FFDNet: Toward a Fast and Flexible Solution for CNN based Image Denoising

Kai Zhang, Wangmeng Zuo, *Senior Member, IEEE,* and Lei Zhang, *Fellow, IEEE*

***Abstract*—Due to the fast inference and good performance, discriminative learning methods have been widely studied in image denoising. However, these methods mostly learn a specific model for each noise level, and require multiple models for denoising images with different noise levels. They also lack flexibility to deal with spatially variant noise, limiting their applications in practical denoising. To address these issues, we present a fast and flexible denoising convolutional neural network, namely FFDNet, with a tunable noise level map as the input. The proposed FFDNet works on downsampled sub-images, achieving a good trade-off between inference speed and denoising performance. In contrast to the existing discriminative denoisers, FFDNet enjoys several desirable properties, including (i) the ability to handle a wide range of noise levels (i.e., [0, 75]) effectively with a single network, (ii) the ability to remove spatially variant noise by specifying a non-uniform noise level map, and (iii) faster speed than benchmark BM3D even on CPU without sacrificing denoising performance. Extensive experiments on synthetic and real noisy images are conducted to evaluate FFDNet in comparison with state-of-the-art denoisers. The results show that FFDNet is effective and efficient, making it highly attractive for practical denoising applications.**

***Index Terms*—Image denoising, convolutional neural networks, Gaussian noise, spatially variant noise**

I. Introduction

THE importance of image denoising in low level vision can be revealed from many aspects. First, noise corruption is inevitable during the image sensing process and it may heavily degrade the visual quality of acquired image. Removing noise from the observed image is an essential step in various image processing and computer vision tasks [1], [2]. Second, from the Bayesian perspective, image denoising is an ideal test bed for evaluating image prior models and optimization methods [3], [4], [5]. Last but not least, in the unrolled inference via variable splitting techniques, many image restoration problems can be addressed by sequentially solving a series of denoising subproblems, which further broadens the application fields of image denoising [6], [7], [8], [9].

As in many previous literature of image denoising [10], [11], [12], [13], in this paper we assume that the noise is additive white Gaussian noise (AWGN) and the noise level is given. In order to handle practical image denoising problems, a flexible image denoiser is expected to have the following desirable properties: (i) it is able to perform denoising using a single model; (ii) it is efficient, effective and user-friendly; and (iii) it can handle spatially variant noise. Such a denoiser can be directly deployed to recover the clean image when the noise level is known or can be well estimated. When the noise level is unknown or is difficult to estimate, the denoiser should allow the user to adaptively control the trade-off between noise reduction and detail preservation. Furthermore, the noise can be spatially variant and the denoiser should be flexible enough to handle spatially variant noise.

However, state-of-the-art image denoising methods are still limited in flexibility or efficiency. In general, image denoising methods can be grouped into two major categories, model-based methods and discriminative learning based ones. Model-based methods such as BM3D [11] and WNNM [5] are flexible in handling denoising problems with various noise levels, but they suffer from several drawbacks. For example, their optimization algorithms are generally time-consuming, and cannot be directly used to remove spatially variant noise. Moreover, model-based methods usually employ hand-crafted image priors (e.g., sparsity [14], [15] and nonlocal self-similarity [12], [13], [16]), which may not be strong enough to characterize complex image structures.

As an alternative, discriminative denoising methods aim to learn the underlying image prior and fast inference from a training set of degraded and ground-truth image pairs. One approach is to learn stage-wise image priors in the context of truncated inference procedure [17]. Another more popular approach is plain discriminative learning, such as the MLP [18] and convolutional neural network (CNN) based methods [19], [20], among which the DnCNN [20] method has achieved very competitive denoising performance. The success of CNN for image denoising is attributed to its large modeling capacity and tremendous advances in network training and design. However, existing discriminative denoising methods are limited in flexibility, and the learned model is usually tailored to a specific noise level. From the perspective of regression, they aim to learn a mapping function $\boldsymbol{x} = \mathcal{F}(\boldsymbol{y}; \Theta_\sigma)$ between the input noisy observation $\boldsymbol{y}$ and the desired output $\boldsymbol{x}$. The model parameters Θ_σ are trained for noisy images corrupted by AWGN with a fixed noise level σ, while the trained model with Θ_σ is hard to be directly deployed to images with other noise levels. Though a single CNN model (i.e., DnCNN-B) is trained in [20] for Gaussian denoising, it does not generalize well to real noisy images and works only

This project is partially supported by the National Natural Scientific Foundation of China (NSFC) under Grant No. 61671182 and 61471146, and the HK RGC GRF grant (under no. PolyU 152124/15E).

K. Zhang is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China, and also with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: cskaizhang@gmail.com).

W. Zuo is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: cswmzuo@gmail.com).

L. Zhang is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: cslzhang@comp.polyu.edu.hk).

if the noise level is in the preset range, e.g., $[0, 55]$. Besides, all the existing discriminative learning based methods lack flexibility to deal with spatially variant noise.

To overcome the drawbacks of existing CNN based denoising methods, we present a fast and flexible denoising convolutional neural network (FFDNet). Specifically, our FFDNet is formulated as $\boldsymbol{x} = \mathcal{F}(\boldsymbol{y}, \boldsymbol{M}; \Theta)$, where $\boldsymbol{M}$ is a noise level map. In the DnCNN model $\boldsymbol{x} = \mathcal{F}(\boldsymbol{y}; \Theta_\sigma)$, the parameters Θ_σ vary with the change of noise level σ, while in the FFDNet model, 1) the noise level map is modeled as an input and the model parameters Θ are invariant to noise level. Thus, FFDNet provides a flexible way to handle different noise levels with a single network.

By introducing a noise level map as input, it is natural to expect that the model performs well when the noise level map matches the ground-truth one of noisy input. Furthermore, the noise level map should also play the role of controlling the trade-off between noise reduction and detail preservation. It is found that heavy visual quality degradation may be engendered when setting a larger noise level to smooth out the details. We highlight this problem and adopt a method of 2) orthogonal initialization on convolutional filters to alleviate this. Besides, the proposed FFDNet 3) works on downsampled sub-images, which largely accelerates the training and testing speed, and enlarges the receptive field as well.

Using images corrupted by AWGN, we quantitatively compare FFDNet with state-of-the-art denoising methods, including model-based methods such as BM3D [11] and WNNM [5] and discriminative learning based methods such as TNRD [17] and DnCNN [20]. The results clearly demonstrate the superiority of FFDNet in terms of both denoising performance and computational efficiency. In addition, FFDNet performs favorably on images corrupted by spatially variant AWGN. We further evaluate FFDNet on real-world noisy images, where the noise is often signal-dependent, non-Gaussian and spatially variant. The proposed FFDNet model still achieves perceptually convincing results by setting proper noise level maps. Overall, FFDNet enjoys high potentials for practical denoising applications.

The main contribution of our work is summarized as follows:

- A fast and flexible denoising network, namely FFDNet, is proposed for discriminative image denoising. By taking a tunable noise level map as input, a single FFDNet is able to deal with noise on different levels, as well as spatially variant noise.
- We highlight the importance to guarantee the role of the noise level map in controlling the trade-off between noise reduction and detail preservation.
- FFDNet exhibits perceptually appealing results on both synthetic noisy images corrupted by AWGN and real-world noisy images, demonstrating its potential for practical image denoising.

The remainder of this paper is organized as follows. Sec. II reviews existing discriminative denoising methods. Sec. III presents the proposed image denoising model. Sec. IV reports the experimental results. Sec. V concludes the paper.

II. Related Work

In this section, we briefly review and discuss the two major categories of relevant methods to this work, i.e., maximum a posteriori (MAP) inference guided discriminative learning and plain discriminative learning.

A. MAP Inference Guided Discriminative Learning

Instead of first learning the prior and then performing the inference, this category of methods aims to learn the prior parameters along with a compact unrolled inference through minimizing a loss function [21]. Following the pioneer work of fields of experts [3], Barbu [21] trained a discriminative Markov random field (MRF) model together with a gradient descent inference for image denoising. Samuel and Tappen [22] independently proposed a compact gradient descent inference learning framework, and discussed the advantages of discriminative learning over model-based optimization method with MRF prior. Sun and Tappen [23] proposed a novel nonlocal range MRF (NLR-MRF) framework, and employed the gradient-based discriminative learning method to train the model. Generally speaking, the methods above only learn the prior parameters in a discriminative manner, while the inference parameters are stage-invariant.

With the aid of unrolled half quadratic splitting (HQS) techniques, Schmidt et al. [24], [25] proposed a cascade of shrinkage fields (CSF) framework to learn stage-wise inference parameters. Chen et al. [17] further proposed a trainable nonlinear reaction diffusion (TNRD) model through discriminative learning of a compact gradient descent inference step. Recently, Lefkimmiatis [26] and Qiao et al. [27] adopted a proximal gradient-based denoising inference from a variational model to incorporate the nonlocal self-similarity prior. It is worth noting that, apart from MAP inference, Vemulapalli et al. [28] derived an end-to-end trainable patch-based denoising network based on Gaussian Conditional Random Field (GCRF) inference.

MAP inference guided discriminative learning usually requires much fewer inference steps, and is very efficient in image denoising. It also has clear interpretability because the discriminative architecture is derived from optimization algorithms such as HQS and gradient descent [17], [21], [22], [23], [24]. However, the learned priors and inference procedure are limited by the form of MAP model [25], and generally perform inferior to the state-of-the-art CNN-based denoisers. For example, the inference of CSF [24] is not very flexible since it is strictly derived from the HQS optimization under the field of experts (FoE) framework. The capacity of FoE is however not large enough to fully characterize image priors, which in turn makes CSF less effective. For these reasons, Kruse et al. [29] generalized CSF for better performance by replacing some modular parts of unrolled inference with more powerful CNN.

B. Plain Discriminative Learning

Instead of modeling image priors explicitly, the plain discriminative learning methods learn a direct mapping function to model image prior implicitly. The multi-layer perceptron

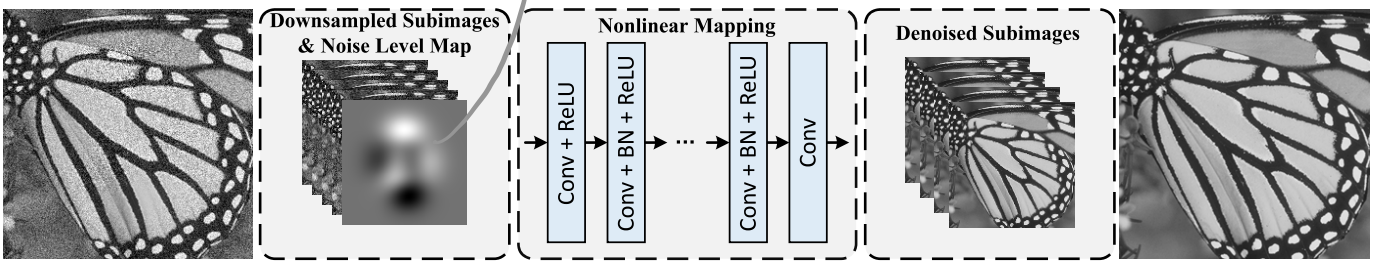


Fig. 1. The architecture of the proposed FFDNet for image denoising. The input image is reshaped to four sub-images, which are then input to the CNN together with a noise level map. The final output is reconstructed by the four denoised sub-images.

(MLP) and CNNs have been adopted to learn such priors. The use of CNN for image denoising can be traced back to [19], where a five-layer network with sigmoid nonlinearity was proposed. Subsequently, auto-encoder based methods have been suggested for image denoising [30], [31]. However, early MLP and CNN-based methods are limited in denoising performance and cannot compete with the benchmark BM3D method [11].

The first discriminative denoising method which achieves comparable performance with BM3D is the plain MLP method proposed by Burger et al. [18]. Benefitted from the advances in deep CNN, Zhang et al. [20] proposed a plain denoising CNN (DnCNN) method which achieves state-of-the-art denoising performance. They showed that residual learning and batch normalization [32] are particularly useful for the success of denoising. For a better trade-off between accuracy and speed, Zhang et al. [9] introduced a 7-layer denoising network with dilated convolution [33] to expand the receptive field of CNN. Mao et al. [34] proposed a very deep fully convolutional encoding-decoding network with symmetric skip connection for image denoising. Santhanam et al. [35] developed a recursively branched deconvolutional network (RBDN) for image denoising as well as generic image-to-image regression. Tai et al. [36] proposed a very deep persistent memory network (MemNet) by introducing a memory block to mine persistent memory through an adaptive learning process.

Plain discriminative learning has shown better performance than MAP inference guided discriminative learning; however, existing discriminative learning methods have to learn multiple models for handling images with different noise levels, and are incapable to deal with spatially variant noise. To the best of our knowledge, it remains an unaddressed issue to develop a single discriminative denoising model which can handle noise of different levels, even spatially variant noise, in a speed even faster than BM3D.

III. Proposed Fast and Flexible Discriminative CNN Denoiser

We present a single discriminative CNN model, namely FFDNet, to achieve the following three objectives:

- Fast speed: The denoiser is expected to be highly efficient without sacrificing denoising performance.
- Flexibility: The denoiser is able to handle images with different noise levels and even spatially variant noise.
- Robustness: The denoiser should introduce no visual artifacts in controlling the trade-off between noise reduction and detail preservation.

In this work, we take a tunable noise level map $\boldsymbol{M}$ as input to make the denoising model flexible to noise levels. To improve the efficiency of the denoiser, a reversible downsampling operator is introduced to reshape the input image of size $W \times H \times C$ into four downsampled sub-images of size $\frac{W}{2} \times \frac{H}{2} \times 4C$. Here C is the number of channels, i.e., $C = 1$ for grayscale image and $C = 3$ for color image. In order to enable the noise level map to robustly control the trade-off between noise reduction and detail preservation by introducing no visual artifacts, we adopt the orthogonal initialization method to the convolution filters.

A. Network Architecture

Fig. 1 illustrates the architecture of FFDNet. The first layer is a reversible downsampling operator which reshapes a noisy image $\boldsymbol{y}$ into four downsampled sub-images. We further concatenate a tunable noise level map $\boldsymbol{M}$ with the downsampled sub-images to form a tensor $\tilde{\boldsymbol{y}}$ of size $\frac{W}{2} \times \frac{H}{2} \times (4C+1)$ as the inputs to CNN. For spatially invariant AWGN with noise level σ, $\boldsymbol{M}$ is a uniform map with all elements being σ.

With the tensor $\tilde{\boldsymbol{y}}$ as input, the following CNN consists of a series of 3×3 convolution layers. Each layer is composed of three types of operations: Convolution (Conv), Rectified Linear Units (ReLU) [37], and Batch Normalization (BN) [32]. More specifically, "Conv+ReLU" is adopted for the first convolution layer, "Conv+BN+ReLU" for the middle layers, and "Conv" for the last convolution layer. Zero-padding is employed to keep the size of feature maps unchanged after each convolution. After the last convolution layer, an upscaling operation is applied as the reverse operator of the downsampling operator applied in the input stage to produce the estimated clean image $\hat{\boldsymbol{x}}$ of size $W \times H \times C$. Different from DnCNN, FFDNet does not predict the noise. The reason is given in Sec. III-F. Since FFDNet operates on downsampled sub-images, it is not necessary to employ the dilated convolution [33] to further increase the receptive field.

By considering the balance of complexity and performance, we empirically set the number of convolution layers as 15 for grayscale image and 12 for color image. As to the channels of feature maps, we set 64 for grayscale image and 96 for color image. The reason that we use different settings for grayscale and color images is twofold. First, since there are

high dependencies among the R, G, B channels, using a smaller number of convolution layers encourages the model to exploit the inter-channel dependency. Second, color image has more channels as input, and hence more feature (i.e., more channels of feature map) is required. According to our experimental results, increasing the number of feature maps contributes more to the denoising performance on color images. Using different settings for color images, FFDNet can bring an average gain of 0.15dB by PSNR on different noise levels. As we shall see from Sec. IV-F, 12-layer FFDNet for color image runs slightly slower than 15-layer FFDNet for grayscale image. Taking both denoising performance and efficiency into account, we set the number of convolution layers as 12 and the number of feature maps as 96 for color image denoising.

B. Noise Level Map

Let's first revisit the model-based image denoising methods to analyze why they are flexible in handling noises at different levels, which will in turn help us to improve the flexibility of CNN-based denoiser. Most of the model-based denoising methods aim to solve the following problem

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{x}\|^2 + \lambda\Phi(\boldsymbol{x}), \tag{1}$$

where $\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{x}\|^2$ is the data fidelity term with noise level σ, $\Phi(\boldsymbol{x})$ is the regularization term associated with image prior, and λ controls the balance between the data fidelity and regularization terms. It is worth noting that in practice λ governs the compromise between noise reduction and detail preservation. When it is too small, much noise will remain; on the opposite, details will be smoothed out along with suppressing noise.

With some optimization algorithms, the solution of Eqn. (1) actually defines an implicit function given by

$$\hat{\boldsymbol{x}} = \mathcal{F}(\boldsymbol{y}, \sigma, \lambda; \Theta). \tag{2}$$

Since λ can be absorbed into σ, Eqn. (2) can be rewritten as

$$\hat{\boldsymbol{x}} = \mathcal{F}(\boldsymbol{y}, \sigma; \Theta). \tag{3}$$

In this sense, setting noise level σ also plays the role of setting λ to control the trade-off between noise reduction and detail preservation. In a word, model-based methods are flexible in handling images with various noise levels by simply specifying σ in Eqn. (3).

According to the above discussion, it is natural to utilize CNN to learn an explicit mapping of Eqn. (3) which takes the noise image and noise level as input. However, since the inputs $\boldsymbol{y}$ and σ have different dimensions, it is not easy to directly feed them into CNN. Inspired by the patch based denoising methods which actually set σ for each patch, we resolve the dimensionality mismatching problem by stretching the noise level σ into a noise level map $\boldsymbol{M}$. In $\boldsymbol{M}$, all the elements are σ. As a result, Eqn. (3) can be further rewritten as

$$\hat{\boldsymbol{x}} = \mathcal{F}(\boldsymbol{y}, \boldsymbol{M}; \Theta). \tag{4}$$

It is worth emphasizing that $\boldsymbol{M}$ can be extended to degradation maps with multiple channels for more general noise models such as the multivariate (3D) Gaussian noise model $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with zero mean and covariance matrix $\boldsymbol{\Sigma}$ in the RGB color space [38]. As such, a single CNN model is expected to inherit the flexibility of handling noise model with different parameters, even spatially variant noises by noting $\boldsymbol{M}$ can be non-uniform.

C. Denoising on Sub-images

Efficiency is another crucial issue for practical CNN-based denoising. One straightforward idea is to reduce the depth and number of filters. However, such a strategy will sacrifice much the modeling capacity and receptive field of CNN [20]. In [9], dilated convolution is introduced to expand receptive field without the increase of network depth, resulting in a 7-layer denoising CNN. Unfortunately, we empirically find that FFDNet with dilated convolution tends to generate artifacts around sharp edges.

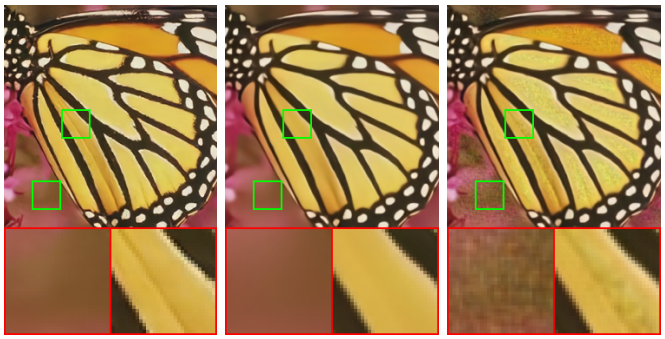
Shi et al. [39] proposed to extract deep features directly from the low-resolution image for super-resolution, and introduced a sub-pixel convolution layer to improve computational efficiency. In the application of image denoising, we introduce a reversible downsampling layer to reshape the input image into a set of small sub-images. Here the downsampling factor is set to 2 since it can largely improve the speed without reducing modeling capacity. The CNN is deployed on the sub-images, and finally a sub-pixel convolution layer is adopted to reverse the downsampling process.

Denoising on downsampled sub-images can also effectively expand the receptive field which in turn leads to a moderate network depth. For example, the proposed network with a depth of 15 and 3×3 convolution will have a large receptive field of 62×62. In contrast, a plain 15-layer CNN only has a receptive field size of 31×31. We note that the receptive field of most state-of-the-art denoising methods ranges from 35×35 to 61×61 [20]. Further increase of receptive field actually benefits little in improving denoising performance [40]. What is more, the introduction of subsampling and sub-pixel convolution is effective in reducing the memory burden.

Experiments are conducted to validate the effectiveness of downsampling for balancing denoising accuracy and efficiency on the BSD68 dataset with $\sigma = 15$ and 50. For grayscale image denoising, we train a baseline CNN which has the same depth as FFDNet without downsampling. The comparison of average PSNR values is given as follows: (i) when σ is small (i.e., 15), the baseline CNN slightly outperforms FFDNet by 0.02dB; (ii) when σ is large (i.e., 50), FFDNet performs better than the baseline CNN by 0.09dB. However, FFDNet is nearly 3 times faster and is more memory-friendly than the baseline CNN. As a result, by performing denoising on sub-images, FFDNet significantly improves efficiency while maintaining denoising performance.

D. Examining the Role of Noise Level Map

By training the model with abundant data units $(\boldsymbol{y}, \boldsymbol{M}; \boldsymbol{x})$, where $\boldsymbol{M}$ is exactly the noise level map of $\boldsymbol{y}$, the model is expected to perform well when the noise level matches the ground-truth one (see Fig. 2(a)). On the other hand, in practice,



(a) (b) (c)

Fig. 2. An example to show the importance of guaranteeing the role of noise level map in controlling the trade-off between noise reduction and detail preservation. The input is a noisy image with noise level 25. (a) Result without visual artifacts by matched noise level 25. (b) Result without visual artifacts by mismatched noise level 60. (c) Result with visual artifacts by mismatched noise level 60.

we may need to use the learned model to smooth out some details with a higher noise level map than the ground-truth one (see Fig. 2(b)). In other words, one may take advantage of the role of λ to control the trade-off between noise reduction and detail preservation. Hence, it is very necessary to further examine whether $\boldsymbol{M}$ can play the role of λ.

Unfortunately, the use of both $\boldsymbol{M}$ and $\boldsymbol{y}$ as input also increases the difficulty to train the model. According to our experiments on several learned models, the model may give rise to visual artifacts especially when the input noise level is much higher than the ground-truth one (see Fig. 2(c)), which indicates $\boldsymbol{M}$ fails to control the trade-off between noise reduction and detail preservation. Note that it does not mean all the models suffer from such problem. One possible solution to avoid this is to regularize the convolution filters. As a widely-used regularization method, orthogonal regularization has proven to be effective in eliminating the correlation between convolution filters, facilitating gradient propagation and improving the compactness of the learned model. In addition, recent studies have demonstrated the advantage of orthogonal regularization in enhancing the network generalization ability in applications of deep hashing and image classification [41], [42], [43], [44], [45]. According to our experiments, we empirically find that the orthogonal initialization of the convolution filters [43], [46] works well in suppressing the above mentioned visual artifacts.

It is worth emphasising that this section aims to highlight the necessity of guaranteeing the role of $\boldsymbol{M}$ in controlling the trade-off between noise reduction and detail preservation rather than proposing a method to avoid the possible visual artifacts caused by noise level mismatch. In practice, one may retrain the model until $\boldsymbol{M}$ plays its role and results in no visual artifacts with a lager noise level.

E. FFDNet vs. a Single Blind Model

So far, we have known that it is possible to learn a single model for blind and non-blind Gaussian denoising, respectively. And it is of significant importance to clarify their differences.

First, the generalization ability is different. Although the blind model performs favorably for synthetic AWGN removal without knowing the noise level, it does not generalize well to real noisy images whose noises are much more complex than AWGN (see the results of DnCNN-B in Fig. 8). Actually, since the CNN model can be treated as the inference of Eqn. (1) and the data fidelity term corresponds to the degradation process (or the noise model), the modeling accuracy of the degradation process is very important for the success of a denoising model. For example, a model trained for AWGN removal is not expected to be still effective for Poisson noise removal. By contrast, the non-blind FFDNet model can be viewed as multiple denoisers, each of which is anchored with a noise level. Accordingly, it has the ability to control the trade-off between noise removal and detail preservation which in turn facilitates the removal of real noise to some extent (see the results of DnCNN and FFDNet in Fig. 8).

Second, the performance for AWGN removal is different. The non-blind model with noise level map has moderately better performance for AWGN removal than the blind one (about 0.1dB gain on average for the BSD68 dataset), possibly because the noise level map provides additional information to the input. Similar phenomenon has also been recognized in the task of single image super-resolution (SISR) [47].

Third, the application range is different. In the variable splitting algorithms for general image restoration tasks, the prior term involves a denoising subproblem with a current noise level [6], [7], [8]. Thus, the non-blind model can be easily plugged into variable splitting algorithms to solve various image restoration tasks, such as image deblurring, SISR, and image inpainting [9], [48]. However, the blind model does not have this merit.

F. Residual vs. Non-residual Learning of Plain CNN

It has been pointed out that the integration of residual learning for plain CNN and batch normalization is beneficial to the removal of AWGN as it eases the training and tends to deliver better performance [20]. The main reason is that the residual (noise) output follows a Gaussian distribution which facilitates the Gaussian normalization step of batch normalization. The denoising network gains most from such a task-specific merit especially when a single noise level is considered.

In FFDNet, we instead consider a wide range of noise level and introduce a noise level map as input. Thus, it is interesting to revisit the integration of residual learning and batch normalization for plain CNN. According to our experiments, batch normalization can always accelerate the training of denoising network regardless of the residual or non-residual learning strategy of plain CNN. In particular, with batch normalization, while the residual learning enjoys a faster convergence than non-residual learning, their final performances after fine-tuning are almost exactly the same. Some recent works have proposed to train very deep plain networks with nearly the same performance to that with residual learning [44], [49]. In fact, when a network is moderately

deep (e.g., less than 20), it is feasible to train a plain network without the residual learning strategy by using advanced CNN training and design techniques such as ReLU [37], batch normalization [32] and Adam [50]. For simplicity, we do not use residual learning for network design.

G. Un-clipping vs. Clipping of Noisy Images for Training

In the AWGN denoising literature, there exist two widely-used settings, i.e., un-clipping [5], [11], [17], [18] and clipping [24], [28], of synthetic noisy image to evaluate the performance of denoising methods. The main difference between the two settings lies in whether the noisy image is clipped into the range of 0-255 (or more precisely, quantized into 8-bit format) after adding the noise.

On the one hand, the un-clipping setting which is also the most widely-used setting serves an ideal test bed for evaluating the denoising methods. This is because most denoising methods assume the noise is ideal AWGN, and the clipping of noisy input would make the noise characteristics deviate from being AWGN. Furthermore, in the variable splitting algorithms for solving general image restoration problems, there exists a subproblem which, from a Bayesian perspective, corresponds to a Gaussian denoising problem [9], [48]. This further broadens the use of the un-clipping setting. Thus, unless otherwise specified, FFDNet in this work refers to the model trained on images without clipping or quantization.

On the other hand, since real noisy images are always integer-valued and range-limited, it has been argued that the clipping setting of noisy image makes the data more realistic [24]. However, when the noise level is high, the noise will be not zero-mean any more due to clipping effects [51]. This in turn will lead to unreliable denoiser for plugging into the variable splitting algorithms to solve other image restoration problems.

To thoroughly evaluate the proposed method, we also train an FFDNet model with clipping setting of noisy image, namely FFDNet-Clip, for comparison. During training and testing of FFDNet-Clip, the noisy images are quantized into 8-bit format. Specifically, for a clean image $\boldsymbol{x}$, we use the matlab function `imnoise`$(\boldsymbol{x}$, `'gaussian'`, 0, $(\frac{\sigma}{255})^2)$ to generate the quantized noisy $\boldsymbol{y}$ with noise level σ.

IV. Experiments

A. Dataset Generation and Network Training

To train the FFDNet model, we need to prepare a training dataset of input-output pairs $\{(\boldsymbol{y}_i, \boldsymbol{M}_i; \boldsymbol{x}_i)\}_{i=1}^{N}$. Here, $\boldsymbol{y}_i$ is obtained by adding AWGN to latent image $\boldsymbol{x}_i$, and $\boldsymbol{M}_i$ is the noise level map. The reason to use AWGN to generate the training dataset is two-fold. First, AWGN is a natural choice when there is no specific prior information on noise source. Second, real-world noise can be approximated as locally AWGN [52]. More specifically, FFDNet model is trained on the noisy images $\boldsymbol{y}_i = \boldsymbol{x}_i + \boldsymbol{v}_i$ without quantization to 8-bit integer values. Though the real noisy images are generally 8-bit quantized, we empirically found that the learned model still works effectively on real noisy images. For FFDNet-Clip, as mentioned in Sec. III-G, we use the matlab function `imnoise` to generate the quantized noisy image from a clean one.

TABLE I
Main specifications of the proposed FFDNet

FFDNet models	Number of layers	Number of channels	Noise level range	Training patch size
Grayscale	15	64	$[0, 75]$	70×70
Color	12	96	$[0, 75]$	50×50

We collected a large dataset of source images, including 400 BSD images, 400 images selected from the validation set of imageNet [53], and the 4,744 images from the Waterloo Exploration Database [54]. In each epoch, we randomly crop $N = 128 \times 8,000$ patches from these images for training. The patch size should be larger than the receptive field of FFDNet, and we set it to 70×70 and 50×50 for grayscale images and color images, respectively. The noisy patches are obtained by adding AWGN of noise level $\sigma \in [0, 75]$ to the clean patches. For each noisy patch, the noise level map is uniform. Since FFDNet is a fully convolutional neural network, it inherits the local connectivity property that the output pixel is determined by the local noisy input and local noise level. Hence, the trained FFDNet naturally has the ability to handle spatially variant noise by specifying a non-uniform noise level map. For clarity, in Table I we list the main specifications of the FFDNet models.

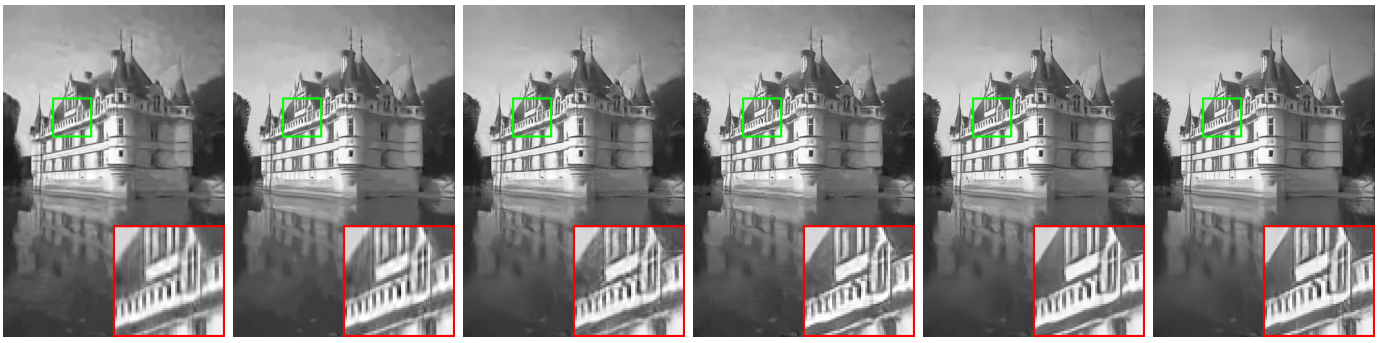
The ADAM algorithm [50] is adopted to optimize FFDNet by minimizing the following loss function,

$$\mathcal{L}(\Theta) = \frac{1}{2N} \sum_{i=1}^{N} \|\mathcal{F}(\boldsymbol{y}_i, \boldsymbol{M}_i; \Theta) - \boldsymbol{x}_i\|^2. \quad (5)$$

The learning rate starts from 10^{-3} and reduces to 10^{-4} when the training error stops decreasing. When the training error keeps unchanged in five sequential epochs, we merge the parameters of each batch normalization into the adjacent convolution filters. Then, a smaller learning rate of 10^{-6} is adopted for additional 50 epochs to fine-tune the FFDNet model. As for the other hyper-parameters of ADAM, we use their default settings. The mini-batch size is set as 128, and the rotation and flip based data augmentation is also adopted during training. The FFDNet models are trained in Matlab (R2015b) environment with MatConvNet package [55] and an Nvidia Titan X Pascal GPU. The training of a single model can be done in about two days.

To evaluate the proposed FFDNet denoisers on grayscale image denoising, we use BSD68 [3] and Set12 datasets to test FFDNet for removing AWGN noise, and use the "RNI6" dataset [56] to test FFDNet for removing real noise. The BSD68 dataset consists of 68 images from the separate test set of the BSD300 dataset [57]. The Set12 dataset is a collection of widely-used testing images. The RNI6 dataset contains 6 real noisy images without ground-truth. In particular, to evaluate FFDNet-Clip, we use the quantized "Clip300" dataset which comprises the 100 images of test set from the BSD300 dataset [57] and 200 images from PASCALVOC 2012 [58] dataset. Note that all the testing images are not included in the training dataset.

As for color image denoising, we employ four datasets,



(a) BM3D (26.21dB) (b) WNNM (26.51dB) (c) MLP (26.54dB) (d) TNRD (26.59dB) (e) DnCNN (26.89dB) (f) FFDNet (26.93dB)

Fig. 3. Denoising results on image "*102061*" from the BSD68 dataset with noise level 50 by different methods.

namely CBSD68, Kodak24 [59], McMaster [60], and "RNI15" [56], [61]. The CBSD68 dataset is the corresponding color version of the grayscale BSD68 dataset. The Kodak24 dataset consists of 24 center-cropped images of size 500×500 from the original Kodak dataset. The McMaster dataset is a widely-used dataset for color demosaicing, which contains 18 cropped images of size 500×500. Compared to the Kodak24 images, the images in McMaster dataset exhibit more saturated colors [60]. The RNI15 dataset consists of 15 real noisy images. We note that RNI6 and RNI15 cover a variety of real noise types, such as camera noise and JPEG compression noise. Since the ground-truth clean images are unavailable for real noisy images, we thus only provide the visual comparisons on these images. The source codes of FFDNet and its extension to multivariate Gaussian noise are available at https://github.com/cszn/FFDNet.

(a) CBM3D (25.49dB) (b) CDnCNN (26.19dB) (c) FFDNet (26.28dB)

Fig. 4. Color image denoising results by CBM3D, CDnCNN and FFDNet on noise level $\sigma = 50$.

B. Experiments on AWGN Removal

In this subsection, we test FFDNet on noisy images corrupted by spatially invariant AWGN. For grayscale image denoising, we mainly compare FFDNet with state-of-the-art methods BM3D [11], WNNM [5], MLP [18], TNRD [17], and DnCNN [20]. Note that BM3D and WNNM are two representative model-based methods based on nonlocal self-similarity prior, whereas TNRD, MLP and DnCNN are discriminative learning based methods. Tables II and III report the PSNR results on BSD68 and Set12 datasets, respectively. We also use two CNN-based denoising methods, i.e., RED30 [34] and MemNet [36], for further comparison. Their PSNR results on BSD68 dataset with noise level 50 are 26.34dB and 26.35dB, respectively. Note that RED30 and MemNet are trained on a specific noise level and are less efficient than DnCNN. From Tables II and III, one can have the following observations.

First, FFDNet surpasses BM3D by a large margin and outperforms WNNM, MLP and TNRD by about 0.2dB for a wide range of noise levels on BSD68. Second, FFDNet is slightly inferior to DnCNN when the noise level is low (e.g., $\sigma \leq 25$), but gradually outperforms DnCNN with the increase of noise level (e.g., $\sigma > 25$). This phenomenon may be resulted from the trade-off between receptive field size and modeling capacity. FFDNet has a larger receptive field than DnCNN, thus favoring for removing strong noise, while DnCNN has better modeling capacity which is beneficial for denoising images with lower noise level. Third, FFDNet outperforms WNNM on images such as "*House*", while it is inferior to WNNM on image "*Barbara*". This is because "*Barbara*" has a rich amount of repetitive structures, which can be effectively exploited by nonlocal self-similarity based WNNM method. The visual comparisons of different methods are given in Fig. 3. Overall, FFDNet produces the best perceptual quality of denoised images.

To evaluate FFDNet-Clip, Table IV shows the PSNR comparison with DCGRF [28] and RBDN [35] on the Clip300 dataset. It can be seen that FFDNet-Clip with matched noise level achieves better performance than DCGRF and RBDN, showing that FFDNet performs well under the clipping setting. We also tested FFDNet-Clip on BSD68 dataset with clipping setting, it has been found that the PSNR result is similar to that of FFDNet with un-clipping setting.

For color image denoising, we compare FFDNet with CBM3D [11] and CDnCNN [20]. Table V reports the performance of different methods on CBSD68, Kodak24, and McMaster datasets, and Fig. 4 presents the visual comparisons. It can be seen that FFDNet consistently outperforms CBM3D on different noise levels in terms of both quantitative and qualitative evaluation, and has competing performance with CDnCNN.

TABLE II
THE PSNR(DB) RESULTS OF DIFFERENT METHODS ON SET12 DATASET WITH NOISE LEVELS 15, 25 35, 50 AND 75. THE BEST TWO RESULTS ARE HIGHLIGHTED IN RED AND BLUE COLORS, RESPECTIVELY

Images	*C.man*	*House*	*Peppers*	*Starfish*	*Monarch*	*Airplane*	*Parrot*	*Lena*	*Barbara*	*Boat*	*Man*	*Couple*	*Average*
Noise Level	$\sigma = 15$												
BM3D	31.91	34.93	32.69	31.14	31.85	31.07	31.37	34.26	33.10	32.13	31.92	32.10	32.37
WNNM	32.17	35.13	32.99	31.82	32.71	31.39	31.62	34.27	33.60	32.27	32.11	32.17	32.70
TNRD	32.19	34.53	33.04	31.75	32.56	31.46	31.63	34.24	32.13	32.14	32.23	32.11	32.50
DnCNN	32.61	34.97	33.30	32.20	33.09	31.70	31.83	34.62	32.64	32.42	32.46	32.47	32.86
FFDNet	32.42	35.01	33.10	32.02	32.77	31.58	31.77	34.63	32.50	32.35	32.40	32.45	32.75
Noise Level	$\sigma = 25$												
BM3D	29.45	32.85	30.16	28.56	29.25	28.42	28.93	32.07	30.71	29.90	29.61	29.71	29.97
WNNM	29.64	33.22	30.42	29.03	29.84	28.69	29.15	32.24	31.24	30.03	29.76	29.82	30.26
MLP	29.61	32.56	30.30	28.82	29.61	28.82	29.25	32.25	29.54	29.97	29.88	29.73	30.03
TNRD	29.72	32.53	30.57	29.02	29.85	28.88	29.18	32.00	29.41	29.91	29.87	29.71	30.06
DnCNN	30.18	33.06	30.87	29.41	30.28	29.13	29.43	32.44	30.00	30.21	30.10	30.12	30.43
FFDNet	30.06	33.27	30.79	29.33	30.14	29.05	29.43	32.59	29.98	30.23	30.10	30.18	30.43
Noise Level	$\sigma = 35$												
BM3D	27.92	31.36	28.51	26.86	27.58	26.83	27.40	30.56	28.98	28.43	28.22	28.15	28.40
WNNM	28.08	31.92	28.75	27.27	28.13	27.10	27.69	30.73	29.48	28.54	28.33	28.24	28.69
MLP	28.08	31.18	28.54	27.12	27.97	27.22	27.72	30.82	27.62	28.53	28.47	28.24	28.46
DnCNN	28.61	31.61	29.14	27.53	28.51	27.52	27.94	30.91	28.09	28.72	28.66	28.52	28.82
FFDNet	28.54	31.99	29.18	27.58	28.54	27.47	28.02	31.20	28.29	28.82	28.70	28.68	28.92
Noise Level	$\sigma = 50$												
BM3D	26.13	29.69	26.68	25.04	25.82	25.10	25.90	29.05	27.22	26.78	26.81	26.46	26.72
WNNM	26.45	30.33	26.95	25.44	26.32	25.42	26.14	29.25	27.79	26.97	26.94	26.64	27.05
MLP	26.37	29.64	26.68	25.43	26.26	25.56	26.12	29.32	25.24	27.03	27.06	26.67	26.78
TNRD	26.62	29.48	27.10	25.42	26.31	25.59	26.16	28.93	25.70	26.94	26.98	26.50	26.81
DnCNN	27.03	30.00	27.32	25.70	26.78	25.87	26.48	29.39	26.22	27.20	27.24	26.90	27.18
FFDNet	27.03	30.43	27.43	25.77	26.88	25.90	26.58	29.68	26.48	27.32	27.30	27.07	27.32
Noise Level	$\sigma = 75$												
BM3D	24.32	27.51	24.73	23.27	23.91	23.48	24.18	27.25	25.12	25.12	25.32	24.70	24.91
WNNM	24.60	28.24	24.96	23.49	24.31	23.74	24.43	27.54	25.81	25.29	25.42	24.86	25.23
MLP	24.63	27.78	24.88	23.57	24.40	23.87	24.55	27.68	23.39	25.44	25.59	25.02	25.07
DnCNN	25.07	27.85	25.17	23.64	24.71	24.03	24.71	27.54	23.63	25.47	25.64	24.97	25.20
FFDNet	25.29	28.43	25.39	23.82	24.99	24.18	24.94	27.97	24.24	25.64	25.75	25.29	25.49

C. Experiments on Spatially Variant AWGN Removal

We then test the flexibility of FFDNet to deal with spatially variant AWGN. To synthesize spatially variant AWGN, we first generate an AWGN image $\boldsymbol{v}_1$ with unit standard deviation and a noise level map $\boldsymbol{M}$ of the same size. Then, element-wise multiplication is applied on $\boldsymbol{v}_1$ and $\boldsymbol{M}$ to produce the spatially variant AWGN, i.e., $\boldsymbol{v} = \boldsymbol{v}_1 \odot \boldsymbol{M}$. In the denoising stage, we take the bilinearly downsampled noise level map as the input to FFDNet. Since the noise level map is spatially smooth, the use of downsampled noise level map generally has very little effect on the final denoising performance.

Fig. 5 gives an example to show the effectiveness of FFDNet on removing spatially variant AWGN. We do not compare FFDNet with other methods because no state-of-the-art AWGN denoising method can be readily extended to handle spatially variant AWGN. From Fig. 5, one can see that FFDNet with non-uniform noise level map is flexible and powerful to remove spatially variant AWGN. In contrast, FFDNet with uniform noise level map would fail to remove strong noise at the region with higher noise level while smoothing out the

TABLE III
THE AVERAGE PSNR(DB) RESULTS OF DIFFERENT METHODS ON BSD68 WITH NOISE LEVELS 15, 25 35, 50 AND 75

Methods	BM3D	WNNM	MLP	TNRD	DnCNN	FFDNet
$\sigma = 15$	31.07	31.37	–	31.42	31.72	31.63
$\sigma = 25$	28.57	28.83	28.96	28.92	29.23	29.19
$\sigma = 35$	27.08	27.30	27.50	–	27.69	27.73
$\sigma = 50$	25.62	25.87	26.03	25.97	26.23	26.29
$\sigma = 75$	24.21	24.40	24.59	–	24.64	24.79

TABLE IV
THE AVERAGE PSNR(DB) RESULTS OF DIFFERENT METHODS ON CLIP300 WITH NOISE LEVELS 15, 25 35, 50 AND 60

Methods	$\sigma = 15$	$\sigma = 25$	$\sigma = 35$	$\sigma = 50$	$\sigma = 60$
DCGRF	31.35	28.67	27.08	25.38	24.45
RBDN	31.05	28.77	27.31	25.80	23.25
FFDNet-Clip	31.68	29.25	27.75	26.25	25.51

TABLE V
THE AVERAGE PSNR(DB) RESULTS OF CBM3D, CDNCNN AND FFDNET ON CBSD68, KODAK24 AND MCMASTER DATASETS WITH NOISE LEVELS 15, 25 35, 50 AND 75

Datasets	Methods	σ=15	σ=25	σ=35	σ=50	σ=75
CBSD68	CBM3D	33.52	30.71	28.89	27.38	25.74
	CDnCNN	33.89	31.23	29.58	27.92	24.47
	FFDNet	33.87	31.21	29.58	27.96	26.24
Kodak24	CBM3D	34.28	31.68	29.90	28.46	26.82
	CDnCNN	34.48	32.03	30.46	28.85	25.04
	FFDNet	34.63	32.13	30.57	28.98	27.27
McMaster	CBM3D	34.06	31.66	29.92	28.51	26.79
	CDnCNN	33.44	31.51	30.14	28.61	25.10
	FFDNet	34.66	32.35	30.81	29.18	27.33

details at the region with lower noise level.

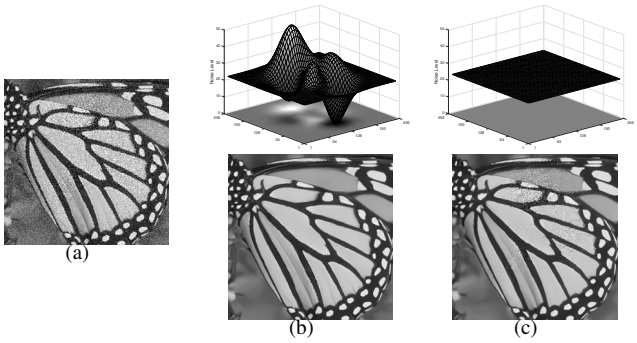


Fig. 5. Examples of FFDNet on removing spatially variant AWGN. (a) Noisy image (20.55dB) with spatially variant AWGN. (b) Ground-truth noise level map and corresponding denoised image (30.08dB) by FFDNet; (c) Uniform noise level map constructed by using the mean value of ground-truth noise level map and corresponding denoised image (27.45dB) by FFDNet.

D. Experiments on Noise Level Sensitivity

In practical applications, the noise level map may not be accurately estimated from the noisy observation, and mismatch between the input and real noise levels is inevitable. If the input noise level is lower than the real noise level, the noise cannot be completely removed. Therefore, users often prefer to set a higher noise level to remove more noise. However, this may also remove some image details together with noise. A practical denoiser should tolerate certain mismatch of noise levels. In this subsection, we evaluate FFDNet in comparison with benchmark BM3D and DnCNN by varying different input noise levels for a given ground-truth noise level.

Fig. 6 illustrates the noise level sensitivity curves of BM3D, DnCNN and FFDNet. Different methods with different input noise levels (e.g., "FFDNet-15" represents FFDNet with input noise level fixed as 15) are evaluated on BSD68 images with noise level ranging from 0 to 50. Fig. 7 shows the visual comparisons between BM3D/CBM3D and FFDNet by setting different input noise levels to denoise a noisy image. Four typical image structures, including flat region, sharp edge, line with high contrast, and line with low contrast, are selected for visual comparison to investigate the noise level sensitivity of BM3D and FFDNet. From Figs. 6 and 7, we have the following observations.

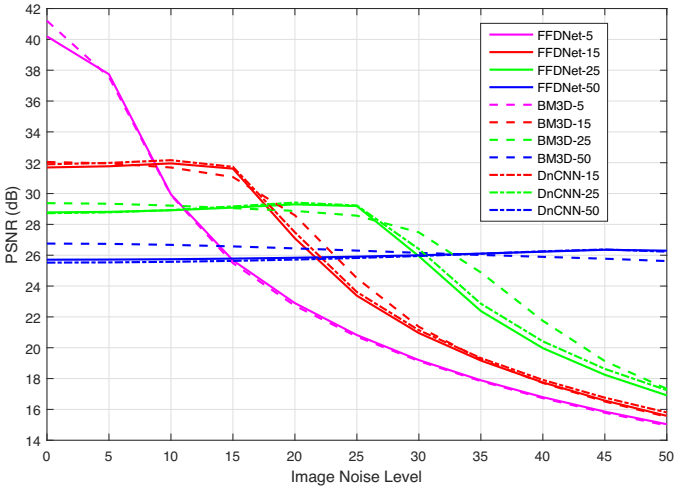


Fig. 6. Noise level sensitivity curves of BM3D, DnCNN and FFDNet. The averaged PSNR results are evaluated on BSD68.

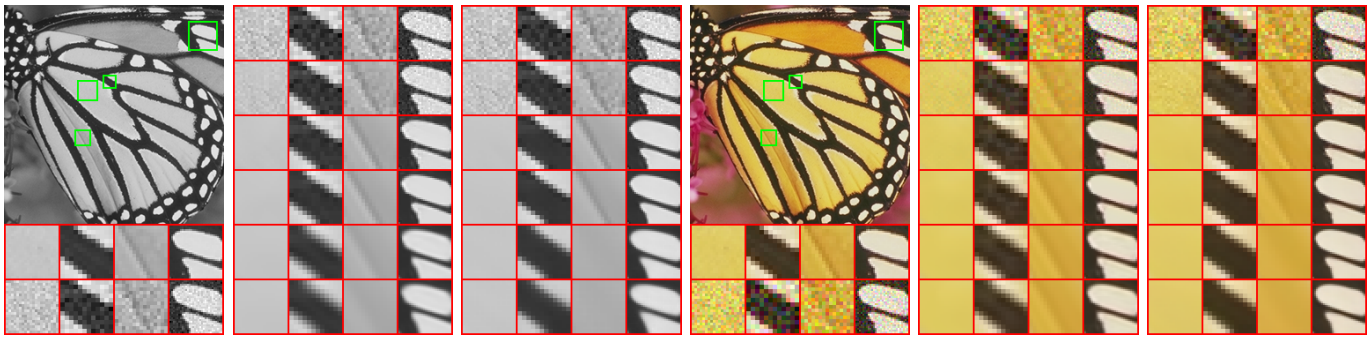
- On all noise levels, FFDNet achieves similar denoising results to BM3D and DnCNN when their input noise levels are the same.
- With the fixed input noise level, for all the three methods, the PSNR value tends to stay the same when the ground-truth noise level is lower, and begins to decrease when the ground-truth noise level is higher.
- The best visual quality is obtained when the input noise level matches the ground-truth one. BM3D and FFDNet produce similar visual results with lower input noise levels, while they exhibit certain difference with higher input noise levels. Both of them will smooth out noise in flat regions, and gradually smooth out image structures with the increase of input noise levels. Particularly, FFDNet may wipe out some low contrast line structure, whereas BM3D can still preserve the mean patch regardless of the input noise levels due to its use of nonlocal information.
- Using a higher input noise level can generally produce better visual results than using a lower one. In addition, there is no much visual difference when the input noise level is a little higher than the ground-truth one.

According to above observations, FFDNet exhibits similar noise level sensitivity performance to BM3D and DnCNN in balancing noise reduction and detail preservation. When the ground-truth noise level is unknown, it is more preferable to set a larger input noise level than a lower one to remove noise with better perceptual quality.

E. Experiments on Real Noisy Images

In this subsection, real noisy images are used to further assess the practicability of FFDNet. However, such an evaluation is difficult to conduct due to the following reasons. (i) Both the ground-truth clean image and noise level are unknown for real noisy image. (ii) The real noise comes from various sources such as camera imaging pipeline (e.g., shot noise, amplifier noise and quantization noise), scanning, lossy compression and image resizing [62], [63], and it is generally non-Gaussian, spatially variant, and signal-dependent. As a result, the AWGN assumption in many denoising algorithms does not hold, and the associated noise level estimation methods do not work well for real noisy images.

Instead of adopting any noise level estimation methods, we adopt an interactive strategy to handle real noisy images. First of all, we empirically found that the assumption of spatially invariant noise usually works well for most real noisy images. We then employ a set of typical input noise levels to produce multiple outputs, and select the one which has best trade-off between noise reduction and detail preservation. Second, the spatially variant noise in most real-world images is signal-dependent. In this case, we first sample several typical regions of distinct colors. For each typical region, we apply different



(a) (b) (c) (d) (e) (f)

Fig. 7. Visual comparisons between FFDNet and BM3D/CBM3D by setting different input noise levels to denoise a noisy image. (a) From top to bottom: ground-truth image, four clean zoom-in regions, and the corresponding noisy regions (AWGN, noise level 15). (b) From top to bottom: denoising results by BM3D with input noise levels 5, 10, 15, 20, 50, and 75, respectively. (c) Results by FFDNet with the same settings as in (b). (d) From top to bottom: ground-truth image, four clean zoom-in regions, and the corresponding noisy regions (AWGN, noise level 25). (e) From top to bottom: denoising results by CBM3D with input noise levels 10, 20, 25, 30, 45 and 60, respectively. (f) Results by FFDNet with the same settings as in (e).

noise levels with an interval of 5, and choose the best noise level by observing the denoising results. The noise levels at other regions are then interpolated from the noise levels of the typical regions to constitute an approximated non-uniform noise level map. Our FFDNet focuses on non-blind denoising and assumes the noise level map is known. In practice, some advanced noise level estimation methods [62], [64] can be adopted to assist the estimation of noise level map. In our following experiments, unless otherwise specified, we assume spatially invariant noise for the real noisy images.

Since there is no ground-truth image for a real noisy image, visual comparison is employed to evaluate the performance of FFDNet. We choose BM3D for comparison because it is widely accepted as a benchmark for denoising applications. Given a noisy image, the same input noise level is used for BM3D and FFDNet. Another CNN-based denoising method DnCNN and a blind denoising method Noise Clinic [56] are also used for comparison. Note that, apart from the non-blind DnCNN models for specific noise levels, the blind DnCNN model (i.e., DnCNN-B) trained on noise level range of $[0, 55]$ is also used for grayscale image denoising. For color image denoising, the blind CDnCNN-B is used for comparison.

Fig. 8 compares the grayscale image denoising results of Noise Clinic, BM3D, DnCNN, DnCNN-B and FFDNet on RNI6 images. As one can see, Noise Clinic reduces much the noise, but it also generates many algorithm-induced artifacts. BM3D, DnCNN and FFDNet produce more visually pleasant results. While the non-blind DnCNN models perform favorably, the blind DnCNN-B model performs poorly in removing the non-AWGN real noise. This phenomenon clearly demonstrates the better generalization ability of non-blind model over blind one for controlling the trade-off between noise removal and detail preservation. It is worth noting that, for image "*Building*" which contains structured noise, Noise Clinic and BM3D fail to remove those structured noises since the structured noises fit the nonlocal self-similarity prior adopted in Noise Clinic and BM3D. In contrast, FFDNet and DnCNN successfully remove such noise without losing underlying image textures.

Fig. 9 shows the denoising results of Noise Clinic, CBM3D, CDnCNN-B and FFDNet on five color noisy images from RNI15. It can be seen that CDnCNN-B yields very pleasing results for noisy image with AWGN-like noise such as image "*Frog*", and is still unable to handle non-AWGN noise. Notably, from the denoising results of "*Boy*", one can see that CBM3D remains the structured color noise unremoved whereas FFDNet removes successfully such kind of noise. We can conclude that while the nonlocal self-similarity prior helps to remove random noise, it hinders the removal of structured noise. In comparison, the prior implicitly learned by CNN is able to remove both random noise and structured noise.

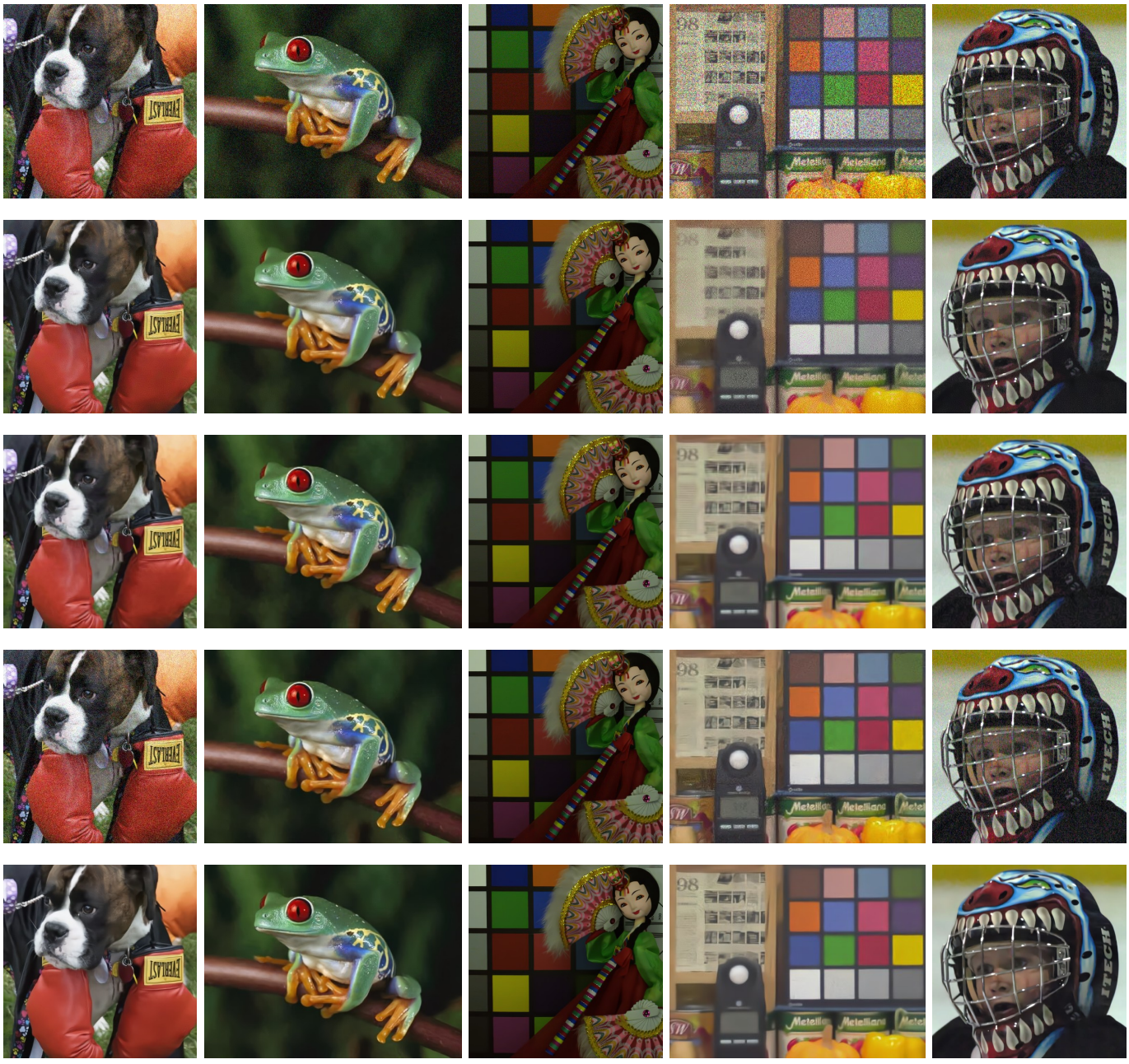
Fig. 10 further shows more visual results of FFDNet on the other nine images from RNI15. It can be seen that FFDNet can handle various kinds of noises, such as JPEG lossy compression noise (see image "*Audrey Hepburn*"), and video noise (see image "*Movie*").

Fig. 11 shows a more challenging example to demonstrate the advantage of FFDNet for denoising noisy images with spatially variant noise. We select five typical regions to estimate the noise levels, including two background regions, the coffee region, the milk-foam region, and the specular reflection region. In our experiment, we manually and interactively set σ = 10 for the milk-foam and specular reflection regions, σ = 35 for the background region with high noise (i.e., green region), and σ = 25 for the other regions. We then interpolate the non-uniform noise level map for the whole image based on the estimated five noise levels. As one can see, while FFDNet with a small uniform input noise level can recover the details of regions with low noise level, it fails to remove strong noise. On the other hand, FFDNet with a large uniform input noise level can remove strong noise but it will also smooth out the details in the region with low noise level. In contrast, the denoising result with a proper non-uniform noise level map not only preserves image details but also removes the strong noise.

Finally, according to the above experiments on real noisy images, we can see that the FFDNet model trained with unquantized image data performs well on 8-bit quantized real noisy images.

(a) *David Hilbert* (b) *Old Tom Morris* (c) *Chupa Chups* (d) *Vinegar* (e) *Building* (f) *Marilyn*

Fig. 8. Grayscale image denoising results by different methods on real noisy images. From top to bottom: noisy images, denoised images by Noise Clinic, denoised images by BM3D, denoised images by DnCNN, denoised images by DnCNN-B, denoised images by FFDNet. (a) σ = 14 (15 for DnCNN); (b) σ = 15; (c) σ = 10; (d) σ = 20; (e) σ = 20; (f) σ = 7 (10 for DnCNN).



(a) *Dog* (b) *Frog* (c) *Pattern1* (d) *Pattern2* (e) *Boy*

Fig. 9. Color image denoising results by different methods on real noisy images. From top to bottom: noisy images, denoised images by Noise Clinic, denoised images by CBM3D, denoised images by CDnCNN-B, denoised images by FFDNet. (a) σ = 28; (b) σ = 15; (c) σ = 12; (d) σ = 40; (e) σ = 45.

F. Running Time

Table VI lists the running time results of BM3D, DnCNN and FFDNet for denoising grayscale and color images with size 256×256, 512×512 and 1,024×1,024. The evaluation was performed in Matlab (R2015b) environment on a computer with a six-core Intel(R) Core(TM) i7-5820K CPU @ 3.3GHz, 32 GB of RAM and an Nvidia Titan X Pascal GPU. For BM3D, we evaluate its running time by denoising images with noise level 25. For DnCNN, the grayscale and color image denoising models have 17 and 20 convolution layers, respectively. The Nvidia cuDNN-v5.1 deep learning library is used to accelerate the computation of DnCNN and FFDNet. The memory transfer time between CPU and GPU is also counted. Note that DnCNN and FFDNet can be implemented with both single-threaded (ST) and multi-threaded (MT) CPU computations.

这次又计算在内了?

From Table VI, we have the following observations. First, BM3D spends much more time on denoising color images than grayscale images. The reason is that, compared to gray-BM3D, CBM3D needs extra time to denoise the chrominance components after luminance-chrominance color transformation. Second, while DnCNN can benefit from GPU computation for

TABLE VI
RUNNING TIME (IN SECONDS) OF DIFFERENT METHODS FOR DENOISING IMAGES WITH SIZE 256×256, 512×512 AND 1,024×1,024

Methods	Device	256×256		512×512		1,024×1,024	
		Gray	Color	Gray	Color	Gray	Color
BM3D	CPU(ST)	0.59	0.98	2.52	3.57	10.77	20.15
DnCNN	CPU(ST)	2.14	2.44	8.63	9.85	32.82	38.11
	CPU(MT)	0.74	0.98	3.41	4.10	12.10	15.48
	GPU	0.011	0.014	0.033	0.040	0.124	0.167
FFDNet	CPU(ST)	0.44	0.62	1.81	2.51	7.24	10.17
	CPU(MT)	0.18	0.21	0.73	0.98	2.96	3.95
	GPU	0.006	0.008	0.012	0.017	0.038	0.057

fast implementation, it has comparable CPU time to BM3D. Third, FFDNet spends almost the same time for processing grayscale and color images. More specifically, FFDNet with multi-threaded implementation is about three times faster than DnCNN and BM3D on CPU, and much faster than DnCNN on GPU. Even with single-threaded implementation, FFDNet is also faster than BM3D. Taking denoising performance and flexibility into consideration, FFDNet is very competitive for practical applications.

V. CONCLUSION

In this paper, we proposed a new CNN model, namely FFDNet, for fast, effective and flexible discriminative denoising. To achieve this goal, several techniques were utilized in network design and training, such as the use of noise level map as input and denoising in downsampled sub-images space. The results on synthetic images with AWGN demonstrated that FFDNet can not only produce state-of-the-art results when input noise level matches ground-truth noise level, but also have the ability to robustly control the trade-off between noise reduction and detail preservation. The results on images with spatially variant AWGN validated the flexibility of FFDNet for handing inhomogeneous noise. The results on real noisy images further demonstrated that FFDNet can deliver perceptually appealing denoising results. Finally, the running time comparisons showed the faster speed of FFDNet over other competing methods such as BM3D. Considering its flexibility, efficiency and effectiveness, FFDNet provides a practical solution to CNN denoising applications.

REFERENCES

[1] H. C. Andrews and B. R. Hunt, "Digital image restoration," *Prentice-Hall Signal Processing Series, Englewood Cliffs: Prentice-Hall, 1977*, vol. 1, 1977.

[2] P. Chatterjee and P. Milanfar, "Is denoising dead?" *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 895–911, 2010.

[3] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 860–867.

[4] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *IEEE International Conference on Computer Vision*, 2011, pp. 479–486.

[5] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2862–2869.

[6] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2345–2356, 2010.

[7] F. Heide, M. Steinberger, Y.-T. Tsai, M. Rouf, D. Pajak, D. Reddy, O. Gallo, J. Liu, W. Heidrich, K. Egiazarian *et al.*, "FlexISP: A flexible camera image processing framework," *ACM Transactions on Graphics*, vol. 33, no. 6, p. 231, 2014.

[8] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *submitted to SIAM Journal on Imaging Sciences*, 2016.

[9] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3929–3938.

[10] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of gaussians in the wavelet domain," *IEEE Transactions on Image processing*, vol. 12, no. 11, pp. 1338–1351, 2003.

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

[12] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *IEEE International Conference on Computer Vision*, 2009, pp. 2272–2279.

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

[14] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[15] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, 2008.

[16] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 60–65.

[17] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1256–1272, 2017.

[18] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2392–2399.

[19] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Advances in Neural Information Processing Systems*, 2009, pp. 769–776.

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

[21] A. Barbu, "Training an active random field for real-time image denoising," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2451–2462, 2009.

[22] K. G. Samuel and M. F. Tappen, "Learning optimized MAP estimates in continuously-valued MRF models," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 477–484.

[23] J. Sun and M. F. Tappen, "Learning non-local range markov random field for image restoration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2745–2752.

[24] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2774–2781.

[25] U. Schmidt, "Half-quadratic inference and learning for natural images," Ph.D. dissertation, Technische Universität, Darmstadt, 2017. [Online]. Available: http://tuprints.ulb.tu-darmstadt.de/6044/

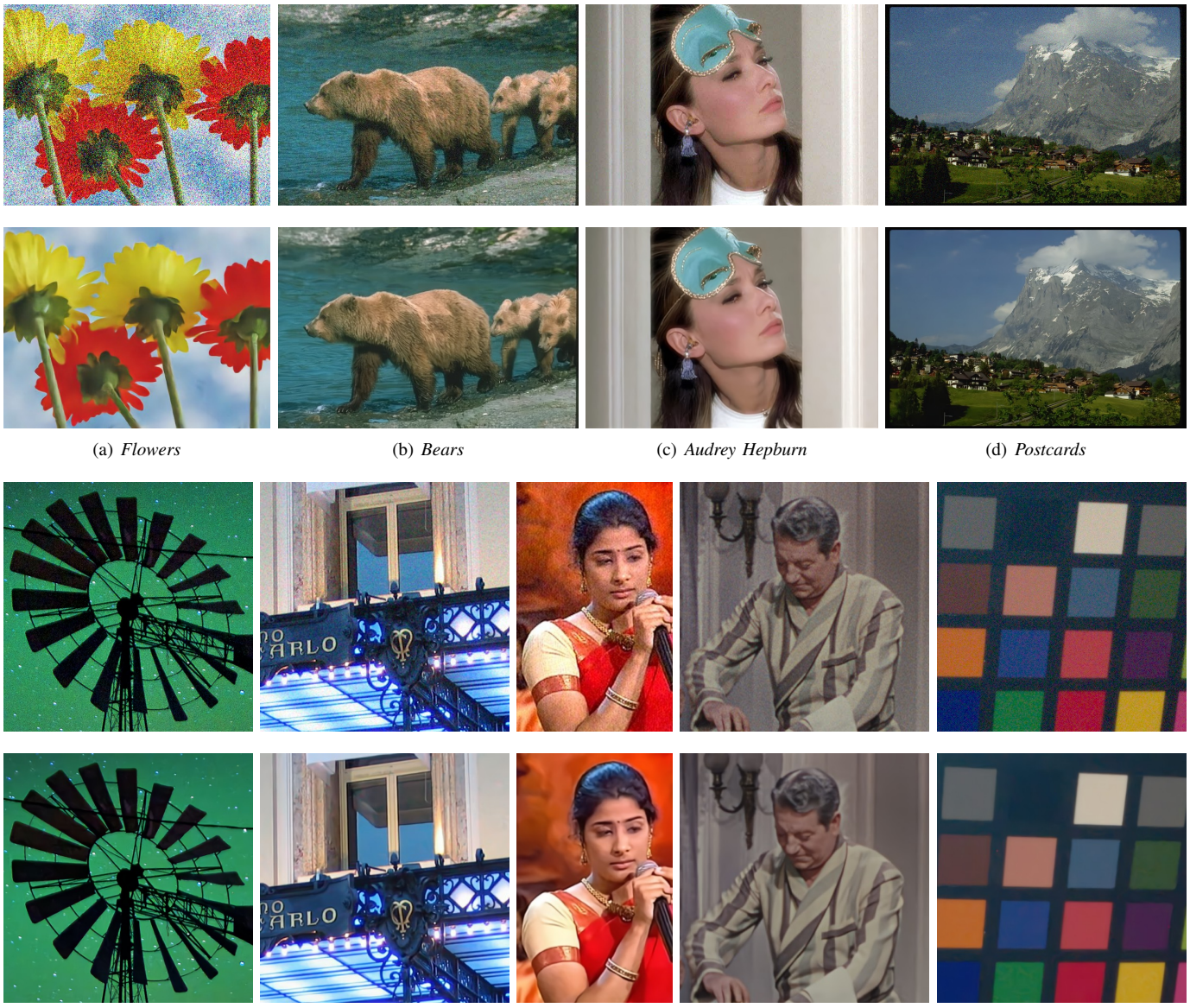
[26] S. Lefkimmiatis, "Non-local color image denoising with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3587–3596.

[27] P. Qiao, Y. Dou, W. Feng, R. Li, and Y. Chen, "Learning non-local image diffusion for image denoising," in *Proceedings of the 2017 ACM on Multimedia Conference*, 2017, pp. 1847–1855.

[28] R. Vemulapalli, O. Tuzel, and M.-Y. Liu, "Deep gaussian conditional random field network: A model-based deep network for discriminative denoising," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

[29] J. Kruse, C. Rother, and U. Schmidt, "Learning to push the limits of efficient FFT-based image deconvolution," in *IEEE International Conference on Computer Vision*, Oct 2017.

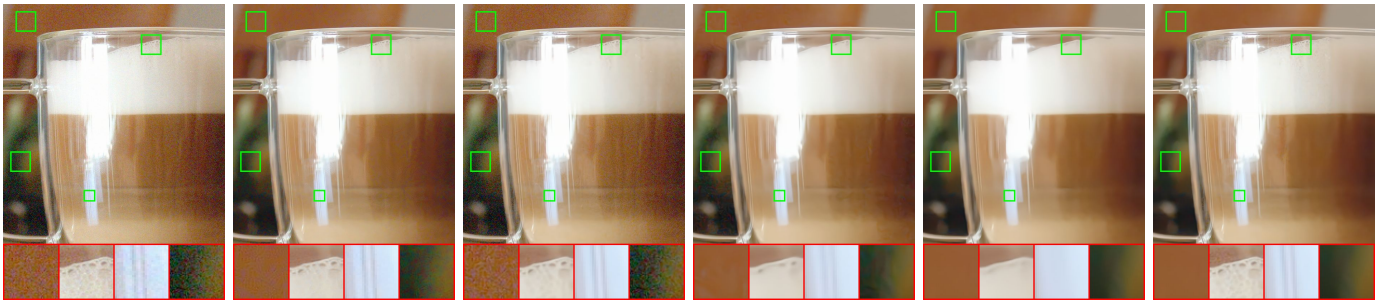
[30] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 341–349.



(a) *Flowers* (b) *Bears* (c) *Audrey Hepburn* (d) *Postcards*

(e) *Stars* (f) *Window* (g) *Singer* (h) *Movie* (i) *Pattern3*

Fig. 10. More denoising results of FFDNet on real image denoising. (a) σ = 70; (b) σ = 15; (c) σ = 10; (d) σ = 15; (e) σ = 18; (f) σ = 15; (g) σ = 30; (h) σ = 12; (i) σ = 25.



(a) (b) (c) (d) (e) (f)

Fig. 11. An example of FFDNet on image "*Glass*" with spatially variant noise. (a) Noisy image; (b) Denoised image by Noise Clinic; (c) Denoised image by FFDNet with σ = 10; (d) Denoised image by FFDNet with σ = 25; (e) Denoised image by FFDNet with σ = 35; (f) Denoised image by FFDNet with non-uniform noise level map.

[31] F. Agostinelli, M. R. Anderson, and H. Lee, "Robust image denoising with multi-column deep neural networks," in *Advances in Neural Information Processing Systems*, 2013, pp. 1493–1501.

[32] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[33] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *International Conference on Learning Representations*, 2016.

[34] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in Neural Information Processing Systems*, 2016, pp. 2802–2810.

[35] V. Santhanam, V. I. Morariu, and L. S. Davis, "Generalized deep image to image regression," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5609–5619.

[36] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *IEEE International Conference on Computer Vision*, 2017, pp. 4539–4547.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[38] S. Nam, Y. Hwang, Y. Matsushita, and S. Joo Kim, "A holistic approach to cross-channel image noise modeling and its application to image denoising," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

[39] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.

[40] A. Levin and B. Nadler, "Natural image denoising: Optimality and inherent bounds," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2833–2840.

[41] D. Wang, P. Cui, M. Ou, and W. Zhu, "Deep multimodal hashing with orthogonal regularization." in *International Joint Conference on Artificial Intelligence*, 2015, pp. 2291–2297.

[42] Z. Mhammedi, A. Hellicar, A. Rahman, and J. Bailey, "Efficient orthogonal parametrisation of recurrent neural networks using householder reflections," *arXiv preprint arXiv:1612.00188*, 2016.

[43] K. Jia, "Improving training of deep neural networks via singular value bounding," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4344–4352.

[44] D. Xie, J. Xiong, and S. Pu, "All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6176–6185.

[45] Y. Sun, L. Zheng, W. Deng, and S. Wang, "SVDNet for pedestrian retrieval," *arXiv preprint arXiv:1703.05693*, 2017.

[46] D. Mishkin and J. Matas, "All you need is a good init," *ArXiv e-prints*, 2015.

[47] G. Riegler, S. Schulter, M. Ruther, and H. Bischof, "Conditioned regression models for non-blind single image super-resolution," in *IEEE International Conference on Computer Vision*, 2015, pp. 522–530.

[48] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-Play ADMM for image restoration: Fixed-point convergence and applications," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 84–98, 2017.

[49] S. Zagoruyko and N. Komodakis, "Diracnets: training very deep neural networks without skip-connections," *arXiv preprint arXiv:1706.00388*, 2017.

[50] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference for Learning Representations*, 2015.

[51] T. Plotz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[52] J.-S. Lee, "Refined filtering of image noise using local statistics," *Computer graphics and image processing*, vol. 15, no. 4, pp. 380–389, 1981.

[53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[54] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang, "Waterloo exploration database: New challenges for image quality assessment models," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, 2017.

[55] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for matlab," in *ACM Conference on Multimedia Conference*, 2015, pp. 689–692.

[56] M. Lebrun, M. Colom, and J.-M. Morel, "The noise clinic: A blind image denoising algorithm," *Image Processing On Line*, vol. 5, pp. 1–54, 2015. [Online]. Available: http://demo.ipol.im/demo/125/

[57] 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. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.

[58] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan 2015.

[59] R. Franzen, "Kodak lossless true color image suite," *source: http://r0k.us/graphics/kodak*, vol. 4, 1999.

[60] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *Journal of Electronic Imaging*, vol. 20, no. 2, pp. 1–15, 2011.

[61] [Online]. Available: https://ni.neatvideo.com/home

[62] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman, "Automatic estimation and removal of noise from a single image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 299–314, 2008.

[63] M. Colom, M. Lebrun, A. Buades, and J.-M. Morel, "A non-parametric approach for the estimation of intensity-frequency dependent noise," in *IEEE International Conference on Image Processing*, 2014, pp. 4261–4265.

[64] L. Azzari and A. Foi, "Gaussian-cauchy mixture modeling for robust signal-dependent noise estimation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 5357–5361.