IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 July 2022, accepted 25 July 2022, date of publication 1 August 2022, date of current version 23 February 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3195295

RESEARCH ARTICLE

Improving Image Compression With Adjacent Attention and Refinement Block

AFSANA AHSAN JENY[1,2], (Member, IEEE),
MD BAHARUL ISLAM[1,2,3], (Senior Member, IEEE),
MASUM SHAH JUNAYED[2], (Member, IEEE),
AND DEBASHISH DAS[4]

[1]Department of Computer Science and Engineering, Daffodil International University, Dhaka 1207, Bangladesh
[2]Department of Computer Engineering, Bahçeşehir University, 34349 İstanbul, Turkey
[3]College of Data Science and Engineering, American University of Malta, 1013 Bormla, Malta
[4]Faculty of Computing, Engineering and the Built Environment, Birmingham City University, Birmingham B4 7XG, U.K.

Corresponding author: Afsana Ahsan Jeny (afsanaahsan1996@gmail.com)

This work was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) under the 2232 Outstanding Researchers Program under Project 118C301.

ABSTRACT Recently, learned image compression algorithms have shown incredible performance compared to classic hand-crafted image codecs. Despite its considerable achievements, the fundamental disadvantage is not optimized for retaining local redundancies, particularly non-repetitive patterns, which have a detrimental influence on the reconstruction quality. This paper introduces the autoencoder-style network-based efficient image compression method, which contains three novel blocks, i.e., adjacent attention block, Gaussian merge block, and decoded image refinement block, to improve the overall image compression performance. The adjacent attention block allocates the additional bits required to capture spatial correlations (both vertical and horizontal) and effectively remove worthless information. The Gaussian merge block assists the rate-distortion optimization performance, while the decoded image refinement block improves the defects in low-resolution reconstructed images. A comprehensive ablation study analyzes and evaluates the qualitative and quantitative capabilities of the proposed model. Experimental results on two publicly available datasets reveal that our method outperforms the state-of-the-art methods on the KODAK dataset (by around 4dB and 5dB) and CLIC dataset (by about 4dB and 3dB) in terms of PSNR and MS-SSIM.

INDEX TERMS Image compression, attention mechanisms, Gaussian merge block, refinement block, autoencoder.

I. INTRODUCTION

(**Revision**) Image compression reduces spatial redundancy in images and optimizes bandwidth and storage space in various applications, including video compression, online advertising, professional photographic exchange, etc. Traditional image compression algorithms [1]–[4] depend on hand-crafted processes with intricate dependencies to increase compression efficiency. For example, JPEG [1] employs the discrete cosine transform (DCT). On the other hand, JPEG2000 [2] uses discrete wavelet transforms (DWT) to transfer an image pixel to the frequency domain and decompose multi-scale decomposition into spectral bands, respectively. However, they cause artifacts along the image borders, invisible at high bit rates. Recent video codecs, such as VVC [3] incorporate intra prediction and an in-loop filter for intra-frame coding. It is also utilized in BPG [4], an image codec, to minimize redundant and irrelevant features to improve the quality of the reconstruction frame. However, traditional compression techniques cannot be optimized end-to-end, limiting their overall rate-distortion (RD) optimization performance (particularly in similarity index) and learning ability.

Nowadays, deep learning-based image compression methods [5]–[10] outperform traditional algorithms in terms of rate-distortion (RD) performance. For example,

The associate editor coordinating the review of this manuscript and approving it for publication was Joewono Widjaja.

Ballé *et al.* [5] provide an end-to-end image compression using a convolutional neural network (CNN) based autoencoder. In particular, context-adaptive entropy models for learned image compression are renowned for achieving higher performance across traditional codecs. The study [6] introduced a hyperprior to add more bits to the entropy model to describe it more accurately. Minnen *et al.* [8] used the auto-regressive previous information to build an accurate entropy model and achieve equivalent or even higher compression efficiency performance than the conventional codec [4]. The work in [10] introduced a very similar notion by taking into account two sorts of contexts, bit consuming contexts (that is, hyperprior) and bit-free contexts (that is, auto-regressive model), achieving a context-adaptive entropy model. Although these methods enhance the compression performance, they also greatly raise the compression artifacts [11] due to the quantization process during the entropy coding and have stacked by limited respective fields in latent space.

To boost the overall image compression performance, the attention mechanism is being utilized to gather more details from the latent space while suppressing irrelevant information to allocate more bits [12]–[14]. The non-local attention mechanism [15] is effective in many visual tasks (i.e., semantic segmentation). Liu *et al.* [12] use the non-local attention to build implicit significance masks for leading the adaptive processing of latent features. On the other hand, Cheng *et al.* [13] remove the non-local block to make it easier to learn image compression. The most recent research in [14] also employed the non-local attention processes to enhance the adaptive processing of latent features. It helps the compression algorithm allocate additional bits to complicated areas (e.g., edges and textures). However, this work suffers some drawbacks. Firstly, their non-local attention (working in a single direction) has no impact on the vertical and horizontal weights to produce the respective wide field and acquire valuable features to improve RD performance. Secondly, a single mask in entropy coding will not be able to eliminate latent feature data redundancy. Thirdly, the compression artifacts are dramatically increased due to assigning bits to non-essential areas, resulting in poor reconstructed images. Motivated by it, we propose an efficient end-to-end image compression method that significantly improves the overall RD performance. Our contributions to this paper are summarized as follows:

- We present an end-to-end autoencoder-based image compression model to improve the overall image compression performance. Three new blocks, i.e., an adjacent attention block (AAB), a Gaussian merge block (GMB), and a decoded image refinement block (DIRB), are included in this model.
- A plug-and-play AAB is applied to capture spatial correlations (both vertically and horizontally), suppress unnecessary information, and boost entropycoding efficiency with more crucial features by allocating additional bits.
- The GMB simulates the distribution of the latent representation in a precise manner to boost the rate-distortion optimization performance.
- Compression artifacts are inevitable on the final reconstructed images since our approach is lossy image compression. A DIRB is used to leverage global information with rich texture information and vibrant features to improve the reconstructed image quality.
- An extensive experiment is conducted on two publicly available datasets. Our method shows state-of-the-art performance in both datasets and reduces the computational complexity simultaneously.

The remainder of the paper is arranged in the following manner. In Section II, traditional and existing deep learning based works are reviewed. The proposed architecture for image compression with three new blocks, e.g., AAB, GMB, and DIRB are described in detail in Section III. Section IV represents the dataset, training details, and the evaluation metrics. The qualitative and quantitative results with some ablation studies are presented in section V. Finally, section VI concludes the paper with our future research works.

II. RELATED WORKS

In this section, we briefly discuss the classical and deep learning-based image compression methods.

A. CLASSICAL METHODS

Image compression techniques are primarily concerned with reducing the levels of spatial redundancies present in images. For example, converting photos from the pixel domain to the frequency domain is simpler to compress. For instance, JPEG [1] applies the discrete cosine transform. In contrast, JPEG2000 [2] applies the discrete wavelet transform, which is handcrafted. To reduce data redundancy, high-frequency information is separated from low-frequency information, and bits are allocated according to the signal significance. Entropy coding such as Huffman [16], [17], hashing [18], and arithmetic coding [19], [20] is also utilized to increase the lossless compression performance of the image.

Currently, the intra-prediction approach [3], [4], which is often used in video compression, has been employed for image compression as well. The BPG [4] standard, for example, is based on the HEVC/H.265 [21] image compression standard, which delivers the highest possible image compression results in comparison to prior methods, such as JPEG and JPEG2000. The prediction-transform approach is used in the BPG standard [4], and 35 encoding options are utilized to create the reconstructed image, which also decreases redundant data. Then, bigger computing units, more forecast methods, more transform varieties, and more coding facilities are all supported by VVC [3]. Furthermore, the hybrid techniques employ both conventional compression techniques and the most current learning super-resolution strategies, such as [22], to achieve higher compression ratios. However, traditional algorithms are created by hand-crafted components (such as entropy coding).

B. DEEP LEARNING-BASED METHODS

Deep neural networks (DNNs) have shown to be useful for various computer vision applications in recent years, namely super-resolution, denoising, and object recognition. Some recent studies have attempted to conduct neural networks' excellent representation capabilities to improve the performance of image compression [5], [6], [8], [10], [13], [23]–[32]. Toderici *et al.* [23] developed the first learning-based image compression framework, which was based on a recurrent neural network (RNN). Various bitrates may be generated using a single model in their method. When compared to BPG, [28] introduces more complex RNN components and efficient reconstruction approaches to obtain equivalent or even superior MS-SSIM [33] results. Although some of these approaches [23], [25], [28] are aimed to reduce the bitrates, the rate-distortion (RD) trade-off is not considered.

By improving the RD performance, Ballé *et al.* [24] introduced a CNN-based framework with the generalized divisive normalization (GDN) layer, which is effective for simulating nonlinear transformations that have been frequently employed in subsequent approaches [5], [6], [8], [10], [13], [14], [34]. However, to improve the RD performance, these methods conduct the Gaussian Model (GM) distribution that is still short of encoding latent features by effectively estimating the conditional statistics. According to Rippel and Bourdev [35], a feature pyramid network (FPN) was introduced to obtain more valuable features. However, this would also lead to redundant information since convolutional methods exchange features. Li *et al.* [29] suggested the use of a significance map to alter the bit allocation of images, which they found to be effective. To create the significance map, a branch of a three-layer convolutional neural network was trained. However, the explicit learning material requires weight, which raises the computing cost. It is also tricky to adaptively assign bits for in-depth features, as described in [29].

In the training process, some methods [27], [32] employed an adversarial network (GAN) as a distortion assessment to lead the decoder to create more feasible pattern structures, which tends to result in reconstructed images of decent visual quality. But the pattern structures obtained in this way are not actual textures and lack fidelity. Recent studies on adaptive learning of feature significance have shown that attention strategies are quite effective. Considerable progress has been achieved in areas like as natural language processing [36] and semantic segmentation [15]. Moreover, the efficiency of noise removal and super-resolution can be dramatically improved by incorporating non-local block (NLB) into neural networks [37], [38]. In image compression, some methods [12]–[14], [39] employ attention mechanisms that allow spatially adaptive feature response for more difficult locations (i.e., patterns, saliency) in order to allocate more bits. For example, [39] introduced an improvement unit that functions on full-resolution photos to eliminate compression artifacts by filtering the reconstructed images using a simple neural network. [12]–[14] employed residual non-local attention mechanisms to improve the RD performance and compression artifacts due to the quantization procedure. However, these proposed attention mechanisms can't be exploited features in both directions (vertical and horizontal) because of their one-way weight allocation. Therefore, allocating more bits in complex regions (i.e., patterns, edges) is not fully explored to improve the final reconstructed image.

In contrast, we propose an adjacent attention block that uses distinct weights in the horizontal and vertical directions for feature maps to maintain only the most relevant information while eliminating unnecessary information, such as a complicated natural background, which has a significant impact on the performance of RD. Furthermore, in order to decrease compression artifacts, we have included a refinement block, which is capable of smoothing out and improving the visualization of the reconstructed image.

III. METHODOLOGY

This section presents the proposed deep image compression framework in detail. In Figure 1, the architecture is shown. Typically, well-known autoencoders are used in CNN-based compression techniques [5], [6], [8], [12], [29], [30], [32], [35]. Among them, variational autoencoder (VAE) has been shown to be a successful architecture for compression as first described in [6]. In this network [6], to successfully capture spatial relationships while boosting the compression performance by the entropy model efficiently, the hyper-encoder and the hyper-decoder network are employed with two times quantization. Therefore, motivated by [6], we adopt the network of the autoencoder type for learning-based image compression with three new blocks to improve the overall performance. In particular, four modules are employed in the proposed system, which are the main encoder and decoder, as well as the hyper-encoder and the hyper-decoder network, respectively. The proposed attention mechanism, referred to as the adjacent attention block (AAB), is included in each architecture module. Two additional blocks, the Gaussian merge block (GMB) and decoded image refinement block (DIRB) are introduced to increase the overall performance of the RD and improve the reconstructed image, respectively.

At first, the original image I is taken through the main encoder network and creates the corresponding latent representations l_a by employing four convolutional layers with non-linear functions (e.g., GDN). After that l_a is quantized to $\hat{l}_a$. The quantized latent forms $\hat{l}_a$ are delivered to the decoder network to generate the final reconstructed image $\hat{I}$ after arithmetic encoding (AE) and decoding (AD) [19]. Similarly, we utilize the same quantization method as [6], [8] with some modifications in the latent state (i.e., added the GMB block) in a precious way. When it comes to image compression, the goal is to obtain high-quality reconstructed images at a certain bitrate, and the entropy model is utilized to predict

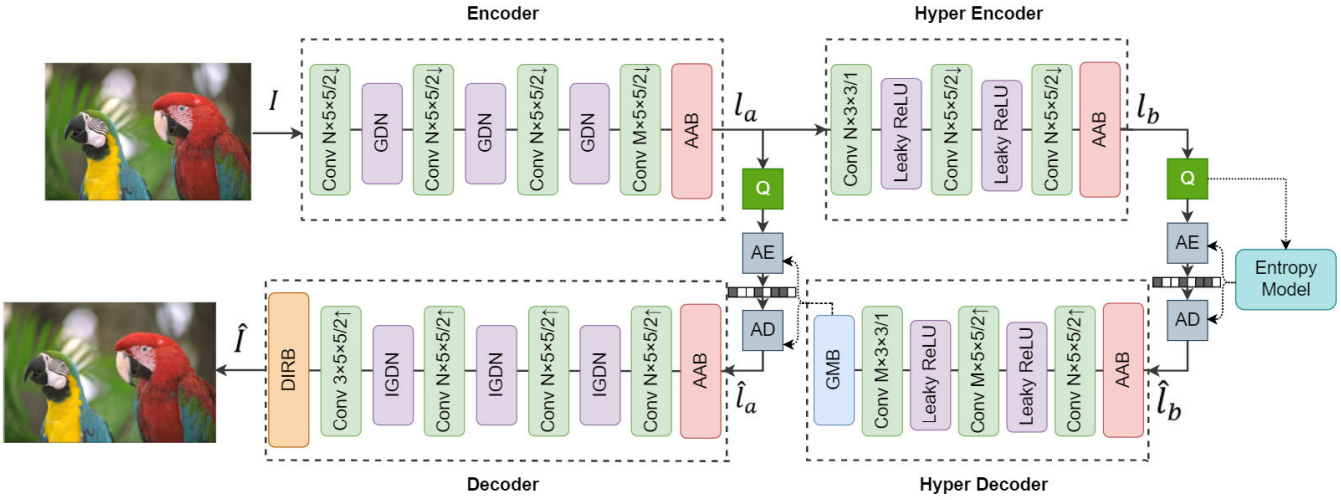

FIGURE 1. Proposed architecture. The AAB, GMB, and Dt attention block, Gaussian merge block, and decoded image refinement block. Q, AE, and AD correspondingly indicate the quantization, arithmetic encoder, and arithmetic decoder. The parameters of Conv (convolution) layers indicate the number of filters × (kernel height) × (kernel width) × stride (up or downsampling). Here, upsampling and downsampling are represented by ↑ and ↓, respectively. For the feature map values of N and M, we employ 192 and 320, respectively.

the bitrate target. The entropy model uses the hyperprior module in conjunction with the factorized module. This method of entropy coding uses a hyperprior network to produce an estimate of latent forms before quantizing and encoding the output of the hyperprior encoder into the bitstream. It will be encoded into the bitstream since this information is necessary for decoding, and the proper entropy model will increase compression effectiveness. In this work, the hyper-encoder module received the hyper-prior information from the latent forms $\hat{l}_a$ and encoded them into latent representations l_b. After that, it is quantized to $\hat{l}_b$ and passed to the hyper-decoder after AE and AD process. The hyper-decoder module again retrieves the hyper-prior information from $\hat{l}_b$ and estimates the relevant entropy model parameters (φ, ϑ) accordingly. In the following three subsections, we will go through our proposed three blocks, i.e., AAB, GMB, and DIRB of the framework.

The below loss function (Υ) is employed to optimize the whole training process of the compression technique:

$$\Upsilon = \lambda \mathrm{D} + \mathrm{R} = \lambda \mathrm{d}(\mathrm{I}, \hat{I}) + \mathrm{H}_{\left(\hat{l}_a\right)} + \mathrm{H}_{\left(\hat{l}_b\right)} \tag{1}$$

The D and R are the distortion and bitrate, respectively, in this equation. The amount of distortion and the bit rate are both taken into consideration by λ. The distortion measure (MS-SSIM [33]) is denoted by d(.). H is the bitrate utilizing for encoding the latent visualization $\hat{l}_a$ and $\hat{l}_b$, respectively.

During the training phase, we use an entropy estimation method that is presented in [8], and we represent the latent features in the following way:

$$P_{\hat{l}_a|\hat{l}_b}\left(\hat{l}_a \mid \hat{l}_b\right) = \prod_i \mathcal{N}\left(\boldsymbol{\varphi}^i, \boldsymbol{\vartheta}^{2(i)}\right) * \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right)\left(\hat{l}_{a_\mathrm{i}}\right). \tag{2}$$

Every latent portrayal $\hat{l}_{a_\mathrm{i}}$ is represented as a Gaussian distribution with its parameters φ^i and $\boldsymbol{\vartheta}^i$ which are predicted by the probability of the hidden element $\hat{l}_b$. $\hat{l}_b$ is referred to as the hyperprior, $\mathcal{U}$ stands for a uniform distribution, and $*$ is the convolution process. The hyperprior $\hat{l}_b$ is represented as below:

$$p_{\hat{l}_b|\psi}(\hat{l}_b|\psi) = \prod_i \left(p^{(i)}_{\hat{l}_b}|\psi^{(i)}(\psi^{(i)}) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2})\right)(\hat{l}_{b_i}) \tag{3}$$

where every univariate's distribution is represented by $p_{\hat{l}_b^{(i)}|\psi^{(i)}}$ and its parameters are represented by $\psi^{(i)}$. The bit rate in our technique is made up of the bit rates for the hidden variable $\hat{l}_b$ and the latent representations $\hat{l}_a$. However, the bit rates of Equation (1) are indicated as:

$$H_{\left(\hat{l}_a\right)} = \sum_i -\log_2\left(P_{\hat{l}_{a_\mathrm{i}}|\hat{l}_{b_\mathrm{i}}}\left(\hat{l}_{a_\mathrm{i}} \mid \hat{l}_{b_\mathrm{i}}\right)\right) \tag{4}$$

$$H_{\left(\hat{l}_b\right)} = \sum_i -\log_2\left(P_{\hat{l}_{b_\mathrm{i}}|\psi^{(i)}}\left(\hat{l}_{b_\mathrm{i}} \mid \psi^{(i)}\right)\right). \tag{5}$$

A. ADJACENT ATTENTION BLOCK

In deep neural networks, the attention mechanism is an effort to emulate the similar behavior of deliberately focusing on a few significant elements while disregarding the rest. Nowadays, there are now three primary techniques to include attention mechanisms: spatial [40], channel [41], and Convolution Block Attention Module (CBAM) [42]. In the meanwhile, several researchers have adapted spatial attention processes by non-local blocks [43] to image compression [14] and [12], intending to reduce spatial redundancy. Furthermore, to construct an image generation model, [44] employed a transformer-based self-attention block which increased the size of the images. However, these methods concentrate only

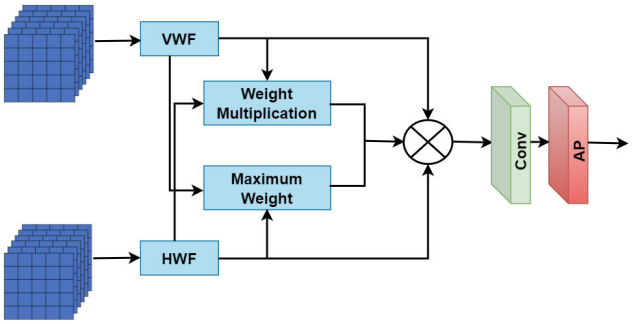

FIGURE 2. Proposed architecture of AAB. VWF and HWF represent the coefficients of the feature maps (size 320 for encoder and 192 for hyper encoder-decoder and decoder): vertical weight features (VWF) coefficients and horizontal weight features (HWF) coefficients, respectively. The softmax function recognizes the weight coefficients, then extended by the weight multiplication block, and the maximum weight block selects the most significant weight coefficients. After that, the weight coefficients are passed to the convolution layer (Conv) and average pooling layer (AP) to produce deep features.

on building deep networks to increase the models' representation capability, which results in high computation and memory demands. Besides, in most cases, the conventional spatial attention mechanism [45] only provides one-direction weight allocation [12]–[14], which results in the loss of vital information up to a specific level.

We propose a spatial adjacent attention mechanism, namely, AAB, which allocates weights coefficients based on distinct methods from both the vertical and horizontal directions. In addition to successfully suppressing irrelevant information, it may also ensure that the loss of critical information is kept to an absolute minimum. Besides, it concentrates the texture on the edges of the image with much contrast and allocates additional bits to them. Figure 2 depicts the proposed structure of AAB. Three parts are included in the structure.

- First, the coefficients of weight features are selected by the vertical weight features (VWF) and horizontal weight features (HWF) blocks. It works crosswise to obtain more stable features for allocating more bits in edge areas.
- Second, the two types of weight features are multiplied through the structure's weight multiplication (WM) module to increase the weight coefficients (for example, a tiny weight could be 0.1×0.2, while the highest weight could be 0.9×0.7).
- Third, the softmax function recognizes the weight coefficients, then extended by the weight multiplication block, and the maximum weight (MW) block selects the most significant weight coefficients [for instance, max (0.1, 0.9)].

To connect and concatenate the weights coefficients of the three parts of the model are arranged as follows:

$$w_i = \sum_{l=1}^{n} \frac{a^{a_{i,l}}}{\sum_{q=1}^{n} a^{a_{i,q}}} d_l \tag{6}$$

$$wm = concat\left([w_s, w_r, (w_s * w_r), (w_s, w_r)]\right) \tag{7}$$

The weights (w_i) of VWF and HWF allocated by the attention process are denoted by $a^{a_{i,l}}$, pixel i and I denote the feature at a specific instant and the sequential feature, and the hidden layer characteristics of the feature sequence I are indicated by d_l. In equation 2, m indicates the weight multiplication, and w_s represents the weight coefficient of VWF in the feature space ($w_s = [w_1, w_2, \ldots . \ w_{i-1}, w_i]$). Then the weight operation of WM and MW is denoted by $(w_s * w_r)$, and (w_s, w_r), respectively. After completing all the weight operations of VWF and HWF, one convolutional layer (Conv) and average pooling (AP) layer produce the deep feature.

According to Figure 1, for high-quality compression, the suggested AAB is incorporated into the encoding, decoding, hyper-encoding, and hyper encoding networks for leveraging the channel relationship. The re-weighted feature map from AAB is included in the subsequent quantization and entropy coding components.

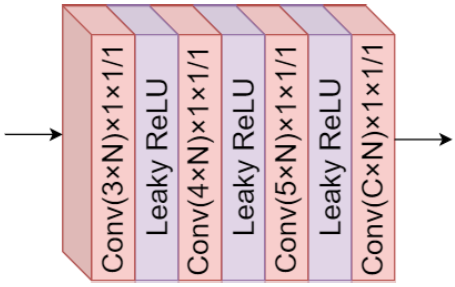

FIGURE 3. Architecture of GMB. For each layer, N specifies the hyper-parameter that determines how many channels will be available, and C indicates how many different Gaussian models will be available.

B. GAUSSIAN MERGE BLOCK

Estimating bit rates is critical in learning-based image compression techniques. Minnen *et al.* [8] and Lee *et al.* [10] demonstrate learning-based systems in which the hyper-prior compression technique is employed and a Gaussian Model (GM) distribution is used to represent the latent representations $\left(\hat{l}_a\right)$ in the model.

$$E_{\hat{l}_a|\hat{l}_b}\left(\hat{l}_a \mid \hat{l}_b\right) \sim \partial(\varphi, \vartheta) \tag{8}$$

where $E_{\hat{l}_b}(\hat{l}_b)$ denotes the quantized entropy model [5]. The purpose of the hyper-encoder and hyper-decoder is to predict the parameters (φ, ϑ) of the GM. Though the single GM-based entropy model has significantly improved over prior work [5], the representation capabilities of single GM are still inadequate, particularly for complicated components. As a result, we conduct the Gaussian Merge Block (GMB) to boost the image compression performance. In our proposed GMB, the $\hat{l}_a$ is expressed as below:

$$E_{\hat{l}_a|\hat{l}_b}\left(\hat{l}_a \mid \hat{l}_b\right) \sim \sum_{i=1}^{G} W_i \partial\left(\varphi_i, \vartheta_i\right) \tag{9}$$

where W_i and G denote the weights assigned to various GMs and the number of GMs, respectively. To estimate the

parameters (ψ) of the GMB, we generate three convolutional layers with three LeakyReLU layers, as illustrated in Figure 3. In our proposed GMB, the value of G is set to three. A total of 6 × N output channels are employed, with the first 5 × N channels being used for predicting the mean and variance of three GMs. A sigmoid layer is included in the output of the final N channels in estimating the weights of every GM. For example, the weight of the first GM is denoted by W, and the next one will be (1-W), respectively. Furthermore, by creating $G(G \geq 4)$ GMs, we may increase the number of output channels on the GMB block to $4 \times G \times N$ (C = 4 × G). For G of GMs, the mean and variance parameters are estimated by the first 3 × G × N channels in the same manner. The softmax layer is utilized after the final G × N channels to figure out each GM weight.

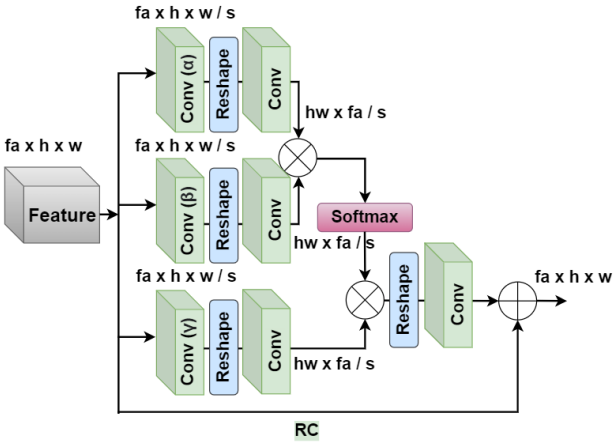

FIGURE 4. Architecture of DIRB. Conv indicates the convolution layer and RC means residual connection. See Section III (C) for more details.

C. DECODED IMAGE REFINEMENT BLOCK

The proposed compression approach for the entropy model employs a quantization procedure. As a result, compression artifacts may appear in the reconstructed image. Thus, a proposed DIRB, at the decoder side, is adjoined after image reconstruction, which significantly improves the performance of the decoded image. To improve the representations of feature maps, the proposed refinement block uses a self-similarity measures and inter-spatial relationship information. The following is a concept of a deep neural network process:

$$I_i = \frac{1}{f(O)} \sum_{\omega_j} F\left(O_i, O_j\right) \gamma\left(O_j\right) \tag{10}$$

$$f(O) = \sum_{\omega_j} F\left(O_i, O_j\right) \tag{11}$$

where i is the position reference of the feature reaction awaiting to be computed, and j is the counted position reference of input features. The input and output signals are represented by I and O, respectively, with the same area and channel number. At the input feature map, $F(.)$ calculates the similar reaction between i and all j. The response is multiplied by the matching features representation calculated by γ (.) after normalizing with a coefficient $f(O)$. Refinement block can extract the long-distance dependency between multiple places by calculating the reaction matrix, which may efficiently enlarge the receptive fields of deep convolution layers. It solves the shortcomings of traditional standard convolution operations, which can only gather minimal data from nearby regions. Figure 4 depicts our proposed DIRB for obtaining spatial relevant information in a feature space.

$$F\left(O_i, O_j\right) = \alpha\left(X_0\right)^t \beta\left(X_0\right) \tag{12}$$

where X_0 represents the input features and $F\left(O_i, O_j\right)$ represents the reaction weight vector for every position. Convolution operations ($\alpha(.)$ and $\beta(.)$) are used to produce the features descriptions, which are multiplied to create the matching matrix.

$$X_{IF} = \text{softmax}\left(\alpha\left(X_0\right)^t \beta\left(X_0\right)\right) \bigotimes \gamma\left(X_0\right) \tag{13}$$

where softmax (.) and X_{IF} denote the normalized operation and improved features, respectively. Improved features X_{IF} are calculated by multiplying the reaction weight vector for the feature representations, produced by the 1×1 convolution operation $\gamma(.)$.

$$X_{out} = X_0 \bigoplus X_{IF} \tag{14}$$

In the refinement block, we included a residual connection that constructs similar to a residual learning network by combining input feature X_0 and improved feature X_{IF}. It enables the component to concentrate on improving high-frequency information rather than low-frequency information.

Comparing the ways of gradually expanding the receptive fields in typical regular convolution procedures, our proposed refinement block can acquire the spatial dependency between any two locations for the purpose of further refining and improving the flow of gradients and information. Our DIRB can also add global information to the features that allow our network to utilize better the promising information contained within the low-resolution reconstructed images.

IV. EXPERIMENTS

A. DATASET

The experimental datasets are primarily separated into two types: training data and test data. We randomly select 300k images from the Open Images dataset [46] and crop them to a 256 × 256 pixel size for training. For testing, the KODAK image dataset [47] and CLIC professional validation dataset [48] are employed, including high-resolution natural images. The KODAK dataset comprises 24 photos with a resolution of 512 × 768 pixels and a broad range of contents and patterns, which are artifact-sensitive (restricted color gradients). As a result, it's frequently been employed to test image compression techniques. The CLIC dataset [48] includes 41 pictures acquired by mobile phones and professional cameras. The images have greater resolutions, with

an average size of 1913 × 1361 pixels for mobile shots and 1803 × 1175 pixels for professional photos.

B. TRAINING DETAILS

All experiments are carried out on a Windows 10 workstation with an Intel Core i7 processor, 32GB of RAM, and a single NVIDIA GeForce RTX 2070 GPU with 8GB of memory running under the CUDA 10.0. To finish the experiment's code, we used Python 3.7.0 with Conda environment. Pytorch 1.0.0 is used as the deep learning framework. For the model implementation process, the Adam optimizer [49] is conducted to train all models for 1.8M steps with a batch size of 8. For the first $110k$ iterations, the learning rate is determined to 0.0003, then reduces to 0.00003 for the other $35k$ iterations, and finally to 0.00001 for the final $35k$ iterations. The channel numbers of the latent and hyper latent variables are set in the proposed model at 320 and 192, respectively.

C. EVALUATION METRICS

This article evaluates the rate-distortion in bits per pixel (bpp) while the model is optimized by employing the PSNR [50] and MS-SSIM [33]. To show their coding efficiency, rate-distortion (RD) curves are generated. We followed the same setting of [51], and for MS-SSIM, the λ values are fixed to 2.41, 5.24, 8.31, 15.65, 30.43, and 60.56.

V. RESULTS AND DISCUSSIONS

A. QUALITATIVE RESULTS

We present some visualization outcomes to make the efficiency of our approach more apparent. Figure 5 and Figure 6 demonstrate the qualitative comparisons (final reconstructed images) of some images from KODAK dataset [47] dataset. In Figure 5, we compare our results with existing methods [2]–[5]. To illustrate the efficiency of our proposed technique, we highlight a few specific areas of the reconstructed images for a more in-depth examination. Our reconstructed images have a higher PSNR, i.e., 37.9dB (Kodim 23.png), 34.21dB (Kodim 24.png), and 30.01dB (Kodim 19.png), and maintain around the same bit rates as other methods. Besides, the texture of the images is more vibrant (especially the patterns around the eyes of the birds (row 1), the drawing (row 2), and the window (row 3), allowing us to preserve the finer feature pleasingly. In Figure 6, we show the comparison of the reconstructed images with original images in different zoom-in ways for better qualitative visualization.

Usually, textured areas (high contrast) are often allocated more bits than non-textured areas (low contrast), resulting in better visual quality at the same bit rate. To display the efficiency of the proposed AAB, the visualizations of kodim23.png, kodim19.png, and kodim24.png from the KODAK [47] dataset are depicted in Figure 7. From Figure 7 (b and c), it can be shown that AAB distributes weights vertically and horizontally to suppress irrelevant information effectively. As a result, in latent (Figure 7(b)), it assigns more bits to regions of high contrast (objects) while assigning fewer bits to regions of low contrast (background). However, Figures 7 (b and c) are expressed for 1.19 Bpp and 0.17 Bpp, respectively. It can be clearly said that by AAB, we can allocate more bits not only at higher bit rates Figure 7 (b) but also at lower bit rates Figure 7 (c). Even we keep both kernels and feature maps maintain a virtually identical pattern, although each element's intensity is adjusted differently. In summary, our proposed AAB is very effective in the latent representations because it can provide the almost same pattern at lower bit rates.

In order to demonstrate the efficiency of our proposed DIRB, the visualizations are portrayed in Figure 8 of kodim23.png, kodim19.png, and kodim24.png from KODAK [47] dataset. The result after the last convolution layer is shown in Figure8 (b), and after applying the DIRB, the final reconstructed images are shown in Figure 8 (c). We can see that the learned residual images (Figure 8 (b)) include a disproportionate amount of high-frequency information. On the other hand, the final reconstructed images (Figure 8 (c)) also aid in perfectly predicting the spectral analysis of the images with a better display.

B. QUANTITATIVE RESULTS

To evaluate our proposed model, the RD performance is computed. We employ the PSNR as the quality measure, as illustrated in Figure 9 (a). Our technique is evaluated against a variety of well-known image compression algorithms (both classical and deep learning-based), including [3]–[5], [8], [10], [13], [14]. When compared to [4], [5], [8], [10], [13], , our method outperforms them by a large margin, specially from the most popular methods Chen *et al.* [14] (around 41.12 vs. 37.4), and Cheng *et al.* [13] (around 41.12 vs. 37.1). However, the bit rates are slightly lower (about 0.7%) when comparing our approach to the traditional method [3] (around 41.4 vs. 41.12).

The experiments are also carried out using the MS-SSIM quality measure, as seen in Figure 9 (b). We provide MS-SSIM values in decibels (i.e., $-10log_{10}(1 - MS - SSIM)$) to better illustrate the progress. It is clearly said that our method shows state-of-the-art performance against both the traditional methods, including [4] and [3], and deep learning-based methods, including [8], [10], [13], [14], and [5]. Therefore, we can say that the AAB, GMB, and DIRB we have presented have a significant influence on showing higher RD performance and improving the reconstructed image's similarity. Please refer the ablation study (in next sub-section) to get a better idea of the modules' efficacy.

We employ another CLIC [48] professional validation dataset to confirm the robustness of our technique, and the results are shown in Table 1. It is noteworthy that our approach also yields state-of-the-art results in terms of MS-SSIM which we express in decibels (i.e., $-10log_{10}(1-MS-SSIM)$). However, regarding PSNR, our method achieves

FIGURE 5. Qualitative performance comparison of the our reconstructed images with existing methods, such as Ball é *et al.* [5], BPG444 [4], JPEG [2], and VTM 8.0 [3]. These images are taken from KODAK [47] dataset.

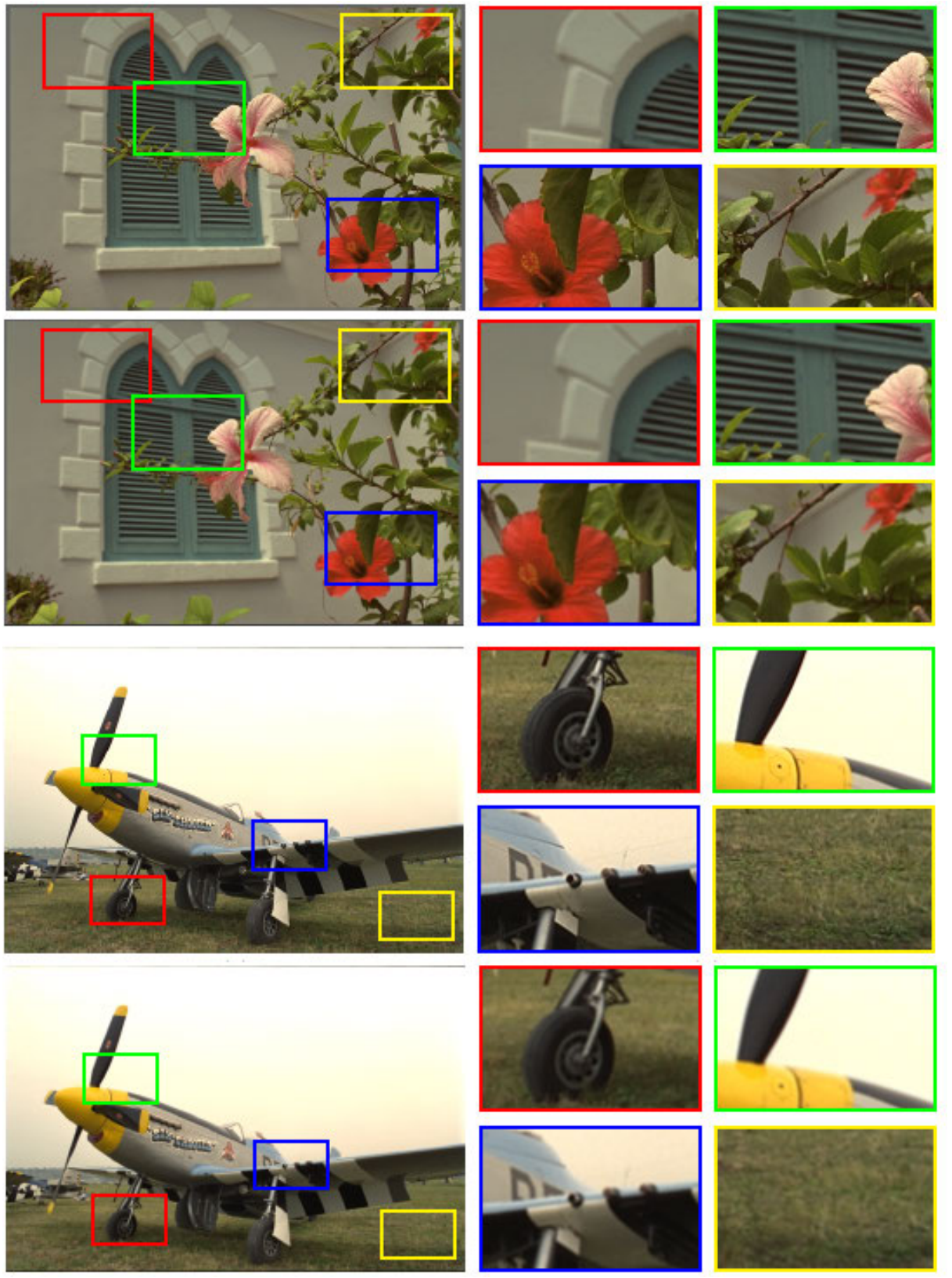

FIGURE 6. Qualitative comparison of the reconstructed images (row 2, 4) with the ground truth (row 1, 3) images from KODAK [47] dataset (Kodim 07.png and Kodim 20.png). The highlighted rectangular area zoom-in by $\times 2$ (in row 1, 2) and $\times 3$ (in row 3, 4) for better visualization.

TABLE 1. RD performance (MS-SSIM and PSNR) of the CLIC [48] Professional Validation dataset. Bold indicates the highest performance and underline indicates the second highest.

Approaches	MS-SSIM	PSNR	bpp
JPEG [2]	17.6	35.4	0.80
BPG44 [4]	22.3	41.2	0.75
VTM 8.0 [3]	22.6	**42.5**	0.70
Lee et al. [10]	26.6	40.4	0.74
Balle et al. [6]	22.5	40.1	0.89
Minnen et al. [8]	<u>27.1</u>	40.0	0.88
Cheng et al. [13]	24.1	36.4	0.63
Ours	**27.4**	<u>40.8</u>	0.58

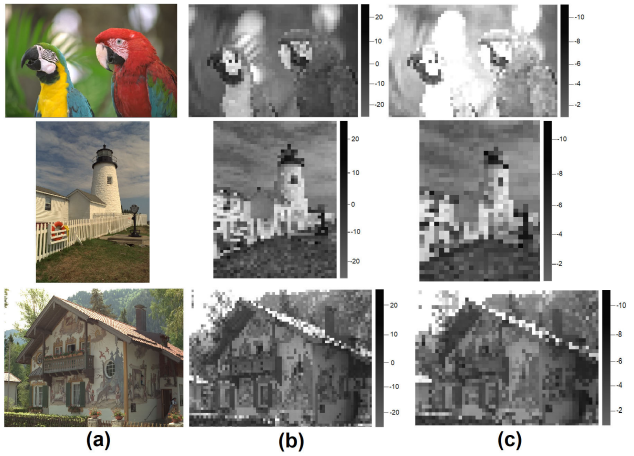

FIGURE 7. Visualization result of AAB regarding kodim23.png, kodim19.png, and kodim24.png from KODAK [47] dataset. (a) original image, (b) Latent, and (c) Allocated bits (see the edges of the objects).

TABLE 2. Performance comparison of different prior attention blocks with our AAB in terms of PSNR with bit rates (bpp) on KODAK dataset [47].

Approaches	λ	PSNR	bpp
Proposed method	2.41	27.23	0.201
Proposed method + attention module [13]	2.41	31.89	0.211
Proposed method + NLAM [14]	2.41	32.21	0.202
Proposed method + AAB	2.41	**32.98**	**0.208**
Proposed method	8.31	27.51	0.657
Proposed method + attention module [13]	8.31	33.87	0.678
Proposed method + NLAM [14]	8.31	34.01	0.651
Proposed method + AAB	8.31	**35.67**	**0.591**

the second-highest result (underline results in Table 1), which is approximately 4% less than the traditional

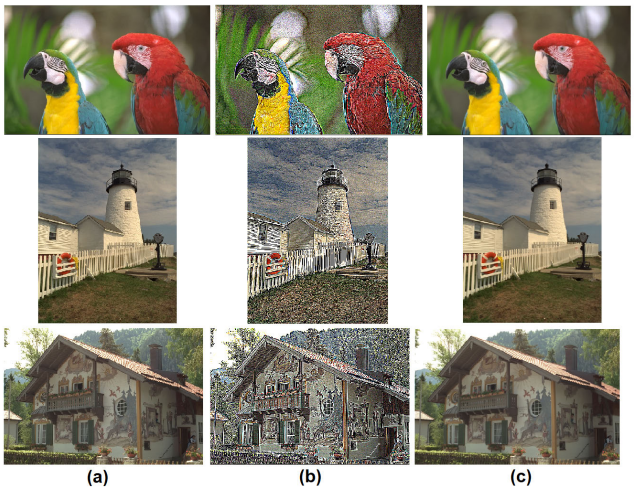

FIGURE 8. Visualization result of DIRB in terms of kodim23.png, kodim19.png, and kodim24.png from KODAK [47] dataset. (a) original image, (b) the reconstructed image without applying DIRB, and (c) the reconstructed image after applying DIRB.

method [3] (around 42.5 vs. 40.8) at lower bit rates (0.58). However, it outperforms all existing deep learning approaches.

C. ABLATION STUDY

We perform some ablation studies on the KODAK dataset [47] to further illustrate the robustness and effectiveness of our proposed approach.

In Table.2, we provide an investigation by adopting current attention modules replacing our suggested attention module in our proposed approach for two kinds of λ values. PSNR performance is relatively poor (27.23) for $\lambda = 2.41$ and 8.31 at low and high bit rates when the attention module is not included in the baseline model. When the attention modules of Cheng *et al.* [13] and Chen *et al.* [14], and ours are utilized, the PSNR values improve by around 15% (31.89 vs. 27.23 and 32.21 vs. 27.23) for [13] and [14], and by about 17% (32.98 vs. 27.23) for ours at low bit rates, respectively. The PSNR improves significantly when $\lambda = 8.31$, for example, for our suggested adjacent attention module, the PSNR is improved by roughly 23% (35.67 vs. 27.51) and even by around 5% (35.67 vs. 33.87 and 35.67 vs. 34.01) over the prior modules of [13] and [14].

TABLE 3. Performance measurement of the proposed modules in trems of PSNR, MS-SSIM, and Inference Time on KODAK dataset [47]. The AAB, GMB and DIRB indicate Adjacent Attention Block, Gaussian Merge Block, and Decoded Image Refinement Block, respectively.

S.N.	Baseline	AAM	GMB	DIRB	PSNR	MS-SSIM	Inference Time (ms)
1	✓	×	×	×	26.27	18.71	522
2	✓	✓	×	×	27.95	19.95	734
3	✓	×	✓	×	26.89	18.78	591
4	✓	×	×	✓	30.33	20.32	822
5	✓	✓	✓	×	32.51	23.41	883
6	✓	×	✓	✓	31.23	21.87	861
7	✓	✓	×	✓	33.96	24.45	901
8	✓	✓	✓	✓	**35.89**	**26.01**	**1067**

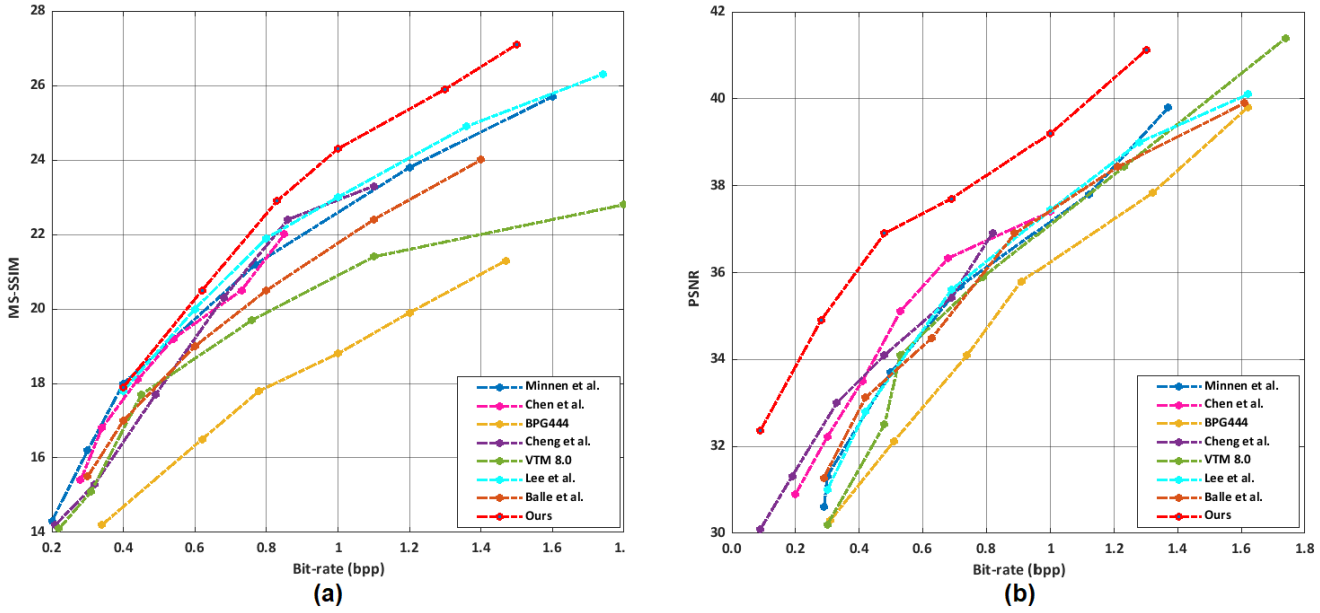

FIGURE 9. RD performance assessment on the KODAK dataset [47]. (a) the performance of MS-SSIM in decibels (i.e., $-10log_{10}(1 - MS - SSIM)$), and (b) the performance of PSNR.

To further verify the effectiveness of our proposed three modules in the main architecture, we have carried out another experiment in terms of PSNR, MS-SSIM, and Inference Time by replacing and adding the modules to bring the bpp close to 1 in Table 3. The PSNR and MS-SSIM performance of the baseline model are 26.27 and 18.71, respectively, when the three modules are not included as well as the inference time of 522ms. The value of PSNR and MS-SSIM dramatically increases with the inference time of 1067ms when all components are taken into account. For example, the improvement in PSNR and MS-SSIM is roughly 27% (35.89 vs. 26.27) and 28% (26.01 vs. 18.71). Among them, the proposed adjacent attention module and refinement block are able to boost the RD performance more, for instance, approximately 23% (33.96 vs. 26.27) in PSNR and 23% (24.45 vs. 18.71) in MS-SSIM. Eventually, it can be concluded that our proposed modules are very effective in boosting the state-of-the-art RD performance.

VI. CONCLUSION

This paper introduces a deep learning-based efficient image compression model that utilizes the autoencoder-style network. To increase the overall performance of image compression, three additional components, namely Adjacent Attention Block (AAB), Gaussian Merge Block (GMB), and Decoder Image Refinement Block (DIRB), are included in this model. The AAB is used to concentrate the texture on the edges of the image in order to allocate additional bits for capturing spatial correlations and concealing irrelevant features. The GMB and DIRB are applied to simulate the distribution of the latent representation and improve the defects in low-resolution decoded images, respectively. Two publicly available datasets (KODAK and CLIC) are employed in this experiment. Experimental findings reveal that the proposed model outperforms existing deep learning-based techniques in terms of MS-SSIM and PSNR. In the future, we will investigate additional components that influence reconstructed images, such as the entropy model.

REFERENCES

[1] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 34, no. 1, pp. 30–44, Apr. 1992.
[2] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 36–58, Sep. 2001.
[3] *VTM10.0. VTM Reference Software for VVC*. Accessed: Mar. 12, 2022. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-10.0
[4] F. Bellard. *BPG Image Format*. Accessed: Jan. 15, 2022. [Online]. Available: https://bellard.org/bpg/evaluation
[5] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," 2016, *arXiv:1611.01704*.
[6] J. Ballé, D. Minnen, S. Singh, S. Jin Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," 2018, *arXiv:1802.01436*.
[7] Y. Kim, J. W. Soh, and N. I. Cho, "AGARNet: Adaptively gated JPEG compression artifacts removal network for a wide range quality factor," *IEEE Access*, vol. 8, pp. 20160–20170, 2020.
[8] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–10.
[9] W. Li, W. Sun, Y. Zhao, Z. Yuan, and Y. Liu, "Deep image compression with residual learning," *Appl. Sci.*, vol. 10, no. 11, p. 4023, Jun. 2020.
[10] J. Lee, S. Cho, and S.-K. Beack, "Context-adaptive entropy model for end-to-end optimized image compression," 2018, *arXiv:1809.10452*.
[11] T. T. Pham, X. V. Hoang, N. T. Nguyen, D. T. Dinh, and L. T. Ha, "End-to-end image patch quality assessment for image/video with compression artifacts," *IEEE Access*, vol. 8, pp. 215157–215172, 2020.
[12] H. Liu, T. Chen, P. Guo, Q. Shen, X. Cao, Y. Wang, and Z. Ma, "Non-local attention optimized deep image compression," 2019, *arXiv:1904.09757*.

[13] 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. 7939–7948.
[14] 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.
[15] Q. Song, J. Li, H. Guo, and R. Huang, "Denoised non-local neural network for semantic segmentation," 2021, *arXiv:2110.14200*.
[16] X. Liu, P. An, Y. Chen, and X. Huang, "An improved lossless image compression algorithm based on Huffman coding," *Multimedia Tools Appl.*, vol. 81, no. 4, pp. 4781–4795, Feb. 2022.
[17] R. Ranjan, "Canonical Huffman coding based image compression using wavelet," *Wireless Pers. Commun.*, vol. 117, no. 3, pp. 2193–2206, Apr. 2021.
[18] V. Monga and B. L. Evans, "Perceptual image hashing via feature points: Performance evaluation and tradeoffs," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3452–3465, Nov. 2006.
[19] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.
[20] Z. Guo, J. Fu, R. Feng, and Z. Chen, "Accelerate neural image compression with channel-adaptive arithmetic coding," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.
[21] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[22] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Performance comparison of convolutional autoencoders, generative adversarial networks and super-resolution for image compression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2018, pp. 2613–2616.
[23] 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," 2015, *arXiv:1511.06085*.
[24] J. Ballé, V. Laparra, and E. P. Simoncelli, "Density modeling of images using a generalized normalization transformation," 2015, *arXiv:1511.06281*.
[25] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5306–5314.
[26] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," 2017, *arXiv:1703.00395*.
[27] 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.*, vol. 30, 2017, pp. 1–11.
[28] 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.
[29] 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.
[30] 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.
[31] X. Lu, H. Wang, W. Dong, F. Wu, Z. Zheng, and G. Shi, "Learning a deep vector quantization network for image compression," *IEEE Access*, vol. 7, pp. 118815–118825, 2019.
[32] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool, "Generative adversarial networks for extreme learned image compression," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 221–231.
[33] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, vol. 2, Nov. 2004, pp. 1398–1402.
[34] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Deep residual learning for image compression," in *Proc. CVPR Workshops*, 2019, pp. 1–5.
[35] O. Rippel and L. Bourdev, "Real-time adaptive image compression," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2017, pp. 2922–2930.
[36] S. Jain and B. C. Wallace, "Attention is not explanation," 2019, *arXiv:1902.10186*.
[37] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," 2019, *arXiv:1903.10082*.
[38] J. Sun and M. F. Tappen, "Learning non-local range Markov random field for image restoration," in *Proc. CVPR*, Jun. 2011, pp. 2745–2752.
[39] L. D. Chamain, F. Racape, J. Begaint, A. Pushparaja, and S. Feltman, "End-to-end optimized image compression with attention mechanism," in *Proc. CVPR Workshop*, Mar. 2021, pp. 1–4.
[40] X. Zhu, D. Cheng, Z. Zhang, S. Lin, and J. Dai, "An empirical study of spatial attention mechanisms in deep networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6688–6697.
[41] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[42] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.
[43] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.
[44] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2018, pp. 4055–4064.
[45] B. Li, H. Ren, X. Jiang, F. Miao, F. Feng, and L. Jin, "Scep—A new image dimensional emotion recognition model based on spatial and channel-wise attention mechanisms," *IEEE Access*, vol. 9, pp. 25278–25290, 2021.
[46] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, and A. Veit, "OpenImages: A public dataset for large-scale multi-label and multi-class image classification," *Dataset*, vol. 2, no. 3, p. 18, 2017.
[47] E. Kodak. (1993). *Kodak Lossless True Color Image Suite (Photocd Pcd0992)*. vol. 6. [Online]. Available: http://r0k.us/graphics/kodak
[48] *Workshop Challenge Learned Image Compression 2020*. Accessed: Feb. 27, 2022. [Online]. Available: https://www.compression.cc/
[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, vol. 5, 2015.
[50] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 2366–2369.
[51] J. Bégaint, F. Racapé, S. Feltman, and A. Pushparaja, "CompressAI: A PyTorch library and evaluation platform for end-to-end compression research," 2020, *arXiv:2011.03029*.

AFSANA AHSAN JENY (Member, IEEE) received the bachelor's degree in computer science and engineering from Daffodil International University, in 2019. She is currently a Postgraduate Research Scholar with the Department of Computer Engineering, Bahçeşehir University (BAU), İstanbul, Turkey, and a Research Scholar at Daffodil International University, Bangladesh. She is working with the BAU Computer Vision Laboratory under the TUBITAK 2232 Project. She is a member of the Machine Intelligence Research Laboratory (MIR Lab). Previously, she has worked as a Research Assistant at the Gradient Laboratory for AI Research (GLAIR). She has published several peer-reviewed conferences and journal articles. Her research interests include computer vision, machine learning, video processing, medical image processing, and artificial intelligence.

MD BAHARUL ISLAM (Senior Member, IEEE) received the B.Sc. degree in computer science and engineering from RUET, Bangladesh, the M.Sc. degree in digital media from Nanyang Technological University, Singapore, and the Ph.D. degree in computer science from Multimedia University, Malaysia. He is currently a Faculty Member (research) and a TUBITAK 2232 Project Coordinator with the Computer Engineering Department, Bahçeşehir University (BAU). He is also an Associate Professor with the American University of Malta. Prior to that, he was a Postdoctoral Research Fellow at the AI and Augmented Vision Laboratory, Miller School of Medicine, University of Miami, USA. He has more than 12 years of working experience in teaching and cutting-edge research in image processing and computer vision. He has authored/coauthored more than 40 international peer-reviewed research papers, including journal articles, conference proceedings, books, and book chapters. His current research interests include 3D processing and AR/VR-based vision rehabilitation. He has secured several gold medals and best paper awards from different national and/or international scientific and technological competitions and conferences. His Ph.D. thesis has been selected for the Best Ph.D. Thesis by the IEEE SPS Research Excellence Award. He was awarded the International Fellowship and a Grant for Outstanding Young Researchers from the Scientific and Technological Research Council of Turkey (TUBITAK), in 2019.

MASUM SHAH JUNAYED (Member, IEEE) received the bachelor's degree in computer science and engineering from Daffodil International University, in 2019. He has worked as a Research Assistant at the Gradient Laboratory for AI Research (GLAIR). He is currently a Research Assistant at the Computer Vision Laboratory, Department of Computer Engineering, Bahçeşehir University (BAU), İstanbul, Turkey. He is a member of the Machine Intelligence Research Laboratory (MIR Lab) and the Founder and the Director of the Vision Research Laboratory of AI (VRLAI). He has published several peer-reviewed conferences and journal articles. His research interests include computer vision, machine learning, image and video processing, and medical image analysis.

DEBASHISH DAS received the Ph.D. degree in computer science from Universiti Malaysia Pahang, Malaysia. He is currently a Faculty Member and the Director of postgraduate teaching and learning at the Faculty of Computing, Engineering and the Built Environment, Birmingham City University (BCU), U.K. Overall, he has 20 years of teaching and research experience at leading. He is the author or the coauthor of many articles in international journals and conferences. His research interests include but not limited to artificial intelligence, optimization, classification, prediction, data science, data analytics, data mining and machine learning algorithms, biomedical, artificial intelligent applications, and programming languages. He has received several important recognitions for quality teaching and research, including the Top Performing Academic, the UMP Research Grant, the XMUM Research Fund, the Doctoral Scholarship Scheme (DSS), the Gold Medal in Three Minutes Thesis Completion, and the Merit Award (GOLD) for SCI Indexed Journal Publication.