

Research Article

Image Compression for Wireless Sensor Network: A Model Segmentation-Based Compressive Autoencoder

Xuecai Bao ^{1,2}, Chen Ye ^{1,2}, Longzhe Han ^{1,2} and Xiaohua Xu ³

¹Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing, Nanchang Institute of Technology, 330099 Nanchang, Jiangxi, China

²School of Information Engineering, Nanchang Institute of Technology, 330099 Nanchang, Jiangxi, China

³Jiangxi Academy of Water Science and Engineering, 330029 Nanchang, Jiangxi, China

Correspondence should be addressed to Xuecai Bao; lx97821@126.com

Received 12 August 2022; Revised 27 September 2022; Accepted 16 February 2023; Published 25 October 2023

Academic Editor: Yuanlong Cao

Copyright © 2023 Xuecai Bao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at the problems of image quality, compression performance, and transmission efficiency of image compression in wireless sensor networks (WSN), a model segmentation-based compressive autoencoder (MS-CAE) is proposed. In the proposed algorithm, we first divide each image in the dataset into pixel blocks and design a novel deep image compression network with a compressive autoencoder to form a compressed feature map by encoding pixel blocks. Then, the reconstructed image is obtained by using the quantized coefficients of the quantizer and splicing the decoded feature maps in order. Finally, the deep network model is segmented into two parts: the encoding network and the decoding network. The weight parameters of the encoding network are deployed to the edge device for the compressed image in the sensor network. For high-quality reconstructed images, the weight parameters of the decoding network are deployed to the cloud system. Experimental results demonstrate that the proposed MS-CAE obtains a high signal-to-noise ratio (PSNR) for the details of the image, and the compression ratio at the same bit per pixel (bpp) is significantly higher than that of the compared image compression algorithms. It also indicates that the MS-CAE not only greatly relieves the pressure of the hardware system in sensor network but also effectively improves image transmission efficiency and solves the deployment problem of image monitoring in remote and energy-poor areas.

1. Introduction

The wireless sensor network (WSN) is widely deployed in many applications, such as ecological environment monitoring, water quality monitoring, and mine safety monitoring [1–4]. Image monitoring in WSN is an important topic in the monitoring field. It has a visual effect and can provide image information to the management platform. However, the massive amounts of image information cause network congestion. Although some novel technologies of congestion control and packet reordering algorithms are proposed to solve this problem [5–7], the image compression technology in the image sensor device has attracted an increasing attention and is considered as an effective solution in terms of improving energy and transmission efficiency. Until now,

many image compression algorithms for WSN have been proposed [8]. However, owing to the functional limitations of hardware equipment for WSN and the high energy consumption of image transmission, it also poses significant challenges to WSN deployment in remote areas with limited energy.

For traditional image compression techniques in WSN, the research on image compression can be categorized as lossless and lossy image compression. JPEG [9] and JPEG 2000 [10] are typical representations of lossy image compression and have been widely applied to WSN. Aiming at transmission efficiency and memory saving, lossy image compression draws more attention in WSN than lossless image compression. In particular, the emergence of image compression techniques based on Deep Learning Models (DLMs) provides a completely new direction [11].

In the field of deep learning image compression, a great number of efforts have been devoted to improving the resolution of reconstructed compressed images. Using the Convolutional Neural Network (CNN) structure, the methods of training Compact CNN (ComCNN) and Reconstructed CNN (RecCNN) are proposed simultaneously in [12]. ComCNN mainly optimizes the compression effect, and RecCNN is used to reconstruct high-quality images. Kuang et al. propose a new model for a single-image Super-Resolution (SR) task by utilizing the design of densely connected convolutional networks (DenseNet) [13], which has a lightweight structure and is extensively evaluated on datasets. They attempt to optimize the deep network and adjust parameter settings to achieve trade-offs between image resolution and running time. The advantage of deep CNNs lies in its powerful capability to handle large-scale image datasets. These works, on the other hand, are complex, making them difficult to deploy in WSN edge devices.

Currently, autoencoders based on CNN have become a significant research interest, which are more simple than deep CNN in network architecture. In an earlier period, most learning autoencoders were used for dimensionality reduction for high-efficiency image compression. On the other hand, the autoencoder, with its relatively simple network architecture, is also faster than CNN in the inference process. Huang et al. propose a multiscale autoencoder (MSAE) to improve the compression effect and adopt the generative adversarial network (GAN) with multiscale discriminators to perform the end-to-end trainable rate-distortion optimization. This framework achieves excellent reconstruction effects at a low bit rate [14]. Cheng et al. use Principal Component Analysis (PCA) to generate an energy-efficient representation for the CAE architecture to achieve high coding efficiency, and the algorithm mainly preserves the principal components in the model training process and greatly improves the compression ratio [15]. Furthermore, when compared to the traditional deep CNN architecture, CAE-based image compression is a complete deep learning architecture that reduces its own network layers [16]. Based on an autoencoder, the authors in [17] append quantization and entropy rate estimation to the CNN structure. Furthermore, in [18], a three-dimensional convolutional autoencoder (3D-CAE) is proposed, which has greatly improved the reconstruction precision. All these algorithms mentioned above improve the network architecture of the compressive autoencoder, which performs well in reconstructed image detail extraction. In addition, the end-to-end architecture also offers the possibility of deployment for WSN. However, some of these algorithms will occupy a great deal of memory at runtime, which impacts the efficiency of image monitoring for WSN.

Moreover, most of the above-mentioned works focus on the optimization of rate distortion, visual effect, and image compression ratio, but the limited memory capacity of the hardware system in the WSN is not considered. Aiming to solve the above-mentioned problems, we propose a novel MS-CAE algorithm to satisfy the demands of WSN image monitoring in remote areas. The main contributions of the proposed MS-CAE algorithm areas are as follows:

- (1) To address the issue of large networks not being deployed in sensor nodes due to functional constraints, we proposed a model segmentation-based compressive autoencoder
- (2) We proposed an asymmetric architecture for the encoding and decoding networks in MS-CAE. We design the simplified encoding network and the more complex decoding network properly to improve the resolution of the reconstructed compressed image

The rest of this paper is organized as follows: Section 2 describes the related work of image compression. Section 3 presents the principles of the architecture of a compressive autoencoder (CAE). In Section 4, we present a novel MS-CAE image compression algorithm for image monitoring in WSN. Section 5 evaluates the performance of the proposed MS-CAE algorithm, followed by concluding remarks in Section 6.

2. Related Work

2.1. Image Compression Based on Deep Learning. Recent works on the CNN network have made contributions to image compression, especially in DLMS. To achieve high-quality image compression at low bit rates, Jiang et al. propose two CNNs as the pre- and postprocessing steps [12]. Toderici et al. utilize a long short-term memory (LSTM) recurrent network to compress small patch images and also adopt quantization to realize the decrease in the encoding coefficient scale [19]. Li et al. are motivated by the character of the local information content in a single image, and they propose learning convolution networks for content-weight image compression to solve the problem of encoder rate distortion [20]. The DSSLIC framework is used to obtain the semantic segmentation map of the input image and encode it as the base layer of the bitstream [21]. Sushma and Fatimah improve the reconstructed image detail information by predicting chroma at the decoder, which serves as side information for decoding chroma components [22]. These algorithms optimize the quality of reconstructed images in various aspects. For instance, these authors make great progress in the aspects of high compression ratio, compression efficiency, high-resolution image, and detail image reconstruction, whereas the operations mentioned before usually consume a large amount of storage space in computer equipment.

2.2. Image Compression Based on CAE. There exist numerous works on variants of compressive autoencoders (CAE). In different ways, these techniques reduce the distortion of the reconstructed image for lossy image compression. In [23], Shi et al. introduce an efficient subpixel convolution layer learned from an array of upscaling filters to upscale the final low-resolution feature maps into the high-resolution output image. Inspired by the work of Shi et al. [23], Theis and Shi [16] utilize the CAE structure by optimizing quantization and entropy rate estimation to acquire excellent training model results. Following the above architectures, the authors

in [17] append a nonlinear analysis transformation, a uniform quantizer, and a nonlinear synthesis transformation to a convolutional network. Cheng et al. train the improved CAE architecture to generate a more compact representation of feature maps, and they optimize the rate-distortion loss function of CAE to improve image-coding efficiency [15]. An energy compaction-based image compression using a convolutional autoencoder is proposed. This work optimizes the CAE architecture by decomposing it into several down- and upsample operations and proposes a normalized coding gain metric in neural networks [24]. Based on the previous high-precision CAE, Chong et al. [18] exploit a 3D-CAE architecture that precisely achieves end-to-end joint spectral-spatial compression and reconstruction. These works in the literature [15, 18, 24] primarily employ a compact compression network and various upsample operations to trade-off the optimization of the compression ratio and rate distortion.

2.3. Image Compression Work in the Field of WSN. Efficient DLM models will be applicable to the interconnection between hardware systems and cloud devices. From the requirements of image monitoring, our work is divided into two aspects: edge devices and cloud-based devices. An edge device is used to obtain image information [25] and cloud-based device analysis image-coding coefficients [26]. Ding et al. deploy DLMs to edge devices and cloud-based devices, which advance the running speed of the corresponding device [27]. However, high-performance DLMs usually require numerous storage and computing resources, which make the deployment work difficult on an edge device. To solve this problem, many researchers attempt to improve the efficiency of DLMs by pruning the convolution layers or convolution kernels [28, 29]. Some works combine gradient-based optimization [30, 31] and residual learning [32] to implement steps to speed up inference in image compression algorithms. These works have made great progress toward obtaining excellent effects. Because a cloud-based device is deployed near monitor operators, it is technically reasonable for a decoder to obtain high-resolution images.

Through comparison and analysis, we found that the CAE architecture is suitable for image compression in WSN and presents excellent performance. Furthermore, CAE is simpler than CNN in network architecture. Therefore, we design a novel network architecture based on CAE and propose an image compression algorithm based on a model segmentation-based compressive autoencoder (MS-CAE), which not only segments the model to alleviate the pressure of the hardware system and promote the transmission efficiency of the sensor network but also improves the image quality and monitoring energy efficiency, so as to achieve the purpose of improving the energy efficiency for WSN image monitoring.

3. Architecture of Compressive Autoencoder

The network architecture of a compressive autoencoder consists of three modules: an encoder E , a decoder D , and a quantizer Q :

$$E : \mathbb{R}^N \longrightarrow \mathbb{R}^M, \quad (1)$$

$$D : \mathbb{R}^M \longrightarrow \widehat{\mathbb{R}}^N, \quad (2)$$

$$Q : e \longrightarrow Q(e). \quad (3)$$

The encoder E maps the original image $x \in \mathbb{R}$ to a latent representation $e \in E(x)$. The quantizer Q maps each element of e to $Q(e)$, which generates the quantized coefficients $\tilde{e} = Q(e)$. Then, the decoder D attempts to reconstruct the original image $\hat{x} = D(\tilde{e})$ from the quantized coefficients \tilde{e} .

Figure 1 clearly illustrates the flow diagram of the CAE network. The original image is gradually compressed by the convolution layers to generate compressed data in the encoder. Then, the compressed data is quantized through the quantizer. Subsequently, the decoder reconstructs the image through the decompressed data.

To assist understanding, we assume that the original image dataset was encoded using linear mapping and a nonlinear activation function. As a result, the process of an encoder producing compressed data can be defined as

$$\mathbb{R}^M = g(W_{i+2}\mathbb{R}^N + b_{i+2}), \quad (4)$$

where \mathbb{R}^N and \mathbb{R}^M represent the original image and compressed data of the original image, respectively. The weight and the bias of the Conv3 layer are W_{i+2} and b_{i+2} , respectively. Moreover, the corresponding node activation function is defined as $g(\cdot)$.

After the encoding process, the quantizer transforms compressed data into decompressed data. The decoder obtains the decompressed data and calculates the reconstructed image sample. Obviously, the decoding process is the inverse of the encoding process, which is defined as

$$\widehat{\mathbb{R}}^N = g\left(W_{j+2}^T\mathbb{R}^M + b_{j+2}\right), \quad (5)$$

where $\widehat{\mathbb{R}}^N$ is the reconstructed image sample. The weight and the bias of the DeConv3 layer are W_{j+2} and b_{j+2} , respectively.

Next, we introduce the quantizer in Figure 1. The quantization is one of the approaches to decrease the complexity of encoding coefficients. The encoding network exploits the rounding function in the early period of the deep neural network. The rounding function is used to obtain the nearest integer of the coefficient. It is denoted as

$$f(x) = \text{round}(x, d), \quad (6)$$

where x and d are the coefficient and accuracy retained after the decimal point, respectively. Thus, to quantize the coefficients in more detail, Agustsson et al. in [33] adopt the uniform scalar quantizer, which is similar with the rounding function, as follows:

$$f(x_i) = \text{round}(x_i), \quad (7)$$

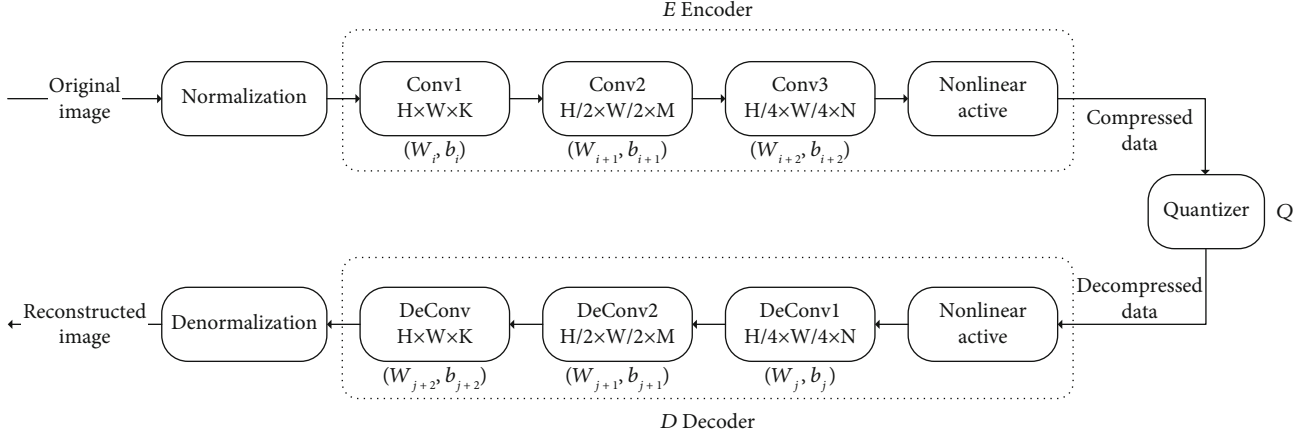


FIGURE 1: The network architecture of CAE.

where x and i are the coefficient and the number of equal points, respectively. Accordingly, $f(x_i)$ represents quantization through equipartition to the nearest interval.

Moreover, Toderici et al. in [19] propose a stochastic rounding function of binarization, which is written as

$$\{y\} \approx \lfloor y \rfloor + \varepsilon, \quad \varepsilon \in \{0, 1\}, \quad (8)$$

$$P(\varepsilon = 1) = y - \lfloor y \rfloor. \quad (9)$$

The stochastic rounding function is different from above-mentioned two rounding functions. The operation mainly uses the round-down method, namely, the integer of not more than x . Furthermore, the stochastic rounding function obtains round results by expectation x and random probability ε .

In the process of quantization, the rounding function exists more or less as a deviation. Therefore, rounding and the uniform scalar quantizer have more deviations. Thus, CAE uses the loss function to evaluate the train loss. From the above description, we know that the input original image sample is $X : x \in \mathbb{R}^N$, and the output reconstructed image is $\hat{X} : \hat{x} \in \hat{\mathbb{R}}^N$. CAE evaluates the loss rate between \mathbb{R}^N and $\hat{\mathbb{R}}^N$ by the cross-entropy loss function and mean square error (MSE) loss function. These two loss functions are defined as

$$J(X, \hat{X}) = - \sum_{i=1}^n [x_i \log(\hat{x}_i) + (1 - x_i) \log(1 - \hat{x}_i)]. \quad (10)$$

$$J(X, \hat{X}) = \frac{1}{2} \sum_{i=1}^n \|\hat{x}_i - x_i\|_2^2. \quad (11)$$

Following the above analysis, the loss function is minimized to acquire an excellent trained result, which is written as

$$\arg \min_{W, b} J(W, b). \quad (12)$$

4. MS-CAE Architecture and Implementation Method

In this section, we propose an image compression network architecture based on a model segmentation-based compressive autoencoder (MS-CAE) for WSN. We first present the proposed MS-CAE framework. Then, the corresponding implementation process is described. Finally, we provide the achievement of model segmentation and weight deployment for WSN.

4.1. MS-CAE Framework for WSN. The existing image compression algorithms based on CAE mainly focus on compression performance. However, few algorithms based on CAE consider the limited computing resources and the practical deployment of WSN.

Therefore, we present a novel MS-CAE framework to solve two problems:

- (1) The image sensor node in the WSN cannot carry a complete trained image dataset for the deep neural network
- (2) A cloud-computing platform makes it difficult to parse and reconstruct high-quality images from a simple network with insufficiently encoded data

We illustrate the proposed MS-CAE framework for WSN in Figure 2. Firstly, we divide the image dataset into several small pixel blocks by preprocessing the image. Then, the encoding network implements image compression through image feature extraction, quantification, and data compression. Subsequently, the decoding coefficients are obtained by the quantizer in the decoding network, and then, they are used to reconstruct images by the data filtering of the residual block network. In the implementation process, the obtained weight parameters by training the MS-CAE network are divided into two parts, namely, the encoding and decoding networks. Accordingly, the weight parameter information in the encoding and decoding networks is deployed to edge devices and cloud devices, respectively.

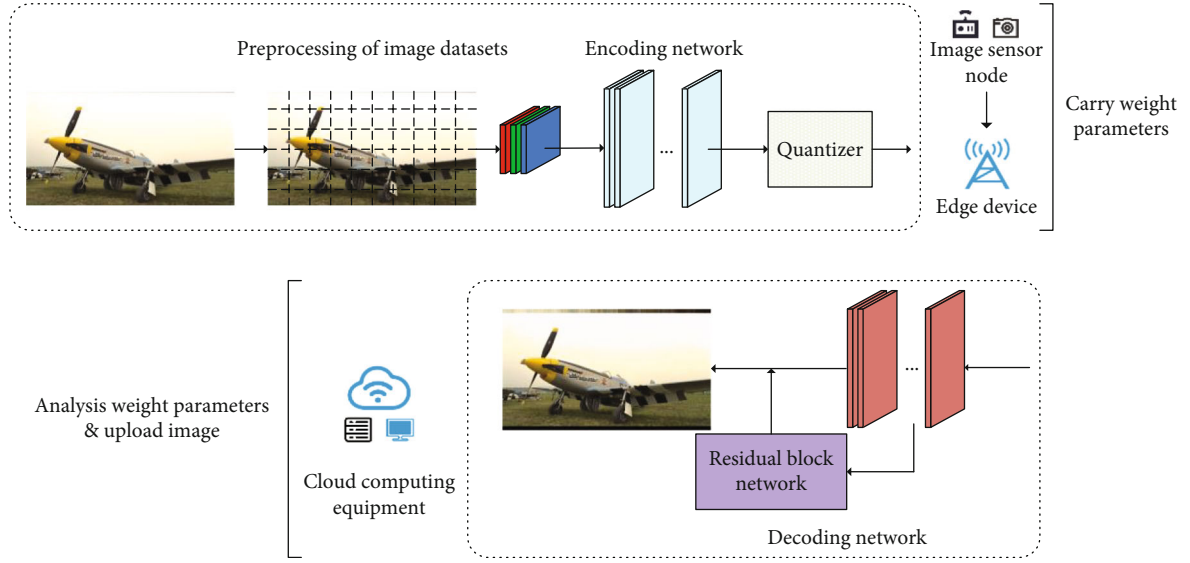


FIGURE 2: MS-CAE framework of image compression for WSN.

4.2. MS-CAE Network Architecture and Implementation Process

4.2.1. MS-CAE Network Architecture. The encoding and decoding networks in traditional CAE architecture are symmetrical. The symmetrical CAE architecture, on the other hand, necessitates a relatively high level of computation complexity and storage space in edge devices. It is unsuitable for an edge device with limited resources. To satisfy the demand for the edge device and cloud device, we propose a novel asymmetrical MS-CAE architecture, which is shown in Figure 3. In the proposed MS-CAE architecture, we simplify the encoding network. Meanwhile, we increase the complexity of the decoding network to improve the resolution of the reconstructed compressed image. The detailed description is as follows.

In Figure 3, after the above preprocessing data based on pixel block segmentation, each picture is decomposed into 60 three-channel (RGB) pixel blocks. The encoding and decoding networks generate three kinds of feature maps by the convolution operation. These feature maps are 128×128 , 64×64 , and 32×32 .

In the MS-CAE network, there are five convolution kernel units. As shown in Table 1, “ConvK/S P” stands for a convolution layer with kernel size $K \times K$, a stride of S and a reflection padding size of P . For instance, “Conv5/2 p1.5” is a convolution unit with 5×5 convolution kernel size, 2-stride size, and 1.5 padding size.

Moreover, the reflection-padding mode is different from zero-padding. The input matrix of the reflection-padding mode is (N, C, H_{in}, W_{in}) , and the output matrix of the reflection-padding mode is (N, C, H_{in}, W_{out}) , where N is the number, C is the channel number, and H and W are the matrix height and width, respectively. The corresponding padding mode is written as

$$H_{out} = H_{in} + padding_top + padding_bottom, \quad (13)$$

$$W_{out} = W_{in} + padding_left + padding_right. \quad (14)$$

Furthermore, Figure 4 illustrates the zero-padding mode and the reflection-padding mode. Actually, the filled coefficients in the reflection-padding mode follow the sequence of left, right, top, and bottom. Since most deep networks adopt the zero-padding mode, the boundary pixels cannot accurately extract the coefficients through convolution operations, which causes the boundary-blurring effect. Thus, in our proposed MS-CAE, we use reflection padding to compensate for pixel gaps caused by boundary-blurring effects. Moreover, by utilizing the reflection padding in the training process, the boundary of the reconstructed image pixel blocks does not cause pixel cracks and improves the overall image quality.

4.2.2. Implementation Process

(1) *Preprocessing Data: Pixel Block Segmentation.* The purpose of pixel block segmentation is to divide the training images with pixel 720p ($1280 \times 720 \times 3$) into pixel 128p ($128 \times 128 \times 3$). The specific operation is as follows: We first fill the width of the image ($1280 \times 720 \times 3$ to $1280 \times 768 \times 3$). Then, the images are divided into small pixel blocks ($128 \times 128 \times 3$). Subsequently, the batches of patches are packed into the CAE-training network.

(2) *Encoder Network.* In the proposed MS-CAE in Figure 3, the encoder network consists of 9 convolutional layers that contain the labeled different convolution kernel units and the subsequent nonlinear operation of the parameterized rectified linear units. We adopt PReLU as an active function, which is defined as

$$\text{PReLU}(x_i) = \begin{cases} x_i, & \text{if } x_i > 0, \\ a_i x_i, & \text{if } x_i \leq 0, \end{cases} \quad (15)$$

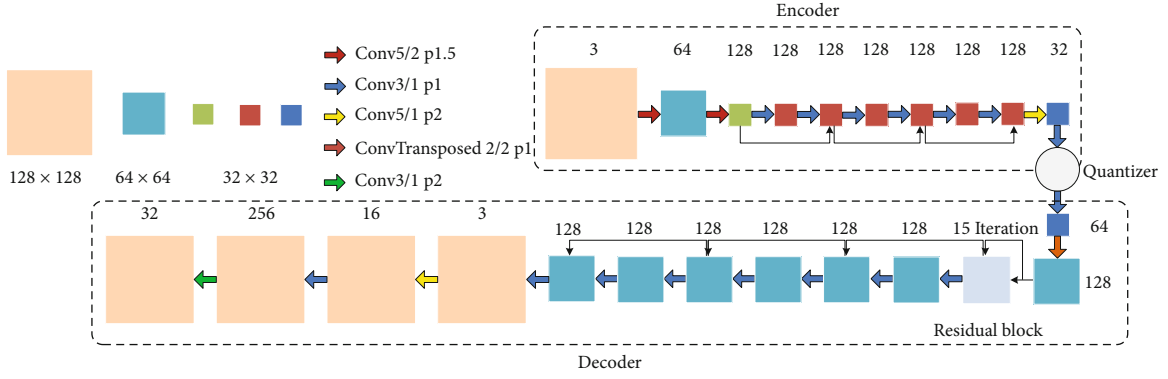


FIGURE 3: MS-CAE network architecture for enhancing decoder feature extraction.

TABLE 1: Description of different convolution kernel units.

Name	Kernel size ($K \times K$)	Stride (S)	Padding size (P)
Conv5/2 p1.5	5×5	2	1.5
Conv3/1 p1	3×3	1	1
Conv5/1 p2	5×5	1	2
Conv3/1 p2	3×3	1	2
ConvTransposed 2/2 p1	2×2	2	1

where x_i is the input of the nonlinear activation function in the matching channel i and a_i is the gradient of the negative axis of the activation function.

PReLU's nonlinear operation is conducive to the extraction and retention of negative coefficients. Through the feature linear superposition of 128 channels 32×32 with two similar Conv3/1 p1 convolution layers in three layers, the feature matrix coefficients with a lower frequency are retained as much as possible for image feature extraction.

(3) *Decoder Network.* The decoder of MS-CAE in Figure 3 reconstructs the 32×32 compressed feature maps obtained by the encoder. The function of the convolution layer between the encoder and the decoder network is to transform 32×32 feature blocks into 64×64 feature blocks before the process of the residual block network. As shown in Figure 3, following 15 iterations of the residual block network, 6 convolution layers are applied to increase the sample. The residual block network in the decoder relieves the gradient-vanishing problem, which efficiently avoids degradation in the next network layer.

The detailed description of the residual block network is shown in Figure 5. It consists of three convolution layers. Both the first and third convolution layers employ an 11-convolution kernel with a stride length of 1. The second layer uses a 3×3 -convolution kernel with a stride length of 1. Three convolution layers are normalized and nonlinearly activated by the PReLU function. Following the filtering of the feature coefficients by the residual block network, the

64×64 feature maps of 128 channels with six Conv3/1 p1 convolution layers are used to effectively retain the nonredundant and high correlation coefficients as the foundation for reconstructing the image. Finally, the decoder obtains a reconstructed image by using 4 convolution layers.

4.3. *Model Segmentation and Weight Deployment for WSN.* As shown in Figures 2 and 3, we know that the proposed MS-CAE is divided into two parts, namely, the encoder network and the decoder network. Furthermore, the scale of the designed encoder network is relatively small, and the decoding network is more complex than the encoder network. The purpose of this design is to consider the resource limitations of an image monitoring node for the WSN in remote areas. We train the novel MS-CAE network model and extract the weight parameters of the whole model after several periodic iterations. The weight parameters of the well-trained model are divided into two parts, the weight parameters of the encoding network and the decoding network. For the practical deployment of image monitoring for WSN, we require the proposed MS-CAE model to be segmented. The encoder and decoder networks in MS-CAE are deployed to the edge device and cloud-computing device, respectively.

The divided weight parameters are then loaded into the edge device's encoding network and the cloud-computing device's decoding network. For remote monitoring, an edge device is used to collect and compress image data from sensor nodes, which are based on resource-constrained micro-controllers. A cloud-computing device usually has strong computing capability and large storage capacity. Thus, a cloud-computing device is used to parse and restore a large number of reconstructed images.

Therefore, in order to reduce the burden of the edge device in WSN, the relatively small-scale encoding network model parameters are deployed to the edge device. In addition, to improve the quality of the reconstructed image, the weight parameters of the more complex decoding network model are deployed to the cloud device.

5. Experiment Result

5.1. *Dataset.* Considering the deployment work of the edge device in WSN, for our experiments, we chose a relatively

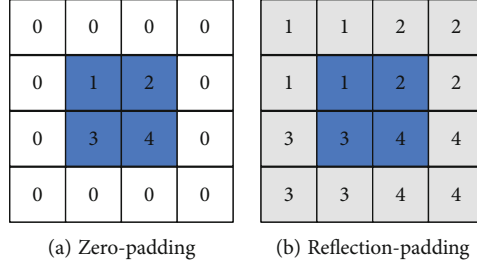


FIGURE 4: Two kinds of padding mode.

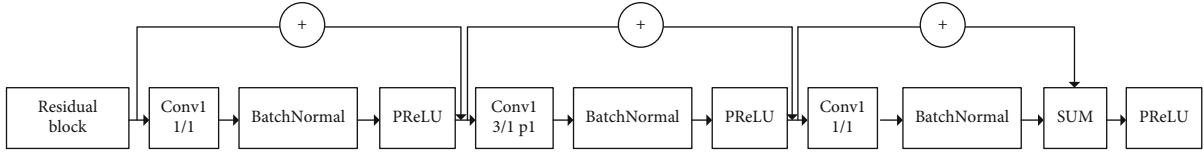


FIGURE 5: Residual block network.

small image dataset (yt_small_720p) to train and evaluate the performance of the proposed MS-CAE. The dataset covers seven categories: portrait, cartoon, game, natural scenery, advertisement pattern, city scene, and medical image. Furthermore, it collects 2285 images with a resolution of 1280×720 . According to the above introduction of pixel block division, we train the proposed MS-CAE network using 60 pixel blocks for each image. In the testing process, we use the Kodak 720p dataset with high-resolution photographs. All procedures are implemented in PyTorch. Each model is trained for 143 epochs on the NVIDIA GeForce RTX 2070 with Max-Q Design GPU.

5.2. Evaluation Indicators. To verify the effectiveness of the proposed MS-CAE, we study the performance with respect to mean square error (MSE), average loss, peak signal-to-noise ratio (PSNR), and structural similarity index measurement (SSIM) for reconstructed compressed image quality. These evaluation indicators are written as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^m w_i (y_i - \hat{y}_i), \quad (16)$$

$$\text{Avgloss}_{\text{per patch}} = \frac{1}{60} \sum_i \text{MSEloss}_{\text{per patch}}, \quad (17)$$

$$\text{PSNR} = 10 \times \log_{10} \left(\frac{(2^n - 1)^2}{\text{MSE}} \right), \quad (18)$$

$$\text{SSIM}(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (19)$$

where n is the number of samples and y_i and \hat{y}_i are the real value and predict value, respectively. In (19), μ_x and μ_y refer to the average values of x and y , respectively. Accordingly, the σ_x is the variance of x and σ_y is the var-

iance of y . σ_{xy} is the covariance of x and y . The $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ are constants used to maintain stability, where L is the dynamic range of the pixel value and $k_1 = 0.01$ and $k_2 = 0.03$.

5.3. Results

5.3.1. Evaluation for Average Loss Rates. The mean square error is calculated by (16) to measure the error between the real coefficients and the reconstructed coefficients. Average loss reflects the difference in loss between the original image and the reconstructed compressed image. Then, the average loss of the whole image is evaluated by calculating the average loss of 60 pixel blocks by (17). The training loss of a single image can be estimated by averaging the loss of 60 pixel blocks. In Figure 6, we present the average loss of each pixel block between the MS-CAE and CAE models in training over 143 epochs. As shown in Figure 6, the average loss of MS-CAE gradually stabilized and was less than CAE after 80 training epochs. Namely, the training effect of each pixel block of our proposed MS-CAE is better than that of CAE. Moreover, Figure 7 shows the comparison result for the average loss for 24 Kodak images in the test dataset. The result in Figure 7 shows that the average loss of the proposed MS-CAE is obviously lower than that of the CAE.

5.3.2. Quality Evaluation of Reconstructed Images. According to the indicators of PSNR and SSIM in (18) and (19), we evaluate the quality of the reconstructed compressed image for our proposed MS-CAE by setting different bits per pixel (bpp). PSNR is a comprehensive, objective image evaluation indicator that is based on the difference between the corresponding pixels. SSIM focuses on full-reference image quality, which evaluates image similarity based on luminance, contrast, and structure. As a result, these two indicators evaluate the quality of reconstructed compressed images from different perspectives, with higher indexes indicating less distortion. Furthermore, we compute the average PSNR and SSIM values for 24 Kodak images to validate the

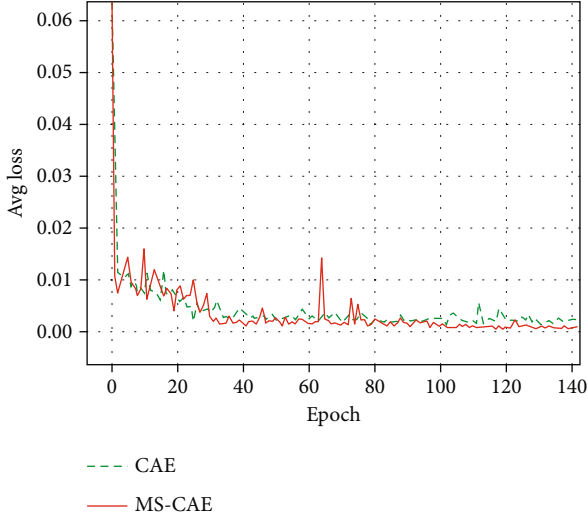


FIGURE 6: The average training loss of yt_small_720p dataset.

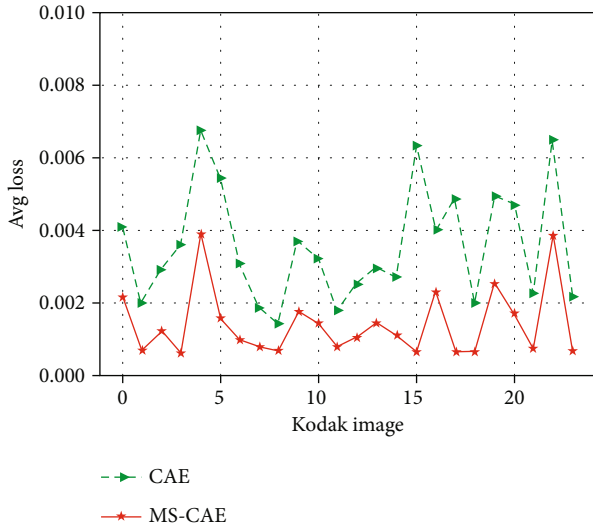


FIGURE 7: The average testing loss of 24 Kodak images.

performance of five algorithms. The bpp represents the ratio of the number of valid bits in a compressed image to the total number of pixels. Thus, the image compression ratio can also be reflected by the bpp value. The higher bpp value represents a lower image compression ratio and vice versa.

To further verify the performance of the reconstructed compressed image, we compare MS-CAE with JPEG, JPEG 2000, CAE, and Toderici's Full-Resolution Image Compression with Recurrent Neural Networks (FRIC-RNN) [19]. Figure 8 depicts the PSNR comparison value of reconstructed images at various bpp. It can be seen that the PSNR values of MS-CAE are significantly higher than those of JPEG and FRIC-RNN image compression algorithms. Between 0.1042 and ~ 0.7083 bpp, the reconstructed image quality of MS-CAE is better than that of CAE and JPEG 2000. The results also show that the proposed MS-CAE outperforms other algorithms in terms of high compression ratio. Although the PSNR of MS-CAE is slightly lower than

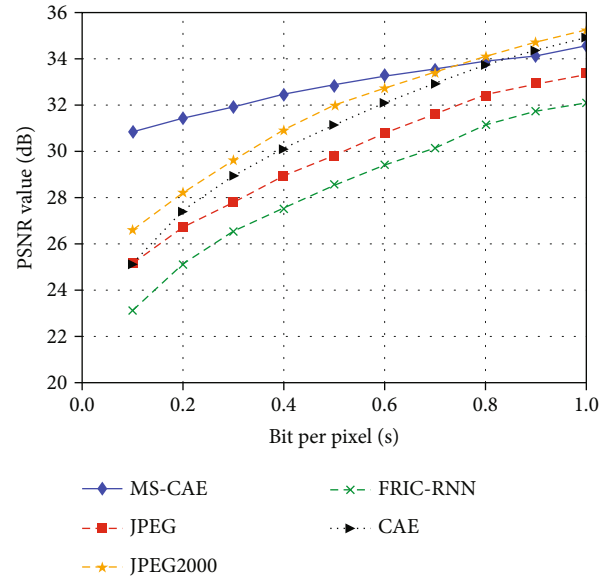


FIGURE 8: PSNR comparison for different bpp.

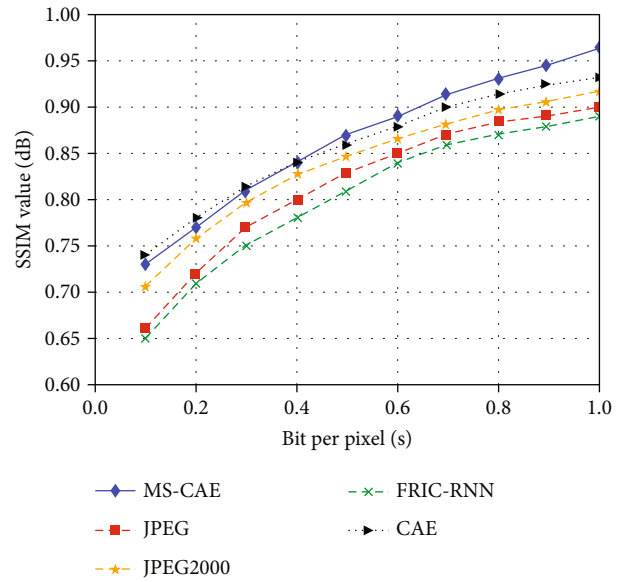


FIGURE 9: SSIM comparison for different bpp.

that of CAE and JPEG 2000 in the range of 0.7083~1.0 bpp, the PSNR performance in the range of 0.7083~1.0 bpp which represents a low compression ratio is not a cause for concern for WSN. Furthermore, Figure 9 illustrates the SSIM values of reconstructed images at different bpp. Figure 9 shows that the proposed MS-CAE's structural similarity is greatly improved in the range of 0~1.0 bpp and only slightly lower than that of CAE in the range of 0~0.4 bpp. The reason is that since the proposed MS-CAE algorithm adopts residual block network iterations in the decoding process, it can reduce the network generalization to a small range and is conducive to the feature extraction of high-correlation coefficients.



(a) CAE (PSNR: 28.67 dB, SSIM: 0.8301)



(b) JPEG (PSNR: 28.35 dB, SSIM: 0.76)



(c) MS-CAE (PSNR: 29.94 dB, SSIM: 0.8718)



(d) JPEG 2000 (PSNR: 30.15 dB, SSIM: 0.8215)

FIGURE 10: Visual effect comparison of Kodak image at 0.3125 bpp.

Therefore, from the above results, our proposed MS-CAE not only improves the decoding network performance of the reconstructed compressed image but also achieves the low-complexity requirement of the encoding network for energy-limited WSN deployment in remote areas.

5.3.3. Visual Effect of Reconstructed Image. In this section, we present the comparison of visual effects between MS-CAE and CAE, JPEG, and JPEG 2000 at 0.3125 bpp for reconstructed images using the Kodak image dataset. The overall comparison results are shown in Figure 10. It can be seen from Figure 10 that the visual effect of the proposed MS-CAE algorithm is the best at a low 0.3125 bpp. This is because the MS-CAE effectively avoids the boundary-blurring effect through reflection-padding. The visual effect of the JPEG algorithm has severe image information distortion. The reason for the phenomenon is that the JPEG algorithm uses an 8×8 matrix of the Discrete Cosine Transform (DCT) to produce a boundary-blurring effect when the pixel blocks are spliced. The original image is compressed by the traditional CAE architecture in Figure 10(a). We can clearly see that the chroma and pixels of the reconstructed compressed image are severely distorted at 0.3125 bpp. We use the JPEG algorithm to compress and reconstruct the same image, as shown in Figure 10(b). Clearly, the PSNR of the reconstructed image in Figure 10(b) is higher than that of the CAE in Figure 10(a). However, because of the boundary-blurring effect caused by the DCT, the SSIM value of JPEG in Figure 10(b) is slightly lower than that of CAE in

Figure 10(a). The visual effect of the reconstructed image for JPEG is similar to that of CAE, as shown in Figures 10(c) and 10(d). JPEG 2000's vision effects are also comparable to the proposed MS-CAE. This is because the overall vision effect of the JPEG 2000 algorithm improves significantly as a result of the algorithm's use of the preprocessing procedure, coding, and quantization mode. Furthermore, by utilizing the residual block network and sufficient train epochs, the proposed MS-CAE algorithm avoids block effects and maintains detail elements.

In order to further verify the performance of restoring the image detail texture part, we take the character image in the Kodak image dataset as an example, and the comparison results are shown in Figure 11. Figures 11(a)–11(d) depict the visual effects of CAE, JPEG, MS-CAE, and JPEG 2000, respectively. Figures 11(a) and 11(b) show that the reconstructed image details are not very clear. Their effects in Figures 11(a) and 11(b) are worse than those of MS-CAE and JPEG 2000 in Figures 11(c) and 11(d). From Figures 11(a)–11(d), we know that the proposed MS-CAE algorithm is much clearer in terms of eyelash and hair texture than CAE, JPEG, and JPEG 2000 and has a much higher SSIM value than other algorithms while still maintaining good PSNR performance.

5.3.4. Complexity Analysis of Algorithm. We know from the above sections that our proposed MS-CAE clearly distinguishes itself from the traditional symmetric CAE architecture, and the corresponding encoder and decoder networks



FIGURE 11: Visual effect comparison of Kodak image details at 0.3125 bpp.

are asymmetric architectures. The purpose of designing the asymmetric architecture is to reduce the parameters of the encoder network for the deployment of an edge device in WSN and to utilize the resource advantages of a cloud-computing device. The encoder network of the proposed MS-CAE reduces the number of network layers, channels, and feature iterations and further improves the computation complexity of image compression. Then, the decoder utilizes three layers of a small residual block network to solve the problem of parameter redundancy and insufficient analytical accuracy so that the quality of the reconstructed image is improved. To analyze the computing complexity of the proposed MS-CAE, we evaluate the average running time of the above-mentioned algorithms in the same experimental environment. The results are shown in Table 2.

As shown in Table 2, the average running time of encoding an image with the proposed MS-CAE is shorter than that of JPEG and JPEG 2000 when using the same computing

TABLE 2: The average running time of encoding an image for different image compression algorithms.

Algorithm	Average time (s)
JPEG	1.25
JPEG 2000	26
CAE	0.56
MS-CAE	0.67

resource. Although our proposed MS-CAE algorithm consumes slightly more than CAE in the time of single image compression, the accuracy of the reconstructed image is better than that of JPEG and JPEG 2000 at low bpp. This consequence results from many operations of the JPEG and JPEG 2000 image compression algorithms, such as brightness matrix quantization, Huffman coding, DCT, or discrete wavelet transform (DWT). The computation complexity of these operations is high.

6. Conclusions

In this paper, a model segmentation-based compressive autoencoder (MS-CAE) image compression algorithm for image monitoring of WSN in remote areas is proposed. We first present the MS-CAE network architecture, which considers the limited computing resources and the practical deployment of WSN. Then, we provide the implementation method for the MS-CAE network. The decoder with a residual block network optimizes the problem of vanishing gradient and gradient exploration. Finally, we split the trained network model and deploy the weight parameters of the encoder and decoder into the edge device and cloud-computing device, respectively. Moreover, for the purpose of obtaining a high-resolution reconstructed compressed image, we appropriately increase the complexity of the decoding network. In addition, we also compare the performance with JPEG, JPEG 2000, FRIC-RNN, and CAE algorithms between 0 and ~1bpp. The experimental results show that the MS-CAE improves image resolution, compression performance, and transmission efficiency. Based on model segmentation, the designed model MS-CAE has achieved excellent performance in resource savings for edge hardware devices. It also has the ability to completely express the image content. Therefore, it also indicates that the proposed approach effectively improves the monitoring efficiency of long-term environmental image monitoring for WSN.

Data Availability

The image dataset (yt_small_720p) used to support the findings of this study are available from: <https://drive.google.com/file/d/1wbwkpz38stSFMwgEKhoDCQCMiLLFVC4T/view>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant Nos. 61961026 and 61962036); Natural Science Foundation of Jiangxi Province, China (Grant No. 20202BABL202003); China Postdoctoral Science Foundation (Grant No. 2020M671556), and Major Science and Technology Projects in Jiangxi Province (Grant No. 20213AAG01012).

References

- [1] X. Niu, X. Huang, Z. Zhao, Y. Zhang, C. Huang, and L. Cui, "The design and evaluation of a wireless sensor network for mine safety monitoring," in *IEEE GLOBECOM 2007-2007 IEEE Global Telecommunications Conference*, vol. 3, pp. 1291–1295, Washington, DC, USA, 2007.
- [2] M. F. Othman and K. Shazali, "Wireless sensor network applications: a study in environment monitoring system," *Procedia Engineering*, vol. 41, no. 1, pp. 1204–1210, 2012.
- [3] A. T. Demetillo, M. V. Japitana, and E. B. Taboada, "A system for monitoring water quality in a large aquatic area using wireless sensor network technology," *Sustainable Environment Research*, vol. 29, no. 1, pp. 1–9, 2019.
- [4] X. Bao, L. Han, X. He, W. Tan, and T. Fan, "Optimizing maximum monitoring frequency and guaranteeing target coverage and connectivity in energy harvesting wireless sensor networks," *Mobile Information Systems*, vol. 2019, Article ID 6312589, 14 pages, 2019.
- [5] Y. Cao, R. Ji, L. Ji, G. Lei, H. Wang, and X. Shao, " I^2 -MPTCP: a learning-driven latency-aware multipath transport scheme for industrial internet applications," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 8456–8466, 2022.
- [6] Y. Cao, R. Ji, X. Huang, G. Lei, X. Shao, and I. You, "Empirical mode decomposition-empowered network traffic anomaly detection for secure multipath TCP communications," *Mobile Networks and Applications*, vol. 27, no. 6, pp. 2254–2263, 2022.
- [7] J. Ma, M. Li, and H.-J. Li, "Traffic dynamics on multilayer networks with different speeds," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 1697–1701, 2022.
- [8] J. Peak, J. Hicks, S. Coe, and R. Govindan, "Image-based environmental monitoring sensor application using an embedded wireless sensor network," *Sensors*, vol. 14, no. 9, pp. 15981–16002, 2014.
- [9] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. 43–59, 1991.
- [10] M. Rabbani and R. Joshi, "An overview of the JPEG2000 still image compression standard," *Signal Processing: Image Communication*, vol. 17, no. 1, pp. 3–48, 2002.
- [11] J. Balle, V. Laparra, and E. P. Simoncelli, "End-to-end optimization of nonlinear transform codes for perceptual quality," in *2016 Picture Coding Symposium (PCS)*, pp. 1–5, Nuremberg, Germany, 2016.
- [12] F. Jiang, W. Tao, S. Liu, J. Ren, X. Guo, and D. Zhao, "An end-to-end compression framework based on convolutional neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 3007–3018, 2017.
- [13] P. Kuang, T. Ma, Z. Chen, and F. Li, "Image super-resolution with densely connected convolutional networks," *Applied Intelligence*, vol. 49, no. 1, pp. 125–136, 2019.
- [14] C. Huang, H. Liu, T. Chen, Q. Shen, and Z. Ma, "Extreme image coding via multiscale autoencoders with generative adversarial optimization," in *2019 IEEE Visual Communications and Image Processing (VCIP)*, pp. 1–4, Sydney, NSW, Australia, 2019.
- [15] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Deep convolutional autoencoder-based lossy image compression," in *2018 Picture Coding Symposium (PCS)*, pp. 253–257, San Francisco, CA, USA, 2018.
- [16] L. Theis, W. Shi, A. Cunningham, and F. Huszar, "Lossy image compression with compressive autoencoders," in *International Conference on Learning Representations (ICLR)*, pp. 1–19, Palais des Congrès Neptune, Toulon, France, 2017.
- [17] J. Balle, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *International Conference on Learning Representations (ICLR)*, pp. 1–27, Palais des Congrès Neptune, Toulon, France, 2017.
- [18] Y. Chong, L. Chen, and S. Pan, "End-to-end joint spectral-spatial compression and reconstruction of hyperspectral images

- using a 3D convolutional autoencoder,” *Journal of Electronic Imaging*, vol. 30, no. 4, 2021.
- [19] G. Toderici, D. Vincent, N. Johnston et al., “Full resolution image compression with recurrent neural networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5435–5443, Honolulu, HI, USA, 2017.
- [20] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, “Learning convolutional networks for content-weighted image compression,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3214–3223, Salt Lake City, UT, USA, 2018.
- [21] M. Akbari, J. Liang, and J. Han, “DSSLIC: deep semantic segmentation-based layered image compression,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2042–2046, Brighton, UK, 2019.
- [22] B. Sushma and B. Fatimah, “Wyner-Ziv coding of chroma in wireless capsule endoscopy image compression using deep side information generation,” in *2020 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, pp. 58–62, Chennai, India, 2020.
- [23] W. Shi, J. Caballero, F. Huszar et al., “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1874–1883, Las Vegas, NV, USA, 2016.
- [24] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, “Energy compaction-based image compression using convolutional autoencoder,” *IEEE Transactions on Multimedia*, vol. 22, no. 4, pp. 860–873, 2019.
- [25] U. Drolia, K. Guo, J. Tan, R. Gandhi, and P. Narasimhan, “Cachier: edge-caching for recognition applications,” in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 276–286, Atlanta, GA, USA, 2017.
- [26] Y. Kang, J. Hauswald, C. Gao et al., “Neurosurgeon,” *ACM SIGARCH Computer Architecture News*, vol. 45, no. 1, pp. 615–629, 2017.
- [27] C. Ding, A. Zhou, Y. Liu, R. N. Chang, C.-H. Hsu, and S. Wang, “A cloud-edge collaboration framework for cognitive service,” *IEEE Transactions on Cloud Computing*, vol. 1, no. 1, p. 99, 2020.
- [28] G. Li, J. Wang, H.-W. Shen, K. Chen, G. Shan, and Z. Lu, “CNNPruner: pruning convolutional neural networks with visual analytics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1364–1373, 2020.
- [29] S. Roy, P. Panda, G. Srinivasan, and A. Raghunathan, “Pruning filters while training for efficiently optimizing deep learning networks,” in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, Glasgow, UK, 2020.
- [30] M. Tian and M. Tong, “Self-adaptive subgradient extragradient method with inertial modification for solving monotone variational inequality problems and quasi-nonexpansive fixed point problems,” *Journal of Inequalities and Applications*, vol. 2019, no. 1, p. 19, 2019.
- [31] Z. Izzo, L. Ying, and J. Zou, “How to learn when data reacts to your model: performative gradient descent,” *International Conference on Machine Learning*, vol. 139, pp. 4641–4650, 2021.
- [32] A. Jafar and L. Myungho, “Hyperparameter optimization for deep residual learning in image classification,” in *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*, pp. 24–29, Washington, DC, USA, 2020.
- [33] E. Agustsson, F. Mentzer, M. Tschannen et al., “Soft-to-hard vector quantization for end-to-end learning compressible representations,” *Advances in Neural Information Processing Systems*, vol. 2017, p. 30, 2017.