

Variational Image Restoration Network

Zongsheng Yue, Hongwei Yong, Qian Zhao, Lei Zhang, *Fellow, IEEE*, and Deyu Meng, *Member, IEEE*

Abstract—Deep neural networks (DNNs) have achieved significant success in image restoration tasks by directly learning a powerful non-linear mapping function from corrupted images to their latent clean ones. However, there still exist two major limitations for these deep learning (DL)-based methods. Firstly, the noises contained in the real corrupted images are very complex, usually neglected and largely under-estimated in most current methods. Secondly, existing DL methods are mostly trained on one pre-assumed degradation process for all of the training image pairs, such as the widely used bicubic downsampling assumption in the image super-resolution task. This always inevitably leads to poor generalization performance when the true degradation does not match with such subjectively assumed one. To address these issues, we propose a unified generative model for the image restoration tasks, which elaborately configures the degradation process from the latent clean image to the observed corrupted one. Specifically, different from most of current methods, the pixel-wisely non-i.i.d. Gaussian distribution, being with more flexibility, is adopted in our method to fit the complex real noise shapes. Furthermore, the method is built on the general image degradation process, making it capable of adapting diverse degradations under one single model. Besides, we design a variational inference algorithm to learn all parameters involved in the proposed model with explicit form of objective loss. Specifically, beyond traditional variational methodology, two DNNs are employed to parameterize the posteriori distributions, including a restoration network, to infer the distribution of the latent clean image, and a sigma network, to infer the distribution of the non-i.i.d. image noise. Extensive experiments demonstrate the superiority of the proposed method on three classical image restoration tasks, including image denoising, image super-resolution and JPEG image deblocking. And the source code is available at <https://github.com/zsyAOA/VIRNet>.

Index Terms—Image restoration, denoising, super-resolution, deblocking, generative model, variational inference, deep learning.

1 INTRODUCTION

IMAGE restoration (IR) is one of the hottest research topics in signal processing and computer vision fields, aiming at recovering the latent clean image z from its observed corrupted one y , i.e.,

$$y = Hz + n, \quad (1)$$

where H represents the degraded operator, and n represents the noise contained in y . Depending on different degradation configurations, Eq. (1) corresponds to different IR tasks. For example, the three classical IR tasks, image denoising, image deblurring and image super-resolution, can be obtained by setting H as an identity matrix, a blurring operator and a composition of blurring and down-sampling operators, respectively. The IR task is generally challenging mainly due to two reasons. Firstly, the degradation processes for real images are always diverse and complex. The operator H might be even with infinite dimension and varied from one image to another even for a single IR mission. Secondly, the noise n is generally accumulated from multiple sources (e.g., capturing instrument, data transmissions media and image quantization) [1], and also affected by the camera pipelines that depends on their own hardware settings. This makes this inverse problem even more ill-posed and hardly to be resolved.

Z. Yue, Q. Zhao and D. Meng (corresponding author) are with School of Mathematics and Statistics and Ministry of Education Key Lab of Intelligent Networks and Network Security, Xi'an Jiaotong University, Shaanxi, P.R.China. Email:zsyam, timmy.zhaqian@gmail.com; dymeng@gmail.xjtu.edu.cn.
H. Yong and L. Zhang are with Department of Computing, The Hong Kong Polytechnic University, Hong Kong. Email:cshyong, cslzhang@comp.polyu.edu.hk.

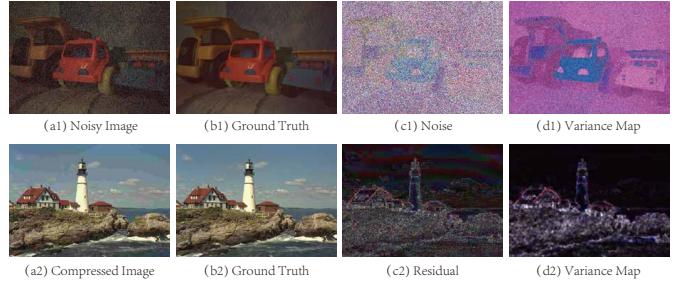


Fig. 1. Visualization of real noise or residual in image denoising (upper) or JPEG deblocking (lower) tasks. (a1)(a2): Real-world noisy or compressed images y . (b1)(b2): Corresponding noise-free or ground truth images x of (a1)(a2). (c1)(c2): Noise or residual demonstrations with values calculated by $|y - x|$. (d1)(d2): Variance maps predicted by the proposed method. For better visualization, the magnitudes of the noise and variance maps have been amplified.

In the past decades, various methods have been proposed for IR tasks under the maximum a posteriori (MAP) framework, generally involving one fidelity term and one prior term from the Bayesian perspective. To alleviate the serious ill-posed issue in IR tasks, most earlier works focused on elaborately designing the prior term based on intrinsic image structures to enforce the desired property of the recovered image. Typical presented priors include total variation (TV) [2], non-local similarity [3], [4], sparsity [4], [5], [6], [7], low-rankness [8], [9], [10] and so on. More recently, some works began to concentrate on the careful designing of the fidelity term by constructing more flexible distributions to fit image noise, typically including mixture of Gaussian (MoG) [11], mixture of Exponential (MoEP) [12] and Dirichlet Process mixture of Gaussian (DP-MoG) [13], [14], [15].

Albeit substantiated to be effective in some specific scenarios, these MAP-based methods still have evident defects in real IR tasks. Firstly, the rationality of a MAP model highly relies on the basis of the correctness of the manually designed image prior and noise distribution assumptions. However, human understandings on images are always subjective and not sufficiently comprehensive to faithfully and completely reflect the the real complex image knowledge, making the result deviating from the groundtruth. Secondly, most these methods are hardly to be used in practice for their relatively large computational burden, mainly attributed to their requirement of re-implementing the whole algorithm for any new testing image.

Different from the traditional MAP based methods, discriminative learning methods represent another research trend along this line. The aim of these methods is to obtain a well-trained prediction model, which can be readily used in the testing phase and avoids the iterative optimization procedures, like cascade of shrinkage fields (CSF) [16], trainable nonlinear reaction diffusions (TNRD) [17] and other related ones [18], [19], [20], [21]. In recent years, discriminative learning methods equipped with DNNs as their fundamental model forms have achieved unprecedent success in IR tasks, mainly owing to the powerful non-linear fitting capability of DNNs and tremendous advance in deep network training strategies. Specifically, DnCNN [22] method, simply composed of convolution layers, BatchNorm [23] layers and ReLU [24] activation layers, obtained competitive performance in several IR tasks, including image denoising, image super-resolution and JPEG deblocking. The success of DnCNN stimulated a wave of DL-based IR methods [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], which mainly focus on designing more effective network architectures.

Albeit achieving a huge boost in performance, most of these DL-based methods did not consider the image degradation process during modeling, while just forcibly fitting a mapping function from the observed corrupted image to the expected latent clean one based on the pre-collected image pairs, simulating the pseudo mapping input-output pairs, in a purely data-driven manner, naturally leading to two limitations of them. Firstly, the complex noise in general IR tasks, i.e., n in Eq. (1), is always not able to be well modeled. Two typical cases are shown in Fig. 1, corresponding to denoising and JPEG deblocking tasks on two real-world images, respectively. As can be evidently seen, the noises involved in two images are signal-dependent and spatially variant, implying that they are non-independently and non-identically distributed (non-i.i.d.) in statistics. Neglecting the fine modeling of such complex noise shapes always inclines to degenerate their generalization performance in real cases with complicated noises. Secondly, most of the DL-based methods are constructed based on one pre-assumed and fixed degraded operator H in Eq. (1). For example, H is usually set as bicubic downsample in the image super-resolution task [22], [27], [35], [36], [37]. With such a degraded operator H , they are trained by firstly generating large amount of image pairs and then use them to train the DNN model. Naturally, the learned model under this manner tends to deteriorate when the degradation setting on testing images does not well match that on the training data. In order to deal with general IR tasks in different

scenarios, it is thus critical to build a single IR model that can be finely generalized to diverse kinds of degradations.

To address the aforementioned issues, in this work we propose a general IR framework that is hopeful to integrate both advantages of traditional MAP-based methods and current DL-based methods. On one hand, it explicitly specifies the generation process from the latent clean image to the observed corrupted one, which not only comprehensively models the complex non-i.i.d. noise but also is able to deal with multiple degradations with one learned model. Such generative model also leads to better interpretability compared with current DL-based methods. On the other hand, DNNs are used to explicitly parameterize the inference process for the latent variables, which makes our model capable of being directly employed to get recovery image in the test stage, and thus more convenient and faster to be used than traditional MAP-based methods. In summary, the contributions of this work are as follows:

Firstly, we construct a Bayesian generative model for general IR tasks, finely representing the corruption process of the observed image from the Bayesian perspective. Specifically, we fit the noise term n in Eq. (1) with more flexible pixel-wisely non-i.i.d. Gaussian distributions, i.e., setting each image pixel with its own noise variance. This leads to a learnable likelihood purely relying on data. Besides, the degraded operator H is naturally encoded into our generative model as an additional regularizer for the recovered image. It should be noted that H in our model can be any general degraded operator and is not constrained to be the same for all training data. This makes our model capable of intrinsically exploring the underlying consistent prediction principle from diverse degradation operators on training data, as validated in our image super-resolution experiments.

Secondly, instead of only calculating the latent clean image, we regard both the noise variance and the clean image as latent variables, which are simultaneously inferred at the training stage in our method. In other words, the noise estimation and the IR tasks are intergraded into one unique Bayesian framework and thus can be mutually ameliorated during the inference process.

Thirdly, we design an amortized variational inference (VI) algorithm to guide the training of all variables involved in the proposed generative model, in which the posteriors of the latent clean image and the variance map are both explicitly parameterized by DNNs. Such an explicit mapping makes our model capable of being readily employed for directly achieving the posterior of any testing corrupted image, and thus can efficiently making inference for the IR task as compared with conventional MAP-based paradigm.

Fourthly, our proposed method exhibits superior performance in three classical IR tasks, including image denoising, image super-resolution and JPEG image deblocking, as compared with the state-of-the-art (SOTA) methods specifically designed for these tasks, demonstrating its potential in real IR applications. Especially, our method demonstrates specific advantages on its fine generalization capability in testing stage, like on those testing images with different noise distributions and degraded operators from all training ones.

The remainder of the paper is organized as follows: Section II introduces the related work. Section III proposes

our generative model, and discusses three specific IR tasks. Section IV presents the designed stochastic VI algorithm for solving our model. In Section V, comprehensive experiments are demonstrated to evaluate the performance of our method. Section VI concludes the paper.

2 RELATED WORK

In this section, we briefly review two major categories of relevant IR approaches to our work, including the traditional MAP-based approach and more recent DL-based approach.

2.1 MAP-Based Approach

As a long-standing research problem in computer vision, various methods have been proposed to address IR tasks during the past decades. Most of the early methods can be reformulated into the MAP framework, involving one likelihood (fidelity) term and one prior (regularization) term from the Bayesian perspective. The developments are also along two research lines accordingly:

Prior Modeling Methods: Aiming at alleviating the serious ill-posed issue of IR, many studies focused on exploiting rational image prior knowledge. The statistical regularities exhibited in images were firstly employed in IR task, including anisotropic diffusion [38], TV denoising [2] and wavelet coring [39]. Then NLM [3] and BM3D [4] were both proposed to remove the image noise using the non-local self-similarity prior, meaning that many small image patches in a non-local image area possess similar configurations. Later, low-rankness [8], [9], [10] and sparsity [4], [5], [6], [7] priors, which also aim to explore the characteristics of image patches, became popular and were widely used in IR tasks. To further increase the model's capacity and expressiveness, some other methods moved from the analytical technologies to data-driven approaches. E.g., Roth and Black [40] proposed the fields of experts (FoE) to learn image priors. Barbu [19] trained a discriminative model for the Markov random field (MRF) prior, while Sun and Tappen [20] proposed a non-local range MRF (NLR-MRF) model. Besides, Vemulapalli *et al.* [41] adopted the Gaussian conditional random field to derive an end-to-end trainable network. In addition, Qiao *et al.* [42] incorporated the non-local self-similarity prior into the variational model to obtain a proximal gradient based inference method.

Noise Modeling Methods: Different from the prior based methods, noise modeling methods concentrated on the likelihood (fidelity) term of MAP framework. In fact, the widely used L_1 and L_2 loss functions implicitly represent the i.i.d. Gaussian and Laplacian assumptions on image noise, respectively, which largely under-estimate the complexity of real noise. Based on such understanding, Meng *et al.* [11] proposed MoG noise modeling method under the low-rankness framework due to the universal approximated capacity of MoG for any continuous distribution. Furthermore, Zhu *et al.* [13], [43] and Yue *et al.* [14], [15] both introduced the non-parametric Dirichlet Process into MoG to increase its flexibility, leading to the adaptive adjustment for the component number of MoG.

Recently, the generative adversarial network (GAN) [44] provides a powerful tool to approximate one complex distribution with the aid of DNN. Inspired by GAN, some

works [37], [45], [46], [47] attempted to fit the noises contained in real images using DNN based on the adversarial training mechanism in IR tasks. Typically, Kim *et al.* [48] trained a noise generator conditioned on some camera settings, including ISO level, shutter speed and smartphone code. More recently, Yue *et al.* [49] proposed a dual adversarial loss to implement the noise removal and noise generation tasks in one unique Bayesian framework.

2.2 DL-Based Approach

Different from the traditional model-driven MAP methods, DL approach represents a data-driven trend to solve the IR problem. The main idea is to straightforwardly train a explicit mapping function parameterized by DNN on the pre-collected large amount of image pairs in an end-to-end manner. The earliest convolutional neural network (CNN) method can be traced back to [50], in which a five-layer network was employed. Then, some auto-encoder based methods were proposed [51], [52]. However, due to insufficient research in DNN, these methods are always inferior to the MAP-based methods in performance.

The first significant improvement of DNN on IR tasks was achieved by Burger *et al.* [53], which obtained comparable performance with BM3D in image denoising using a plain multilayer perceptron (MLP). Benefitted from the development of deep CNN technologies, Zhang *et al.* [22] trained a deep CNN model named DnCNN and achieved SOTA performance in several IR tasks. Then, deep CNN was attracted more research attention and successfully used in almost all of the IR tasks, especially in image denoising [25], [26], [32], [34], [54], [55] and super-resolution [35], [36], [56], [57], [58], [59], [60].

Compared with the traditional MAP approach, DL-based approach largely improves the IR performance by increasing the model capacity through DNN. However, such straightforward data-driven regime easily overfits to the training images because they do not sufficiently consider the intrinsic degradation process and the noise characteristic underlying the observed corrupted images. Therefore, it is necessary to develop a more comprehensive method to address the issues existed in both approaches.

3 THE PROPOSED METHOD

In this section, we firstly construct a generative model for general IR tasks, and then discuss how to simplify and optimize the model to adapt some specific IR tasks, like image denoising, image super-resolution, and JPEG image deblocking.

Before presenting our model, we briefly introduce some necessary notations. Given training data set $\mathcal{D} = \{\mathbf{y}^{(j)}, \mathbf{x}^{(j)}, H^{(j)}\}_{j=1}^N$, where $\mathbf{y}^{(j)}$ and $H^{(j)}$ denote the corrupted image and the corresponding degraded operator, respectively. The superscript j on H indicates that each image pair has its own degraded configuration, called "multiple degradations" in some image super-resolution works [54], [60]. As for $\mathbf{x}^{(j)}$, it denotes the manually simulated "clean" image in real data set and true clean image in synthetic data set. Specifically, in real-world image denoising tasks, $\mathbf{x}^{(j)}$ is usually obtained by averaging many noisy ones taken

under the same camera conditions [61], [62]. Our aim is then to formulate a rational full Bayesian model based on the beneficial knowledge delivered by the training image pairs.

3.1 Generative Model on General IR Tasks

Let $\mathbf{y} = [y_1, y_2, \dots, y_d]^T$, $\mathbf{x} = [x_1, x_2, \dots, x_d]^T$, and H denote any image pair and the degraded operator in training set \mathcal{D} , where $d = hw$ is the dimension of the image, h and w represent the image height and width, respectively. According to Eq. (1), we assume that the corrupted image \mathbf{y} is generated by the following model:

$$y_i \sim \mathcal{N}(y_i | (Hz)_i, \sigma_i^2), \quad i = 1, 2, \dots, d, \quad (2)$$

where $\mathbf{z} \in \mathbb{R}^d$ denotes the latent clean image, $(Hz)_i$ denotes the i -th pixel value of $H\mathbf{z}$, and $\mathcal{N}(\cdot | \mu, \sigma^2)$ denotes the Gaussian distribution with mean μ and variance σ^2 . Different from the commonly-used i.i.d. Gaussian/Laplacian assumption as convolutional, we model the noise as pixel-wisely non-i.i.d. Gaussian distribution in Eq. (2), expecting to better fit the complex noise in real scenarios, such as the spatially variant and signal-dependent noises in real-world image denoising tasks.

Next, we introduce possibly faithful priors for latent variables \mathbf{z} and σ^2 . Obviously, the simulated “clean” image \mathbf{x} provides an approximate estimation to \mathbf{z} , and thus we embed \mathbf{x} into the following prior distribution for \mathbf{z} :

$$z_i \sim \mathcal{N}(z_i | x_i, \varepsilon_0^2), \quad i = 1, 2, \dots, d, \quad (3)$$

where ε_0 is a hyper-parameter that reflects the closeness between \mathbf{x} and \mathbf{z} . For synthetic data set, \mathbf{x} is indeed the true latent clean image \mathbf{z} , which can be easily represented by setting ε_0 as a small number close to 0. Under such setting, Eq. (3) degenerates to the Dirac distribution centered at \mathbf{x} .

As for the variance $\sigma^2 = \{\sigma_1^2, \sigma_2^2, \dots, \sigma_d^2\}$, we construct the following conjugate prior:

$$\sigma_i^2 \sim \text{IG}\left(\sigma_i^2 \middle| \frac{p^2}{2} - 1, \frac{p^2 \xi_i}{2}\right), \quad i = 1, 2, \dots, d, \quad (4)$$

where

$$\xi = \mathcal{G}((\hat{\mathbf{y}} - H\hat{\mathbf{x}})^2; p), \quad (5)$$

where $\text{IG}(\cdot | \alpha, \beta)$ is the inverse Gamma distribution with parameters α and β , $\mathcal{G}(\cdot; p)$ represents Gaussian or average filter with window size p , and $\hat{\mathbf{y}}, \hat{\mathbf{x}} \in \mathbb{R}^{h \times w}$ are the matrix (image) forms of $\mathbf{y}, \mathbf{x} \in \mathbb{R}^d$, respectively. Actually, ξ_i is an estimation for the variance σ_i^2 based on the pixels in the $p \times p$ window centered at the i -th pixel. Inspired by [63], Eq. (4) is constructed to guarantee that the mode of this inverse Gamma distribution is ξ_i exactly.

Combining Eqs. (2)-(5), a full Bayesian model for a general IR task can be obtained. The corresponding graphical model is depicted in Fig. 2. Then the goal turns to infer the posterior of latent variable \mathbf{z} and σ^2 from \mathbf{y} and H , i.e., $p(\mathbf{z}, \sigma^2 | \mathbf{y}, H)$.

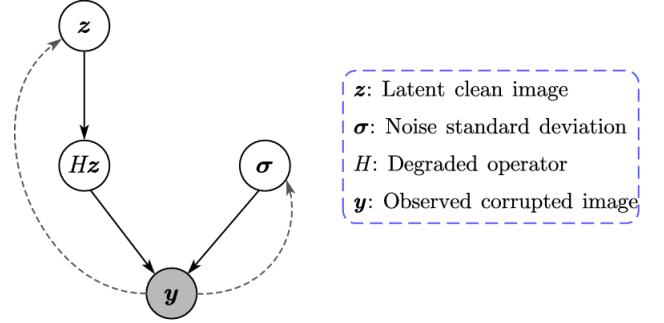


Fig. 2. Illustration of the graphical model of the proposed method. Black solid lines denote the generative process of the observed image, while gray dashed lines denote the inference process for the latent variables.

3.2 Discussions on Specific IR Tasks

Even though our proposed generative model provides a general formulation for IR tasks, different focused points based on their own characteristics. For example, image denoising takes more consideration on how to model the complex noise, while image super-resolution concentrates more on how to estimate the inverse mapping for various degradations. Here, we discuss how to simply adjust our model to make it better adapt some specific IR tasks, including image denoising, image super-resolution and JPEG image deblocking.

Image denoising: In real-world image denoising task, the degraded operator H degenerates into identity matrix, and thus Eq. (2) can be simplified as follows:

$$y_i \sim \mathcal{N}(y_i | z_i, \sigma_i^2), \quad i = 1, 2, \dots, d. \quad (6)$$

This simplification makes the objective function able to be analytically calculated during inference as shown in Sec. 4.2, and helps our model be trained more exactly and easily.

As is well known, the most challenging problem in this task is the signal-dependent property of the noise as shown in Fig. 1. In Eq. (4), we employ one filtering operation in a $p \times p$ window to obtain an estimation, i.e., ξ_i , for the noise variance σ_i^2 as prior. Due to the evident non-i.i.d. property of the noise conducted by its signal-dependence, we suggest to estimate the variance by a spatially weighted manner using Gaussian filter as listed in Eq. (5), mainly due to the fact that the closer two pixels are, the more similar distributions they are with. Besides, the window size p should not be too large, and we easily set it as 7 throughout all our image denoising experiments. More experimental comparisons can be seen in Sec. 5.4.1.

Image Super-resolution: Different from the denoising task, the noise involved in image super-resolution is always simply assumed as i.i.d. Gaussian [35], [58], [64]. Under such assumption, we can directly use the average filter in a relatively larger window to obtain the prior parameter ξ in Eq. (5). Empirically, we find that our model performs stably well when setting the window size p as 11 in all of the image super-resolution experiments.

In traditional non-blind image super-resolution research [54], the degraded operator H and the noise level σ are both required to be pre-known. However, in this work, we allow the noise level to be unknown latent knowledge

and automatically estimated by our model. We refer this as “partially non-blind image super-resolution”. Actually, the degraded operator H plays a role of regularization in our model, enforcing the recovered high-resolution (HR) image able to be mapped to the low-resolution (LR) image again through the corrupted process as defined in Eq. (1). More analysis can be seen in Sec. 4.4.

JPEG Image Deblocking: As shown in Fig. 1, the residual between the original and the compressed images is also signal-dependent in this situation, and thus can be regarded as one kind of special “noise”. Therefore, the JPEG image deblocking task is able to be reformulated into a special denoising problem and solved as discussed above.

4 STOCHASTIC VARIATIONAL INFERENCE

4.1 Form of Variational Posterior

We firstly construct a variational distribution $q(\mathbf{z}, \sigma^2 | \mathbf{y}, H)$ to approximate the true posterior $p(\mathbf{z}, \sigma^2 | \mathbf{y}, H)$ led by our generative model. As general VI techniques [65], we assume the conditional independence between latent variables \mathbf{z} and σ^2 , and factorize $q(\mathbf{z}, \sigma^2 | \mathbf{y}, H)$ as follows,

$$q(\mathbf{z}, \sigma^2 | \mathbf{y}, H) = q(\mathbf{z} | \mathbf{y}, H)q(\sigma^2 | \mathbf{y}). \quad (7)$$

Note that we simplify $q(\sigma^2 | \mathbf{y}, H)$ as $q(\sigma^2 | \mathbf{y})$ in Eq. (7) since the additive noise term \mathbf{n} does not rely on H .

Based on the conjugate prior in Eq. (4), it is natural to formulate the variational posterior form of σ^2 as the following inverse Gamma distribution:

$$q(\sigma^2 | \mathbf{y}) = \prod_i^d \text{IG}(\sigma_i^2 | \alpha_i(\mathbf{y}; W_S), \beta_i(\mathbf{y}; W_S)), \quad (8)$$

where $\alpha_i(\mathbf{y}; W_S)$ and $\beta_i(\mathbf{y}; W_S)$ are designed as mapping functions for getting the posteriori parameters of the latent variable σ^2 directly from the corrupted image \mathbf{y} . These two mapping functions are jointly parameterized as a neural network, called the sigma network or *SNet* as shown in Fig. 3, with learnable parameters W_S .

As for $q(\mathbf{z} | \mathbf{y}, H)$, we set it as Gaussian distribution, i.e.,

$$q(\mathbf{z} | \mathbf{y}, H) = \prod_i^d \mathcal{N}(z_i | \mu_i(\mathbf{y}, H; W_R), m_i^2(\mathbf{y}, H; W_R)). \quad (9)$$

In fact, such setting is inspired by the fact that Eq. (3) corresponds to the conjugate prior of \mathbf{z} when H degenerates into an identity matrix in some special case (e.g., in image denoising task), which leads to a Gaussian posteriori distribution. Similar to Eq. (8), $\mu_i(\mathbf{y}, H; W_R)$ and $m_i^2(\mathbf{y}, H; W_R)$ represent the mapping functions for evaluating posteriori parameters of \mathbf{z} from \mathbf{y} and H . And these two mapping functions are also jointly parameterized as a neural network, called the restoration network or *RNet* as shown in Fig. 3, with learnable parameters W_R .

With such parameterization using DNNs, it is able to get the explicit functions for predicting the latent clean image \mathbf{z} as well as the noise variance σ^2 for any testing image, avoiding the heavy iteration computations and subjective prior assumptions as traditional VI algorithms. Note that the network parameters W_R and W_S are shared and optimized across all training data, and thus our method is expected to induce a general statistical inference principle from the corrupted image to its latent clean image and noise variance.

4.2 Evidence Lower Bound

In order to train parameters W_R and W_S involved in *RNet* and *SNet*, we need to induce a rational objective function. For notation convenience, we simply denote $\alpha_i(\mathbf{y}; W_S)$, $\beta_i(\mathbf{y}; W_S)$, $\mu_i(\mathbf{y}, H; W_R)$ and $m_i(\mathbf{y}, H; W_R)$ as α_i , β_i , μ_i and m_i , respectively, in the following presentation. For any corrupted image \mathbf{y} and its corresponding H , its conditional log marginal probability can be decomposed as follows [65]:

$$\log p(\mathbf{y} | H) = \mathcal{L}(\mathbf{z}, \sigma; \mathbf{y}, H) + D_{KL}(q(\mathbf{z}, \sigma^2 | \mathbf{y}, H) \| p(\mathbf{z}, \sigma^2 | \mathbf{y}, H)), \quad (10)$$

where

$$\mathcal{L}(\mathbf{z}, \sigma; \mathbf{y}, H) = E_{q(\mathbf{z}, \sigma^2 | \mathbf{y}, H)} [\log p(\mathbf{y} | \mathbf{z}, \sigma^2, H)p(\mathbf{z})p(\sigma^2) - \log q(\mathbf{z}, \sigma^2 | \mathbf{y}, H)], \quad (11)$$

and $E_{p(x)}[f(x)]$ denotes the expectation of $f(x)$ w.r.t. random variable X with probability density function $p(x)$. The second term of Eq. (10) represents the KL divergence between the variational approximate posterior $q(\mathbf{z}, \sigma^2 | \mathbf{y}, H)$ and the ideal posterior $p(\mathbf{z}, \sigma^2 | \mathbf{y}, H)$. Due to the non-negative property of KL divergence, the first term $\mathcal{L}(\mathbf{z}, \sigma; \mathbf{y}, H)$ constitutes a lower bound of $\log p(\mathbf{y} | H)$, which is often called as evidence lower bound (ELBO), i.e.,

$$\log p(\mathbf{y} | H) \geq \mathcal{L}(\mathbf{z}, \sigma; \mathbf{y}, H). \quad (12)$$

Combining Eqs. (7)-(9) and (11), the ELBO can be rewritten as follows:

$$\mathcal{L}(\mathbf{z}, \sigma; \mathbf{y}, H) = E_{q(\mathbf{z}, \sigma^2 | \mathbf{y}, H)} [\log p(\mathbf{y} | \mathbf{z}, \sigma^2, H)] - D_{KL}(q(\mathbf{z} | \mathbf{y}, H) \| p(\mathbf{z})) - D_{KL}(q(\sigma^2 | \mathbf{y}) \| p(\sigma^2)). \quad (13)$$

And the KL divergences in the second and third terms both can be analytically calculated as follows:

$$D_{KL}(q(\mathbf{z} | \mathbf{y}, H) \| p(\mathbf{z})) = \sum_{i=1}^d \left\{ \frac{(\mu_i - x_i)^2}{2\varepsilon_0^2} + \frac{1}{2} \left[\frac{m_i^2}{\varepsilon_0^2} - \log \frac{m_i^2}{\varepsilon_0^2} - 1 \right] \right\}, \quad (14)$$

$$D_{KL}(q(\sigma^2 | \mathbf{y}) \| p(\sigma^2)) = \sum_{i=1}^d \left\{ [\log \Gamma(\alpha_i^0) - \log \Gamma(\alpha_i)] + \alpha_i \left(\frac{\beta_i^0}{\beta_i} - 1 \right) + (\alpha_i - \alpha_i^0) \varphi(\alpha_i) + \alpha_i^0 (\log \beta_i - \log \beta_i^0) \right\}, \quad (15)$$

where $\alpha_i^0 = \frac{p^2}{2} - 1$, $\beta_i^0 = \frac{p^2 \xi_i}{2}$, $\Gamma(\cdot)$ and $\varphi(\cdot)$ denotes the Gamma and Digamma function, respectively.

As for the first term of Eq. (13), it is intractable for general degraded operator H , such as blurring and downsampling in the image super-resolution task. However, we can use the reparameterization trick [66] to approximate it by Monte Carlo (MC) estimation, i.e.,

$$E_{q(\mathbf{z}, \sigma^2 | \mathbf{y}, H)} [\log p(\mathbf{y} | \mathbf{z}, \sigma^2, H)] = \sum_{i=1}^d \left\{ -\frac{1}{2} \log 2\pi - \frac{1}{2} [\log \beta_i - \varphi(\alpha_i)] - \frac{\alpha_i}{2\beta_i} [y_i - (H\tilde{\mathbf{z}})_i]^2 \right\}, \quad (16)$$

where

$$\tilde{\mathbf{z}} = \boldsymbol{\mu} + \mathbf{m} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d). \quad (17)$$

Note that we can use only one MC sample in Eq. (16) during the training process as suggested in [66]. Moreover,

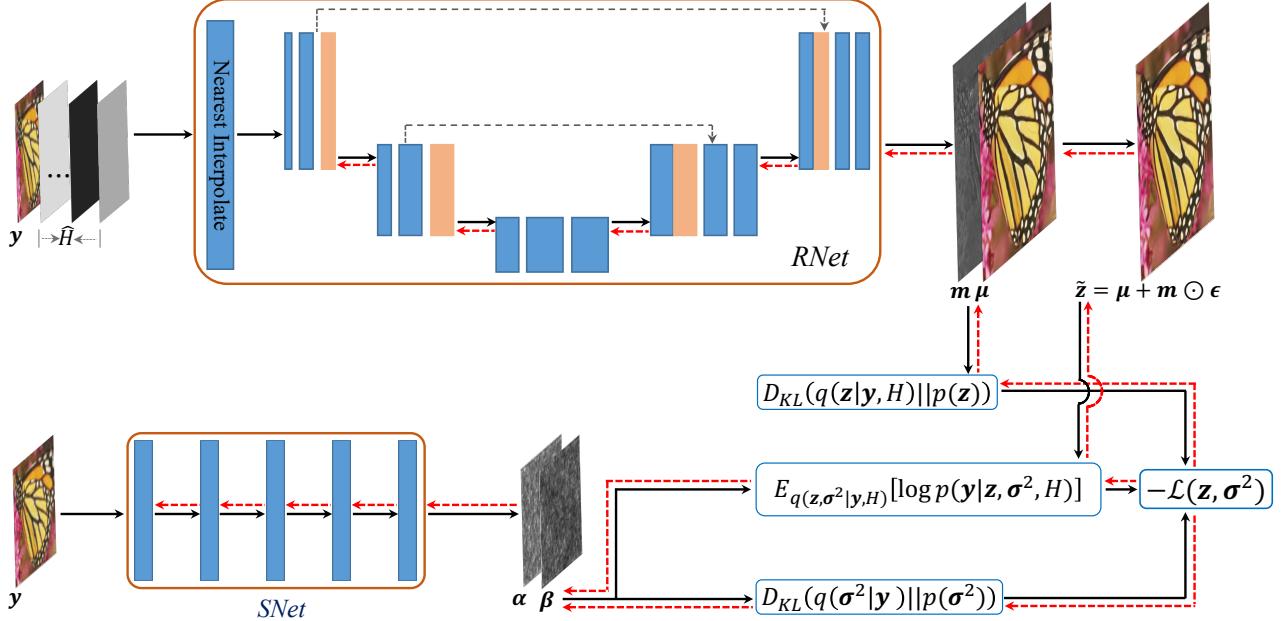


Fig. 3. The inference framework of the proposed generative model for image super-resolution task. The black solid lines denote the forward process, and the red dotted lines mark the gradient flow in the BP algorithm. For $RNet$, we concatenate y and H as input to infer the parameters of $q(z|y, H)$. Before concatenation, H is projected into lower space by PCA and then spatially stretched to the same size with y , i.e., \tilde{H} .

when H degenerates to identity matrix in some special task (e.g., image denoising), this term can also be analytically calculated instead of MC estimation, i.e.,

$$E_{q(z, \sigma^2 | y)} [\log p(y | z, \sigma^2)] = \sum_{i=1}^d \left\{ -\frac{1}{2} \log 2\pi - \frac{1}{2} [\log \beta_i - \varphi(\alpha_i)] - \frac{\alpha_i}{2\beta_i} [(y_i - \mu_i)^2 + m_i^2] \right\}. \quad (18)$$

Now we can easily get the expected objective function (i.e., the negative ELBO on the entire training set) for optimizing the network parameters of $RNet$ and $SNet$ as follows:

$$\min_{W_R, W_S} - \sum_{j=1}^N \mathcal{L}(z^{(j)}, \sigma^{(j)}; y^{(j)}, H^{(j)}), \quad (19)$$

where $z^{(j)}$ and $\sigma^{(j)}$ denotes the latent variables for the j -th image pair in training data set \mathcal{D} .

4.3 Network Structure and Learning

As aforementioned, $SNet$ is employed to infer the variational posterior parameters α and β from the corrupted image y as shown in Fig. 3. As for $RNet$, representing the inference of posterior $q(z|y, H)$, it thus takes the concatenation of the corrupted image y and the degraded operator H as input and outputs the posterior parameters μ and m . However, it is not feasible to directly concatenate them, because the dimensionality of H does not match with y . Against this problem, we adopt the dimensionality stretching strategy of [54], i.e., projecting H onto t -dimensional linear space by PCA and then stretching to the same spatial size with y .

In this work, we aim to propose a general framework that does not depend on specific network architectures for IR tasks. Therefore, we simply select the commonly-used networks in low-level vision tasks as our backbones

for $SNet$ and $RNet$. Inspired by [31], $SNet$ consists of five plain convolution layers, and each layer is followed with a LeakyReLU [67] activation except the last one. As for $RNet$, we adopt one tiny U-Net [68] in the synthetic image denoising, image super-resolution and JPEG image deblocking tasks, which contains 3 encoder blocks, 2 decoder blocks and symmetric skip connection under each scale. While in the real-world image denoising task, we add one more encoder and decoder blocks to increase the model capacity. More details about network architectures can be found in the supplementary material.

Using the stochastic gradient descend (SGD) or its any variant, it is easy to train our model based on the objective function presented in Eq. (19). Actually, each term of the ELBO can be intuitively explained: the last two KL divergence terms (Eqs. (14) and (15)) control the discrepancy between the variational posteriors and the priors, mainly used to update $SNet$ and $RNet$, respectively. The first term (Eq. (16) or (18)) corresponds to likelihood of the observed corrupted images in training data set. It couples $SNet$ and $DNet$ together and generates gradients to them simultaneously during backpropagation as marked by the red dotted lines in Fig. 3. In other words, $SNet$ and $DNet$ are guided and finetuned by each other under the supervision of such likelihood term.

At the testing stage, for any corrupted image y , through feeding it (with concatenation with H) into the trained $RNet$, the final recovered result can be directly obtained by μ . Additionally, through inputting the corrupted noisy image to $SNet$, the noise distribution knowledge (i.e., σ^2) is easily inferred. Specifically, the noise variance in each pixel can be directly obtained by using the mode of the inferred inverse Gamma distribution, i.e., $\sigma_i^2 = \frac{\beta_i}{\alpha_i + 1}$.

4.4 Remarks

Relationship to MSE: If we set the hyper-parameter ε_0^2 in Eq. (3) as an extremely small value close to 0, it is easy to see that the objective of the proposed method is dominated by the second term of Eq. (13), which makes the objective function degenerates as the MSE loss generally used in traditional DL methods. With more considerations on noise variation, our method can better generalize to more complex noise, and even performs well on unseen noise types in testing stages, as shown in our experiments of Sec. 5.1.1.

On the other hand, this also provides a new perspective to explain why the MSE loss inclines to overfit to noise bias in training data. Specifically, the posterior inference process is dominated by priors imposed on the latent clean image under MSE loss, while almost neglects the likelihood term (or data fidelity) of Eq. (16). In contrast, such likelihood term of Eq. (16) can be regarded as an additional regularization term for MSE loss in our method, which enforces the recovered image \tilde{z} being able to be mapped to the observed image y by the same degradation process.

Learnable re-weighted data fidelity: Most of the traditional IR methods assume that each element of the data fidelity term is with the same importance, i.e., $\sum_i [y_i - (H\tilde{z})_i]_p$. In this work, we novelly exploit an adaptive manner to re-weight the data fidelity in terms of l_2 -norm, i.e., $\sum_i \frac{\alpha_i}{\beta_i} [y_i - (H\tilde{z})_i]^2$ in Eq. (16). Each pixel is re-weighted by value $\frac{\alpha_i}{\beta_i}$, in which α_i and β_i are both learned by SNet. The weight $\frac{\alpha_i}{\beta_i}$ can be more clearly explained from statistics:

$$E \left[\frac{1}{\sigma_i^2} \right] = \frac{\alpha_i}{\beta_i}, \quad (20)$$

meaning that the weight is related to the inferred posterior of noise variance, i.e., $q(\sigma^2 | y)$.

Extension of VDN: This is an extended work for the image denoising method VDN [34] that is designed for the degenerated version of Eq. (1), i.e.,

$$y = z + n. \quad (21)$$

Compared with VDN, in this paper, we consider a more general corrupted process, i.e., general degraded operator H in Eq. (1), making our model capable of handling more IR tasks other than denoising. Besides, more optimization strategies are discussed as shown in Sec. 3.2 for different IR tasks. Such extension not only leads to a further capability amelioration than original VDN in real-world denoising tasks, but also substantially prompt the methodology developments of general IR tasks, e.g., image super-resolution and JPEG image deblocking, beyond conventional model-driven or data-driven manner. Our experiments comprehensively validate the advantages of such advancement, as shown in the next section.

5 EXPERIMENTAL RESULTS

In this section, we evaluate the effectiveness of our proposed method on three typical IR tasks, i.e., image denoising, image super-resolution, and JPEG image deblocking. We briefly denote our Variational Image Restoration Network as VIRNet in the following.

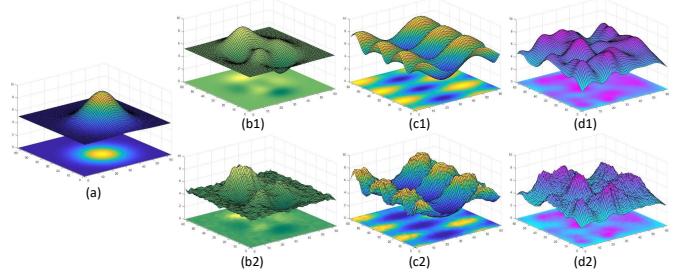


Fig. 4. (a) The spatially variant map M for noise generation in training data. (b1)-(d1): Three different M 's on testing data in Cases 1-3. (b2)-(d2): Predicted M 's by our method on testing data.

The weights of VIRNet were initialized according to [69]. We cropped small patches with size 128×128 for training images in denoising and JPEG deblocking tasks, and in super-resolution task we set the HR patch size as 96×96 . The Adam [70] algorithm with mini-batch size 32 is adopted to optimize the network parameters. The initial learning rate is set as 2×10^{-4} and gradually decays with a factor 0.5 until 3×10^{-6} after totally 8×10^5 iterations. We use an adaptive gradient clipping strategy to stabilize the training. Specifically, we recorded the gradient norm of the network parameters of each iteration in current epoch, and calculated their mean as threshold to clip gradients in next epoch.

5.1 Image Denoising Experiments

5.1.1 Synthetic Non-I.I.D. Gaussian Noise Removal

In order to verify the effectiveness and robustness of VIRNet under the non-i.i.d. noise configurations, we simulate clean-noisy image pairs to train our VIRNet as follows: 1) similar to [29], we collected a set of high quality source images as clean images, including 432 images from BSD500 [71], 400 images from the ImageNet [72] validation set and 4744 images from Waterloo Database [73]; 2) non-i.i.d. Gaussian noise is generated as follows:

$$n = n^1 \odot M, \quad n_{ij}^1 \sim \mathcal{N}(n_{ij}^1 | 0, 1), \quad (22)$$

where M is a spatially variant map with the same size as the source image, and the noisy image is obtained by adding the generated noise n to each source image. As for the testing images, three commonly-used ones in IR tasks are adopted to evaluate the performance of different methods, i.e., BSD68 [71], Kodak24 [74] and McMaster [75]. Note that we have totally generated four kinds of M 's as shown in Fig. 4. The first (Fig. 4 (a)) is used for generating noisy images in training data set, and the others (Fig. 4 (b1)-(d1)) for three groups of testing data (denoted as Case 1-3). Under such noise generation mechanism, the noise in training data and testing data are evidently different, which is suitable to verify the generalization capability of our VIRNet.

Comparison with the State-of-the-art: We compared the VIRNet with several Gaussian denoising methods, including three traditional methods, including CBM3D [4], NCSR [6] and WNNM [9], and five DL methods, including DnCNN [22], MemNet [26], FFDNet [29], UDNNet [76] and SGN [33]. The PSNR and SSIM results of all competing methods on three groups of testing data are listed in Table 1. From Table 1, we can easily see that: 1) on the whole,

TABLE 1

The PSNR and SSIM results of all competing methods on three groups of testing data sets. The best and second best results in each series of experiments are highlighted in bold and italic, respectively.

Cases	Datasets	Metrics	Methods									
			CBM3D	NCSR	WNNM	DnCNN	MemNet	UDNet	FFDNet	FFDNet _{Est}	SGN	VIRNet
Case 1	CBSD68	PSNR	26.73	24.97	25.14	28.74	28.29	27.15	28.79	28.78	28.89	29.02
		SSIM	0.766	0.666	0.682	0.819	0.801	0.766	0.818	0.818	0.819	0.828
	Kodak24	PSNR	27.35	25.87	26.09	29.69	29.20	27.88	29.87	29.86	30.02	30.16
		SSIM	0.769	0.693	0.701	0.818	0.801	0.763	0.823	0.823	0.825	0.833
	McMaster	PSNR	27.47	26.56	26.75	29.48	29.34	27.88	30.15	30.13	30.36	30.50
		SSIM	0.780	0.751	0.747	0.822	0.812	0.774	0.840	0.839	0.848	0.852
Case 2	CBSD68	PSNR	25.42	24.28	23.53	28.15	27.87	25.17	28.43	28.42	28.35	28.65
		SSIM	0.704	0.638	0.616	0.799	0.788	0.652	0.809	0.809	0.805	0.818
	Kodak24	PSNR	25.73	24.99	24.26	28.97	28.71	25.71	29.42	29.42	29.57	29.71
		SSIM	0.701	0.664	0.636	0.798	0.789	0.644	0.814	0.814	0.817	0.824
	McMaster	PSNR	25.82	25.68	24.68	28.84	28.87	25.88	29.74	29.72	29.91	30.08
		SSIM	0.712	0.719	0.674	0.800	0.801	0.661	0.832	0.831	0.840	0.844
Case 3	CBSD68	PSNR	26.85	24.95	24.59	28.11	27.70	26.45	28.22	28.21	28.26	28.46
		SSIM	0.736	0.651	0.634	0.794	0.776	0.715	0.798	0.797	0.797	0.808
	Kodak24	PSNR	27.65	25.96	25.53	29.00	28.56	27.11	29.23	29.23	29.39	29.53
		SSIM	0.742	0.683	0.655	0.795	0.778	0.707	0.803	0.803	0.807	0.814
	McMaster	PSNR	27.62	26.60	26.02	28.81	28.67	27.16	29.48	29.46	29.68	29.85
		SSIM	0.752	0.738	0.697	0.801	0.791	0.722	0.823	0.822	0.832	0.836

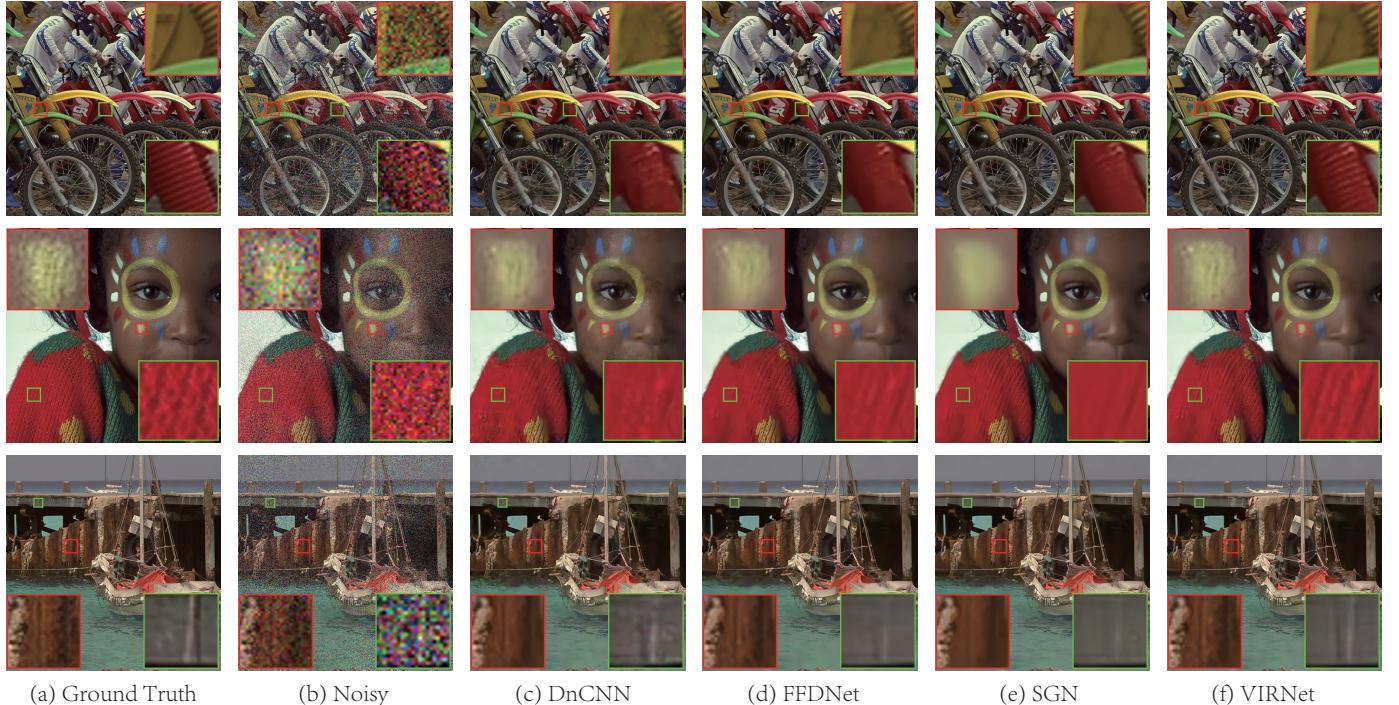


Fig. 5. Denoising results of three typical test examples in synthetic experiments. The first, second and third row correspond to cases 1, 2 and 3, respectively. From left to right: (a) Ground truth image, (b) noisy image, (c)-(f): denoising results by DnCNN, FFDNet, SGN and VIRNet, respectively.

the DL-based methods evidently outperform the traditional methods due to the powerful non-linear fitting capability of DNNs; 2) the proposed VIRNet outperforms other competing methods in all cases, indicating its superiority on handling such complex non-i.i.d. noise; 3) VIRNet and FFDNet both achieve better results than the benchmark DL method DnCNN because they make use of the noise information to guide the denoising task. However, VIRNet still surpasses FFDNet about 0.20dB PSNR even though FFDNet is guided by the manually provided true noise level while VIRNet infers the noise distribution automatically.

Fig. 5 shows visual results of different methods under testing cases 1-3 of Table 1. Note that we only display the best DL-based methods due to page limitation. It is seen

that the denoising results of DnCNN still contain certain amount of noises and SGN generates over-smooth and blurry recovery, especially in the heavy-noise areas. This can be explained by the fact that they do not consider the spatial noise variations. FFDNet and VIRNet adopt different strategies to handle such non-i.i.d. noise, and VIRNet preserves more image details (e.g., edges, structures) than FFDNet.

Even though our VIRNet is designed and trained on the non-i.i.d. noise case, it also performs well in additive white Gaussian noise (AWGN) removal tasks. Note that AWGN is one special case of the pixel-wisely non-i.i.d. Gaussian noise. Table 2 lists the average PSNR and SSIM results of different methods under three noise levels (i.e., $\sigma=15, 25, 50$) of AWGN. It is easy to see that our method obtains

TABLE 2
The PSNR and SSIM results of all competing methods under AWGN noise cases on three testing data sets. The best and second best results in each series of experiments are highlighted in bold and italic, respectively.

Cases	Simga	Metrics	Methods									
			CBM3D	NCSR	WNNM	DnCNN	MemNet	UDNet	FFDNet	FFDNet _{Est}	SGN	VIRNet
$\sigma=15$	CBSD68	PSNR	33.55	31.16	30.69	33.88	33.51	33.77	33.88	33.87	33.96	34.03
		SSIM	0.923	0.873	0.845	0.929	0.924	0.926	0.929	0.926	0.931	0.932
	Kodak24	PSNR	34.30	32.17	32.06	34.47	34.15	34.41	34.63	34.61	34.72	34.81
		SSIM	0.916	0.871	0.858	0.920	0.914	0.917	0.921	0.921	0.924	0.925
$\sigma=25$	McMaster	PSNR	34.03	33.20	33.34	33.44	33.89	33.94	34.65	34.61	34.66	34.83
		SSIM	0.911	0.898	0.896	0.903	0.908	0.909	0.921	0.921	0.923	0.924
	CBSD68	PSNR	30.82	28.59	28.54	31.23	30.84	31.04	31.22	31.21	31.39	31.42
		SSIM	0.870	0.800	0.788	0.883	0.870	0.876	0.882	0.881	0.887	0.887
$\sigma=50$	Kodak24	PSNR	31.75	29.70	29.87	32.02	31.62	31.84	32.13	32.11	32.34	32.38
		SSIM	0.868	0.807	0.806	0.876	0.862	0.869	0.878	0.877	0.884	0.884
	McMaster	PSNR	31.69	30.71	30.95	31.52	31.61	31.62	32.36	32.33	32.53	32.63
		SSIM	0.870	0.850	0.852	0.869	0.863	0.868	0.886	0.885	0.892	0.892
	CBSD68	PSNR	27.48	25.59	25.79	27.93	27.27	27.73	27.97	27.96	28.24	28.19
		SSIM	0.766	0.681	0.686	0.788	0.732	0.781	0.788	0.787	0.801	0.798
	Kodak24	PSNR	28.55	26.74	27.08	28.85	28.12	28.71	28.99	28.98	29.34	29.29
		SSIM	0.779	0.707	0.716	0.791	0.729	0.786	0.794	0.794	0.809	0.805
	McMaster	PSNR	28.55	27.43	27.80	28.63	28.17	28.51	29.19	29.18	29.58	29.55
		SSIM	0.791	0.763	0.769	0.799	0.744	0.791	0.815	0.814	0.832	0.829

TABLE 3

The PSNR and SSIM results of different methods on SIDD validation and testing data sets. The best results are highlighted in bold.

Methods	Testing		Validation	
	PSNR	SSIM	PSNR	SSIM
NLM [3]	26.75	0.699	26.30	0.505
CBM3D [4]	25.65	0.685	25.29	0.412
WNNM [9]	25.78	0.809	26.31	0.524
DnCNN [22]	23.66	0.583	38.56	0.910
CBDNet [31]	33.28	0.868	38.68	0.909
RIDNet [32]	-	-	38.71	0.913
VIRNet	39.28	0.955	39.31	0.917

the best 12 out of 18 cases) or second best (6 out of 18 cases) performance compared with the SOTA method SGN. Combining the results in Table 1 and Table 2, it should be rational to say that our VIRNet is robust and able to handle a wide range of noise types, due to its more flexible noise modeling essence.

5.1.2 Real-world Noise Removal

In this part, we evaluate the performance of VIRNet on real-world image denoising benchmark data sets, including DND [77] and SIDD [62]. DND¹ consists of 50 high-resolution images with realistic noise from 50 scenes taken by 4 consumer cameras, but it does not provide any other noisy-clean image pairs as training data. SIDD² is another real-world denoising benchmark, containing about 30,000 real noisy images captured by 5 cameras under 10 scenes. For each noisy image, it estimates one simulated “clean” image through some statistical methods [62]. 80% of them are provided to us for the purpose of training, and 40 images are selected from the remaining 20% as testing data set and validation data set, respectively. Note that the PSNR/SSIM results on the DND and SIDD testing data sets can only be obtained by online submission system, and the results on the SIDD validation data set can be calculated by ourself because the noise-free images are also available. Following the experimental setting of the SOTA real-world

image denoising method RIDNet [32], we trained our model using the noisy and noise-free image pairs from SIDD [62], Poly [78] and RENOIR [61] for fair comparison.

Results on Two Benchmarks: Table 3 lists the PSNR and SSIM results of different methods on SIDD testing and validation data sets³. Note that we only list the results reported in the published works that are available on the official benchmark website. It is obvious that VIRNet performs evidently better than other competing methods both on SIDD testing and validation data sets. Specifically, the performance of traditional methods (i.e., NLM, CBM3D and WNNM) all deteriorate seriously, since most of them are designed on i.i.d. Gaussian noise assumption, largely deviated from the real noise. As for DL-based methods, RIDNet achieves relatively satisfied denoising results by introducing the feature attention module. However, neither DnCNN nor CBDNet perform well on the testing data set, mainly because they were trained on other data sets, whose noise type is different from SIDD. For fair comparison, we retrained DnCNN and CBDNet on our training set and evaluated their performance on SIDD validation set. Under these same training settings, VIRNet still surpasses other DL-based methods significantly, indicating that our non-i.i.d. noise modeling manner is more effective when encountering real-world noise.

For easy visualization, we display two typical denoising results of SIDD validation data set in Fig. 6, from which we can observed that: 1) In summary, the proposed VIRNet achieves the best visual results compared with all the other methods; 2) The denoised images by WNNM still contain a lot of noise, while the DL-based methods are able to remove most of the noise; 3) In the first example (1st row of Fig. 6), the results of DnCNN, CBDNet and RIDNet are over-smoothed, while VIRNet better preserves the image details; 4) In the second example (2nd row of Fig. 6), VIRNet obviously recovers more structure details (e.g., stripes) than other methods.

3. We employed the function “compare_ssim” in scikit-image library to calculate the SSIM value on validatein data set, which is a little non-consistent with the SIDD official results on testing data set.

1. <https://noise.visinf.tu-darmstadt.de>

2. <https://www.eecs.yorku.ca/~kamel/sidd/benchmark.php>

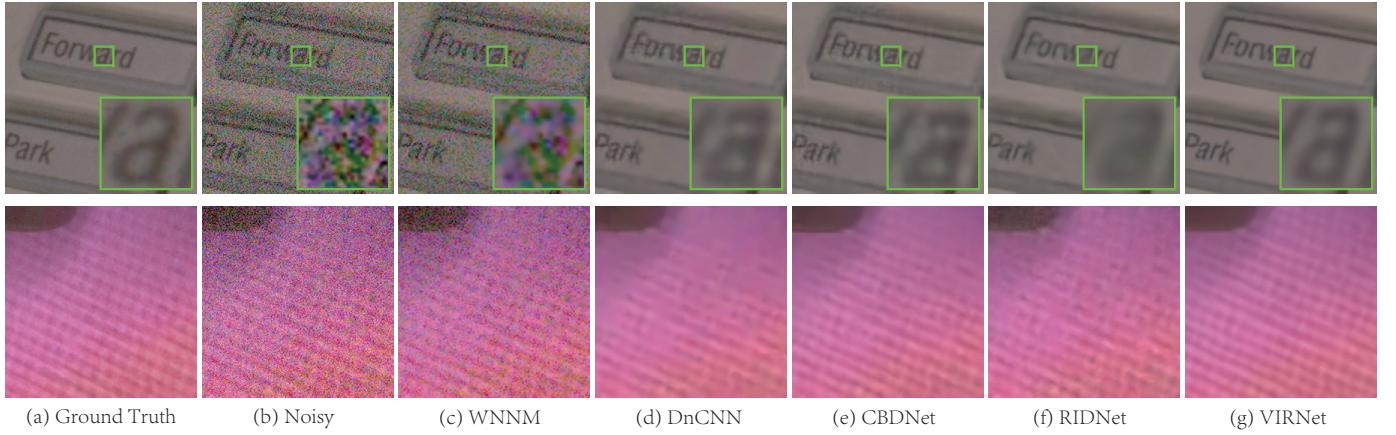


Fig. 6. Denoising results of two typical noisy images in SIDD validation data. From left to right: (a) ground truth image, (b) noisy image, (c)-(f): denoised images by WNNM, DnCNN, CBDNet, RIDNet and VIRNet, respectively.

TABLE 4

The PSNR and SSIM results of different methods on DND benchmark. The best results are highlighted as bold.

Metrics	Methods								
	CBM3D [4]	WNNM [9]	NCSR [6]	MLP [53]	DnCNN [22]	FFDNet [29]	CBDNet [31]	RIDNet [32]	VIRNet
PSNR	34.51	34.67	34.05	34.23	37.90	37.61	38.06	39.26	39.50
SSIM	0.8244	0.8646	0.8351	0.8331	0.9430	0.9415	0.9421	0.9528	0.9529

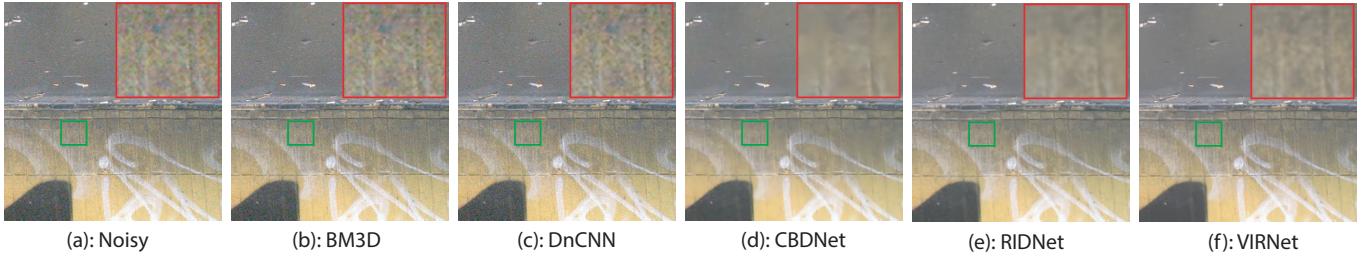


Fig. 7. One typical denoising example of different methods on DND benchmark. From left to right: (a) original noisy image, (b)-(f): denoised images by BM3D, DnCNN, CBDNet, RIDNet and VIRNet, respectively.

Table 4 lists the performance of all competing methods on DND benchmark. From the table, it is easy to see that the proposed VIRNet is superior than other competing methods both in terms of PSNR and SSIM, indicating that VIRNet is more feasible in real-world blind denoising tasks. Fig. 7 displays one typical visual result of different methods in DND benchmark, the better denoising quality of VIRNet is consistent with the quantitative results in Table 4.

5.2 Image Super-resolution

In this section, we apply our proposed VIRNet in the task of single image super-resolution. To train VIRNet, DIV2K [82] (containing 800 high quality images) are employed as the HR training data set. The LR images are synthesized by the following equation:

$$\mathbf{y} = (\mathbf{z} \otimes \mathbf{k}) \downarrow_s^d + \mathbf{n}, \quad (23)$$

where \mathbf{y} and \mathbf{z} denote the LR and HR image, respectively, \otimes represents 2-D convolution operation between \mathbf{z} and blur kernel \mathbf{k} , \downarrow_s^d denotes the direct down-sampler with scale factor s , i.e., keeping the upper-left pixel for each distinct $s \times s$ patch and discarding others, and \mathbf{n} is usually set as i.i.d. Gaussian noise with noise level σ . For blur kernel \mathbf{k} , we adopt isotropic Gaussian kernel determined by kernel width

d and anisotropic Gaussian kernel determined by covariance matrix Σ that is generated as follows [83]:

$$\mathbf{U} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad \mathbf{\Lambda} = \begin{bmatrix} l_1 & 0 \\ 0 & l_2 \end{bmatrix}, \quad \Sigma = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T. \quad (24)$$

For noise levels, we set the range to [0, 25] as suggested in [64]. And more settings about d , θ , l_1 , and l_2 on training data set can be seen in the supplementary material.

Two commonly-used data sets, i.e., Set5 [84] and Set14 [85], are adopted as testing data sets to quantitatively evaluate the performance of different methods. In order to conduct a thorough comparison on various degraded configurations, we consider 8 representative and diverse kernels to synthesize the LR images following Eq. (23), including 4 isotropic Gaussian kernels with different kernel widths (i.e., 0.7, 1.2, 1.6, and 2.0), and 4 anisotropic Gaussian kernels from [54]. For noise level, we set it as 0, 2.55 and 7.65 following [64]. Like most of the related literatures, the PSNR and SSIM [86] metrics are both calculated on Y channel (i.e., luminance) of transformed YCbCr space.

Comparing methods include four representative ones, including RCAN [27], ZSSR [83], IRCNN [87], and SRMD [54]. Specifically, RCAN is the current SOTA method for bicubic degradation in terms of PSNR metric. ZSSR is a zero-shot learning method to handle different degradations using the

TABLE 5

Average PSNR/SSIM results of comparing methods under different combinations of scale factors, blur kernels and noise levels on Set14. The best results are highlighted in bold. The results highlighted in gray color indicate unfair comparison due to mismatched degraded assumptions.

Methods	Scale Factor	Noise Level	Blur Kernel							
RCAN	x2	0	27.93/0.8426	27.62/0.8045	26.45/0.7459	25.52/0.6973	25.41/0.7013	25.08/0.6848	25.43/0.7001	24.76/0.6570
	x3	0	22.71/0.6612	24.76/0.7142	25.03/0.7034	24.68/0.6742	24.50/0.6698	24.02/0.6419	24.85/0.6913	24.24/0.6449
	x4	0	20.08/0.5403	21.73/0.5982	22.97/0.6274	23.47/0.6324	23.20/0.6226	22.52/0.5820	23.61/0.6499	23.51/0.6241
ZSSR	x2	0	29.93/0.8727	29.83/0.8665	29.21/0.8445	28.42/0.8130	27.90/0.7981	27.46/0.7800	27.74/0.7912	27.47/0.7723
	x3	0	26.73/0.7661	26.98/0.7702	26.89/0.7642	26.67/0.7507	26.45/0.7388	26.14/0.7264	26.30/0.7378	26.30/0.7303
	x3	2.55	26.68/0.7599	26.84/0.7592	26.66/0.7465	26.37/0.7268	26.20/0.7192	25.98/0.7099	26.09/0.7178	25.97/0.7011
	x3	7.65	26.36/0.7361	26.33/0.7239	26.09/0.7055	25.68/0.6824	25.59/0.6808	25.41/0.6718	25.53/0.6775	25.24/0.6583
	x4	0	25.30/0.6961	25.60/0.7023	25.69/0.7017	25.56/0.6949	25.38/0.6877	25.36/0.6834	25.38/0.6882	25.34/0.6841
IRCNN	x2	0	31.96/0.8935	30.08/0.8526	28.07/0.8018	26.98/0.7583	26.67/0.7611	26.42/0.7476	26.71/0.7526	25.87/0.7141
	x3	0	28.43/0.8047	28.61/0.7944	27.83/0.7630	26.68/0.7233	26.57/0.7243	26.28/0.7172	26.43/0.7199	25.74/0.6874
	x3	2.55	28.20/0.7942	28.16/0.7736	27.31/0.7368	26.33/0.6996	26.22/0.7017	25.99/0.6945	26.17/0.6970	25.53/0.6681
	x3	7.65	27.50/0.7611	27.12/0.7274	26.29/0.6911	25.54/0.6615	25.41/0.6621	25.26/0.6555	25.41/0.6586	24.88/0.6355
	x4	0	26.13/0.7243	26.84/0.7298	26.68/0.7130	26.13/0.6887	25.85/0.6851	25.81/0.6841	25.98/0.6868	25.39/0.6620
SRMD	x2	0	27.79/0.8435	28.10/0.8483	28.05/0.8366	27.71/0.8082	27.15/0.8107	24.42/0.7136	27.60/0.8142	27.34/0.7959
	x3	0	23.89/0.7012	24.33/0.7126	24.59/0.7152	24.53/0.7141	20.21/0.5046	18.71/0.4623	23.88/0.7137	23.08/0.6866
	x3	2.55	23.91/0.6989	24.38/0.7087	24.59/0.7065	24.51/0.6990	20.08/0.4927	20.58/0.5877	24.26/0.7041	23.87/0.6850
	x3	7.65	23.91/0.6910	24.32/0.6945	24.49/0.6864	24.36/0.6748	22.37/0.6310	22.61/0.6429	24.26/0.6776	24.02/0.6598
VIRNet	x2	0	32.79/0.9075	32.93/0.9082	32.09/0.8840	30.42/0.8281	30.21/0.8319	30.05/0.8405	30.41/0.8445	29.08/0.7818
	x3	0	29.21/0.8211	29.72/0.8281	29.73/0.8266	29.54/0.8194	29.27/0.8108	29.02/0.8068	29.12/0.8091	29.07/0.8020
	x3	2.55	29.10/0.8161	29.48/0.8172	29.26/0.8064	28.75/0.7862	28.46/0.7780	28.34/0.7774	28.49/0.7805	28.04/0.7578
	x3	7.65	28.61/0.7958	28.66/0.7847	28.18/0.7633	27.54/0.7366	27.39/0.7346	27.24/0.7299	27.35/0.7305	26.83/0.7068
	x4	0	27.18/0.7546	27.84/0.7650	28.01/0.7668	28.03/0.7652	27.87/0.7610	27.69/0.7571	27.71/0.7548	27.91/0.7594

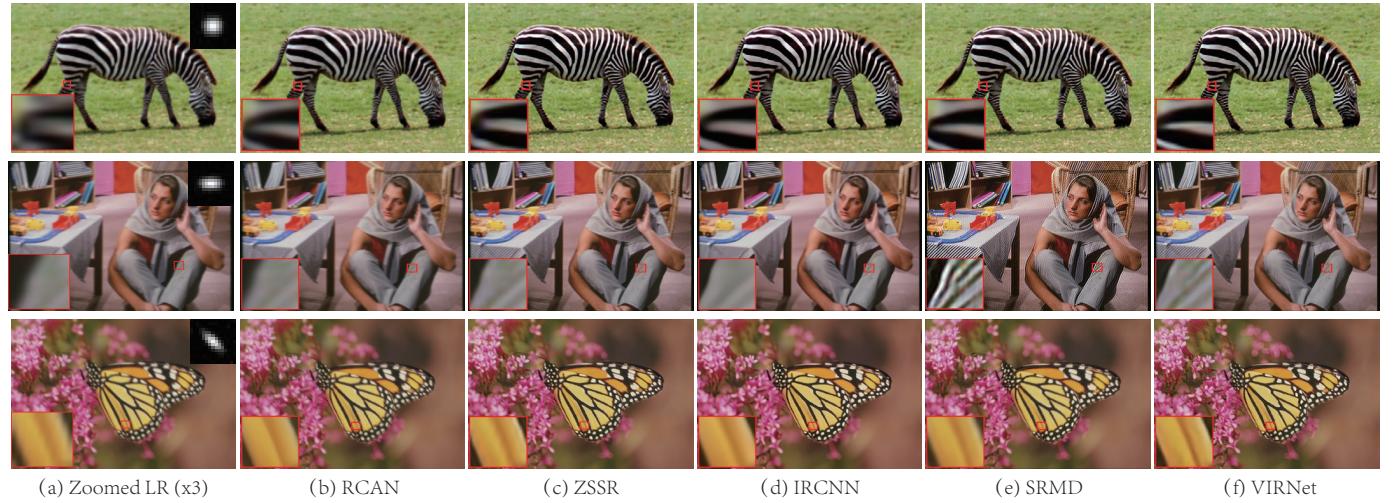


Fig. 8. Three typical visual results of different methods on super-resolving noise-free LR images in Set14 with scale factor 3. The blur kernel is shown on the upper right corner of the LR image.

internal recurrence of information inside a single image. IRCNN is a non-blind plug-and-play method which embeds a deep denoiser into the traditional HQS algorithm. SRMD attempts to learn a single model for multiple degradations by taking the kernel and noise level as input. Note that SRMD is assumed on the bicubic downsample, and we re-estimate the blur kernel under our adopted direct downsample by solving the following problem with a data-driven method as suggested by the authors,

$$\mathbf{k}_b = \arg \min_{\mathbf{k}_b} \|(\mathbf{z} \otimes \mathbf{k}_b) \downarrow_s^b - (\mathbf{z} \otimes \mathbf{k}_d) \downarrow_s^d\|_2^2, \forall \mathbf{z}, \quad (25)$$

where \downarrow_s^b is the bicubic downsample with scale factor s , \mathbf{k}_b and \mathbf{k}_d represent the corresponding blur kernels under bicubic and direct downsampled settings, respectively,

Table 5 lists the average PSNR/SSIM results of different methods on Set14, and more results on Set5 are put

into the supplementary material due to page limitation. From Table 5, it can be seen that our VIRNet obtains the best results under different scale factors, blur kernels and noise levels, since the degradation process is embedded into our inference framework to constrain the restorated image. RCAN deteriorates certainly when encountering different degradation settings other than the bicubic degradation that it adopted. ZSSR performs well for small scale factors (i.e., x2 and x3 cases) but cannot finely handle x4 case, because it only relies on the limited image knowledge in the single image. IRCNN obtains relatively good results under most of the cases, because the analytical solution for the data fidelity subproblem makes it generalize well to different degradation assumptions.

Fig. 8 shows three typical visual results of different methods on super-resolving noise-free LR images with scale

TABLE 6

The PSNR and SSIM results of different methods on LIVE1 and Classic5 data sets in JPEG image deblocking task with QF 10, 20, 30 and 40.

Datasets	QF	Methods															
		JPEG		SA-DCT [79]		ARCNN [80]		TNRD [17]		DnCNN [22]		DDFN [81]		RNAN [30]		VIRNet	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
LIVE1	10	27.77	0.7905	28.65	0.8093	28.98	0.8217	29.15	0.8111	29.19	0.8123	29.39	0.8136	29.63	0.8239	29.93	0.8347
	20	30.07	0.8683	30.81	0.8781	31.29	0.8871	31.46	0.8769	31.59	0.8802	31.76	0.8136	32.03	0.8239	32.29	0.8950
	30	31.41	0.9000	32.08	0.9078	32.69	0.9166	32.84	0.9059	32.98	0.9090	33.19	0.9117	33.45	0.9149	33.69	0.9202
	40	32.35	0.9173	32.99	0.9240	33.63	0.9306	-	-	33.96	0.9247	34.20	0.9273	34.47	0.9299	34.67	0.9341
Classic5	10	27.82	0.7800	28.88	0.8071	29.04	0.8111	29.28	0.7992	29.40	0.8026	29.55	0.8086	29.96	0.8178	31.00	0.8317
	20	30.12	0.8541	30.92	0.8663	31.16	0.8694	31.47	0.8576	31.63	0.8610	31.70	0.8636	32.11	0.8693	33.21	0.8819
	30	31.48	0.8844	32.14	0.8914	32.52	0.8967	32.78	0.8837	32.91	0.8861	33.03	0.8881	33.38	0.8924	34.44	0.9034
	40	32.43	0.9011	33.00	0.9055	33.34	0.9101	-	-	33.77	0.9003	33.90	0.9023	34.27	0.9061	35.28	0.9156



Fig. 9. Two typical JPEG deblocking examples with quality 10 (the 1st row) and 20 (the 2nd row). From left to right: (a) original image, (b) compressed image, (c)-(g): recovered results by TNRD, DnCNN, DDFN, RNAN and VIRNet, respectively.

factor 3 in Set14. It can be seen that VIRNet is able to better recover realistic and sharper HR images, and most of the other comparing methods obtain relatively blurry results and lose more image details. SRMD produces some artifacts in the “barbara” image (2nd row in Fig. 8) attributed to its not sufficiently accurate kernel re-estimations.

5.3 JPEG Image Deblocking

In this part, we test the effectiveness of our VIRNet on the JPEG image deblocking task. Following the settings in RNAN [30], 800 images in DIV2K [82] are used as training data set. During training, we randomly select a quality factor (QF) in range [5, 99] to generate the compressed images using OpenCV⁴, and the corresponding source images are regarded as ground truth. As for evaluation, two commonly-used data sets LIVE1 [88] and Classic5 [79] are employed as testing data sets, and we report the PSNR and SSIM [86] results under QF=10, 20, 30, 40, respectively. Note that the PSNR and SSIM metrics are calculated on the Y channel (in YCbCr space) as other methods for fair comparison.

Table 6 lists the PSNR/SSIM results of different methods on two testing data sets under different QF settings. As we can see, our VIRNet achieves the best performance in all

TABLE 7
Performance of VIRNet under different ε_0^2 values on different IR tasks.

Tasks	Metrics	ε_0^2					
		1e-4	1e-5	1e-6	1e-7	1e-8	MSE
Denoising	PSNR	39.04	39.29	39.31	39.30	39.23	39.18
	SSIM	0.9140	0.9167	0.9170	0.9167	0.9164	0.9155
Super-Resolution	PSNR	27.22	27.84	27.91	27.89	27.82	27.77
	SSIM	0.7350	0.7558	0.7593	0.7602	0.7565	0.7542
JPEG Deblocking	PSNR	33.19	33.21	33.21	33.19	33.15	33.12
	SSIM	0.8817	0.8819	0.8819	0.8816	0.8810	0.8804

cases, which indicates its effectiveness in this task. Furthermore, we show two typical visual comparisons under low image quality (QF=10, 20) in Fig. 9. It can be seen that the compared methods always over-smooth some areas when removing the blocking artifacts, while VIRNet obviously preserves more details and structures. That’s because the compared methods mostly adopt the i.i.d. Gaussian assumption (i.e., MSE loss) for the residual, which tends to result in blurry and averaged results. Comparatively, the re-weighted mechanism led by our non-i.i.d. noise modeling is able to recover more sharp and consistent images.

4. <https://opencv.org/>

TABLE 8

Performance of VIRNet under different p values on different IR tasks.

Tasks	Metrics	p					
		5	7	9	11	15	19
Denoising	PSNR	39.29	39.31	39.29	39.27	39.28	39.26
	SSIM	0.9166	0.9170	0.9168	0.9166	0.9167	0.9166
Super-Resolution	PSNR	27.87	27.88	27.90	27.91	27.91	27.91
	SSIM	0.7590	0.7592	0.7594	0.7594	0.7593	0.7594
JPEG Deblocking	PSNR	33.19	33.21	33.21	33.20	33.20	33.20
	SSIM	0.8819	0.8819	0.8819	0.8818	0.8818	0.8817

5.4 Additional Analysis

5.4.1 Hyper-parameters Analysis

Our method mainly involves two hyper-parameters, i.e., ε_0 in Eq. (3) and p in Eq. (5). To test the sensitiveness of VIRNet to both hyper-parameters, two series of experiments are conducted on different IR tasks. Specifically, we report the performance of VIRNet under different hyper-parameter settings on SIDD validation set for real-world image denoising, on Classic5 data set under QF= 20 for JPEG image deblocking, and on Set14 with the last anisotropic Gaussian blur kernel of Table 5 and noise level 0 under downsampling scale 3 for image super-resolution, respectively.

As discussed in Sec. 4.4, the ELBO of Eq. (13) degenerates to MSE loss when ε_0 is set as an extremely small value. Thus we directly trained the *RNet* separately under MSE loss as baseline for explicit comparison. The performances of VIRNet under different ε_0^2 values on three different IR tasks are listed in Table 7. From the table, we can see that: 1) when ε_0 is too large, the proposed VIRNet obtains relatively worse results since the prior constraint on z becomes unsubstancial; 2) VIRNet achieves the best results when ε_0^2 is 1e-6 in most of the cases, and performs stably well in the range [1e-5, 1e-7]; 3) with ε_0^2 decreasing after 1e-7, the performance of VIRNet tends to become gradually worse; 4) the result of VIRNet surpasses MSE loss 0.13/0.14/0.09dB PSNR on three different tasks when ε_0^2 is 1e-6, which indicates the importance of noise modeling in our method. Therefore, we suggest to set ε_0^2 as 1e-6 in these three IR tasks.

In Eq. (4) and (5), we introduce a conjugate inverse Gamma distribution as prior for σ^2 . The mode of this inverse Gamma distribution ξ_i provides a rational approximate evaluation for σ_i^2 , which is a local estimation in a $p \times p$ window centered at the i -th pixel. We compared the performance of VIRNet under different p values on different tasks in Table 8. It is seen that: Firstly, VIRNet is not sensitive to this hyper-parameter and achieves relatively satisfied results for most of p values. That's because ξ_i only provides a prior to constrain *SNet*, and *SNet* can be further adatptively adjusted by the likelihood (i.e., the first term of Eq. (13)) during training. Secondly, as discussed in Sec. 3.2, VIRNet performs a little better for signal-dependent noise in the real-world image denoising and JPEG deblocking tasks, when p is set as a moderate value 7. However, for i.i.d. Gaussian noise in the image super-resolution task, we can obtain a small improvement of performance by setting p as a larger value. Therefore, we set p as 7 in the denoising and deblocking tasks and 11 in the super-resolution task.

5.4.2 Variance Map Prediction

Different from other IR methods, the pixel-wisely non-i.i.d. Gaussian assumption is adopted to fit the noise distribution in our method. In the following, we analyse the performance of such assumption when encountering the common noise types in IR takes:

I.I.D. Gaussian Noise: Even though VIRNet is designed on the basis of non-i.i.d. Gaussian noise assumption, its superiority in the AWGN noise removal task of Table 2 in Sec. 5.1.1 and the image super-resolution task of Sec. 5.2 demonstrate it can be generalized well to such i.i.d. Gaussian noise.

To further quantitatively illustrate this point, we apply the estimated noise variance map by VIRNet in another non-blind i.i.d. Gaussian denoising method, i.e., FFDNet [29], which takes the pre-provided noise level as input. In Table 2, FFDNet denotes the results that take the true noise level as input while FFDNet_{Est} the results that take the predicted variance map by VIRNet as input. We can observe that FFDNet only surpasses FFDNet_{Est} 0.02db PSNR on average, even though it makes use of the true noise level, which indicates the accurate noise estimation of VIRNet on i.i.d. Gaussian noise. What's more, with the aid of VIRNet, FFDNet not only can be used in the case where the true noise level is unknown but also maintains comparable performance.

Non-I.I.D. Gaussian Noise: In Sec. 5.1.1, we choose three specific noise variance maps as shown in Fig. 4 (b1-d1) to synthesize non-i.i.d. Gaussian noise as testing data sets. For easy visulization, we display the predicted variance maps by VIRNet in Fig. 4 (b2-d2). It can be seen that the predicted variance map has very similar spatial variation with the groundtruth, which leads to the superiorities of VIRNet on such noise type. Similarly, the predicted variance maps are also provided as input of FFDNet to test its practical value. The PSNR and SSIM results of FFDNet and FFDNet_{Est} are listed in Table 1. It is obvious that FFDNet and FFDNet_{Est} almost have the same performance, indicating that VIRNet effectively captures proper noise information.

Signal-dependent Noise: One of the main challenges in IR tasks, such as the real-world image denoising and JPEG image deblocking, is the signal-dependence of real-world noise. Fig. 1 shows two typical examples of such noise type and the corresponding variance maps estimated by VIRNet. Note that the variance maps had been enlarged several times for easy visualization. It is easy to see that the predicted noise variance map depicts strong relevance to the pixel illumination, implying that our method is able to approximate the signal-dependent real noise to some extent.

6 CONCLUSION

In this paper, we have proposed a generative model to handle the IR tasks. In our model, we carefully take the image degradation process into consideration. On one hand, we novelly adopt one pixel-wisely non-i.i.d. Gaussian distribution to fit the complex noise contained in corrupted images, which is more powerful and flexible than the commonly-used i.i.d. Gaussian or Laplacian distribution. On the other hand, our model does not require that all the training image pairs are with the same degradation process. In other words, our method is able to handle multiple degradations

only with one single model. As for the inference, we have explicitly parameterized the posterior distributions of latent variables using DNNs under the amortized VI framework, which makes it possible to learn more abundant image knowledge from large training image pairs comparing with traditional VI algorithm. Extensive experimental results demonstrate the superiority of our method on typical IR tasks, including image denoising, JPEG image deblocking, and image super-resolution. In future, we will make further effort on extending our method to deal with more blind IR tasks under such variational framework, such as blind image super-resolution and image deblurring.

REFERENCES

- [1] Y. Tsin, V. Ramesh, and T. Kanade, "Statistical calibration of ccd imaging process," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 1. IEEE, 2001, pp. 480–487.
- [2] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.
- [3] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE, 2005, pp. 60–65.
- [4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [5] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on image processing*, vol. 17, no. 1, pp. 53–69, 2007.
- [6] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, 2012.
- [7] J. Xu, L. Zhang, and D. Zhang, "A trilateral weighted sparse coding scheme for real-world image denoising," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 20–36.
- [8] W. Dong, G. Shi, and X. Li, "Nonlocal image restoration with bilateral variance estimation: a low-rank approach," *IEEE transactions on image processing*, vol. 22, no. 2, pp. 700–711, 2012.
- [9] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2862–2869.
- [10] J. Xu, L. Zhang, D. Zhang, and X. Feng, "Multi-channel weighted nuclear norm minimization for real color image denoising," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1096–1104.
- [11] D. Meng and F. De La Torre, "Robust matrix factorization with unknown noise," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1337–1344.
- [12] X. Cao, Y. Chen, Q. Zhao, D. Meng, Y. Wang, D. Wang, and Z. Xu, "Low-rank matrix factorization under general mixture noise distributions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1493–1501.
- [13] F. Zhu, G. Chen, J. Hao, and P.-A. Heng, "Blind image denoising via dependent dirichlet process tree," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 8, pp. 1518–1531, 2016.
- [14] Z. Yue, D. Meng, Y. Sun, and Q. Zhao, "Hyperspectral image restoration under complex multi-band noises," *Remote Sensing*, vol. 10, no. 10, p. 1631, 2018.
- [15] Z. Yue, H. Yong, D. Meng, Q. Zhao, Y. Leung, and L. Zhang, "Robust multiview subspace learning with nonindependently and nonidentically distributed complex noise," *IEEE transactions on neural networks and learning systems*, 2019.
- [16] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2774–2781.
- [17] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1256–1272, 2016.
- [18] K. G. Samuel and M. F. Tappen, "Learning optimized map estimates in continuously-valued mrf models," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 477–484.
- [19] A. Barbu, "Learning real-time mrf inference for image denoising," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1574–1581.
- [20] J. Sun and M. F. Tappen, "Separable markov random field model and its applications in low level vision," *IEEE transactions on image processing*, vol. 22, no. 1, pp. 402–407, 2012.
- [21] U. Schmidt, J. Jancsary, S. Nowozin, S. Roth, and C. Rother, "Cascades of regression tree fields for image restoration," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 677–689, 2015.
- [22] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of The 32nd International Conference on Machine Learning*, 2015, pp. 448–456.
- [24] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.
- [25] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in neural information processing systems*, 2016, pp. 2802–2810.
- [26] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4539–4547.
- [27] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 294–310.
- [28] T. Plötz and S. Roth, "Neural nearest neighbors networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 1087–1098.
- [29] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018.
- [30] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," in *ICLR 2019 : 7th International Conference on Learning Representations*, 2019.
- [31] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1712–1722.
- [32] S. Anwar and N. Barnes, "Real image denoising with feature attention," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
- [33] S. Gu, Y. Li, L. V. Gool, and R. Timofte, "Self-guided network for fast image denoising," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2511–2520.
- [34] Z. Yue, H. Yong, Q. Zhao, D. Meng, and L. Zhang, "Variational denoising network: Toward blind noise modeling and removal," in *Advances in Neural Information Processing Systems*, 2019, pp. 1688–1699.
- [35] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*. Springer, 2014, pp. 184–199.
- [36] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1664–1673.
- [37] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.
- [38] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 7, pp. 629–639, 1990.
- [39] E. P. Simoncelli and E. H. Adelson, "Noise removal via bayesian wavelet coring," in *Proceedings of 3rd IEEE International Conference on Image Processing*, vol. 1. IEEE, 1996, pp. 379–382.
- [40] S. Roth and M. J. Black, "Fields of experts," *International Journal of Computer Vision*, vol. 82, no. 2, p. 205, 2009.

- [41] R. Vemulapalli, O. Tuzel, and M.-Y. Liu, "Deep gaussian conditional random field network: A model-based deep network for discriminative denoising," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4801–4809.
- [42] P. Qiao, Y. Dou, W. Feng, R. Li, and Y. Chen, "Learning non-local image diffusion for image denoising," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1847–1855.
- [43] F. Zhu, G. Chen, and P.-A. Heng, "From noise modeling to blind image denoising," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 420–429.
- [44] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [45] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3155–3164.
- [46] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.
- [47] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "Deblurgan: Blind motion deblurring using conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8183–8192.
- [48] D.-W. Kim, J. Ryun Chung, and S.-W. Jung, "Grdn: Grouped residual dense network for real image denoising and gan-based real-world noise modeling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.
- [49] Z. Yue, Q. Zhao, L. Zhang, and D. Meng, "Dual adversarial network: Toward real-world noise removal and noise generation," *arXiv preprint arXiv:2007.05946*, 2020.
- [50] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Advances in neural information processing systems*, 2009, pp. 769–776.
- [51] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Advances in neural information processing systems*, 2012, pp. 341–349.
- [52] F. Agostinelli, M. R. Anderson, and H. Lee, "Adaptive multi-column deep neural networks with application to robust image denoising," in *Advances in Neural Information Processing Systems*, 2013, pp. 1493–1501.
- [53] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with bm3d?" in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 2392–2399.
- [54] K. Zhang, W. Zuo, and L. Zhang, "Learning a single convolutional super-resolution network for multiple degradations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3262–3271.
- [55] V. Santhanam, V. I. Morariu, and L. S. Davis, "Generalized deep image to image regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5609–5619.
- [56] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.
- [57] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2472–2481.
- [58] K. Zhang, W. Zuo, and L. Zhang, "Deep plug-and-play super-resolution for arbitrary blur kernels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1671–1681.
- [59] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, and J. Sun, "Metasr: a magnification-arbitrary network for super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1575–1584.
- [60] J. Gu, H. Lu, W. Zuo, and C. Dong, "Blind super-resolution with iterative kernel correction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 1604–1613.
- [61] J. Anaya and A. Barbu, "Renoir - a benchmark dataset for real noise reduction evaluation," *arXiv: Computer Vision and Pattern Recognition*, 2014.
- [62] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1692–1700.
- [63] H. Yong, D. Meng, W. Zuo, and L. Zhang, "Robust online matrix factorization for dynamic background subtraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 7, pp. 1726–1740, 2018.
- [64] K. Zhang, L. V. Gool, and R. Timofte, "Deep unfolding network for image super-resolution," 2020.
- [65] C. M. Bishop, *Pattern Recognition and Machine Learning*, 2006.
- [66] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR 2014 : International Conference on Learning Representations (ICLR) 2014*, 2014.
- [67] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.
- [68] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [69] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," in *ICLR 2014 : International Conference on Learning Representations (ICLR) 2014*, 2014.
- [70] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *ICLR 2015 : International Conference on Learning Representations 2015*, 2015.
- [71] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2010.
- [72] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [73] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang, "Waterloo exploration database: New challenges for image quality assessment models," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, 2017.
- [74] R. Franzen, "Kodak lossless true color image suite," <http://r0k.us/graphics/kodak/>, 1999.
- [75] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *Journal of Electronic Imaging*, vol. 20, no. 2, p. 23016, 2011.
- [76] S. Lefkimiatis, "Universal denoising networks : A novel cnn architecture for image denoising," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3204–3213.
- [77] T. Plotz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2750–2759.
- [78] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang, "Real-world noisy image denoising: A new benchmark." *arXiv preprint arXiv:1804.02603*, 2018.
- [79] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1395–1411, 2007.
- [80] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 576–584.
- [81] C. Chen, Z. Xiong, X. Tian, and F. Wu, "Deep boosting for image denoising," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.
- [82] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, L. Zhang, B. Lim *et al.*, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.
- [83] A. Shocher, N. Cohen, and M. Irani, "Zero-shot super-resolution using deep internal learning," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3118–3126.
- [84] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *British Machine Vision Conference 2012*, 2012, pp. 1–10.

- [85] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proceedings of the 7th international conference on Curves and Surfaces*, 2010, pp. 711–730.
- [86] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [87] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3929–3938.
- [88] L. C. A. C. B. Hamid Rahim Sheikh, Zhou Wang, "Live image quality assessment database release 2," <http://www.live.ece.utexas.edu/research/quality/subjective.htm>, 2005.