High ISO JPEG Image Denoising by Deep Fusion of Collaborative and Convolutional Filtering

Huanjing Yue, *Member, IEEE*, Jianjun Liu, Jingyu Yang, *Senior Member, IEEE*, Truong Q. Nguyen, *Fellow, IEEE*, and Feng Wu, *Fellow, IEEE*

***Abstract*— Capturing images at high ISO modes will introduce much realistic noise, which is difficult to be removed by traditional denoising methods. In this paper, we propose a novel denoising method for high ISO JPEG images via deep fusion of collaborative and convolutional filtering. Collaborative filtering explores the non-local similarity of natural images, while convolutional filtering takes advantage of the large capacity of convolutional neural networks (CNNs) to infer noise from noisy images. We observe that the noise variance map of a high ISO JPEG image is spatial-dependent and has a Bayer-like pattern. Therefore, we introduce the Bayer pattern prior in our noise estimation and collaborative filtering stages. Since collaborative filtering is good at recovering repeatable structures and convolutional filtering is good at recovering irregular patterns and removing noise in flat regions, we propose to fuse the strengths of the two methods via deep CNN. The experimental results demonstrate that our method outperforms the state-of-the-art realistic noise removal methods for a wide variety of testing images in both subjective and objective measurements. In addition, we construct a dataset with noisy and clean image pairs for high ISO JPEG images to facilitate research on this topic.**

***Index Terms*— Realistic noise, Bayer pattern, convolutional neural network, collaborative filtering, high ISO images.**

I. Introduction

CAPTURING images under a high ISO (ISO is short for International Standards Organization, which also standardizes the sensitivity ratings for camera sensors) mode enables us to capture fast motion objects, record the details in dark scenes, and avoid blur artifacts when taking images without tripods. Consequently, many cameras with a large ISO range have been developed, such as Nikon D810 (ISO: 64-12800), Canon EOS 5D Mark IV (ISO: 100-32000), Panasonic gx8 (ISO: 100-25600). However, the read noise and shot noise will increase at high ISO conditions. Due to the highly nonlinear imaging process introduced by demosaicing, white balance, tone mapping, JPEG compression etc., the statistic property of this kind of noise (named realistic noise hereafter) is more complex than that of Gaussian noise [1]. Therefore, directly utilizing the well developed Gaussian noise removal method to blindly handle this kind of noise cannot produce satisfactory results.

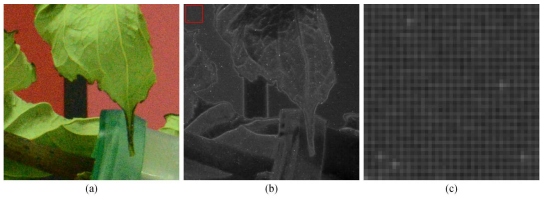

Fig. 1. From left to right: (a) the input noisy image, (b) the noise variance map (for the red channel) of (a), and (c) the close-up of (b).

Blind image denoising usually involves two stages, i.e. noise estimation and denoising with the estimated noise level. The models used for noise level estimation can be categorized as point, curve (line) and map models. The point model corresponds to the homogeneous white Gaussian noise, which can be described by the noise variance [2]. The line model corresponds to the Poisson-Gaussian noise, whose noise level depends on the image intensity. However, this model is only suitable for raw images or linear imaging system. To cope with the non-linear imaging pipeline, the works in [3], [4] proposed to utilize a noise level function (NLF) to model the relationship between noise level and image intensity. Nevertheless, this is still not suitable to model the noise in high ISO JPEG images. Recently, Nam *et. al* proposed to utilize noise variance map to describe the noise level of JPEG images, which demonstrates that the noise is not only content-correlated but also channel-correlated [1]. Fig. 1 (b) presents the noise variance map for the image shown in Fig. 1 (a) (captured by Nikon D800 at ISO 6400). In addition to the intensity-dependent characteristic, we observe that the noise is also spatial correlated due to the interpolation in demosaicing from sensor outputs. Specifically, noise in the observed samples is spread into the interpolated ones, which makes the noise variance map has a pattern similar to the Bayer color filter array as shown in Fig. 1 (c). Therefore, in this paper we propose a novel noise estimation method by taking advantage of the Bayer pattern prior.

Manuscript received November 12, 2017; revised December 17, 2018; accepted March 27, 2019. Date of publication April 9, 2019; date of current version July 1, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61672378, Grant 61771339, Grant 61425026, and Grant 61520106002, and in part by the China Scholarship Council. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Gang Hua. *(Corresponding author: Jingyu Yang.)*

H. Yue, J. Liu, and J. Yang are with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: dayueer@tju.edu.cn; LJJ_ZD@tju.edu.cn; yjy@tju.edu.cn).

T. Q. Nguyen is with the Electrical and Computer Engineering Department, University of California at San Diego, San Diego, CA 92093 USA (e-mail: tqn001@eng.ucsd.edu).

F. Wu is with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China (e-mail: fengwu@ustc.edu.cn).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TIP.2019.2909805

1057-7149 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

For image denoising, many methods have been proposed to remove the white Gaussian noise, including non-local similartiy (internal-correlation) based methods [5]–[8], learning (external-correlation) based [9]–[11], and the combination of internal and external denoising [12]–[14]. With the rapid development of deep learning in vision-based research, the convolutional neural network (CNN) based denoising methods have emerged and achieved promising results [15]–[18]. Compared with Gaussian noise removal, only a few researches on blind realistic noise removal exist and they mainly focus on internal-correlation [19]–[22]. Recently, Xu *et al.* proposed to denoise real noisy images by utilizing external priors to guide internal prior learning [23], which demonstrates that the external-correlation is important for realistic noise removal.

Based on the above observations, we propose to remove the realistic noise introduced in high ISO JPEG imaging by deep fusion of external and internal denoising.[1] For external denoising, we utilize CNN to learn the mapping function between noisy image and the residual noise. For simplicity, we denote the CNN based denoising process as convolutional filtering in the following. For internal denoising, we utilize collaborative filtering to explore the non-local similarity in noisy images. There are mainly three contributions in this paper. First, to our best knowledge, we are the first to take advantage of the Bayer pattern of noise variance map in noise estimation and collaborative filtering. Second, we propose to fuse the convolutional and collaborative filtering results by deep CNN learning, which combines the advantages of learning-based denoising and non-local similarity based denoising. Third, we construct a benchmark dataset with noisy and noise-free image pairs for high ISO JPEG images. The diversity of the captured dataset enables us to learn a CNN to remove realistic noise.

The remainder of this paper is organized as follows. Section II gives a brief introduction of related work on noise estimation and realistic noise removal. Sec. III illustrates the framework of the proposed method. The proposed noise estimation and removal method is detailed in Sec. IV and V, respectively. Experimental results and discussions are given in Sec. VI. Sec. VII concludes the paper.

II. Related Work

In this section, we briefly introduce the two main components in blind noise removal, namely noise estimation and noise removal.

A. Noise Estimation

There are many noise estimation methods for white Gaussian noise, such as the block-based methods [24]–[26], PCA-based methods [2], [27], and statistic-based methods [28]. These methods usually take advantage of the flat patches in the target image [24], [26] or utilize the low-rank patches to estimate the noise variance [2], [27]. However, the noise in captured images are usually not white Gaussian noise. Therefore, it is not suitable to model real noise with only one noise variance.

To solve this problem, some methods are proposed to estimate signal dependent noise, including multi-image based and single-image based. The multi-image based methods estimate noise variance by capturing the same scene with the same camera setting for multi-times [29], which is unreasonable in real applications. In contrast, estimating the signal dependent noise variance from a single image is challenging. The work in [30] proposed to estimate both additive and signal dependent band noise of hyperspectral images using the maximum-likelihood method. Alparone *et al.* demonstrated that the homogeneous pixels produce scatter-points along a straight line and the slope of the line is the signal-dependent noise level [31]. Moreover, Abramov *et al.* proposed weighted LMS line fitting model which considers the number of points in clusters, to estimate the noise parameters [32]. The work in [3] estimated the NLF using the bounds derived from segmented smooth regions with Bayesian MAP method. Yang *et al.* proposed to estimate NLF based on its sparse representation [4]. However, NLF is still not suitable to describe the noise statistics since it assumes that there is only one noise level for each intensity.

The noise in real captured images is very complex, which is spatial, frequency, channel, and content correlated. Nam *et al.* proposed a data-driven approach to estimate the content and channel correlated noise variance [1]. Inspired by this work, we also propose to use a noise variance map to represent the noise level in high ISO JPEG images. In addition, we take the Bayer pattern of noise variance into consideration in the noise estimation process.

B. Realistic Noise Removal

Although hundreds of methods have been proposed to deal with Gaussian noise, the methods for realistic noise removal are rare. Rabie proposed a blind statistical framework for blind denoising, which models the noise as outliers [33]. Yang *et al.* proposed to remove CCD noise utilizing adaptive BM3D, whose parameters are adjusted according to the estimated noise level [4]. Lebrun *et al.* modeled the noise as spatial, frequency and scale dependent. They first estimated the noise covariance matrices in each scale, and then denoised the image using nonlocal-Bayes method with the estimated covariance as parameters [19], [20]. Zhu *et al.* utilized the mixture of Gaussian distribution to model the noise, and utilized a low-rank mixture of Gaussian model to remove the noise [21]. Xu *et al.* proposed a multi-channel optimization model for real color image denoising which introduced a weight matrix to adjust the contributions of R, G, and B channels based on the noise levels [22]. The software Neat Image [34] aimed at removing the noise introduced in high ISO capturing. It first estimated the noise parameters from flat regions and then removed the noise using the estimated parameters. These methods generally first estimate noise levels and then remove noise correspondingly. In contrast, the work in [23] focused on the prior learning rather than noise modeling and achieved

[1]Note that, in this paper we focus on JPEG images since JPEG is the most popular image format used in our daily life.

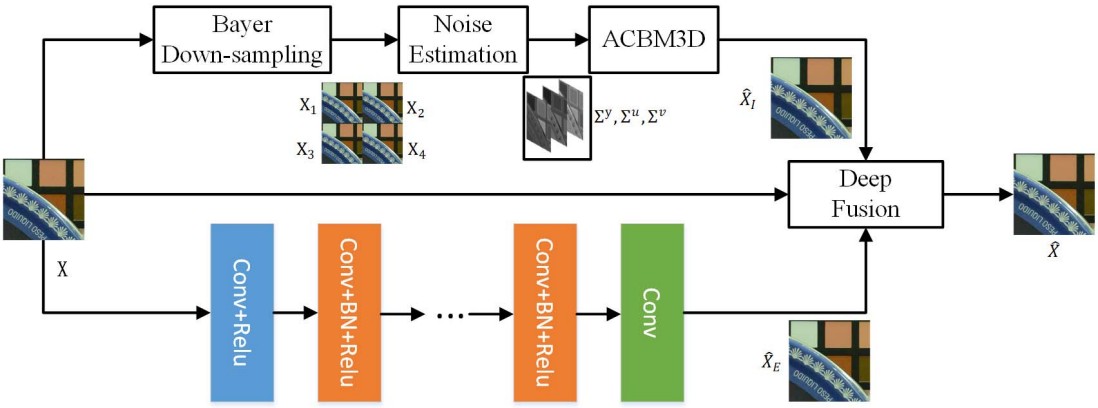

Fig. 2. Framework of the proposed denoising method.

promising denoising results by external prior guided internal prior learning.

Inspired by the work in [23], we propose to remove noise by taking advantages of both external and internal priors. For internal denoising, we follow the collaborative filtering strategy in BM3D, and integrate the estimated noise variance map and the Bayer pattern of noise variance in the filtering process. For external denoising, to fully take advantage of external examples, we propose to utilize convolutional filtering to remove the noise. Finally, we combine the strengths of collaborative and convolutional filtering via deep CNN fusion.

This work extends our previous work [35] in three aspects. First, we give a more detailed illustration of the Bayer prior based noise estimation and collaborative filtering, and analyze the frequency distribution of realistic noise. Second, we propose to fuse the convolutional and collaborative filtering results together via a deep CNN. Third, we construct a dataset with noisy and clean image pairs for high ISO JPEG images to facilitate research on this topic.

III. Framework Overview

Fig. 2 presents the framework of the proposed noise removal method. Given a noisy image X, we first estimate its noise variance map Σ according to the non-local similarity of image content. Then, in the collaborative filtering stage, we extend color BM3D (CBM3D) [6] to adaptive CBM3D (ACBM3D), which is able to handle realistic noise by integrating the estimated Σ and Bayer-pattern down-sampling into the denoising process. In the convolutional filtering stage, we utilize CNN to learn the mapping function from the noisy image to the latent noise. We observe that the collaborative and convolutional filtering results are complementary to each other. Therefore, utilizing deep CNN to fuse the two results together generates the final denoising results $\hat{X}$. Each module will be discussed in detail in the following sections.

IV. Noise Level Estimation

As introduced in Sec. II A, many noise estimation methods are smooth-block-based, which assume that the variance of flat regions are introduced by noise. Since the flat regions are not always available and the noise variance in texture regions

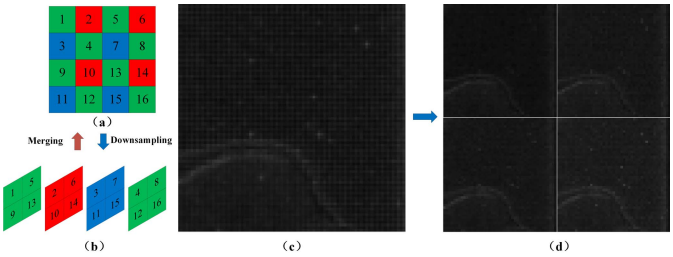

Fig. 3. Illustration of the Bayer downsampling process. From left to right: (a) Bayer pattern, (b) the downsampled version of (a), (c) the noise variance map of one block, and (d) the four Bayer-downsampled blocks of (c).

are different from that in flat regions, we propose to estimate noise variance by taking advantage of the non-local similarity of image blocks.

However, we observe that the noise variances of neighboring pixels are not similar, as shown in Fig. 3 (c). This is due to the color filter array (CFA). CFA is used in single-chip digital image sensors to create color images and is mostly Bayer filter. The filter pattern is 50% green, 25% red and 25% blue, as shown in Fig. 3 (a). Therefore, the color image produced by cameras is reconstructed by interpolating missing color information, i.e. demosaicing interpolation, in each channel. Since interpolating will reduce the noise variance, the noise variances of neighboring pixels in each channel are not similar, as shown in Fig. 3 (c). Therefore, we propose to downsample the images in each channel into sub images according to the Bayer pattern. As shown in Fig. 3 (a) and (b), the input is downsampled into four sub-images. In this way, the pixels in one sub-image are either all interpolated or all captured by the camera sensor. As shown in Fig. 3 (c) and (d), the noise variances of neighboring pixels are not similar in the original block, but similar in Bayer-down-sampled block. Therefore, we split X into four sub-images $\{X_1, X_2, X_3, X_4\}$ according to the Bayer pattern and estimate their noise variance maps separately. Since the noise estimation process is the same for the four sub-images, in the following we use X_i to denote the sub-image.

We observe that the mean noise level of JPEG images captured at high ISO mode depends on the ISO value.

Therefore, we propose to remove the noise using CBM3D with a coarse noise variance $\bar{\sigma}^2$, which is defined as $\bar{\sigma} = 3\log_2 \frac{\text{ISO}}{100}$. Generally, $\bar{\sigma}$ is much larger than the true noise level in order to remove the noise thoroughly. We denote the coarsely denoised version of X_i as $\bar{X}_i$. Compared with X_i, the noise in $\bar{X}_i$ is greatly reduced. Then, we split $\bar{X}_i$ into $m \times m$ blocks (m is set to 4 in this paper). For each block $\bar{B}$ in $\bar{X}_i$, we search for its most similar blocks in a local region of $\bar{X}_i$ centered at $\bar{B}$. We observe that the noise variance in the Y channel is smaller than that in R, G, and B channels. Therefore, we first transform the input noisy RGB image into YUV space using the transformation matrix A, defined as

$$A = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix}. \quad (1)$$

Then, we search similar blocks according to the Euclidean distance in the Y channel.

Suppose we retrieve n similar blocks for each query block $\bar{B}$, denoted as $\{\bar{B}_1, \bar{B}_2, \ldots, \bar{B}_n\}$. n is set to 16 in this paper. Their corresponding noisy blocks are $\{B_1, B_2, \ldots, B_n\}$. The residual block $\mathcal{D}_j = B_j - \bar{B}_j$ contains noise and some high frequency details. Therefore, it is not suitable to use the variance of $\mathcal{D}_j$ as the estimated variance of noise. Meanwhile, $\{\mathcal{D}_j(k)\}_{j=1}^n$ (k is the pixel index) have similar content and their noise variances are also similar to each other. Therefore, the noise variance for the k^{th} pixel in block B can be calculated as

$$\sigma_{B(k)}^{c\,2} = \text{var}\{\mathcal{D}_1^c(k), \mathcal{D}_2^c(k), \ldots, \mathcal{D}_n^c(k)\}, \quad c = y, u, v, \quad (2)$$

where c is the channel index and var$\{\Omega\}$ represents calculating the variance of the set Ω. In this way, we obtain the pixel-wise noise variance σ_B for the whole block B. After merging them together via averaging, we obtain the noise variance map Σ_i^y, Σ_i^u, and Σ_i^v for sub-image X_i. Finally, Σ_i^c ($i = 1, 2, 3, 4$) is mapped to its original position in X^c according to the Bayer pattern, as denoted by the merging operator between Fig. 3 (a) and (b), producing the final noise variance map Σ^c.

V. Noise Removal

In this section, we present our denoising algorithm with the estimated noise variance map as input, including collaborative filtering, convolutional filtering and the deep fusion of the two filtering results.

A. Collaborative Filtering Based Denoising

We observe that many realistic noise removal methods are built on Gaussian noise removal methods. Recently, the work in [36] demonstrates that BM3D is the most effective method for removing realistic noise compared with other Gaussian noise removal methods. Therefore, in this paper, instead of developing the denoising algorithm from scratch, we follow the collaborative filtering strategy of BM3D, and incorporate the estimated noise variance map into the denoising process to handle realistic noise.

A straight forward way is to adaptively change the noise variance for each block in CBM3D. However, as shown in Fig. 1, the noise variances in one block are not uniform, and thus not reasonable to process the whole block using the same variance. Hence, similar to the strategy in noise estimation, we down-sample the noisy input X into four sub-images and remove their noise separately in the first stage of CBM3D. For each block B in image X_i, we search for its similar blocks in a local region centered at B according to the Euclidean distance in the Y channel. The retrieved similar blocks form a 3D cube, denoted as B_{3D}. The noise in B_{3D} is removed by hard-thresholding of the transformed coefficients, namely

$$\tilde{B}_{3D}^c = \mathcal{T}_{3D}^{-1}(\gamma(\mathcal{T}_{3D}(B_{3D}^c))), \quad c = y, u, v, \quad (3)$$

where $\mathcal{T}_{3D}$ is a 3D transform, including a 2D wavelet transform and a 1D Hadamard transform along the third dimension. $\gamma(\cdot)$ is a hard thresholding operator with threshold $\delta^c \lambda^c \bar{\sigma}_B^c$, where $\bar{\sigma}_B^c$ is the mean noise variance of block B, δ^c and λ^c are constant coefficients. After the inverse 3D transform $\mathcal{T}_{3D}^{-1}$, we obtain the initial denoised blocks $\tilde{B}_{3D}^c$. After generating the initial denoising result for each block, these denoised blocks are merged together via weighted averaging, producing the initial denoising result $\tilde{X}_i$. Finally, the four denoised images $\{\tilde{X}_1, \tilde{X}_2, \tilde{X}_3, \tilde{X}_4\}$ are fused together according to the Bayer-pattern, generating an initial denoising image $\tilde{X}$, which cannot only improve patch matching accuracy but also provide guidance for filtering in the following stage.

We would like to point out that although the noise in $\tilde{X}$ is greatly reduced compared with X, the smoothness of neighboring pixels is destroyed due to the proposed sub-image based denoising strategy. Therefore, in the second stage of CBM3D, we propose to denoise image X as a whole with image $\tilde{X}$ as guidance. To avoid ambiguity, instead of using B, we use P to denote the block in the whole image. For each block $\tilde{P}$ in image $\tilde{X}$, we retrieve its similar blocks according to the Euclidean distance in the Y channel, and these similar blocks form a 3D cube $\tilde{P}_{3D}$. Correspondingly, we build another 3D cube P_{3D} from image X using the patch index of $\tilde{P}_{3D}$. Then the final denoised blocks $\hat{P}_{3D}$ is produced by

$$\hat{P}_{3D}^c = \mathcal{T}_{3D}^{wie^{-1}}(W^c \odot (\mathcal{T}_{3D}^{wie}(P_{3D}^c))), \quad c = y, u, v, \quad (4)$$

where $\mathcal{T}_{3D}^{wie}$ is a 3D transform, consisted by a 2D DCT transform and a 1D Hadamard transform. $\odot$ represents element-wise multiplication operation. W^c is a Wiener weighting parameter for coefficients thresholding, and is calculated as

$$W^c = \frac{|\mathcal{T}_{3D}^{wie}(\tilde{P}_{3D}^c)|^2}{|\mathcal{T}_{3D}^{wie}(\tilde{P}_{3D}^c)|^2 + (\beta^c \bar{\sigma}_P^c)^2}, \quad c = y, u, v, \quad (5)$$

where $\bar{\sigma}_P^c$ is the mean noise deviance of block P^c.

Fig. 4 presents the noise standard deviation distribution in frequency domain for Gaussian noise and realistic noise in the Y channel. We select three blocks, i. e. P_1, P_2, and P_3, with different contents from one noisy image captured by Nikon D800 at ISO 6400 to show their noise standard deviations. For one patch, we have 100 noisy samples since we captured the same scene for 100 times. Then, we apply the 8×8 DCT transform on the 100 noisy patches and their

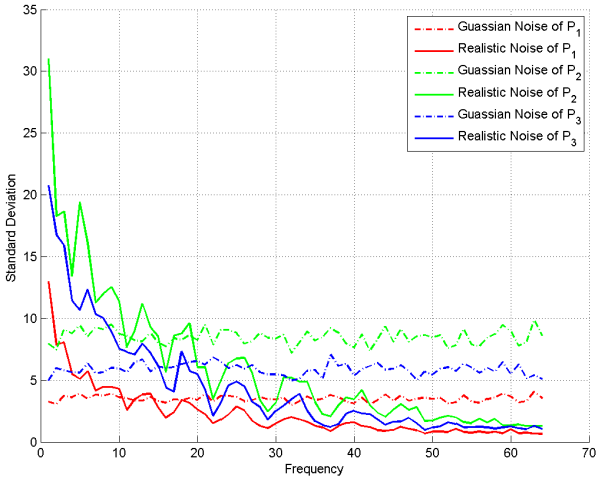

Fig. 4. Comparison of noise deviation distribution in frequency domain (zig-zag scanning order) for Gaussian noise and realistic noise.

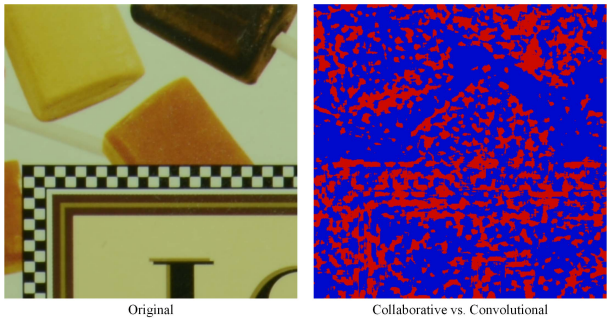

Fig. 5. Preference of denoising patches by convolutional or collaborative filtering. The original noisy image is captured by Nikon D800 at ISO 3200. Red marks collaborative filtering preference and blue marks convolutional filtering preference.

mean patch, respectively. Hereafter, we calculate the deviation for each frequency using the frequency residual between the 100 noisy patches and the mean patch. For Gaussian noise, we add white Gaussian noise 100 times to the mean patch with the mean deviation of its realistic noisy version, and compute the deviation for each frequency using the same method. It can be observed that the deviation distribution for Gaussian noise in frequency domain is nearly uniform.[2] However, for realistic noise, the noise variance in low-frequency band is much higher than that in high-frequency band for each block P. Consequently, we introduce the parameter β^c, as shown in Eq. 5, to adjust the Wiener thresholds so as to remove the noise in low-frequency band thoroughly. We first tried to adjust β^c for different frequencies according to the curve shown in Fig. 4, but the result is not good. This is due to the fact that the characteristic of realistic noise is very complex. It cannot be modeled well using only the frequency distribution. Therefore, we used a fixed β^c, which is larger than 1, for each block.

After obtaining the denoised blocks $\hat{P}$ for each block P, we merge them together via weighted averaging, generating the final internal denoising image $\hat{X}_I$.

B. Convolutional Filtering Based Denoising

The collaborative filtering method only takes advantage of the non-local similarity of the noisy image itself. In the past decades, many image denoising methods have benefited from the external priors learned from clean images. Recently, the CNN has shown its great potential in removing Gaussian noise. In this paper, we extend CNN to process realistic noise.

Note that the work in [23] demonstrates that the trained Blind-DnCnn [16] for arbitrary noise level cannot handle realistic noise. The reason is that realistic noise is not a combination of Gaussian noise with different variances. The statistics of realistic noise is more complex than that of Gaussian noise in both frequency and spatial domains. Its complexity motivates us to adopt convolutional filtering, which has a high capacity to infer the noise.

1) Network Architecture: Recent developments of CNN in image super-resolution and denoising have demonstrated that the residual learning can improve the learning accuracy and accelerate the learning speed [16], [37]. In addition, to reduce the internal covariance shift, batch normalization is usually adopted to stabilize the learning process. For the activation function, rectified linear units (ReLu) is widely used in recent years since it can accelerate the convergence of stochastic gradient descent and its computing complexity is much lower compared with Sigmoid and Tanh. Therefore, we use the combination of Conv+BatchNorm+ReLu as the main operators in our denoising network. As shown in Fig. 2, the proposed denoising network is similar to that of DnCnn [16]. For the first layer, we utilize k convolutional filters with size $3 \times 3 \times \kappa$, where κ is the channel number. This paper processes color images, i. e. κ is 3. Followed by the convolution layer is the Relu layer. The following layers are composed by convolution, batch normalization, and Relu layers. For each convolution layer, there are k convolution filters with size is $3 \times 3 \times k$. The end layer is a $3 \times 3 \times k$ convolution filter, which reconstructs the residual noise. In this paper, k is set to 64. Using thin (e.g. 64 channels) and small filter size (e.g. 3×3) in deep network is an effective strategy for improving performance with reasonable number of parameters [16], [37]–[39].

2) Training Details: Generally, the peak signal noise ratio (PSNR) is utilized to evaluate the denoising performance. Therefore, we utilize the mean square error as the loss function to learn the mapping parameters. We denote the clean image corresponding to the noisy input X as Y. Then, the loss function is defined as

$$\underset{\mathcal{W}}{\operatorname{argmin}} \frac{1}{2N} \sum_{j=1}^{N} \|\mathcal{F}(X_j; \mathcal{W}) - (X_j - Y_j)\|_F^2, \tag{6}$$

where $\{X_j, Y_j\}$ are the noisy-clean training pairs, $\mathcal{F}(X_j; \mathcal{W})$ is the learned operators to map X to the residual noise, $\mathcal{W}$ is the parameters to be learned, and N is the number of training pairs in each batch.

[2]Since we only simulate the noise for 100 times, there is fluctuation in Gaussian noise variance curve. Ideally, the white Gaussian noise variance should be constant in frequency domain.

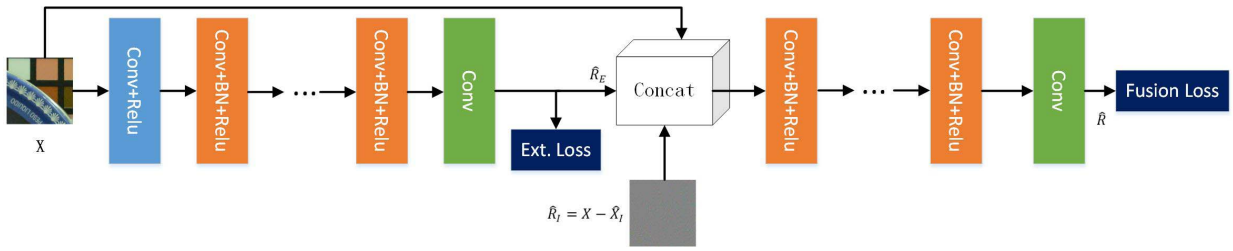

Fig. 6. The end to end training of the convolutional filtering and deep fusion network.

In the training process, we extract noisy-clean patch pairs of size 40×40 from the noisy input and its corresponding noise-free image. To reduce the boundary artifacts caused by convolution, we utilize zero padding before applying convolution. We utilize adaptive moment estimation (Adam) method with mini-batch size of 128 for training. The learning rate is initialized as 1e-4 and the training process converges in 20 epochs. Since the noise cannot be simulated online, the diversity of training samples is reduced to some extent. Therefore, data augmentation is important to enrich the training dataset. We augment the training pairs via scaling, rotation and flipping.

C. Deep Fusion

Fig. 5 presents the preference of collaborative filtering and convolutional filtering results. For each patch (8×8), if the collaborative filtering result is better than convolutional filtering result, it is marked in red. Otherwise, this patch is marked in blue. We observe that the collaborative filtering is good at keeping the repetitive edges, while the convolutional filtering is good at removing noise in flat regions. In other words, the collaborative and convolutional filtering based denoising results are complementary to each other. This inspires us to combine the two results together.

In the literature, there are some combined denoising methods proposed for Gaussian noise removal, e.g. weighted average in spatial domain [12] or frequency domain [14], and using neural network to learn the weighting parameters [13]. The work in [13] utilizes a fully connected neural network, which takes one month of training time. In contrast, we propose to utilize deep CNN to train the fusion filters. In addition, we utilize residual learning which learns the residual noise rather than the clean image.

The structure of proposed fusion network is the same as that of external denoising network, except for the first convolution layer. The first convolution layer is composed by 64 filters with size of $3 \times 3 \times 9$. The channel number is 9 since the input is three color images, i.e. the noisy input X, the noise residual generated in collaborative filtering $\hat{R}_I = X - \hat{X}_I$, and the residual generated by convolutional filtering $\hat{R}_E = \mathcal{F}(X; \mathcal{W})$. The training parameter settings, i.e. batch size and patch size, are the same with those of external denoising network.

We would like to point out that the proposed convolutional filtering network and deep fusion network can be trained in an end-to-end fashion. As shown in Fig. 6, the two networks are connected by a concatenation layer. There are two loss functions in the connected network, i.e. the noise residual loss for convolutional filtering, named as external loss, and the final noise residual loss after deep fusion, named as fusion loss. Therefore, the total loss during training is

$$\mathcal{L}(\hat{R}, R; \mathcal{W}) = \frac{1}{2N} \sum_{j=1}^{N} (\|\hat{R}_{E_j} - R_j\|_F^2 + \|\hat{R}_j - R_j\|_F^2), \quad (7)$$

where R_j is the ground truth residual, $\hat{R}_{E_j}$ is the output of the external denoising network, and $\hat{R}_j$ is the final output.

In the experiments, we train the two networks together since this can save lots of training data compared with training the two networks separately.

VI. Experiments

A. Dataset and Parameter Settings

One limit in realistic noise removal is the lack of training data. In this paper, we construct a dataset with noisy and clean image pairs for high ISO JPEG images. The clean image is obtained by shooting the same scene for 100 times and their average is treated as the noise-free image. This is a common strategy in previous realistic noise removal work to generate noisy-clean pairs [1]. We shot seven groups of images using Canon 60D and Nikon D800 camera at different ISO settings. For each group, we capture four static indoor scenes. Namely, there are totally 28 scenes captured. The details about the constructed dataset is presented in the supplemental material. In addition, we adopt the dataset published in [1], which contains 11 static scenes with noisy-clean image pairs, to enrich our training and testing set. In total, we construct nine groups of training and testing images according to the camera type and ISO settings. Table I presents the detail settings for each group. Each group contains 10-20 testing images with size 512×512, and there are 110 images in total for testing. Fig. 7 (top two rows) presents a few examples of our testing indoor images from the nine groups.[3]

We first train a model for each camera at a specific ISO setting, named as FCCF-s (Fusion of Collaborative and Convolutional Filtering-specific). We extract 128×3100 patches of size 40×40 as the training data for each camera setting. The convolutional layer depth of the external convolutional filtering and deep fusion network is set as 17 and 12, respectively, namely that the convolutional layer depth of the whole network

[3] After publication of this paper, we will release our dataset and code to inspire more works on this topic.

Fig. 7. A few examples of our indoor (top two rows) and outdoor (bottom two rows) testing images.

is 29. We find that using more convolutional layers can slightly boost the denoising performance. Considering the tradeoff between computing complexity and denoising performance, we use the above setting in this paper. To demonstrate the generality and scalability of the proposed method, we train a general network for all test cameras at a variety of ISO settings. In this case, the training data is extracted from all the training scenes. The trained model is named as FCCF-b (FCCF-blind). In the following experiments, we will compare their performance.

For the parameters of collaborative filtering, we set δ^c in Eq. (3) to 2.7. The thresholding parameters λ^c and β^c in Eq. (3) and (5) depend on the camera setting. Generally, $\lambda^y \in [1, 1.7]$, $\lambda^u \in [2.5, 3.5]$, $\lambda^v \in [2.5, 5]$, and $\beta^y \in [2, 3.7]$, $\beta^u \in [4, 8]$, $\beta^v \in [6, 8]$. λ^c and β^c are fixed for each camera setting, and their settings are detailed in the supplemental material.

B. Analysis of the Proposed Method

In this subsection, we demonstrate the effectiveness of the proposed denoising method via presenting intermediate results. We first present the collaborative (convolutional) filtering based denoising and their deep fusion (FCCF-s) results in both subjective and objective measurements. Then, we present the denoising results generated by the general model FCCF-b trained by all kinds of camera settings. Finally, we compare with convolutional filtering by setting its convolutional layer number to 29, the same as that of our FCCF network.

1) Intermediate Results: Fig. 8 presents the intermediate visual results of two images cropped from the scene captured by Canon 60D at ISO 6400. To show the details clearly, we show the close-up of the region marked by red boxes for each result. It can be observed that collaborative filtering is good at recovering repetitive structures, such as the stripes on the ornamental band. However, it cannot remove the noise in the flat regions well. For example, the flat regions of the second image in the internal denoising result is still noisy. On the contrary, the convolutional filtering is good at removing the noise in flat regions, but cannot work well for repetitive structures. As expected, the FCCF-s result combines the strength of the two results, achieving the best denoising result.

In addition, we present the objective intermediate results in terms of PSNR and SSIM [40] in Table I. For color images, we average the SSIM values of R, G, and B channels as the final SSIM value. For each group, the PSNR and SSIM results are the average results of all the test images in this group. It can be observed that FCCF-s outperforms collaborative and convolutional filtering results for all the nine groups. On average, FCCF-s outperforms collaborative and convolutional filtering results by 0.86 dB and 0.3 dB, respectively.

2) Comparison of Specific Model and Blind Model: As stated in Section VI-A, we not only train a specific denoising model for each camera setting (i. e. FCCF-s), but also train a general model for all the camera settings (i. e. FCCF-b), which could be utilized for blind denoising of images captured by different cameras. We select 2300 images with size of 512×512 from those used in FCCF-s training to ensure there is no overlap between training and testing contents. In total, we extract 128×3100 patches of size 40×40 as the training data for FCCF-b. Table I compares the results of FCCF-s and FCCF-b for nine groups of testing images. It can be observed that the results of FCCF-b are comparable with that of FCCF-s. It demonstrates the feasibility of training a single model for all the camera settings. Another interesting phenomenon is that some results of FCCF-b even outperforms the results of FCCF-s. The reason is that the training data of FCCF-b is much richer than that of FCCF-s, even if the noise type of FCCF-b is more complex than that of FCCF-s.

3) Comparison with Convolutional Filtering: From Table I we observe that the proposed FCCF-s outperforms convolutional filtering by 0.3 dB. To demonstrate the improvement coming from the fusion of convolutional and collaborative filtering results, rather than the increase of convolutional layers, we compare with convolutional filtering (CF) by setting the number of its convolutional layers to 29, the same as that of the proposed FCCF-s. We compare the results of CF-29 and the proposed FCCF-s using the testing images in Group "8". The average result of CF-29 is 41.39 dB, which is slightly better than the original CF-17 (41.35 dB). However, it is still worse than our result, which is 41.70 dB.

4) Ablation Experiments: To demonstrate the effectiveness of the proposed Bayer-pattern downsampling in image denoising, we present the denoising results by removing the Bayer-pattern downsampling while keeping other configurations constant. We use the images in Group 8 for comparison. The detailed quantitative results are listed in the supplementary file. The average denoising result of collaborative filtering without Bayer prior is 40.02 dB, which is much lower than that of collaborative filtering with Bayer-pattern downsampling (40.82 dB). This is consistent with the following noise estimation results in Table II: noise estimation results without Bayer prior are much lower than those of the proposed Bayer prior based estimation.

We also observe that the PSNR improvement of FCCF-s over the version without Bayer prior is not so significant as the collaborative filtering over its counterpart without Bayer

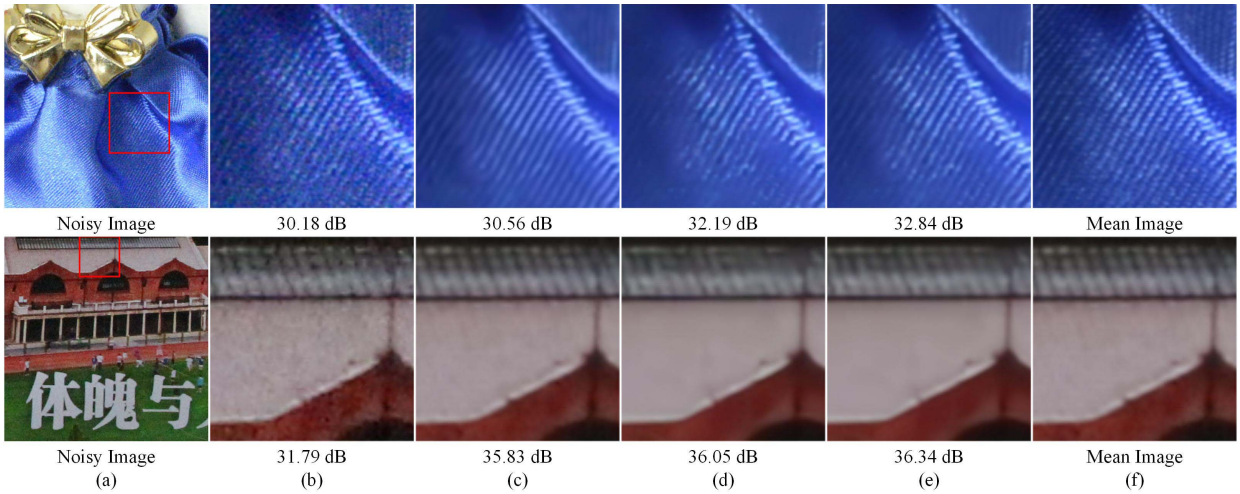

Fig. 8. The intermediate results of two images captured by Nikon D800 (top) and Canon 60D (bottom) at ISO 6400. From left to right: (a) is the original noisy image, (b), (c), (d), and (e) are the highlighted regions cropped from the noisy image, collaborative filtering, convolutional filtering, and FCCF-s denoising results. (f) is the corresponding noise-free image.

TABLE I

COMPARISON OF INTERMEDIATE DENOISING RESULTS IN TERMS OF PSNR AND SSIM VALUES. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Group #	Camera	ISO	Image Num	Collaborative		Convolutional		FCCF-s		FCCF-b	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1	Canon EOS 5D Mark3	3200	10	37.07	0.964	37.81	0.972	**38.00**	**0.973**	37.77	0.971
2	Canon 60D	3200	10	40.42	0.981	40.56	0.982	40.89	**0.983**	**40.99**	**0.983**
3		4000	10	40.48	0.977	40.93	0.979	**41.13**	**0.980**	41.08	0.979
4		5000	10	39.40	0.977	40.14	0.980	**40.36**	**0.981**	40.11	0.980
5		6400	10	37.84	0.962	38.24	0.967	38.62	**0.969**	**38.69**	0.968
6	NiKon D600	3200	10	37.40	0.970	38.31	0.973	38.45	**0.975**	**38.68**	0.974
7	NiKon D800	1600	15	41.37	0.977	41.53	0.980	**42.17**	**0.981**	41.71	0.979
8		3200	20	40.82	0.972	41.35	0.976	41.70	**0.978**	**41.73**	0.977
9		6400	15	35.42	0.935	36.34	0.946	36.62	0.952	**36.80**	**0.953**
Ave.			110	38.91	0.968	39.47	0.973	**39.77**	**0.975**	39.73	0.974

prior. This attributes to the high performance of the proposed fusion network in integrating advantages of the two stream results. As a result, the fused results are always higher than both results before the fusion and the merit of the Bayer prior seems to be absorbed by the fusion module.

C. Noise Estimation Results

To verify the superiority of our noise estimation method, we compare our noise estimation method with [1], [4], which are the most related works to ours. We utilize mean square error (MSE) to measure the accuracy of estimated noise deviation maps. The distance D^c between the estimated noise variance and the ground truth is calculated as

$$D^c = \frac{\|\sqrt{\Sigma^c} - \sqrt{\Sigma^c_{gt}}\|_F^2}{K}, \tag{8}$$

where Σ^c_{gt} and Σ^c are the ground truth and estimated noise variance maps. K is the number of pixels in Σ^c and $\|*\|_F$ represents the Frobenius norm. It is hard to directly compare with [1], [4], since their estimated noise variance is presented in different forms. Therefore, we transform their results to be consistent with ours for the ease of comparison. For [1], we calculate their noise variance in YUV channels from their estimated covariance maps in RGB channels via

$$\begin{aligned} \sigma^c &= \sqrt{\phi_A + \phi_B} \\ \phi_A &= a_{c1}^2\sigma_r^2 + a_{c2}^2\sigma_g^2 + a_{c3}^2\sigma_b^2, \\ \phi_B &= 2a_{c1}a_{c2}\sigma_{rg} + 2a_{c1}a_{c3}\sigma_{rb} + 2a_{c2}a_{c3}\sigma_{gb}, \end{aligned} \tag{9}$$

where σ^c represents the standard deviation in the c channel of one pixel and a_{ci} is the transform coefficient $A(c,i)$ defined in Eq. (1). The estimated noise variance in [4] is presented in NLF, which describes the mapping relationship between intensity and noise variance. So, we map their NLF to a noise variance map according to the denoised intensity of the Y channel. Since the noise levels in U and V channels cannot be represented by the estimated NLF, we only compare with [4] for the Y channel. The results of [1], [4] are generated using the authors' codes.

In addition, to demonstrate the effectiveness of the proposed Bayer prior based noise estimation method, we compare with the noise estimation results generated by removing the Bayer down-sampling process described in Sec. IV.

The noise estimation performance is evaluated on five different camera settings since the results of [1] are only available in these camera settings. We utilize ten test images for each camera setting. Table II presents the mean noise estimation results for each group of images. It can be observed that the

TABLE II

COMPARISON OF THE NOISE ESTIMATION RESULTS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Camera	Image Num	ISO	[4]		[1]			Ours w/o Bayer Prior			Ours		
			D^y	D^{uv}	D^y	D^u	D^v	D^y	D^u	D^v	D^y	D^u	D^v
Canon EOS 5D Mark3	10	3200	6.525	-	20.96	0.575	0.767	7.211	0.593	2.091	**5.867**	**0.140**	**0.569**
Nikon D600	10	3200	0.485	-	0.683	1.852	1.141	1.681	1.191	2.987	**0.468**	**0.255**	**0.764**
Nikon D800	10	1600	1.789	-	3.722	1.116	1.524	2.206	0.746	2.193	**1.456**	**0.183**	**0.574**
	10	3200	0.977	-	1.414	0.937	2.158	1.537	1.551	3.794	**0.405**	**0.370**	**0.961**
	10	6400	2.362	-	6.801	1.319	7.536	5.298	2.901	7.927	**1.250**	**0.760**	**2.401**

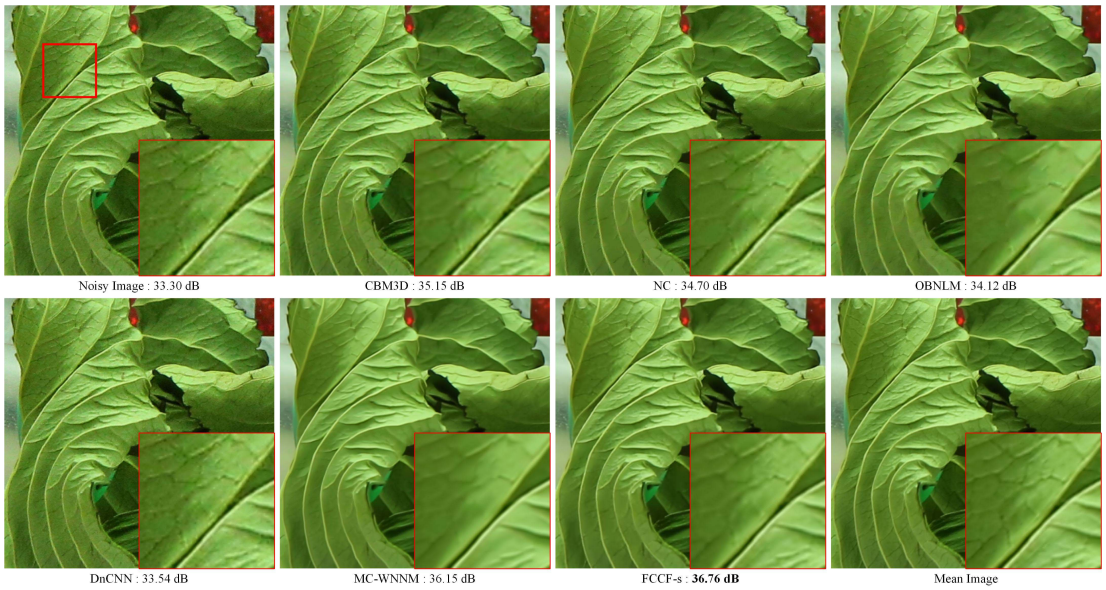

Fig. 9. Comparison of denoising results for an indoor image. The noisy image is captured with the camera setting "Canon 5D Mark 3 ISO 3200".

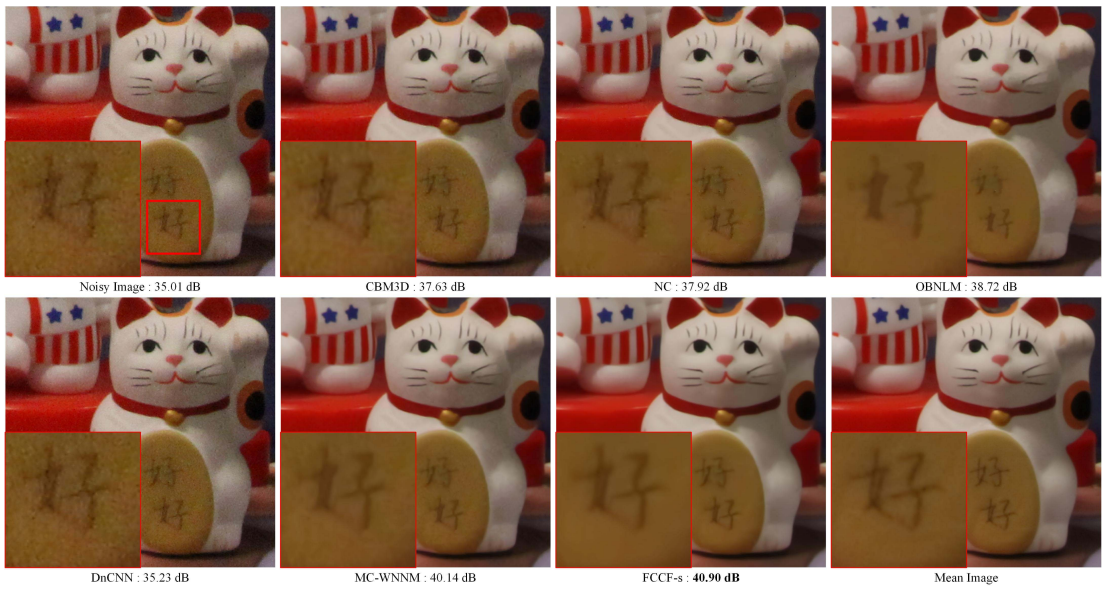

Fig. 10. Comparison of denoising results for an indoor image. The noisy image is captured with the camera setting "Canon 60D ISO 4000".

proposed noise estimation method outperforms [1] and [4] for all the three channels. In addition, our results without Bayer prior is much worse than those with Bayer prior, which further demonstrates the importance of Bayer downsampling in noise estimation.

D. Image Denoising Results

We compare the proposed denoising method with five state-of-the-art denoising methods, including original CBM3D [6], the optimized Bayesian non-local means (OBNLM) algorithm [41] utilized in [1] to process realistic noise,

TABLE III

COMPARISON OF DENOISING RESULTS IN TERMS OF PSNR AND SSIM VALUES. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Group #	CBM3D		NC		OBNLM		DnCNN		MC-WNNM		FCCF-s	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1	35.92	0.960	35.64	0.952	35.20	0.942	34.40	0.961	36.92	0.965	**38.00**	**0.973**
2	38.15	0.962	38.06	0.964	37.34	0.951	35.98	0.944	40.46	0.980	**40.89**	**0.983**
3	38.12	0.951	38.28	0.953	38.12	0.947	35.70	0.925	40.45	0.975	**41.13**	**0.980**
4	36.97	0.951	36.97	0.951	37.91	0.945	34.64	0.929	39.53	0.975	**40.36**	**0.981**
5	35.78	0.931	36.19	0.942	35.02	0.910	33.46	0.895	38.13	0.962	**38.62**	**0.969**
6	35.32	0.940	36.88	0.955	35.65	0.935	34.43	0.934	37.02	0.966	**38.45**	**0.975**
7	38.39	0.950	39.49	0.963	38.88	0.954	36.86	0.937	40.87	0.975	**42.17**	**0.981**
8	36.17	0.913	38.26	0.945	37.53	0.929	34.82	0.890	39.89	0.965	**41.70**	**0.978**
9	33.04	0.879	34.34	0.903	33.58	0.885	31.60	0.858	35.29	0.932	**36.62**	**0.952**
Ave.	36.43	0.937	37.12	0.948	36.58	0.933	34.65	0.919	38.73	0.966	**39.77**	**0.975**
Time (s)	31.92		10		3863		0.17		330		146	

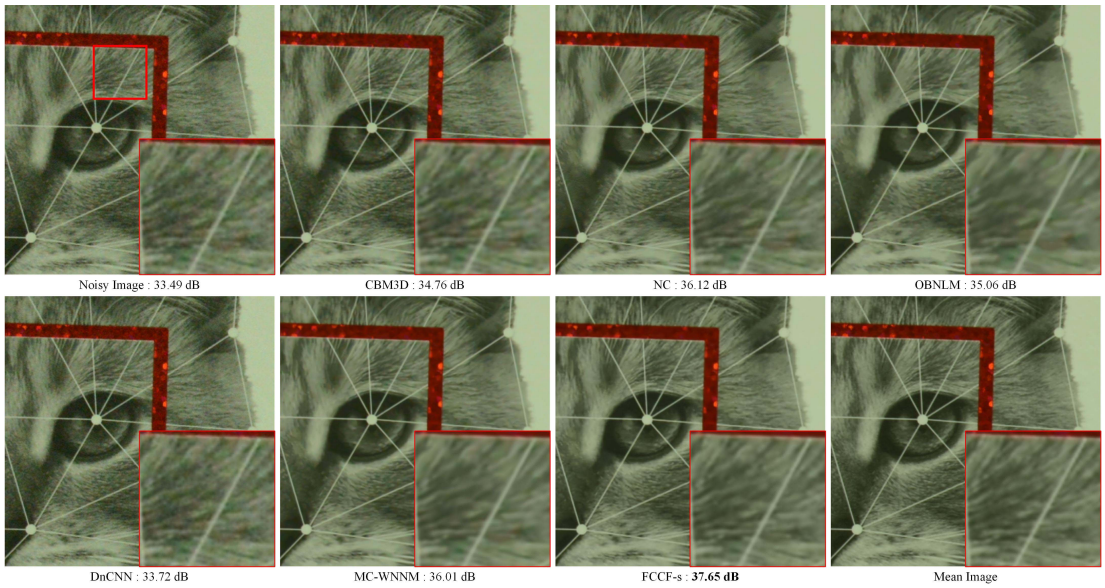

Fig. 11. Comparison of denoising results for an indoor image. The noisy image is captured with the camera setting "NiKon D600 ISO 3200".

the blind denoising method noise clinic (NC) [20], the CNN based denoising method DnCnn [16] and multi-channel weighted nuclear norm minimization (MC-WNNM) method for realistic noise removal [22]. We would like to compare the denoising performance of different algorithms regardless of the noise estimation accuracy. Therefore, for CBM3D, which requires a single parameter for noise level, we utilize the average of the ground truth noise variance map as its parameter. For OBNLM, we utilize the ground truth noise covariance maps as its input. For DnCnn, we utilize the blind color image denoising model trained by the authors since this model is able to process a large range of noise levels. For MC-WNNM, we set their iteration number to 2, which generates their best denoising results. Except OBNLM, which is realized by ourselves, all the other comparison results are generated by the authors' codes.

We utilize both indoor and outdoor scenes to evaluate the denoising performance. Since it is very difficult to capture the outdoor scenes for hundreds of times without shifting, we only capture the outdoor scenes for testing. Namely, there is no ground truth images for outdoor scenes. As stated in Sec. VI-A, we have nine groups of indoor images for testing the denoising performance objectively. Table III presents the denoising results for the nine groups of images in terms of PSNR and SSIM values. The best results are highlighted in bold. We observe that the proposed method consistently outperforms the compared five methods for all the nine groups. DnCnn generates the worst denoising results since its training data is synthesized Gaussian noise. Although it can deal with a large range of Gaussian noise, the noise statistic in high ISO JPEG images is more complex than that of Gaussian noise. Compared with CBM3D, our method achieves a gain of 3.34 dB (0.038) in terms of PSNR (SSIM). This is because that CBM3D is designed for uniform distributed Gaussian noise, while realistic noise is out of its scope. NC, OBNLM, and MC-WNNM are all designed for realistic noise removal. Although we utilize the ground truth noise variance map to calculate the patch similarity of OBNLM, it still cannot generate satisfactory results, comparable with CBM3D. Compared with NC, our method achieves more than 2 dB gain. Compared with the second best result, MC-WNNM, our method still

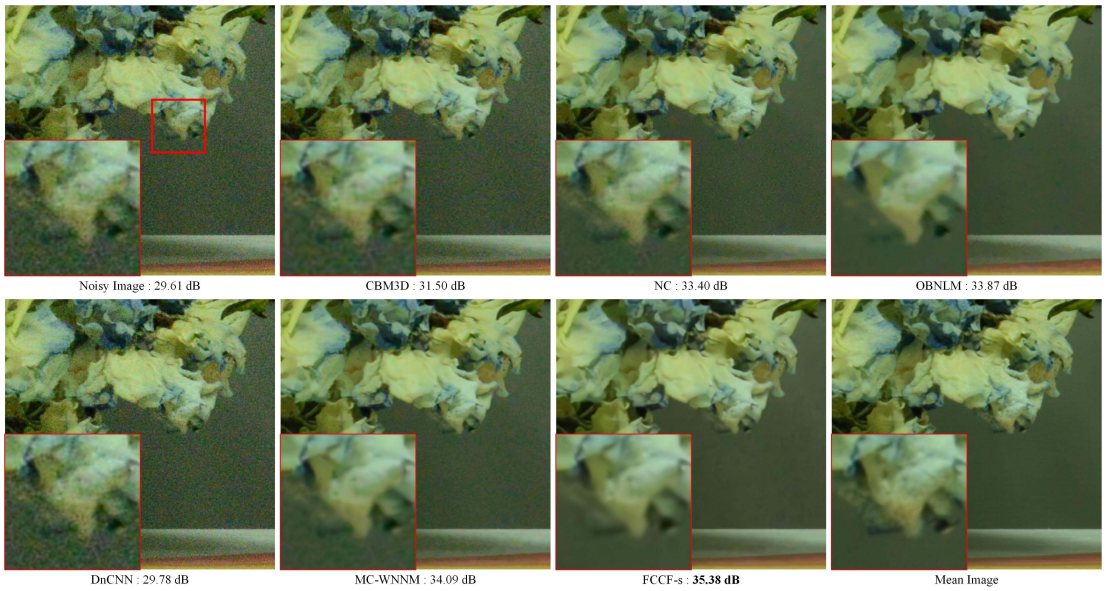

Fig. 12. Comparison of denoising results for an indoor image. The noisy image is captured with the camera setting "NiKon D800 ISO 6400".

Fig. 13. Comparison of denoising results for two outdoor images. The left noisy image is captured with the camera setting "Canon 60D ISO 4000" and the right one is "NiKon D800 ISO 1600".

achieves about 1 dB gain. The reason is that MC-WNNM only considers the noise variance differences in different channels, and it models the noise in each channel as white Gaussian noise, which is not suitable for the noise in high ISO JPEG images. Our method takes advantage of the huge modeling capacity of deep neural network to infer the noise in high ISO JPEG images and achieves the best denoising results.

Fig. 9 to Fig. 12 present the visual comparison results for four indoor images captured using different

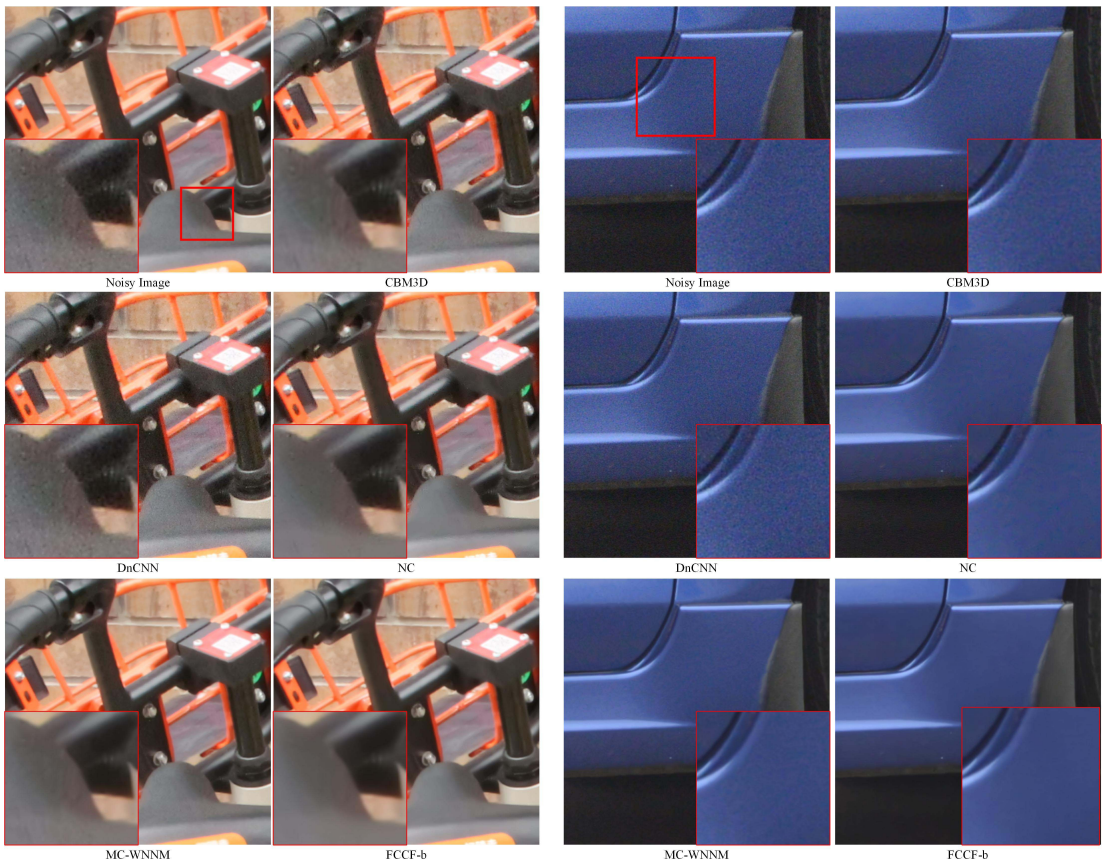

Fig. 14. Comparison of denoising results for two outdoor images. The left noisy image is captured with the camera setting "Canon 60D ISO 5000" and the right one is "NiKon D800 ISO 3200".

cameras.[4] To facilitate the comparison, we display zoomed parts of the images. Please zoom in the figures for better observation. It can be observed that the proposed method removes the noise in flat regions well and recovers rich texture details. As shown in Fig. 9, the results of CBM3D, NC, and DnCnn still contain some noise. Meanwhile the results of OBNLM and MC-WNNM is too smooth. In contrast, our method keeps the texture details better and removes the noise thoroughly compared with the other five methods. From Fig. 10, the results of CBM3D, NC, and DnCnn are still very noisy. The Chinese character in the result of OBNLM does not look like the ground truth. The result of MC-WNNM is a bit noisy in the flat region. Meanwhile, our method recovers a clear Chinese character without the disturbance of noise. The results in Fig. 12 have similar phenomena. For Fig. 11, all the compared methods generate pseudo color while our method recovers the most realistic details. In summary, our method generates the best visual results compared with the competing methods.

The indoor scenes utilized in this paper are mostly printed photos. To verify that the good performance of the proposed method is not because that the testing data have similar properties with that of the training data, we further evaluate the proposed method using outdoor scenes. Fig. 7 provides some samples of our outdoor testing images, which include a variety of objects with different materials, such as flowers, cars, and buildings. Furthermore, we utilize the trained blind model FCCF-b to process the outdoor noisy images captured with different camera settings, as shown in Fig. 13, 14, and 15. More comparison results are provided in the supplemental material. Note that the OBNLM algorithm requires the covariances between the R, G, and B channels as input, but the outdoor scene has no available covariance matrixes. Therefore we do not compare with OBNLM for outdoor images. For CBM3D, we utilize the average of estimated noise variance map as the noise level. We observe that our method removes the noise in flat regions well, while the results of CBM3D and DnCnn still contain much noise. NC cannot remove the noise well in some cases, as the results shown in Fig. 14 and 15. The results of MC-WNNM are smooth, as shown in Fig. 13. Meanwhile, it does not perform well in smooth regions, as shown in Fig. 14. In summary, our method achieves the best visual results for outdoor images.

E. Extension to Poisson-Gaussian Noise Removal

To demonstrate the effectiveness of the proposed collaborative-convolutional filtering framework, we further evaluate the proposed method on Poisson-Gaussian noise removal. We follow the same way of [42] to simulate Poisson-Gaussian noise. Specifically, the noise is added by first scaling the input image to a peak value of 10,

[4]More visual comparison results are presented in the supplemental material.

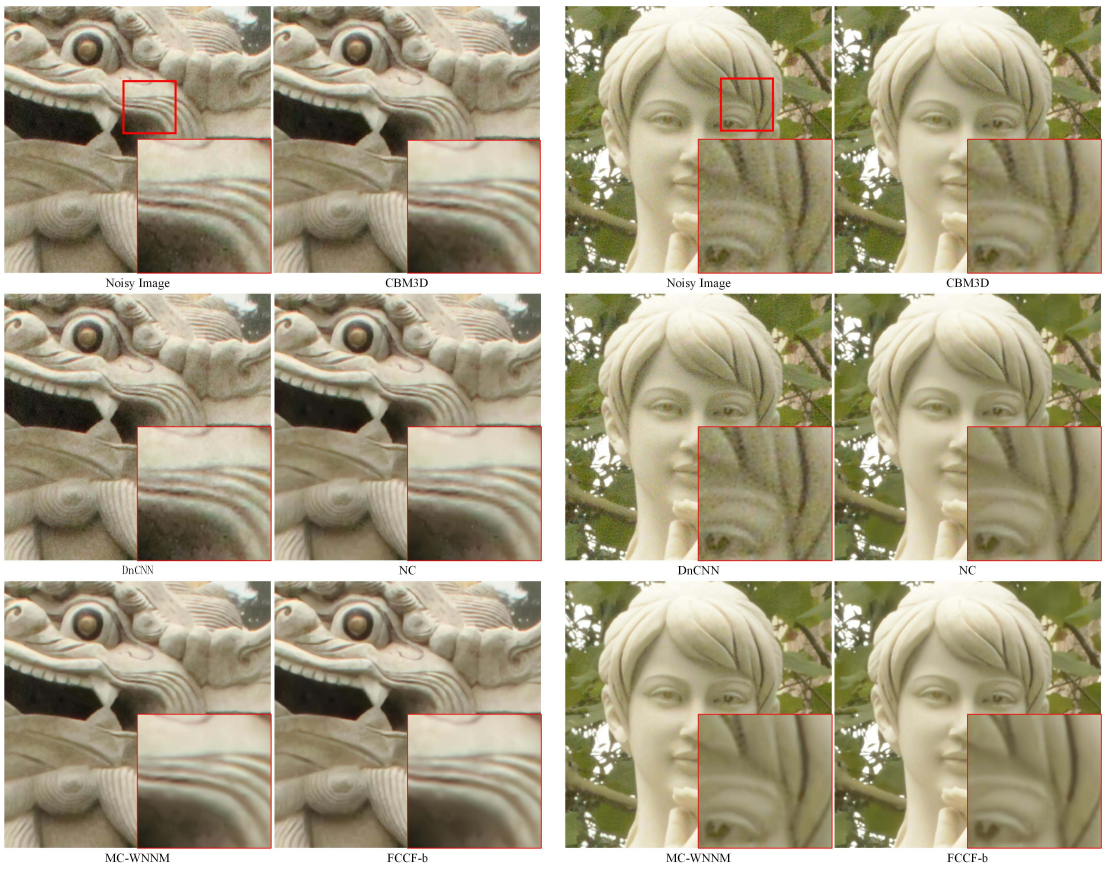

Fig. 15. Comparison of denoising results for two outdoor images. The left noisy image is captured with the camera setting "Canon 60D ISO 6400" and the right one is "NiKon D800 ISO 6400".

instantiating Poisson random variables with the scaled values, and then adding the Gaussian noise with stand variation of 1. Our collaborative filtering result is obtained by first utilizing the generalized Anscombe transformation to stabilize the variance, the same as that of [42]. For the convolutional filtering and deep fusion, we utilize the color version of the BSD68 dataset [43] for testing and the remaining 432 images of the Berkeley segmentation dataset are used for training. The training process converges in 20 epochs. The average denoising results on BSD68 dataset for collaborative and convolutional filtering are 25.47 dB and 26.80 dB, respectively. Our fusion result is 27.20 dB, which is much better than collaborative and convolutional filtering. We also present a visual comparison result in Fig. 16. It can be observed that there is much pseudo-color remaining in the collaborative filtering result, meanwhile convolutional filtering cannot recover the repeating structures well. Our fusion result combines the strengths of the two results, achieving the best visual quality. This experiment demonstrates the generality of proposed method in dealing with different kinds of noise.

F. Computing Complexity

In this section, we discuss the computing complexity of the proposed scheme. Our scheme is realized in Matlab with MatConvnet installed. It takes about 24 hours to train one model on our server with a 1080 Ti GPU and 32 GB RAM. In the test stage, for one image of size 512×512, the noise estimation stage (implemented in Matlab code) takes about 96 seconds and the collaborative filtering process (implemented based on the code of [44]) takes about 50 seconds. The convolutional filtering and deep fusion process takes 0.19 second since it is implemented using GPU. In total, our method takes 146 seconds to process one image. Table III lists the computing time for all the compared methods. The computing time is generated using their reference codes in our server except NC, whose computing time is generated by their online demo code. DnCnn is the fastest code since it is implemented using GPU. CBM3D and NC cost less running time than OBNLM, MC-WNNM and our method, since they are implemented in C codes, while OBNLM and MC-WNNM are all written in Matlab codes. Our method takes less time than the second best method MC-WNNM. Our method can be further accelerated using GPU since the proposed noise estimation and collaborative filtering processes are parallel-friendly.

G. Limitations and Future Work

Since the noise in our training images are introduced by the high ISO mode in imaging, our trained model cannot be directly utilized to process other kinds of noise, such as Gaussian noise, Gamma noise or scanning noise. Fortunately, our model could be easily extended to process these kinds of

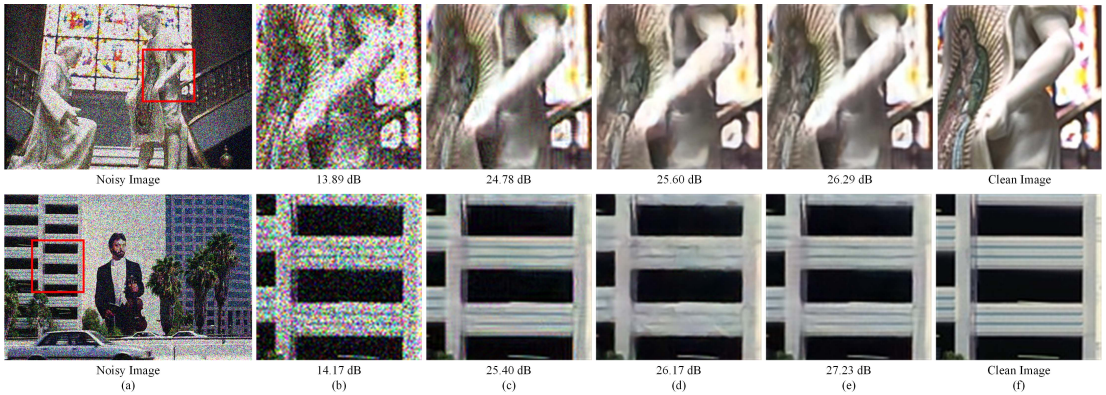

Fig. 16. Poisson-Gaussian noise removal results. From left to right: (a) the synthetic noisy image, (b), (c), (d), and (e) are the highlighted regions cropped from the noisy image, collaborative filtering, convolutional filtering, and fusion results. (f) is the corresponding noise-free image.

noise by enriching our training data with these kinds of noise. In our current network, we do not take the Bayer pattern of noise map into consideration. In the future, we would like to modify the network according to the characteristic of noise. For example, changing the input of the network to the Bayer-down-sampled version of the noisy image, integrating noise estimation and estimated noise maps into the network.

VII. Conclusion

In this paper, we have proposed a novel denoising scheme by deep fusion of the strengths of collaborative and convolutional filtering. For collaborative filtering, we first estimate the noise variance according to the Bayer pattern of noise variance maps. Then, we extend CBM3D to ACBM3D by integrating the estimated noise variance maps and Bayer down-sampling into the denoising process. For convolutional filtering, we utilize the CNN, which includes convolution, batch normalization and relu layers, to remove the noise. Hereafter, the two results are fused together to generate the final denoising result via the proposed deep CNN. Experimental results demonstrate that our method is robust in processing both indoor and outdoor test images, and outperforms state-of-the-art realistic noise removal methods. In addition, we construct a large data set with noisy and noise-free image pairs for high ISO JPEG images, which will facilitate research on this topic.

References

[1] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim, "A holistic approach to cross-channel image noise modeling and its application to image denoising," in *Proc. CVPR*, Jun. 2016, pp. 1683–1691.

[2] X. Liu, M. Tanaka, and M. Okutomi, "Single-image noise level estimation for blind denoising," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5226–5237, Dec. 2013.

[3] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang, "Noise estimation from a single image," in *Proc. CVPR*, Jun. 2006, pp. 901–908.

[4] J. Yang, Z. Gan, Z. Wu, and C. Hou, "Estimation of signal-dependent noise level function in transform domain via a sparse recovery model," *IEEE Trans. Image Process.*, vol. 24, no. 5, pp. 1561–1572, May 2015.

[5] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. CVPR*, Jun. 2005, pp. 60–65.

[6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[7] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.

[8] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. CVPR*, Jun. 2014, pp. 2862–2869.

[9] S. Roth and M. J. Black, "Fields of experts," *Int. J. Comput. Vis.*, vol. 82, no. 2, pp. 205–229, Apr. 2009.

[10] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. ICCV*, Nov. 2011, pp. 479–486.

[11] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in *Proc. CVPR*, Jun. 2012, pp. 2392–2399.

[12] I. Mosseri, M. Zontak, and M. Irani, "Combining the power of internal and external denoising," in *Proc. IEEE Int. Conf. Comput. Photogr.*, Apr. 2013, pp. 1–9.

[13] H. C. Burger, C. Schuler, and S. Harmeling, "Learning how to combine internal and external denoising methods," in *Proc. German Conf. Pattern Recognit.* Berlin, Germany: Springer, 2013, pp. 121–130.

[14] H. Yue, X. Sun, J. Yang, and F. Wu, "Image denoising by exploring external and internal correlations," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1967–1982, Jun. 2015.

[15] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2802–2810.

[16] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[17] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. CVPR*, Jul. 2017, pp. 3929–3938.

[18] S. Lefkimmiatis, "Non-local color image denoising with convolutional neural networks," in *Proc. CVPR*, Jul. 2017, pp. 3587–3596.

[19] M. Lebrun, M. Colom, and J.-M. Morel, "Multiscale image blind denoising," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3149–3161, Oct. 2015.

[20] M. Lebrun, M. Colom, and J.-M. Morel, "The noise clinic: A blind image denoising algorithm," *Image Process. Line*, vol. 5, pp. 1–54, Jan. 2015.

[21] F. Zhu, G. Chen, J. Hao, and P.-A. Heng, "Blind image denoising via dependent Dirichlet process tree," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1518–1531, Aug. 2017.

[22] J. Xu, L. Zhang, D. Zhang, and X. Feng, "Multi-channel weighted nuclear norm minimization for real color image denoising," in *Proc. ICCV*, Oct. 2017, pp. 1096–1104.

[23] J. Xu, L. Zhang, and D. Zhang. (2017). "External prior guided internal prior learning for real-world noisy image denoising." [Online]. Available: https://arxiv.org/abs/1705.04505

[24] P. Meer, J. Jolion, and A. Rosenfeld, "A fast parallel algorithm for blind estimation of noise variance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 2, pp. 216–223, Feb. 1990.

[25] A. Amer and E. Dubois, "Fast and reliable structure-oriented video noise estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 113–118, Jan. 2005.

[26] M. Ghazal and A. Amer, "Homogeneity localization using particle filters with application to noise estimation," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1788–1796, Jul. 2011.
[27] S. Pyatykh, J. Hesser, and L. Zheng, "Image noise level estimation by principal component analysis," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 687–699, Feb. 2013.
[28] G. Chen, F. Zhu, and P. Ann Heng, "An efficient statistical method for image noise level estimation," in *Proc. ICCV*, Dec. 2015, pp. 477–485.
[29] M. Kumar and R. L. Miller, "An image fusion approach for denoising signal-dependent noise," in *Proc. ICASSP*, Mar. 2010, pp. 1438–1441.
[30] M. L. Uss, B. Vozel, V. V. Lukin, and K. Chehdi, "Local signal-dependent noise variance estimation from hyperspectral textural images," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 3, pp. 469–486, Jun. 2011.
[31] L. Alparone, M. Selva, B. Aiazzi, S. Baronti, F. Butera, and L. Chiarantini, "Signal-dependent noise modelling and estimation of new-generation imaging spectrometers," in *Proc. 1st Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, Aug. 2009, pp. 1–4.
[32] S. Abramov, V. Zabrodina, V. Lukin, B. Vozel, K. Chehdi, and J. Astola, "Improved method for blind estimation of the variance of mixed noise using weighted LMS line fitting algorithm," in *Proc. ISCAS*, 2010, pp. 2642–2645.
[33] T. Rabie, "Robust estimation approach for blind denoising," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1755–1765, Nov. 2005.
[34] Neatlab ABSoft. *Neat Image*. [Online]. Available: https://ni.neatvideo.com/home
[35] H. Yue, J. Liu, J. Yang, T. Nguyen, and C. Hou, "Image noise estimation and removal considering the Bayer pattern of noise variance," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2017, pp. 2976–2980.
[36] T. Plotz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *Proc. CVPR*, Jul. 2017, pp. 1586–1595.
[37] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. CVPR*, Jun. 2016, pp. 1646–1654.
[38] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[39] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. (2017). "Deep laplacian pyramid networks for fast and accurate super-resolution." [Online]. Available: https://arxiv.org/abs/1704.03915
[40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[41] C. Kervrann, J. Boulanger, and P. Coupé, "Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal," in *Proc. Int. Conf. Scale Space Variational Methods Comput. Vis.* Berlin, Germany: Springer, 2007, pp. 520–532.
[42] M. Makitalo and A. Foi, "Optimal inversion of the generalized Anscombe transformation for Poisson-Gaussian noise," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 91–103, Jan. 2013.
[43] 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 *Proc. ICCV*, vol. 2, Jul. 2001, pp. 416–423.
[44] M. Lebrun, "An analysis and implementation of the BM3D image denoising method," *Image Process. On Line*, vol. 2, pp. 175–213, Aug. 2012.

Huanjing Yue (M'17) received the B.S. and Ph.D. degrees from Tianjin University, Tianjin, China, in 2010 and 2015, respectively. She was an Intern with Microsoft Research Asia from 2011 to 2012 and from 2013 to 2015. She visited the Video Processing Laboratory, University of California at San Diego, from 2016 to 2017. She is currently an Associate Professor with the School of Electrical and Information Engineering, Tianjin University. Her current research interests include image processing and computer vision. She received the Microsoft Research Asia Fellowship Honor in 2013 and was selected into the Elite Scholar Program of Tianjin University in 2017.

Jianjun Liu received the M.E. degree from the School of Electrical and Information Engineering, Tianjin University, in 2019. His research interests include image denoising and super resolution.

Jingyu Yang (M'10–SM'17) received the B.E. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2003, and the Ph.D. degree (Hons.) from Tsinghua University, Beijing, in 2009. He has been a Faculty Member with Tianjin University, China, since 2009, where he is currently a Professor with the School of Electrical and Information Engineering. He was with Microsoft Research Asia (MSRA) in 2011, and the Signal Processing Laboratory, EPFL, Lausanne, Switzerland, in 2012, and from 2014 to 2015. His research interests include image/video processing, 3D imaging, and computer vision. He has authored or coauthored over 90 high quality research papers (including dozens of IEEE TRANSACTIONS and top conference papers). As a co-author, he received the Best 10% Paper Award in IEEE VCIP 2016 and the Platinum Best Paper Award in IEEE ICME 2017. He served as the Special Session Chair for VCIP 2016 and an Area Chair for ICIP 2017. He was selected into the program for New Century Excellent Talents in University (NCET) from the Ministry of Education, China, in 2011, the Reserved Peiyang Scholar Program of Tianjin University in 2014, and the Tianjin Municipal Innovation Talent Promotion Program in 2015.

Truong Q. Nguyen (F'05) is currently a Professor and the Chair of the ECE Department, UC SanDiego. He is the coauthor (with Prof. Gilbert Strang) of a popular textbook, *Wavelets & Filter Banks* (Wellesley-Cambridge Press, 1997) and the author of several MATLAB-based toolboxes on image compression, electrocardiogram compression, and filter bank design. He has over 400 publications. His current research interests are 3D video processing and communications and their efficient implementation.

Prof. Nguyen received the IEEE TRANSACTIONS ON SIGNAL PROCESSING Paper Award (Image and Multidimensional Processing area) for the paper he co-wrote with Prof. P. P. Vaidyanathan on linear-phase perfect-reconstruction filter banks (1992). He received the NSF Career Award in 1995 and is currently the Series Editor (Digital Signal Processing) for Academic Press. He served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING 1994–1996, the *Signal Processing Letters* 2001–2003, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1996 to 1997, from 2001 to 2004, and the IEEE TRANSACTIONS ON IMAGE PROCESSING from 2004 to 2005.

Feng Wu (M'99–SM'06–F'13) received the B.S. degree in electrical engineering from Xidian University in 1992, and the M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology in 1996 and 1999, respectively.

He is currently a Professor with the University of Science and Technology of China and the Dean of the School of Information Science and Technology. Prior to that, he was a Principle Researcher and a Research Manager with Microsoft Research Asia. He has authored or coauthored over 200 high quality papers (including several dozens of IEEE TRANSACTIONS papers) and top conference papers on MOBICOM, SIGIR, CVPR, and ACM MM. He has 77 granted U.S. patents. His 15 techniques have been adopted into international video coding standards. His research interests include image and video compression, media communication, and media analysis and synthesis. As a coauthor, he received the Best Paper Award in IEEE T-CSVT 2009, PCM 2008, and SPIE VCIP 2007. He serves as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, and several other International journals. He received the IEEE Circuits and Systems Society 2012 Best Associate Editor Award. He served as the TPC Chair for MMSP 2011, VCIP 2010, and PCM 2009, and Special Sessions Chair in ICME 2010 and ISCAS 2013.