivers facilitate communication between the MCU and various peripherals. Structural analysis libraries, crucial for processing raw images, provide primitive structural analysis capabilities. Additionally, high-level algorithmic libraries handle more complex tasks. Sensor applications execute sensing operations tailored to the host's requirements. Cyclops also boasts compatibility with other sensor nodes, utilizing the Micaz node for communication [15].

The CMUcam3 stands as an accessible, open-source solution tailored for straightforward vision tasks, offering affordability and full programmability. Its hardware architecture centers around three key elements: a microprocessor, a frame buffer, and a CMOS camera chip. Due to its limited 64KB of RAM, standard Vision Libraries aren't compatible. To address this limitation, the CMUcam3 incorporates its own software, the CC3 software vision system, presenting a C Application Programming Interface (API) for vision and control tasks [16]. For expanded functionality, the CMUcam3 can interface with a TelosB mote for communication with additional sensor nodes. While not suited for complex vision algorithms due to its performance and memory constraints, the CMUcam3 finds utility in low-resolution video applications [17]. The algorithm employed a pruning method for approximating the Discrete Cosine Transform, wherein the transformation matrix undergoes modification to streamline the process, resulting in enhanced compression ratios. This optimization facilitates easier data storage compared to the original image. The efficacy of this approach is validated through real-time image capture of concrete walls in buildings using a Raspberry Pi3B+WSN gateway equipped with camera modules. Furthermore, the scheme is thoroughly scrutinized across various parameters including PSNR, MSE, etc., culminating in a balanced trade-off between energy consumption and image quality [18].

3. Proposed Methodology

This research aims to enhance real-time image coding so that transmission of information over WMSN can be easily processed. Good compression of data opens up various applications requiring prompt delivery of high-quality image data. Data processing is carried out in three phases: Acquisition of real time video and extract in image frames, apply optimal compression technique on extracted image frames, transmit the compressed image frames over wireless sensor network. At the receiver side, receive the compressed image for retrieving original image with minimum loss. Proposed algorithm is discussed in table-1.

Table 1: Proposed algorithm

Step 1: Analyze Video Content: Analyze the video content in real-time (≥25 fps).

Step 2: Partition Video Sequence: Partition the video sequence into multiple frames to process each frame individually.

Step 3: Apply Compression Technique: Apply an optimal compression technique to reduce irrelevance and redundancy of image data, ensuring efficient storage or transmission.

Step 4: Transmit Compressed Image: Transmit the compressed image over a WSN to the receiver side.

Step 5: Receive: At the receiver side, receive the compressed image data.

Step 6: Decompress: Reconstruct the original image by applying the appropriate decompression technique.

Step 7: Calculate Accuracy Rate: Calculate the transmission accuracy rate.

The core concept involves capturing instantaneous video using a camera for real-time image acquisition. However, the challenge lies in the large image frame sizes, often exceeding 78KB, which strain the limited resources of multimedia devices. An effective compression algorithm can extend sensor operation, reduce data size and save energy. By employing diverse compression techniques, best strategy for achieving optimal outcomes in terms of compressed image dimensions, compression ratio, bit rate, and PSNR through real-time processing will be opted. Reduced file size allows for storing more images within limited storage capacity, ultimately improving sensor longevity and reducing data size and power consumption [19,20].

4. EVALUATION METRICS FOR COMPRESSION METHODS

In order to transport image and meet Quality of Service (QoS) requirement for multimedia communication, focus is laid on optimal compression technique. In order to effectively compare and select an optimal compression technique compression ratio (CR), bit rate, Peak signal to noise ratio (PSNR) and Mean Square Error (MSE) etc play an important role.

A. Mean Square Error:

MSE is a universal metric for evaluating image quality, serving as a comprehensive reference by comparing the reconstructed image to a reference image. As the MSE approaches zero, it indicates a significant improvement in image quality. Equation (1) provides the mathematical expression for calculating the MSE.

$$MSE = \frac{1}{mn}\sum_{0}^{m-1}\sum_{0}^{n-1}\|f(i,j) - g(i,j)\|^2$$

(1)

Where m is pixel row count of images, i is specific row index, n is pixel column count of image, j is particular column index in consideration, f is initial image matrix data and g is deteriorated image matrix data.

B. Peak Signal to Noise Ratio (PSNR):

PSNR is a parameter for assessing the quality between original image and compressed image. High value of PSNR results in better quality of compressed image [6]. Equation (2) provides the mathematical representation for PSNR:

$$PSNR = 20\log\log_{10}\left(\frac{MAX_f}{\sqrt{MSE}}\right)$$

(2)

MAXf refers to highest signal value present within the original “known to be good” image.

C. Compression Ratio (CR):

Image compression is the process of reduces the image data file size while preserving the crucial details. CR quantifies the relationship between original image file size and compressed image file size, and is mathematically represented as;

$$CR = \frac{Usize}{Csize}$$

(3)

Usize - original image size, Csize - compressed image size.

D. Bit Rate:

It pertains to the volume of data or information transmitted within a specific time frame. It signifies the total bits needed to state the image per unit of time. The calculation for determining bit rate in bps is expressed as;

$$\text{Bite rate} = \frac{\text{File size in bits}}{\text{Duration in seconds}} \times 8$$

(4)

Here, the bit rate signifies the speed at which data is transmitted or processed, indicating the amount of data processed per unit of time.

E. Structural Similarity Index:

Although the PSNR is a useful and a demanding metric for evaluating the quality of image, it may not fully encompass the image's structural nuances [19]. As a result, to utilize an alternative metric for quality evaluation known as the Structural Similarity (SSIM) index to overcome this limitation.

$$\text{SSIM}(x, y) = \frac{(2\,\mu_x \mu_y + C_1)(2\,\sigma_{xy} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)}$$

(3)

When calculating the Structural Similarity (SSIM) index, first local samples μ_x and μ_y are computed for windows x and y, both of which have a uniform size of N × N. Additionally, standard deviations σ_x and σ_y are calculated for windows x and y, sample cross-correlation σ_{xy} is calculated by subtracting their respective means.

$C_1 = (k_1L)^2$,

(4)

$C_2 = (k_2L)^2$

(5)

The default method for stabilizing division when dealing with a weak denominator relies on the utilization of two variables, C_1 and C_2. The pixel values' dynamic range, represented as L, which is expressed as,

$L = 2^{\text{number of bits per sample (bps)}} - 1.$

Moreover, the preset values for the constants k1 and k2 are established at 0.01 and 0.03, respectively.

5. H.265 COMPRESSION

H.265 is also named as High-Efficiency Video Coding (HEVC), represents a cutting-edge compression technique applicable on videos to significantly enhance the efficiency of video compression.

The HEVC method is well-suited for compressing real-time images. It is recognized for its versatility in video coding, making it adaptable to a broad scale of applications, comprising real-time image compression. This technique surpasses previous standards by enhancing compression efficiency, enabling the real-time transmission and storage of high-quality images [21]. HEVC's progress in compression performance, parallel processing, and compatibility with diverse resolutions and bit depths positions, make it a suitable choice for compressing real-time images. This allows for the reduction of bit rates while preserving acceptable visual quality [22].

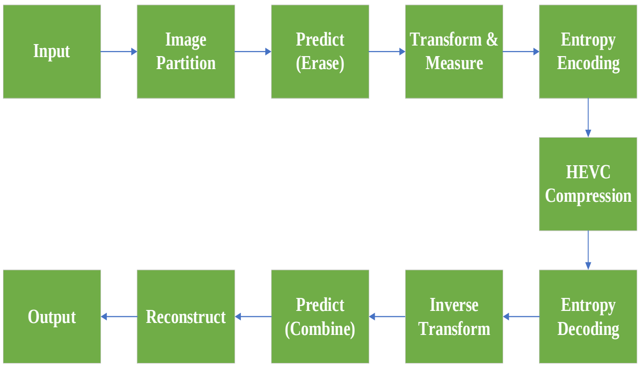


Figure 2- Block diagram of HEVC CODEC

The HEVC codec encoder undergoes several stages as shown in Fig. 2:

- Prediction: The encoder predicts each unit by analyzing a frame consisting of 16x16 displayed pixels (macroblocks). It generates predictions based on the current frame or previously transmitted frames.
- Quantization and Transformation: Following prediction, the encoder quantizes and transforms the residual, which represents the variance between the prediction and the original picture unit. This process utilizes an approximate form of Discrete Cosine Transform, resulting in a set of coefficients representing the weighting value of fundamental patterns. These coefficients are then combined to produce the initial residual.
- Entropy Encoding: The encoder employs entropy encoding, a lossless data compression method, to compress the transformed output, mode information, prediction details, and headers.

On the other hand, an HEVC decoder performs the following tasks:

- Entropy Decoding: The decoder reverses the entropy encoding steps, extracting the original elements from the coded sequence.
- Transformation Reversal and Rescaling: It inverses the transformations applied during encoding and rescale the data.
- Prediction: The decoder predicts each unit and adds it to the output of the inverse transform.
- Reconstruction: Finally, the decoder reconstructs the final decoded video image [23, 24].

6. EXPERIMENTAL SETUP

One of the main objectives of this research is to design, implement the hardware setup for WMSN and test for continuous transmission of multiple compressed frames [25]. For this, real time images are captured by using a Raspberry pi 4 module which is fitted with digital camera as shown in Fig. 3. The setup of Raspberry pi 4 module is equipped with digital camera can be installed at each node of WSN which switches it in wireless multimedia sensor network (WMSN), makes it suitable for recording video and clicking images [26]. The same module is used to run the algorithm to compress the continuous image frames, transmit it and to evaluate the time taken to compute 640×480 dimensional images.

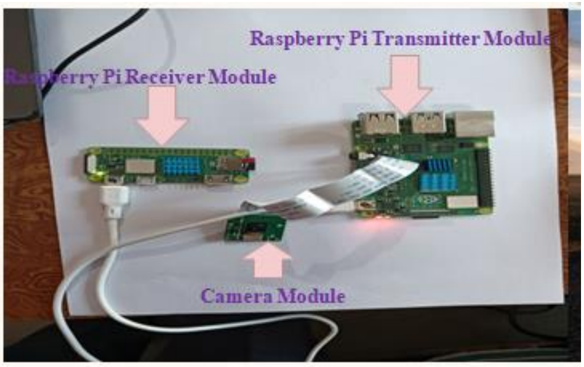


Figure 3- Raspberry Pi 4 Setup

For the end user, the Raspberry Pi 4 Model B provides desktop performance comparable to that of entry-level x86 PC systems [27]. VNC stands for Virtual Network Computing. Any device connected to a Local Area Network is assigned an IP address. To connect to the Raspberry Pi from another machine using VNC, the Raspberry Pi's IP address is required. This is easy if a display is connected and there are number of methods for finding it remotely from another machine on the network. The best thing is that the Raspberry Pi allows more than one display at the same time, so it will be easy to analyze the output [28]. In this research, the RealVNC viewer open-source platform has been selected, which provides a separate desktop other than the PC desktop for coding and viewing the output, as shown in Figures 4 & 5. VNC is a cross-platform screen-sharing system created to remotely control another computer. This means that a computer's screen, keyboard and mouse can be used from a distance by a remote user on a secondary device, as though they were sitting right in front of it.

Figure 4- Desktop of transmitting device

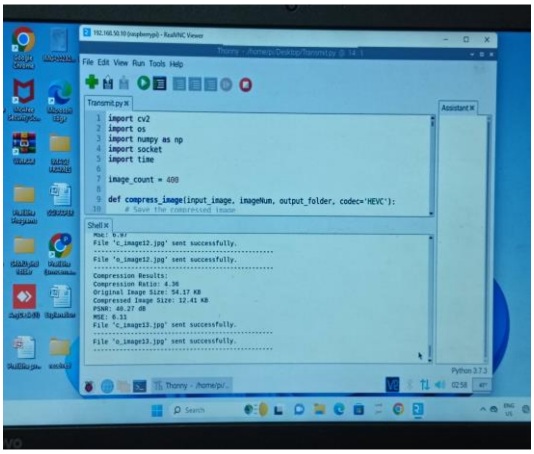


Figure 5- Coding for image acquisition, compressing & transmitting image frames

7. SIMULATION RESULTS AND ANALYSIS

The Raspberry Pi camera module captures images of size 640×480, enabling efficient algorithm testing. Fig. 6 demonstrates real-time video acquisition and reconstruction into multiple original image frames at 30 frames per second, which were utilized for simulation and testing. The algorithm underwent rigorous testing using standard office environment image sets. Fig. 7 showcases sampled compressed simulation results derived from the modified algorithm applied to real-time images. It's essential to assess evaluation metrics during image frame compression and transmission from the transmitter to the receiver node. Fig. 8 displays a screenshot illustrating the calculation of evaluation metrics for each frame, crucial for measuring compressed image quality. Table-2 shows the simulation parameters for the experimentation.

Table 2. Simulation Parameters

Model Variables	Values
Image Length	640

Image Width	480
Depth of Pixel	8
Frames Per Second (FPS)	30 fps
M	10, 20, 30, 40
Tt	1/250 s
Vt	3.3 V
It	19.5 mA

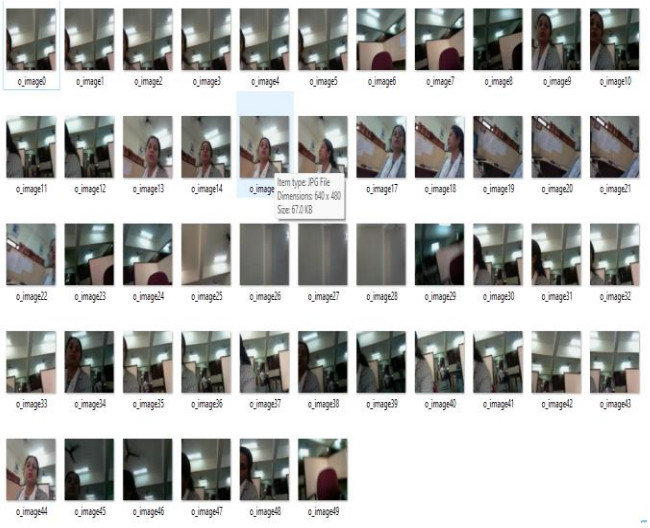


Figure 6- Real time video acquisition and rebuild into multiple original image frames (30 frames per sec)

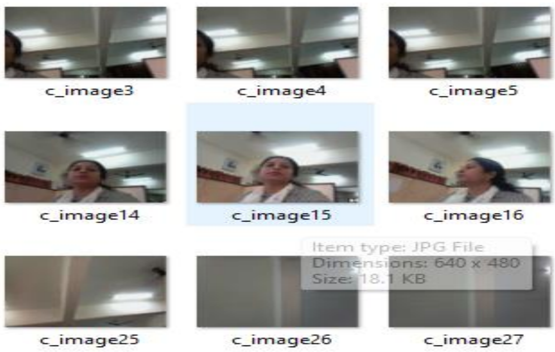


Figure 7- Sample of compressed image frames

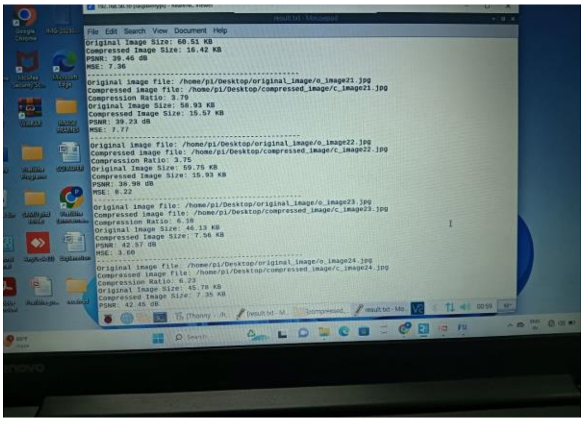


Figure 8- Calculation of evaluation metrics of each frame

Table-3 shows the comparison of original size and compressed size of 50 image frames along with generating the processing time of each image frame during the transmission from sender to receiver. It is found that the size of real time original image frame lies between 78.45KB to 30.85KB during this experimentation and the compressed image frame size ranges from 21.16KB to 7.47KB. The processing time of each frame varying between 0.4592µS to 0.1062µS according to the compressed image size. Table-4 shows the comparison of evaluation metrics such as compression ratio, PSNR, bit rate, MSE, SSIM between original image and compressed image. The MSE varies directly with the PSNR, where higher PSNR values correspond to lower MSE values. Additionally, the Structural Similarity Index (SSIM) of reconstructed images at the receiver end indicates the resemblance between the original and reconstructed images, measured on a scale from 0 to 1. A value of 1 signifies 100% similarity, while 0 denotes no similarity. SSIM leverages pixel interdependencies to quantify image quality degradation resulting from image and data processing during compression, providing a perceptual metric for assessing losses. The bit rate signifies the speed at which data is transmitted or processed, indicating the amount of data processed per unit of time. In this result, the maximum value of bit rate is 169.27 bps for the image frame 40.

Table 3- Set of image frames with original & compressed

size, processing time for a 640×480 Dimension

Image Frame No.	Original Size (KB)	Compressed Size (KB)	Processing time (µs)
Image 0	54.78	14.73	0.4592
Image 1	54.88	14.71	0.3715
Image 2	54.78	14.62	0.3567
Image 3	55.09	14.75	0.3647
Image 4	55.34	14.79	0.3584
Image 5	55.45	14.78	0.3496
Image 6	51.49	13.86	0.3418
Image 7	50.07	13.52	0.3599
Image 8	54.22	13.87	0.3536
Image 9	71.25	19.56	0.3658
Image 10	65.88	17.73	0.3712
Image 11	68.56	18.89	0.3637
Image 12	69.13	18.91	0.3974
Image 13	64.44	16.87	0.3748
Image 14	60.22	16.23	0.3726
Image 15	67.1	18.1	0.3628
Image 16	63.57	16.85	0.3857
Image 17	78.45	21.05	0.3865
Image 18	77.61	20.65	0.3947
Image 19	72.73	19.64	0.1062
Image 20	75.79	20.39	0.3802
Image 21	75.6	20.24	0.3880
Image 22	53.46	14.02	0.4050

Image 23	56.21	14.29	0.3817
Image 24	54.43	14.78	0.3816
Image 25	36.97	9.49	0.1577
Image 26	30.85	7.48	0.3712
Image 27	31.64	7.47	0.3968
Image 28	31.74	7.53	0.3794
Image 29	42.01	11.36	0.3809
Image 30	61.4	16.85	0.3960
Image 31	58.48	15.84	0.3962
Image 32	56.04	15.25	0.2353
Image 33	54.04	14.77	0.3839
Image 34	65.1	17.67	0.3982
Image 35	66.91	18.59	0.3977
Image 36	68.96	19.26	0.3920
Image 37	69.25	19.35	0.3948
Image 38	53.59	14.42	0.3898
Image 39	64.8	17.86	0.4000
Image 40	75.77	21.16	0.4032
Image 41	75.25	20.03	0.3971
Image 42	53.33	14.78	0.3822
Image 43	55.88	15.65	0.3945
Image 44	65.38	17.76	0.3921
Image 45	47.71	11.42	0.3798
Image 46	51.08	10.97	0.3670
Image 47	51.08	13.42	0.3837
Image 48	60.72	16.50	0.3821
Image 49	50.85	12.91	0.3904

Table 4- Comparison of evaluation metrics

Image Frame No.	Compression Ratio	PSNR (dB)	Bit Rate (bps)	MSE	SSIM
Image 0	3.72	39.92	117.82	6.63	0.99
Image 1	3.73	39.93	117.65	6.6	0.99
Image 2	3.75	39.98	11695	6.53	0.99
Image 3	3.74	39.86	117.96	6.72	0.99
Image 4	3.74	39.87	118.34	6.7	0.99
Image 5	3.75	39.87	118.25	6.7	0.99
Image 6	3.72	40.63	110.84	5.62	0.99
Image 7	3.7	40.9	108.15	5.29	0.99
Image 8	3.91	40.4	110.98	5.93	0.99
Image 9	3.64	37.82	156.45	10.74	0.99
Image 10	3.72	38.49	141.85	9.21	0.99
Image 11	3.63	37.93	151.16	10.47	0.99
Image 12	3.66	37.96	151.27	10.4	0.99
Image 13	3.82	38.81	134.95	8.55	0.99
Image 14	3.71	39.24	129.83	7.74	0.97
Image 15	3.71	38.19	144.8	9.87	0.99
Image 16	3.77	38.7	134.78	8.78	0.94
Image 17	3.73	37.17	168.38	12.49	0.98
Image 18	3.76	37.26	165.16	12.21	0.98
Image 19	3.7	38.01	157.1	10.29	0.98
Image 20	3.72	37.39	163.12	11.86	0.98
Image 21	3.74	37.43	161.88	11.75	0.98
Image 22	3.81	40.42	112.16	5.9	0.99
Image 23	3.93	40.05	114.34	6.43	0.99
Image 24	3.68	40.12	118.27	6.32	0.99
Image 25	3.9	42.93	75.91	3.31	1
Image 26	4.13	44.12	59.82	2.52	1
Image 27	4.24	44.17	59.75	2.49	1
Image 28	4.21	44.37	60.27	2.38	1
Image 29	3.7	42.21	90.86	3.91	0.99
Image 30	3.64	38.94	134.8	8.31	0.99
Image 31	3.69	39.33	126.75	7.59	0.99
Image 32	3.67	39.91	122.02	6.63	0.99
Image 33	3.66	39.96	118.16	6.61	0.99
Image 34	3.69	38.75	141.32	8.68	0.99
Image 35	3.6	38.21	148.72	9.83	0.96
Image 36	3.58	37.99	154.06	10.32	0.99
Image 37	3.6	38.01	154.78	10.27	0.99
Image 38	3.72	40.54	115.36	5.74	0.99
Image 39	3.63	38.67	142.86	8.83	0.99
Image 40	3.58	37.49	169.27	11.58	0.92
Image 41	3.61	37.66	160.25	11.15	0.99
Image 42	3.61	40.14	118.22	6.3	0.99
Image 43	3.57	39.67	125.22	7.02	0.99
Image 44	3.68	38.66	142.05	8.84	0.99
Image 45	4.18	41.39	91.38	4.73	0.99
Image 46	4.18	41.81	87.75	5.53	0.99
Image 47	3.81	40.71	107.35	5.53	0.99
Image 48	3.68	39.12	131.98	7.97	0.99
Image 49	3.94	41	103.27	5.08	0.99

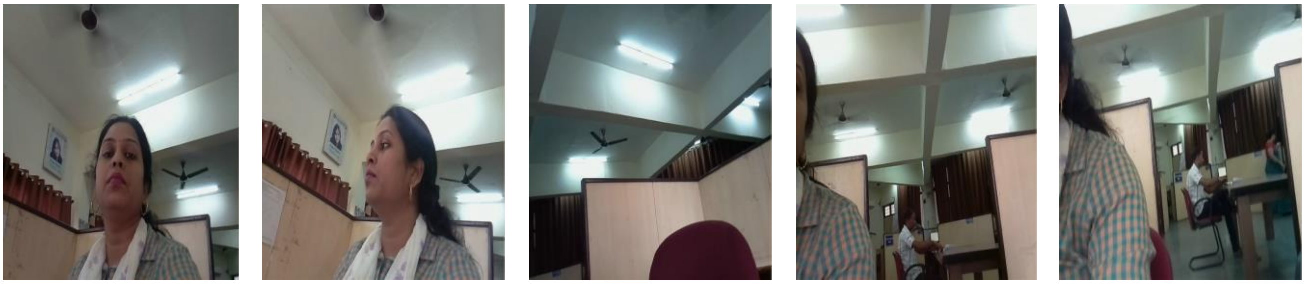
Sampled simulation results from the modified algorithm applied to five selected real-time images in an office environment are depicted in Figs. 9-11, showcasing: the original images of five selected frames, compressed output of the same selected frames and the reconstructed images at the receiving end. These set of image frames are chosen randomly for further analysis, comparison and graphical representation. The parameters, evaluation metrics and the image quality of image frames in all stages starting from sender end to receiver end can be analyzed easily. Image 24 generally performs the best across all metrics, with the highest PSNR, lowest MSE, and bit rate, indicating efficient compression with minimal loss of quality. Image 40 consistently performs the worst, with the lowest PSNR, highest MSE, and bit rate, indicating poorer compression quality compared to other five image frames. Overall, the compression algorithm seems very effective for real time images, as indicated by the high SSIM values across all images. However, Image 24 stands out as the most efficiently compressed image in this dataset.

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

Figure 9- Original image set: (a) Image 14, (b) Image 16, (c) Image 24, (d) Image 35, (e) Image 40

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

Figure 10 - Compressed image set: (a) Image 14, (b) Image 16, (c) Image 24, (d) Image 35, (e) Image 40



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

Figure 11 - Reconstructed image set: (a) Image 14, (b) Image 16, (c) Image 24, (d) Image 35, (e) Image 40

Table 5- Simulation results for selected image sets

Image Frame No.	Original Size (KB)	Compressed Size (KB)	Processing time (µs)	Compression Ratio	PSNR (dB)	Bit Rate (bps)	MSE	SSIM
Image 14	60.22	16.23	0.373	3.71	39.24	129.83	7.74	0.97
Image 16	63.57	16.85	0.386	3.77	38.70	134.78	8.78.	0.94
Image 24	54.43	14.78	0.382	3.68	40.12	118.27	6.32	0.99
Image 35	66.91	18.59	0.398	3.60	38.21	148.72	9.83	0.96
Image 40	75.77	21.16	0.403	3.58	37.49	169.27	11.58	0.92

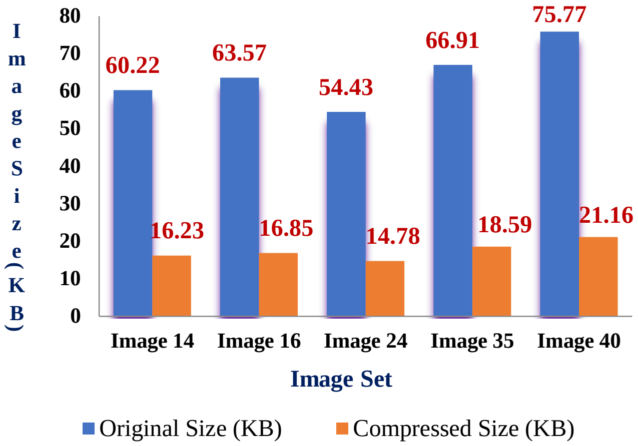


Figure 12- Comparison of original image size and compressed image size

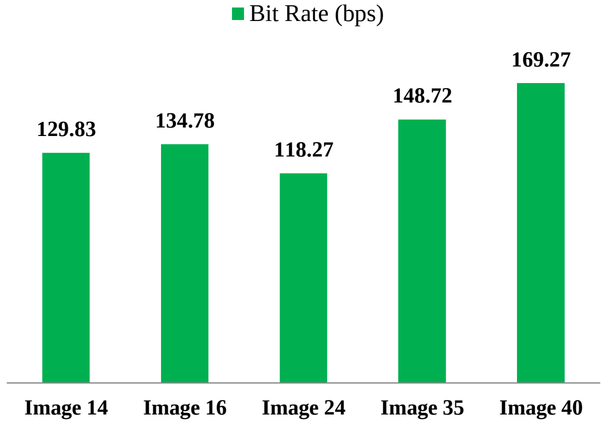


Figure 15- Comparison of bit rate for different image sets

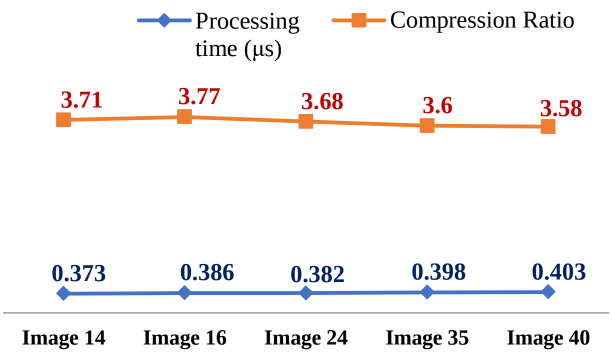


Figure 13- Comparison of processing time and compressed ratio

Figure 16- Comparison of MSE for different image sets

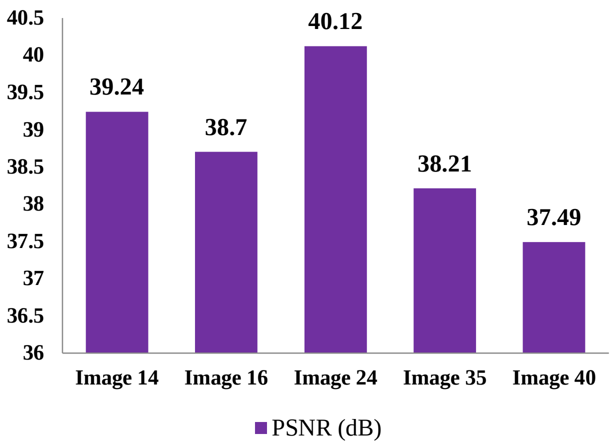


Figure 14- Comparison of PSNR for different image sets

8. CONCLUSION

Ensuring real-time data transmission is critical for both system security and usability when processing and transferring multimedia data. It is required to swiftly transfer data captured by sensors to either another sensor or a base station. To transfer the scalar data or text is quit easier over WMSN but while transferring set of images from one node to another node is very challenging due to their data-intensive nature and limitations of WMSN. To overcome these challenges the data transmission is controlled by the synchronization mechanism without collision. Then, the image data is compressed locally at the monitoring nodes (over 85%) equipped with raspberry pi4 fitted with camera to capture the images in run time, so that the image of each node can meet the needs of real-time data transmission and the overall power consumption of the system is greatly reduced.

In this research HEVC is experimented on real time images which are captured by multimedia devices on continuity basis on each node because there is need of compression technique which respond faster in running time mode and stand compatibly with real time setup. HEVC offers a solution by effectively compressing image data, thereby reducing its size without compromising quality at the sensor node only. This compression not only facilitates efficient transmission but also extends the lifespan of sensors by minimizing the volume of data transmitted. The algorithm achieves a remarkably high compression ratio, removing redundant information before transmission. Despite this aggressive compression, both the Peak PSNR and SSIM values remain high. This indicates not only excellent compression efficiency but also ensures that the reproduced images at the receiving end closely resemble the original captures. In essence, this algorithm not only addresses the challenges of data-intensive image processing but also ensures optimal quality and efficiency throughout the transmission process.

The future research focuses on the integration of the existing research prototype system, making it suitable for all kinds of platforms; in the network management architecture, it supports the function of routing and transferring and realizes

the image data transmission of longer distance.

Conflict of Interest

On behalf of all the authors, corresponding author declares that there is no conflict of interest involved in conducting the study.

REFERENCES

[1] Pratibha, Sandip Vijay, "A Review of Image Transmission using Real Time Technique over WMSN," *International Journal of Applied Engineering Research*, vol. 13, no. 5, ISSN 0973-4562, pp. 2477-2483, 2018.

[2] Ali Hamzah Najim, Sefer Kurnaz "Study of Integration of Wireless Sensor Network and Internet of Things (IoT)," *Wireless Personal Communications*, August 2023.

[3] Akyildiz F, "Asurvey on wireless multimedia sensor networks" *Elsevier Comput Netw J*, 51(4):921–960, 2007.

doi:10.1016/j.comnet.2006.10.002

[4] R. Pitchai, G. Reshma, J. Raja, " An efficient packet image transmission based on texture content for border side security using sensor networks ," *International Journal of Computer Aided Engineering and Technology*, Vol.13, No.1/2, pp.63–72, 2020.

[5] Xianhui Che, Barry Ip, Zhuge Yan, "Field test of multi-hop image sensing network prototype on a city-wide scale" *Digital Communications and Networks*, Volume 5, Issue 2,pp. 131-137, May 2019

[6] G. Sudha, C. Tharini, "Energy-efficient image compression algorithm for wireless sensor networks in structural health monitoring applications" Automatika, 65:2, 483-495, 2024, DOI: 10.1080/00051144.2023.2295145

[7] Hui Huang, Zhe Li, " Real-Time Image Transmission Algorithm in WSN with Limited Bandwidth," *International Journal of Online and Biomedical Engineering (iJOE)* – ISSN: 1861-2121 – Vol. 14, No. 12, 2018, DOI:https://doi.org/10.3991/ijoe.v14i12.9494

[8] R. Pitchai, G. Reshma, J. Raja, " An efficient packet image transmission based on texture content for border side security using sensor networks ," *International Journal of Computer Aided Engineering and Technology*, 2020 Vol.13 No.1/2, pp.63 – 72, DOI: 10.1504/IJCAET.2020.108104

[9] C. Conti, P. Nunes and L. D. Soares, "HEVC-based light field image coding with bi-predicted self-similarity compensation," *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1-4, 2016, doi: 10.1109/ICMEW.2016.7574667.

[10] J. Hao, 'Image processing and transmission scheme based on generalized Gaussian mixture with opportunistic networking for wireless sensor networks', *EURASIP Journal on Wireless Communications and Networking*, 4228, no. 226, 2015.

[11] Moumita Deb, Abantika Choudhury, "A Review of Wireless Multimedia Sensor Network and existing Routing Protocols," *International Journal Computer Science Engg*, vol. 6(3), pp. 343-358 2018.

[12] Shalin Hai-Jew, "Techniques for Coding Imagery and Multimedia: Emerging Research and Opportunities" Publisher: IGI Global, ISBN: 9781522526797, 2017, DOI: 10.4018/978-1-5225-2679-7.

[13] M. Horton and J. Suh, "A vision for wireless sensor networks," in Microwave Symposium Digest, *IEEE*, pp. 361–364, 2005.

[14] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, D. Culler, "The nesC language: A holistic approach to networked embedded systems", *ACM SIGPLAN Notices*, vol. 49, Issue 45, pp 41–51, https://doi.org/10.1145/2641638.2641652.

[15] N. A. Ali, M. Drieberg and P. Sebastian, "Deployment of MICAz mote for Wireless Sensor Network applications," *IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE)*, 2011, pp. 303-308, doi: 10.1109/ICCAIE.2011.6162150.

[16] C. R., I. N. A. Rowe, "A low cost embedded color vision system," *Intell. Robot. Syst. IEEE*, vol. 1,

pp. 208–213, 2012.

[17] TelosB mote specifications, http://bullseye.xbow.com:81/Products.pdf,Wireless.pdf,TelosBDatasheet.

[18] G. Sudha, C. Tharini, "Energy-efficient image compression algorithm for wireless sensor networks in structural health monitoring applications" *Automatika*, 65:2, 483-495, 2024, DOI: 10.1080/00051144.2023.2295145

[19] Huang Hui, Li Zhe, "Real-Time Image Transmission Algorithm in WSN with Limited Bandwidth" *International Journal of Online and Biomedical Engineering*, Vol. 14, ISSN: 1861-2121, No. 12, 2018.

[20] Roman Strukov, Vassilis Athitsos, "Evaluation of Video Compression Methods for Network Transmission on Diverse Data: A Case Study" *16 International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '23)*, pp. 300-305 July 2023.

[21] G. J. Sullivan, "Overview of the High-Efficiency Video Coding (HEVC) Standard," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. no. 22, pp. 1649-1668, Dec. 2012.

[22] V. Sze, M. Budagavi, "Design and Implementation of Next-Generation Video Coding Systems (H.265/HEVC Tutorial)" *IEEE International Symposium on Circuits and Systems*, Melbourne, Australia, June 2014.

[23] T. Kavitha, K. Jaya Sankar, "H.264 Video Compression Using Novel Refined Huffman Codes for Omnipresent Applications" *Wireless Personal Communications*, vol. 131, pp. 2949–2967, 2023.

[24] Y. Zhang, C. Zhang, R. Fan, S. Ma, Z. Chen, C. J. Kuo, "Recent Advances on HEVC Inter-Frame Coding: From Optimization to Implementation and Beyond," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 4321-4339, Nov. 2020, doi: 10.1109/TCSVT.2019.2954474.

[25] Bilal Al-Hayani, Haci Ilhan, "Efficient cooperative image transmission in one-way multi-hop sensor network" International Journal of Electrical Engineering Education, Volume 57, Issue 4, December 2018 https://doi.org/10.1177/0020720918816009

[26] Vujovic V, Maksimovic M. "Raspberry Pi as a wireless sensor node: performance and constraints*"* *37 International Convention in Information and Communication Technology, Electronics and Microelectronics(MIPRO)*, 1013–1018; 2014. https://doi.org/10.1109/MIPRO.2014.6859717

[27] Astudillo-Salinas F "Minimizing the power consumption in Raspberry Pi to use as a WSN gateway*"* *8th IEEE Latin-American Conference on Communications* (LATINCOM), 2016.

[28] S Prasath kumar, P Rayavel "Raspberry pi based secured cloud data" *J. Phys.: Conf. Ser. ICACSE 2020*, 1964 042101, 2021, DOI 10.1088/1742-6596/1964/4/042101.