Cross-Patch Graph Convolutional Network for Image Denoising

Yao Li, Xueyang Fu, Zheng-Jun Zha*
University of Science and Technology of China, China
xslx@mail.ustc.edu.cn, {xyfu, zhazj}@ustc.edu.cn

Abstract

Recently, deep learning-based image denoising methods have achieved significant improvements over traditional methods. Due to the hardware limitation, most deep learning-based image denoising methods utilize cropped small patches to train a convolutional neural network to infer the clean images. However, the real noisy images in practical are mostly of high resolution rather than the cropped small patches and the vanilla training strategies ignore the cross-patch contextual dependency in the whole image. In this paper, we propose Cross-Patch Net (CPNet), which is the first deep- learning-based real image denoising method for HR (high resolution) input. Furthermore, we design a novel loss guided by the noise level map to obtain better performance. Compared with the vanilla patch-based training strategies, our approach effectively exploits the cross-patch contextual dependency. Besides, owing to the difficulty in capturing real noisy and noise-free image paired training data, we propose an effective method to generate realistic sRGB noisy images from their corresponding clean sRGB images for denoiser training. Denoising experiments on real-world sRGB images show the effectiveness of the proposed method. More importantly, our method achieves state-of-the-art performance on practical sRGB noisy image denoising.

1. Introduction

Since image denoising can help downstream computer vision tasks [44, 28, 43, 27, 26, 32], it has attracted extensive interest in related fields. The majority of denoising methods take the cropped small patches as the training dataset due to the hardware storage limitation like GPU memory. However, these methods trained with the cropped patches may fail when denoising the real noisy images in the practical situation. Nowadays, the images captured by the cameras always have high-resolution and there exists context consistency between the patches in a whole image.

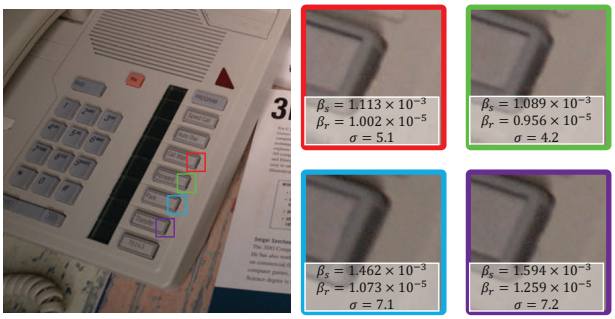

Figure 1. An example of the patch-wise noise consistency in SIDD [1]. The four similar patches have almost the same NLFs (β_s and β_r for noise level function – see Section 3.1). The NLFs and the Gaussian noise level are estimated by using the method proposed by [13]and [8], respectively.

A large number of denoising methods can achieve considerable performance when they adopt the cropped real noisy image patches as training data set dealing with the Gaussian noise. Nevertheless, there exist virtual differences between the Gaussian noise and the real noise. The noise levels of patches in a Gaussian noisy image are the same which is the fixed variance. Therefore, it is unnecessary to consider the context consistency between Gaussian noisy patches.

It is quite different from the real noise. The real sRGB noise is generated by the raw image noise through the image processing pipeline (ISP). The raw noise can be divided into two categories: shot noise and read noise[13, 29], which obey Poisson distribution and Gaussian distribution respectively. Poisson noise is highly relevant to the image pixel values. Besides, the noise on each pixel is affected by the pixels in the adjacent region when the raw image is converted to an sRGB image. As a result, similar patches generate similar noise distribution. Figure 1 gives an example of this phenomenon, illustrating that the high-resolution noisy images consist of a series of similar patches, and the noise level functions (NLFs) are almost the same (shot noise σ_s and read noise σ_r). All the NLFs are estimated by using the SIDD[1] raw noisy images with the NLFs estimation method [13]. Therefore, It is essential to take the cross-

*Corresponding author.

patch consistency into account dealing with the real noise.

To obtain better real image denoising performance, noise level maps are widely used as inputs in nowadays denoising methods[18, 4, 42]. Noise level map consists of not only the original image pixel values information but also the real noise distribution information of each pixel. Both the real noise and the noise level map record these two kinds of information mentioned above. But unlike the noise level map, the real noise is accompanied by uncertainty and randomness, which makes it difficult to extract the original pixel values and the noise levels. Although the noise level map has superior properties, most existing methods only concatenate it with the noisy image as the network input and do not leverage it efficiently.

In addition to the cross-patch consistency, the real image denoising performance is limited by the lack of real image training data. Due to the difficulties during image capturing (object movement, camera motion, and lighting changes), all these real noisy image data sets[1, 34, 3, 39, 31] have limited numbers of scenes and images. For example, the largest data set SIDD has only 10 scenes and 160 training image pairs. Each real HR noisy image can be cropped into multiple noisy patch training pairs, but the real noise parameters of each image are fixed which makes a large number of training pairs share the same real noise parameters. However, there is a wide range of noise parameters in the practical situation so that the lack of real noise parameters in the training images affects the robustness of denoiser to unknown noise parameters. Different from Gaussian noise, the noise of the real image is often complex and difficult to simulate. To tackle this issue, we propose a method to synthesize realistic sRGB noisy images from clean images.

In this paper, we propose CPNet, a novel patch-based deep learning method for real image denoising. Specifically, we crop an input image into patches and a primary patches selection is made according to the semantic relevance among them. Then we construct a cross-patch graph and propose Cross-Patch Graph Convolutional Network. we aggregate both the local and non-local information to the decoder to obtain the predicted clean image. Cross-Patch GCN explicitly captures cross-patch long-range contextual dependency. For each given patch to be estimated (query patch), CPNet aggregates other patches which are highly relevant to the query one. Then CPNet ensembles those correlated features towards a more faithful predicted clean image. To solve the problem of insufficient training data set, we propose an effective method to generate realistic sRGB noisy images from their corresponding clean sRGB images for denoiser training.

In summary, our contributions are as follows:

- We propose a cross-patch strategy to explore the contextual consistency between patches in a high-resolution real noisy image.
- We propose a novel loss to leverage the noise level map. Instead of merely regarding the noise level map as the input in previous work, we further use it to supervise the network training.
- We design an effective pipeline to generate realistic sRGB noisy images from their corresponding clean sRGB images for denoiser training.
- We propose a graph convolutional network CPNet to practical HR real image denoising under hardware resources constraints. Through extensive experiments on different datasets, we show that the proposed method is able to achieve state-of-the-art performance.

2. Related Work

Image denoising has attracted wide interest in computer vision. There are two main approaches to denoise an image. One is a classic technique that uses hand-engineered algorithms to model the image priors which play an essential role in image denoising from a Bayesian viewpoint. These methods include, but are not limited to, non-local self-similarity (NSS) models [6], sparse models [11], gradient models [33], and Markov random field (MRF) models [23]. Self-similarity-driven techniques are still popular in recent methods such as BM3D [10] and WNNM [17]. Despite the progress in image denoising, the process is time-consuming as a result of complicated optimization in the test stage. Besides, the classic technique involves a number of manually chosen parameters that have a significant effect on denoising performance. Due to these two issues, CNN-based denoising methods come to power [22, 41, 12, 9, 25, 35, 7]. Benefited from the modeling capability of CNNs, these methods generally achieve state-of-the-art performance for blind denoising of images with simulated Gaussian noise in the sRGB space. Nevertheless, owing to the unsatisfactory denoising performance in real noisy images, denoising of real images has been the focus of recent research in image denoising [4, 18, 42]. All of them use noise level maps to help denoising.

Cross-patch consistency has been widely considered in image restoration. Similar patches frequently recur in a natural image inherently, leading to many classical methods, e.g., non-local means [6] and BM3D[10] aggregate similar patches to infer clean images. Nevertheless, the majority of deep learning-based denoising methods ignore the cross-patch consistency.

Since the noise level map contains abundant real noise information, a large number of methods regard the noise level map as additional input, such as[4, 40]. But they only take the noise map as part of the input. There do not exist subsequent processing on it which leads to insufficient utilization of the real noise parameters.

Graph convolutional network[21, 19, 14, 15] has been successfully applied in image restoration. GCDN[37] introduce a lightweight Edge-Conditioned Convolution that addresses vanishing gradient and over-parameterization issues of this particular graph convolution for image denoising. IGNN[45] propose a graph network that explores the internal recurrence property for image super-resolution. The graph network learns the repetitive textures, corners and edges from the real image. We use GCN for three reasons: Firstly, there exist a number of small similar structures in the real noisy image. They are suitable to be modeled as graph data to learn informative representations for nodes based on the original node features and the structure information. Secondly, the relevance between similar patches can just be represented by the edge weights of the two nodes in GCN. Thirdly, the traditional non-local approaches involve many manually chosen parameters that affect denoising performance, while GCN learns the parameters adaptively and boosts the denoising performance.

There are plenty of real image data set[1, 34, 3, 39, 31, 5]. However, making a high-quality training data set is highly expensive[1] so that the number of the images and scenes is limited. Because of the difficulty of capturing a large amount of real noisy and noise-free image pairs, some methods[4, 40] synthesize realistic sRGB noisy images based on modeling the key components of the ISP pipeline for denoiser training. However, the generated noisy images are not realistic enough since there are some non-invertible ISP components such as demosaicing[30]. We design a network to learn non-invertible components in ISP for realistic sRGB noisy image generation.

In this paper, we utilize the GCN to capture the cross-patch contextual dependency and optimize the training loss to exploit the properties of the noise level map. Besides, we design a network to enlarge the training dataset when training the CPNet.

3. Our method

In image denoising, we crop an input image and its noise level map (Sec. 3.1) into patches for training our network. Given a query patch, we find image patches similar to the query patch in the whole image (Sec. 3.2). To leverage cross-patch information efficiently, we propose a novel Cross-Patch GCN (Sec. 3.3). Besides, we propose a new loss to efficiently exploit the properties of the noise level map (Sec. 3.4). Then, we describe the details of our network (Sec. 3.5). The proposed procedure is shown in Figure 2. Finally, we design a noisy image generation network to circumvent the lack of the training dataset (Sec. 3.6).

3.1. Background of Noise Level Map

Noise level map, an image containing original pixel values and real noise parameters information, is widely used in nowadays image denoising methods[18, 4, 42]. We can obtain noise level map n by Eqn.1:

$$\begin{aligned} \mathbf{n} &= \beta_s * \mathbf{L} + \beta_r, \\ \mathbf{L} &= f^{-1}(\mathbf{i}), \end{aligned} \tag{1}$$

where $\mathbf{i}$ is the clean image and f is the camera response function (CRF)[30]. β_s: the signal-dependent multiplicative component of the shot noise. β_r:the independent additive Gaussian component of the read noise.

The raw noise model can be presented as [13]:

$$\mathbf{y} = \mathbf{x} + \mathcal{N}\left(\mathbf{0}, \sigma\left(\mathbf{x}\right)\right), \qquad \sigma^2\left(\mathbf{x}\right) = \beta_s \mathbf{x} + \beta_r, \tag{2}$$

$\mathbf{x}$ and $\mathbf{y}$ represent the clean and noisy raw images, respectively. From above we can see that even though all the patches in one image share the same NLFs (β_s and β_r), most patches have different noise levels due to the difference of the original pixel values. As for similar patches, the real noise is unlikely the same owing to the randomness of the noise, but the noise distribution and noise level is most likely the same. Consequently, the cross-patch consistency is worthy to be studied in the real image denoising task. Empirically, we follow [4] and define f^{-1} as the inverse ISP for computation simplification without sacrificing the accuracy. Each noise level map and the noisy image are concatenated as the input.

3.2. Cross-Patch Sampling

We assume the $l \times l$ query patch as $\mathbf{I}_q$. Our goal is to find patches similar to query patch $\mathbf{I}_q$ in the whole image. Instead of using all the copped patches in the whole image, we only sample N candidates $\mathbf{I}_c^i, i \leq N$ by the stride s_1 and then select top-K neighboring patches to save computation without sacrificing the accuracy. Specifically, we cropped the whole noisy image concatenated with its noise level map into patches $\mathbf{I}_c^i$. To find the K neighboring feature patches, we first extract the semantic features E_q and $\mathbf{E}_c^i$ of $\mathbf{I}_q$ and $\mathbf{I}_c^i$ by the encoder. We find K $l \times l$ nearest neighboring patches $\mathbf{E}_n^i, i \leq K$ according to the Euclidean distance between the query feature map $\mathbf{E}_q$ and other candidates $\mathbf{E}_c^i$. The smaller distance indicates that the candidate patch is more correlated to the query patch, and thus should play a more essential role in information propagation. Empirically, we find that K = 3 can already achieve comparable accuracy compared to utilizing all N context patches.

3.3. Cross-Patch GCN

To further relieve the computation burden of the GCN, we set stride s_2 to extract $d \times d$ patches $\mathbf{E}_{q_j}^1$ and $\mathbf{E}_{n_j}^i$ from $\mathbf{E}_q$ and $\mathbf{E}_n^i$. As illustrated in Figure 3, for each $\mathbf{E}_{q_j}^1$, the Euclidean distances between the feature $\mathbf{E}_{q_j}^1$ and all of the feature of $\mathbf{E}_{n_j}^i$ are computed and an edge is drawn between

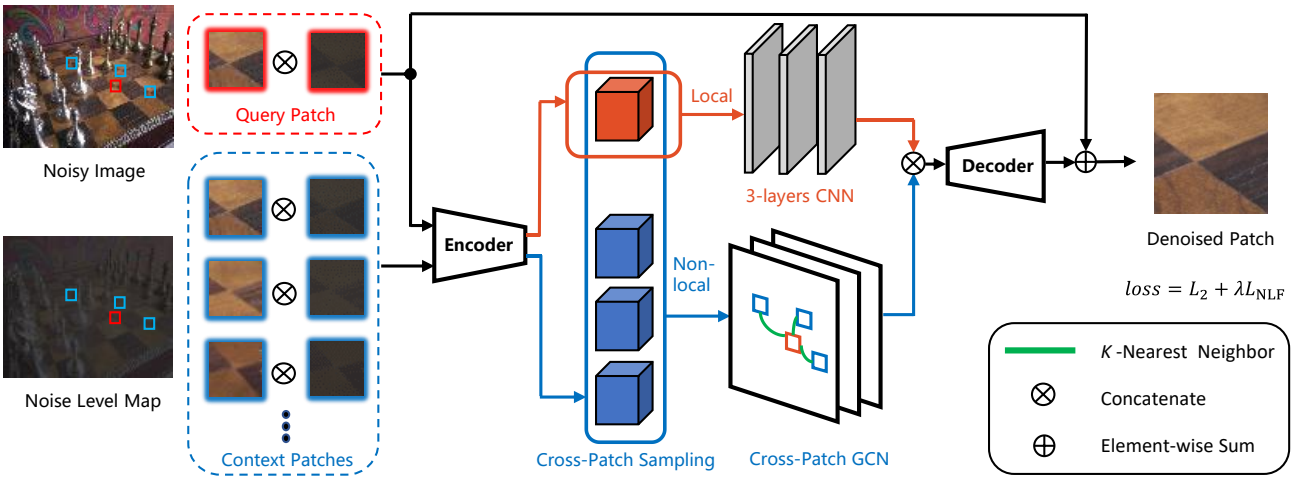

Figure 2. The architecture of CPNet. The whole image concatenated with its noise level map is taken as the encoder input. Given a query patch, the similar context patches will be found by cross-patch sampling. The following framework comprises two branches: local and non-local networks. For the former, the branch extracts the correlation of the inner patch. For the latter, Cross-Patch GCN extracts patch-wise consistency. Then we aggregate the local and non-local features as the input of the decoder. Finally, a residual connection is applied to help generate the clean image.

$\mathbf{E}^1_{q_j}$ and the K $\mathbf{E}^i_{n_j}$ with smallest distance. As verified in [46], there are plenty of recurring patches in the whole image, hence we can assume K patches $\mathbf{E}^i_{n_j}$ are similar to $\mathbf{E}^1_{q_j}$.

The connections between cross-patches can be well constructed as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where every patch is a vertex $\mathcal{V}$ and edge $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ is a similarity-weighted connection of two vertices. In this work, we assume $\mathcal{G}$ as a labeled directed graph without self-loops. The graph is constructed as a $K-$nearest neighbor graph in the feature space. We also assume that each edge has its label, and set the edge labeling function as the difference between the two features at layer m: $\mathbf{d}(m, i, j) = \mathbf{E}^m_{q_j} - \mathbf{E}^i_{n_j}$.

Inspired by the Edge-Conditioned Convolution [36], we aggregate k patches $\mathbf{E}^i_{n_j}$ and the non-local aggregation at layer m is computed as :

$$\begin{aligned} \mathbf{E}^m_{q_j} &= \frac{1}{S^m_j} \sum_i \exp(\mathcal{F}^m \mathbf{d}(m, i, j) \mathbf{E}^i_{n_j}), \\ S^m_j &= \sum_i \exp(\mathcal{F}^m \mathbf{d}(m, i, j)), \end{aligned} \tag{3}$$

where $\mathcal{F}^m$ is a fully connected edge conditioned convolutional layer at layer m that takes as input the edge labels. $\exp(\cdot)$ denotes the exponential function. S^m_j represent the normalization factor. In our Cross-Patch GCN, convolutional layers $M = 3$ are performed. The $K-$nearest neighbor in the graph is only calculated once. The rest two convolutional layers share the same vertices. By exploiting the edge labels, the proposed GCN aggregates K semantic similar features robustly and flexibly.

Despite the non-local operation of the Cross-Patch GCN, a classical local convolution processes the local neighboring

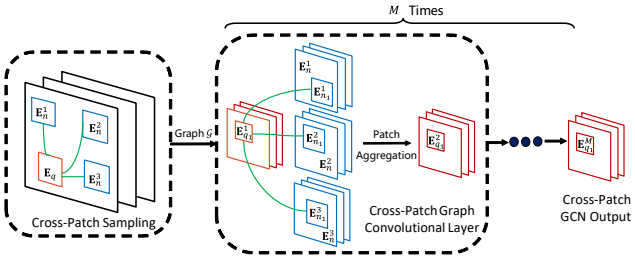

Figure 3. An illustration of the Cross-Patch GCN. $\mathbf{E}^i_n, i \leq K$ are the candidates similar to the query patch $\mathbf{E}^q_n$, which can be obtained from Cross-Patch Sampling. Then we crop $\mathbf{E}^q_n$ and candidates $\mathbf{E}^i_n$ into smaller patches such as $\mathbf{E}^1_{q_j}$ and its nearest neighbors $\mathbf{E}^i_{n_j}$ to construct the graph $\mathcal{G}$, After the processing of the GCN, the output is embedded with cross-patch contextual consistency.

to provide the output feature. Empirically, we design three layers 3×3 convolution network. Finally, we combine both the local feature and the non-local feature to generate the output feature D as the input of the decoder.

3.4. Losses

3.4.1 L_2 Loss

At a first glance, the network has a global input-output residual connection whereby the network learns to estimate the noise rather than successively clean the image. This has been shown [41] to improve training convergence for the denoising problem. Following the traditional methods like [41], we adopt the L2-norm loss function which is the mean squared error (MSE) between the denoised patch output by the network and the ground truth.

3.4.2 NLF Loss

We assume that the irradiance of the ground truth, noisy input and predicted clean image as $\widetilde{\mathbf{y}}, \mathbf{x}, \mathbf{y}$. We have the real noise in input :$\mathbf{n}_i = \mathbf{x} - \widetilde{\mathbf{y}}$, with $\mathbf{n}_i \sim \mathcal{N}(\mathbf{0}, \sigma_{\mathbf{x}})$. We also define the noise in output as :$\mathbf{n}_o = \mathbf{y} - \widetilde{\mathbf{y}}$, with $\mathbf{n}_o \sim \mathcal{N}(\mathbf{0}, \sigma_{\mathbf{y}})$. We have $|\sigma_{\mathbf{x}}| \gg |\sigma_{\mathbf{y}}| \approx 0$ owing to that $\mathbf{y}$ is closer to $\widetilde{\mathbf{y}}$ than $\mathbf{x}$. Considering a new variable $\mathbf{n}_v = \mathbf{x} - \mathbf{y}$, it has $\mathbf{n}_v \sim \mathcal{N}(\mathbf{0}, \sigma_{\mathbf{x}} + \sigma_{\mathbf{y}})$. For a given position $k, \mathbf{x}_k, \mathbf{y}_k, \mathbf{n}_{v_k}$ are $C \times 1$ vectors ($C = 1$ for gray level image and 3 for color image), and $\sigma_{\mathbf{x}_k}, \sigma_{\mathbf{y}_k}$ is a $C \times C$ covariance matrix. If we ignore the noise channel correlation, the covariance matrix turns into a diagonal matrix.

Motivated by [38], we take the sum of negative log-likehood of $\mathbf{n}_{v_k}$ as the NLF loss:

$$\mathcal{L}_{\mathrm{NLF}} = \sum_k \left\{ \frac{1}{2} \left(\mathbf{y}_k - \mathbf{x}_k\right)^T \left(\sigma_{\mathbf{x}_k} + \sigma_{\mathbf{y}_k}\right)^{-1} \left(\mathbf{y}_k - \mathbf{x}_k\right) + \frac{1}{2} \log \left|\sigma_{\mathbf{x}_\mathrm{k}} + \sigma_{\mathbf{y}_\mathrm{k}}\right| \right\}, \tag{4}$$

where $|\cdot|$ indicates the determinant of a matrix. We take the approximation $|\sigma_{\mathbf{x}_k}| \gg \left|\sigma_{\mathbf{y}_k}\right| \approx 0$ into account, and the log term can be approximated by its first order Taylor expansion at the point $|\sigma_{\mathbf{x}_k}|$: $\log \left|\sigma_{\mathbf{x}_\mathrm{k}} + \sigma_{\mathbf{y}_\mathrm{k}}\right| \approx \log |\sigma_{\mathbf{x}_\mathrm{k}}| + \mathrm{tr}\left(\left(\sigma_{\mathbf{x}_\mathrm{k}}\right)^{-1} \sigma_{\mathbf{y}_\mathrm{k}}\right)$, where $\mathrm{tr}\left(\cdot\right)$ indicates the trace of a matrix. The irradiance of $\mathbf{x}$ and $\mathbf{y}$ is obtained by using the inverse ISP (Sec.3.6) to convert the noisy input and the predicted clean image. The noise level $\sigma_{\mathbf{x}}$ is provided by the real noisy dataset SIDD and DND. The noise level $\sigma_{\mathbf{y}}$ is estimated by [8]. By citing the conclusion of the [8], the estimated order of magnitude of $\sigma_{\mathbf{y}}$ is accurate. According to the approximation, $\sigma_{\mathbf{y}} + \sigma_{\mathbf{x}} \approx \sigma_{\mathbf{x}}$ and the trace of $\left(\sigma_{\mathbf{x}}\right)^{-1} \sigma_{\mathbf{y}}$ can be well estimated. Thus, the total loss can boost the performance.

Finally, the following NLF loss is employed to supervise the training process:

$$\mathcal{L}_{\mathrm{NLF}} = \sum_k \left\{ \frac{1}{2} \left(\mathbf{y}_k - \mathbf{x}_k\right)^T \left(\sigma_{\mathbf{x}_k} + \sigma_{\mathbf{y}_k}\right)^{-1} \left(\mathbf{y}_k - \mathbf{x}_k\right) + \frac{1}{2} \log |\sigma_{\mathbf{x}_\mathrm{k}}| + \frac{1}{2} \mathrm{tr}\left(\left(\sigma_{\mathbf{x}_\mathrm{k}}\right)^{-1} \sigma_{\mathbf{y}_\mathrm{k}}\right) \right\}. \tag{5}$$

The total loss can be written as:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_2 + \lambda \mathcal{L}_{\mathrm{NLF}}. \tag{6}$$

3.5. Network Structure

The encoder consists of 16 residual blocks which are proposed by [24]. The decoder consists of 16 residual blocks. The Cross-Patch GCN is inserted after the encoder and is a quite small network only containing three convolutional layers for both efficiency and accuracy.

3.6. Data Augmentation

Inspired by [16, 1], as shown in Figure 4, we propose a method to synthesize the realistic sRGB noisy images from the clean images. It consists of three steps: transforming a clean sRGB image to the raw space (inverse ISP), adding Poisson-Gaussian noise to the raw image, and converting the noisy raw image back to the sRGB space (ISP). The ISP pipeline in order consists of white balancing, Bayer rearrangement, DC-Net, color space conversion, gamma transform, and tone mapping. In the inverse ISP pipeline, the input is an sRGB image and the output is a simulated raw image which in order goes through inverse tone mapping, inverse gamma transformation, inverse color space conversion, inverse demosaicing, inverse white balancing. A network called DC-Net has been designed to learn the processing of the non-invertible components of ISP including demosaicing.

To generate the realistic sRGB noisy images, we need to train DC-Net in advance. First, we take the real noisy sRGB images as the inverse ISP pipeline input to generate the simulated noisy raw images, then turn off the noise model component, and process the raw images through the ISP pipeline to generate realistic noisy images. The noise of the generated sRGB images shares a similar distribution as the original real noisy images. After the framework is trained, turn on the noise model component, and take a large amount of clean sRGB images as the input to generate their corresponding sRGB noisy images. Then we can leverage these realistic noisy images as augmentation for real image training. More detailed implementations are provided in the supplementary material.

4. Experiment

In this section, we examine the effectiveness of our method for real image denoising.

4.1. Datasets

We mainly conduct experiments on a real-world dataset: SIDD [1] which is currently the most informative dataset captured by smartphone cameras. It releases 160 pairs of clean and noisy images. All the images are with metadata. We randomly select 140 clean sRGB images and the corresponding synthetic noisy images to train a denoising model. All the images are cropped into 128×128 non-overlap training query patches, of which the total number is 110810. The rest 20 real noisy images are cropped into 16210 128×128 non-overlap patches for testing.

For more comparisons with recent methods, we also perform experiments on the DND[34] datasets. This dataset

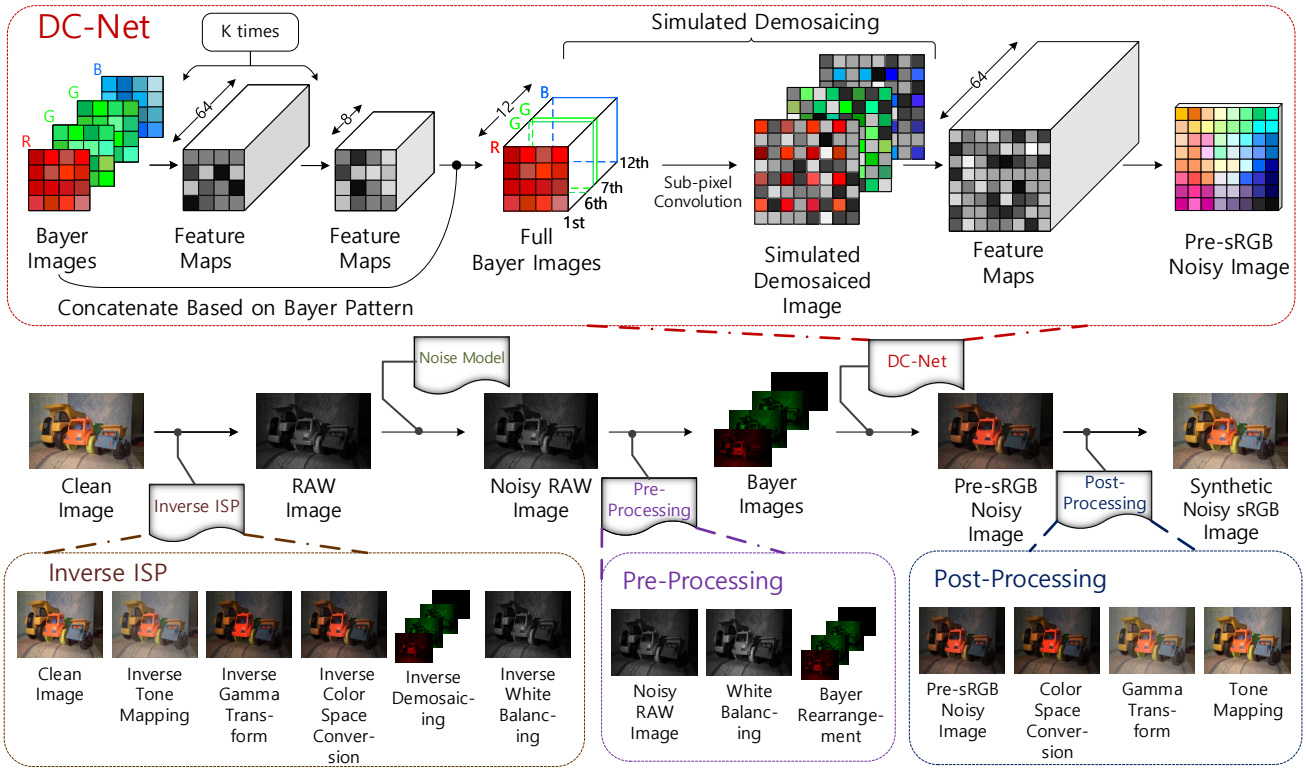

Figure 4. A flowchart illustrating the main steps of realistic noisy image generation in our procedure. A clean sRGB image is passed through the inverse ISP module to get a raw image, and after been added noise, the raw image is pre-processed to obtain the input to the DC-Net, which generates a pre-sRGB noisy image. Finally, the realistic noisy sRGB image is obtained by the post-processing.

consists of 50 pairs of noisy and noise-free images captured with four consumer cameras. Since the images are of very high-resolution, the providers extract 20 crops of size 512 × 512 from each image, thus yielding a total of 1000 patches. The complete dataset is used for testing because the ground truth noise-free images are not publicly available. Quantitative evaluation in terms of PSNR and SSIM can only be performed through an online server.

We adopt DIV2K[2] as the augmentation of the real noisy training dataset. DIV2K has 800 (relatively) clean training images without the paired noisy ones with high quality (2K resolution). We randomly cropped 50 non-overlap 128×128 patches in each image thus yielding a total of 40000 training query patches.

In the cross-patch sampling, we set the stride s_1 as 200. The number of the neighbors K is set as 3. In the Cross-Patch GCN, we set the stride s_2 as 4. The size d of the GCN feature maps is 3.

4.2. Implementation Details

4.2.1 CPNet Settings

The model is trained for approximately 800000 iterations with a minibatch size of 8. The Adam optimizer[20] has been used with the settings of $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10 - 8$ and an exponentially decaying learning rate between 10^{-4} and 10^{-5}. The framework is implemented on the Pytorch on 8 NVIDIA 1080Ti GPU.

4.2.2 DC-Net Settings

In the training of DC-Net, ADAM [20] is used as the optimizer with a learning rate set to 5×10^{-4}. The batch size is 64. In each epoch, 30000 patches are sampled from the whole training patches. Totally, 1000K epochs are carried out during training. L_2 loss is adopted.

4.2.3 Training Strategies

According to the difference of testing data SIDD and DND, we use two training strategies. For SIDD testing, we adopt SIDD noisy data as the DC-Net training data. The generated DIV2K noisy images have similar noise distribution with SIDD noisy images. Both of the two data sets are regarded as the training data set for CPNet training. For DND testing, we adopt DND noisy data as the DC-Net training data and the generated noisy images share a similar distribution with the DND noisy images. Nevertheless, DND does not provide training pairs so that only DIV2K noisy pairs are taken as the CPNet training input.

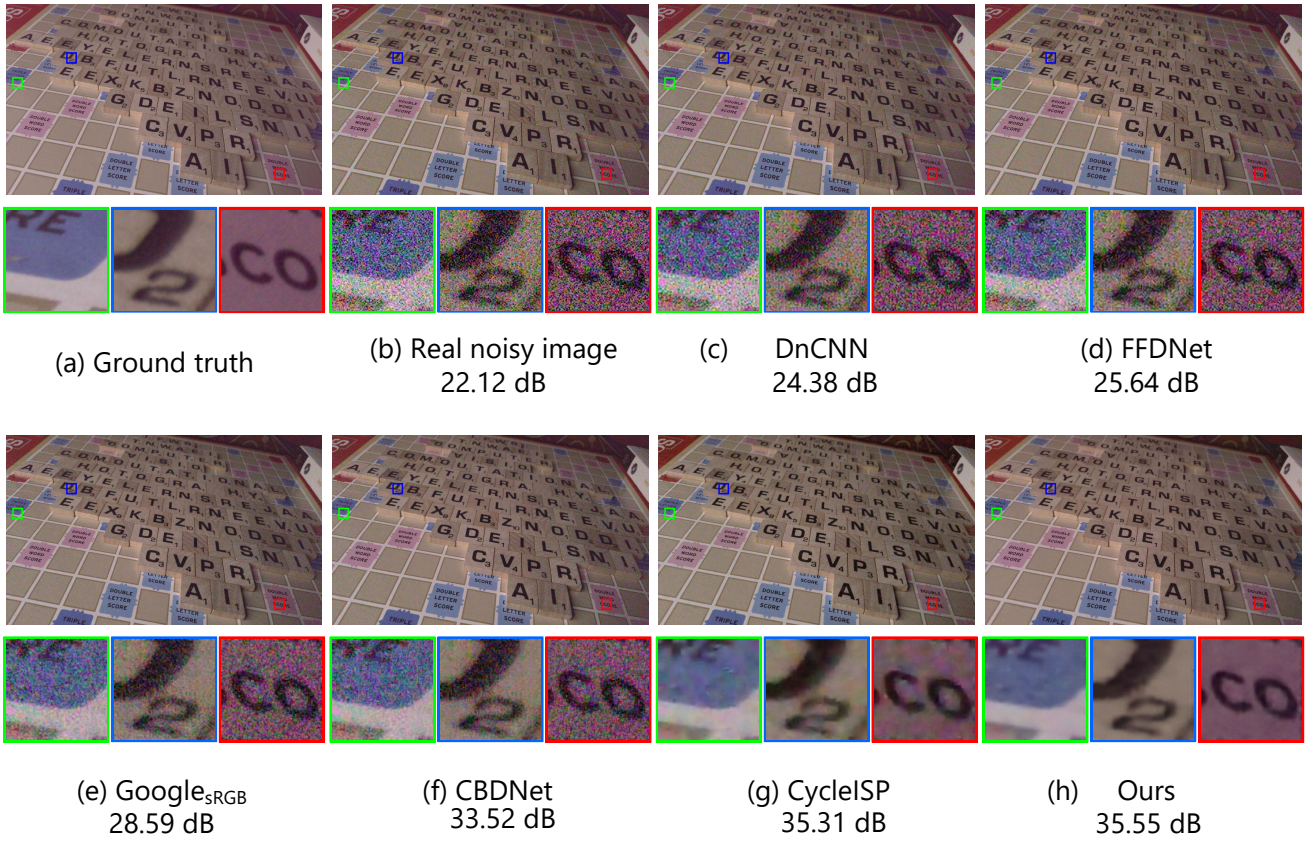

Figure 5. Visual comparison of the six models on the SIDD dataset.

4.3. Denoising Performance on SIDD

We validate the usefulness of our denoising model on SIDD. Five recent models are used to compare with CPNet. The first is DnCNN[41], which is a classical deep-learning-based denoising method. The second model is FFDNet [42], which is trained by using sRGB images with estimated noise level maps, where the noisy images are added with different levels of Gaussian noise. The third model is $\text{Google}_{\text{sRGB}}$ [4], which is trained with synthetic raw images and the loss is imposed in the sRGB space. The fourth model is CBDNet [18], which is trained using sRGB images with their noise level maps generated by a deep network. The fifth model is CycleISP[40], which models the camera imaging pipeline forward and trains a new framework on realistic synthetic data generated by its pipeline. All these models are trained with the SIDD and DIV2K training set.

Table 1 shows the denoising performance comparison among these six models on practical noisy sRGB images. We can see that CPNet performs significantly better than DnCNN, FFDNet, $\text{Google}_{\text{sRGB}}$, and CBDNet in terms of both PSNR and SSIM. It also outperforms CycleISP by 0.21dB on PSNR.

Table 1. Denoising performance of the six models on the 16210 128×128 non-overlap patches from the 20 practical noisy images in SIDD.

Method	PSNR (dB)	SSIM
DnCNN	31.96	0.6970
FFDNet	34.66	0.7781
$\text{Google}_{\text{sRGB}}$	35.66	0.8485
CBDNet	37.08	0.9236
CycleISP	38.13	0.9524
CPNet	**38.34**	**0.9571**

A visual comparison of the six models is given in Figure 5. The original real noisy image is quite dark. For better observation, the intensities of the three R, G, and B channels of the image and the denoising results are stretched by a linear function $y = 2x$. It is easy to see that CPNet removes most noise, especially in the dark regions.

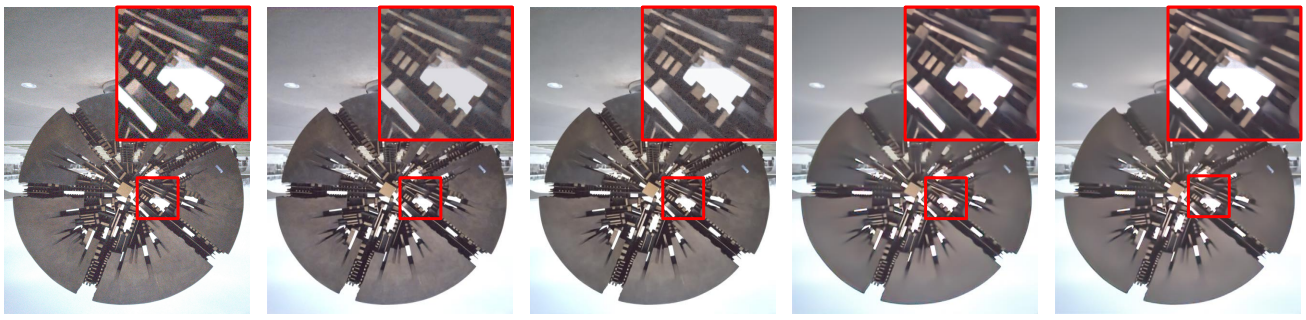

Figure 6. Visual comparison of the five models on the DND dataset.

Table 2. Denoising performance of the five models on the DND dataset.

Method	PSNR (dB)	SSIM
FFDNet	36.88	0.9252
CBDNet	38.06	0.9421
Google$_{\text{sRGB}}$	38.10	0.9436
CycleISP	39.56	0.9564
CPNet	**39.78**	**0.9566**

Table 3. Results on SIDD for variants of CPNet. The (w/o CPGCN), (w/o L_{NLF}), and (w/o data augmentation) denote removing Cross-Patch GCN, removing noise level map loss, and only using SIDD training data set, respectively.

	PSNR (dB)	SSIM
baseline	36.85	0.9103
w/o CPGCN	37.34	0.9311
w/o $\mathcal{L}_{\text{NLF}}$	38.15	0.9531
w/o data augmentation	37.61	0.9382
CPNet	**38.34**	**0.9571**

4.4. Denoising Performance on DND

We also conduct an experiment on another commonly-used real-world dataset DND [34]. Since DND has only 50 real noisy images without their ground truth released, all the models are trained with the 800 clean sRGB images from DIV2K and tested on the DND dataset. As shown in Table 2, our model outperforms the others, even though the training and test datasets are different. A visual denoising comparison is given in Figure 6.

4.5. Ablation Study

We study the impact of various design parameters on denoising performance. To validate the effectiveness of Cross-Patch GCN, the NLF loss, and the data augmentation, we conduct ablation experiments to evaluate the effectiveness of each key component in our proposed method: removing NLF loss (w/o $\mathcal{L}_{\text{NLF}}$) which means the loss is L_2 loss only, removing Cross-Patch GCN (w/o CPGCN) which means only remains the local convolution, and removing data augmentation which means only use SIDD training data set. Table 3 shows that the performance becomes increasingly better as any one of the adaptation components being included. The above experimental results demonstrate that CPGCN enhances the denoising performance and the noise level map can be leveraged to supervise the network training. Besides, we can improve a denoiser by using our method to enlarge the training dataset.

5. Conclusion

We have proposed a graph convolutional network CPNet to explore cross-patch contextual consistency for high-resolution real image denoising. Furthermore, a novel noise level map loss is applied to our model and promotes the denoiser performance. To improve the robustness and flexibility of the real image denoising, we design an effective pipeline to generate realistic sRGB noisy images for enlarging the training dataset and achieve satisfactory results.

Acknowledgments: This work was supported by the National Key R&D Program of China under Grand 2020AAA0105702, the National Natural Science Foundation of China (NSFC) under Grants U19B2038 and 61901433, the University Synergy Innovation Program of Anhui Province under Grant GXXT-2019-025, and the USTC Research Funds of the Double First-Class Initiative under Grant YD2100002003.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018.

[2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, 2017.

[3] Josue Anaya and Adrian Barbu. RENOIR - A benchmark dataset for real noise reduction evaluation. *CoRR*, abs/1409.8230, 2014.

[4] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T. Barron. Unprocessing images for learned raw denoising. In *CVPR*, 2019.

[5] Benoit Brummer and Christophe De Vleeschouwer. Natural image noise dataset. In *CVPRW*, 2019.

[6] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *CVPR*, 2005.

[7] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Real-world image denoising with deep boosting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(12):3071–3087, 2020.

[8] Guangyong Chen, Fengyuan Zhu, and Pheng-Ann Heng. An efficient statistical method for image noise level estimation. In *ICCV*, 2015.

[9] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1256–1272, 2017.

[10] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen O. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Processing*, 16(8):2080–2095, 2007.

[11] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *IEEE Trans. Image Processing*, 22(4):1620–1630, 2013.

[12] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.*, 15(12):3736–3745, 2006.

[13] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen O. Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Trans. Image Processing*, 17(10):1737–1754, 2008.

[14] Xueyang Fu, Qi Qi, Zheng-Jun Zha, Xinghao Ding, Feng Wu, and John W. Paisley. Successive graph convolutional network for image de-raining. *Int. J. Comput. Vis.*, 129(5):1691–1711, 2021.

[15] Xueyang Fu, Qi Qi, Zheng-Jun Zha, Yurui Zhu, and Xinghao Ding. Rain streak removal via dual graph convolutional network. In *AAAI*, 2021.

[16] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Trans. Graph.*, 35(6):191:1–191:12, 2016.

[17] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, 2014.

[18] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *CVPR*, 2019.

[19] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *CoRR*, abs/1506.05163, 2015.

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[21] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[22] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In *ICML*, 2018.

[23] Stan Z. Li. *Markov Random Field Modeling in Image Analysis*. Advances in Pattern Recognition. Springer, 2009.

[24] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017.

[25] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S. Huang. Non-local recurrent network for image restoration. In *NeurIPS*, 2018.

[26] Ding Liu, Bihan Wen, Xianming Liu, Zhangyang Wang, and Thomas S. Huang. When image denoising meets high-level vision tasks: A deep learning approach. In *IJCAI*, 2018.

[27] Daqing Liu, Hanwang Zhang, Zheng-Jun Zha, and Feng Wu. Learning to assemble neural module tree networks for visual grounding. In *ICCV*, 2019.

[28] Jiawei Liu, Zheng-Jun Zha, Xuejin Chen, Zilei Wang, and Yongdong Zhang. Dense 3d-convolutional neural network for person re-identification in videos. *ACM Trans. Multim. Comput. Commun. Appl.*, 15(1s):8:1–8:19, 2019.

[29] Jan Lukás, Jessica J. Fridrich, and Miroslav Goljan. Digital camera identification from sensor pattern noise. *IEEE Trans. Inf. Forensics Secur.*, 1(2):205–214, 2006.

[30] Henrique S. Malvar, Li-wei He, and Ross Cutler. High-quality linear interpolation for demosaicing of bayer-patterned color images. In *ICASSP*, 2004.

[31] Seonghyeon Nam, Youngbae Hwang, Yasuyuki Matsushita, and Seon Joo Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. In *CVPR*, 2016.

[32] Xuejing Niu, Bo Yan, Weimin Tan, and Junyi Wang. Effective image restoration for semantic segmentation. *Neurocomputing*, 374:100–108, 2020.

[33] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.

[34] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017.

[35] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *NeurIPS*, 2018.

[36] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *CVPR*, 2017.

[37] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Deep graph-convolutional image denoising. *IEEE Trans. Image Process.*, 29:8226–8237, 2020.

[38] Xiaohe Wu, Ming Liu, Yue Cao, Dongwei Ren, and Wangmeng Zuo. Unpaired learning of deep image denoising. In *ECCV*, 2020.

[39] Jun Xu, Hui Li, Zhetong Liang, David Zhang, and Lei Zhang. Real-world noisy image denoising: A new benchmark. *CoRR*, abs/1804.02603, 2018.

[40] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis.

[41] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Processing*, 26(7):3142–3155, 2017.

[42] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Trans. Image Processing*, 27(9):4608–4622, 2018.

[43] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. Object relational graph with teacher-recommended learning for video captioning. In *CVPR*, 2020.

[44] Zheng-Jun, Zha, Daqing, Liu, Hanwang, Zhang, Yongdong, Feng, and Wu. Context-aware visual policy network for fine-grained image captioning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[45] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, and Chen Change Loy. Cross-scale internal graph neural network for image super-resolution. In *NeurIPS*, 2020.

[46] Maria Zontak and Michal Irani. Internal statistics of a single natural image. In *CVPR*, 2011.