

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications



ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

Energy-efficient image compression algorithm for wireless sensor networks in structural health monitoring applications

G. Sudha & C. Tharini

To cite this article: G. Sudha & C. Tharini (2024) Energy-efficient image compression algorithm for wireless sensor networks in structural health monitoring applications, *Automatika*, 65:2, 483-495, DOI: [10.1080/00051144.2023.2295145](https://doi.org/10.1080/00051144.2023.2295145)

To link to this article: <https://doi.org/10.1080/00051144.2023.2295145>



© 2024 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 15 Jan 2024.



Submit your article to this journal [↗](#)



Article views: 497



View related articles [↗](#)



View Crossmark data [↗](#)



Citing articles: 2 View citing articles [↗](#)



Energy-efficient image compression algorithm for wireless sensor networks in structural health monitoring applications

G. Sudha and C. Tharini

Department of Electronics and Communication Engineering, B. S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India

ABSTRACT

In Wireless Sensor Networks (WSN), transmitting an uncompressed image consumes more energy than a compressed image, and it is, therefore, the prime requirement to establish energy-aware compression methods to extend the life of the sensor node, and ultimately the network as a whole. This work suggests an image compression algorithm with a low degree of complexity for WSNs in structural health monitoring applications. This algorithm represents a pruning approach to a Discrete Cosine Transform approximation transform in which the transformation matrix is modified to reduce the series of steps and the compression ratio achieved is better compared to the actual image which makes it easy for the data storage. Because of the reduction in the number of data bits, it enhances the lifetime of the network by reducing the number of node failures caused by resource scarcity. The implementation is tested by capturing real-time images of concrete walls in buildings using Raspberry Pi3B + WSN gateway fitted with camera modules. The scheme is also investigated in terms of a variety of parameters like Peak Signal to Noise Ratio, Mean Square Error, Structural SIMilarity Index and Compression Ratio. This technique achieves the best arbitration of energy consumption and image quality.

ARTICLE HISTORY

Received 17 September 2023
Accepted 8 December 2023

KEYWORDS

WSNs; image compression; approximate DCT; pruned 8-point DCT; lifetime improvement; WSN gateway

1. Introduction



Wireless Sensor Networks (WSNs) are distributed intelligent systems equipped with smart cameras with the capability to sense, process and to communicate multimedia information from the environment in real time. These data are rich in multimedia content and hence are bandwidth intensive [1]. But WSNs are resource constrained to handle such bandwidth intensive content. The transmission of this raw data content also consumes considerable network resources. This demands the need for efficient algorithms to reduce the amount of information content by compression, and effective mechanisms to overcome the inefficient use of network resources by defining efficient power optimization techniques. These techniques enhance the capacity of the communication channels.

In this paper, a pruned approach to normal Discrete Cosine Transform (DCT) is considered with the reduced complexity in terms of the number of additions which implies reduced computations to conserve energy. The existing and proposed algorithms have been tested with a finite number of test image sets and the results thus attained by simulation are compared on the basis of some standard performance metrics like Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), Compression Ratio (CR) and Structural

Similarity index (SSIM). The processing time and implementation are tested using Raspberry Pi3B+ and Python and the graphs are plotted using Python. From the results obtained, it is inferred that improved values of the above said metrics are achieved. The section of the following work is written as follows: Section 2 discusses an overview of works and surveys carried out thus far for compression. The current DCTs are covered in Section 3. Section 4 deals with the approach proposed and Section 5 deals with the propagation model. Section 6 discusses the results acquired by the simulations, and Section 7 summarizes the conclusion of the article.

2. Related works

There are several efficient techniques in the literature for image compression currently, and the primary goal of these techniques is to omit non-relevant image data and to efficiently process only the useful information. However, in the process of obtaining the necessary compression ratio, both the useful and irrelevant information are affected and thus suitable methods need to be used depending on the applications. In general, the image compression schemes are classified based on the type of transform used as Discrete Cosine

CONTACT C. Tharini  tharini@crescent.education  Professor and Dean, Department of Electronics and Communication Engineering, B. S. Abdur Rahman Crescent Institute of Science and Technology, Vandalur, Tamil Nadu, 600048 India

© 2024 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

Transform (DCT) and Discrete Wavelet Transform (DWT). [2,3] examine and classify images compression algorithms into Transform Based Schemes and Non-Transform Based schemes which are used in resource-limited systems.

DCT is a block-based image compression scheme that utilizes more energy and produces blocking artefacts, causing performance degradation at low bit rates [2,3]. Although a key disadvantage is blocking artefacts created by the DCT, its limited processing ability, and strong localization in the frequency and time (space) domains lead to significant improvements.

All transform-based schemes under DWT require almost high memory due to the very high number of operations required for effective reconstruction in the decoding end, long processing times, etc. which makes it inappropriate for WSNs and can be used only after effective modifications. With the progressive encoding property, the EZW method is simple and effective, but it takes more passes to compress an input image for efficient reconstruction. As a result, a lot of memory is needed. It is also affected by errors in transmission and packet loss, and it lacks multiresolution scalability [2,3]. As compared to EZW, SPIHT produces an output bit stream with a lower bit rate and compactness, as well as reduced computational complexity since the threshold is halved for each sorting pass and progressive encoding is supported. However, the main disadvantage of SPIHT is its high memory capacity requirements and is vulnerable to packet failures [2,3]. Bit streams in EBCOT can be organized and arranged in a variety of ways. However, the EBCOT algorithm has a higher memory requirement, a longer processing time, and a higher computational complexity, all of which increase power dissipation.

Non-transform based schemes are often computationally and time-intensive, making them unsuitable for wireless sensor networks [2,3]. As a result, the major disadvantages of the techniques discussed thus far that prevent them from being implemented in WSNs are primarily the high memory requirements, the increased number of operations needed to provide successful reconstruction, which results in long processing times, packet losses, and so on, as well as the fact that they can only be used with suitable modifications in the WSNs.

For WSNs, an image compression approach is proposed in [4] that use a combination of DCT and Huffman coding by leveraging the similarity between adjacent pixels, which reduces sensor energy consumption and thus maintains a long network lifetime by putting less strain on nodes. In [5], the author proposes DCT along with Least Squares Interpolation to suggest the usage of the DCT algorithm for image compression in real-time applications with the bit-based transmission. In [6], the author implements an in-exact adder architecture with the implementation of DCT transform using 180 nm CMOS process technology and

proved its efficiency with minimal area and power requirement without any significant compromise in PSNR to assist in the creation of an error-resistant image and video compression applications. In [7], a three-stage DCT is proposed which is a step-by-step process that requires less energy and less delay while the accuracy level is maintained perfectly. In the first stage, all computationally expensive floating-point multiplications are removed and the DCT uses only shift operations and additions. In the second level, an appropriate filtering method is used to remove the information that is not sensitive to humans, and in the third level, an inexact adder implementation is employed to prove the reduction in energy and delay.

In [8], a set of approximations are introduced for the DCT algorithm like floor, ceil, truncation, etc. to make DCT less complex in terms of arithmetic operations with a multiplier less DCT that is incorporated with only the additions and bit shift operations. The approximations were assessed in terms of computations and JPEG-like compression efficiency and found to be a good compromise. A set of different transformations were tested in [9] to prove the effectiveness of DCT algorithms in image compression. Because of its outstanding energy compaction capabilities, DCT is used widely as discussed in [10], which proposes multiplier-free approximate DCT transformations that give greater compression performance at very minimal circuit complexity with 14 additions and extends the network lifetime. In [11], a trade-off strategy for energy consumption and distortion is proposed using fast zonal DCT. A low complexity approach was discussed in [12] that use only additions and bit shift operations to reduce the arithmetic complicity of the DCT process, but bit shift operations consume certain power.

3. DCT-based image compression

The 2-D DCT computation using the direct method is highly computationally intensive with $O(N^4)$ multiplications and $O(N^4)-O(N^2)$ additions for an $N \times N$ block. For example, the common value of N used in any DCT is 8, and the process requires 4096 and 4032 multiplications and additions respectively. DCT and IDCT are given by the formulae (1) and (3) respectively.

$$X(n, m) = \frac{1}{\sqrt{2N}} C(n) * C(m) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x(i, j) \times \cos\left(\frac{(2i+1)n\pi}{2N}\right) \cos\left(\frac{(2j+1)m\pi}{2N}\right) \quad (1)$$

with $n, m = 0, 1, \dots, N-1$

$$C(n) * C(m) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } n, m = 0 \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

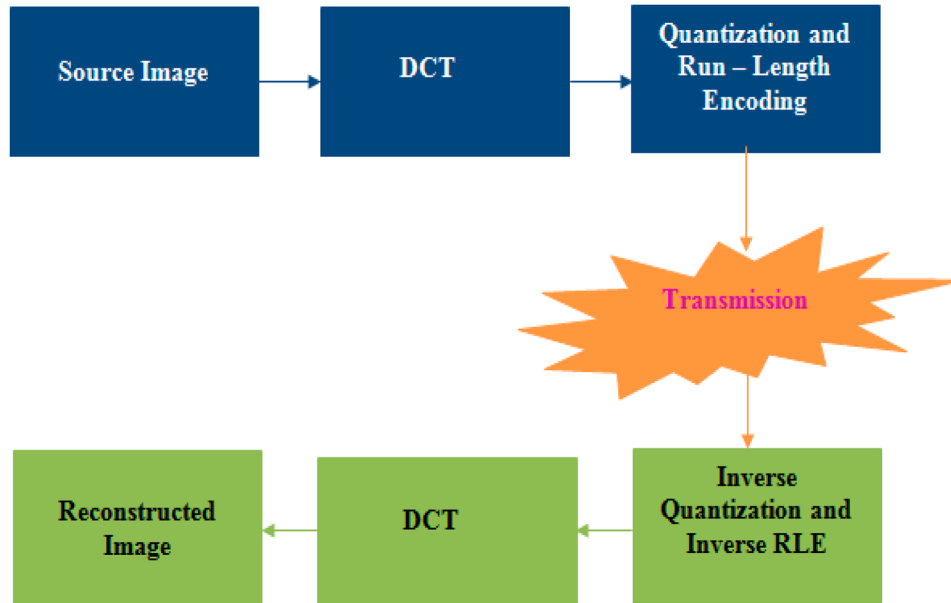


Figure 1. DCT block diagram.

Theoretically to reduce the number of operations, a 2D-DCT computation is done by using two separate 1-D DCTs, by utilizing DCT's separability characteristic. First, the transformation of 1-D DCT occurs over each row to obtain the coefficients 1-D DCT. The calculated data is further applied with 1-D DCT to provide the 2-D DCT coefficients as in (4), where X is an image block of size 8×8 , D is the 8×8 DCT matrix, R is the domain transformation obtained from 8×8 blocks and "T" signifies the transpose matrix operation (Figure 1).

$$R = D.X.D^T \quad (4)$$

The D matrix is orthogonal, i.e. $D^{-1} = D^T$. Therefore, the inverse 2-D DCT is expressed in (5) as

$$X = D^T.R.D \quad (5)$$

The above decomposition decreases the complexity of computation from $O(N^4)$ to $O(2N^3)$ multiplications and $O(N^4)-O(N^2)$ to $O(2N^3)-O(2N^2)$. For example, with the value of $N = 8$, the number of additions reduces from 4096 to 1024 and the number of multiplications reduces from 4032 to 896, indicating at least a fourfold decrease in the number of operations. The general concept of DCT process is such that the source image is captured, applied with the DCT technique, and the output is quantized and then encoded using the run-length encoding technique which generates a stream of bits to be transmitted. At the received end, the reverse processes are carried out like inverse run-length encoding, inverse quantization, inverse DCT and the image is then reconstructed which almost looks like the original source image. But DCT corresponds to a lossy compression technique and hence the exact replica of the source image cannot be obtained which can be validated using the performance parameters like PSNR, CR, SSIM and MSE.

The DCT transform utilizes considerable energy in the processing steps involved for JPEG or any other image formats. Hence the need for appropriate methods to save time and energy is required by reducing the numerical complexity. Several techniques to minimize DCT complexity are in use and these techniques are broadly classified into two methodologies like Fast DCT and Zonal DCT. The requirement for effective implementation of this algorithm is the number of operations is reduced by assessing only a segment of the DCT coefficients by exploiting interpixel redundancy and at the same time provides effective reconstruction at the receiving end without any loss of information, generally, the lower frequency coefficients as in Figure 2 and they are chosen based on how the transform matrix "C" is modified. A slight variation in the transform matrix called pruned approach will reduce the number of computations accordingly and an efficient transform matrix can be utilized in DCT computations.

A pruned 2-D DCT computes only a segment of DCT coefficients of order $N \times N$ with $N = 8$. A pruned approach can be used both with a square and triangular-shaped area with the number of coefficients given by K^2 and $K \times (K + 1)/2$ respectively with $N > K$. Among the two forms, the square-shaped area provides a better balance of energy distortion which computes only the lower frequency coefficients. The number of computations can be reduced effectively by pruning procedure.

4. Proposed algorithm – DCT with modified transform matrix

The transform matrix in DCT plays a major role in reducing the number of bits to be transmitted after compression in which it exhibits strong energy

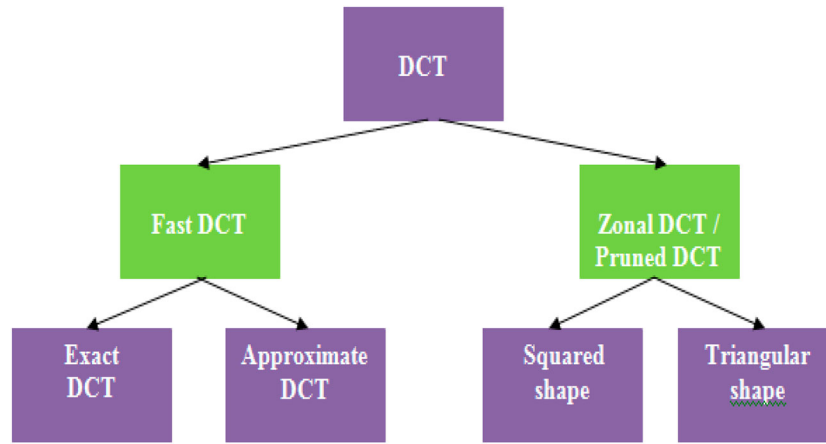


Figure 2. DCT optimization techniques.

compaction and the significant energy of an image is accumulated only on a few coefficients. As many of the higher frequency coefficients are 0, and only the low frequency pixels are concentrated in the upper portion after the 2D-DCT operation, here, the transform matrix is modified with 0s and 1/2s in appropriate positions by exploiting the redundancy between pixels and computing only a subset of coefficients. Based on this modification, the resultant transform is given by (6) which results in 12 additions and no shift operations. As shift operation implies multiple repeated bit shift operations depending on number of bit shifts, the complexity is also reduced further by modifying the transform matrix as given by (6), (7) and (8).

$$D = C.T \tag{6}$$

where C is the diagonal matrix and T is the transform matrix given by (7) and (8)

$$C = \text{diag} \left(\frac{1}{2\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}} \right) \tag{7}$$

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \tag{8}$$

This is compared with the state of the art Bouguezel-Ahmed-Swamy (BAS-2008) [9] DCT process, and Pruned BAS – 2008 (P.BAS – 2008) which used the C matrix and the transform matrix “ T ” as mentioned in (9), (10) and (11), (12) respectively, that produces

$$C = \text{diag} \left(\frac{1}{2\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}} \right) \tag{9}$$

The DCT operation using Equations (9) and (10) makes in 18 additions and 2 shift operations and the DCT operation using Equations (11) and (12) produces 14 additions and 1 shift operation. The complexity is further reduced in the proposed method by the modified

values of the transform matrix given by (7) and (8) that results in 12 additions and no shift operations.

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 0 & 0 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} & -\frac{1}{2} & 1 & -1 & \frac{1}{2} \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{10}$$

$$C = \text{diag} \left(\frac{1}{2\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}} \right) \tag{11}$$

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \tag{12}$$

The signal flow graph for the proposed algorithm and its comparison with the existing algorithms are shown in Figures 3–5 respectively.

The proposed model is shown in Figure 6. Basically, a DCT involves a DCT matrix, quantization, and run-length encoding. In the proposed model, a Cohen–Daubechies–Feauveau (CDF) filter which is a set of analysis filters and synthesis filters for decomposition and reconstruction respectively is also included. CDF filter possesses good compression properties and it is included to minimize the number of bits further to improve network lifetime with respect to the nodes.

The proposed model operation is such that the image is captured by a sensor node and processing of the same is done with the modified DCT algorithm. Here for the experimental purposes a Raspberry Pi 4 board fitted with camera modules is used for image capturing. Raspberry Pi can be customized and programmable based on the requirements and is cheaper with an extendable memory capacity compared to the wireless sensor

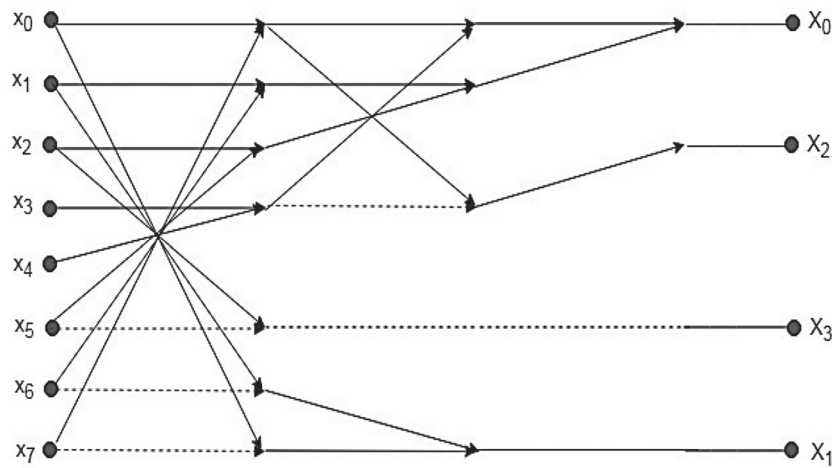


Figure 3. Modified pruned DCT with 12 additions, no shift operations.

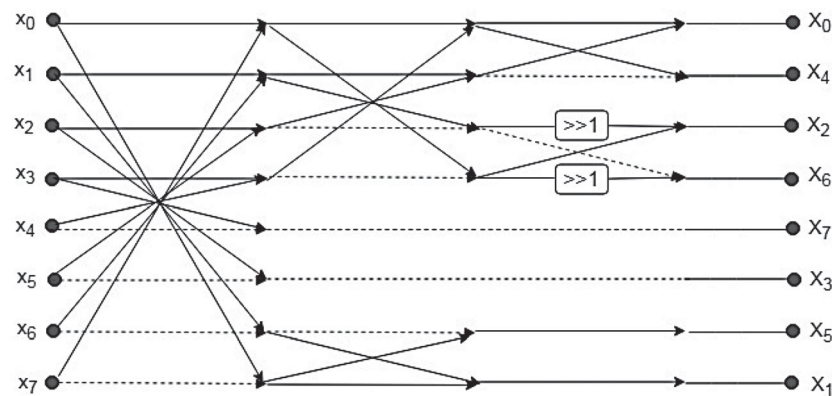


Figure 4. DCT with 18 additions, 2 shift operations.

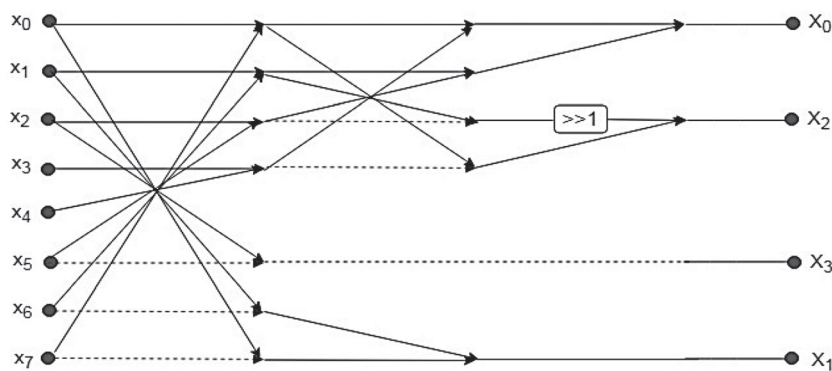


Figure 5. DCT with 14 additions, 1 shift operations – pruned DCT.

nodes like MicaZ, TelosB, Iris, etc. [13]. A Raspberry Pi 4 can also be deployed as a WSN gateway with a minimum power consumption profile [14]. The image is captured and applied with modified DCT. For modified DCT, the transform matrix is modified as in (10) with the elimination of fractional values in the transform matrix. This is done to avoid the bit shift operations, as the bit shift operations consume energy depending on the number of bit shift operations. The 2D-DCT operation is performed based on (6) and its output is then quantized. The quantization process quantizes the image coefficients into the nearby quantization levels.

The quantization step introduces quantization error, and hence the step size must be chosen appropriately so as not to produce so much loss in the reconstructed image.

The resultant coefficients are further processed by a 9/7 CDF filter [15] which is a wavelet filter normally used by JPEG images for limiting the amount of data and CDF filter proves to have very good multiresolutional analysis. CDF filter consists of a set of decomposition low-pass (LP) and high-pass (HP) analysis filters and reconstruction low-pass (LPR) and high-pass (HPR) filters synthesis filters. The filter coefficients are

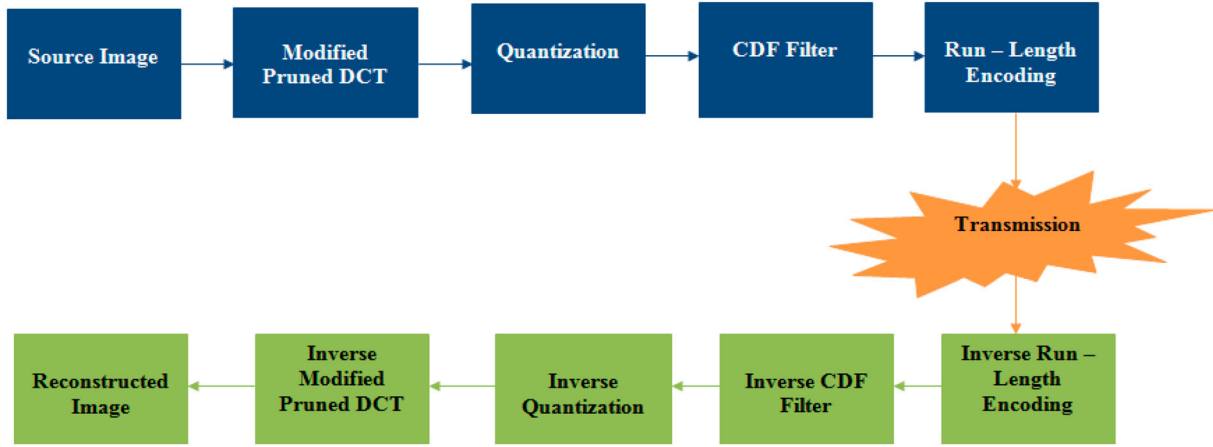


Figure 6. Proposed model.

adjusted to provide lossless reconstruction but with a reduction in the data size. The filter coefficients used are as shown in (11), (12), (13) and (14) for the analysis and synthesis filters respectively.

$$LP = \begin{bmatrix} 0.026748757411, -0.016864118443, \\ -0.07822326652, 0.266864118443, \\ 0.602949018236, 0.266864118443, \\ -0.07822326652, \\ -0.016864118443, 0.026748757411 \end{bmatrix} \quad (11)$$

$$HP = \begin{bmatrix} 0.045635881557, -0.028771763114, \\ -0.295635881557, 0.557543526229, \\ -0.295635881557, -0.028771763114, \\ 0.045635881557 \end{bmatrix} \quad (12)$$

$$LPR = \begin{bmatrix} -0.045635881557, -0.028771763114, \\ 0.295635881557, 0.557543526229, \\ 0.295635881557, -0.028771763114, \\ -0.045635881557 \end{bmatrix} \quad (13)$$

$$HPR = \begin{bmatrix} 0.026748757411, 0.016864118443, \\ -0.07822326652, -0.266864118443, \\ 0.602949018236, -0.266864118443, \\ -0.07822326652, \\ 0.016864118443, 0.026748757411 \end{bmatrix} \quad (14)$$

The output of the CDF filtering process is then passed through a Run Length Encoded (RLE) step to produce a stream of bits to be transmitted through a communication channel. RLE is a lossless encoding procedure that encodes the data bits from the CDF filter stage into runs of 1s and 0s as a (data value, count) pair. After CDF filtering, the image coefficients are represented as 1s and 0s with many of the coefficients as 0s. This can be represented in RLE format and transmitted.

For reconstruction of the image in the receiver end, inverse RLE, Inverse CDF filtering, Inverse quantization, and Inverse DCT are done and the reconstructed image is analysed with PSNR, MSE, CR, SSIM, etc.

4.1. Procedure for the proposed process

4.1.1. Compression

1. Decompose the image into 8×8 blocks
2. For the 1-D DCT, Multiply the 4×8 transform matrix with 8×8 image blocks with which results in a 4×8 output
3. Take transpose of the transform matrix
4. Multiply the output from step 2 with the transposed matrix from step 3 to obtain the 2D-DCT output of 4×4 image elements
5. Quantize the coefficients obtained from step 4
6. The output from step 5 is passed through a CDF9/7 filter
7. CDF9/7 filter performs convolution of the low pass, high pass coefficients along horizontal, vertical directions and the output is in the form of LL, LH, HL, and HH sub-bands
8. Encode the output obtained from step 7 using RLE.

4.1.2. Decompression

The final image can be reconstructed via reverse RLE, applying reverse CDF 9/7 filtering and reverse 2D-DCT coefficients in all stages. Figure 7 shows the process as depicted in Sections 4.1.1 and 4.1.2.

5. Performance metrics

The compression algorithm is tested on real-time images of concrete walls of buildings captured using Raspberry pi 4 module integrated with the camera. This setup will be very much helpful in the monitoring of heritage buildings. All the images are of size 512×512 and a comparison of the suggested method to existing

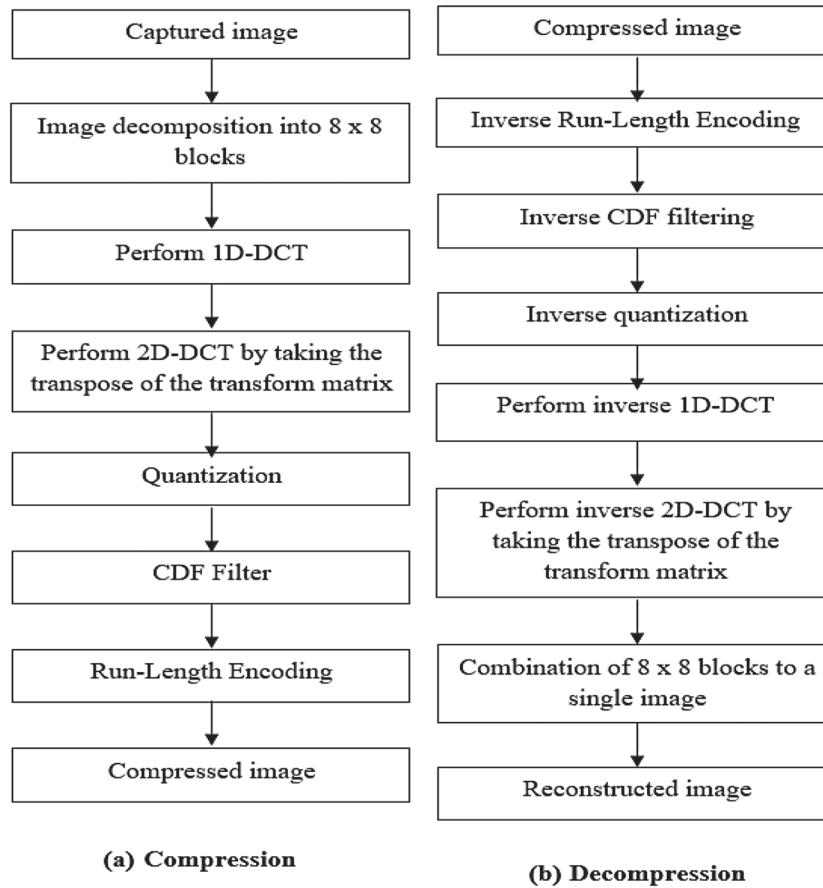


Figure 7. Flowchart of the proposed model.

methods of DCT is established based on different performance metrics. The standard concrete crack images dataset [16] is also analysed.

5.1. Peak signal to noise ratio and mean square error

To determine the accuracy, PSNR and MSE are the most widely used parameters, where $X(i,j)$ is the actual image before compression and $Y(i,j)$ is the recovered image after decompression. The absolute error between the original image and the reconstructed image is based on PSNR and MSE. PSNR displays the accuracy of the reconstructed image based on the MSE of the reconstructed image as given by (15) and represents the maximum error between the original image and the reconstructed image. Numerator term in (15) represents an image's maximum pixel value and the denominator term is the MSE that represents the collective error and the image size is represented by $m \times n$ in the MSE term. PSNR indicates that the higher the PSNR value, the lower the MSE (less error) value, which in turn provides better reconstruction.

$$PSNR = 20 \log_{10} \frac{255}{\sqrt{\frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (X(i,j) - Y(i,j))^2}} \quad (15)$$

5.2. Compression ratio

Compression Ratio is the proportion of the size of the actual image to the compressed image. The compressed image size shall be computed after compression using the formula as given in (16). The greater the CR value, the more the picture gets compressed.

$$\begin{aligned} \text{Compression Ratio (CR)} \\ = \frac{\text{Size of the Original Image}}{\text{Size of the Compressed Image}} \end{aligned} \quad (16)$$

Based on the value of CR from (16), the space savings is given by (17) as

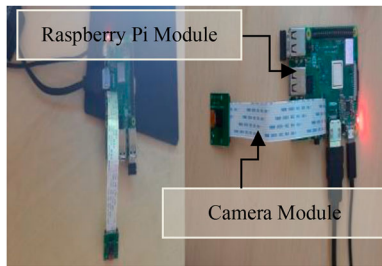
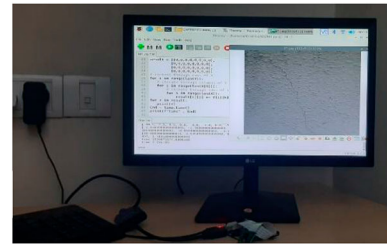
$$\text{Percentage of Space Savings} = \left(1 - \frac{1}{CR}\right) * 100 \quad (17)$$

5.3. Experimental setup and complexity analysis of various DCT methods

Table 1 shows the total number of adds and multiplications that must be computed for various DCT methods. The consolidation is done based on the number of operations for 1D as well as 2D-DCTs and the proposed method has a lesser number of operations when compared to the various methods. The real-time images are captured using a Raspberry pi 4 module fitted with the camera as shown in Figures 8–10, and the same module is used to run the algorithm to evaluate the time

Table 1. Complexity analysis of various DCT methods.

Method	1-D			2-D		
	Multiplications	Additions	Shift	Multiplications	Additions	Shift
Standard DCT	64	56	0	1024	896	0
BAS-2008	0	18	2	0	288	32
BAS-2013	0	24	0	0	384	0
RDCT	0	22	0	0	352	0
MRDCT	0	14	0	0	224	0
P. BAS-2008	0	14	1	0	168	12
Proposed method (modified pruned DCT)	0	12	0	0	144	0

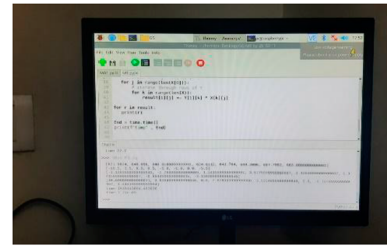
**Figure 8.** Raspberry Pi 4 setup.**Figure 9.** Transmitter output.**Table 2.** Code size and processing time for a 8×8 block.

Method	Code size (bytes)	Processing time (μ s)	Energy consumption (μ J)
BAS-2008	1994	173.25	3.81
BAS-2013	2278	192.75	4.24
RDCT	2102	181.5	3.99
MRDCT	1756	161.25	3.65
P. BAS-2008	1250	114.88	2.65
Proposed method (modified pruned DCT)	745	72.7	1.65
Proposed method with CDF	837	79.75	1.8
Proposed method with CDF and RLE	924	89.89	1.95

taken to compute 8×8 pixel blocks of an image and the code size thus obtained also shows an appropriate validation of the proposed algorithm as shown in Table 2. In Table 2, the code size, processing time, and energy consumed are calculated for a block of DCT and it is found that the values obtained are even less for the combined methods of DCT, CDF, and RLE when compared to the P.BAS-2008.

6. Simulation results and analysis

Images of size 512×512 are captured using the Raspberry Pi camera module and the algorithm's efficiency has been tested and Figure 11 indicates the set of images captured for which the simulation and testing were conducted. The algorithm is tested for the standard concrete image set also [16]. The sample simulation results obtained from the modified algorithm for the captured images are shown in Figure 12–17 which indicates the (a) original image, (b) DCT output, (c) CDF filter output and (d) reconstructed image at the receiving end.

**Figure 10.** Output screen.

The PSNR and MSE values closely approximate to the existing methods that have been compared with as the change after the DCT compression is very minimal as represented in Figures 18 and 19.

Table 3 shows the comparison of PSNR, MSE and SSIM between the two algorithms and shows increase in PSNR. The MSE value differs proportionately on the basis of PSNR which indicates the higher values of PSNR correspond to lower MSE values. The Structural Similarity Index (SSIM) of the images that are reconstructed at the receiver end, SSIM indicates the similitude between the actual image and the reconstructed image with a value between 0 and 1 with 1 relates to 100% similarity and 0 relates to no similarity. SSIM exploits the interdependencies among the pixels and quantifies the image quality degradation induced by processing images and data during compression and the losses using this perceptual metric. The plots of PSNR, MSE, and SSIM are done using Python software which is shown in Figures 20–22 respectively, and the values obtained are better when compared to the existing method. Furthermore, the suggested method is also validated in terms of rate distortion curve. It calculates the minimum number of bits per pixel that should be sufficient so that the input image (X) can be

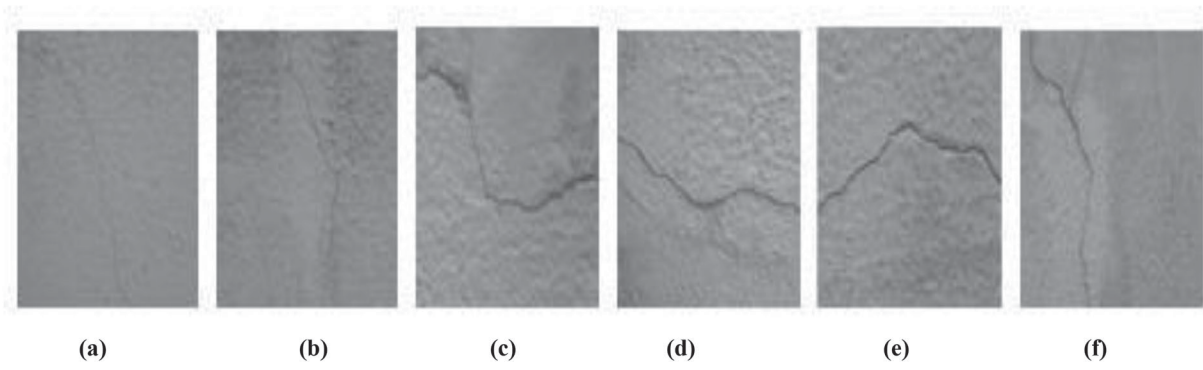


Figure 11. Image Set: (a) Image 1, (b) Image 2, (c) Image 3, (d) Image 4, (e) Image 5 and (f) Image 6.

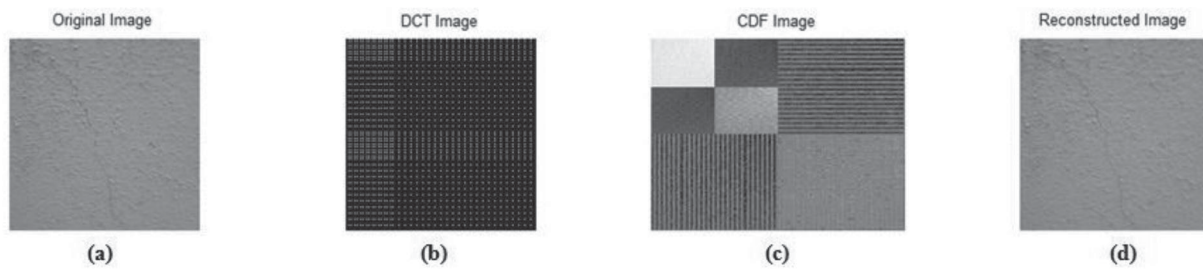


Figure 12. (a) Original Image, (b) DCT Output, (c) CDF Output and (d) Reconstructed Image.

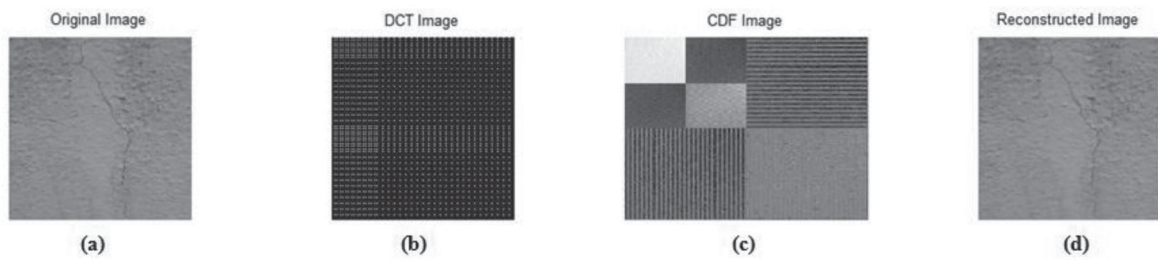


Figure 13. (a) Original Image, (b) DCT Output, (c) CDF Output and (d) Reconstructed Image.

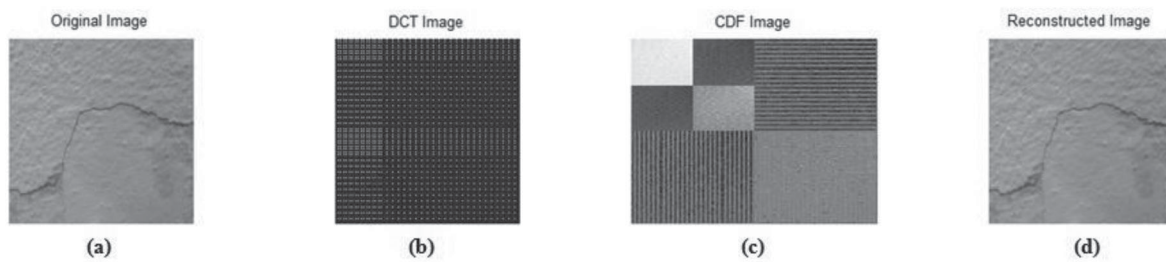


Figure 14. (a) Original Image, (b) DCT Output, (c) CDF Output and (d) Reconstructed Image.

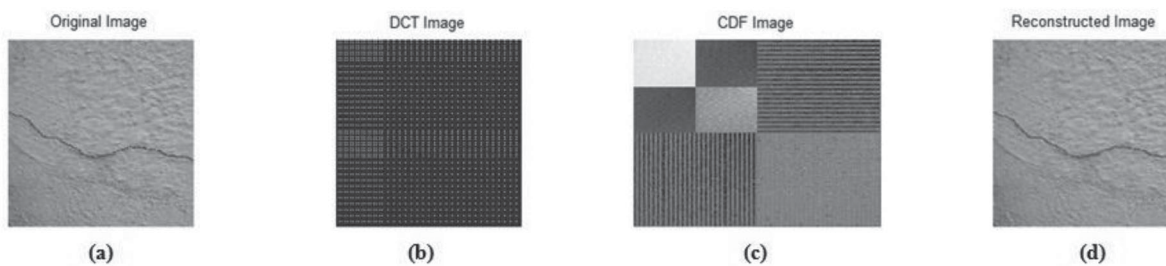


Figure 15. (a) Original Image, (b) DCT Output, (c) CDF Output and (d) Reconstructed Image.

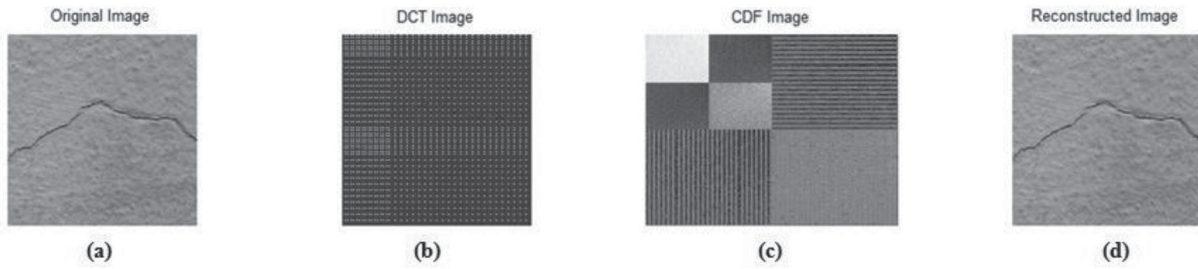


Figure 16. (a) Original Image, (b) DCT Output, (c) CDF Output and (d) Reconstructed Image.

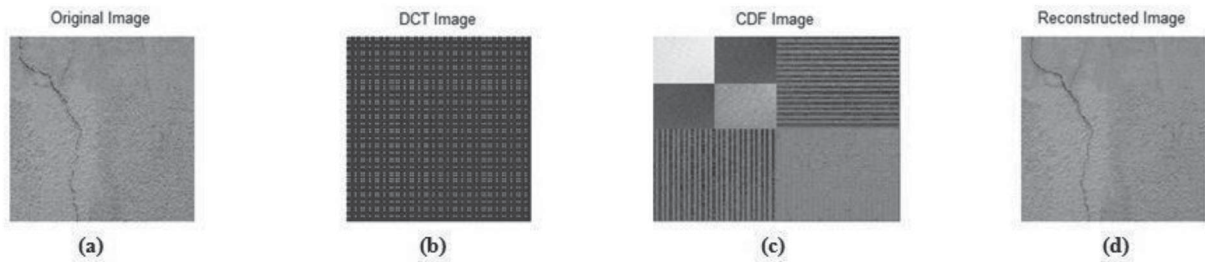


Figure 17. (a) Original Image, (b) DCT Output, (c) CDF Output and (d) Reconstructed Image.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.9062	-0.0117	0.0012	0.0059	0.9990	-0.0318	-0.0086	0.0156	1.0933	-0.0428	0	0.0186
2	-0.0221	0.0078	-2.1684e-19	-0.0055	-0.0035	0.0059	-0.0114	0	-0.0262	-9.7656e-04	8.7346e-04	0.0014
3	-6.1763e-04	0	6.2063e-18	0.0012	0.0130	0.0087	0.0023	0.0148	0.0130	8.7346e-04	-2.1684e-19	-0.0012
4	0.0107	-0.0014	0	0.0020	-0.0117	0.0124	0.0049	-0.0039	-0.0020	0.0014	0	0.0020
5	1.0000	-0.0525	0.0099	0.0244	1.0464	-0.0615	0.0395	0.0273	1.2026	-0.0228	0.0111	0.0107
6	-0.0269	0.0195	8.7346e-04	-0.0097	-6.9053e-04	-0.0059	-0.0122	-0.0110	-0.0104	-0.0117	0.0017	0.0028
7	0.0012	0.0105	-7.8125e-04	-0.0062	0.0012	-0.0026	-0.0102	-4.7705e-18	-0.0142	0.0017	-0.0117	0.0012
8	0.0254	-0.0097	-6.2884e-18	0.0059	-4.8789e-18	0.0028	0.0037	0.0059	0.0039	0.0028	6.2884e-18	-0.0020
9	1.0312	-0.0387	0.0198	0.0234	1.1392	-0.0097	0.0105	0.0039	1.2500	-0.0207	-2.7756e-17	0.0078
10	0	0	0	0	-0.0207	-0.0029	-8.7346e-04	0.0166	-0.0442	0	0	0
11	0	0	0	0	-6.1763e-04	-8.7346e-04	-3.1032e-18	-4.9065e-18	-1.4617e-17	-0.0017	6.1800e-18	9.8131e-18
12	0	0	0	0	0.0088	0.0028	-6.5052e-18	-0.0098	0.0176	0	1.3010e-17	0
13	1.1104	-0.0414	0.0093	0.0186	1.2485	-0.0421	0.0093	0.0176	1.3428	-0.0532	0.0179	0.0225
14	-0.0435	0	-8.7346e-04	0	-0.0214	-0.0029	0.0017	0.0152	-0.0249	-9.7656e-04	-0.0052	-0.0014
15	-1.2766e-20	-1.0408e-17	-7.8125e-04	0.0012	-0.0247	0.0044	-7.8125e-04	-0.0210	-0.0117	-8.7346e-04	-7.8125e-04	0
16	0.0195	-0.0014	-0.0025	0.0020	0.0098	0.0028	0.0012	-0.0078	-0.0020	-0.0028	0.0185	0

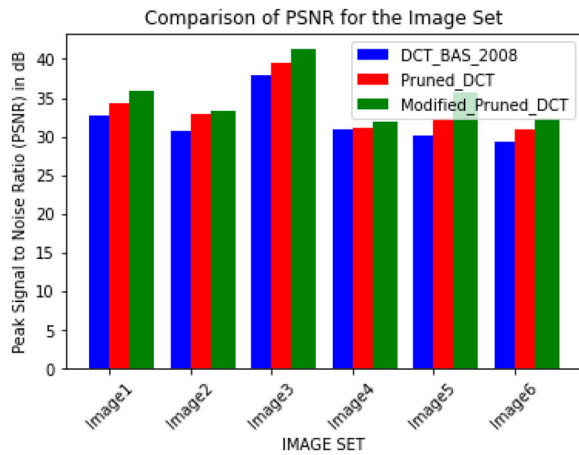
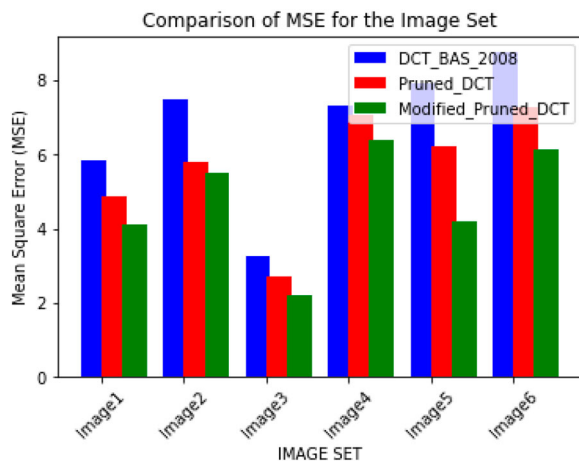
Figure 18. Coefficients obtained from modified pruned DCT.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.9062	-0.0117	0.0012	0.0059	0.9990	-0.0318	-0.0111	0.0156	1.0933	-0.0428	3.0882e-04	0.0186
2	-0.0221	0.0078	0	-0.0055	-0.0035	0.0059	-0.0140	0	-0.0262	-9.7656e-04	4.3673e-04	0.0014
3	-3.0882e-04	4.3673e-04	-5.8594e-04	0.0019	0.0151	0.0105	0.0027	0.0173	0.0154	4.3673e-04	-3.9062e-04	-0.0012
4	0.0107	-0.0014	-6.1763e-04	0.0020	-0.0117	0.0124	0.0062	-0.0039	-0.0020	0.0014	3.1442e-18	0.0020
5	1.0000	-0.0525	0.0124	0.0244	1.0464	-0.0615	0.0466	0.0273	1.2026	-0.0228	0.0139	0.0107
6	-0.0269	0.0195	-8.6736e-19	-0.0097	-6.9053e-04	-0.0059	-0.0144	-0.0110	-0.0104	-0.0117	0.0022	0.0028
7	0.0012	0.0122	-0.0014	-0.0074	3.0882e-04	-0.0026	-0.0160	-4.7705e-18	-0.0173	0.0013	-0.0178	0.0012
8	0.0254	-0.0097	-6.0715e-18	0.0059	-4.8789e-18	0.0028	0.0043	0.0059	0.0039	0.0028	9.4326e-18	-0.0020
9	1.0312	-0.0387	0.0247	0.0234	1.1392	-0.0097	0.0124	0.0039	1.2500	-0.0207	-2.7756e-17	0.0078
10	0	0	0	0	-0.0207	-0.0029	-4.3673e-04	0.0166	-0.0442	0	0	0
11	0	0	0	0	-6.1763e-04	-4.3673e-04	1.9531e-04	7.3598e-18	-4.8718e-18	-0.0017	1.7347e-18	0
12	0	0	0	0	0.0088	0.0028	-6.1763e-04	-0.0098	0.0176	0	-0.0012	0
13	1.1104	-0.0414	0.0114	0.0186	1.2485	-0.0421	0.0111	0.0176	1.3428	-0.0532	0.0225	0.0225
14	-0.0435	0	-8.7346e-04	0	-0.0214	-0.0029	0.0013	0.0152	-0.0249	-9.7656e-04	-0.0070	-0.0014
15	9.5654e-20	-1.2143e-17	-3.9062e-04	0.0012	-0.0306	0.0052	-0.0018	-0.0253	-0.0139	-8.7346e-04	-0.0018	-6.1763e-04
16	0.0195	-0.0014	-0.0025	0.0020	0.0098	0.0028	0.0019	-0.0078	-0.0020	-0.0028	0.0229	0

Figure 19. Coefficients obtained from pruned DCT.

Table 3. Comparison of PSNR, MSE and SSIM for the image set.

Image name	PSNR			DCT – BAS-2008	MSE		DCT – BAS-2008	SSIM	
	DCT – BAS-2008	Pruned DCT	Modified pruned DCT		Pruned DCT	Modified pruned DCT		Pruned DCT	Modified pruned DCT
Image 1	32.79	34.36	35.85	5.85	4.88	4.11	0.86	0.92	0.95
Image 2	30.64	32.86	33.34	7.49	5.80	5.49	0.84	0.87	0.92
Image 3	37.81	39.45	40.26	3.28	2.72	2.21	0.89	0.92	0.96
Image 4	30.86	31.13	32.00	7.30	7.08	6.41	0.80	0.83	0.89
Image 5	30.18	32.27	33.66	7.90	6.21	4.20	0.90	0.94	0.96
Image 6	29.29	30.91	31.40	8.75	7.26	6.12	0.91	0.96	0.98

**Figure 20.** Comparison of PSNR.**Figure 21.** Comparison of MSE.

reproduced image at the receiver (Y) efficiently. The Rate distortion curves for a specific set of images are compared in Figure 23 in terms of PSNR, which also depicts the effectiveness of the suggested algorithm, with the increase in bpp, results in an increase in PSNR and considerable PSNR improvement in the proposed method.

Table 4 shows the number of bits to be transmitted from the source node to the receiving end and the comparison shows proves that Pruned DCT with Modified Transform Matrix with CDF and RLE is best suited for sensor nodes as the memory requirement is very much less when compared to the other cases. The representation is shown in Figure 24.

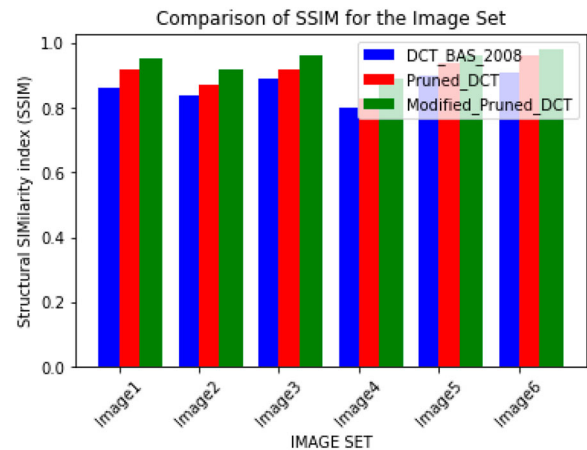
**Figure 22.** Comparison of SSIM.**Table 4.** Comparison of bit count for the image set with various DCT algorithms.

Image	Actual image (before compression)	DCT – BAS-2008	Pruned DCT	Modified pruned DCT with CDF and RLE
Image 1	335052	77198	76120	5552
Image 2	626688	88930	80685	8974
Image 3	601292	94160	85960	8048
Image 4	697958	69234	66814	6495
Image 5	491520	95680	90700	7515
Image 6	472156	90457	87054	5152

7. Conclusion

Real-time capturing and transmission of images and its subsequent storage is a prime concern, as it is data intensive. The proposed algorithm helps to reduce the data size by effective compression and also enables the images to be reproduced efficiently at the receiving end. Due to the very high compression ratio achieved, the lifetime of the sensors is also enhanced as the image size and the numbers of bits are reduced before the transmission which relates to the removal of redundant information. While having good compression ratio, the PSNR is also proved to be high and high SSIM value which indicates the structural similarity of both the captured image at the transmitting end and the reproduced image at the receiving end. Even though the CDF filter is an additional block, combined with RLE, it provides a considerable decrease in the number of bits to be transmitted, by which the time required for the operation of this filter is considerably compensated and make

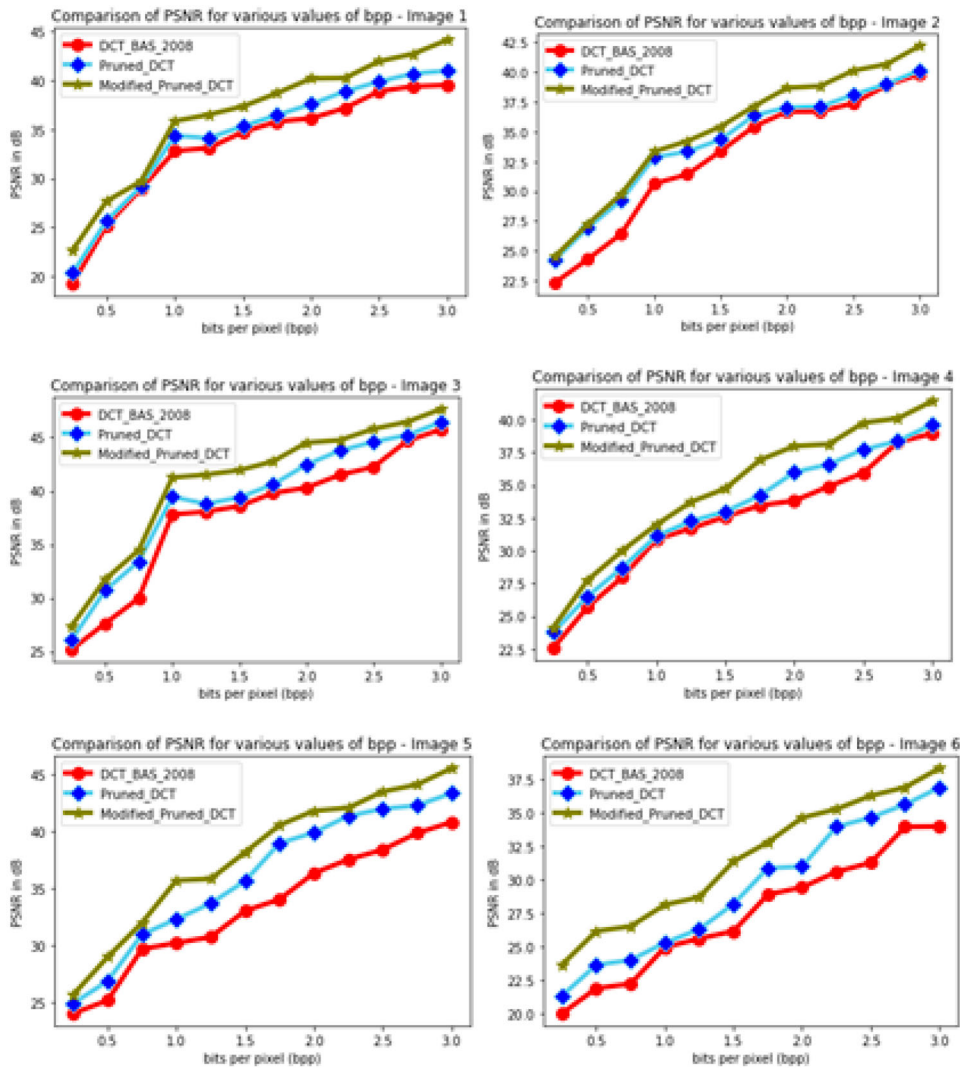


Figure 23. Comparison of rate distortion curves.

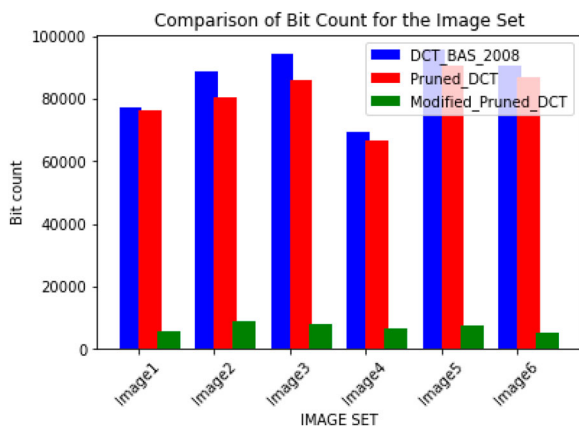


Figure 24. Comparison of bit count.

it suitable for implementing this algorithm in a network of nodes.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

- [1] Akyildiz F, et al. A survey on wireless multimedia sensor networks. Elsevier Comput Netw J. 2007;51(4):921–960. doi:10.1016/j.comnet.2006.10.002
- [2] Ma T, Hempel M, Peng D, et al. A survey of energy-efficient compression and communication techniques for multimedia in resource-constrained systems. IEEE Commun Surv Tutor. 2012;15(3):963–972. doi:10.1109/SURV.2012.060912.00149
- [3] Mammari A, et al. A survey of compression algorithms for visual sensor networks. Int Scholarly Res Netw, ISRN Sens Netw. 2012;2012:Article ID: 760320. doi:10.5402/2012/760320
- [4] Gao R, Wu H, Cheng C. Image compression algorithm for WSN. Adv Inf Sci Serv Sci (AISS). 2013;5(9).
- [5] Eisa SA, Lotfy W. Image compression using two dimensional DCT and least squares interpolation. Wirel Pers Commun. 2017;97(3):4773–4787. doi:10.1007/s11277-017-4750-x
- [6] Kumar UA, et al. Evaluation of multiplier-less DCT transform using in-exact computing. Second International Conference (MIND 2020), Springer; 2020.
- [7] Almurib HAF, et al. Approximate DCT image compression using inexact computing. IEEE Trans Comput. 2018;67(2):149–159. doi:10.1109/TC.2017.2731770

- [8] Cintra RJ, et al. Low-complexity 8-point DCT approximations based on integer functions. *Sig Process.* 2014;99:201–214. doi:10.1016/j.sigpro.2013.12.027
- [9] Bouguezel S, Ahmad MO, Swamy MNS. A low-complexity parametric transform for image compression 2011 IEEE International Symposium of Circuits and Systems (ISCAS). Brazil; 2011. p. 2145–2148. doi:10.1109/ISCAS.2011.5938023
- [10] Potluri US, Madanayake A, Cintra RJ, et al. Improved 8-point approximate DCT for image and video compression requiring only 14 additions. *IEEE Trans Circuits Syst Regul Pap.* 2014;61(6):1727–1740. doi:10.1109/TCSI.2013.2295022
- [11] Makkaoui L, et al. Fast zonal DCT-based image compression for wireless camera sensor networks. 2nd International Conference on Image Processing Theory Tools and Applications (IPTA), 126–129; 2010.
- [12] Araar C, et al. Low complexity image compression using pruned 8-point DCT approximation in wireless visual sensor networks. 2017 International Conference on Mathematics and Information Technology (ICMIT); 2017. <http://doi.org/10.1109/MATHIT.2017.8259730>
- [13] Vujovic V, Maksimovic M. Raspberry Pi as a wireless sensor node: performance and constraints. 2014 37th International Convention in Information and Communication Technology, Electronics and Microelectronics (MIPRO), 1013–1018; 2014. <https://doi.org/10.1109/MIPRO.2014.6859717>
- [14] Astudillo-Salinas F, et al. Minimizing the power consumption in Raspberry Pi to use as a WSN gateway. 8th IEEE Latin-American Conference on Communications (LATINCOM); 2016.
- [15] Sudha G, Tharini C. Modified contourlet transform based effective image compression framework for wireless sensor networks. *J Adv Res Dyn Control Syst.* 2018;10(01). Special Issue. <https://www.jardcs.org/backissues/abstract.php?archiveid=2675>.
- [16] Concrete Crack images for classification dataset <https://data.mendeley.com/datasets/5y9wdsg2zt/2>.