frontiers | Frontiers in Signal Processing

TYPE Original Research
PUBLISHED 02 September 2022
DOI 10.3389/frsip.2022.932873

OPEN ACCESS

EDITED BY
Wenhan Yang,
Nanyang Technological University, Singapore

REVIEWED BY
Ionut Schiopu,
Huawei Technologies Oy, Finland
Miaohui Wang,
Shenzhen University, China

*CORRESPONDENCE
Ivan V. Bajić,
ibajic@ensc.sfu.ca

SPECIALTY SECTION
This article was submitted to Image Processing, a section of the journal Frontiers in Signal Processing

RECEIVED 30 April 2022
ACCEPTED 08 August 2022
PUBLISHED 02 September 2022

CITATION
Ranjbar Alvar S, Ulhaq M, Choi H and Bajić IV (2022), Joint image compression and denoising via latent-space scalability.
Front. Sig. Proc. 2:932873.
doi: 10.3389/frsip.2022.932873

Joint image compression and denoising *via* latent-space scalability

Saeed Ranjbar Alvar, Mateen Ulhaq, Hyomin Choi and Ivan V. Bajić*

School of Engineering Science, Simon Fraser University, Burnaby, BC, Canada

When it comes to image compression in digital cameras, denoising is traditionally performed prior to compression. However, there are applications where image noise may be necessary to demonstrate the trustworthiness of the image, such as court evidence and image forensics. This means that noise itself needs to be coded, in addition to the clean image itself. In this paper, we present a learning-based image compression framework where image denoising and compression are performed jointly. The latent space of the image codec is organized in a scalable manner such that the clean image can be decoded from a subset of the latent space (the base layer), while the noisy image is decoded from the full latent space at a higher rate. Using a subset of the latent space for the denoised image allows denoising to be carried out at a lower rate. Besides providing a scalable representation of the noisy input image, performing denoising jointly with compression makes intuitive sense because noise is hard to compress; hence, compressibility is one of the criteria that may help distinguish noise from the signal. The proposed codec is compared against established compression and denoising benchmarks, and the experiments reveal considerable bitrate savings compared to a cascade combination of a state-of-the-art codec and a state-of-the-art denoiser.

KEYWORDS

image denoising, image compression, deep learning, multi-task compression, scalable coding

1 Introduction

Images obtained from digital imaging sensors are degraded by the noise generated due to many factors such as lighting of the scene, sensors, shutter speed, etc. In practice, noticeable noise is often encountered in low-light conditions, as illustrated in the Smartphone Image Denoising Dataset (SIDD) Abdelhamed et al. (2018). In a typical image processing pipeline, noise in the captured image is attenuated or removed before compressing the image. The noise removed in the pre-processing stage cannot be restored, and the compressed image does not carry information about the original noise. While it is a desirable feature not to have noise in the stored image for the majority of applications, the captured noise may carry useful information for certain applications, such as court evidence, image forensics, and artistic intent. For such applications, the noise needs to be

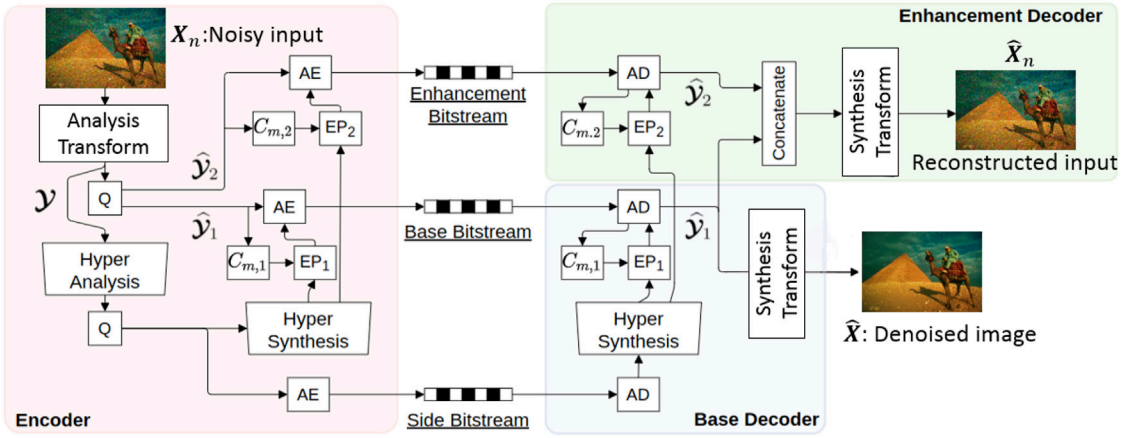

FIGURE 1
The proposed joint compression and denoising (JICD) framework. AE/AD represent arithmetic encoder and decoder, respectively. C_m and EP stand for the context model and entropy parameters, respectively. Q represents the quantizer, which is simple rounding to the nearest integer. The architecture of the individual building blocks is the same as in Cheng et al. (2020); Choi and Bajić (2021, 2022), but they have been retrained to support different tasks, as explained in the text.

preserved in the compressed image. In fact, compressed-domain denoising together with techniques to preserve the noise is part of the recent JPEG AI call for proposals ISO/IEC and ITU-T (2022a). The major drawback of encoding the noise is that it significantly increases the bitrate required for storing and transferring the images. As an example, it is known that independent and identically distributed (iid) Gaussian source, which is a common noise model, has the worst rate-distortion performance among all the sources with the same variance Cover and Thomas (2006). Another issue is that when the clean (denoised) image is needed, the denoising should be applied to the reconstructed noisy images. The additional denoising step may increase the run time and the complexity of the pipeline.

To overcome the mentioned drawbacks of encoding the noisy image and performing denoising in cascade, we present a scalable multi-task image compression framework that performs compression and denoising jointly. We borrow the terminology from scalable video coding Schwarz et al. (2007), where the input video is encoded into a scalable representation consisting of a *base layer* and one or more *enhancement layers*, which enables reconstructing various representations of the original video - different resolutions and/or frame rates and/or qualities. In the proposed Joint Image Compression and Denoising (JICD) framework, the encoder maps the noisy input to a latent representation that is partitioned into a base layer and an enhancement layer. The base layer contains the information about the clean image, while the enhancement layer contains information about noise. When the denoised image is needed, only the base layer needs to be encoded (and decoded), thereby avoiding noise coding. The enhancement layer is encoded only when the noisy input reconstruction is needed.

The scalable design of the system provides several advantages. Since only a subset of latent features is encoded for the denoised image, the bitrate is reduced compared to using the entire latent space. Another advantage is that the noise is not completely removed from the latent features, only separated from the features corresponding to the denoised image. Therefore, when the noisy input reconstruction is needed, the enhancement features are used in addition to the base features to decode the noisy input. The multi-task nature of the framework means that compression and denoising are trained jointly, and it also allows us to obtain both reconstructed noisy input and the corresponding denoised image in single forward pass, which reduces the complexity compared to the cascade implementation of compression and denoising. In fact, our results demonstrate that such a system provides improved performance—better denoising accuracy at the same bitrate—compared to a cascade combination of a state-of-the-art codec and a state-of-the-art denoiser.

The novel contributions of this paper are as follows:

- We develop JICD, the first multi-task image coding framework that supports both image denoising and noisy image reconstruction.
- JICD employs latent space scalability, such that the information about the clean image is mapped to a subset of the latent space (base layer) while noise information is mapped to the remainder (enhancement layer).
- Unlike many methods in the literature, which are either developed for a particular type of noise and/or require some noise parameter(s) in order to operate

properly, the proposed JICD is capable of handling unseen noise.

The remainder of the paper is organized as follows. Section 2 briefly describes prior work related to compression, denoising, and joint compression and denoising. Section 3 discusses the preliminaries related to learning-based multi-task image compression. Section 4 presents the proposed method. Section 5 describes the experiments and analyzes the experimental results. Finally, Section 6 presents concluding remarks.

2 Related works

The proposed JICD framework is a multi-task image codec that performs image compression and denoising jointly. In this section, we briefly discuss the most relevant works related to image denoising (Section 2.1), learning-based image compression (Section 2.2), and multi-task image compression including joint compression and denoising (Section 2.3).

2.1 Image denoising

State-of-the-art classical image denoising methods are based on Non-local Self Similarity (NSS). In these methods, repetitive local patterns in a noisy image are used to capture signal and noise characteristics, and perform denoising. In BM3D Dabov et al. (2007b), similar patches are first found by block matching. Then, they are stacked to form a 3D block. Finally, transform-domain collaborative filtering is applied to obtain the clean patch. Yahya et al. (2020) used adaptive filtering to improve BM3D. WNNM Gu et al. (2014) performs denoisng by applying low rank matrix approximation to the stacked noisy patches. In Xu et al. (2015), a patch group based NSS prior learning scheme to learn explicit NSS models from natural images is proposed. The denoising method in Zha et al. (2019) used NSS priors in both the degraded images and the external clean images to perform denoising. CBM3D Dabov et al. (2007a) and MCWNNM Xu et al. (2017) are the extensions of BM3D and WNNM, respectively, created to handle color images.

More recently (deep) learning-based denoising methods have gained popularity and surpassed the performance of classical methods. Burger et al. (2012) used a multi-layer perceptron (MLP) to achieve denoising results comparable to the state-of-the-art classic method. Among the learning-based denoisers, DnCNN Zhang et al. (2017) was the first Convolutional Neural Network (CNN) to perform blind Gaussian denoising. FFDNet Zhang et al. (2018) improved upon DnCNN by proposing a fast and flexible denoising CNN that could handle different noise levels with a single model. In Guo et al. (2019), noise estimation subnetwork is added prior to the CNN-based denoiser to get an accurate estimate of the noise level in the real-world noisy photographs. A Generative Adversarial Networks (GAN)-based denoising method is proposed in Chen et al. (2018). The mentioned works are supervised methods where clean reference image is needed for training. In Laine et al. (2019); Quan et al. (2020), self-supervised denoising methods are proposed.

2.2 Learning-based image compression

In recent years, there has been an increasing interest in the development of learning-based image codecs. Some of the early works Toderici et al. (2016); Minnen et al. (2017); Johnston et al. (2018) were based on Recurrent Neural Networks (RNNs), whose purpose was to model spatial dependence of pixels in an image. More recently, the focus has shifted to Convolutional Neural Network (CNN)-based autoencoders. Ballé et al. (2017) introduced Generalized Divisive Normalization (GDN) as a key component of the nonlinear transform in the encoder. The image codec based on GDN was improved by introducing a hyperprior to capture spatial dependencies and take advantage of statistical redundancy in the entropy model Ballé et al. (2018). To further improve the coding gains, discretized Gaussian mixture likelihoods are used in Cheng et al. (2020) to parameterize the distributions of latent codes. Most recently, this approach has been extended using advanced latent-space context modelling Guo et al. (2022) to achieve even better performance.

Most state-of-the-art learning-based image codding approaches Ballé et al. (2018); Cheng et al. (2020); Guo et al. (2022) train different models for different bitrates, by changing the Lagrange multiplier that trades-off rate and distortion. Such approach is meant to explore the potential of learning-based compression, rather than be used in practice as is. There has also been a considerable amount of work on variable-rate learning-based compression Toderici et al. (2016); Choi et al. (2019); Yang et al. (2020); Sebai (2021); Yin et al. (2022), where a single model is able to produce multiple rate-distortion points. However, in terms of rate-distortion performance, "fixed-rate" approaches such as Cheng et al. (2020); Guo et al. (2022) currently seem to have an advantage over variable-rate ones.

2.3 Multi-task image compression

The mentioned learning-based codec are single-task models, where the task is the reconstruction of the input image, just like with conventional codecs. However, the real power of learning-based codecs is their ability to be trained for multiple tasks, for example, image processing or computer vision tasks, besides the usual input reconstruction. In fact, the goal of JPEG AI standardization is to develop such a coding framework that could support multiple tasks from a common compressed representation ISO/IEC and ITU-T (2022b).

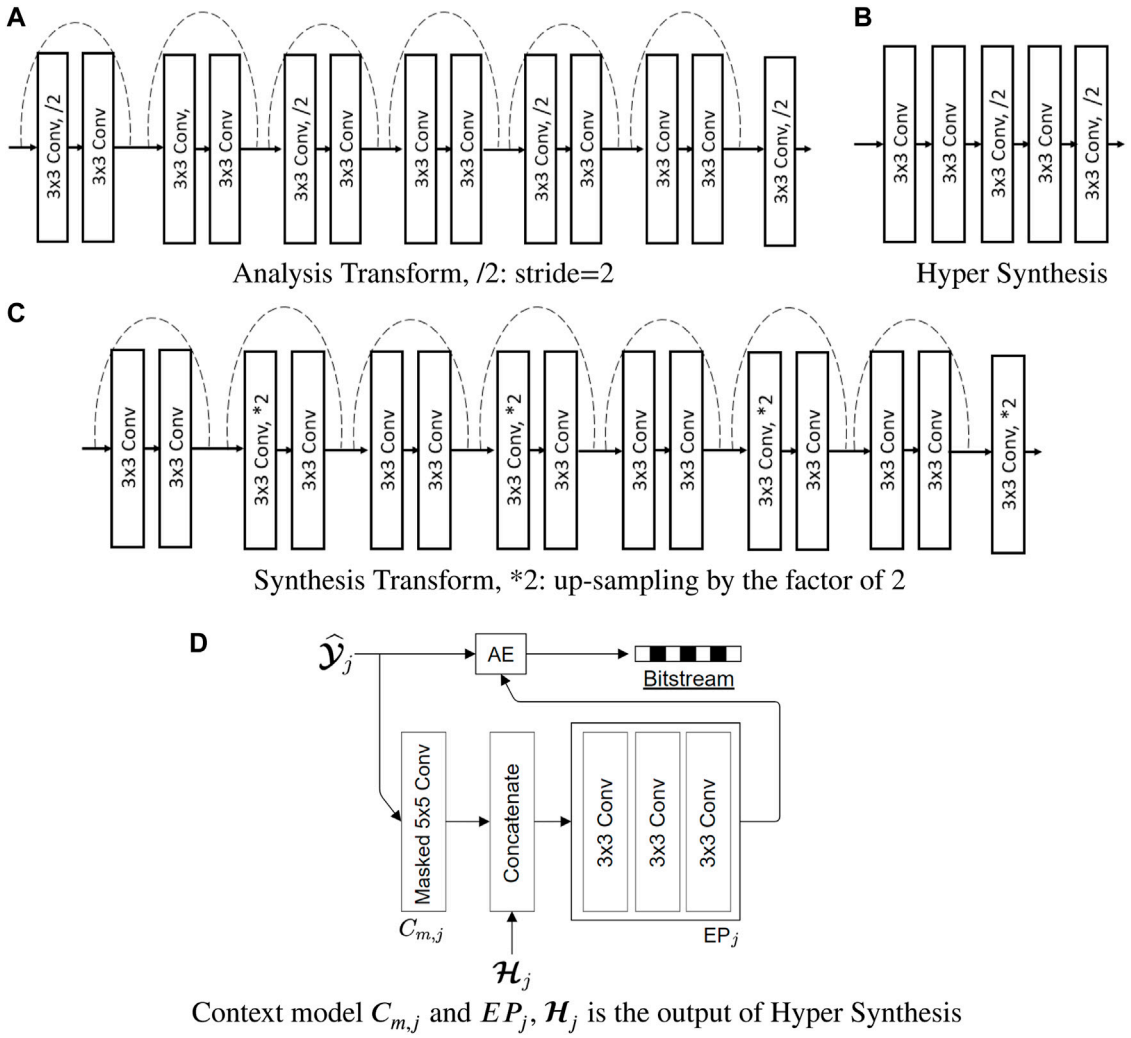

FIGURE 2
The architecture of the components inside the proposed JICD framework.

Choi and Bajić (2022) proposed a scalable multi-task model with multiple segments in the latent space to handle computer vision tasks in addition to input reconstruction. The concept was based on latent-space scalability Choi and Bajić (2021), where the latent space is partitioned in a scalable manner, from tasks that require less information to tasks that require more information. Our JICD framework is also based on latent-space scalability Choi and Bajić (2021). However, unlike these earlier works, the latent space is organized such that it supports image denoising from the base layer and noisy input reconstruction from the full latent space. In other words, the tasks are different compared to these earlier works.

Recently, Testolina et al. (2021) and Alves de Oliveira et al. (2022) developed joint image compression and denoising pipelines built upon learning-based image codecs, where the pipeline is trained to take the input noisy image, compress it, and decode a denoised image. However, with these approaches, it is not possible to reconstruct the original noisy image, hence they are not multi-task models. Our proposed JICD performs the denoising task in its base layer, but it keeps the noise information in the enhancement layer, thereby also enabling noisy input reconstruction if needed.

3 Prelimineries

In thinking about how to construct a learning-based system that can produce both the denoised image and reconstruct the noisy image, it is useful to consider the processing pipeline in which noisy image is first compressed, then decoded, and then denoising is applied to obtain the denoised image. Let $\mathbf{X}_n$ be the noisy input image. If such an image is input to a learning-based

codec Ballé et al. (2017, 2018); Minnen et al. (2018); Cheng et al. (2020), encoding would proceed in three steps:

$$\mathcal{Y} = g_a(\mathbf{X}_n; \phi) \tag{1}$$

$$\hat{\mathcal{Y}} = Q(\mathcal{Y}) \tag{2}$$

$$\mathbf{B} = A_E\left(\hat{\mathcal{Y}}\right) \tag{3}$$

where g_a is the analysis transform, ϕ represents the parameters of g_a, Q is the quantization function, and $\mathbf{B}$ is the bitstream obtained by applying the arithmetic encoder A_E to $\hat{\mathcal{Y}}$.

The noisy input image is reconstructed at the decoder by applying the entropy decoding and synthesis transform to the encoded bitstream as:

$$\hat{\mathcal{Y}} = A_D(\mathbf{B}) \tag{4}$$

$$\hat{\mathbf{X}}_n = g_s\left(\hat{\mathcal{Y}}; \theta\right) \tag{5}$$

where A_D is the entropy decoder, g_s and θ are the synthesis transform and its parameters, respectively. Then the denoised image can be obtained by applying a denoiser to the reconstructed noisy input as:

$$\hat{\mathbf{X}} = F\left(\hat{\mathbf{X}}_n, \psi\right) \tag{6}$$

where F and ψ are the denoiser and its parameters, respectively, and $\hat{\mathbf{X}}$ is the denoised image.

This processing pipeline forms a Markov chain $\mathbf{X}_n \rightarrow \hat{\mathcal{Y}} \rightarrow \hat{\mathbf{X}}_n \rightarrow \hat{\mathbf{X}}$. Applying the data processing inequality (DPI) Cover and Thomas (2006) to this Markov chain, we get

$$I\left(\hat{\mathcal{Y}}; \hat{\mathbf{X}}_n\right) \geq I\left(\hat{\mathcal{Y}}; \hat{\mathbf{X}}\right), \tag{7}$$

where $I(\cdot;\cdot)$ is the mutual information Cover and Thomas (2006) between two random quantities. Based on (7) we can conclude that latent representation $\hat{\mathcal{Y}}$ carries less information about the denoised image $\hat{\mathbf{X}}$ than it does about the noisy reconstructed image $\hat{\mathbf{X}}_n$. Moreover, because $\hat{\mathbf{X}}$ is obtained from $\hat{\mathbf{X}}_n$, the information that $\hat{\mathcal{Y}}$ carries about $\hat{\mathbf{X}}$ is a subset of the information that it carries about $\hat{\mathbf{X}}_n$. This motivates us to structure the latent representation $\hat{\mathcal{Y}}$ in such a way that only a part of it (the base layer) is used to reconstruct the denoised image $\hat{\mathbf{X}}$, while the whole of $\hat{\mathcal{Y}}$ (base + enhancement) is used to reconstruct the noisy image $\hat{\mathbf{X}}_n$.

4 Proposed method

The proposed joint image compression and denoising (JICD) framework consists of an encoder and two task-specific decoders, as illustrated in Figure 1. The architecture of the blocks that make up the encoder and two decoders in Figure 1 is shown in Figure 2. Note that the architecture of the individual building blocks (Analysis Transform, Synthesis Transform, etc.) is the same as in Cheng et al. (2020); Choi and Bajić (2021, 2022), but these blocks have been retrained to support a scalable latent representation for joint compression and denoising. Specifically, compared to Cheng et al. (2020), our encoder is trained to produce a scalable latent space that enables both denoising and noisy input reconstruction. Compared to Choi and Bajić (2021, 2022), our system is trained to support different tasks, and correspondingly the structure of the latent space and the training procedure is different. Details of individual components are described below.

TABLE 1 λ values used for training various models. Higher λ leads to higher qualities and higher bitrates.

Model index	1	2	3	4	5	6
λ	0.0035	0.0067	0.013	0.025	0.0483	0.09

4.1 Encoder

The encoder employs an analysis transform to obtain a high fidelity latent-space representation for the input image. In addition, the encoder has blocks to efficiently encode the obtained latent-space tensor. The encoder's analysis transform is borrowed from Cheng et al. (2020) due to its high compression efficiency. In addition to the analysis transform, we also adopted the entropy parameter (EP) module, the context model (CTX) for arithmetic encoder/decoder (AE/AD), synthesis transform and hyper analysis/synthesis without attention layers from Cheng et al. (2020).

The analysis transform converts the input image $\mathbf{X}$ into $\mathcal{Y} \in \mathbb{R}^{N \times M \times C}$, with $C = 192$ as in Minnen et al. (2018); Cheng et al. (2020). Unlike Minnen et al. (2018); Cheng et al. (2020), the latent representation $\mathcal{Y}$ is split into two separate sub-latents $\mathcal{Y} = \mathcal{Y}_1 \cup \mathcal{Y}_2$, $\mathcal{Y}_1 \cap \mathcal{Y}_2 = \varnothing$, where $\mathcal{Y}_1$ is the base layer containing i channels, $\mathcal{Y}_1 = \{\mathbf{Y}_1, \mathbf{Y}_2, \ldots \mathbf{Y}_i\}$, and $\mathcal{Y}_2$ is the enhancement layer containing $C - i$ channels, $\mathcal{Y}_2 = \{\mathbf{Y}_{i+1}, \mathbf{Y}_{i+2}, \ldots, \mathbf{Y}_C\}$. This allows the latent representation to be used efficiently for multiple purposes, namely denoising (from $\mathcal{Y}_1$) and noisy input reconstruction (from $\mathcal{Y}_1 \cup \mathcal{Y}_2$). Since denoising requires only $\mathcal{Y}_1$, it can be accomplished at a lower bitrate compared to decoding the full latent space. The sub-latents are then quantized to produce $\hat{\mathcal{Y}}_1$ and $\hat{\mathcal{Y}}_2$, respectively, and then coded using their respective context models to produce two independently-decodable bitstreams, as discussed in Choi and Bajić (2022, 2021). The side bitstream shown in Figure 1 is considered to be a part of the base layer and its rate is included in bitrate calculations for the base layer bitstream in the experiments.

4.2 Decoder

Two task-specific decoders are constructed: one for denoised image decoding and one for noisy input image reconstruction.

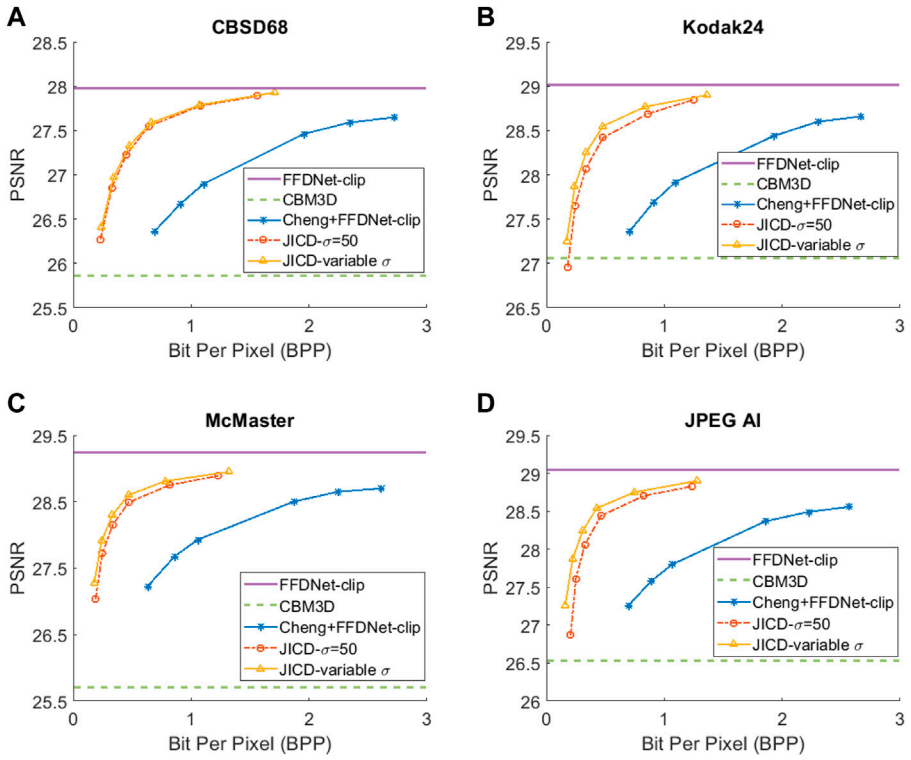

FIGURE 3
Denoising rate-PSNR curves for σ =50: **(A)** CBSD68, **(B)** Kodak24, **(C)** McMaster, **(D)** JPEG AI.

The hyperpriors used in both decoders are reconstructed from the side bitstream which, as mentioned above, is considered to be a part of the base layer. Quantized base representation $\hat{\mathcal{Y}}_1$ is reconstructed in the base decoder by decoding the base bitstream, and used to produce the denoised image $\hat{\mathbf{X}}$. Unlike Choi and Bajić (2021, 2022), where the base layer was dedicated to object detection/segmentation, our decoder does not require latent space transformation from $\hat{\mathcal{Y}}_1$ into another latent space; the synthesis transform (Figure 1) produces the denoised image $\hat{\mathbf{X}}$ directly from $\hat{\mathcal{Y}}_1$. Quantized enhancement representation $\hat{\mathcal{Y}}_2$ is decoded only when noisy input reconstruction is needed. The reconstructed noisy input image $\hat{\mathbf{X}}_n$ is produced by the second decoder using $\hat{\mathcal{Y}} = \hat{\mathcal{Y}}_1 \cup \hat{\mathcal{Y}}_2$.

Although not pursued in this work, it is worth mentioning that the proposed JICD framework can be extended to perform various computer vision tasks as well, such as image classification or object detection. These tasks typically require clean images, so one can think of the processing pipeline described by the following Markov chain: $\mathbf{X}_n \rightarrow \hat{\mathcal{Y}}_1 \rightarrow \hat{\mathbf{X}} \rightarrow T$, where T is the output of a computer vision task, for example a class label or object bounding boxes. Applying the DPI to this Markov chain we have

$$I\left(\hat{\mathcal{Y}}_1; \hat{\mathbf{X}}\right) \geq I\left(\hat{\mathcal{Y}}_1; T\right), \tag{8}$$

which implies that a subset of information from $\hat{\mathcal{Y}}_1$ is sufficient to produce T. Hence, if such tasks are required, the encoder's latent space can be further partitioned by splitting $\hat{\mathcal{Y}}_1$, in a manner similar to Choi and Bajić (2021, 2022), to support such tasks at an even lower bitrate than our base layer.

4.3 Training

The model is trained end-to-end with a rate-distortion Lagrangian loss function in the form of:

$$\mathcal{L} = R + \lambda \cdot D, \tag{9}$$

where R is an estimate of rate, D is the total distortion of both tasks, and λ is the Lagrange multiplier. The estimated rate is affected by latent and hyper-priors as in Minnen et al. (2018),

$$R = \underbrace{\mathbb{E}_{x\sim p_x}\left[-\log_2 p_{\hat{y}}(\hat{y})\right]}_{\text{latent}} + \underbrace{\mathbb{E}_{x\sim p_x}\left[-\log_2 p_{\hat{z}}(\hat{z})\right]}_{\text{hyper-priors}}, \tag{10}$$

where x denotes input data, $\hat{y}$ is the quantized latent data and $\hat{z}$ is the quantized hyper-prior. Total distortion D is computed as the

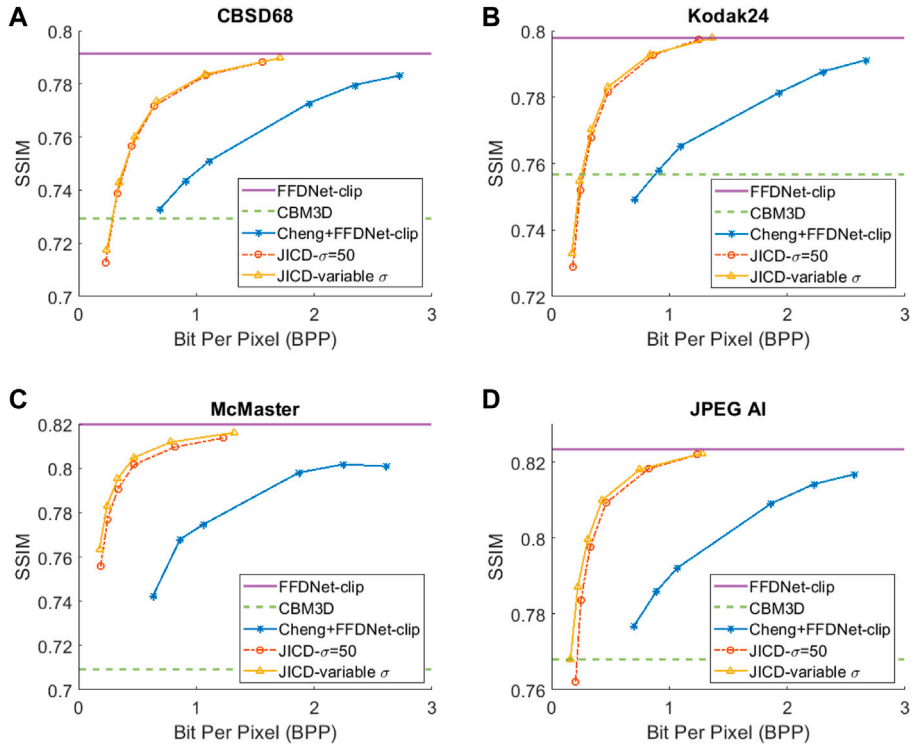

FIGURE 4
Denoising rate-SSIM curves for σ =50: **(A)** CBSD68, **(B)** Kodak24, **(C)** McMaster, **(D)** JPEG AI.

TABLE 2 The PSNR-based BD-rate of the proposed JICD compared to Cheng + FFDNet-clip on the image denoising task.

Noise type	Model	CBSD68	Kodak24	McMaster	JPEG AI
AWGN σ = 50	σ = 50	−69.28%	−72.91%	−72.69%	−74.45%
	variable σ	−70.66%	−77.27%	−76.55%	−80.20%
AWGN σ = 25	σ = 25	−30.58%	−41.00%	−33.18%	−45.13%
	variable σ	−30.28%	−42.61%	−33.52%	−45.77%
AWGN σ = 15	σ = 15	1.07%	−11.99%	−4.95%	−15.82%
	variable σ	8.00%	−2.99%	9.22%	−5.78%
Practical noise simulator	variable σ	−23.25%	−33.83%	−21.51%	−23.42%

TABLE 3 The SSIM-based BD-rate of the proposed JICD compared to Cheng + FFDNet-clip on the image denoising task.

Noise type	Model	CBSD68	Kodak24	McMaster	JPEG AI
AWGN σ = 50	σ = 50	−63.34%	−72.08%	−72.29%	−72.89%
	variable σ	−64.09%	−73.64%	−75.57%	−76.43%
AWGN σ = 25	σ = 25	−24.14%	−38.99%	−53.40%	−43.11%
	variable σ	−24.45%	−39.86%	−52.21%	−43.97%
AWGN σ = 15	σ = 15	4.52%	−11.85%	−21.57%	−15.32%
	variable σ	9.14%	−5.96%	−8.60%	−8.78%
Practical noise simulator	variable σ	−15.83%	−27.66%	−28.18%	−37.72%

weighted average of image denoising distortion and noisy input reconstruction distortion:

$$D = (1 - w) \cdot \mathrm{MSE}\left(\mathbf{X}, \hat{\mathbf{X}}\right) + w \cdot \mathrm{MSE}\left(\mathbf{X}_n, \hat{\mathbf{X}}_n\right), \qquad (11)$$

where w is the trade-off factor to adjust the importance of the tasks. The automatic differentiation Paszke et al. (2019) ensures that the gradients from D flow through the corresponding parameters without further modification to the back-propagation algorithm.

5 Experimental results

5.1 Network training

The proposed multi-task model is trained from scratch using the randomly cropped 256 × 256 patches from the CLIC dataset CLIC (2019). The noisy images are obtained using additive white Gaussian noise (AWGN) with three noise levels σ = {15, 25, 50},

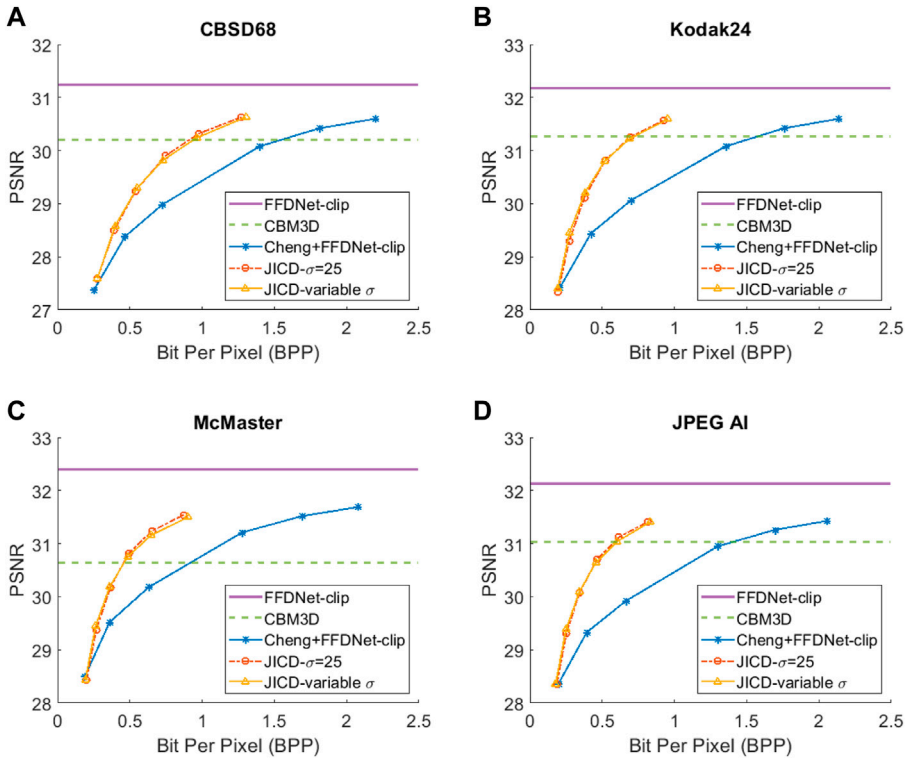

FIGURE 5
Denoising rate-PSNR curves for σ =25: **(A)** CBSD68, **(B)** Kodak24, **(C)** McMaster, **(D)** JPEG AI.

clipping the resulting values to [0, 255] and quantizing the clipped values to mimic how noisy images are stored in practice. The batch size is set to 16. Training is ran for 300 epochs using the Adam optimizer with initial learning rate of 1×10^{-4}. The learning rate is reduced by factor of 0.5 when the training loss plateaus. We trained six different models by changing the value of λ in (9). The list of different values for λ is shown in Table 1. For all the models we used $w = 0.05$ in (11). We trained the model for the first rate point (lowest λ) from scratch. However, for the remaining rate points we fine-tune the model starting from the previous rate point's weights.

We trained models under two different settings. In the first setting, a given model is trained for each noise level. For this case, the number of enhancement channels $C - i$ is chosen according to the strength of the noise. For stronger noise, we allocate more channels to the enhancement layer, so that it can capture enough information to reconstruct the noise. The number of enhancement channels is reduced as the noise gets weaker. Specifically, the number of enhancement channels is empirically set to 32, 12, and two for $\sigma = 50$, $\sigma = 25$, and $\sigma = 15$, respectively. The second training setting is to train a single model with different noise levels $\sigma \in \{50, 25, 15\}$ simultaneously, and use the final trained model to perform denoising for all noise levels. This is beneficial when the noise level information is not given. In this model, we used 180 base channels and 12 enhancement channels. σ at each training iteration is uniformly chosen from {50, 25, 15}.

5.2 Data

To evaluate the performance of the proposed JICD framework, four color image datasets are used: 1) CBSD68 Martin et al. (2001), 2) Kodak24 Franzen (1999), 3) McMaster Zhang et al. (2011) and 4) JPEG AI testset ISO/IEC and ITU-T (2022b), which is used in the JPEG AI exploration experiments. The mentioned datasets contain 68, 24, 18, and 16 images, respectively. The resolution of the images in the Kodak24 and McMaster dataset is fixed to 500, ×, 500. CBS68 dataset contains the lowest-resolution images among the four datasets, with the height and width of images ranging between 321 and 481. The images in the JPEG AI testset are high-resolution images with the height varying between 872 and 2,456 pixels and width varying between 1,336 and 3,680 pixels. The results are reported for two sets of noisy images. In the first set, we added synthesized AWGN to the testing images with three noise levels: $\sigma = \{15, 25, 50\}$ and tested the results with the quantized noisy images. In the second set, we used the synthesized noise obtained from the noise simulator in

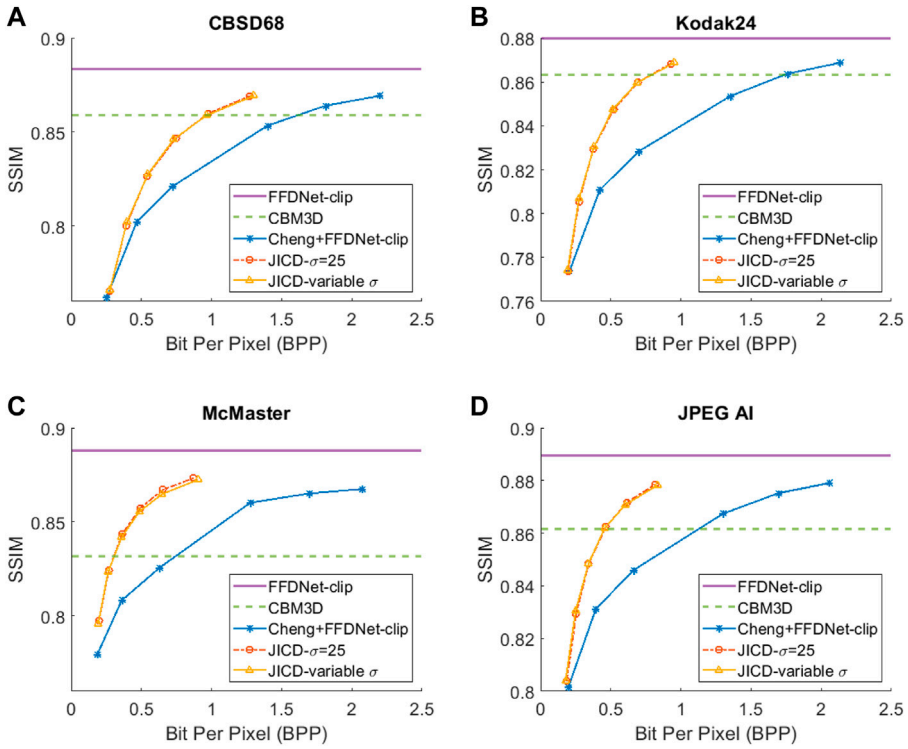

FIGURE 6
Denoising rate-SSIM curves for σ =25: **(A)** CBSD68, **(B)** Kodak24, **(C)** McMaster, **(D)** JPEG AI.

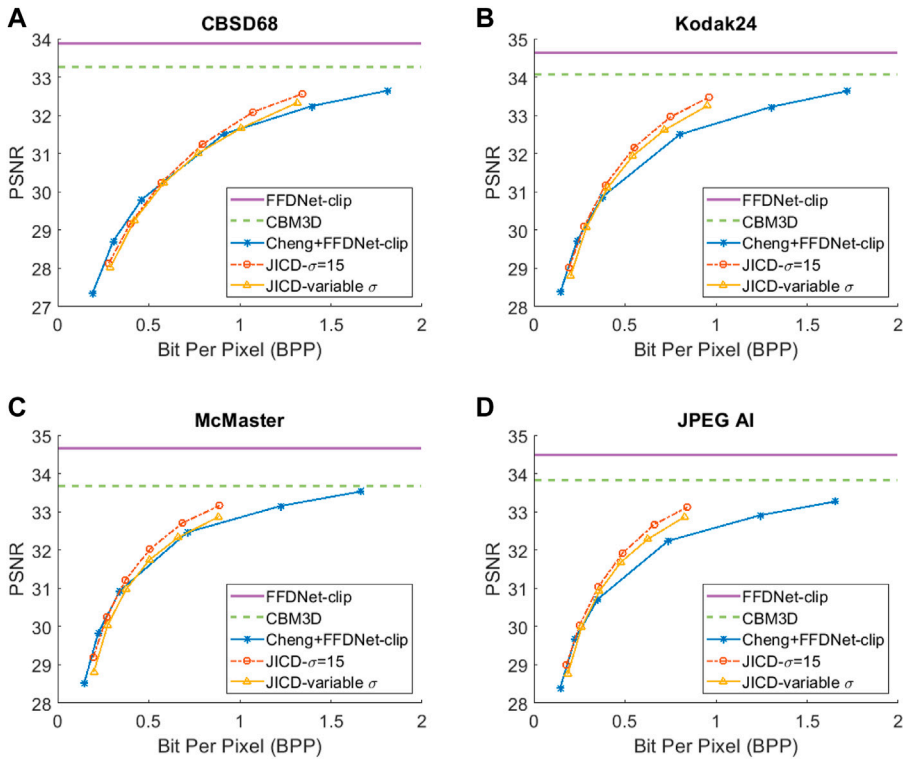

FIGURE 7
Denoising rate-PSNR curves for σ =15: **(A)** CBSD68, **(B)** Kodak24, **(C)** McMaster, **(D)** JPEG AI.

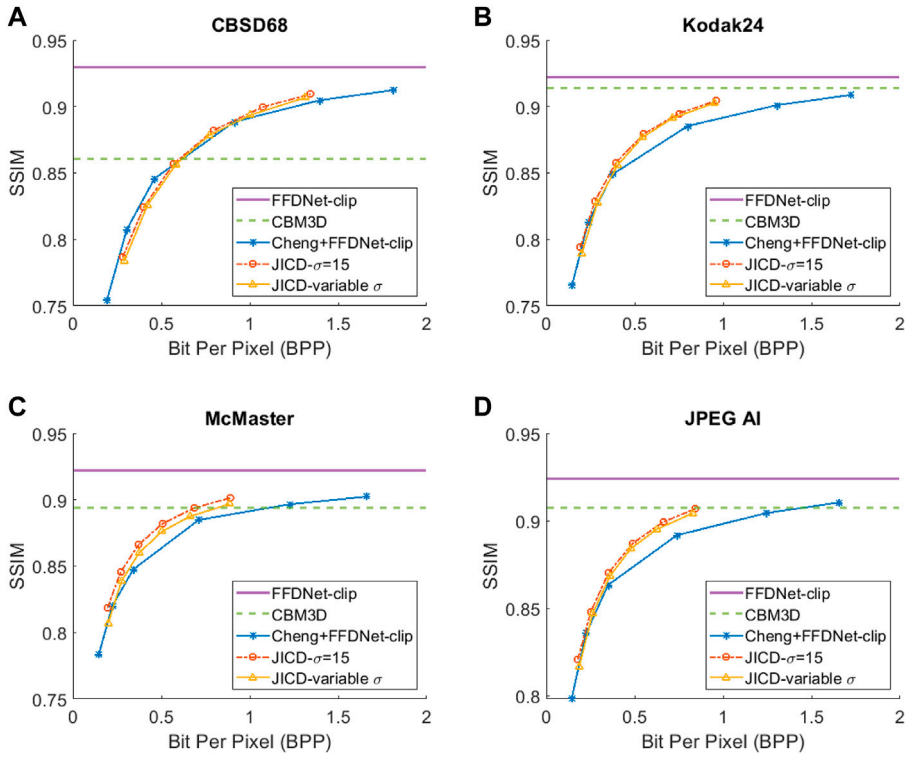

FIGURE 8
Denoising rate-SSIM curves for σ =15: **(A)** CBSD68, **(B)** Kodak24, **(C)** McMaster, **(D)** JPEG AI.

Ranjbar Alvar and Bajić (2022), which was also used to generate the final test images for the denoising tasks in the ongoing JPEG AI standardization. This type of noise was not used during the training of the proposed JICD framework. Hence, the goal of testing with this second set of images is to evaluate how well the proposed JICD generalizes to the noise that is not seen during the training.

5.3 Baselines

The denoising performance of the proposed JICD framework is compared against well-established baselines: CBM3D Dabov et al. (2007a) and FFDNet Zhang et al. (2018). CBM3D is a NSS-based denoising method, and FFDNet belongs to the learning-based denoising category. FFDNet was trained using AWGN with different noise levels during the training. At inference time, FFDNet needs the variance of the noise as input. FFDNet-clip Zhang et al. (2018) is a version of FFDNet that is trained with quantized noisy images. Since our focus is on practical settings with quantized noisy images, we used FFDNet-clip as a baseline in the experiments. We also tested the DRUNet denoiser Zhang et al. (2021), which is one of the latest state-of-the-art denoisers. DRUNet assumes that the noise is not quantized, and when tested with quantized noise, it performs worse than FFDNet-clip. As a result, we did not include it in the experiments.

Two baselines are established by applying CBM3D and FFDNet-clip directly on noisy images, without compression. However, to assess the interaction of compression and denoising, we establish one more baseline. In this third baseline, the noisy image is first compressed using the end-to-end image compression model from Cheng et al. (2020) (the "Cheng model") with an implementation from CompressAI Bégaint et al. (2020), and then decoded. Then FFDNet-clip is used to denoise the decoded noisy image. We call this cascade denoising approach as Cheng + FFDNet-clip. It is worth mentioning that Cheng + FFDNet-clip, similar to the proposed JICD framework, is able to obtain both the reconstructed noisy images and denoised images, hence it could be considered a multi-task approach.

5.4 Experiments on AWGN removal

We evaluate the baselines and the proposed JICD method using the quantized noisy images obtained using AWGN with three noise levels, $\sigma \in \{15, 25, 50\}$. The test results with the strongest noise (σ = 50) across the four datasets (CBSD68, Kodak24, McMaster, and

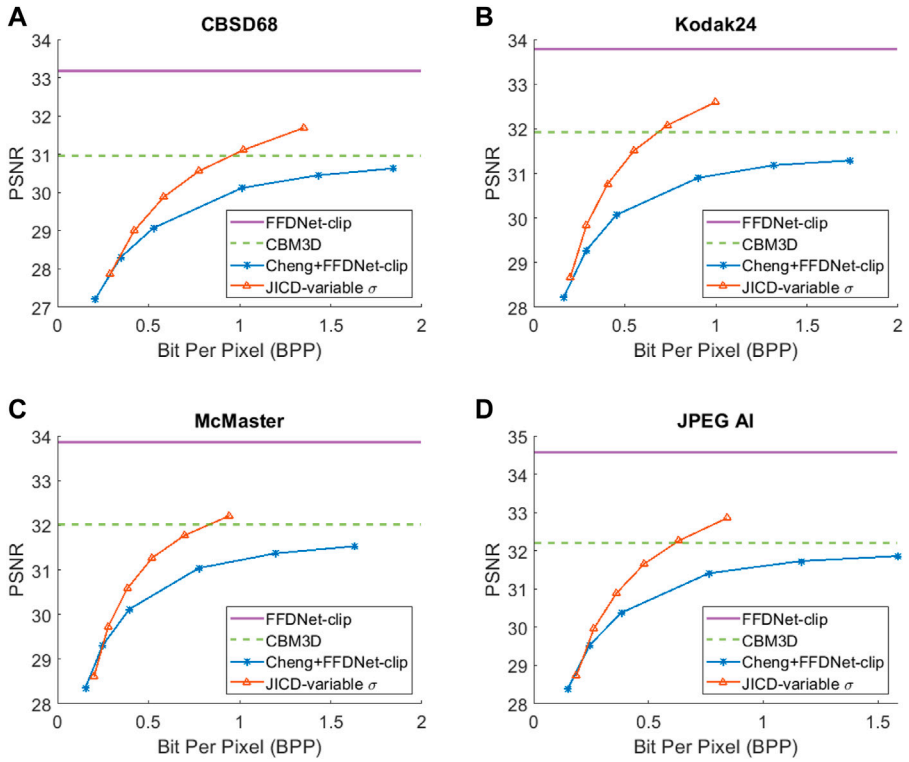

FIGURE 9
Denoising rate-PSNR curves for the unseen noise: **(A)** CBSD68, **(B)** Kodak24, **(C)** McMaster, **(D)** JPEG AI.

JPEG AI) are shown Figure 3 in terms of rate vs Peak Signal-to-Noise Ratio (PSNR) and in Figure 4 in terms of rate vs Structural Similarity Index Measure (SSIM). The horizontal lines in the figure correspond to applying CBM3D and FFDNet-clip to the raw (uncompressed) noisy images. The blue curve shows the results for Cheng + FFDNet-clip. The six points on this curve correspond to the six Cheng models from CompressAI Bégaint et al. (2020). For JICD, two curves are shown. The orange curve shows the results obtained from the models trained for $\sigma = 50$ with 160 base feature channels and 32 enhancement channels. The yellow curve corresponds to the results obtained using the model that was trained with variable σ values and has 180 base and 12 enhancement channels. The six points on the orange and yellow curves correspond to the six JICD models we trained with λ values shown in Table 1.

As seen in Figure 3, for $\sigma = 50$, the quality of the images denoised by CBM3D is considerably lower compared to those obtained using FFDNet-clip. It was shown in Zhang et al. (2018) that CBM3D and FFDNet-clip achieve comparable performance for non-quantized noisy images. Our results show that CBM3D's performance is degraded when the noise deviates (due to clipping and quantization) from the assumed model, at least at high noise levels.

The comparison of the results obtained by JICD and Cheng + FFDNet-clip reveal that JICD is able to reduce the bitrate substantially while achieving the same denoising performance as Cheng + FFDNet-clip. This is due to the fact that the Cheng model allocates the entire latent representation to noisy input reconstruction, whereas the proposed method uses a subset of the latent features to perform denoising. The results of JICD trained with variable σ are also shown in the curves. Since the number of base channels is larger in this model compared to the model trained for $\sigma = 50$, its denoising performance is improved.

To summarize the differences between the performance-rate curves, we compute Bjøntegaard Delta-rate (BD-rate) Bjøntegaard (2001). The BD-rate of the proposed JICD compared to Cheng + FFDNet-clip on the four datasets is given in the first two rows of Table 2 for PSNR, and Table 3 for SSIM. It can be seen that the proposed method achieves up to 80.2% BD-rate savings compared to Cheng + FFDNet-clip. Both JICD and Cheng + FFDNet-clip denoising methods outperform CBMD3D for all the tested rate points at $\sigma = 50$. Using the proposed JICD method, we are able to denoise images at a quality close to what FFDNet-clip achieves on raw images, and at the same time compress the input.

We repeat the denoising experiment for $\sigma = 25$, and the results are shown in Figure 5 for PSNR and Figure 6 for SSIM. As seen in the figures, the gap between the CBM3D and FFDNet-clip performance is now reduced, and the compression-based methods now outperform CBM3D only at the higher rates. The

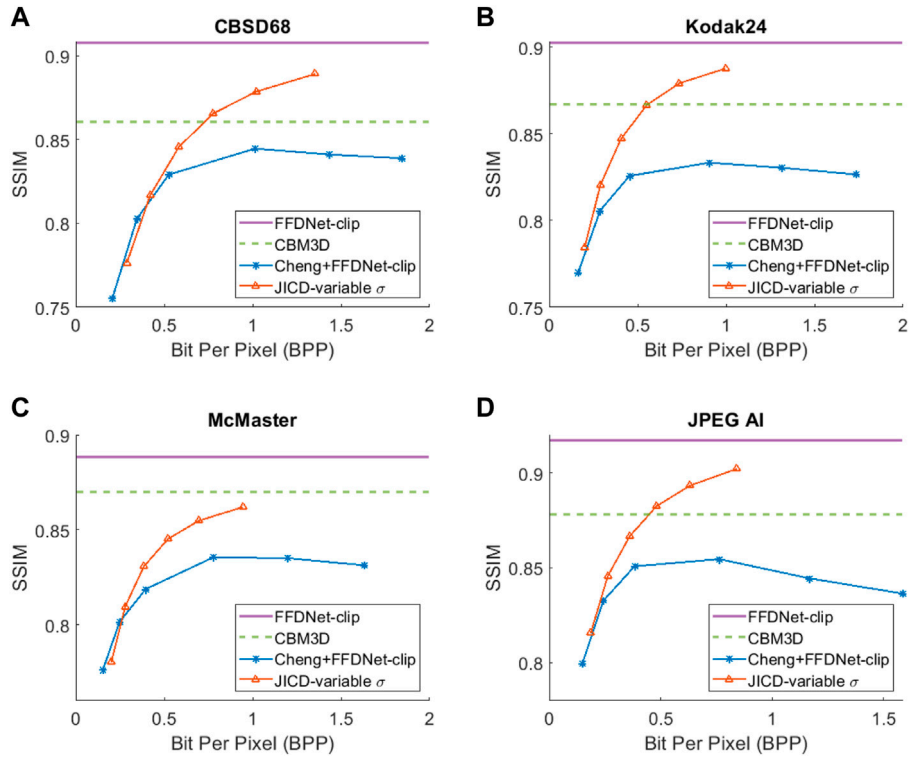

FIGURE 10
Denoising rate-SSIM curves for the unseen noise: **(A)** CBSD68, **(B)** Kodak24, **(C)** McMaster, **(D)** JPEG AI.

gap between the curves corresponding to JICD and Cheng + FFDNet-clip is also reduced. However, JICD still achieves a considerable BD-rate saving compared to Cheng + FFDNet-clip, as shown in the third row of Table 2 and Table 3. JICD trained with variable σ has slightly better PSNR performance compared to the noise-specific model on three datasets, and a slightly worse performance (by 0.3%) on the low-resolution CBSD68 dataset.

At the lowest noise level ($\sigma = 15$), the gap between CBM3D and FFDNet-clip shrinks further. It can be seen in the denoised rate-PSNR curves in Figure 7 and rate-SSIM curves in Figure 8 that when the noise is weak, applying denoising to the raw images achieves high PSNR, and the compression-based methods cannot outperform either CBM3D, or FFDNet-clip at the tested rates. The gap between JICD and Cheng + FFDNet-clip curves is also reduced compared to the higher noise levels. This can also be seen from the BD-rates in the fourth row of Tables 2, 3. JICD trained for $\sigma = 15$ outperforms Cheng + FFDNet-clip on three datasets, but it suffers a 1% (4.5% for SSIM) loss on the low-resolution CBSD68.

As seen above, the performance of the proposed JICD framework is lower on the low-resolution CBSD68 dataset than on other datasets. The reason is the following. The processing pipeline USED in JICD expects the input dimensions to be multiples of 64. For images whose dimensions do not satisfy this requirement, the input is padded up to the nearest multiple of 64. At low resolutions, the padded area may be somewhat large in relation to the original image, which causes noticeable performance degradation. At high resolutions, the padded area is insignificant compared to the original image, and the impact on JICD's performance is correspondingly smaller. It is worth mentioning that for $\sigma = 15$, the JICD trained with variable σ has a weaker denoising performance compared to the model trained specifically for $\sigma = 15$. This is because the number of base channels in the variable-σ model (180) is smaller than the number of base channels in the noise-specific model (190). At low noise levels, fewer channels are needed to hold noise information, which means the number of base channels could be higher. Hence, the structure chosen for the noise-specific model is better suited for this case. However, we show in the next subsection that the variable-σ model is more useful when the noise parameters are not known.

5.5 Experiments on unseen noise removal

5.5.1 Image compression and denoising

The proposed JICD denoiser and the baselines are also tested with the noise that was not used in the training. The purpose of this experiment is to evaluate how well the denoisers are able to handle unseen noise. To generate unseen noise, we used the noise

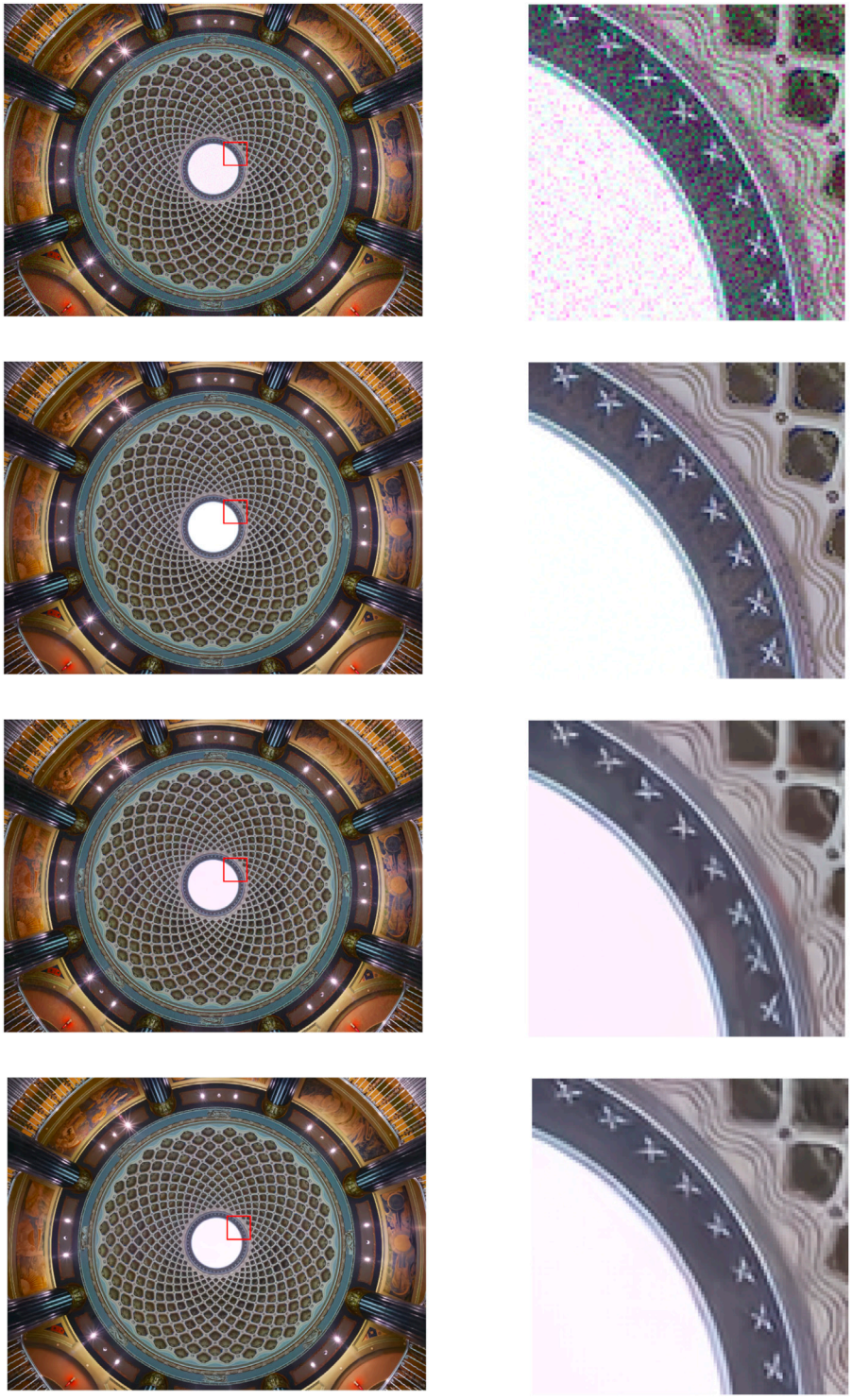

FIGURE 11
An example of denoised images. Top to Bottom: noisy image, clean image, Denoised: Cheng + FFDNet-clip (bpp = 0.57), Denoised: proposed (bpp = 0.55). Images in the right column show the red square in the left images.

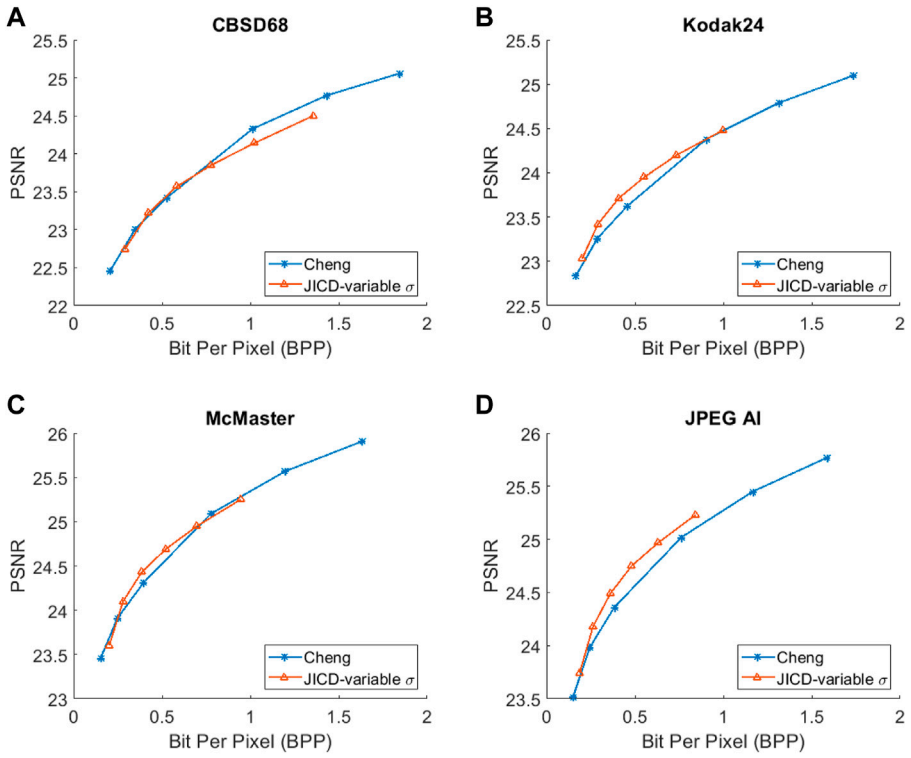

FIGURE 12
The rate-PSNR curves for noisy input reconstruction. **(A)** CBSD68, **(B)** Kodak24, **(C)** McMaster, **(D)** JPEG-AI.

simulator from Ranjbar Alvar and Bajić (2022). This noise simulator, which we subsequently refer to as "practical noise simulator," was created by fitting the Poissonian-Gaussian noise model Foi et al. (2008) to the noise from the Smartphone Image Denoising Dataset (SIDD) Abdelhamed et al. (2018). It is worth mentioning that this noise simulator is used in the evaluation of the image denoising task in JPEG AI standardization.

For this experiment we use the JICD model trained with variable σ. One advantage of this model is that, unlike some of the baselines, it does not require any additional input or noise information, besides the noisy image. On the other hand, FFDNet needs σ to perform denoising. In the experiment, the σ is estimated for each image by computing the standard deviation of the difference between the noisy test image and the corresponding clean image.

The denoising rate-PSNR and rate-SSIM curves are illustrated in Figures 9, 10, respectively. Since the variance of the noise obtained from the practical noise simulator is not large, the PSNR range of the denoised images is close to that observed in the AWGN experiments with $\sigma = 15$ and $\sigma = 25$. The results indicate that JICD achieves better denoising performance compared to Cheng + FFDNet-clip across all four datasets. Moreover, at higher bitrates (1 bpp and above), JICD outperforms CBM3D applied to uncompressed noisy images. BD-rate results are summarized in the last row of Tables 2, 3. It is seen in the table that JICD achieves 15–30% gain over Cheng + FFDNer-clip across the four datasets.

A visual example comparing the denoised images obtained from JICD and Cheng + FFDNet-clip encoded at similar bitrates is shown in Figure 11. As seen in the figure, JICD preserves more details compared to Cheng + FFDNet-clip. In addition, the colors inside the white circle are reproduced closer to the ground truth with JICD compared to the image produced by Cheng + FFDNet-clip.

5.5.2 Noisy image reconstruction

Besides denoising, the proposed JICD framework is also able to reconstruct the noisy input image when enhancement features are decoded together with base features. While the main focus of this work was on denoising (and the majority of experiments devoted to that goal), for completeness we also evaluate the noisy image reconstruction performance using unseen noise. We compare the noisy input reconstruction performance of JICD against Cheng et al. (2020), i.e., the compression model used earlier in the Cheng + FFDNet-clip baseline. The PSNR between the noisy input and the reconstructed noisy images is shown against bitrate in Figure 12, while Figure 13 shows SSIM vs bitrate. As illustrated in Figure 12, our JICD achieves better noisy input reconstruction compared to Cheng et al. (2020) in most cases. BD-rate results corresponding to Figures 12, 13 are given in Tables 4, 5, respectively. As the numbers in Table 4 indicate, the proposed JICD achieves noticeable BD-rate savings on three of the four test datasets; the only exception is, again, the low-resolution CBSD68 dataset, where the loss is mainly

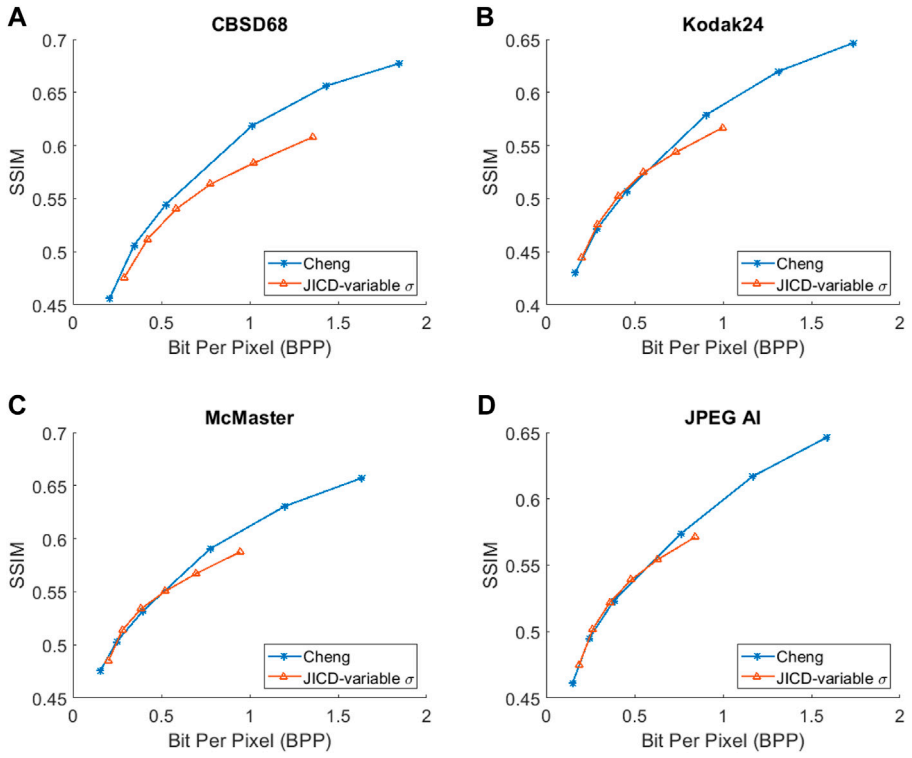

FIGURE 13
The rate-SSIM curves for noisy input reconstruction. **(A)** CBSD68, **(B)** Kodak24, **(C)** McMaster, **(D)** JPEG-AI.

TABLE 4 The PSNR-based BD-rate of the proposed JICD compared to the Cheng model on noisy input reconstruction.

Noise type	Model	CBSD68	Kodak24	McMaster	JPEG AI
Practical noise simulator	variable σ	5.50%	−11.74%	−3.97%	−13.49%

TABLE 5 The SSIM-based BD-rate of the proposed JICD compared to the Cheng model on noisy input reconstruction.

Noise type	Model	CBSD68	Kodak24	McMaster	JPEG AI
Practical noise simulator	variable σ	22.58%	1.90%	4.05%	0.58%

concentrated at higher bitrates. It is worth noting that, since our proposed method is trained using the MSE loss, it performs better in terms of PSNR than SSIM. Overall, the proposed JICD framework achieves gains on both denoising and compression tasks compared to Cheng + FFDNet-clip and Cheng et al. (2020) models.

6 Conclusion

In this work, we presented a joint image compression and denoising framework. The proposed framework is a scalable multi-task image compression model based on the latent-space scalability. The base features are used to perform the denoising and the enhancement features are used when the noisy input reconstruction is needed. Extensive experiments show that the proposed framework achieves significant BD-rate savings up to 80.20% across different dataset compared to the cascade compression and denoising method. The experimental results also indicate that the proposed method achieves improved results for the unseen noise for both denoising and noisy input reconstruction tasks.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: For CBSD8 and Kodak24 datasets: https://github.com/cszn/KAIR, For McMaster dataset: https://www4.comp.polyu.edu.hk/~cslzhang/CDM_Dataset.htm, For JPEG AI: https://jpeg.org/jpegai/dataset.html.

Author contributions

SA and IB contributed to conception and design of the study. HC developed the initial code. MU and SA contributed to further code development and optimization. SA wrote the first draft of the manuscript and worked with IB on the revisions.

Funding

Funding for this work was provided by the Natural Sciences and Engineering Research Council (NSERC) of Canada under the grants RGPIN-2021-02485 and RGPAS-2021-00038, and by Huawei Technologies.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abdelhamed, A., Lin, S., and Brown, M. S. (2018). A high-quality denoising dataset for smartphone cameras. *Proc. CVPR* 18, 1692–1700. doi:10.1109/cvpr.2018.00182

Alves de Oliveira, V., Chabert, M., Oberlin, T., Poulliat, C., Bruno, M., Latry, C., et al. (2022). Satellite image compression and denoising with neural networks. *IEEE Geosci. Remote Sens. Lett.* 19, 1–5. doi:10.1109/lgrs.2022.3145992

Ballé, J., Laparra, V., and Simoncelli, E. (2017). End-to-end optimized image compression. *Proc. ICLR* 17.

Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. (2018). Variational image compression with a scale hyperprior. *Proc*, ICLR'18.

Bégaint, J., Racapé, F., Feltman, S., and Pushparaja, A. (2020). Compressai: A pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*.

Bjøntegaard, G. (2001). "VCEG-M33: Calculation of average PSNR differences between RD-curves," in Video coding experts group (VCEG) *(ITU –telecommunications standardization)*.

Burger, H. C., Schuler, C. J., and Harmeling, S. (2012). "Image denoising: Can plain neural networks compete with bm3d?," in 2012 IEEE conference on computer vision and pattern recognition (IEEE), 2392–2399.

Chen, J., Chen, J., Chao, H., and Yang, M. (2018). "Image blind denoising with generative adversarial network based noise modeling," in Proceedings of the IEEE conference on computer vision and pattern recognition, 3155–3164.

Cheng, Z., Sun, H., Takeuchi, M., and Katto, J. (2020). Learned image compression with discretized Gaussian mixture likelihoods and attention modules. *Proc. IEEE CVPR* 20, 7936–7945. doi:10.1109/cvpr42600.2020.00796

Choi, H., and Bajić, I. V. (2021). Latent-space scalability for multi-task collaborative intelligence. *Proc. IEEE ICIP.*, 3562–3566. doi:10.1109/icip42928.2021.9506712

Choi, H., and Bajić, I. V. (2022). Scalable image coding for humans and machines. *IEEE Trans. Image Process.* 31, 2739–2754. doi:10.1109/tip.2022.3160602

Choi, Y., El-Khamy, M., and Lee, J. (2019). Variable rate deep image compression with a conditional autoencoder. *Proc. IEEE/CVF ICCV.*, 3146–3154.

CLIC (2019). Challenge on learned image compression (CLIC). [Online]: http://www.compression.cc/.

Cover, T. M., and Thomas, J. A. (2006). *Elements of information theory*. 2nd edn. Wiley.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2007a). Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space. *Proc. IEEE ICIP* 07 (1), 313–316. I –.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2007b). Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Process.* 16, 2080–2095. doi:10.1109/tip.2007.901238

Foi, A., Trimeche, M., Katkovnik, V., and Egiazarian, K. (2008). Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *IEEE Trans. Image Process.* 17, 1737–1754. doi:10.1109/tip.2008.2001399

Franzen, R. (1999). *Kodak lossless true color image suite. source:* http://r0k.us/graphics/kodak 4.

Gu, S., Zhang, L., Zuo, W., and Feng, X. (2014). Weighted nuclear norm minimization with application to image denoising. *Proc. IEEE CVPR* 14, 2862–2869.

Guo, S., Yan, Z., Zhang, K., Zuo, W., and Zhang, L. (2019). Toward convolutional blind denoising of real photographs. *Proc. IEEE CVPR* 19, 1712–1722.

Guo, Z., Zhang, Z., Feng, R., and Chen, Z. (2022). Causal contextual prediction for learned image compression. *IEEE Trans. Circuits Syst. Video Technol.* 32, 2329–2341. doi:10.1109/tcsvt.2021.3089491

ISO/IEC and ITU-T (2022a). Final call for proposals for JPEG AI. ISO/IEC JTC 1/SC29/WG1 N100095

ISO/IEC and ITU-T (2022b). JPEG AI use cases and requirements. ISO/IEC JTC 1/SC29/WG1 N100094

Johnston, N., Vincent, D., Minnen, D., Covell, M., Singh, S., Chinen, T., et al. (2018). Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. *Proc. IEEE/CVF CVPR.*, 4385–4393. doi:10.1109/cvpr.2018.00461

Laine, S., Karras, T., Lehtinen, J., and Aila, T. (2019). High-quality self-supervised deep image denoising. *Adv. Neural Inf. Process. Syst.* 32.

Martin, D. R., Fowlkes, C. C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *Proc. IEEE ICCV* 01, 416–425.

Minnen, D., Ballé, J., and Toderici, G. D. (2018). Joint autoregressive and hierarchical priors for learned image compression. *Adv. Neural Inf. Process. Syst.* 31, 10771–10780.

Minnen, D., Toderici, G., Covell, M., Chinen, T., Johnston, N., Shor, J., et al. (2017). Spatially adaptive image compression using a tiled deep network. *Proc. IEEE ICIP.*, 2796–2800.

Paszke, A., Gross S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems* (MIT Press), 32, 8024–8035. Curran Associates, Inc.

Quan, Y., Chen, M., Pang, T., and Ji, H. (2020). "Self2self with dropout: Learning self-supervised denoising from single image," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 1890–1898.

Ranjbar Alvar, S., and Bajić, I. V. (2022). Practical noise simulation for RGB images. *arXiv preprint arXiv:2201.12773*.

Schwarz, H., Marpe, D., and Wiegand, T. (2007). Overview of the scalable video coding extension of the h.264/avc standard. *IEEE Trans. Circuits Syst. Video Technol.* 17, 1103–1120. doi:10.1109/TCSVT.2007.905532

Sebai, D. (2021). Multi-rate deep semantic image compression with quantized modulated autoencoder. *Proc. IEEE MMSP.*, 1–6.

Testolina, M., Upenik, E., and Ebrahimi, T. (2021)., 11842. San Diego, CA: SPIE, 412–422.Towards image denoising in the latent space of learning-based compression*Appl. Digital Image Process. XLIV*

Toderici, G., O'Malley, S. M., Hwang, S. J., Vincent, D., Minnen, D., Baluja, S., et al. (2016). Variable rate image compression with recurrent neural networks. In ICLR

Xu, J., Zhang, L., Zhang, D., and Feng, X. (2017). Multi-channel weighted nuclear norm minimization for real color image denoising. *Proc. IEEE Int. Conf. Comput. Vis.*, 1096–1104. doi:10.1109/iccv.2017.125

Xu, J., Zhang, L., Zuo, W., Zhang, D., and Feng, X. (2015). Patch group based nonlocal self-similarity prior learning for image denoising. *Proc. IEEE Int. Conf. Comput. Vis.*, 244–252. doi:10.1109/iccv.2015.36

Yahya, A. A., Tan, J., Su, B., Hu, M., Wang, Y., Liu, K., et al. (2020). Bm3d image denoising algorithm based on an adaptive filtering. *Multimed. Tools Appl.* 79, 20391–20427. doi:10.1007/s11042-020-08815-8

Yang, F., Herranz, L., Weijer, J. v. d., Guitián, J. A. I., López, A. M., and Mozerov, M. G. (2020). Variable rate deep image compression with modulated autoencoder. *IEEE Signal Process. Lett.* 27, 331–335. doi:10.1109/lsp.2020.2970539

Yin, S., Li, C., Bao, Y., Liang, Y., Meng, F., and Liu, W. (2022). Universal efficient variable-rate neural image compression. *Proc. IEEE ICASSP.*, 2025–2029.

Zha, Z., Yuan, X., Wen, B., Zhang, J., Zhou, J., and Zhu, C. (2019). "Simultaneous nonlocal self-similarity prior for image denoising," in 2019 IEEE International Conference on Image Processing (ICIP) (IEEE), 1119–1123.

Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., and Timofte, R. (2021). Plug-and-play image restoration with deep denoiser prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1. doi:10.1109/tpami.2021.3088914

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017). Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.* 26, 3142–3155. doi:10.1109/tip.2017.2662206

Zhang, K., Zuo, W., and Zhang, L. (2018). FFDNet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Trans. Image Process.* 27, 4608–4622. doi:10.1109/tip.2018.2839891

Zhang, L., Wu, X., Buades, A., and Li, X. (2011). Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *J. Electron. Imaging* 20, 023016. doi:10.1117/1.3600632