

# Deep Unfolding Multi-scale Regularizer Network for Image Denoising

Jingzhao Xu<sup>1</sup>, Mengke Yuan<sup>2,3</sup>, Dong-Ming Yan<sup>2,3</sup>, Tieru Wu<sup>1</sup>

<sup>1</sup>School of Mathematics, Jilin University, Changchun, China

<sup>2</sup>NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China

<sup>3</sup>School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

xujz19@mails.jlu.edu.cn mengke.yuan@nlpr.ia.ac.cn yandongming@gmail.com wutr@jlu.edu.cn

## Abstract

Existing deep unfolding methods unroll an optimization algorithm with a fixed number of steps, and utilize convolutional neural networks (CNNs) to learn data-driven priors. However, their performance is limited for two main reasons. Firstly, priors learned in deep feature space need to be converted to the image space at each iteration step, which limits the depth of CNNs and prevents CNNs from exploiting contextual information. Secondly, existing methods only learn deep priors at the single full-resolution scale, so ignore the benefits of multi-scale context in dealing with high level noise. To address these issues, we explicitly consider the image denoising process in the deep feature space and propose the *deep unfolding multi-scale regularizer network* (DUMRN) for image denoising. The core of DUMRN is the *feature-based denoising module* (FDM) that directly removes noise in the deep feature space. In each FDM, we construct a multi-scale regularizer block to learn deep prior information from multi-resolution features. We build the DUMRN by stacking a sequence of FDMs and train it in an end-to-end manner. Experimental results on synthetic and real-world benchmarks demonstrate that DUMRN performs favorably compared to state-of-the-art methods.

## 1. Introduction

Image denoising is a fundamental problem in low-level vision since corruption by noise is inevitable during the image acquisition process. Image denoising aims to recover the latent clean image  $x$  from the corresponding noisy observation  $y$ . Mathematically, the degradation model for the denoising problem can be formulated as

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

where  $n$  is generally assumed to be *additive white Gaussian noise* (AWGN) with standard deviation  $\sigma$ . Due to the ill-posed nature of the image denoising problem, many conventional methods use various image prior terms based on

the statistics of natural images, including sparse models [18, 39, 17], non-local self-similarity models [7, 8, 13, 54, 55], low-rank models [16, 24] and Markov random field models [35, 48]. Despite their significant progress, these model-based methods usually reconstruct latent clean images by solving complicated optimization problems, which limits their practical application. Other methods [10, 16, 24] sacrifice flexibility and efficiency to achieve high performance.

With the development of deep *convolutional neural networks* (CNNs), many learning-based methods [61, 62, 9, 63, 41] have been proposed for image denoising. With the powerful representation ability of deep CNNs, these methods outperform traditional model-based methods by a large margin. However, most learning-based methods directly learn the mapping between noisy and clean image pairs without considering a physical model of the noise process, which makes them more efficient but less interpretable than model-based methods.

Taking advantages of model-based and learning-based methods, many deep unfolding methods incorporate standard optimization methods into deep CNNs. They unfold the image denoising problem through various optimization algorithms (*e.g.* gradient descent [11, 22], alternating direction method of multipliers [56], and primal-dual [2]), and implement the regularization term using deep CNNs, which can implicitly learn deep priors in the feature space. By integrating the image degradation constraint into CNNs, deep unfolding methods maintain the efficiency and improve the interpretability of deep learning. However, the performance of current deep unfolding methods is still limited for two main reasons. Firstly, in each iteration step of deep unfolding methods, noise is removed in the standard image space, but deep priors are learned in feature space. The transformation from deep feature space to image space limits the depth and receptive field of CNNs, which prevents CNNs from extracting non-local dependencies within images. Secondly, existing deep unfolding methods only learn deep priors at the original full-resolution, and thus cannot effectively capture spatial contextual information and restore clear edges for images that suffer from heavy noise [19].

In this paper, we propose the *deep unfolding multi-scale regularizer network* (DUMRN) to more closely integrate model-based and learning-based methods. To reduce the number of space transformations and improve information flow within the network, we explicitly consider the image denoising process in the deep feature space, and propose a *feature-based denoising module* (FDM) based on the image degradation model. By mimicking the gradient descent optimization process, a sequence of FDMs is stacked to build the DUMRN, so that we can obtain a deep CNN with a large receptive field and train it in an end-to-end manner. In each FDM, we construct a *multi-scale regularizer block* (MSRB) to learn deep prior information from multi-resolution features, which is able to capture local details at high resolution and large-scale contextual information at low resolution.

To summarize, the main contributions of this paper are three-fold:

- explicit consideration of the image denoising process in deep feature space, leading to a feature-based denoising module (FDM) based on the iterative optimization steps of the image degradation model, and a deep unfolding multi-scale regularizer network (DUMRN) produced by stacking a series of FDMs, which can effectively leverage deep features and learn deep prior information,
- a multi-scale regularizer block (MSRB) to learn deep prior information from features of different resolutions, which can capture fine-scale detail information as well as long-range contextual information, and
- comprehensive experiments to demonstrate DUMRN achieves competitive performance for synthetic and real-world image denoising; DUMRN is still effective and robust on the challenging blind Gaussian denoising task.

## 2. Related work

Image denoising methods can be roughly categorized as traditional model-based methods, deep learning-based methods, or deep unfolding methods. In this section, we briefly review these methods.

### 2.1. Traditional methods

Since image denoising is an ill-posed problem, various regularization or prior terms have been proposed to constrain the solution space. For example, Elad and Aharon [18] enforced sparsity on image patches by constructing highly over-complete dictionaries. Dabov *et al.* [13] proposed the well-known block-matching and 3D filter (BM3D) method to combine non-local self-similarity with sparsity for image denoising. Chen *et al.* [10] learned a Gaussian mixture model prior from external image patches

and utilized it to find similar patches in input noisy images for denoising. However, hand-crafted priors are not strong enough to characterize complex image structures and usually involve non-convex and time-consuming optimization problems.

### 2.2. Deep learning methods

Motivated by the great success of CNNs in high-level vision tasks, various learning-based methods have been proposed for image denoising. Zhang *et al.* [61] proposed DnCNN which incorporates residual learning [27] and batch normalization [31] in a CNN to learn residuals between the noisy input and the corresponding clean image. To increase the flexibility of the network to deal with noise of different levels, Zhang *et al.* [62] utilized a noise level map as input and performed denoising in the down-sampled sub-image space. Inspired by DenseNet [28], Zhang *et al.* [63] utilized dense connections to construct a residual dense network (RDN) for image restoration and achieved state-of-the-art results. For real-world image denoising, Guo *et al.* [25] proposed the convolutional blind denoising network (CBDNet) with a noise estimation subnetwork, while Kim *et al.* [33] constructed the adaptive instance normalization network (AINDNet) to transfer the Gaussian denoiser to various real noisy scenes. To decrease the computational cost on low-noise images, Yu *et al.* [57] proposed a multi-path CNN with a dynamic path selection module to adaptively select appropriate routes for different image regions. Recently, several methods [23, 9, 59, 58] have employed multi-scale strategies to enlarge the receptive field and improve the performance of deep networks. For example, Chang *et al.* [9] incorporated multi-size dilated convolutions into a U-Net [47] structure to capture multi-scale contextual information, which helps to restore rich details in complex scenes.

In order to increase the modeling capacity of deep priors, we utilize deep CNNs to learn deep prior information at different scales, which helps to enlarge the receptive field and capture long-range contextual information.

### 2.3. Deep unfolding methods

Deep unfolding methods integrate the advantages of model-based methods (*e.g.* good interpretability) and learning-based methods (*e.g.* efficiency and strong representation capability). They usually unfold iterative optimization algorithms as a cascade scheme with a fixed number of steps. Deep CNNs are utilized as regularizers in each step, which implicitly learn deep image priors. Many unfolding methods have been proposed for various image restoration tasks, including image denoising [6, 51, 50], image deblurring [49, 11, 22], and image super-resolution [60]. Schmidt and Roth [49] unrolled the half-quadratic optimization procedure into an end-to-end

learning framework and proposed a random field-based architecture to learn an image restoration regularizer. Chen *et al.* [11] proposed a trainable nonlinear reaction diffusion (TNRD) model by unfolding the gradient descent procedure to a fixed number of iterations. To integrate sparsity regularization with deep CNNs, Simon *et al.* [50] unfolded the convolutional sparse coding model by the learned iterative shrinkage threshold algorithm. By incorporating deep CNNs into a fully parameterized gradient descent scheme, Gong *et al.* [22] proposed to learn a universal gradient descent optimizer and construct a recurrent gradient descent network (RGDN) for image restoration. Recently, Ren *et al.* [46] incorporated the adaptive consistency prior into the maximum a posterior framework, and proposed DeamNet based on unfolding the optimization problem. Driven by a large training set, deep unfolding methods optimize the parameters in an end-to-end manner and surpass model-based methods.

Existing deep unfolding methods utilize CNNs to learn data-driven priors in deep feature space, but they remove noise in image space. Thus deep features are transformed into image space at each iteration step, which limits the depth of CNNs and makes it difficult from them to exploit contextual information within images. In order to make full use of deep CNNs, we explicitly consider the image denoising problem in deep feature space and construct a deep network for image denoising.

### 3. Proposed method

In this section, we first propose the feature-based denoising module (FDM) which performs image denoising in deep feature space. Then we introduce the multi-scale regularizer block (MSRB) which learns deep prior information from features at different resolutions. Finally, we stack several FDMs using an unfolding strategy to build the DUMRN for image denoising.

#### 3.1. Feature-based denoising module

From a Bayesian perspective, the maximum a posterior (MAP) model for denoising can be formulated as:

$$\hat{x} = \arg \min_x \frac{1}{2} \|y - x\|_2^2 + \lambda \phi(x), \quad (2)$$

where the first, fidelity, term guarantees the solution  $\hat{x}$  is in accordance with the degradation process in Eq. (1),  $\phi(x)$  is the regularization term associated with prior information, and  $\lambda$  is the regularization parameter that controls the trade-off between these terms. Eq. (2) can be solved by various optimization algorithms, such as gradient descent [11, 22], alternating direction of multipliers [56], and the primal-dual method [2]. Here we utilize the momentum gradient descent method due to its simplicity and effectiveness [45, 52].

Thus,  $\hat{x}$  can be obtained through the following iteration:

$$x^{t+1} = x^t - \alpha^t ((x^t - y) - \lambda \nabla \phi^t(x^t)) + \beta^t (x^t - x^{t-1}), \quad (3)$$

$$y = x^0, \quad (4)$$

where  $t$  denotes the step index,  $\alpha^t$  is the step size,  $\nabla \phi^t(\cdot)$  denotes the gradient of  $\phi(\cdot)$ ,  $\beta^t$  is the momentum parameter at step  $t$  and  $(x^t - x^{t-1})$  is the momentum term. In order to learn deep image priors, deep unfolding methods employ CNNs to calculate the gradient of the regularizer  $\nabla \phi(\cdot)$ , which implicitly provides a deep image prior. Then the iterative procedure in Eq. (3) is unrolled with a fixed number of steps  $T$ . However, in each iteration step, noise in  $x^t$  is removed in the standard image space while the prior  $\nabla \phi(\cdot)$  is learned in deep feature space. The mapping from deep feature space to image space limits the depth and receptive field size of CNNs, making it difficult for unfolding methods to capture non-local dependencies inside images.

In order to make full use of deep CNNs and improve the information flow in the network, we explicitly consider removing noise in deep feature space. Specifically, we first utilize a feature extractor  $f(\cdot)$  to map image  $x$  to the feature space. Then we can use Eq. (3) to iteratively denoise  $f(x)$  without mapping it into image space. In order to scale the gradient and momentum terms adaptively, we replace the step size  $\alpha^t$  and momentum parameter  $\beta^t$  by  $\mathcal{A}^t(\cdot)$  and  $\mathcal{B}^t(\cdot)$  respectively. We let  $\mathcal{S}(\cdot)$  replace  $\nabla \phi(\cdot)$  to supplant the gradient of the regularizer;  $\mathcal{S}(\cdot)$  absorbs the trade-off weight  $\lambda$ .  $\mathcal{S}(\cdot)$  implicitly performs the role of the deep prior. Overall, the  $t$ -th feature-based denoising module (FDM) is formulated as:

$$\mathbf{X}^{t+1} = \mathbf{X}^t - \mathcal{A}^t(\mathbf{X}^t - \mathbf{Y} + \mathcal{S}^t(\mathbf{X}^t)) + \mathcal{B}^t(\mathbf{X}^t - \mathbf{X}^{t-1}), \quad (5)$$

$$\mathbf{Y} = f(y) = f(x^0) = \mathbf{X}^0. \quad (6)$$

After  $T$  iterations, we utilize an image reconstructor  $g(\cdot)$  to reconstruct the final denoised image from  $\mathbf{X}^T$ :

$$\hat{x} = g(\mathbf{X}^T). \quad (7)$$

Considering the effectiveness of the residual block (ResBlock) in low-level vision tasks [36, 42], we use a single ResBlock to implement  $\mathcal{A}^t(\cdot)$  and  $\mathcal{B}^t(\cdot)$  respectively. The structure of the ResBlock is shown in Figure 1(a): each ResBlock contains two  $3 \times 3$  convolutional layers and a ReLU activation function [43]. For flexibility, we implement the feature extractor  $f$  and image reconstructor  $g$  using two convolutional layers with  $3 \times 3$  learnable kernels.

#### 3.2. Multi-scale regularizer block

Many unfolding methods [22, 60, 11] implement  $\mathcal{S}(\cdot)$  using deep CNNs due to their strong learning capability. However, limited by the size of the convolution kernel,

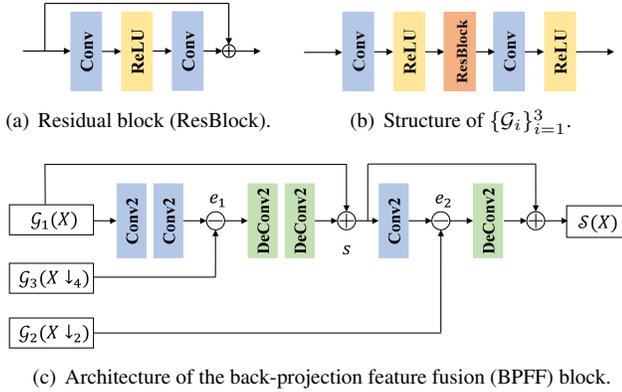


Figure 1. (a) Structure of the residual block (ResBlock). (b)  $\{\mathcal{G}_i(\cdot)\}_{i=1}^3$  in FDM share a common architecture of a CNN with different parameters to be learned. (c) Structure of the BPFF block. **Conv2** and **Deconv2** represent a convolutional layer and deconvolutional layer with stride of 2 respectively.

CNNs fail to capture diverse contextual information, and most existing deep unfolding methods (*e.g.* TNRD [11] and CSCNet [50]) perform poorly when the noise level is high. To overcome this problem, we adopt a multi-scale strategy by down-sampling features to different scales. On one hand, down-sampling can effectively enlarge the receptive field and enable models to exploit more spatial contextual information, which is helpful in denoising images that suffer from heavy noise [19]. On the other hand, as observed in [65], noise decreases while strong edges are less affected by down-sampling. To gain the advantages of a multi-scale strategy, we implement  $\mathcal{S}(\cdot)$  using a multi-scale regularizer block (MSRB) to extract useful feature information at multiple resolutions:

$$\mathcal{S}(\mathbf{X}) = f_{\text{MSRB}}(\mathbf{X}) = \mathcal{F}(\mathcal{G}_1(\mathbf{X}), \mathcal{G}_2(\mathbf{X} \downarrow_2), \mathcal{G}_3(\mathbf{X} \downarrow_4)), \quad (8)$$

where  $f_{\text{MSRB}}(\cdot)$  represents the proposed MSRB function,  $\downarrow_k$  represents the down-sampling operator with a scaling factor of  $k$ ,  $\{\mathcal{G}_i(\cdot)\}_{i=1}^3$  denotes a set of deep CNNs that learn useful prior information from the features of different scales, and  $\mathcal{F}(\cdot)$  denotes the multi-scale feature fusion block utilized to aggregate features from multiple scales. In practice, we use three CNNs with the same structure but different channel dimensions to implement  $\{\mathcal{G}_i(\cdot)\}_{i=1}^3$ . The numbers of channels are set to 64, 128, 256 at resolutions 1, 1/2, 1/4 respectively. As shown in Figure 1(b), each  $\mathcal{G}_i(\cdot)$  consists of two convolutional layers with ReLU activation functions and a ResBlock, and all kernel sizes are set to 3.

Summation and concatenation are the most commonly-used strategies for feature fusion, but directly applying them to multi-scale feature fusion is not effective [15]. Motivated by the success of back-projection technique in image super-resolution [32, 26, 14], we implement  $\mathcal{F}(\cdot)$  using a

back-projection feature fusion (BPFF) block [15] to aggregate multi-scale features. As shown in Figure 1(c), we first down-sample  $\mathcal{G}_1(\mathbf{X})$  to the same resolution as  $\mathcal{G}_3(\mathbf{X} \downarrow_4)$ , and compute their difference:

$$e_1 = \mathcal{G}_1(\mathbf{X}) \downarrow_4 - \mathcal{G}_3(\mathbf{X} \downarrow_4), \quad (9)$$

then we enhance the prior information  $\mathcal{G}_1(\mathbf{X})$  using the back-projected difference:

$$s = \mathcal{G}_1(\mathbf{X}) + (e_1) \uparrow_4. \quad (10)$$

We obtain the final  $\mathcal{S}(\mathbf{X})$  by applying a similar update procedure to integrate  $s$  and  $\mathcal{G}_3(\mathbf{X} \downarrow_4)$ :

$$e_2 = (s) \downarrow_2 - \mathcal{G}_2(\mathbf{X} \downarrow_2), \quad (11)$$

$$\mathcal{S}(\mathbf{X}) = s + (e_2) \uparrow_2. \quad (12)$$

In MSRB, we use convolutional layers and deconvolutional layers to implement down-sampling and up-sampling operators respectively; the strides of both convolutional layers and deconvolutional layers are set to 2. By utilizing the BPFF block in MSRB, we can effectively aggregate prior information at different resolutions into a stronger prior.

### 3.3. Overall architecture

The overall architecture of our proposed deep unfolding multi-scale regularizer network (DUMRN) is shown in Figure 2. Let  $y$  and  $x$  denote the input noisy image and the corresponding ground-truth image. The feature extractor  $f$  first extracts features from  $y$ . The extracted features  $\mathbf{Y}$  are set as the initial value  $\mathbf{X}^0$  for the feature denoising process.  $T$  FDMs are stacked to remove noise in feature space: we update  $\mathbf{X}^t$  via  $T$  steps using Eq. (5). Finally, we utilize the image reconstructor  $g$  to reconstruct the final image from the denoised feature  $\mathbf{X}^T$ .

The DUMRN is optimized by minimizing the difference between the input noisy images and the corresponding ground-truth counterparts. To assess the effectiveness of the proposed network, we adopt the same  $L_2$  loss function as previous works [61, 62, 9, 41]. Given a training dataset  $\{y_i, x_i\}_{i=1}^N$ , where  $N$  is the number of the training patch pairs, we obtain the optimal parameters by minimizing the following objective function:

$$\mathcal{L}(\Theta) = \frac{1}{N} \sum_{i=1}^N \|f_{\text{DUMRN}}(y_i) - x_i\|^2, \quad (13)$$

where  $f_{\text{DUMRN}}(\cdot)$  represents the DUMRN function, and  $\Theta$  represents all learnable parameters in DUMRN.

## 4. Experimental results

In this section, we first provide the training and implementation details of the proposed network. Then, we compare our proposed DUMRN with several state-of-the-art

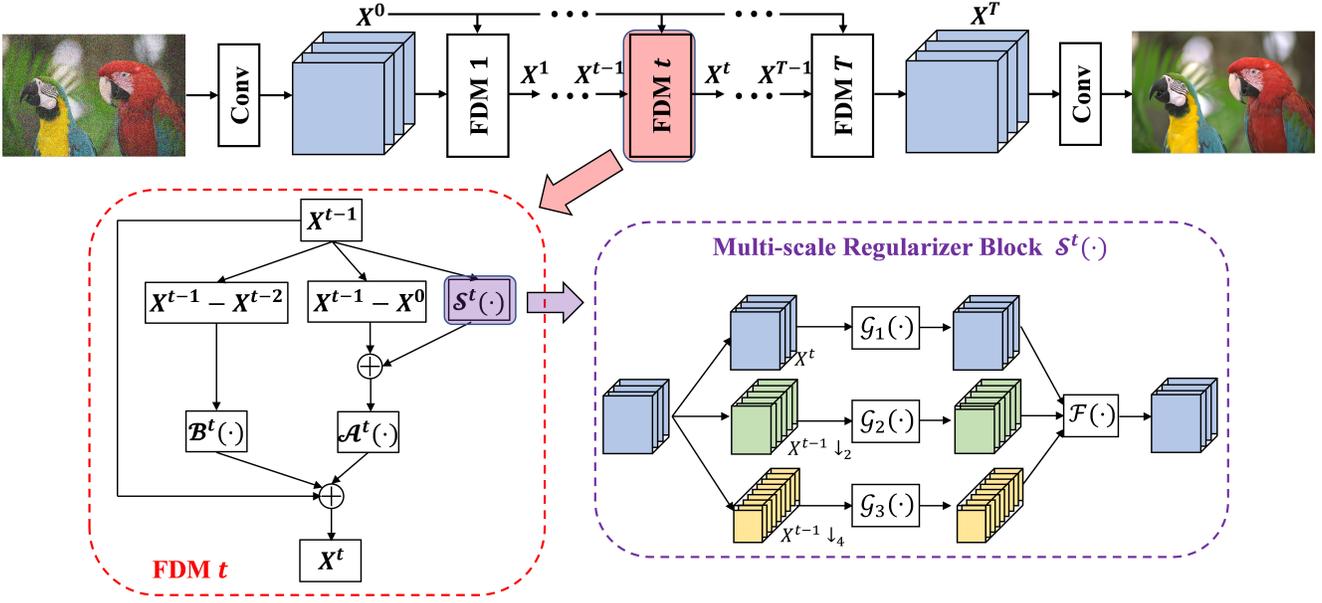


Figure 2. Architecture of DUMRN. The first convolutional layer extracts feature  $\mathbf{X}^0$  from the noisy input, then  $T$  feature-based denoising modules (FDMs) are stacked to remove noise in deep feature space; the structure of FDM is based on the momentum gradient descent algorithm. Benefiting from FDM, there is no need to transform deep features into image space at each step. The last convolutional layer converts the denoised feature  $\mathbf{X}^T$  to a clean image. A multi-scale regularizer block  $\mathcal{S}(\cdot)$  learns deep prior information from features at different resolutions;  $\downarrow_k$  represents down-sampling with a scaling factor of  $k$ .

methods on synthetic and real-world image denoising tasks. Our source code is available at <https://github.com/Xujz19/DUMRN>.

## 4.1. Datasets and implementation

### 4.1.1 Datasets

Following RDN [63] and SADNet [9], we adopt 800 high-resolution training images from the DIV2K dataset [3] to train our models for Gaussian denoising at four different noise levels ( $\sigma = 10, 30, 50, 70$ ). In addition, we use noisy images with varying noise levels  $\sigma \in [0, 75]$  to train a single model for blind Gaussian denoising; it is referred as DUMRN-B. All synthetic Gaussian noisy images were obtained by adding Gaussian noise of different levels to clean images. For real-world image denoising, following DeamNet [46], we trained our model on the SIDD medium dataset [1] and RENOIR dataset [4].

To evaluate DUMRN for gray-scale image denoising, we used three benchmark datasets: Set12[61], BSD68 [40], and Urban100 [29]. For color image denoising, we chose Kodak24 [20], CBSD68, and Urban100 as test datasets. For real image denoising, we evaluated our model on the SIDD Benchmark dataset [1] and Darmstadt Noise Dataset (DnD) [44]. We used peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [64] as the metrics for quanti-

tative evaluation.

### 4.1.2 Implementation details

We implemented our proposed model in the PyTorch environment and adopted the ADAM optimizer [34] with default parameters to optimize the network parameters. We trained our network on an NVIDIA TITAN Xp GPU for a total of 15,000 epochs. For each iteration, we set the batch size to 16 and randomly cropped the noisy and sharp images to size  $96 \times 96$ . Like other denoising methods [61, 62, 63, 9, 41, 53], we performed data augmentation on the training images, using random flipping and rotation. We set the initial learning rate to  $10^{-4}$  and halved it every 2500 epochs. All trainable parameters were initialized using the Xavier method [21]. For the trade-off between efficiency and performance, we set the number of FDMs  $T$  to 6. More details of this choice are provided in Section 5.1.

### 4.2. Synthetic noise removal

To evaluate denoising performance on synthetic noisy images, we compared our proposed DUMRN with several state-of-the-art denoising methods including classical model-based methods (BM3D [13] and CBM3D [12]), deep unfolding methods (TNRD[11], CSCNet [50] and DeamNet [46]), and deep-learning based methods (DnCNN [61],

Table 1. Average PSNR (dB) and SSIM results for different methods on the gray-scale datasets for noise levels  $\sigma = 10, 30, 50$  and  $70$ . Best results are highlighted in bold. ‘-’ means that a result is unavailable.

Method	Set12				BSD68				Urban100			
	10	30	50	70	10	30	50	70	10	30	50	70
BM3D [13]	34.38	29.13	26.72	25.22	33.23	27.74	25.61	24.42	34.47	28.75	25.94	24.27
	0.9233	0.8315	0.7676	0.7174	0.9148	0.7721	0.6855	0.6318	0.9459	0.8567	0.7791	0.7163
TNRD [11]	-	-	26.88	-	-	-	26.01	-	-	-	25.71	-
	-	-	0.7717	-	-	-	0.7057	-	-	-	0.7757	-
CSCNet [50]	-	-	27.15	-	-	-	26.23	-	-	-	26.21	-
	-	-	0.7823	-	-	-	0.7199	-	-	-	0.7957	-
DnCNN [61]	34.78	29.52	27.18	25.52	33.88	28.36	26.23	24.90	34.73	28.88	26.28	24.36
	0.9270	0.8420	0.7827	0.7273	0.9270	0.7999	0.7189	0.6567	0.9486	0.8566	0.7869	0.7178
FFDNet [62]	34.64	29.61	27.32	25.81	33.76	28.39	26.29	25.04	34.45	29.03	26.52	24.86
	0.9271	0.8465	0.7903	0.7451	0.9266	0.8032	0.7245	0.6700	0.9489	0.8707	0.8057	0.7495
NLRN [38]	-	-	27.64	-	-	-	26.47	-	-	-	27.49	-
	-	-	0.7980	-	-	-	0.7298	-	-	-	0.8279	-
SADNet [9]	-	29.94	27.67	26.19	-	28.58	26.49	25.23	-	29.83	27.42	25.82
	-	0.8551	0.8032	0.7622	-	0.8106	0.7340	0.6819	-	0.8876	0.8347	0.7891
DudeNet [53]	34.65	29.54	27.21	-	33.76	28.37	26.25	-	34.41	29.00	26.46	-
	0.9256	0.8430	0.7839	-	0.9251	0.7991	0.7167	-	0.9444	0.8629	0.7945	-
COLANet [41]	34.97	29.81	27.50	25.98	33.93	28.47	26.35	25.10	35.37	30.01	27.47	25.79
	0.9310	0.8486	0.7929	0.7490	0.9284	0.8054	0.7269	0.6704	0.9545	0.8864	0.8289	0.7782
RDN [63]	35.06	29.94	27.60	26.05	33.91	28.54	26.40	25.10	35.42	30.04	27.44	25.69
	0.9315	0.8537	0.7998	0.7558	0.9281	0.8071	0.7275	0.6723	0.9550	0.8886	0.8314	0.7801
DeamNet [46]	-	-	27.74	-	-	-	26.53	-	-	-	27.51	-
	-	-	0.8052	-	-	-	0.7364	-	-	-	0.8374	-
DUMRN	<b>35.12</b>	<b>30.06</b>	<b>27.83</b>	<b>26.33</b>	<b>34.04</b>	<b>28.65</b>	<b>26.56</b>	<b>25.30</b>	<b>35.44</b>	<b>30.19</b>	<b>27.82</b>	<b>26.11</b>
	<b>0.9328</b>	<b>0.8566</b>	<b>0.8068</b>	<b>0.7663</b>	<b>0.9298</b>	<b>0.8120</b>	<b>0.7372</b>	<b>0.6835</b>	<b>0.9555</b>	<b>0.8931</b>	<b>0.8435</b>	<b>0.7976</b>
DUMRN-B	34.94	29.98	27.77	26.32	33.93	28.60	26.52	25.27	35.01	30.00	27.66	26.10
	0.9300	0.8549	0.8047	0.7648	0.9284	0.8107	0.7351	0.6816	0.9519	0.8899	0.8397	0.7957

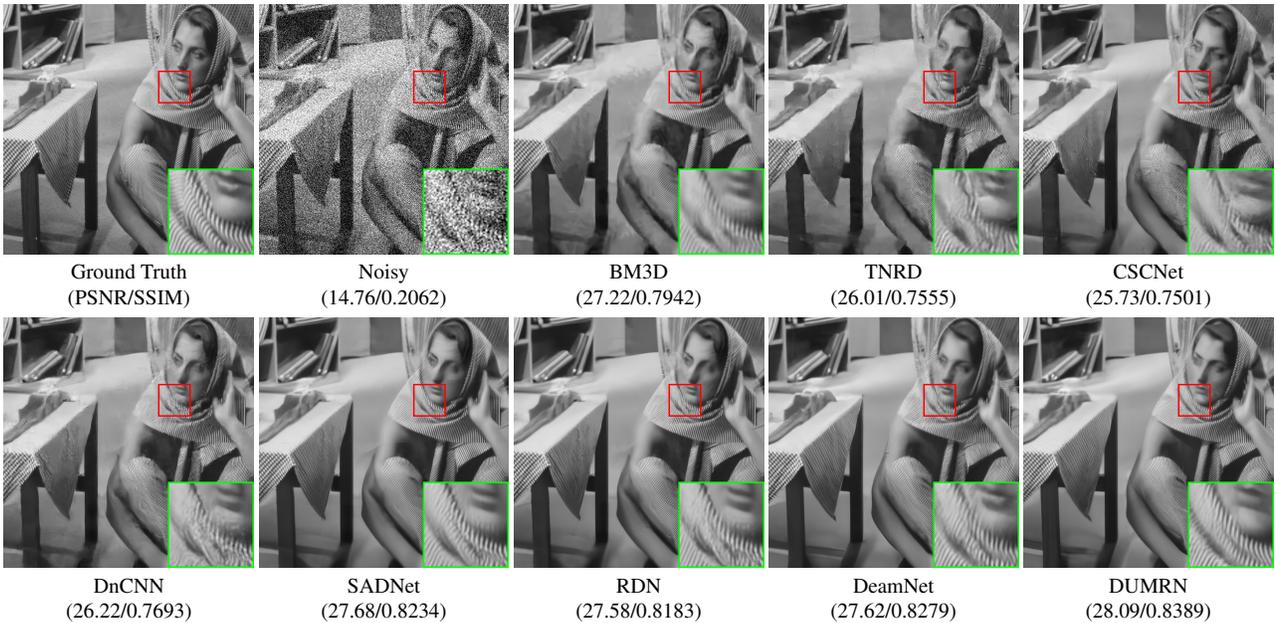


Figure 3. Gray-scale image denoising results on image 09 in Set12 with noise level  $\sigma = 50$ .

FFDNet [62], NLRN[38], SADNet [9], DudeNet [53], COLANet [41], Neb2Neb [30], and RDN [63]).

Tables 1 and 2 summarize quantitative results for different methods on gray-scale and color image denoising, re-

spectively. Our proposed DUMRN achieves the best PSNR under all experiment settings, demonstrating the superiority of our model. Specifically, DUMRN outperforms the most representative traditional method BM3D by 0.74dB

Table 2. Average PSNR (dB) and SSIM results for different methods on the color datasets for noise levels  $\sigma = 10, 30, 50$  and  $70$ . Best results are highlighted in bold. ‘-’ means the result is unavailable.

Method	Kodak24				CBSD68				Urban100			
	10	30	50	70	10	30	50	70	10	30	50	70
CBM3D [12]	36.57	30.89	28.63	27.27	35.91	29.73	27.38	26.00	36.00	30.36	27.94	26.31
	0.9425	0.8450	0.7764	0.7290	0.9507	0.8422	0.7627	0.7073	0.9586	0.8930	0.8404	0.7934
TNRD [11]	-	-	27.24	-	-	-	26.01	-	-	-	25.64	-
	-	-	0.7210	-	-	-	0.7013	-	-	-	0.7696	-
DnCNN [61]	36.99	31.39	29.16	27.64	36.31	30.40	28.01	26.56	36.21	30.28	28.16	26.17
	0.9472	0.8611	0.7985	0.7504	0.9552	0.8642	0.7925	0.7404	0.9607	0.8922	0.8490	0.7924
FFDNet [62]	36.81	31.39	29.10	27.68	36.14	30.31	27.96	26.53	35.77	30.53	28.05	26.39
	0.9461	0.8595	0.7947	0.7473	0.9540	0.8603	0.7881	0.7332	0.9585	0.8983	0.8476	0.8028
SADNet [9]	-	31.80	29.58	28.22	-	30.62	28.31	26.91	-	31.26	28.97	27.45
	-	0.8707	0.8140	0.7718	-	0.8697	0.8041	0.7540	-	0.9108	0.8712	0.8368
DudeNet [53]	36.72	31.41	29.10	-	36.06	30.33	27.96	-	35.65	30.63	28.16	-
	0.9452	0.8605	0.7952	-	0.9531	0.8616	0.7893	-	0.9545	0.8962	0.8454	-
Neb2Neb [30]	36.17	31.24	28.93	-	35.49	30.22	27.86	-	34.85	30.22	27.82	-
	0.9403	0.8594	0.7896	-	0.9492	0.8611	0.7839	-	0.9541	0.8961	0.8423	-
RDN [63]	37.31	31.94	29.65	28.19	36.47	30.67	28.31	26.85	36.69	31.69	29.29	27.64
	0.9498	0.8714	0.8132	0.7659	0.9565	0.8695	0.8024	0.7481	0.9635	0.9147	0.8748	0.8377
DUMRN	<b>37.33</b>	<b>32.02</b>	<b>29.81</b>	<b>28.44</b>	<b>36.50</b>	<b>30.76</b>	<b>28.46</b>	<b>27.06</b>	<b>36.71</b>	<b>31.81</b>	<b>29.53</b>	<b>28.01</b>
	<b>0.9504</b>	<b>0.8743</b>	<b>0.8192</b>	<b>0.7779</b>	<b>0.9570</b>	<b>0.8725</b>	<b>0.8086</b>	<b>0.7593</b>	<b>0.9640</b>	<b>0.9174</b>	<b>0.8810</b>	<b>0.8489</b>
DUMRN-B	37.24	31.99	29.79	28.44	36.42	30.73	28.45	27.06	36.44	31.71	29.49	28.01
	0.9500	0.8740	0.8191	0.7777	0.9565	0.8722	0.8084	0.7593	0.9626	0.9164	0.8804	0.8487

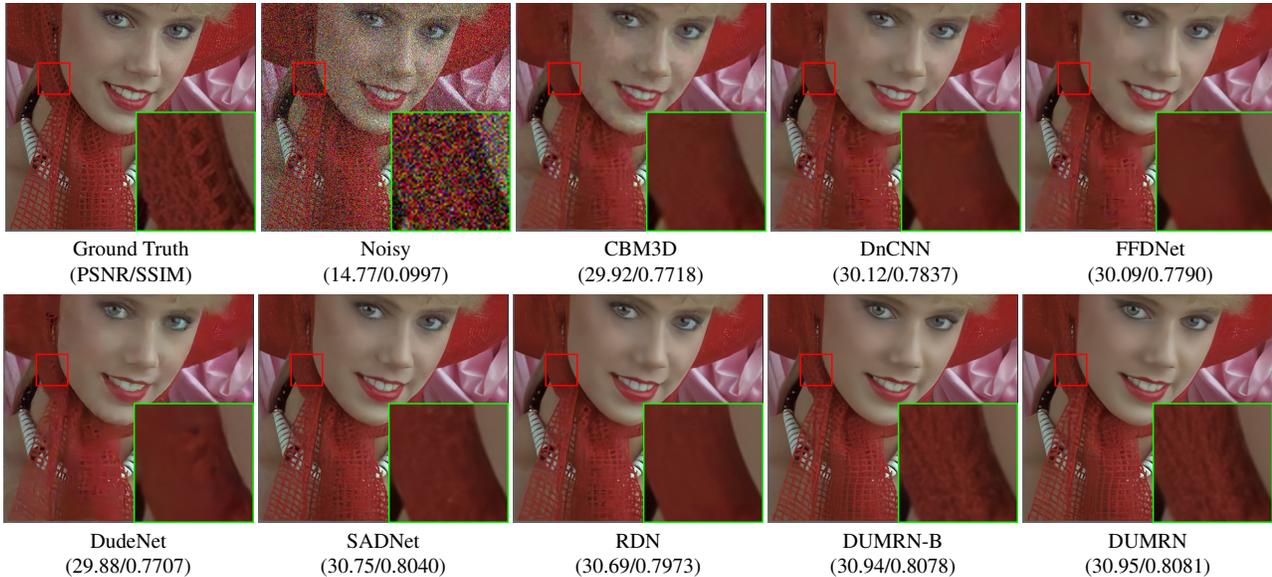


Figure 4. Color image denoising results on image kodim04 from Kodak24 with noise level  $\sigma = 50$ .

to 1.84dB in different settings. For gray-scale image denoising on the Urban100 dataset with noise level  $\sigma = 50$ , DUMRN achieves 2.11dB/0.0678 and 0.31dB/0.0061 improvements over the previous deep unfolding methods TNRD and DeamNet, respectively. Benefiting from the incorporation of the physical model and deep CNNs, our model also outperforms state-of-the-art deep learning-based methods DnCNN, SADNet, COLANet, and RDN. Taking color image denoising with noise level  $\sigma = 50$  as an example, our DUMRN obtains 0.65dB/0.0207, 0.45dB/0.0161,

and 1.37dB/0.0320 improvements over DnCNN on Kodak24, CBSD68, and Urban100 respectively. Due to the proposed FDM and multi-scale strategy that enlarge the receptive field to exploit more spatial contextual information, DUMRN is especially effective when the noise level is high ( $\sigma = 50, 70$ ). Specifically, on the challenging Urban100 dataset, DUMRN outperforms RDN by 0.42dB/0.0175 on gray-scale image denoising and 0.37dB/0.0112 on color image denoising when the noise level is  $\sigma = 70$ .

Figure 3 shows a visual comparison of results from dif-

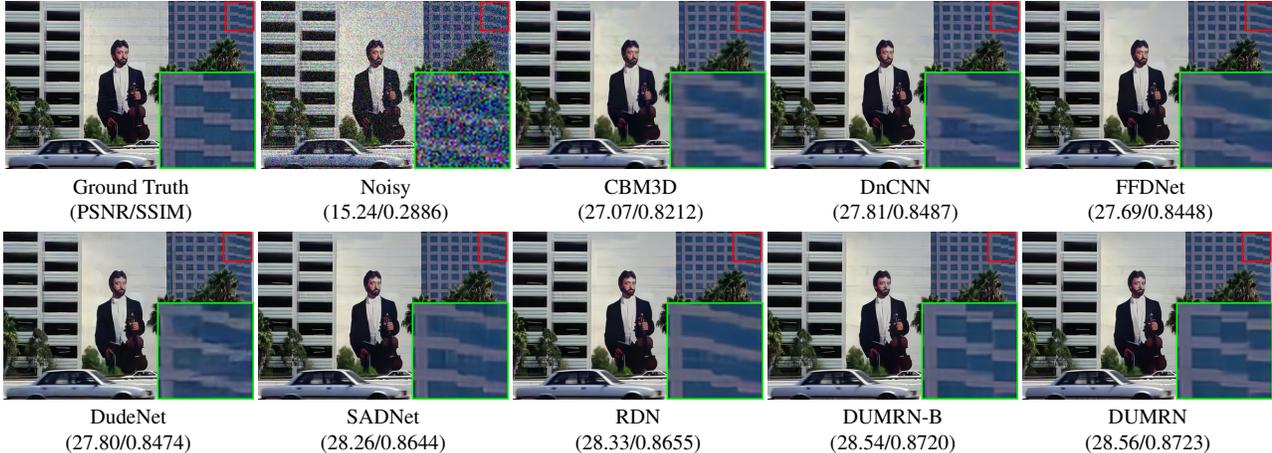


Figure 5. Color image denoising results on image 119082 from CBSD68 with noise level  $\sigma = 50$ .

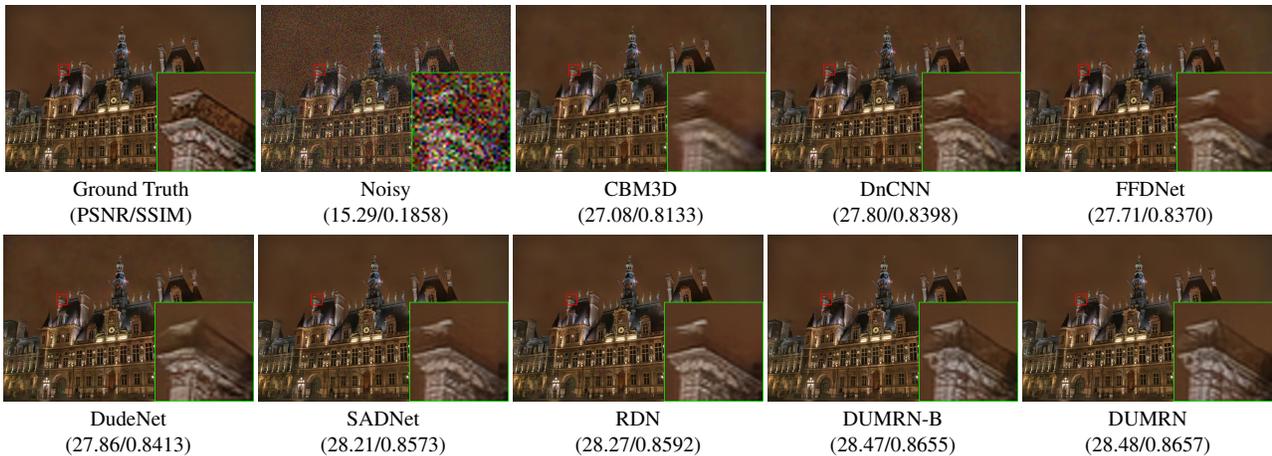


Figure 6. Color image denoising results on image img\_022 from Urban100 with noise level  $\sigma = 50$ .

ferent methods on gray-scale image denoising. We can observe that TNRD and CSCNet generate results with severe distortion and noticeable artifacts, and no competing methods recover a clear edge of the chin in the photo, while DUMRN produces more faithful results. Visual comparisons for color images are shown in Figures 4–6. It can be seen that our DUMRN produces sharper edges and recovers more details while other methods suffer from over-smoothing, demonstrating the powerful representation ability of our DUMRN model.

Quantitative results for our blind denoising model are also given in Tables 1 and 2. The performance of DUMRN-B slightly decreases due to the unknown noise level, but it is still higher than most deep learning-based methods which are trained for a known specific noise level, indicating that our model is robust in blind image denoising. Taking a noise level  $\sigma = 50$  as an example, DUMRN-B outperforms the non-blind RDN by 0.14dB/0.0026, 0.14dB/0.0020, and 0.20dB/0.0056 on the Kodak24, CBSD68, and Urban100 datasets, respectively. As shown in Figures 4–6, DUMRN-

B also generates more faithful results than the other competing methods, further demonstrating that our model can effectively tackle the challenging blind Gaussian denoising task.

### 4.3. Real-world noise removal

Actually, it is hard to model real-world noise precisely. Since investigating the degradation processes of real noise is not our main goal, we simply assume it is a kind of additive noise that can be removed by Eq. (2). We compare the DUMRN to several state-of-the-art methods, including CBM3D [12], DnCNN[61], FFDNet [62], CBDNet [25], AINDNet(TF) [33], PathRestore [57], COLANet [41], R<sup>2</sup>Net [5], and DeamNet [46].

Quantitative results for different methods on the SIDD benchmark and DnD dataset are provided in Table 3. All PSNR and SSIM results were obtained from public benchmark websites. It can be seen that our DUMRN achieves comparable performance to these competing methods. Visual comparisons of results of different methods on an im-

Table 3. Quantitative results for different methods on the SIDD and DnD datasets. Best results are highlighted in bold.

Dataset	SIDD		DnD	
	PSNR	SSIM	PSNR	SSIM
Noisy	23.70	0.480	29.84	0.7018
CBM3D [13]	25.65	0.685	34.51	0.8507
DnCNN [61]	23.66	0.583	37.90	0.9430
FFDNet [62]	29.30	0.694	37.61	0.9415
CBDNet [25]	33.28	0.868	38.06	0.9421
AINDNet(TF) [33]	38.95	0.952	39.37	0.9505
PathRestore[57]	38.21	0.946	39.00	<b>0.9542</b>
COLANet [41]	38.99	0.951	39.45	0.9626
R <sup>2</sup> Net [5]	37.87	0.943	39.25	0.9528
DeamNet [46]	39.35	0.955	39.63	0.9531
DUMRN	<b>39.44</b>	<b>0.956</b>	<b>39.66</b>	0.9529

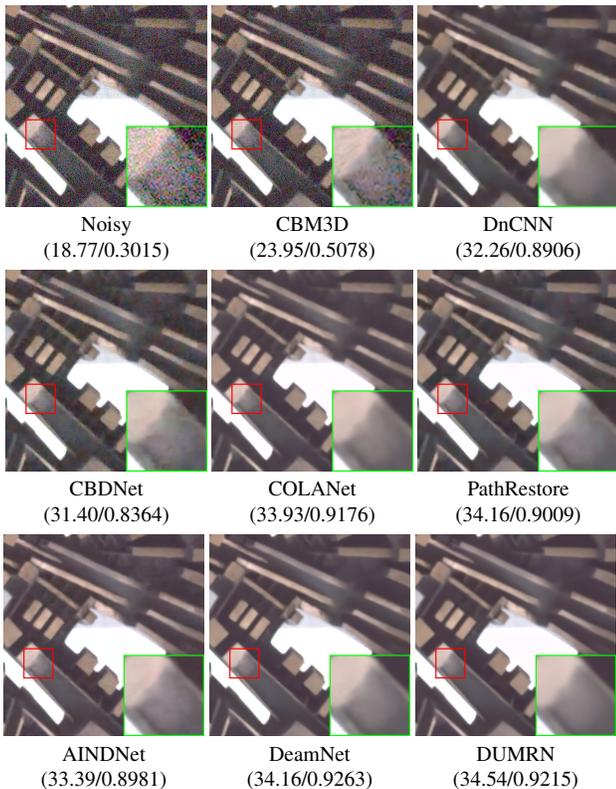


Figure 7. Real image denoising results of different methods on the image from DnD dataset.

age from the DnD dataset are shown in Figure 7. The conventional CBM3D cannot effectively remove real-world noise, and most competing methods (e.g. DnCNN and CBDNet) over-smooth structures while removing noise. In contrast, DUMRN effectively removes noise in constant regions and recovers clear edges with few artifacts.

## 5. Model analysis

In this section, we conduct ablation studies to analyze the effects of different components, including the FDM,

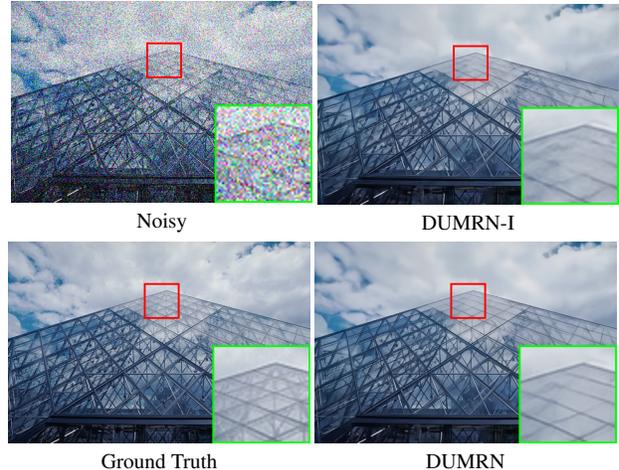


Figure 8. Effectiveness of the FDM on image denoising. Visual comparison on image 223061 from CBSD68 with noise level  $\sigma = 50$ .

MSRB, and consider the unrolling length  $T$ . In addition, we also compare the computational requirements of our DUMRN network and competing methods.

### 5.1. Analysis on FDM

The structure of the proposed DUMRN is similar to TNRD [11] and RGDN [22]; all are based on the gradient descent method. The main difference is that our DUMRN learns a data-driven prior from deep features, but TNRD and RGDN learn priors from the original images. The performance improvement shown in Table 4 of DUMRN over TNRD and RGDN illustrates the superiority of the proposed feature-based denoising framework. To further demonstrate the effect of the deep feature space, we set the feature extractor  $f(\cdot)$  and image reconstructor  $g(\cdot)$  as identity functions (DUMRN-I for short) to learn deep prior information from the raw image space. Table 4 shows that the PSNR of our whole model is 0.19dB higher than DUMRN-I, which illustrates the importance of the feature-based denoising module. A visual comparison of results in Figure 8 further demonstrates the effectiveness of the FDM. It can be seen that the proposed DUMRN with FDM generates fewer artifacts and sharper edges than DUMRN-I: FDM can learn more useful features for high-quality image denoising.

To better demonstrate the effect of the proposed FDM intuitively, we visualize some intermediate features  $\mathbf{X}^t$ . Using the visualization method in [37], we compute the top three principal components of the intermediate features and map them to the principal components of RGB space. The visualized features are shown in Figure 9. It can be seen that noise is removed progressively from  $\mathbf{X}^2$  to  $\mathbf{X}^6$ . Feature  $\mathbf{X}^6$  contains the least noise and has the sharpest texture: the proposed FDM effectively removes noise and reconstructs details.



Figure 9. Visualizations of intermediate features in DUMRN.

Table 4. Analysis of the proposed MSRB. PSNR (dB) results of color image denoising are evaluated on the CBSD68 dataset with noise level  $\sigma = 50$ .

Method	TNRD [11]	RGDN [22]	DUMRN-I	DUMRN
PSNR (dB)	25.96	27.66	28.27	28.46

Table 5. Analysis of the proposed MSRB. PSNR (dB) and SSIM results of color image denoising are evaluated on the Urban100 dataset with noise level  $\sigma = 50$ .

Method	PSNR	SSIM
DUMRN-1	28.81	0.8646
DUMRN-2	29.22	0.8747
DUMRN-3	29.07	0.8710
DUMRN-C	29.41	0.8779
DUMRN-S	29.38	0.8778
DUMRN	29.53	0.8810

The number of FDMs determines the depth of the DUMRN. In Figure 10, we show the variation in PSNR performance and inference time with increasing numbers of steps  $T$ . PSNR values and inference time both increase as  $T$  rises. As  $T$  increases from 5 to 6, an obvious PSNR improvement is still obtained. However, when  $T$  becomes bigger than 6, the curve in Figure 10 becomes flatter, and PSNR gains become minor. Thus, we conclude that DUMRN has almost converged when  $T = 6$ . Although DUMRN achieves the best PSNR performance when  $T = 8$ , it takes the most inference time. Considering the trade-off between the performance and inference time, we adopt  $T = 6$  in our experiments.

## 5.2. Analysis on MSRB

To investigate the effect of the proposed multi-scale strategy and the adopted BPFF block, we designed several baseline models. Specifically, we trained the following alternatives of our model: (i) removing  $\mathcal{G}_2(\cdot)$  and  $\mathcal{G}_3(\cdot)$  in MSRB (DUMRN-1 for short), (ii) removing  $\mathcal{G}_3(\cdot)$  in MSRB

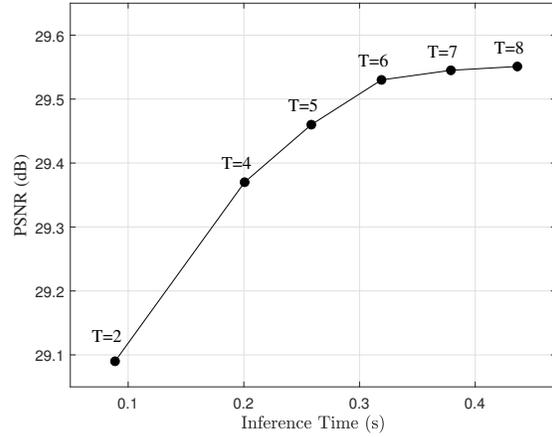


Figure 10. Varying the number  $T$  of FDMs. PSNR (dB) results for color image denoising are evaluated on the Urban100 dataset with noise level  $\sigma = 50$ .

(DUMRN-2 for short), (iii) changing the number of filters in  $\mathcal{G}_2(\cdot)$  and  $\mathcal{G}_3(\cdot)$  to 64 and letting them learn deep prior information at the original full-resolution (DUMRN-3 for short), (iv) replacing BPFF block with concatenation (DUMRN-C for short), (v) replacing the BPFF block with summation (DUMRN-S for short). Quantitative results are shown in Table 5. Compared to DUMRN-1, DUMRN-2 has one more branch that learns deep prior information at the coarse resolution, and DUMRN-3 has two more branches that learn deep prior information at the original full-resolution. It can be seen that DUMRN-2 achieves better denoising performance than DUMRN-3, indicating that using features at different scales can provide more effective prior information. In addition, DUMRN outperforms DUMRN-1, DUMRN-2, and DUMRN-3, which further demonstrates that multi-scale information facilitates image denoising. Visual comparisons are given in Figure 11; we can observe that a learning deep prior at the original full-resolution (DUMRN-1) is

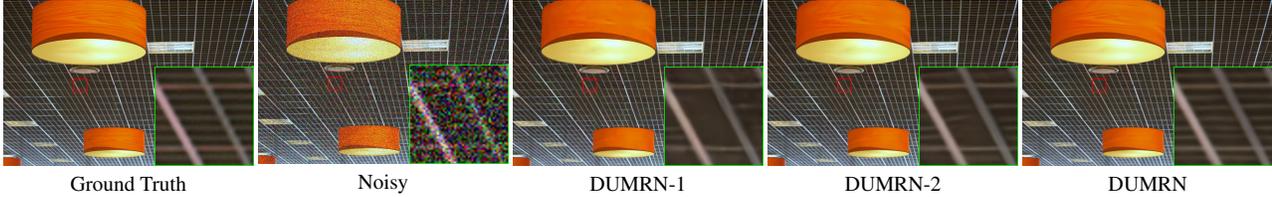


Figure 11. Effectiveness of the MSRB on image denoising. Visual comparison on image `img_044` from Urban100 with noise level  $\sigma = 50$ .

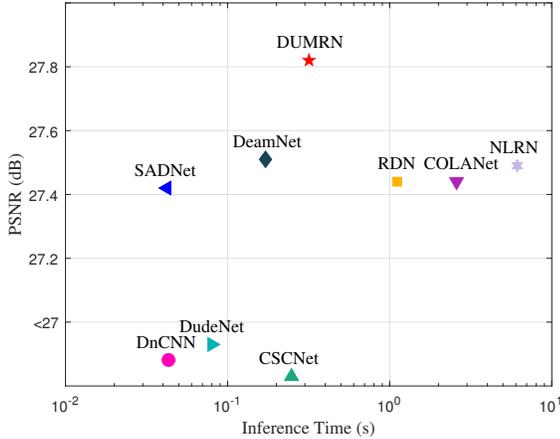


Figure 12. Inference time versus PSNR for different methods. PSNR (dB) results for gray-scale image denoising were evaluated on the Urban100 dataset with noise level  $\sigma = 50$ .

insufficient to recover fine texture details. Taking advantage of the multi-scale prior information, the denoised result of DUMRN contains fewer artifacts and much more detail than the results of DUMRN-1 and DUMRN-2.

From Table 5, it can also be observed that DUMRN generates better results than DUMRN-C and DUMRN-S: simply using concatenation or summation can not effectively integrate multi-scale information.

### 5.3. Speed

We further evaluated the inference time for different methods for processing a  $512 \times 512$  gray-scale image. All inference times were tested on an NVIDIA TITAN Xp GPU. Figure 12 shows that our DUMRN achieves better PSNR performance with a lower inference time than the state-of-the-art methods RDN [63], COLANet [41], and NLRN [38]. Although DUMRN is a little slower than DnCNN [61], DudeNet [53], and SADNet [9], it achieves much better denoising performance. Overall, DUMRN performs well in terms of both effectiveness and efficiency.

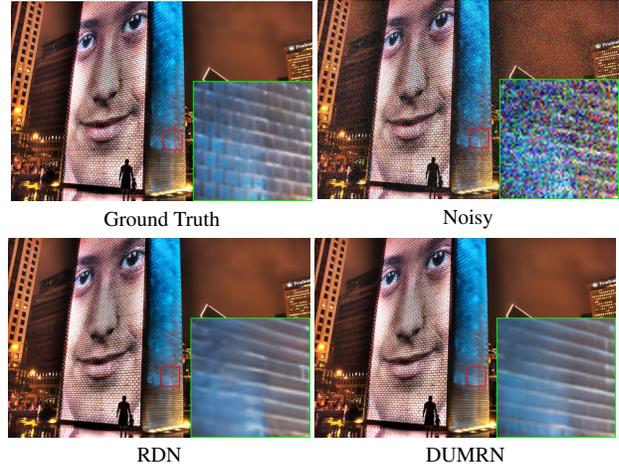


Figure 13. Unsatisfactory results. Visual comparison on image `img_076` from Urban100 with noise level  $\sigma = 50$ .

## 6. Limitations

Like other denoising methods, DUMRN may fail to reconstruct proper details in some challenging cases. As Figure 13 shows, strong noise may make vertical textures imperceptible in the noisy input, and as a result DUMRN and the state-of-the-art RDN cannot correctly recover the vertical textures shown in the close-up. Severe corruption makes it difficult to restore the textures, and DUMRN generates the most likely texture patterns learned from the training dataset.

## 7. Conclusions

In this paper, we proposed the deep unfolding multi-scale regularizer network (DUMRN) to better integrate the traditional image denoising model with deep neural networks. We explicitly consider the image denoising process in deep feature space, and propose a feature-based denoising module (FDM) following the iterative optimization steps of the image degradation model. Benefiting from the FDM, we can construct a deep network with a large receptive field to effectively learn deep prior information. In addition, we proposed the multi-scale regularizer block (MSRB) to extract more spatial contextual information from features of different scales, which is beneficial for images that suffer from heavy noise. We also analyzed the

effect of each component in the proposed DUMRN. Extensive experiments demonstrate that our proposed DUMRN can effectively and robustly denoise images, assessed both by quantitative metrics and visual quality. In future, we will investigate real noise models and modify DUMRN to achieve better results for real image denoising. We will also extend our model to other image restoration tasks, such as image deblurring and rain removal.

## Acknowledgements

This work was partially supported by the National Key R&D Program of China (No.2020YFA0714100) and the National Nature Science Foundation of China (Nos. 61872162, 62102414, 62172415, 52175493).

## References

- [1] A. Abdelhamed, S. Lin, and M. S. Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2018. 5
- [2] J. Adler and O. Öktem. Learned primal-dual reconstruction. *IEEE Transactions on Medical Imaging*, 37(6):1322–1332, 2018. 1, 3
- [3] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1122–1131, 2017. 5
- [4] J. Anaya and A. Barbu. Renoir – a dataset for real low-light image noise reduction. *Journal of Visual Communication and Image Representation*, 51:144–154, 2018. 5
- [5] S. Anwar, N. Barnes, and L. Petersson. Attention-based real image restoration. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–11, 2021. Available at <https://doi.org/10.1109/TNNLS.2021.3131739>. 8, 9
- [6] A. Barbu. Training an active random field for real-time image denoising. *IEEE Transactions on Image Processing*, 18(11):2451–2462, 2009. 2
- [7] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 60–65, 2005. 1
- [8] A. Buades, B. Coll, and J.-M. Morel. Nonlocal image and movie denoising. *International Journal of Computer Vision*, 76(2):123–139, 2008. 1
- [9] M. Chang, Q. Li, H. Feng, and Z. Xu. Spatial-adaptive network for single image denoising. In *Proceedings of the European Conference on Computer Vision*, pages 171–187, 2020. 1, 2, 4, 5, 6, 7, 11
- [10] F. Chen, L. Zhang, and H. Yu. External patch prior guided internal clustering for image denoising. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 603–611, 2015. 1, 2
- [11] 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*, 39(6):1256–1272, 2017. 1, 2, 3, 4, 5, 6, 7, 9, 10
- [12] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 313–316, 2007. 5, 7, 8
- [13] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. 1, 2, 5, 6, 9
- [14] S. Dai, M. Han, Y. Wu, and Y. Gong. Bilateral back-projection for single image super resolution. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 1039–1042, 2007. 4
- [15] H. Dong, J. Pan, L. Xiang, Z. Hu, X. Zhang, F. Wang, and M.-H. Yang. Multi-scale boosted dehazing network with dense feature fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2154–2164, 2020. 4
- [16] W. Dong, G. Shi, and X. Li. Nonlocal image restoration with bilateral variance estimation: A low-rank approach. *IEEE Transactions on Image Processing*, 22(2):700–711, 2013. 1
- [17] W. Dong, L. Zhang, G. Shi, and X. Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, 2013. 1
- [18] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006. 1, 2
- [19] W. Feng, P. Qiao, X. Xi, and Y. Chen. Image denoising via multiscale nonlinear diffusion models. *SIAM Journal on Imaging Sciences*, 10(3):1234–1257, 2017. 2, 4
- [20] R. Franzen. Kodak lossless true color image suite. source: <http://r0k.us/graphics/kodak>, 1999. 5
- [21] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010. 5
- [22] D. Gong, Z. Zhang, Q. Shi, A. van den Hengel, C. Shen, and Y. Zhang. Learning deep gradient descent optimization for image deconvolution. *IEEE Transactions on Neural Networks and Learning Systems*, 31(12):5468–5482, 2020. 1, 2, 3, 9, 10
- [23] S. Gu, Y. Li, L. Van Gool, and R. Timofte. Self-guided network for fast image denoising. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2511–2520, 2019. 2
- [24] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang. Weighted nuclear norm minimization and its applications to low level vision. *International Journal of Computer Vision*, 121(2):183–208, 2017. 1
- [25] 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*, pages 1712–1722, 2019. 2, 8, 9

- [26] M. Haris, G. Shakhnarovich, and N. Ukita. Deep back-projection networks for single image super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4323–4337, 2021. 4
- [27] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2
- [28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2261–2269, 2017. 2
- [29] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015. 5
- [30] T. Huang, S. Li, X. Jia, H. Lu, and J. Liu. Neighbor2neighbor: Self-supervised denoising from single noisy images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 14776–14785, 2021. 6, 7
- [31] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*, page 448–456, 2015. 2
- [32] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical Models and Image Processing*, 53(3):231–239, 1991. 4
- [33] Y. Kim, J. W. Soh, G. Y. Park, and N. I. Cho. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3479–3489, 2020. 2, 8, 9
- [34] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015. 5
- [35] X. Lan, S. Roth, D. Huttenlocher, and M. J. Black. Efficient belief propagation with learned higher-order markov random fields. In *Proceedings of the European Conference on Computer Vision*, pages 269–282, 2006. 1
- [36] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1132–1140, 2017. 3
- [37] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):978–994, 2011. 9
- [38] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang. Non-local recurrent network for image restoration. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 31, 2018. 6, 11
- [39] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2272–2279, 2009. 1
- [40] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 416–423 vol.2, 2001. 5
- [41] C. Mou, J. Zhang, X. Fan, H. Liu, and R. Wang. Colanet: Collaborative attention network for image restoration. *IEEE Transactions on Multimedia*, pages 1–1, 2021. Available at <https://doi.org/10.1109/TMM.2021.3063916>. 1, 4, 5, 6, 8, 9, 11
- [42] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 257–265, 2017. 3
- [43] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning*, 2010. 3
- [44] T. Plötz and S. Roth. Benchmarking denoising algorithms with real photographs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2750–2759, 2017. 5
- [45] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, 1999. 3
- [46] C. Ren, X. He, C. Wang, and Z. Zhao. Adaptive consistency prior based deep network for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8596–8606, 2021. 3, 5, 6, 8, 9
- [47] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. 2
- [48] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 2009. 1
- [49] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2774–2781, 2014. 2
- [50] D. Simon and M. Elad. Rethinking the csc model for natural images. In *Proceedings of the Advances in Neural Information Processing Systems*, 2019. 2, 3, 4, 5, 6
- [51] J. Sun and M. F. Tappen. Learning non-local range markov random field for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2745–2752, 2011. 2
- [52] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the International Conference on Machine Learning*, page III–1139–III–1147, 2013. 3
- [53] C. Tian, Y. Xu, W. Zuo, B. Du, C.-W. Lin, and D. Zhang. Designing and training of a dual cnn for image denoising. *Knowledge-Based Systems*, 226:106949, 2021. 5, 6, 7, 11
- [54] X. Wu, B. Zhou, Q. Ren, and W. Guo. Multispectral image denoising using sparse and graph laplacian tucker decomposition. *Computational Visual Media*, 6(3):319–331, 2020. 1

- [55] Q. Xie, Q. Zhao, Z. Xu, and D. Meng. Color and direction-invariant nonlocal self-similarity prior and its application to color image denoising. *Science China Information Sciences*, 63(12):257–264, 2020. [1](#)
- [56] Y. Yang, J. Sun, H. Li, and Z. Xu. Deep admn-net for compressive sensing mri. In *Proceedings of the Advances in Neural Information Processing Systems*, 2016. [1](#), [3](#)
- [57] K. Yu, X. Wang, C. Dong, X. Tang, and C. C. Loy. Path-restore: Learning network path selection for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. Available at <https://doi.org/10.1109/TPAMI.2021.3096255>. [2](#), [8](#), [9](#)
- [58] X. Yu, Z. Fu, and C. Ge. A multi-scale generative adversarial network for real-world image denoising. *Signal, Image and Video Processing*, 16(1):257–264, 2022. [2](#)
- [59] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao. Learning enriched features for real image restoration and enhancement. In *Proceedings of the European Conference on Computer Vision*, pages 492–511, 2020. [2](#)
- [60] K. Zhang, L. Van Gool, and R. Timofte. Deep unfolding network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3214–3223, 2020. [2](#), [3](#)
- [61] 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*, 26(7):3142–3155, 2017. [1](#), [2](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [11](#)
- [62] K. Zhang, W. Zuo, and L. Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. [1](#), [2](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#)
- [63] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7):2480–2495, 2021. [1](#), [2](#), [5](#), [6](#), [7](#), [11](#)
- [64] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. [5](#)
- [65] M. Zontak, I. Mosseri, and M. Irani. Separating signal from noise using patch recurrence across scales. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1195–1202, 2013. [4](#)