

Optimizing Image Compression via Joint Learning with Denoising

Ka Leong Cheng^{*}, Yueqi Xie^{*}, and Qifeng Chen

The Hong Kong Univeristy of Science and Technology, Hong Kong, China
{klchengad,yxieay}@connect.ust.hk, cqf@ust.hk

Abstract. High levels of noise usually exist in today’s captured images due to the relatively small sensors equipped in the smartphone cameras, where the noise brings extra challenges to lossy image compression algorithms. Without the capacity to tell the difference between image details and noise, general image compression methods allocate additional bits to explicitly store the undesired image noise during compression and restore the unpleasant noisy image during decompression. Based on the observations, we optimize the image compression algorithm to be noise-aware as joint denoising and compression to resolve the bits mis-allocation problem. The key is to transform the original noisy images to noise-free bits by eliminating the undesired noise during compression, where the bits are later decompressed as clean images. Specifically, we propose a novel two-branch, weight-sharing architecture with plug-in feature denoisers to allow a simple and effective realization of the goal with little computational cost. Experimental results show that our method gains a significant improvement over the existing baseline methods on both the synthetic and real-world datasets. Our source code is available at: <https://github.com/felixcheng97/DenoiseCompression>.

Keywords: Joint Method; Image Compression; Image Denoising

1 Introduction

Lossy image compression has been studied for decades with essential applications in media storage and transmission. Many traditional algorithms [50,60] and learned methods [3,5,13,44,65] are proposed and widely used. Thanks to the fast development of mobile devices, smartphones are becoming the most prevalent and convenient choice of photography for sharing. However, the captured images usually contain high levels of noise due to the limited sensor and aperture size in smartphone cameras [1]. Since existing compression approaches are designed for general images, the compressors treat the noise as “crucial” information and explicitly allocate bits to store it, even though noise is usually undesired for common users. The image noise can further degrade the compression quality, especially at medium and high bit rates [2,48]. Concerning these aspects, we see

^{*} Joint first authors

the crucial need for an image compression method with the capacity of noise removal during the compression process.

A natural and straightforward solution is to go through a sequential pipeline of individual denoising and compression methods. However, a simple combination of separate models can be sub-optimal for this joint task. On the one hand, sequential methods introduce additional time overhead due to the intermediate results, leading to a lower efficiency than a united solution. The inferior efficiency can limit their practical applications, especially on mobile devices. On the other hand, a sequential solution suffers from the accumulation of errors and information loss in the individual models. Most image denoising algorithms have strong capabilities of noise removal for the flat regions but somehow over-smooth the image details [66]. However, the details are the critical parts of information that need to be kept for compression. Lossy image compression algorithms save bits through compressing local patterns with a certain level of information loss, particularly for the high-frequency patterns. However, both image details and noise are considered high-frequency information, so a general image compressor is likely to eliminate some useful high-frequency details while misallocating bits to store the unwanted noise instead. In the area of image processing, many researchers explore to develop joint solutions instead of using sequential approaches, such as combined problems of joint image demosaicing, denoising, or super-resolution [21,40,66].

In this paper, we contribute a joint method to optimize the image compression algorithm via joint learning with denoising. The key challenge of this joint task is to resolve the bit misallocation issue on the undesired noise when compressing the noisy images. In other words, the joint denoising and compression method needs to eliminate only the image noise while preserving the desired high-frequency content so that no extra bits are wastefully allocated for encoding the noise information in the images. Some existing works attempt to integrate the denoising problem into image compression algorithms. Prior works [22,49] focus on the decompression procedure and propose joint image denoise-decompression algorithms, which take the noisy wavelets coefficients as input to restore the clean images, but leave the compressing part untouched. A recent work [54] attempts to tackle this task by adding several convolutional layers into the decompressor to denoise the encoded latent features on the decoding side. However, their networks can inevitably use additional bits to store the noise in the latent features since there are no particular designs of modules or supervision for denoising in the compressor, leading to their inferior performance compared to the sequentially combined denoising and compression solutions.

We design an end-to-end trainable network with a simple yet effective novel two-branch design (a denoising branch and a guidance branch) to resolve the bit misallocation problem in joint image denoising and compression. Specifically, we hope to pose explicit supervision on the encoded latent features to ensure it is noise-free so that we can eliminate high-frequency noise while, to a great extent, preserving useful information. During training, the denoising and guidance branches have shared encoding modules to obtain noisy features from the noisy

input image and the guiding features from the clean input image, respectively; efficient denoising modules are plugged into the denoising branch to denoise the noisy features as noise-free latent codes. The explicit supervision is posed in high-dimensional space from the guiding features to the encoded latent codes. In this way, we can train the denoiser to help learn a noise-free representation. Note that the guidance branch is disabled during inference.

We conduct extensive experiments for joint image denoising and compression on both the synthetic data under various noise levels and the real-world SIDD [1]. Our main contributions are as follows:

- We optimize image compression on noisy images through joint learning with denoising, aiming to avoid bit misallocation for the undesired noise. Our method outperforms baseline methods on both the synthetic and real-world datasets by a large margin.
- We propose an end-to-end joint denoising and compression network with a novel two-branch design to explicitly supervise the network to eliminate noise while preserving high-frequency details in the compression process.
- Efficient plug-in feature denoisers are designed and incorporated into the denoising branch to enable the denoising capacity of the compressor with only little addition of complexity during inference time.

2 Related Work

2.1 Image Denoising

Image denoising is an age-long studied task with many traditional methods proposed over the past decades. They typically rely on certain pre-defined assumptions of noise distribution, including sparsity of image gradients [9,53] and similarity of image patches [16,24]. With the rapid development of deep learning, some methods [11,25,62] utilize CNNs to improve the image denoising performance based on the synthetic [20,61,64] and real-world datasets, including DND [47], SIDD [1], and SID [11]. Some works [26,33,69] focus on adapting solutions from synthetic datasets to real-world scenarios. Some current state-of-the-art methods are proposed to enhance performance further, including DANet [68] utilizing an adversarial framework and InvDN [41] leveraging the invertible neural networks. However, many learning-based solutions rely on heavy denoising models, which are practically inefficient for the joint algorithms, especially in real-world applications.

2.2 Lossy Image Compression

Many traditional lossy image compression solutions [7,23,31,50,60] are widely proposed for practical usage. They map an image to quantized latent codes through hand-crafted transformations and compress them using entropy coding. With vast amounts of data available these days, many learning-based solutions are developed to learn a better transformation between image space and feature

space. RNN-based methods [30,56,58] are utilized to iteratively encode residual information in the images, while most of the recent solutions are based on variational autoencoders (VAEs) [4,55] to optimize the whole image compression process directly. Some methods [5,13,27,29,37,44,45] focus on improving the entropy models to parameterize the latent code distribution more accurately. Some others [38,65] design stronger architectures to learn better transformations for image compression. For example, Lin et al. [38] introduce spatial RNNs to reduce spatial redundancy, Mentzer et al. [42] integrate generative adversarial networks for high-fidelity generative image compression, and Xie et al. [65] utilize invertible neural networks to form a better reversible process. However, these existing compression methods generally do not consider the image noise in their designs.

2.3 Joint Solutions

A series of operations are usually included in a whole image or video processing pipeline, while a pipeline with separate solutions can suffer from the accumulation of errors from individual methods. Thus, many joint solutions have been proposed for various combinations of tasks. Several widely-studied ones for image processing include joint denoising and demosaicing [15,18,21,32,35], joint denoising and super-resolution [70], and joint demosaicing and super-resolution [19,59,67]. Recently, Xing et al. [66] further propose to solve a joint triplet problem of image denoising, demosaicing, and super-resolution. As for video processing, Norkin et al. [46] integrate the idea of separating film grain from video content into the AV1 video codec, which can be viewed as a joint solution of video denoising and compression. However, the task for joint image denoising and compression has not received much attention yet important. Some works like [22,49] only target at building decompression methods that restore clean images from the noisy bits by integrating the denoising procedure in the decompression process. A recent work [54] also incorporates the denoising idea into image decompression by performing denoising on the encoded latent codes during decompression. These approaches cannot achieve our goal of solving the bits misallocation problem and cannot achieve pleasant rate-distortion performance since denoising during decompression cannot produce noise-free bits.

3 Problem Specification

We wish to build an image compression method that takes noise removal into consideration during compression since noise is usually unwanted for general users while requiring additional bits for storage. Hence, the benefit of such a compressor lies in saving storage for the unwanted noise during the compression process. Formally, given a noisy image $\tilde{\mathbf{x}}$ with its corresponding clean ground truth image \mathbf{x} , the compressor takes $\tilde{\mathbf{x}}$ as input to denoise and compress it into denoised bitstreams. We can later decompress the bitstreams to get the denoised image $\hat{\mathbf{x}}$. Meanwhile, instead of sequentially doing denoising and successive compression or vice versa, we require the whole process to be end-to-end optimized as a united system to improve efficiency and avoid accumulation of errors.

3.1 Selection of Datasets

It is desirable to have a large number of diverse samples to train a well-performing learned image compression network. Many real-world datasets such as DND [47], SIDD [1], and SID [11] have been proposed for image denoising with noisy-clean image pairs. However, they generally have limited training samples, scene diversities, or noise levels because collecting a rich (large scale, various gains, illuminance, etc.) real-world dataset typically includes much time-consuming labor work. In contrast, synthetic data is cheap and unlimited, and it is flexible to synthesize images with different levels of noise. Therefore, the main experiments in this paper are carried out using synthetic data; additional experiments on the SIDD [1] are also conducted to further verify the effectiveness of our method. We use the SIDD only for real-world datasets because RNI15 [36] has no clean ground truths; SID [11] is for image denoising in the dark; DND [47] only allows 5 monthly submissions, which is not suitable for the image compression task that requires evaluations at various bit rates.

3.2 Noise Synthesis

We use a similar strategy as in [43] to do noise synthesis in raw, where the following sRGB gamma correction function Γ is used to transform between the image sRGB domain \mathbf{X} and the raw linear domain \mathbf{Y} :

$$\mathbf{X} = \Gamma(\mathbf{Y}) = \begin{cases} m\mathbf{Y}, & \mathbf{Y} \leq b, \\ (1+a)\mathbf{Y}^{1/\gamma} - a, & \mathbf{Y} > b, \end{cases} \quad (1)$$

where $a = 0.055$, $b = 0.0031308$, $m = 12.92$, $\gamma = 2.4$. Specifically, the inverse function Γ^{-1} is first applied on sRGB image \mathbf{x} to get the raw image $\mathbf{y} = \Gamma^{-1}(\mathbf{x})$. The noise in raw images is defined as the standard deviation of the linear signal, ranging from 0 to 1. Given a true signal intensity y_p at position p , the corresponding noisy measurement \tilde{y}_p in the noisy raw image $\tilde{\mathbf{y}}$ is estimated by a two-parameter signal-dependent Gaussian distribution [28]:

$$\tilde{y}_p \sim \mathcal{N}(y_p, \sigma_s y_p + \sigma_r^2), \quad (2)$$

where σ_s and σ_r denote the shot and readout noise parameters, respectively, indicating different sensor gains (ISO). After the noise synthesis in raw space, we obtain our noisy sRGB image $\tilde{\mathbf{x}} = \Gamma(\tilde{\mathbf{y}})$.

4 Method

Our joint denoising and compression method is inherently an image compression algorithm with the additional capacity to remove undesirable noise. Hence, the proposed method for image denoise-compression is built upon the learned image compression methods. Fig. 1 shows an overview of the proposed method. Our network contains a novel two-branch design for the training process, where the guiding features in the guidance branch pose explicit supervision on the denoised features in the denoising branch during the compression process.

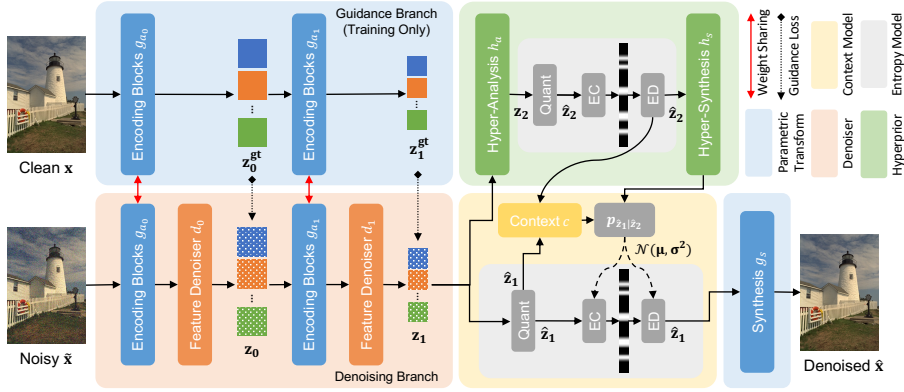


Fig. 1: Overview of the two-branch design of our proposed network, which is first pre-trained on clean images and successively fine-tuned on noisy-clean image pairs. In the top left of the figure, the clean image goes through the guidance branch for the two-level guiding features; in the bottom left, the noisy image is fed into the denoising branch to obtain the two-level denoised features. Note that the guidance branch is for training only, and that the denoising branch (orange part) and the denoisers (orange blocks) are only activated during fine-tuning and used for inference. The right half of the figure contains the common hyperprior, entropy models, context model, and synthesis transform used in the recent learned compression methods [13,44].

4.1 Network Design

Overall workflow. The network contains a parametric analysis transform g_a (containing g_{a_0} and g_{a_1}) with plug-in feature denoisers d (containing d_0 and d_1) to encode and denoise the input noisy image $\tilde{\mathbf{x}}$ into some denoised latent features \mathbf{z}_1 . Discrete quantization is then applied to obtain the quantized latent features $\hat{\mathbf{z}}_1$. Instead of using the non-differentiable discrete rounding function during training, we add a uniform noise $\mathcal{U}(-0.5, 0.5)$ on top of \mathbf{z}_1 to get $\tilde{\mathbf{z}}_1$, which can be view as an approximation of the discrete quantization process [4]. For notation simplicity, we use $\hat{\mathbf{z}}_1$ to represent both $\hat{\mathbf{z}}_1$ and $\tilde{\mathbf{z}}_1$ in this paper. Then accordingly, we have a parametric synthesis transform g_s that decodes $\hat{\mathbf{z}}_1$ to obtain the denoised image $\hat{\mathbf{x}}$. The parametric transforms g_a , g_s and the denoiser d formulate a basic variational model for the joint image denoising and compression task.

As discussed in Ballé et al. [5], there still remain significant spatial dependencies within the latent features $\hat{\mathbf{z}}_1$ using a basic variational model. Hence, a similar scale hyperprior is appended on top of the basic variational model. In particular, the hyperprior contains parametric transform h_a to model the spatial dependencies and obtain the additional latent features \mathbf{z}_2 , so that we can assume that the target variables \mathbf{z}_1 conditioned on the introduced latent \mathbf{z}_1 are indepen-

dent [8]. We adopt the same uniform noise strategy on \mathbf{z}_2 to obtain $\tilde{\mathbf{z}}_2$ during training and perform discrete quantization for $\hat{\mathbf{z}}_2$ during testing. Similarly, we use $\hat{\mathbf{z}}_2$ to represent $\tilde{\mathbf{z}}_2$ and $\tilde{\mathbf{z}}_2$ for notation simplicity. Together with the causal context model c , another parametric synthesis transform h_s transforms $\hat{\mathbf{z}}_2$ to estimate the means $\hat{\boldsymbol{\mu}}$ and standard deviations $\hat{\boldsymbol{\sigma}}$ for the latent features $\hat{\mathbf{z}}_1$ so that each element of the latent features is modeled as mean and scale Gaussian [44]:

$$p_{\hat{\mathbf{z}}_1|\hat{\mathbf{z}}_2} \sim \mathcal{N}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}^2). \quad (3)$$

Similar to [4], the distribution of $\hat{\mathbf{z}}_2$ is modeled as $p_{\hat{\mathbf{z}}_2|\theta}$ by a non-parametric, factorized entropy model θ because the prior knowledge is not available for $\hat{\mathbf{z}}_2$.

Two-branch architecture. The noisy image $\tilde{\mathbf{x}}$ and the corresponding clean image \mathbf{x} are fed into the denoising and guidance branches, respectively. Similar to many denoising methods [10,12], the plug-in denoisers d_0 and d_1 are designed in a multiscale manner. The two-level guiding features \mathbf{z}_0^{gt} and \mathbf{z}_1^{gt} are obtained by the parametric analysis g_{a_0} and g_{a_1} , respectively; the two-level denoised features \mathbf{z}_0 and \mathbf{z}_1 are obtained by the parametric analysis g_{a_0} and g_{a_1} plus the denoisers d_0 and d_1 , respectively:

$$\begin{aligned} \mathbf{z}_0^{\text{gt}} &= g_{a_0}(\mathbf{x}), & \mathbf{z}_0 &= g_{a_0}(\tilde{\mathbf{x}}) + d_0(g_{a_0}(\tilde{\mathbf{x}})), \\ \mathbf{z}_1^{\text{gt}} &= g_{a_1}(\mathbf{z}_0^{\text{gt}}), & \mathbf{z}_1 &= g_{a_1}(\mathbf{z}_0) + d_1(g_{a_1}(\mathbf{z}_0)). \end{aligned} \quad (4)$$

Note that the weights are shared for the parametric analysis g_{a_0} and g_{a_1} in two branches, and the denoisers are implemented in a residual manner. To enable direct supervision for feature denoising, a multiscale guidance loss is posed on the latent space to guide the learning of the denoisers. Specifically, the two-level guidance loss \mathcal{G} is to minimize the \mathcal{L}_1 distance between the denoised and guiding features:

$$\mathcal{G} = \|\mathbf{z}_0 - \mathbf{z}_0^{\text{gt}}\|_1 + \|\mathbf{z}_1 - \mathbf{z}_1^{\text{gt}}\|_1. \quad (5)$$

4.2 Rate-Distortion Optimization

Some entropy coding methods like arithmetic coding [52] or asymmetric numeral systems (ANS) [17] are utilized to losslessly compress the discrete latent features $\hat{\mathbf{z}}_1$ and $\hat{\mathbf{z}}_2$ into bitstreams, which are the two parts of information needed to be stored during compression. As an inherent compression task, we hope the storing bitstreams are as short as possible; as a joint task with denoising, we hope to minimize the difference between the decoded image $\hat{\mathbf{x}}$ and the clean image \mathbf{x} . Hence, it is natural to apply the rate-distortion (RD) objective function \mathcal{L}_{rd} in this joint task:

$$\mathcal{L}_{rd} = \mathcal{R}(\hat{\mathbf{z}}_1) + \mathcal{R}(\hat{\mathbf{z}}_2) + \lambda_d \mathcal{D}(\mathbf{x}, \hat{\mathbf{x}}). \quad (6)$$

In the context of rate-distortion theory, the to-be-coded sources are the noisy images $\tilde{\mathbf{x}}$, and the distortion is measured with respect to the corresponding clean counterparties \mathbf{x} . Similar to [13,44], the rate \mathcal{R} denotes the rate levels for the bitstreams, which is defined as the entropy of the latent variables:

$$\begin{aligned} \mathcal{R}(\hat{\mathbf{z}}_1) &= \mathbb{E}_{\tilde{\mathbf{x}} \sim p_{\tilde{\mathbf{x}}}} [-\log_2 p_{\hat{\mathbf{z}}_1|\hat{\mathbf{z}}_2}(\hat{\mathbf{z}}_1|\hat{\mathbf{z}}_2)], \\ \mathcal{R}(\hat{\mathbf{z}}_2) &= \mathbb{E}_{\tilde{\mathbf{x}} \sim p_{\tilde{\mathbf{x}}}} [-\log_2 p_{\hat{\mathbf{z}}_2|\theta}(\hat{\mathbf{z}}_2|\theta)]. \end{aligned} \quad (7)$$

The formulation of the distortion \mathcal{D} is different for MSE and MS-SSIM [63] optimizations, which is either $\mathcal{D} = \text{MSE}(\mathbf{x}, \hat{\mathbf{x}})$ or $\mathcal{D} = 1 - \text{MS-SSIM}(\mathbf{x}, \hat{\mathbf{x}})$. The factor λ_d governs the trade-off between the bit rates \mathcal{R} and the distortion \mathcal{D} .

4.3 Training Strategy

Pre-training as image compression. Given that our joint image denoising and compression method is an inherent image compression algorithm, we first pre-train our network with only the guidance branch, where to-be-coded sources are the clean images \mathbf{x} and the distortion is also measured with respect to \mathbf{x} . In this way, the compression capacity is enabled for our model with properly trained parameter weights, except for the denoiser d . In the supplements, we further present some ablation studies showing that the pre-training process on image compression can benefit and significantly boost the performance of the joint network.

Fine-tuning under multiscale supervision. The next step is to properly train the plug-in denoisers d_1, d_2 to enable the denoising capacity in the denoising branch. Specifically, noisy-clean image pairs are fed into the denoising and guidance branches accordingly for model fine-tuning, with both the rate-distortion loss \mathcal{L}_{rd} and the guidance loss \mathcal{G} . In this way, the full objective function \mathcal{L} during fine-tuning becomes

$$\mathcal{L} = \mathcal{R}(\hat{\mathbf{z}}_1) + \mathcal{R}(\hat{\mathbf{z}}_2) + \lambda_d \mathcal{D}(\mathbf{x}, \hat{\mathbf{x}}) + \lambda_g \mathcal{G}(\mathbf{z}_0, \mathbf{z}_0^{\text{gt}}, \mathbf{z}_1, \mathbf{z}_1^{\text{gt}}), \quad (8)$$

where $\lambda_g = 3.0$ is empirically set as the weight factor for the guidance loss.

5 Experiments

5.1 Experimental Setup

Synthetic datasets. The Flickr 2W dataset [39] is used for training and validation, which consists of 20,745 general clean images. Similar to [65], images smaller than 256 pixels are dropped for convenience, and around 200 images are selected for validation. The Kodak PhotoCD image dataset (Kodak) [14] and the CLIC Professional Validation dataset (CLIC) [57] are used for testing, which are two common datasets for the image compression task. There are 24 high-quality 768×512 images in the Kodak dataset and 41 higher-resolution images in the CLIC dataset.

We use the same noise sampling strategy as in [43] during training, where the readout noise parameter σ_r and the shot noise parameter σ_s are uniformly sampled from $[10^{-3}, 10^{-1.5}]$ and $[10^{-4}, 10^{-2}]$, respectively. As for the validation and testing, the 4 pre-determined parameter pairs $(\sigma_r, \sigma_s)^*$ in [43]’s official test set are used. Please note that Gain $\propto 4$ (slightly noisier) and Gain $\propto 8$ (significantly noisier) levels are unknown to the network during training. We test at

* Gain $\propto 1 = (10^{-2.1}, 10^{-2.6})$, Gain $\propto 2 = (10^{-1.8}, 10^{-2.3})$, Gain $\propto 4 = (10^{-1.4}, 10^{-1.9})$, Gain $\propto 8 = (10^{-1.1}, 10^{-1.5})$.

full resolution on the Kodak and CLIC datasets with pre-determined levels of noise added.

Real-world datasets. The public SIDD-Medium [1] dataset, containing 320 noisy-clean sRGB image pairs for training, is adopted to further validate our method on real-world noisy images. The SIDD-Medium dataset contains 10 different scenes with 160 scene instances (different cameras, ISOs, shutter speeds, and illuminance), where 2 image pairs are selected from each scene instance. Following the same settings in image denoising tasks, the models are validated on the 1280 patches in the SIDD validation set and tested on the SIDD benchmark patches by submitting the results to the SIDD website.

Training details. For implementation, we use the anchor model [13] as our network architecture (without d_1 and d_2) and choose the bottleneck of a single residual attention block [13] for the plug-in denoisers d_1 and d_2 . During training, the network is optimized using randomly cropped patches at a resolution of 256 pixels. All the models are fine-tuned on the pre-trained anchor models [13] provided by the popular CompressAI PyTorch library [6] using a single RTX 2080 Ti GPU. Some ablation studies on the utilized modules and the training strategy can be found in our supplements.

The networks are optimized using the Adam [34] optimizer with a mini-batch size of 16 for 600 epochs. The initial learning rate is set as 10^{-4} and decayed by a factor of 0.1 at epoch 450 and 550. Some typical techniques are utilized to avoid model collapse due to the random-initialized denoisers at the start of the fine-tuning process: 1) We warm up the fine-tuning process for the first 20 epochs. 2) We have a loss cap for each model so that the network will skip the optimization of a mini step if the training loss is beyond the set threshold value.

We select the same hyperparameters as in [13] to train compression models target a high compression ratio (relatively low bit rate) for practical reasons. Lower-rate models (q_1, q_2, q_3) have channel number $N = 128$, usually accompanied with smaller λ_d values. The channel number N is set as 192 for higher-rate models (q_4, q_5, q_6) and optimized using larger λ_d values. We train our MSE models under all the 6 qualities, with λ_d selected from the set $\{0.0018, 0.0035, 0.0067, 0.0130, 0.0250, 0.0483\}$; the corresponding λ_d values for MS-SSIM (q_2, q_3, q_5, q_6) are chosen from $\{4.58, 8.73, 31.73, 60.50\}$.

Evaluation metrics. For the evaluation of rate-distortion (RD) performance, we use the peak signal-to-noise ratio (PSNR) and the multiscale structural similarity index (MS-SSIM) [63] with the corresponding bits per pixel (bpp). The RD curves are utilized to show the denoising and coding capacity of various models, where the MS-SSIM metric is converted to $-10 \log_{10}(1 - \text{MS-SSIM})$ as prior work [13] for better visualization.

5.2 Rate-Distortion Performance

The sequential methods contains individual models of the state-of-the-art denoising DeamNet [51] and the anchor compression model Cheng2020 [13]. We compare our method with the following baseline methods: 1) ‘‘Cheng2020+DeamNet’’:

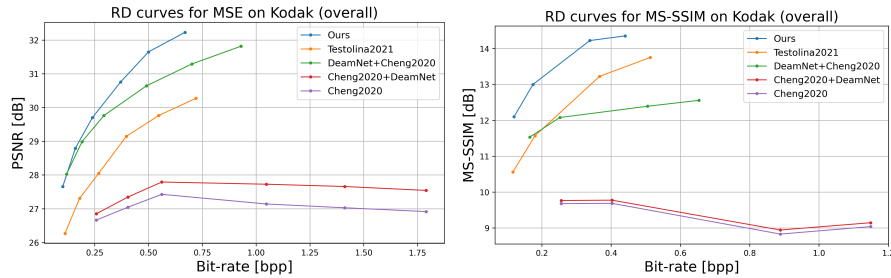


Fig. 2: Overall RD curves on the Kodak dataset at all noise levels. Our method has better RD performance over the pure compression, the sequential, and the joint baseline methods.

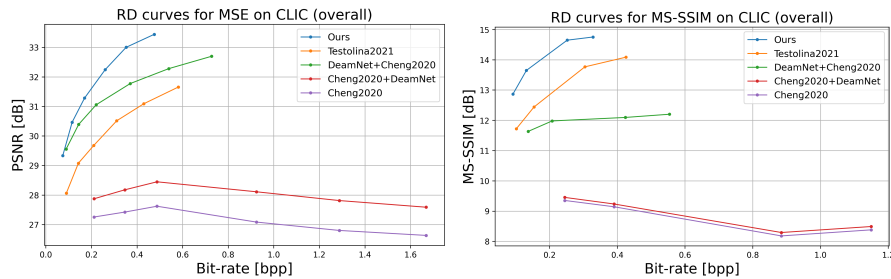


Fig. 3: Overall RD curves on the CLIC dataset at all noise levels. Our method has better RD performance over the pure compression, the sequential, and the joint baseline methods.

sequential method of Cheng2020 and DeamNet; 2) “DeamNet+Cheng2020”: sequential method of DeamNet and Cheng2020; 3) “Testolina2021”: the joint baseline method [54]. We also report the performance of the pure image compression model “Cheng2020” on noisy-clean image pairs. Note that since a pure image compression model is trained to faithfully reconstruct the input image and is not expected to do any extra noisy-to-clean mapping, “Cheng2020” is only for qualitatively demonstrating the limitation of the current pure compression models as a reference.

For compression models, we use the pre-trained model provided by CompressAI [6]. For the denoising models on SIDD, we use the officially pre-trained DeamNet; for models on synthetic data, we retrain DeamNet from scratch on the same synthetic training data as ours. We re-implement “Testolina2021” as the joint baseline method according to their original paper [54]. The RD results are obtained from the CompressAI evaluation platform and the official SIDD website. More quantitative results are available in our supplements.

Synthetic noise (overall). We show the overall (containing all the 4 noise levels) RD curves for both the MSE and MS-SSIM methods evaluated on the

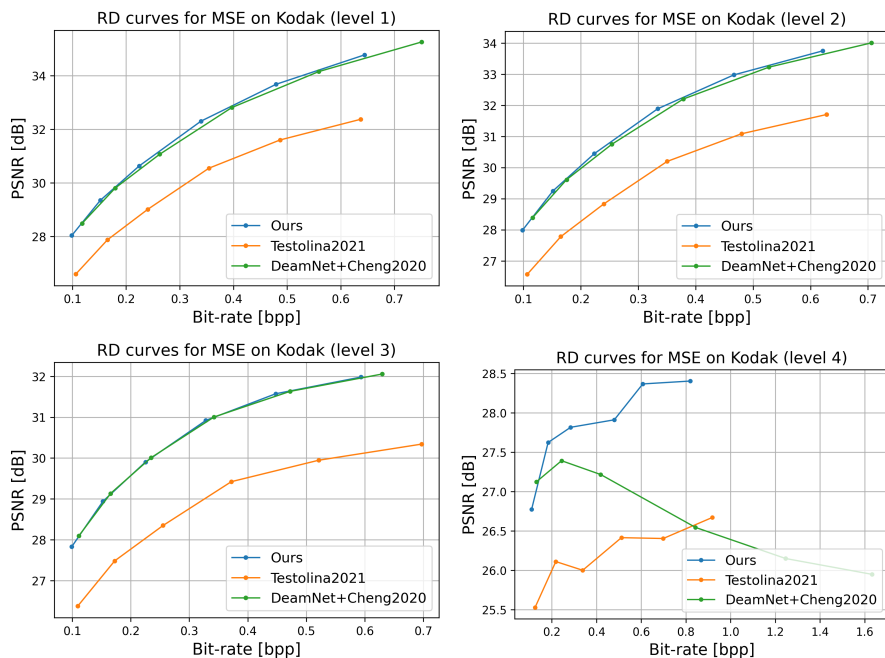


Fig. 4: RD curves on the Kodak dataset at individual noise level. Our method outperforms the baseline solutions, especially at the highest noise level.

Kodak dataset in Fig. 2 and on the CLIC dataset in Fig. 3. We can observe that our method (the blue RD curves) yields much better overall performance than the pure compression method, the sequential methods, and the joint baseline method.

For sequential methods, the green and red RD curves show that both sequential solutions have inferior performance compared to our joint solution. The execution order of the individual methods also matters. Intuitively, the sequential method that performs compression and successively denoising can suffer from the information loss and waste of bits allocating to image noise caused by the bottleneck of the existing general image compression method (see the purple RD curves for reference). The compressed noisy image with information loss makes the successive denoiser harder to reconstruct a pleasing image. Hence, in our remaining discussions, the sequential method specifically refers to the one that does denoising and successive compression.

The orange RD curves show that the joint baseline method [54] cannot outperform the sequential one and have a more significant performance gap between our method due to the better design of our compressor to learn a noise-free representation compared to previous works.

Synthetic noise (individual). To further discuss the effects of different noise levels, Fig. 4 shows the RD curves at individual noise levels for the MSE

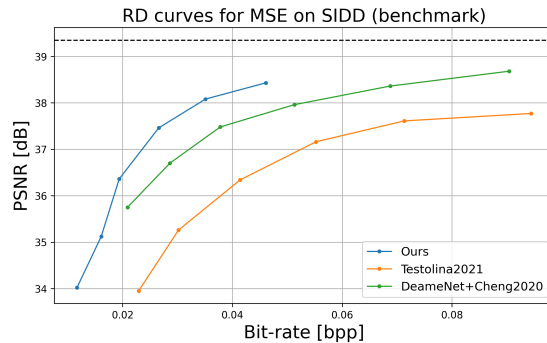


Fig. 5: RD curves optimized for MSE on the SIDD. Our method outperforms all the baseline solutions. The black dotted line is the DeamNet ideal case without compression for reference.

models on the Kodak dataset. We can see that our joint method is slightly better than the sequential method at the first three noise levels and significantly outperforms the sequential one at the highest noise level. Not to mention that our method has a much lower inference time as detailed in Sec. 5.3.

It is interesting to know that the pure denoiser DeamNet (black dotted line) drops significantly down to around 24 PSNR at noise level 4, which is the direct cause of the degraded performance for the sequential method (green curve) in the fourth chart in Fig. 4. Recall that all the models are not trained on synthetic images at noise level 3 (Gain $\propto 4$) and 4 (Gain $\propto 8$), where the Gain $\propto 4$ noise is slightly higher while Gain $\propto 8$ noise is considerably higher than the noisiest level during training. This indicates that the performance of the sequential solutions is somehow limited by the capacity of individual modules and suffers from the accumulation of errors. Our joint method has a beneficial generalization property to the unseen noise level to a certain extent.

Real-world noise. We also provide the RD curves optimized for MSE on the SIDD with real-world noise in Fig. 5. We plot DeamNet (black dotted line) as a pure denoising model to show an ideal case of denoising performance without compression (at 24 bpp) for reference. The results show that our proposed method works well not only on the synthetic dataset but also on the images with real-world noise.

It is worth mentioning that given the same compressor, the compressed bit lengths of different images vary, depending on the amount of information (entropy) inside the images. Here, we can see that all the evaluated RD points are positioned in the very low bpp range (< 0.1 bpp). The very low bit-rate SIDD results are consistent among all methods, indicating inherently low entropy in the test samples, where the official SIDD test patches of size 256×256 contain relatively simple patterns.

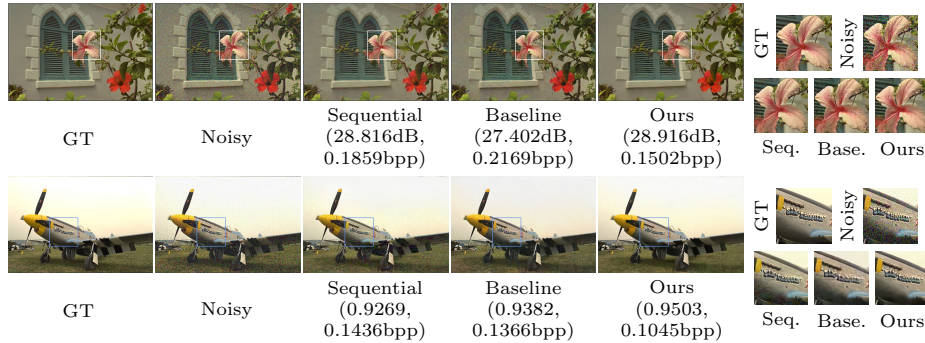


Fig. 6: Comparison results at noise level 4 (Gain $\times 8$) on Kodak image *kodim07* for MSE models and on Kodak image *kodim20* for MS-SSIM models. Apart from the better PSNR values and lower bpp rates, we can see that our solution has a better capacity to restore structural texture and edges.

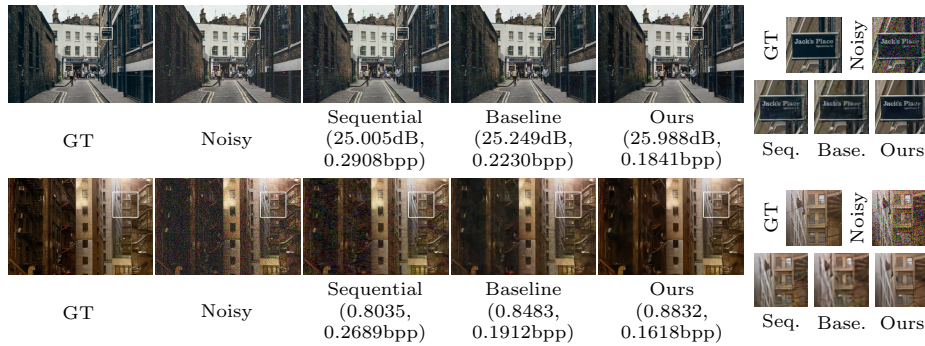


Fig. 7: Comparison results at noise level 4 (Gain $\times 8$) on sample CLIC images for both MSE and MS-SSIM models. Apart from the better PSNR values and lower bpp rates, we can observe that the text and edges are better restored for our method.

5.3 Efficiency Performance

We also compare the efficiency between our method and the sequential solution on the Kodak dataset, where the main difference comes from the compression process. The average elapsed encoding time under all qualities and noise levels for the sequential method is 75.323 seconds, while our joint solution is only 7.948 seconds. The elapsed running time is evaluated on Ubuntu using a single thread on Intel(R) Xeon(R) Gold 5118 CPU with 2.30GHz frequency. The sequential method has considerably longer running time than our joint method, where the additional overhead mainly comes from the heavy individual denoising modules in the encoding process of the sequential method. On the contrary, our joint

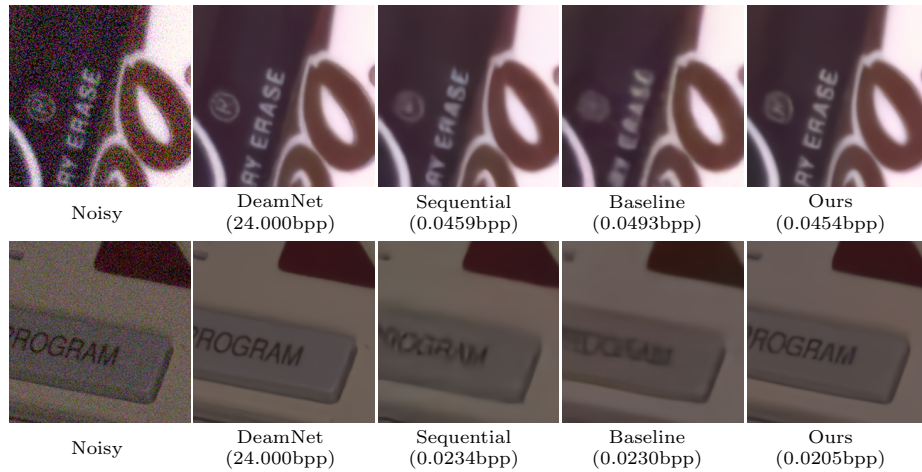


Fig. 8: Sample results on the SIDD. Since no ground-truth image is available for SIDD benchmark dataset, the visual results of DeamNet is shown as a reference for ground truth. We can see that the texts in our results is clearer at even slightly lower bpp rate.

formulation with efficient plug-in feature denoising modules, which pose little burden upon running time, is more attractive in real-world applications.

5.4 Qualitative Results

Some qualitative comparisons are presented to further demonstrate the effectiveness of our method. We show the visual results at noise level 4 ($\text{Gain} \propto 8$) of the sample Kodak images in Fig. 6 and CLIC images in Fig. 7 for both MSE and MS-SSIM models. Fig. 8 shows the results of two sample patches from the SIDD. These results show that our method can obtain better quality images with even lower bit rates. Please check our supplements for more visual results.

6 Conclusion

We propose to optimize image compression via joint learning with denoising, motivated by the observations that existing image compression methods suffer from allocating additional bits to store the undesired noise and thus have limited capacity to compress noisy images. We present a simple and efficient two-branch design with plug-in denoisers to explicitly eliminate noise during the compression process in feature space and learn a noise-free bit representation. Extensive experiments on both the synthetic and real-world data show that our approach outperforms all the baselines significantly in terms of visual and metrical results. We hope our work can inspire more interest from the community in optimizing the image compression algorithm via joint learning with denoising and other aspects.

References

1. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: Proceedings of CVPR (2018) [1](#), [3](#), [5](#), [9](#)
2. Al-Shaykh, O.K., Mersereau, R.M.: Lossy compression of noisy images. IEEE TIP **7**(12), 1641–1652 (1998) [1](#)
3. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimization of nonlinear transform codes for perceptual quality. In: Proceedings of PSC (2016) [1](#)
4. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. In: Proceedings of ICLR (2017) [4](#), [6](#), [7](#)
5. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior. In: Proceedings of ICLR (2018) [1](#), [4](#), [6](#)
6. Bégaint, J., Racapé, F., Feltman, S., Pushparaja, A.: Compressai: a pytorch library and evaluation platform for end-to-end compression research. arXiv:2011.03029 (2020) [9](#), [10](#)
7. Bellard, F.: Bpg image format (2015), <https://bellard.org/bpg/> [3](#)
8. Bishop, C.M.: Latent variable models. In: Learning in Graphical Models, vol. 89, pp. 371–403. Springer Netherlands (1998) [7](#)
9. Chambolle, A.: An algorithm for total variation minimization and applications. Journal of Mathematical Imaging and Vision **20**(1), 89–97 (2004) [3](#)
10. Chang, M., Li, Q., Feng, H., Xu, Z.: Spatial-adaptive network for single image denoising. In: Proceedings of ECCV (2020) [7](#)
11. Chen, C., Chen, Q., Xu, J., Koltun, V.: Learning to see in the dark. In: Proceedings of CVPR (2018) [3](#), [5](#)
12. Cheng, S., Wang, Y., Huang, H., Liu, D., Fan, H., Liu, S.: Nbnnet: Noise basis learning for image denoising with subspace projection. In: Proceedings of CVPR (2021) [7](#)
13. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. In: Proceedings of CVPR. pp. 7939–7948 (2020) [1](#), [4](#), [6](#), [7](#), [9](#)
14. Company, E.K.: Kodak lossless true color image suite (1999), <http://r0k.us/graphics/kodak/> [8](#)
15. Condat, L., Mosaddegh, S.: Joint demosaicking and denoising by total variation minimization. In: Proceedings of ICIP (2012) [4](#)
16. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space. In: Proceedings of ICIP (2007) [3](#)
17. Duda, J.: Asymmetric numeral systems. arXiv:0902.0271 (2009) [7](#)
18. Ehret, T., Davy, A., Arias, P., Facciolo, G.: Joint demosaicking and denoising by fine-tuning of bursts of raw images. In: Proceedings of ICCV. pp. 8868–8877 (2019) [4](#)
19. Farsiu, S., Elad, M., Milanfar, P.: Multiframe demosaicing and super-resolution from undersampled color images. In: Computational Imaging II (2004) [4](#)
20. Foi, A., Trimeche, M., Katkovnik, V., Egiazarian, K.O.: Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. IEEE TIP **17**(10), 1737–1754 (2008) [3](#)
21. Gharbi, M., Chaurasia, G., Paris, S., Durand, F.: Deep joint demosaicking and denoising. ACM TOG **35**(6), 191:1–191:12 (2016) [2](#), [4](#)
22. González, M., Preciozzi, J., Musé, P., Almansa, A.: Joint denoising and decomposition using cnn regularization. In: Proceedings of CVPR Workshops (2018) [2](#), [4](#)

23. Google: Web picture format (2010), <https://chromium.googlesource.com/webm/libwebp> 3
24. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: Proceedings of CVPR (2014) 3
25. Guan, H., Liu, L., Moran, S., Song, F., Slabaugh, G.G.: NODE: extreme low light raw image denoising using a noise decomposition network. arXiv:1909.05249 (2019) 3
26. Guo, S., Yan, Z., Zhang, K., Zuo, W., Zhang, L.: Toward convolutional blind denoising of real photographs. In: Proceedings of CVPR. pp. 1712–1722 (2019) 3
27. Guo, Z., Wu, Y., Feng, R., Zhang, Z., Chen, Z.: 3-d context entropy model for improved practical image compression. In: Proceedings of CVPR Workshops. pp. 116–117 (2020) 4
28. Healey, G., Kondepudy, R.: Radiometric CCD camera calibration and noise estimation. IEEE TPAMI **16**(3), 267–276 (1994) 5
29. Hu, Y., Yang, W., Liu, J.: Coarse-to-fine hyper-prior modeling for learned image compression. In: Proceedings of AAAI. pp. 11013–11020 (2020) 4
30. Johnston, N., Vincent, D., Minnen, D., Covell, M., Singh, S., Chinen, T., Hwang, S.J., Shor, J., Toderici, G.: Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In: Proceedings of CVPR (2018) 4
31. (JVET), J.V.E.T.: Vvc official test model vtm (2021), https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/master 3
32. Khashabi, D., Nowozin, S., Jancsary, J., Fitzgibbon, A.W.: Joint demosaicing and denoising via learned nonparametric random fields. IEEE TIP **23**(12), 4968–4981 (2014) 4
33. Kim, Y., Soh, J.W., Park, G.Y., Cho, N.I.: Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In: Proceedings of CVPR. pp. 3482–3492 (2020) 3
34. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of ICLR (2015) 9
35. Klatzer, T., Hammernik, K., Knobelreiter, P., Pock, T.: Learning joint demosaicing and denoising based on sequential energy minimization. In: Proceedings of ICCP (2016) 4
36. Lebrun, M., Colom, M., Morel, J.: The noise clinic: a blind image denoising algorithm. IPOL **5**, 1–54 (2015) 5
37. Lee, J., Cho, S., Beack, S.: Context-adaptive entropy model for end-to-end optimized image compression. In: Proceedings of ICLR (2019) 4
38. Lin, C., Yao, J., Chen, F., Wang, L.: A spatial rnn codec for end-to-end image compression. In: Proceedings of CVPR (2020) 4
39. Liu, J., Lu, G., Hu, Z., Xu, D.: A unified end-to-end framework for efficient deep image compression. arXiv:2002.03370 (2020) 8
40. Liu, L., Jia, X., Liu, J., Tian, Q.: Joint demosaicing and denoising with self guidance. In: Proceedings of CVPR. pp. 2237–2246 (2020) 2
41. Liu, Y., Qin, Z., Anwar, S., Ji, P., Kim, D., Caldwell, S., Gedeon, T.: Invertible denoising network: A light solution for real noise removal. In: Proceedings of CVPR. pp. 13365–13374 (2021) 3
42. Mentzer, F., Toderici, G., Tschannen, M., Agustsson, E.: High-fidelity generative image compression. In: Advances in NeurIPS (2020) 4
43. Mildenhall, B., Barron, J.T., Chen, J., Sharlet, D., Ng, R., Carroll, R.: Burst denoising with kernel prediction networks. In: Proceedings of CVPR (2018) 5, 8

44. Minnen, D., Ballé, J., Toderici, G.: Joint autoregressive and hierarchical priors for learned image compression. In: *Advances in NeurIPS*. pp. 10794–10803 (2018) [1](#), [4](#), [6](#), [7](#)
45. Minnen, D., Singh, S.: Channel-wise autoregressive entropy models for learned image compression. In: *Proceedings of ICIP* (2020) [4](#)
46. Norkin, A., Birkbeck, N.: Film grain synthesis for AV1 video codec. In: *Proceedings of DCC*. pp. 3–12 (2018) [4](#)
47. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: *Proceedings of CVPR* (2017) [3](#), [5](#)
48. Ponomarenko, N.N., Krivenko, S.S., Lukin, V.V., Egiazarian, K.O., Astola, J.: Lossy compression of noisy images based on visual quality: A comprehensive study. *EURASIP* **2010** (2010) [1](#)
49. Preciozzi, J., González, M., Almansa, A., Musé, P.: Joint denoising and decomposition: A patch-based bayesian approach. In: *Proceedings of ICIP* (2017) [2](#), [4](#)
50. Rabbani, M.: Jpeg2000: Image compression fundamentals, standards and practice. *Journal of Electronic Imaging* **11**(2), 286 (2002) [1](#), [3](#)
51. Ren, C., He, X., Wang, C., Zhao, Z.: Adaptive consistency prior based deep network for image denoising. In: *Proceedings of CVPR*. pp. 8596–8606 (2021) [9](#)
52. Rissanen, J., Langdon, G.G.: Universal modeling and coding. *IEEE TIT* **27**(1), 12–23 (1981) [7](#)
53. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60**(1-4), 259–268 (1992) [3](#)
54. Testolina, M., Upenik, E., Ebrahimi, T.: Towards image denoising in the latent space of learning-based compression. In: *Applications of Digital Image Processing XLIV*. vol. 11842, pp. 412–422 (2021) [2](#), [4](#), [10](#), [11](#)
55. Theis, L., Shi, W., Cunningham, A., Huszár, F.: Lossy image compression with compressive autoencoders. In: *Proceedings of ICLR* (2017) [4](#)
56. Toderici, G., O’Malley, S.M., Hwang, S.J., Vincent, D., Minnen, D., Baluja, S., Covell, M., Sukthakar, R.: Variable rate image compression with recurrent neural networks. In: *Proceedings of ICLR* (2016) [4](#)
57. Toderici, G., Theis, L., Ballé, J., Johnston, N., Shi, W., Agustsson, E., Rapaka, K., Mentzer, F., Sinno, Z., Norkin, A., Noury, E., Timofte, R.: Workshop and challenge on learned image compression (2021), <http://www.compression.cc> [8](#)
58. Toderici, G., Vincent, D., Johnston, N., Jin-Hwang, S., Minnen, D., Shor, J., Covell, M.: Full resolution image compression with recurrent neural networks. In: *Proceedings of CVPR* (2017) [4](#)
59. Vandewalle, P., Krichane, K., Alleysson, D., Süsstrunk, S.: Joint demosaicing and super-resolution imaging from a set of unregistered aliased images. In: *Digital Photography III* (2007) [4](#)
60. Wallace, G.K.: The jpeg still picture compression standard. *IEEE TCE* **38**(1), xviii–xxxiv (1992) [1](#), [3](#)
61. Wang, W., Chen, X., Yang, C., Li, X., Hu, X., Yue, T.: Enhancing low light videos by exploring high sensitivity camera noise. In: *Proceedings of ICCV* (2019) [3](#)
62. Wang, Y., Huang, H., Xu, Q., Liu, J., Liu, Y., Wang, J.: Practical deep raw image denoising on mobile devices. In: *Proceedings of ECCV* (2020) [3](#)
63. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: *Proceedings of ACSSC* (2003) [8](#), [9](#)
64. Wei, K., Fu, Y., Yang, J., Huang, H.: A physics-based noise formation model for extreme low-light raw denoising. In: *Proceedings of CVPR* (2020) [3](#)

65. Xie, Y., Cheng, K.L., Chen, Q.: Enhanced invertible encoding for learned image compression. In: Proceedings of ACM MM. pp. 162–170 (2021) [1](#), [4](#), [8](#)
66. Xing, W., Egiazarian, K.O.: End-to-end learning for joint image demosaicing, denoising and super-resolution. In: Proceedings of CVPR. pp. 3507–3516 (2021) [2](#), [4](#)
67. Xu, X., Ye, Y., Li, X.: Joint demosaicing and super-resolution (jdsr): Network design and perceptual optimization. IEEE TCI **6**, 968–980 (2020) [4](#)
68. Yue, Z., Zhao, Q., Zhang, L., Meng, D.: Dual adversarial network: Toward real-world noise removal and noise generation. In: Proceedings of ECCV. pp. 41–58 (2020) [3](#)
69. Zhang, K., Zuo, W., Zhang, L.: Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. IEEE TIP **27**(9), 4608–4622 (2018) [3](#)
70. Zhang, K., Zuo, W., Zhang, L.: Learning a single convolutional super-resolution network for multiple degradations. In: Proceedings of CVPR (2018) [4](#)