IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 1 November 2023, accepted 21 November 2023, date of publication 23 November 2023, date of current version 29 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3336403

RESEARCH ARTICLE

# Optimized Block-Based Lossy Image Compression Technique for Wireless Sensor Networks

**BOSE A. LUNGISANI, (Member, IEEE), ADAMU M. ZUNGERU, (Senior Member, IEEE), CASPAR K. LEBEKWE, (Member, IEEE), AND ABID YAHYA, (Senior Member, IEEE)**
Department of Electrical, Computer and Telecommunications Engineering, Botswana International University of Science and Technology, Palapye, Botswana

Corresponding author: Adamu M. Zungeru (zungerum@biust.ac.bw)

This work was supported by the Office of Research, Development, and Innovation (ORDI), Botswana International University of Science and Technology, under Grant S00174.

**ABSTRACT** Traditionally, image compression algorithms have primarily focused on optimizing storage without considering resource-constrained applications, such as wireless sensor networks (WSNs). However, for practical application in WSNs, a balanced trade-off between compression ratio, distortion, and energy consumption is crucial to improve the network lifetime while maintaining acceptable image reconstruction quality at lower bit and error rates. Previous studies have focused on higher image compression rates to address storage limitations rather than on the optimization of WSNs. In addition, most previous research on image compression for WSNs either requires an error-bound mechanism or compromises the trade-off between compression ratios and reconstructed image quality measured using image quality assessment metrics, such as root mean square error (RMSE), and coefficient of determination ($R^2$), leading to a reduced network lifetime and uncontrolled reconstructed image quality that is not application-specific. Therefore, we present an optimized block-based image compression algorithm for WSNs with a relative error-bound mechanism that adapts to a given dataset to improve reconstruction fidelity and energy consumption at higher compression ratios. A comparison of our proposed algorithm with existing algorithms demonstrated that using a convolutional variational autoencoder and relative error-bound mechanism leads to a significant trade-off in distortion, compression ratio, and energy consumption in WSNs. Our results demonstrated that an average reconstruction fidelity of more than 90% was achieved using image quality evaluation metrics at compression ratios of 60% or more. Furthermore, more than 50% energy conservation at compression ratios greater than 60% from image compression was achieved compared with the transmission of raw data within a WSN.

**INDEX TERMS** Image compression, neural networks compression, reconstruction fidelity, wireless sensor networks.

## I. INTRODUCTION

The research focus areas in sensor networks include compression, bandwidth management, storage, data transmission, energy conservation, and network interference. The lifetime of a wireless sensor network (WSNs) depends on the battery lifetime of the sensor nodes. Therefore, minimizing the power consumption of sensor nodes has become a research focus to ensure a continuous communication lifetime between the transmitter and receiver nodes. This is achieved via image compression, which involves quantizing the image data. However, quantization leads to lossy compression, which affects the quality of the reconstructed images. Therefore, the level of compression for energy conservation and distortion of the reconstructed image forms the basis for optimization in image compression. Several optimized compression algorithms have been developed based on their achievements and defects [1]. Traditionally, the pipeline for image-compression transformation consists of quantization, transformation, and entropy coding. Image compression transformation involves the transformation of an image signal into decorrelated coefficients. The Discrete Wavelet Transform (DWT) [2] and

The associate editor coordinating the review of this manuscript and approving it for publication was Chan Hwang See.

Discrete Cosine Transform (DCT) [3] form the introductory part of traditional image transformation mechanisms researched in the literature. Less critical information is discarded in quantization according to the vector coefficients. Entropy coding compresses decorrelated coefficients through different coding systems, such as Huffman and arithmetic.

Wireless sensor networks have very limited network lifetimes owing to their power. In a camera equipped with a wireless sensor node, the sensor network lifetime is quickly reduced owing to the constraints of the processing power and battery during image processing and transmission to the destination. However, image compression not only assists in reducing the communication latency but also the energy consumption efficiency. In addition, previous studies on image compression for WSNs have reviewed, surveyed, and ranked image compression techniques, emphasizing shortcomings in areas such as energy efficiency, reconstructed image quality, power consumption, processor performance, and compression ratio. Therefore, investigations have shown that image-compression techniques can be applied to WSNs. However, most of these algorithms must address the trade-off between image reconstruction error, compression ratio, and energy consumption. Those that address compression ratios and energy consumption often compromise image quality based on different applications in WSNs. Therefore, this research intends to address reconstructed image quality while utilizing high compression ratios on resource-constrained WSNs' sensor nodes by applying techniques that do not compromise network lifetime and performance through energy consumption reduction within a wireless sensor network.

In WSNs, lossy image compression compromises the reconstructed image quality owing to image quantization, which creates distortion. In addition, there is a need to balance the energy conservation from lower bit rates during transmission and the reconstructed image quality at the receiver sensor nodes. As highlighted before, previous studies have concentrated more on image compression for WSNs [4], [5], [6] with either fixed to no error-bound mechanisms or similarities of images based on pixel-wise comparisons of whole images [1], [7] that compromise the training set efficiency of autoencoders. Although literature has shown that adopting autoencoders for image compression is helpful for dimensionality reduction, exploiting this type of algorithm using convolutional variational autoencoders in WSNs is still in its infancy [8].

To address these shortcomings, we developed a block-based image compression algorithm for WSNs based on a convolutional variational autoencoder (VAE) [9], [10] with a relative error bound mechanism [11], [12], [13] that can adapt to the image input dataset to control the image reconstruction error rate. However, the effectiveness of artificial neural networks (ANNs) [14] based on autoencoders depends on their training efficiency. Therefore, to improve the training efficiency and avoid overfitting or underfitting the network [15], the input images were divided into blocks [16] of equal size before being fed to the autoencoder as input image data. Furthermore, to achieve constrained optimization between the bit rate and distortion on the two probability distributions, the constrained loss function is calculated from the bit rate, distortion function, and Lagrange multiplier.

To the best of our knowledge, there is no problem formulation for constrained rate-distortion in image compression for wireless sensor networks based on image blocks and relative error bound mechanisms. Therefore, we consider image compression using a convolutional variational autoencoder for dimensionality reduction of image data before transmission, a relative error-bound mechanism to guarantee image quality at acceptable distortion rates between the reconstructed image and the input image. The contributions of this study are as follows.

- The rate and distortion problems of image compression for WSN were formally introduced and formulated.
- We introduce a block-based lossy image compression algorithm that balances the bit rate and distortion using a relative error-bound mechanism on the latent distribution loss and reconstruction loss of a VAE neural network.
- We compare our proposed algorithm with other existing works on image compression.
- We provide a trade-off for rate distortion above 90% when measured using multiscale structural similarity index method (MS-SSIM) and peak signal-to-noise ratio (PSNR).
- Finally, we analyzed energy consumption through image compression with raw data transmission through different hop counts and compression rates.

The rest of the paper is organized as follows: In Section II, related works are discussed; In Section III, the problem is formulated; in Section IV, the proposed solution is presented. An evaluation of the proposed solution is presented in Section V. Section VI provides the conclusions of the research work.

## II. RELATED WORK

Tremendous progress has been made in utilizing machine-learning techniques in image-compression algorithms. Some of the achievements in image compression are discussed: the authors in [17] varied the compression rate using a long short-term memory (LSTM) convolutional neural network (CNN). This was considered a breakthrough in the literature and formed the basis for image compression using neural networks. However, the algorithm did not take advantage of the spatial redundancies of the images, such as relative entropy. Soft-to-hard vector quantization was introduced [18] in non-real-time to improve the network performance through quantization. In [19], the authors developed the first technique for the compression of images in real time, using a framework for multiscale antagonistic loss. To add to the knowledge of image compression, the authors in [20] improved the performance of image compression by adopting a hyperprior. The hyperprior was used to capture the spatiality dependencies on the latent-space data representation by inter-channel reduction with a new nonlinearity to easily model spatial dependencies. Klopp et al. [20] proposed a technique to reduce redundancy in coding through spatial prediction.

In addition, despite their role in improving the quality of reconstructed images at higher compression rates, challenges in balancing compression rate and distortion persist. The authors of [21] introduced a rate-distortion generative model for image compression at low bit rates. It was assumed that the distributions of the training and reconstruction samples followed the same trend. Furthermore, the work by the authors in [22] demonstrated some promising capabilities in image compression through decorrelation, which is trainable with nonlinear normalization. A generalized divisive normalization (GDN) model was introduced in [23] to guarantee an unpartitioned multilayer for end-to-end architectures for image compression. The main objective was to improve the distortion rate in image compression applications. In [24], a local entropy model was developed to optimize latent representation encoding using offline dictionaries.

A content-weighted model was proposed with a feature for guiding latent code bit allocations through an importance map that has been learned [25]. The authors achieved an improved performance for rate distortion [26] by adopting a 3-D context and non-local model network. Moreover, the authors in [27] introduced an *iWave++* model for lossless and lossy compression optimization through wavelet-like transformations in neural networks. The Gaussian Mixture Model (GMM) proposed by the authors in [28] was developed to improve the capability of transformation and accurate estimation of symbol likelihoods. Nevertheless, efforts have been made to solve the image compression optimization problem in the literature [29], [30], [31], [32], [33] with more emphasis on achieving higher compression ratios.

## III. PROBLEM FORMULATION

A mathematical model for balancing the rate and distortion in image compression for wireless using a variational autoencoder (VAE) is presented. Variational autoencoders are generative models that learn sample distributions that are used as estimated distributions to approximate actual sample distributions. The reference is shown in Figures 1–3 for the VAE neural network. For training purposes, VAEs are categorized into three processes: encoding, sampling, and decoding, as illustrated in Figures 1–3.

*Encoder:* Compresses data by producing new features that represent the input data x through feature extraction.

*Sampler:* This forces the latent-space distribution to be a standard normal distribution for easier sampling and generation of new features or data $k$.

*Decoder:* Decompresses latent space data $k$ back to its original features based on the compressed features to produce reconstructed features $\hat{x}$ that should be closely similar to the input data $x$.

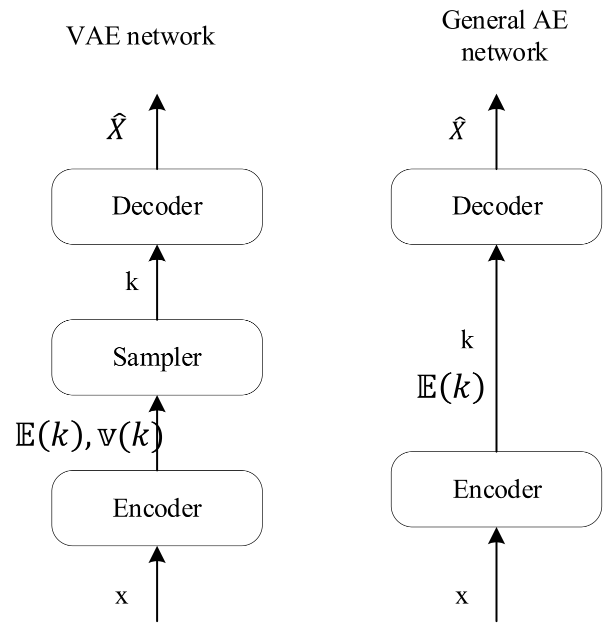


**FIGURE 1.** **A VAE network versus a general AE network.**

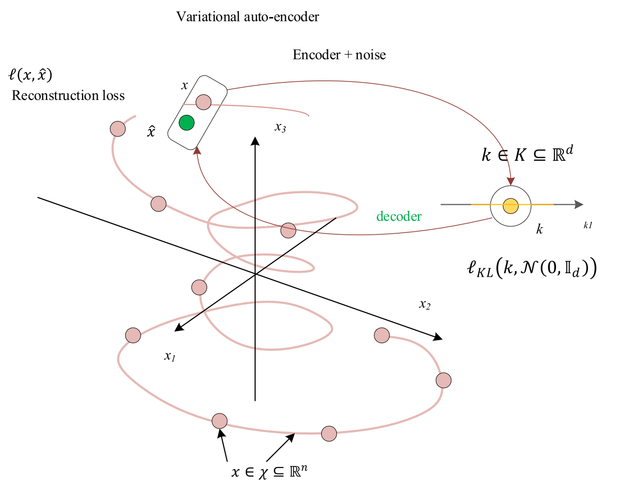


**FIGURE 2.** **Modeled VAE network.**

The mean, $\mathbb{E}(k)$, and the variance, $\mathbb{v}(k)$, represent the parameters that provide a Gaussian representation of the latent variable $k$. They are the outputs of the encoder. $x$ and $\hat{x}$ are the input and reconstructed data, respectively.

In variational autoencoders, there is additional noise, as illustrated in Figure 2, because there is no zero variance compared with a general autoencoder. Each region of space has a variance and mean. The system is trained using latent variable $k$ to obtain the output $\hat{x}$. However, to obtain an output that is as close to the input, the reconstruction loss, $\ell_{(x,\hat{x})}$ must be minimized. The spiral formatted in Figure 2 demonstrates movements in the latent space at different points with the aim of determining what is happening across all those points by enforcing some structure through relative entropy.

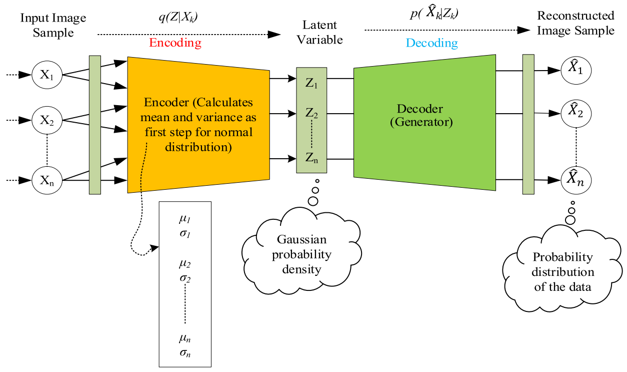


**FIGURE 3.** **A detailed VAE neural network architecture.**

As illustrated in Figure 2, to move from the latent space to the input space, the network learns the distribution or enforcement of some structures through the relative entropy $\ell_{(x,\hat{x})}$. Latent variables were created using (1) to obtain a Gaussian distribution with a specific meaning.

$$k = \mathbb{E}(k) + (\varepsilon \odot \sqrt{\mathbb{V}(k)}), \tag{1}$$

where $\mathbb{E}(k)$ and $\mathbb{V}(k)$ are the mean and variance, respectively for latent variable $k$ Gaussian distribution representation. The symbol $\odot$ is the Hadamard product.

The latent loss $\ell_{KL}(k, \mathcal{N}(0, \mathbb{I}_d))$ penalizes the latent distribution to minimize the loss as much as possible. It enforces some structures in the latent space to allow sampling.

where some noise $\varepsilon \sim \mathcal{N}(0, \mathbb{I}_d)$ represents the normal distribution and $\mathbb{I}_d$ signifies the identity matrix as the covariance matrix.

## IV. BLOCK-BASED VAE IMAGE COMPRESSION SOLUTION PROCEDURE

This section introduces an optimized block-based image compression method for wireless sensor that uses a variational autoencoder. The rate-distortion performance was optimized by adjusting the latent distribution variables in the bottleneck layer. Moreover, blocks of images of equal size were used to improve the autoencoder's training efficiency and algorithm testing. In addition, a relative error bound between the input and reconstructed samples is introduced to further optimize the algorithm.

Figure 3 illustrates the essential VAE network structure. An input image sample goes through an encoder, which produces the mean and variance values for the latent distribution. The latent variable, which is a Gaussian distribution, is the sample based on the variance and mean to generate new data that is reconstructed as the reconstructed image sample. $q(Z|X_k)$, and $p(\hat{X}_k|Z_k)$ are the unknown posterior distribution and reconstructed model distribution, respectively.

The sample distribution $X_n$ was supplied as an input during the encoding process. Second, the unknown posterior distribution was estimated through $q(Z|X)$ model identification. The model follows a normal distribution. The standard deviation $\sigma_n$ and mean $\mu_n$ of the latent Z distribution are then derived. $\sigma_n$ and $\mu_n$ are used to generate a randomized sample that corresponds to $Z_k$ using variational autoencoders. A new sample $X_i$ is generated during decoding of sample $Z_i$ in the latent space using the generational model $p(X|Z)$. The main objective of training the VAE is to regenerate image $\hat{X}$, which is closer to the input image $X_i$. The distance that separates the two probabilities is customarily measured using relative entropy. The two distributions are closer to each other if their relative entropy is as close as possible to zero. Therefore, the targeted function is defined in (2).

$$\ell_{KL} = \left(p(X) \left\| p\left(\hat{X}\right)\right.\right) = \int p(X) \log \frac{p(X)}{p\left(\hat{X}\right)} dX, \tag{2}$$

An unknown posterior distribution $p(Z|X)$ is obtained through the VAE when the recognition model $q(Z|X)$ is introduced. Hence, the target function was optimized using the maximum likelihood method by obtaining the log-likelihood function in (3).

$$\log p(X) = \ell_{KL}(q(Z|X) \| p(Z|X)) + L(X), \tag{3}$$

The relative entropy between the two distributions, denoted as $\ell_{KL}(q(Z|X) \| p(Z|X))$, is initially introduced in (4) and is subsequently elaborated upon explicitly in (5), (6), and (7) as follows:

$$G = \int q(Z|X) \log \frac{q(Z|X)}{p(Z|X)} dZ \tag{4}$$

Additionally, the conditional modeling for the two distributions in (4) is elaborated upon in (5).

$$\begin{aligned} F &= \int q(Z|X) \log \frac{q(Z|X)}{p(Z|X)} dZ \\ &= \int q(Z|X) \left( \log q(Z|X) - \log \frac{p(Z,X)}{p(X)} \right) dZ, \end{aligned} \tag{5}$$

where the ratio $\frac{p(Z,X)}{p(X)}$, is connected to the intractability of the posterior distribution, $p(Z|X)$ in arriving at (6). Since $p(Z,X)$ s feasible to compute, it results in a manageable marginal likelihood, $p(Z)$, which in turn leads to a manageable posterior, $p(Z|X)$ and vice versa.

$$\begin{aligned} H &= \int q(Z|X) (\log q(Z|X) - \log p(Z|X) + \log p(X)) dZ \\ &= \int q(Z|X) (\log q(Z|X) - \log p(Z|X)) dZ + \log p(X), \end{aligned} \tag{6}$$

where (6) is subsequently simplified to obtain (7).

$$\begin{aligned} &\ell_{KL}(q(Z|X) \| p(Z|X)) \\ &\quad = E_{Z \sim q(Z|X)} log \frac{q(Z|X)}{p(Z,X)} + \log p(X), \end{aligned} \tag{7}$$

where $E_{Z \sim q(Z|X)}$, represents the estimated unknown posterior distribution derived from the identification of the $q(Z|X)$ model.

In addition, (3) and (7) are combined to derive the likelihood function $L(X)$ in (10) that is simplified from (8) and (9), which is the lower bound of the variation in the sample. *L(X)* measures the extent to which the model, with its specific parameters, describes and accounts for the observed data.

$$U = E_{Z\sim q(Z|X)} \log \frac{p(Z,X)}{q(Z|X)} = E_{Z\sim q(Z|X)} \log \frac{p(X|Z)\,p(Z)}{q(Z|X)}, \quad (8)$$

where the expression $p(Z, X)$, denotes a joint distribution that is computationally efficient to calculate. t is derived from the posterior distribution, $p(Z|X)$, and is obtained as the product of $p(X|Z)$ and $p(Z)$, representing the posterior distribution and the marginal likelihood, $p(Z)$, respectively.

A more straightforward form of (8) is given in (9), illustrating the calculations for deriving the likelihood function L(X) as $v$.

$$\begin{aligned} V &= \int q(Z|X)\,(\log p(Z) - \log q(Z|X) + \log p(X|Z)\,dZ) \\ &= -\int q(Z|X)\left(\log \frac{q(Z|X)}{p(Z)}\right) dZ \\ &\quad + q(Z|X) \log \int p(X|Z)\,dZ, \end{aligned} \quad (9)$$

Hence, the likelihood function $L(X)$ is obtained as demonstrated in (10).

$$L(X) = -\ell_{KL}\left(q(Z|X)\,\|p(Z)\right) + E_{Z\sim q(Z|X)}\left(\log p(X|Z)\right), \quad (10)$$

*L(X) <= log(P(X)* can be obtained because of the non-negativity of the relative entropy. The closeness of $L(X)$ to $P(X)$ relates to smaller relative entropy values. Hence, (11) is used to calculate the loss function. Therefore, the smaller the target function, the closer the reconstructed image sample to the input image sample.

$$Loss = \ell_{KL}\left(q(Z|X)\,\|p(Z)\right) - E_{Z\sim q(Z|X)}\left(\log p(X|Z)\right), \quad (11)$$

The recognition model likelihood function $\log P(X)$ is presented in (12), in a VAE architecture, with the relative entropy represented by (13). The recognition model serves as an approximate reverse of the generative model. In this research, the neural network architecture is a combination of two variational autoencoders, forming a posterior superposition. To enhance both the primary and hyperprior components, a residual network structure is introduced. The optimization of rate-distortion performance involves fine-tuning the distribution of latent variables y and z within the bottleneck layer. By applying the theoretical principles of the variational autoencoder, we derive the log-likelihood function for the recognition model, represented as in (12).

$$\log P(X) = \ell_{KL}\left(q\,\middle\|p_{\tilde{y},\tilde{z}|x}\right) + L(X), \quad (12)$$

The relative entropy linked to the log-likelihood function of the recognition model within a VAE holds great importance in evaluating the fidelity of the recognition model's approximation to the data distribution. Equation (13) represents the derived relative entropy from this distribution.

$$\begin{aligned} \ell_{KL}\left(q\,\middle\|p_{\tilde{y},\tilde{z}|x}\right) &= E_{x\sim p_x} E_{\tilde{y},\tilde{z}\sim q}\Big(\log q - \log P_{x|\tilde{y}}\left(x|\tilde{y}\right) \\ &\quad - \log p_{\tilde{y}|\tilde{z}}\left(\tilde{y}|\tilde{z}\right) - \log p_{\tilde{z}}\left(\tilde{z}\right)\Big), \end{aligned} \quad (13)$$

The computation of a constrained loss function involves assessing how well the recognition model conforms to specified constraints. This assessment is based on the relative entropy of the log-likelihood function associated with the recognition model. Therefore, the constrained loss function is computed using (14).

$$Loss = R + \lambda * D, \quad (14)$$

where D and R are defined in (15) and (16) and derived from the relative entropy, $\ell_{KL}\left(q\,\middle\|p_{\tilde{y},\tilde{z}|x}\right)$. $\lambda$ represent the Lagrange multipliers in image compression enable you to control the trade-off between image quality and compression rate, assisting in finding the right balance for the specific needs.

$$R = E_{x\sim p_x} E_{\tilde{y},\tilde{z}\sim q}\left(-\log p_{\tilde{y}|\tilde{z}}\left(\tilde{y}|\tilde{z}\right) - \log p_{\tilde{z}}\left(\tilde{z}\right)\right), \quad (15)$$

$$D = E_{x\sim p_x}\left(-\log P_{x|\tilde{y}}\left(x|\tilde{y}\right)\right), \quad (16)$$

Moreover, the compression rate is represented by R, and D represents the distortion calculated using either mean square error (MSE) or MS-SSIM.

### A. OVERVIEW OF THE DESIGN FRAMEWORK

The proposed algorithm comprises offline training, online decompression, and compression. Image data are split into several fixed-size blocks of either 32 × 32 for 2D or 16 × 16 × 16 for 3D, depending on the user's choice. Dividing the image data into blocks improves the network efficiency by capturing only perfect features. Moreover, more samples are created for training the network, which leads to better fine-tuning of the network. The key steps in our design framework are training the VAE network and compression, decompression, and reconstruction of the image data. The input data were split into several blocks and normalized before being supplied to the VAE network. Prediction, sampling, entropy coding, and quantization occur in the VAE network. Our proposed algorithm is a block-based algorithm that uses a relative error bound mechanism instead of a manually set absolute error bound mechanism. These two error bounds are the ones that are primarily used in the literature. Compared to the absolute error bound, the significant advantage of the relative error bound is that it is based on the error approximation between the input matrix and low-rank data approximation [34]. This implies a perfect recovery of the reconstructed image.

In this study, the reconstructed image quality is controlled by the relative error-bound mechanism denoted by Δ,and it is generated adaptively based on the type of input data supplied to the network. Errors due to compression are defined as the difference between the corresponding input and output points of the input and decompressed image data, respectively. Therefore, all image data points are restricted to the error bound, where the decompressed $\hat{X}_i$ values fall between $[X_i - \Delta, X_i + \Delta]$. $X_i$ is the original data point. $\Delta = \beta r$ is a linear function that covers the range of data values globally and represents the relative error bound. $\beta (\in (0, 1))$ is the ratio of the error bound *(er)* and $r$ is the size of the range. For a dataset of $\{X_1, X_2, \ldots\ldots, X_n\}$, $r$ takes the following value for the range size: $\max_{i=1....N} (X_i) - \min_{i=1.....N} (X_i)$.

Algorithm 1 presents the pseudocode for the proposed solution. Similarly, compression occurs in a block-based manner across networks.

**Algorithm 1** Image Compression and Decompression

**Input:** Input image data $X$, block-size *B_Size*, Reconstruction error_bound $\boldsymbol{\varepsilon}_r$, sampling error_bound $\boldsymbol{\varepsilon}_s$
**Output:** Reconstructed image $\hat{X}$

Divide $X$ into image blocks of Size *B_Size*(1-D), *B_Size x B_Size*(2-D), or *B_Size x B_Size x B_Size* (3-D).
**for** (*all blocks in the image data block by block (B)*)
  z ← Enc(B). *//The encoder network Enc encodes the input data block by // block.*
  $z' \leftarrow$ f(z, $\boldsymbol{\varepsilon}_s$)*//Decompressedlatent distribution of the block.*
  $B' \leftarrow$ Dec($z'$) *//Decoded block from the decoder network.*
  $\text{Loss1}_1 \leftarrow \|B - B'\|_1$ *// Calculate the loss between the input and // reconstructed blocks using MSE.*
  $\text{Loss2}_1 \leftarrow \|B - B'\|_1$ *// Calculation of the loss between the input block and // the reconstructed block using MS-SSIM.*
  **if** $\text{Loss1}_1$ >= $\text{Loss2}_1$ *// Comparison of the two losses to pick the most negligible loss //for reconstruction quality.*
    Loss ← $\text{Loss2}_1$ *//minimum loss will be preferred for optimization //purposes.*
  **Else**
    Loss ← $\text{Loss1}_1$
  **End if**
**End for**
Residual ← input Image – reconstructed image
**if** residual <= reconstruction error_bound $\boldsymbol{\varepsilon}_r$
  **Output** ← reconstructed image $\hat{X}$
**End if**

### B. CONVOLUTIONAL VAE NETWORK STRUCTURE

The network input consisted of batches of image blocks, as shown in Figure 4. After sampling, the encoder network generates a compressed distribution of the input data, which is regenerated by the decoder network. The input blocks were fed to the encoder network after normalization based on the minimum and maximum values calculated globally from the input image data. The decoder and encoder were designed from several blocks of deconvolution and convolution with fully connected layers to allow latent rescaling. The deconvolution blocks consist of sequence layers of Deconvolution of Stride1, Stride 2, and Inverse Generalized Divisive Normalization (iGDN) as part of the decoder network. In contrast, the convolutional blocks consist of sequences of layers of convolution of stride 1, stride 2, and Divisive Normalization (GDN) [35] for the encoder network. The kernel size for each convolution or deconvolution for the network is $3 \times 3 \times 3$ for 3-dimensional input or $3 \times 3$ for 2-dimensional input. Stride-1

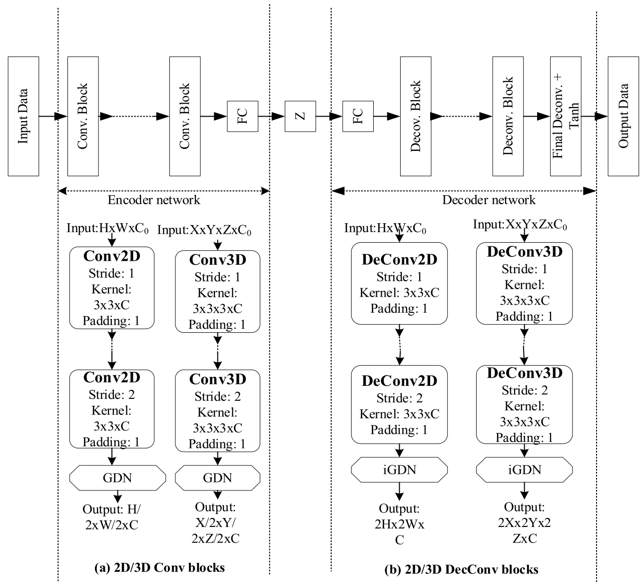


**FIGURE 4. Block-based convolutional VAE network architecture.**

convolutions were applied before stride-2s to allow for an increment of parameters without compromising the reduction speed of the feature map sizes. This alignment minimizes any harm to the network performance. Generalized Divisive Normalization (GDN) is adopted as the activation function in this research because it guarantees better image regeneration quality than the popular sigmoid, Leaky ReLu, and ReLu activation functions. GDN has been widely used with autoencoders, and its strengths have been demonstrated in previous works [35], [36].

The VAE network adapts to varying datasets by changing the channel numbers and (de)convolution blocks while maintaining the same network. Autoencoders can encode datasets irrespective of their dimensionality. This is shown in Fig. 4.

## V. EVALUATION AND EXPERIMENTAL RESULTS

### A. EXPERIMENTAL SETUP

#### 1) THE ENVIRONMENT FOR EXPERIMENTS

The simulations, coding, and testing were performed in a Windows environment using MATLAB R2022b.

#### 2) DATASETS

Commonly used datasets for image compression include ImageNet [37], DIV2K [38], and Kodak [39]. The Kodak, ImageNet, and Image Compression Benchmark [7] datasets were used in this study. The Kodak dataset comprises 24 images that have yet to be lossy compressed at a resolution of $768 \times 512$ pixels. Texture and its content vary and provide sensitive attributes to artifacts. Images from the Image Compression Benchmark were high-precision, high-resolution, and natural images suitable for image compression algorithm evaluation. Images from different sources are available in the form of Red, Green, Blue (RGB), and gray. In addition, they are in 8-bit and 16-bit.

#### 3) COMPARSION WITH OTHER ALGORITHMS

The algorithm used in this study was compared to three (3) other lossy image compression algorithms based on neural networks in the form of autoencoders. Four algorithms were developed by Theis et al. [40], JPEG (4:2:0), Rippel and Bourdev [19], and Abu Alsheikh et al [41]. Other conventional algorithms used for comparison were JPEG [7], JPEG2000 [7], and HDPhoto [7]. Theis et al. 's algorithm is a computationally efficient image compression algorithm that uses a recurrent neural network (RNN), a sub-pixel architecture that makes it relevant for high-resolution images. Moreover, it provides a simple and effective method to address the non-differentiability of autoencoder training for lossy image compression. The method proposed by Rippel et al. is an autoencoder-based algorithm that features pyramidal analysis, coding model that is adaptive, and expected codelength regularization. It also features adversarial training, specifically for use in compression settings, which leads to visually attractive reconstructions at high compression ratios. JPEG (4:2:0) is a state-of-the-art lossy image-compression method. The algorithm proposed by Abu Alsheikh et al. [41] reduced the energy consumption of a WSN through data congestion minimization using a general autoencoder. JPEG, JPEG200, and HDPhoto are state-of-the-art algorithms commonly used for lossy image compression experiments.

#### 4) THE CONFIGURATIONS

The input block sizes were divided into 32 × 32 blocks with a latent size of 16 and a channel of (32,64,128, 256).

#### 5) EVALUATION METRICS

Several evaluation metrics that are more relevant to our research work in measuring the quality of the reconstructed image from image compression by comparing input image $X$ and reconstructed image $\hat{X}$ are discussed herein. Reconstructed image quality measurement also involves visualization of the regenerated image quality within the same bit rate (24). These metrics include the Root Mean Square Error (RMSE) [42], Feature Similarity Indexing Method (FSIM) [42], Structural Similarity Index Method (SSIM) [42], [43], multiscale structural similarity index method (MS-SSIM), peak signal-to-noise ratio (PSNR) [43], Compression Ratio (CR), and Coefficient of Determination ($R^2$). However, the most commonly used ones are

*Multi-Scale Structural Similarity Index Method (MS-SSIM)* [44] - This is an improvement of the single-scale index method, called the Structural Similarity Index Method (SSIM). This metric considers the details of the images within different resolutions. It incorporates the luminance comparison, contrast comparison, and structural comparison, represented by (19), (19), and (20), respectively, at different scales; *the* MS-SSIM is represented in (17).

$$MS-SSIM\left(x,\hat{x}\right)=\left[l_M\left(x,\hat{x}\right)\right]^{\propto M}\cdot\prod_{j=1}^{M}\left[c_j\left(x,\hat{x}\right)\right]^{\beta_j}\left[s_j\left(x,\hat{x}\right)\right]^{\gamma_j},\quad(17)$$

where $\propto M$, $\beta_j$, and $\gamma_j$ are for adjustment of distinct components' relative importance.

$$l_M\left(x,\hat{x}\right)=\frac{2\mu_x\mu_{\hat{x}}+C_1}{\mu_x^2+\mu_{\hat{x}}^2+C_1},\quad(18)$$

$$c\left(x,\hat{x}\right)=\frac{2\sigma_x\sigma_{\hat{x}}+C_2}{\sigma_x^2+\sigma_{\hat{x}}^2+C_2},\quad(19)$$

$$s\left(x,\hat{x}\right)=\frac{\sigma_{x\hat{x}}+C_3}{\sigma_x\sigma_{\hat{x}}+C_3},\quad(20)$$

where $C_1$, $C_2$, and $C_2$ constants are small and represented in (21).

$$C_1=\left(K_1L\right)^2,\ C_2=\left(K_2\right)^2,\ \text{and}\ C_3=\frac{C_2}{2},\quad(21)$$

where an MS-SSIM value of one means that the two compared images are entirely identical, while a value of zero (0) represents completely unidentical images.

*Peak-Signal-to-Noise Ratio (PSNR)* [43] – PSNR value approaches infinity, with values of the mean square error (MSE) in (23) approaching zero. Therefore, the larger the MSE value, the smaller is the PSNR value. This implies that the two images were different. For a grey-level image with 8 bits, the PSNR between the input image $x$ and reconstructed image $\hat{x}$ of size $M \times N$ is given by (22).

$$PSNR\left(x,\hat{x}\right)=10\log_{10}\left(\frac{peakvalue^2}{MSE\left(x,\hat{x}\right)}\right),\quad(22)$$

In this case, the *peak value* is 255 from $2^n-1$, where $n$ is the number of bits and MSE is defined in (23).

$$MSE\left(x,\hat{x}\right)=\frac{1}{MN}\sum\nolimits_{i=1}^{M}\sum\nolimits_{j=1}^{N}\left(x_{ij}-\hat{x}_{ij}\right),^2\quad(23)$$

$$Bit\ Rate=\frac{Size_{of}\ (datatype)}{compression\ ratio\ (CR)},\quad(24)$$

The higher the compression ratio, the smaller is the bit rate, as represented by (24).

According to coding and information theory [45], conventional data and image compression algorithms are not designed for resource-limited applications such as WSNs. Their main target is storage optimization rather than energy conservation [46]. Therefore, it is essential to address the computational requirements for techniques developed for WSNs to avoid a situation in which the energy consumption during CPU computations for complex techniques exceeds the consumed energy for sending fewer bits over the radio frequency (RF) module. To measure the energy conserved from image compression within a wireless sensor network, we considered the power consumed in receiving image data using a complexity analysis metric [47] defined by (25).

**TABLE 1.** **9xTend RF module specifications [49].**

| | Specification | Parameters |
|---|---|---|
| 1. | Frequency Range | 902 - 928$MHz$ |
| 2. | Data Rate of $R_{XTend}$ | 9,600$bps$ |
| 3. | Spread Technology | Frequency-hopping spread spectrum (FHSS) |
| 4. | Transmission Range | 0.9$km$ for urban areas, 22$km$ for outdoor areas |
| 5. | Data compression current consumption during transmission ($I_{TX}$) | 600$mA$ |
| 6. | Data compression current consumption during the reception ($I_{RX}$) | 80$mA$ |
| 7. | Idle mode flow of current | 1$mA$ |
| 8. | Supply voltage ($V_{CC}$) | 3.3$V$ |

Moreover, the energy conserved during image compression was measured using (27). In this study, we considered an MSP430 microcontroller described in [48] with the properties of a 16-bit CPU, 3.3 Volts, 3.3 MHz as clock rate, and 1.85$m$A of the current for an inactive mode. Hence, the power consumption from the MSP430 microcontroller at the clock cycle is 1.85$n$J from the computations using (25).

$$P_{CC} = V_s \times I_{CR}, \tag{25}$$

where $P_{CC}$ is the power consumed per clock cycle by MSP430 microcontroller, $V_s$ being the supplied voltage, and $I_{CR}$ as the current consumption clock rate.

The specifications of the CPU cycle operations used for calculating the energy consumed were derived from a table in [48]. To model the number of cycles in an autoencoder (AE) image compression network $C_{VAE}(X, Z)$ where $X$ is the image data input vector, and the compressed image data vector representation is described as Z. The design described in [41] was adopted. The metric in Eq. (26) was used to measure the conserved energy during image compression.

$$C_{VAE}(X, Z) = N_I \times E_{W+B} + F_c, \tag{26}$$

where $N_I$ represents image data normalization, $E_{W+B}$ as the weights and biases for the encoder, and $F_c$ representing the activation function computation.

The 9XTend RF module was described in [49]. The specifications that make it a more relevant module for backhaul link transmission are summarized in Table 1.

Hence, the energy consumption for the transmission and reception of one bit of image data is 233.75$\mu J$ when calculated using (27).

$$S_{bit} = \frac{V_{CC} \times I_{TX} + V_{CC} \times I_{RX}}{R_{CC2420}}, \tag{27}$$

Therefore, the energy consumed through the transmission and received by the network at the next hop through the transceiver unit is approximated to be similar to the energy consumed by the microcontroller in 125,945 CPU clock cycles. The design components are used for formulating the computational complexity in clock cycles for the input data compression using the variational autoencoder network by using (26). Hence, the complexity in the computational input image data compression through an autoencoder is formulated as in (26). Therefore, the energy consumption for image data transmission with compression is $E_{VAE}$, where $X$ is the input data vector and $Z$ is the compressed data vector (28).

**TABLE 2.** **MSP430 microcontroller clock cycles.**

| | |
|---|---|
| Division | 405 |
| Multiplication | 395 |
| Addition | 184 |
| Subtraction | 177 |
| Comparison | 37 |
| exp(.) | 52000 |

$$E_{VAE} = E_{CLK} \times C_{VAE}(X, Z) + 32bits \times Z \times S_{bit,} \tag{28}$$

It is specifically for sensor readings represented by a floating number of 32 bits. Using Table 2 adopted in [48], the model complexity of the algorithm was calculated with results shown in Figure 15.

## B. RESULTS AND ANALYSIS

### 1) RATE-DISTORTION ANALYSIS

MS-SSIM, PSNR, and other evaluation metrics quantify the distortion between the input and output images. The curved graphs in Figures 6 and 7 represent the trade-offs for rate distortion for different algorithms for distinct values of $\lambda$, the Lagrange multiplier for constrained optimization. The results are shown in Table 3. The results differed depending on the metric used for training the loss function. In addition, the proposed method produced better reconstruction fidelity when compared to the BPG (4:4:4) method, especially at bit rates below 1 bit per pixel (bpp). However, at higher bit rates, the two methods performed similarly. Moreover, at a bit rate of 0.4 to 1 bit per pixel (bpp), the proposed method's reconstruction fidelity measured using PSNR ranges from 33 dB−37dB, and it is more than that of the BPG (4:4:4) method. However, from 1 to 2 bpp, the two methods' PSNR values are almost identical ranging from 37 to 42 dB. Therefore, these results suggest that the proposed method is a good alternative to existing algorithms in the literature, particularly when low bit rates are required. This can be attributed to the introduced relative error bound, which is dependent on the training data and training efficiency using blocks of images together with constrained optimization. Additionally, this is related to the sampling optimization of the latent distribution using the absolute latent loss between the MSE and MS-SSIM. Smaller values of PSNR of less than 30 dB at bit rates of 0.4 bits per pixel (bpp) indicate that the proposed algorithm can reconstruct a close match to the input image at higher compression ratios. Table 3 summarizes the simulation parameters used in the experiments.

The MS-SSIM metric is a widely used image quality assessment metric. It measures how similar a reconstructed image is to the original image. In Figure 7, four different

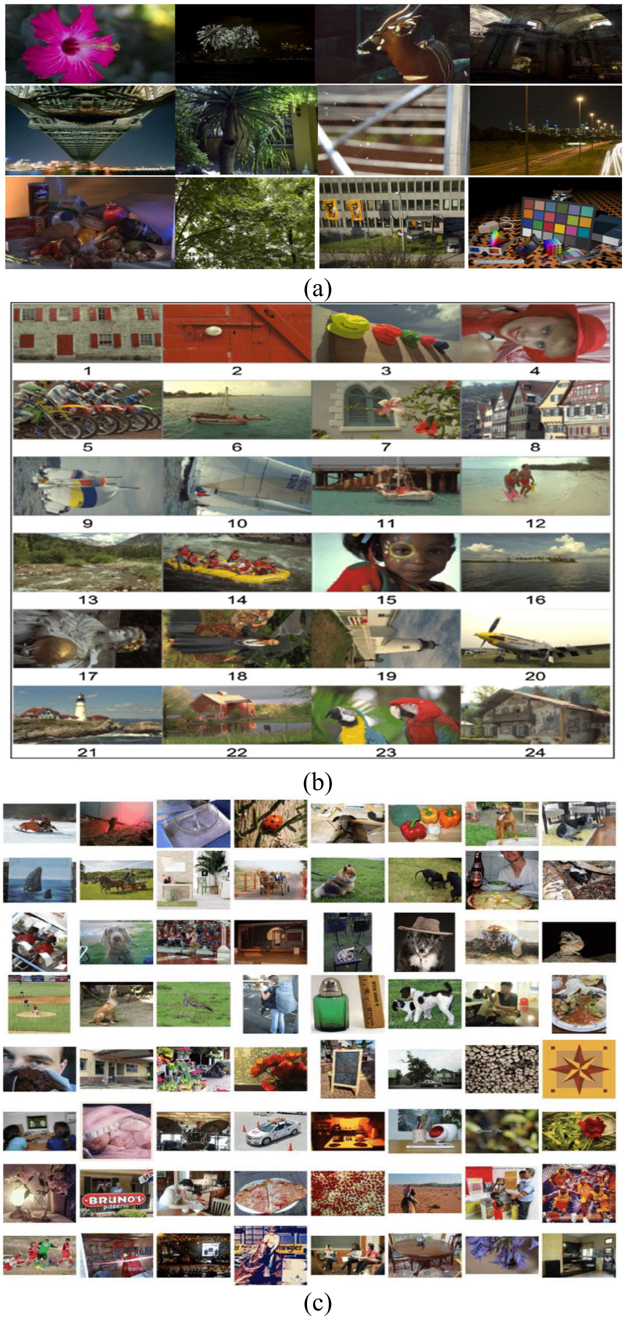


**FIGURE 5.** Image compression benchmark (a), Kodak (b), and ImageNet (c) datasets.

image compression methods were compared using the MS-SSIM metric. As shown in Figure 7, at bit rates between 0.2 and 1.4 bpp, the proposed algorithm had a reconstruction fidelity ranging from 12 to 21.5 dB, which is higher than the other methods. At bit rates above 1 bpp, the proposed algorithm had a slight increase in reconstruction fidelity compared to lower bit rates. Overall, these results demonstrate the strength of the proposed algorithm in image reconstruction at higher compression rates. Therefore, the results show that the proposed algorithm achieves the highest reconstruction

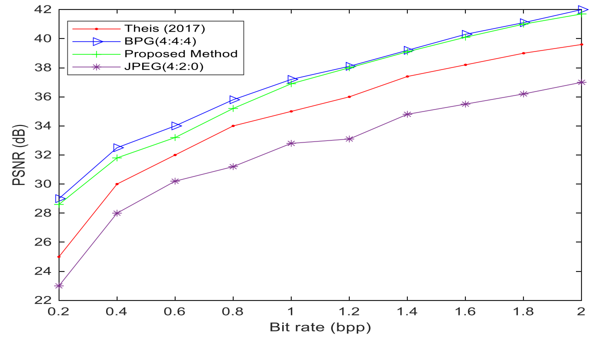


**FIGURE 6.** PSNR on the Kodak dataset.

**TABLE 3.** Simulation parameters.

| Parameter | Value |
|---|---|
| Loss Function | Loss = *BPP*+ $\lambda$***MSE* |
| $\lambda$ values | 0.0045, 0.0065, 0.01,0.011,0.015,0.02,0.09 |
| Iterations | 1.2 M times |
| Batch Size | 8 |
| Patch size | 256*256 |
| Learning Rate | Initial (0.001), After stabilization (0.0001) |

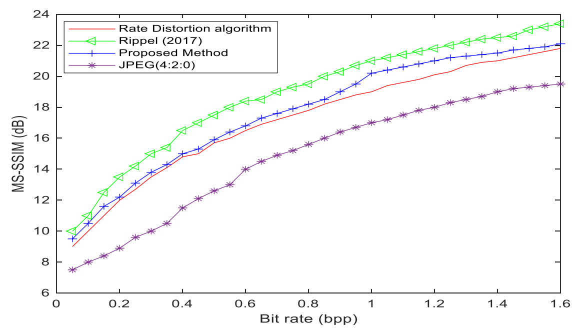


**FIGURE 7.** MS-SSIM on the Kodak dataset.

fidelity among the compared methods when the compression rate is high. This is because the proposed algorithm uses a novel deep learning-based image reconstruction framework that is specifically designed for image compression. This framework is able to better preserve the important features of the original image during compression resulting in a higher reconstructed image.

Other image reconstruction fidelity evaluation metrics reported in the literature include the coefficient of determination ($R^2$), SSIM, RMSE, and FSIM. The results of these evaluation metrics on the Kodak dataset are presented in Figures 8–11. As shown in Figure 8, the proposed algorithm demonstrated a high coefficient of determination at higher compression rates. The proposed algorithm's reconstruction fidelity increased significantly as the compression ratio increased from 40 to 70, and remained constant after that. This means that the proposed algorithm is very effective at higher compression rates. These results guarantee the reconstructed image quality at high compression rates by the proposed method, which is highly desirable for resource-constrained wireless sensor networks.

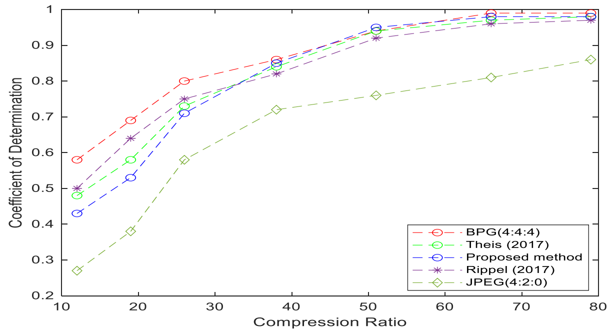


**FIGURE 8. Coefficient of determination ($R^2$) on the Kodak dataset.**

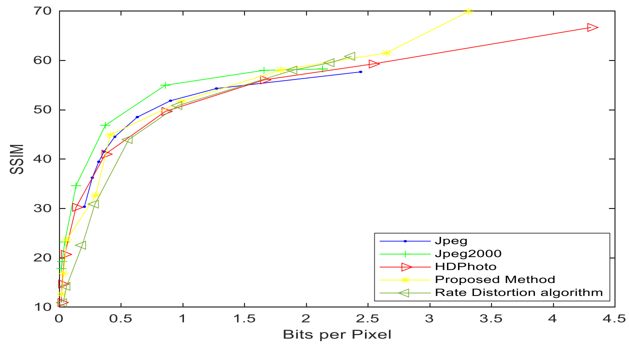


**FIGURE 9. SSIM on the image compression benchmark dataset.**

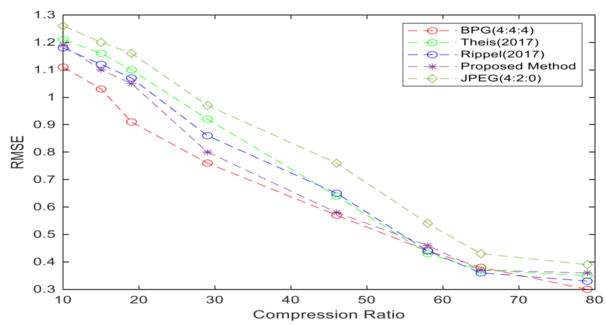


**FIGURE 10. RMSE on the ImageNet dataset.**

Its performance, when compared to conventional and non-conventional algorithms, shows that it performs at a relatively higher level than most of these algorithms. Improvement of training efficiency using blocks of images provided reconstructed images of high quality that were almost closer to the input images. In addition, the use of constrained optimization to balance the reconstruction loss and compression ratio proved to be practical for controlling the distortion levels in the acceptable region.

Figure 9 compares the proposed algorithm with most conventional algorithms and the modified rate-distortion algorithm [41] using SSIM. The dataset used in this experiment was obtained from an image-compression benchmark. The proposed algorithm achieved a significant increase in reconstruction fidelity from a bit rate of 0.1 to 1 bpp, but

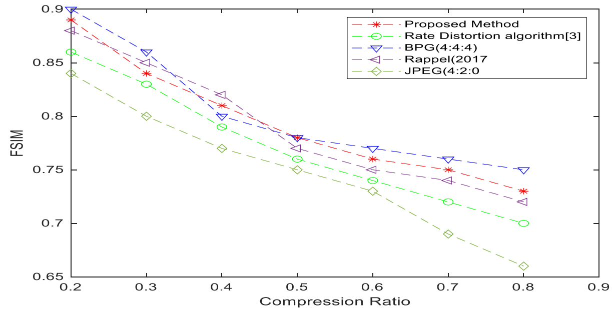


**FIGURE 11. FSIM on the ImageNet dataset.**

only a slight increase at higher bit rates. This shows that the proposed algorithm is most effective at lower bit rates, and its performance plateaus at higher bit rates. Therefore, the results show that the algorithm performs better than the conventional algorithms at higher bit rates. Hence, it emphasizes its strength in balancing the compression rate and distortion of the reconstructed image. Table 4 provides the simulation results for the image compression benchmark dataset from state-of-the-art algorithms, together with the results for our proposed algorithm.

Furthermore, comparing the algorithm with conventional and existing algorithms in the literature using RMSE, the algorithm demonstrated better reconstruction fidelity at high compression ratios with an RMSE of close to zero at a CR of 65%. At a compression ratio of 65%, the proposed algorithm's RMSE value approaches zero, indicating that the reconstructed image is almost indistinguishable from the original image. This is illustrated in Figure 10. The same trend was observed when evaluating the algorithm with the FSIM evaluation similarity metric on the ImageNet dataset, and the results are shown in Figure 11.

The proposed algorithm achieves a higher compression ratio of 0.8 while maintaining a lower FSIM value of less than 0.75, indicating that images can be compressed more efficiently while maintaining image quality. This is demonstrated in Figure 11.

### 2) COMPRESSION RATIO AND ERROR-BOUND ANALYSIS

The strength of an error-bound mechanism on different algorithms was tested to evaluate the impact of the error bounds on image compression, which guarantees reconstruction fidelity at different compression ratios. Figures 12 and 13 show that various error bounds were used to compare the proposed algorithm with other algorithms at different compression ratios. A lower error bound in the proposed algorithm results in less distortion in the reconstructed image, as demonstrated by the decreasing RMSE values from 1 to 3. However, an increase in the error bound beyond 3 results in increased distortion, as evidenced by the increasing RMSE values. This demonstrates that a lower error bound leads to better image quality. In addition, higher image reconstruction quality was achieved at lower error bounds, as shown in Figure 12.

**TABLE 4.** Image compression benchmark table.

| | Jpeg | | Jpeg2000 | | HDPhoto | | Proposed Method | | Rate-Distortion Algorithm [41] |
|---|---|---|---|---|---|---|---|---|---|
| Bits Per Pixel | SSIM | bits per pixel | SSIM | bits per pixel | SSIM | bits per pixel | SSIM | bits per pixel | SSIM |
| 2.43983 | 57.6722 | 2.132075 | 58.2578 | 4.30722 | 66.6781 | 3.31391 | 69.8951 | 2.36481 | 60.8295 |
| 1.27205 | 54.3092 | 2.132075 | 58.2578 | 2.53679 | 59.2681 | 2.64912 | 61.4987 | 2.19642 | 59.5418 |
| 0.899621 | 51.8341 | 1.656938 | 57.991 | 1.65696 | 56.0525 | 1.79215 | 58.0618 | 1.89542 | 58.0437 |
| 0.630604 | 48.5206 | 0.857698 | 54.9734 | 0.857738 | 49.6607 | 0.98412 | 51.6721 | 0.96348 | 50.8679 |
| 0.447303 | 44.5142 | 0.371546 | 46.8514 | 0.371583 | 41.0028 | 0.41289 | 44.9428 | 0.56135 | 43.9743 |
| 0.36161 | 41.529 | 0.13409 | 34.6274 | 0.134113 | 30.2425 | 0.29348 | 32.6548 | 0.28745 | 30.8952 |
| 0.318076 | 39.4692 | 0.043063 | 23.2222 | 0.0430756 | 20.7011 | 0.058413 | 23.6278 | 0.18456 | 22.5672 |
| 0.266094 | 36.2388 | 0.020974 | 19.235 | 0.021005 | 14.7016 | 0.032154 | 16.8124 | 0.057413 | 14.2359 |
| 0.201677 | 30.3556 | 0.015891 | 17.8458 | 0.0159111 | 10.9579 | 0.0186417 | 12.6951 | 0.029845 | 10.9208 |

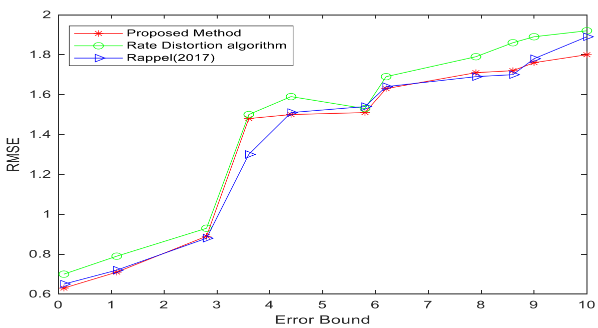


**FIGURE 12.** RMSE and error-bound analysis on the Kodak dataset.

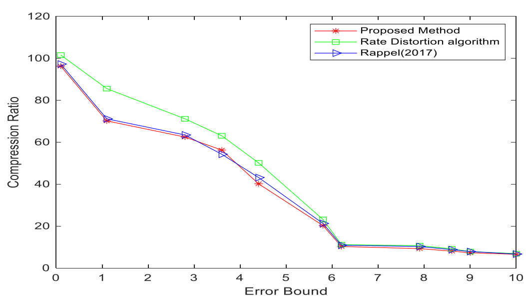


**FIGURE 13.** Compression ratio and error bound analysis on the Kodak dataset.

Therefore, subjecting image compression algorithms to an error-bound mechanism allows the user to balance the reconstruction quality and the application it can be adopted on. Some applications can deal with a certain degree of distortion, whereas others cannot.

Compression ratios above 90% were achieved at the lower error bounds, as illustrated in Figure 13. The proposed method can achieve high compression ratios above 50% when the error bound is 3 or less, the compression ratios become less significant when the error bound is 6 or more. In addition, the compression ratio is fairly constant when the error bound is between 6 and 10. However, the evaluation of the results demonstrated that reasonable compression ratios can be obtained from the proposed algorithm at different error bounds when evaluating using the Kodak dataset.

Table 5 provides the ranking of the compared algorithms based on the commonly used evaluation metrics for image reconstruction fidelity and quality at various compression ratios or bit rates. This table is based on Figures 6,7, 10, and 11.

### 3) ENERGY CONSERVATION FROM IMAGE COMPRESSION

Image compression reduces the energy consumed to transmit image data from the transmitter to the receiver in a wireless sensor network. This is demonstrated in Figure 14. The proposed method saves a significant amount of energy when compressing data and transmitting it across different hop counts compared to transmitting raw data. The proposed method consumes 1155mJ and 1048mJ of energy at compression ratios of 50% and 36% for a hop count of 4, respectively, while transmitting raw data consumes 2750mJ of energy at the same hop count. This represents a significant savings in energy consumption. As illustrated in Figure 14, the amount of energy consumed within a network reduces with the compression level. Transmission of compressed image data at a 60% compression ratio can save more than 50% of the energy required to transmit raw data within a wireless sensor network at different hop counts based on the proposed method.

The model complexity of the proposed algorithm is shown in Figure 15, and it was calculated using the information in Table 2. The computational module requires a constant amount of energy regardless of the compression ratio, but the

**TABLE 5.** Reconstruction fidelity ranking table for the six algorithms with relation to compression ratio.

| Theis (2017) | | | Rippel (2017) | | | | BPG (4:4:4) | | | JPEG (4:2:0) | | | | Proposed Method | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RMSE | $R^2$ | PSNR | RMSE | MS-SSIM | $R^2$ | PSNR | RMSE | $R^2$ | PSNR | RMSE | MS-SSIM | $R^2$ | PSNR | RMSE | MS-SSIM | $R^2$ | PSNR |
| 1.21 | 0.48 | 25 | 1.18 | 10 | 0.50 | 29 | 1.11 | 0.58 | 23 | 1.26 | 7.5 | 0.27 | 28.6 | 1.19 | 9.5 | 0.43 | 23 |
| 1.16 | 0.58 | 30 | 1.12 | 11 | 0.64 | 32.5 | 1.03 | 0.69 | 28 | 1.2 | 8.0 | 0.38 | 31.8 | 1.1 | 10.5 | 0.53 | 28 |
| 1.10 | 0.73 | 32 | 1.07 | 12.5 | 0.75 | 34 | 0.91 | 0.80 | 30.2 | 1.16 | 8.4 | 0.58 | 33.2 | 1.05 | 11.6 | 0.71 | 30.2 |
| 0.92 | 0.84 | 34 | 0.86 | 13.5 | 0.82 | 35.8 | 0.76 | 0.86 | 31.2 | 0.97 | 8.9 | 0.72 | 35.2 | 0.8 | 12.2 | 0.85 | 31.2 |
| 0.64 | 0.94 | 35 | 0.65 | 14.2 | 0.92 | 37.2 | 0.57 | 0.94 | 32.8 | 0.76 | 9.6 | 0.76 | 36.9 | 0.58 | 13.1 | 0.95 | 32.8 |
| 0.43 | 0.97 | 36 | 0.44 | 15 | 0.96 | 38.1 | 0.44 | 0.99 | 33.1 | 0.54 | 10 | 0.81 | 38 | 0.46 | 13.8 | 0.98 | 33.1 |
| 0.37 | 0.98 | 37.4 | 0.36 | 15.4 | 0.97 | 39.2 | 0.38 | 0.99 | 34.8 | 0.43 | 10.5 | 0.86 | 39.1 | 0.37 | 14.3 | 0.98 | 34.8 |
| 0.35 | | 38.2 | 0.33 | 16.5 | | 4.3 | 0.30 | | 35.5 | 0.39 | 11.5 | | 40.1 | 0.36 | 15 | | 35.5 |
| | | 39 | | 17 | | 41.1 | | | 36.2 | | 12.1 | | 41 | | 15.3 | | 36.2 |
| | | 39.6 | | 17.5 | | 42 | | | 37 | | 12.6 | | 41.7 | | 15.9 | | 37 |

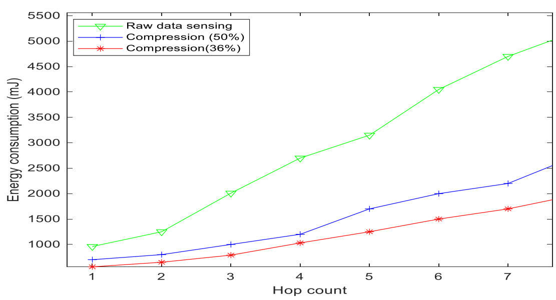


**FIGURE 14.** Consumption of energy at distinct hop counts.

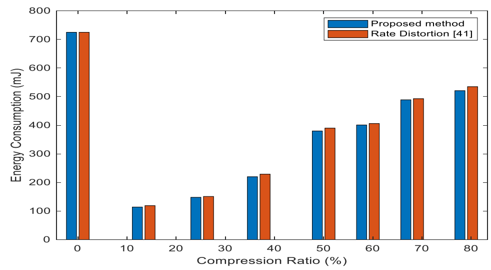


**FIGURE 16.** Energy consumption at different compression ratios on different algorithms.

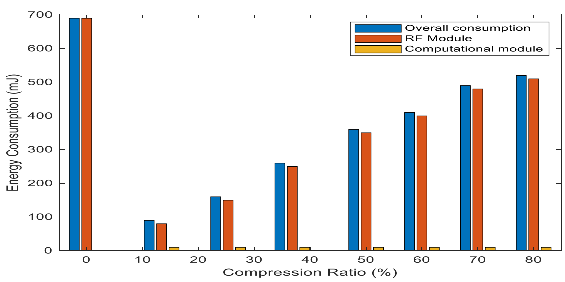


**FIGURE 15.** Energy consumption at different compression ratios for the proposed algorithm.

overall energy consumption decreases proportionally as the compression ratio increases from 80% to 10%. Transmission of uncompressed data requires a significant amount of energy compared to the energy required for the model computation. Additionally, transmission energy contributes more than 90% of the overall energy consumed by the proposed algorithm while less than 10% of the overall energy consumed is from the model computations. However, the overall energy consumed by the proposed method at varying compression ratios decreased as the compression ratio increased, as illustrated in Figure 15. The power required to transmit fewer bits of data can never be equal to that required for transmitting more data bits. At almost 60% compression, image data transmission conserves almost half (420$mJ$) of the energy required to transmit uncompressed data. The proposed algorithm is computationally less based on the operations required in an autoencoder-based image compression method. Figure 15 illustrates the insignificance of the computational complexity of the proposed algorithm in terms of energy consumption. The arithmetic operations of the proposed method are less computationally intensive. Hence, a platform can be used in resource-constrained environments such as WSNs.

When comparing the energy consumption of the proposed algorithm with the rate-distortion algorithm described in [41], the proposed algorithm has similar energy consumption at compression ratios between 10% and 80%. This is demonstrated in Figure 16, which shows that at a compression ratio of 40%, the proposed algorithm's energy consumption is only slightly lower than the referenced algorithm's consumption. Even though this difference may seem small, it represents a meaningful conservation of energy. Therefore, it was found that the trend is like how image compression affects the energy consumed during the transmission of the two algorithms. The algorithms use autoencoder artificial neural networks with more similar computations used during training and testing. The results are shown in Fig. 16. There is a proportionality between the two algorithms in terms of the compression ratio and energy consumption. Therefore, the energy savings achieved through image compression were less than those achieved without compression. As the compression ratio increased, energy savings also increased. Additionally, the consumed energy depends on the level of compression, and it must not compromise the quality of the reconstructed image to balance distortion, energy conservation, and compression. Figure 16 demonstrates that the proposed algorithm significantly reduces the energy consumption from image compression.

## VI. CONCLUSION AND FUTURE DIRECTION

In this study, an optimized block-based lossy image compression algorithm for wireless sensor networks was proposed, coded, simulated in MATLAB, and tested using the Kodak, ImageNet, and Image Compression Benchmark datasets. The approach was optimized through the constrained optimization of the reconstruction loss and minimization of the latent loss through a relative error-bound mechanism. In addition, a comparison was made between existing and proposed algorithms. A reconstruction quality above 80% was achieved with compression ratios above 60% at lower error bounds. Moreover, the energy consumed by the proposed algorithm was significantly reduced by image compression. More than 60% of the compression ratios conserved more than 50% of the energy used to send raw data to WSNs. Therefore, the proposed algorithm provides a significant trade-off between the reconstruction quality and energy conservation at various compression ratios. Something vital for applications with constrained resources, such as wireless sensor networks.

Additionally, the transmission of compressed image data reduces the amount of energy consumed compared to the transmission of raw data. Compared to all other current image compression algorithms that are based on artificial neural networks, our algorithm optimizes the latent-space representation based on the maximum loss between MSE and MS-SSIM. A comparison of our algorithm with existing algorithms from the literature using RMSE shows that the algorithm provides better reconstruction quality at high compression ratios and an RMSE of close to zero at compression ratios of 65% or more. A PSNR of less than 30 dB at 0.2 bits per pixel (bpp) and lower obtained from the proposed algorithm further demonstrated that the algorithm reconstructs a close match to the input image at higher compression ratios. The use of blocks of images provided regenerated images of higher quality, which were almost closer to the input images. Furthermore, the use of constrained optimization to balance the reconstruction loss and compression ratio proved to be practical for controlling the distortion levels in the acceptable region.

Nonetheless, an opportunity for enhancement in future exists within the algorithm to strike a balance between distortion and compression ratio, contingent upon the proximity of the sink that will be done. This approach involves determining whether a greater degree of compression should be applied when the sink is in close proximity or, conversely, when it is further away. Therefore, achieving equilibrium between distortion and compression ratio relies on considering the distance between the algorithm's output and the input data, with the distance to the sink serving as the pivotal factor in determining the optimal trade-off. Furthermore, a comparative assessment with other chaos security-based algorithms may be warranted to gauge the algorithm's robustness in terms of security when transmitting compressed image data.

## ACKNOWLEDGMENT

The authors would like to thank the Botswana International University of Science and Technology for providing different platforms and resources for carrying out these research objectives. The authors also appreciate the support and motivation offered by the Department of Electrical, Computer and Telecommunications Engineering.

## REFERENCES

[1] Y. Hu, W. Yang, Z. Ma, and J. Liu, "Learning end-to-end lossy image compression: A benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4194–4211, Aug. 2022, doi: 10.1109/TPAMI.2021.3065339.

[2] S. Thakral and P. Manhas, "Image processing by using different types of discrete wavelet transform," in *Proc. Int. Conf. Adv. Informat. Comput. Res.*, 2018, pp. 499–507.

[3] V. Dhandapani and S. Ramachandran, "Area and power efficient DCT architecture for image compression," *EURASIP J. Adv. Signal Process.*, vol. 2014, no. 1, pp. 1–9, Dec. 2014.

[4] S. Hariharan and N. Sreelekshmi, "Image compression for wireless sensor networks," *Int. J. Trend Res. Dev.*, vol. 4, no. 5, pp. 161–167, 2017.

[5] J. G. Kolo, S. A. Shanmugam, D. W. G. Lim, L.-M. Ang, and K. P. Seng, "An adaptive lossless data compression scheme for wireless sensor networks," *J. Sensors*, vol. 2012, pp. 1–20, Jan. 2012, doi: 10.1155/2012/539638.

[6] C. Han, S. Zhang, B. Zhang, J. Zhou, and L. Sun, "A distributed image compression scheme for energy harvesting wireless multimedia sensor networks," *Sensors*, vol. 20, no. 3, p. 667, Jan. 2020, doi: 10.3390/s20030667.

[7] B. Becker, A. Richter, T. Frohling, N. Fraser, P. Story, T. Cosshall, W. J. Coffin, and D. Lindbloom. (2015). *Image Compression Benchmark*. Accessed: Aug. 18, 2022. [Online]. Available: https://imagecompression.info/test_images/

[8] M. Khan, A. El Saddik, F. S. Alotaibi, and N. T. Pham, "AAD-Net: Advanced end-to-end signal processing system for human emotion detection & recognition using attention-based deep echo state network," *Knowl.-Based Syst.*, vol. 270, Jun. 2023, Art. no. 110525, doi: 10.1016/j.knosys.2023.110525.

[9] A. Amini, B. Araki, D. Rus, W. Schwarting, G. Rosman, S. Karaman. (2018). *Variational Autoencoder for End-to-End Control of Autonomous Driving With Novelty Detection and Training De-biasing*. Amini. Accessed: Jun. 16, 2023. [Online]. Available: https://dspace.mit.edu/handle/1721.1/118139

[10] M. Connor, G. Canal, and C. Rozell, "Variational autoencoder with learned latent structure," in *Proc. Int. Conf. Artif. Intell. Statist.*, vol. 130, Mar. 2021, pp. 2359–2367. Accessed: Jun. 16, 2023. [Online]. Available: https://proceedings.mlr.press/v130/connor21a.html

[11] X. Zou, T. Lu, W. Xia, X. Wang, W. Zhang, S. Di, D. Tao, and F. Cappello, "Accelerating relative-error bounded lossy compression for HPC datasets with precomputation-based mechanisms," in *Proc. 35th Symp. Mass Storage Syst. Technol. (MSST)*, May 2019, pp. 65–78, doi: 10.1109/MSST.2019.00-15.

[12] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 2, pp. 260–269, Apr. 1967, doi: 10.1109/TIT.1967.1054010.

[13] Z. Zhou and A. M.-C. So, "A unified approach to error bounds for structured convex optimization problems," *Math. Program.*, vol. 165, no. 2, pp. 689–728, Oct. 2017, doi: 10.1007/s10107-016-1100-9.

[14] S. Kwon, "CLSTM: Deep feature-based speech emotion recognition using the hierarchical ConvLSTM network," *Mathematics*, vol. 8, no. 12, p. 2133, Nov. 2020, doi: 10.3390/math8122133.

[15] P. Rafiqul and Z. Khan, "Methods to avoid over-fitting and under-fitting in supervised machine learning (comperative study)," *Comput. Sci., Commun. Instrum. Devices*, vol. 70, pp. 163–172, Jan. 2015.

[16] S. Zafar, N. Iftekhar, A. Yadav, A. Ahilan, S. N. Kumar, and A. Jeyam, "An IoT method for telemedicine: Lossless medical image compression using local adaptive blocks," *IEEE Sensors J.*, vol. 22, no. 15, pp. 15345–15352, Aug. 2022, doi: 10.1109/JSEN.2022.3184423.

[17] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," in *Proc. 4th Int. Conf. Learn. Represent.*, Nov. 2015, pp. 1–12.

[18] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool, ''Soft-to-hard vector quantization for end-to-end learning compressible representations,'' in *Proc. Adv. Neural Inf. Process. Syst.*, Apr. 2017, pp. 1142–1152.
[19] O. Rippel and L. Bourdev, ''Real-time adaptive image compression,'' in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 6, May 2017, pp. 4457–4473.
[20] J. P. Klopp, Y. Wang, S.-Y. Chien, and L.-G. Chen, ''Learning a code-space predictor by exploiting intra-image-dependencies,'' in *Proc. BMVC*, 2018, p. 124.
[21] M. Tschannen, E. Agustsson, and M. Lucic, ''Deep generative models for distribution-preserving lossy compression,'' in *Proc. Adv. Neural Inf. Process. Syst.*, May 2018, pp. 5929–5940.
[22] J. Balle, V. Laparra, and E. P. Simoncelli, ''End-to-end optimization of nonlinear transform codes for perceptual quality,'' in *Proc. Picture Coding Symp. (PCS)*, Dec. 2016, pp. 1–5.
[23] J. Ballé, V. Laparra, and E. P. Simoncelli, ''End-to-end optimized image compression,'' in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–27.
[24] D. Minnen, G. Toderici, S. Singh, S. J. Hwang, and M. Covell, ''Image-dependent local entropy models for learned image compression,'' in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 430–434.
[25] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, ''Learning convolutional networks for content-weighted image compression,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3214–3223.
[26] T. Chen, H. Liu, Z. Ma, Q. Shen, X. Cao, and Y. Wang, ''End-to-end learnt image compression via non-local attention optimization and improved context modeling,'' *IEEE Trans. Image Process.*, vol. 30, pp. 3179–3191, 2021, doi: 10.1109/TIP.2021.3058615.
[27] H. Ma, D. Liu, N. Yan, H. Li, and F. Wu, ''End-to-end optimized versatile image compression with wavelet-like transform,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1247–1263, Mar. 2022, doi: 10.1109/TPAMI.2020.3026003.
[28] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, ''Learned image compression with discretized Gaussian mixture likelihoods and attention modules,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7936–7945.
[29] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, ''Learning image and video compression through spatial–temporal energy compaction,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10063–10072.
[30] N. Johnston, D. Vincent, D. Minnen, M. Covell, S. Singh, T. Chinen, S. J. Hwang, J. Shor, and G. Toderici, ''Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4385–4393.
[31] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. V. Gool, ''Conditional probability models for deep image compression,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4394–4402.
[32] D. Minnen, J. Ballé, and G. Toderici, ''Joint autoregressive and hierarchical priors for learned image compression,'' in *Proc. Adv. Neural Inf. Process. Syst.*, Sep. 2018, pp. 10771–10780.
[33] D. Minnen, G. Toderici, M. Covell, T. Chinen, N. Johnston, J. Shor, S. J. Hwang, D. Vincent, and S. Singh, ''Spatially adaptive image compression using a tiled deep network,'' in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2796–2800.
[34] L. Zhang, T. Yang, R. Jin, and Z.-H. Zhou. (2019). *Relative Error Bound Analysis for Nuclear Norm Regularized Matrix Completion*. [Online]. Available: http://jmlr.org/papers/v20/15-504.html
[35] K. Islam, L. M. Dang, S. Lee, and H. Moon, ''Image compression with recurrent neural network and generalized divisive normalization,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 1875–1879, doi: 10.1109/CVPRW53098.2021.00209.
[36] O. Schwartz and R. Coen-Cagli, ''Visual attention and flexible normalization pools,'' *J. Vis.*, vol. 13, no. 1, p. 25, Jan. 2013, doi: 10.1167/13.1.25.
[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, ''ImageNet large scale visual recognition challenge,'' *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[38] E. Agustsson and R. Timofte, ''NTIRE 2017 challenge on single image super-resolution: Dataset and study,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1122–1131, doi: 10.1109/CVPRW.2017.150.
[39] *True Color Kodak Images*. Accessed: Feb. 22, 2023. [Online]. Available: https://r0k.us/graphics/kodak/
[40] L. Theis, W. Shi, A. Cunningham, and F. Huszár, ''Lossy image compression with compressive autoencoders,'' in *Proc. 5th Int. Conf. Learn. Represent.*, 2017. [Online]. Available: https://arxiv.org/abs/1703.00395
[41] M. Abu Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, ''Rate-distortion balanced data compression for wireless sensor networks,'' *IEEE Sensors J.*, vol. 16, no. 12, pp. 5072–5083, Jun. 2016, doi: 10.1109/JSEN.2016.2550599.
[42] U. Sara, M. Akter, and M. S. Uddin, ''Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study,'' *J. Comput. Commun.*, vol. 7, no. 3, pp. 8–18, 2019, doi: 10.4236/jcc.2019.73002.
[43] A. Horé and D. Ziou, ''Image quality metrics: PSNR vs. SSIM,'' in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 2366–2369, doi: 10.1109/ICPR.2010.579.
[44] Z. Wang, E. P. Simoncelli, and A. C. Bovik, ''Multi-scale structural similarity for image quality assessment,'' in *Proc. 37th IEEE Asilomar Conf. Signals, Syst. Comput.*, Mar. 2002, pp. 1–5. Accessed: Feb. 21, 2023. [Online]. Available: https://ece.uwaterloo.ca/~z70wang/publications/msssim.html
[45] R. W. Hamming, *Coding and Information Theory*, vol. 11, no. 3. NJ, USA: Prentice-Hall, 1969.
[46] M. A. Razzaque, C. Bleakley, and S. Dobson, ''Compression in wireless sensor networks,'' *ACM Trans. Sensor Netw.*, vol. 10, no. 1, pp. 1–44, Dec. 2013, doi: 10.1145/2528948.
[47] D. Zordan, B. Martinez, I. Vilajosana, and M. Rossi, ''To compress or not to compress: Processing vs transmission tradeoffs for energy constrained sensor networking,'' 2012, *arXiv:1206.2129*.
[48] *MSP430 Family Mixed-signal Microcontroller Application Reports—Lutz Bierl—Google Books*. Accessed: Feb. 22, 2023. [Online]. Available: https://books.google.co.bw/books/about/MSP430_Family_Mixed_signal_Microcontroll.html?id=A0OPYgEACAAJ&redir_esc=y
[49] *1-Watt Long-Range OEM RF Modules | Digi International*. Accessed: Feb. 23, 2023. [Online]. Available: https://www.digi.com/products/embedded-systems/digi-xbee/rf-modules/sub-1-ghz-rf-modules/xtend-module

**BOSE A. LUNGISANI** (Member, IEEE) received the B.Eng. degree (Hons.) in computer engineering from the University of Essex, U.K., in 2007, and the M.Sc. degree in computer information systems from the University of Botswana, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electrical, Computer, and Telecommunications Engineering, Botswana International University of Science and Technology, Palapye, Botswana.

**ADAMU M. ZUNGERU** (Senior Member, IEEE) received the B.Eng. degree from the Federal University of Technology Minna, Nigeria, the M.Sc. degree from Ahmadu Bello University, Zaria, Nigeria, and the Ph.D. degree from Nottingham University, U.K. He received the Postgraduate Teaching Certificate from the Massachusetts Institute of Technology (MIT), USA, in 2014. He was a Research Fellow with MIT. In 2015, he was a Senior Lecturer and the Head of the Department of Electrical and Electronics Engineering, Federal University Oye-Ekiti, Nigeria. He is currently a Professor and the Head of the Department of Electrical, Computer, and Telecommunications Engineering, Botswana International University of Science and Technology (BIUST). He is an Inventor of a Termite-Hill routing algorithm for wireless sensor networks and a method and system for sorting diamonds. He has three of the Patent applications registered with the World Intellectual Property Organization (WIPO) and has been granted three patents on the invention. He has authored five academic books and more than 70 international research articles in reputable journals, including the IEEE Systems Journal, IEEE Internet of Things Journal, IEEE Access, and *Journal of Network and Computer Applications* (Elsevier), with more than 1000 citations and an H-index of 15. He served as the Chairperson for the IEEE Botswana Section, from 2019 to 2020. He is serving as an Associate Editor for IEEE Access. He has served as an International Reviewer for IEEE Transactions on Industrial Informatics, IEEE Sensors Journal, IEEE Access, IEEE Transactions on Mobile Computing, IEEE Transactions on Sustainable Computing, and *Journal of Network and Computer Applications* (Elsevier). He holds the Botswana Patent, the Russian Federation Patent, and the Republic of South Africa Patent on the Method and System for Sorting Diamonds. He is a registered Engineer with the Council for the Regulation of Engineering in Nigeria (COREN) and a Professional Engineer registered with the Botswana Engineers Registration Board (ERB) and the Association for Computing Machinery (ACM), USA.

**CASPAR K. LEBEKWE** (Member, IEEE) received the M.Eng. degree in electronics and communications engineering and the Ph.D. degree in electrical and electronics engineering from the University of Bath, in 2008 and 2018, respectively, sponsored by the General Lighthouse Authorities. His Ph.D. project was focused on eLoran Service Volume Coverage Prediction. He is currently a Senior Lecturer with the Botswana International University of Science and Technology, where he teaches optical communications, antennas and propagation, discrete mathematics, telemetry, remote control, and electromagnetic field theory.

**ABID YAHYA** (Senior Member, IEEE) received the bachelor's degree in electrical and electronic engineering, major in telecommunication, from the University of Engineering and Technology Peshawar, Peshawar, Pakistan, and the M.Sc. and Ph.D. degrees in wireless and mobile systems from Universiti Sains Malaysia. He began the career on an engineering path, which is rare among other researcher executives. He is currently with the Botswana International University of Science and Technology. He is a Professional Engineer certified by the Botswana Engineers Registration Board (ERB). He has many research publications in numerous reputable journals, conference articles, and book chapters. He received several awards and grants from various funding agencies and supervised several master's and Ph.D. candidates. His recent four books *Emerging Technologies in Agriculture, Livestock, and Climate* (Springer, 2020), *Mobile WiMAX Systems: Performance Analysis of Fractional Frequency Reuse* (CRC Press | Taylor & Francis, 2019), *Steganography Techniques for Digital Images*, *LTE-A Cellular Networks: Multi-hop Relay for Coverage, Capacity, and Performance Enhancement* (Springer International Publishing, July 2018 and January 2017), and are being followed in national and international universities.