Deep Boosting for Image Denoising

Chang Chen, Zhiwei Xiong(✉), Xinmei Tian, and Feng Wu

University of Science and Technology of China
changc@mail.ustc.edu.cn, {zwxiong,xinmei,fengwu}@ustc.edu.cn

Abstract. Boosting is a classic algorithm which has been successfully applied to diverse computer vision tasks. In the scenario of image denoising, however, the existing boosting algorithms are surpassed by the emerging learning-based models. In this paper, we propose a novel deep boosting framework (DBF) for denoising, which integrates several convolutional networks in a feed-forward fashion. Along with the integrated networks, however, the depth of the boosting framework is substantially increased, which brings difficulty to training. To solve this problem, we introduce the concept of dense connection that overcomes the vanishing of gradients during training. Furthermore, we propose a path-widening fusion scheme cooperated with the dilated convolution to derive a lightweight yet efficient convolutional network as the boosting unit, named Dilated Dense Fusion Network (DDFN). Comprehensive experiments demonstrate that our DBF outperforms existing methods on widely used benchmarks, in terms of different denoising tasks.

1 Introduction

Image denoising is a typical problem in low-level computer vision. Observed a contaminated image with a certain kind of noise (e.g., additive white Gaussian noise), plenty of methods have been investigated to restore the original signal. Among them, modeling image priors for restoration is a prominent approach, such as nonlocal similarity based models [1–3] and sparsity based models [4–6]. Specifically, BM3D [7], CSF [8], and WNNM [9] are several representative methods for image denoising.

Recently, with the rapid advancement of GPU-based parallel computing frameworks, increasingly more learning-based denoising models [10–13] began to adopt the paradigm of end-to-end training based on a convolutional neural network (CNN). These learning-based models have achieved competitive or even better performance than previous methods. On the other hand, several traditional models [14–16] based on the boosting algorithm studied the denoising problem in a unique perspective. By extracting the residual signal or eliminating the noise leftover, these methods boost the restoration quality iteratively. Beyond them, Romano and Elad proposed a notable variant of the boosting algorithm, named Strengthen-Operate-Subtract (SOS) [17]. By combining the denoised image with the original input, it increased the signal-to-noise ratio iteratively and achieved promising improvements.

Nevertheless, the existing boosting algorithms still have performance gaps compared with the learning-based models. In this paper, we embed the deep learning technique into the boosting algorithm and significantly boost its performance in the scenario of image denoising. Specifically, we construct a Deep Boosting Framework (DBF) that integrates several CNNs in a feed-forward fashion, where each network serves as a boosting unit. To the best of our knowledge, this is the first time that deep learning and boosting are jointly investigated for image restoration. Although this paper mainly focuses on denoising, the proposed DBF can be readily generalized to other image restoration tasks.

Theoretically, the boosting unit in the DBF can be any type of networks. In practice, however, we find that not all structures are suitable to be employed as the boosting unit. The reason is that, along with the integrated networks for boosting, the depth of the DBF is substantially increased. It thus causes the difficulty of convergence during training. To fully exploit the potential of the DBF, we further propose a Dilated Dense Fusion Network (DDFN), which is highly optimized to serve as the boosting unit.

We reform the plainly connected network structure in three steps to obtain the DDFN. First, to overcome the vanishing of gradients during training, we introduce the dense connection [18] to construct the boosting unit, which also improves the re-usage of features and thus guarantees the efficiency. Second, to obtain better performance based on the densely connected structure, we adopt the dilated convolution [19] for widening the receptive field without additive parameters, which maintains the lightweight structure of the boosting unit. Last but not least, we further propose a path-widening fusion scheme cooperated with the dilated convolution to make the boosting unit more efficient.

The contributions of this paper are summarized as follows:

1) We propose a novel boosting framework for image denoising by introducing deep learning into the boosting algorithm, named DBF. It not only outperforms existing boosting algorithms by a large margin but also performs better than extensive learning-based models.

2) We optimize a lightweight yet efficient convolutional network as the boosting unit, named DDFN. With the dense connection, we address the difficulty of convergence in DBF. Cooperating with the dilated convolution, we propose a path-widening fusion scheme to expand the capacity of each boosting unit.

3) Our DDFN-based DBF has a clear advantage over existing methods on widely used benchmarks when trained at a specific noise level. If trained for blind Gaussian denoising, it achieves a new state-of-the-art performance within a wide range of noise levels. Also, the proposed method is demonstrated effective when generalized to the image deblocking task.

2 Related Work

CNN-based image denoising. Research along this direction focuses on the exploration of the network structure. Advanced design of architecture yields better restoration quality. For instance, Burger et al. [10] trained a multi-layer perceptrons (MLPs) with a large image database, which achieved comparable results with BM3D [7]. Chen et al. [11] proposed a stage-wise model (i.e., TNRD) which introduced the well-designed convolutional layers into the non-linear diffusion model to derive a flexible framework. And Zhang et al. [12] composed the deep DnCNN model by utilizing batch normalization (BN) [20] and residual connection [21]. Essentially, DnCNN can be viewed as the generalization of one-stage TNRD. Besides, a recently proposed model [13] combined image denoising with semantic classification using CNNs, which bridged the gap between these two different tasks and improved the restoration quality. Following the successful paradigm of end-to-end training, we also adopt the CNN for image denoising. Different from its common usages, however, the employed CNN is just a component of our denoising model. Specifically, it is integrated as a boosting unit in the DBF. Experimental results demonstrate the superior performance of our boosting framework compared with a single CNN model.

Boosting algorithm. Boosting is a widely used algorithm to improve the performance of diverse tasks by cascading several steerable sub-models. A plenty of models based on the boosting algorithm have been investigated for image denoising in literature [14–16], [22]. Generally, the detailed implementation can be divided into 3 classes: (a) re-utilizing the residual [14], (b) re-enhancing the denoised signal [15], and (c) strengthening the SNR iteratively [17]. However, these boosting algorithms with classic models are surpassed by the emerging learning-based models. Contrastively, our proposed DBF inherits both advantages of boosting and CNN and achieves a new state-of-the-art performance for image denoising. Note that, boosting and CNN have been combined for image classification tasks before, e.g., IB-CNN [23] and BoostCNN [24], yet our proposed DBF is the first deep boosting framework in the field of image restoration.

3 Deep Boosting Framework

3.1 Boosting Perspective of Denoising

The fundamental image denoising problem is the recovery of an image $x \in \mathbb{R}^{N\times M}$ from a contaminated measurement $y \in \mathbb{R}^{N\times M}$, which can be formulated as

$$y = x + v, \tag{1}$$

where v stands for the additive noise that is generally modeled as zero-mean white Gaussian noise with a standard deviation σ. The denoising process can be represented as

$$\hat{x} = \mathcal{S}(y) = \mathcal{S}(x + v), \tag{2}$$

where the operator $S(\cdot)$ stands for a general denoising method and $\hat{x}$ stands for an approximation of x.

However, the image $\hat{x}$ recovered by any algorithm cannot ideally equal to x, and the gap between them can be denoted as

$$u = \hat{x} - x = v_r - x_r, \tag{3}$$

where x_r represents the unrecovered signal and v_r stands for the leftover noise in $\hat{x}$. In other words, by adding x_r and subtracting v_r, we then obtain the clean image x from $\hat{x}$.

A straightforward idea to apply the boosting algorithm to image denoising is that we iteratively extract the unrecovered signal x_r from the residual and add it back to $\hat{x}$

$$\hat{x}^{n+1} = \hat{x}^n + \mathcal{H}(y - \hat{x}^n), \tag{4}$$

where $\mathcal{H}(\cdot)$ is an operator for the iterative extraction and we set $\hat{x}^0 = 0$. Note that, however, the residual $y - \hat{x}$ contains not only the unrecovered signal x_r but also a part of noise

$$\begin{aligned} y - \hat{x} &= (x + v) - (x + u) \\ &= (x + v) - (x + v_r - x_r) \\ &= x_r + (v - v_r). \end{aligned} \tag{5}$$

Another native idea is that we remove the leftover noise v_r by filtering the denoised image $\hat{x}$ iteratively

$$\hat{x}^{n+1} = \mathcal{F}(\hat{x}^n), \tag{6}$$

where $\mathcal{F}(\cdot)$ stands for a certain denoising model. However, it could lead to over-smoothing since it neglects x_r which contains most high frequency information.

To further improve the performance of the boosting framework, Romano and Elad proposed a novel SOS algorithm. The denoising target in each iteration step is the "strengthened" image $y + \hat{x}^n$, instead of the residual $y - \hat{x}^n$ or the denoised image $\hat{x}^n$, which improves the signal-to-noise ratio (SNR) [17]. To guarantee the iterability of SOS, however, it has to "subtract" the identical $\hat{x}^n$ in each step as

$$\hat{x}^{n+1} = \mathcal{G}(y + \hat{x}^n) - \hat{x}^n, \tag{7}$$

where $\mathcal{G}(\cdot)$ is a certain denoising model imposed on the strengthened image. To better clarify the insight of the SOS algorithm, we decompose $y + \hat{x}$ as

$$\begin{aligned} y + \hat{x} &= (x + v) + (x + u) \\ &= 2x + (v + u). \end{aligned} \tag{8}$$

Assuming that $||u|| = \delta||v||$, where $\delta \ll 1$. Then we have $SNR(y + \hat{x}) > SNR(y)$ according to the Cauchy-Schwarz inequality [17]. All we need to achieve so is a general denoising method even if it is a "weak" one.

3.2 CNN-Based Boosting

Inspired by SOS [17], we propose a new boosting framework by leveraging deep learning. Specifically, we introduce a CNN to learn the denoising model in each stage. Following Eq. (7), we have

$$\hat{x}^{n+1} = \mathcal{G}_\theta(y + \hat{x}^n) - \hat{x}^n, \tag{9}$$

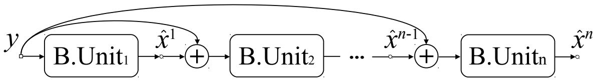


Fig. 1. CNN-based deep boosting framework. The B.Unit$_n$ denotes the n^{th} boosting unit (i.e., $\mathcal{G}_{\theta_n}$) in the framework. The investigation of B.Unit is detailed in Sec. 4

where θ denotes the trainable parameter set of the CNN.

The subtraction of identical $\hat{x}^n$ inherited from Eq. (7) aims to guarantee the iterability of the SOS algorithm. Such constraint in Eq. (9) is no longer needed since we can learn different denoising models in each stage. In other words, our deep boosting framework can adjust its parameters without the constraint of identical subtraction, which actually yields a better performance as will be demonstrated in Sec. 5.3. The output of the final stage can be represented as

$$\hat{x}^n = \mathcal{G}_{\theta_n}\{y + \mathcal{G}_{\theta_{n-1}}[y + \cdots \mathcal{G}_{\theta_2}(y + \mathcal{G}_{\theta_1}(y)) \cdots]\}, \tag{10}$$

where n stands for the serial number of each stage. Fig. 1 illustrates a flowchart for Eq. (10) for a better understanding.

The loss function for training the parameters $\Theta = \{\theta_1, \theta_2, ..., \theta_n\}$ is the mean square error (MSE) between the final output $\hat{x}^n$ and the ground truth x

$$\mathcal{L}_{\Theta}(\hat{x}^n, x) = \frac{1}{2B}\sum_{i=1}^{B} ||\hat{x}_i^n - x_i||_2^2, \tag{11}$$

where B denotes the size of mini-batch for the stochastic gradient descent. Such training scheme is called joint training which optimizes the parameters in all stages simultaneously. We also consider a greedy training scheme, for which the parameters are firstly optimized stage-wise and then fine-tuned among all stages. Related experimental results will be described in Sec. 5.3.

3.3 Relationship to TNRD

The TNRD model proposed in [11] is also a stage-wise model trained jointly, which can be formulated as

$$\hat{x}^n - \hat{x}^{n-1} = -\mathcal{D}(\hat{x}^{n-1}) - \mathcal{R}(\hat{x}^{n-1}, y), \tag{12}$$

where $\mathcal{D}(\cdot)$ stands for the diffusion term which is implemented using a CNN with two layers and $\mathcal{R}(\cdot)$ denotes the reaction term as $\mathcal{R}(\hat{x}^{n-1}, y) = \gamma(\hat{x}^{n-1} - y)$, where γ is a factor which denotes the strength of the reaction term.

Actually, TNRD can be interpreted as a special case of the boosting algorithm. Combining Eq. (4) and Eq. (6), we have

$$\hat{x}^n = \hat{x}^{n-1} + \mathcal{H}(y - \hat{x}^{n-1}) + \mathcal{F}(\hat{x}^{n-1}). \tag{13}$$

Providing $\mathcal{F}(\cdot) = -\mathcal{D}(\cdot)$ and $\mathcal{H}(\cdot) = -\gamma(\cdot)$, we then obtain the basic equation of the TNRD model.

Table 1. From the plain structure to the dilated dense fusion: the evolution of structure for the boosting unit. Part 1 is the feature extraction stage, part 2 is the feature integration stage, and part 3 is the reconstruction stage. $C(\cdot)$ stands for the convolution with (kernel size × kernel size × number of filters) and $D(\cdot)$ denotes the corresponding parameters of dilated convolution. $[\cdot]\times$ or $\{\cdot\}\times$ stands for an operator of concatenation with certain blocks (the ×1 is omitted). And the symbol "/" denotes the path-widening fusion. Detail structures are illustrated in Fig. 2 for a better understanding

Parts	PN	DN	DDN	DDFN
1	$C(3\times3\times24)$	$C(3\times3\times32)$	$\begin{bmatrix} C(3\times3\times16) \\ D(3\times3\times16) \end{bmatrix}$	$\begin{bmatrix} C(3\times3\times8/6) \\ D(3\times3\times8/6) \end{bmatrix}$
2	$C(3\times3\times24)\times8$	$\begin{bmatrix} C(1\times1\times32) \\ C(3\times3\times8) \end{bmatrix}\times8$	$\begin{bmatrix} C(1\times1\times32) \\ D(3\times3\times8) \end{bmatrix}\times8$	$\left\{\begin{matrix} C(1\times1\times24) \\ \begin{bmatrix} C(3\times3\times6/3) \\ D(3\times3\times6/3) \end{bmatrix} \\ C(1\times1\times8) \end{matrix}\right\}\times8$
3	$C(3\times3\times1)$	$C(1\times1\times1)$	$C(1\times1\times1)$	$C(1\times1\times1)$

However, by further decomposing Eq. (12), we demonstrate the fundamentally different insights between TNRD and DBF as follows. Without loss of generality, let $\hat{x}^{n-1} = x + u$ and we discuss a special case when $\gamma = 1$. Considering Eq. (1) and Eq. (3), we have

$$\begin{aligned} \mathcal{R}(\hat{x}^{n-1}, y) &= \hat{x}^{n-1} - y \\ &= (x+u) - (x+v) \\ &= u - v. \end{aligned} \tag{14}$$

Substituting Eq. (14) into Eq. (12), we then have

$$\begin{aligned} \hat{x}^{n} &= \hat{x}^{n-1} - \mathcal{D}(\hat{x}^{n-1}) - \mathcal{R}(\hat{x}^{n-1}, y) \\ &= (x+u) - \mathcal{D}(\hat{x}^{n-1}) - (u - v) \\ &= x - \mathcal{D}(\hat{x}^{n-1}) + v. \end{aligned} \tag{15}$$

The target of TNRD is to let $\hat{x}^n \to x$, i.e., $\mathcal{D}(\hat{x}^{n-1}) \to v$. Thus, the diffusion term is actually trained for fitting the white Gaussian noise v. In contrast, our proposed DBF is trained for directly restoring the original signal x, leveraging on the availability of denoised images and the growth of SNR. Intuitively, it may be more difficult to find correlations between training examples when fitting the irregular noise. Moreover, from the perspective of SNR, it is more difficult to predict the "weaker" noise when the input image has a lower noise level. These are the advantages of our DBF in comparison with TNRD.

4 Dilated Dense Fusion Network

An efficient boosting unit is desired to fully exploit the potential of the proposed DBF. Theoretically, the function $\mathcal{G}_\theta(\cdot)$ in Eq. (10) has no restriction on the

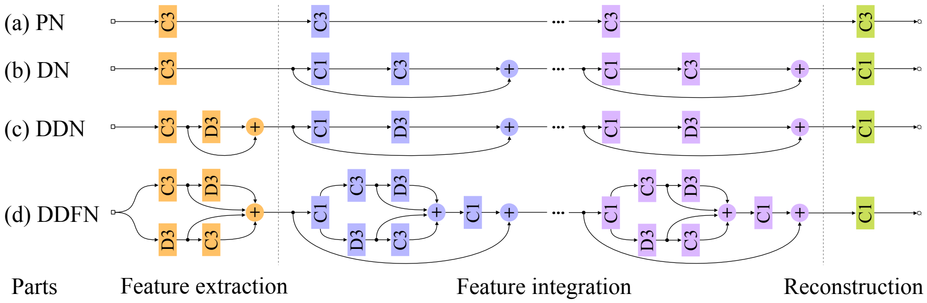


Fig. 2. Details of the evolution for the boosting unit. "C" and "D" with a rectangular block denote the convolution and its dilated variant, respectively. The following "1" and "3" denote the kernel size. "+" with a circular block denotes the concatenation. Each layer in DDFN (except the last one) adopts ReLU [25] as the activation function, which is omitted here for simplifying the illustration

detailed implementation of the boosting unit. Thus, we have a wide choice of diverse network structures. We start our investigation from a simple structure which is the simplified DnCNN [12] without batch normalization and residual connection, i.e., the plain network (PN), as shown in Table 1 and Fig. 2(a). We find in experiments that, given the same number of parameters, deepening a network properly contributes to the efficiency (as detailed in Sec. 5.2). However, when we introduce the PN into our DBF to derive a 2-stage boosting framework, this benefit tends to vanish as the network depth continues to increase, probably due to the vanishing of gradient during the back propagation.

4.1 Dense Connection

To overcome the propagation problem of gradient during training, we introduce the dense connection to derive the dense network (DN), as shown in Table 1 and Fig. 2(b), which is inspired by the successful model for image recognition [18]. The dense connection enables the l^{th} layer to receive the features of all preceding layers (i.e., $f_0, ..., f_{l-1}$) as input

$$f_l = g_l([f_0, f_1, ..., f_{l-1}]), \tag{16}$$

where $g_l(\cdot)$ denotes the l^{th} layer in $\mathcal{G}_\theta$ and $[f_0, f_1, ..., f_{l-1}]$ stands for the concatenation of the features output from preceding layers. We demonstrate in experiments that the dense connection can address the propagation issue of gradient during training (as detailed in Sec. 5.2).

4.2 Dilated Convolution

Widening the receptive field of the CNN is a well-known strategy for enhancing the performance in both image classification [26] and restoration [27] tasks. The convolution with a larger kernel size can widen the receptive field, however, it increases the number of parameters at the same time. Another strategy is stacking multiple convolutional layers with a 3×3 kernel size to obtain a large receptive field equivalently. However, it causes difficulty of convergence due to the increasing of the network depth.

Recently, a notable alternative called dilated convolution has been investigated in semantic segmentation [19] and image classification [28]. The dilated convolution can widen the receptive field without additive parameters and it also prevents the increasing of depth. Inspired by that, we introduce the dilated convolution to derive the dilated dense network (DDN) based on the DN, as shown in Table 1 and Fig. 2(c). By widening the receptive field efficiently, a better denoising performance can be achieved (as detailed in Sec. 5.2).

4.3 Path-widening Fusion

We further propose a path-widening fusion scheme to make the boosting unit more efficient. As shown in Table 1 and Fig. 2(d), we expand the number of forward paths to derive the DDFN from the DDN. Specifically, in a certain block, the order between the dilated convolutions (Dconv for short) and the normal convolutions (Conv for short) is exchanged in different branches. It is very likely that the Conv-ReLU-Dconv and Dconv-ReLU-Conv branches can learn different feature representations. The proposed path-widening fusion exploits the potential of these two orders at the same time, and thus promotes the possibility to learn better representations. Experimental results demonstrate that the denoising performance can be further improved in this way (as detailed in Sec. 5.2). Note that, we restrict the parameter number of DDFN not greater than DDN (i.e., about 4×10^4) to eliminate the influence of additional parameters due to path-widening fusion, and thus the efficiency of DDFN is also justified.

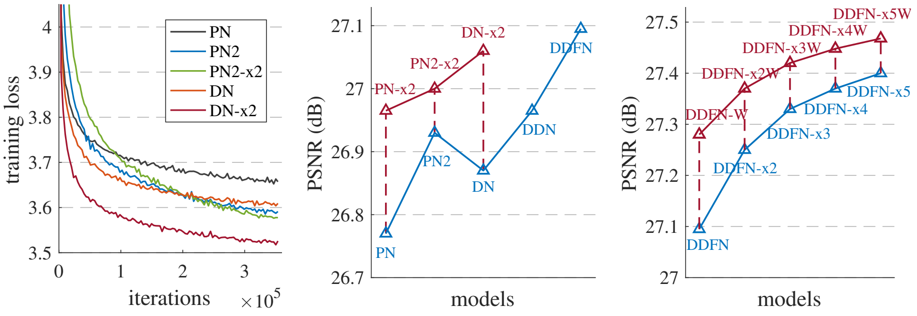
5 Experimental Results

5.1 Datasets and Settings

We adopt 400 images at a 180×180 resolution for training our models following TNRD [11] and DnCNN [12]. The images are partitioned into sub-image patches with a size of 50×50, and the mini-batch number is set to 64 for the stochastic gradient decent. Two widely-used datasets, “Set12” and “BSD68” [29] are employed as the benchmarks for image denoising. Moreover, to compare with the SOS algorithm [17], the “Set5” dataset is adopted following [17].

Besides grey-level image denoising, we also apply our method to two additional tasks, i.e., color image denoising and JPEG image deblocking, following the setting of DnCNN [12]. The color version of “BSD68” is adopted for the color image denoising task. And the “Classic5” and “LIVE1” datasets are adopted for evaluating the deblocking task as in [30].

We use TensorFlow and the “Adam” [31] solver for optimization with the momentum factor set as 0.9 and the coefficient of weight decay (L_2-Norm) as



(a) Curves of losses (b) Evolution of DDFN (c) Evolution of DBF

Fig. 3. Illustrations of the ablation experiments. (a) The curves show the advantage of dense connection over plain structure in terms of convergence. (b) The evolution from plainly connected structure to DDFN. (c) Performance comparisons between DBF and its variants. The symbol "W" means wide and all of these models are tested on the "Set12" dataset at $\sigma = 50$

0.0001. The learning rate is decayed exponentially from 0.001 to 0.0001. We stop training when no notable decay of training loss is observed after 3.6×10^5 iterations. The algorithm proposed in [32] is adopted for initializing the weights except the last layer. Specifically, the last layer for reconstruction is initialized by the random weights drawn from Gaussian distributions with $\sigma = 0.001$. And we set zeros for initializing the biases in each convolutional layer.

5.2 Ablation Experiments of DDFN

The proposed DDFN integrates three concepts: dense connection, dilated convolution, and path-widening fusion, deriving a fundamentally different structure compared with existing models. In this section, we design extensive ablation experiments to evaluate them respectively.

Investigation of depth (PN). We described the structure of PN in Sec. 4. To investigate the effect of depth to the boosting unit, we construct a variant of PN (named PN2) with a deeper yet thinner structure, which has the same number of parameters compared with PN. Specifically, it contains more layers (i.e., 18) in the feature integration part and less filter numbers (i.e., 16) in each layer than PN. Meanwhile, we keep the other hyper-parameters and the training procedure of PN2 the same as PN. As shown in Fig. 3(b), the deeper and thinner PN2 outperforms PN. This observation suggests that deepening the framework gives a better performance. However, when we introduce PN2 into DBF to derive PN2-x2, the advantage of plainly connected deeper structure tends to vanish.

Dense connection (DN). We then introduce the dense connection to address the propagation issue of gradient during training. As shown in Fig. 3(a), DN converges faster than PN2. While maintaining a quicker convergence, the derived DN-x2 shows a clear advantage over PN2-x2 for a 2-stage DBF, as shown

Table 2. Comparisons with the SOS boosting algorithm [17] combined with four classic models. DDFN-x5W is employed as a representative of our DBF. We evaluate the mean PSNR on the "Set5" dataset following [17]

Models	Original		Boosting	
	$\sigma = 25$	$\sigma = 50$	$\sigma = 25$	$\sigma = 50$
NLM [2]	29.93	26.43	30.34	26.73
K-SVD [5]	31.02	27.20	31.28	27.97
EPLL [1]	30.97	27.74	31.23	28.04
BM3D [7]	31.58	28.51	31.61	28.58
DDFN (Ours)	**32.04**	**28.93**	**32.38**	**29.50**

in Fig. 3(b). Note that, the parameters of DN are 15% less than PN2, yet DN-x2 still outperforms PN2-x2.

Dilated convolution (DDN). Based on DN, we adopt the dilated convolution to widen the receptive field. Specifically, we introduce it into two places of the network: the feature extraction part and each dense block, as shown in Fig. 2(c). The ratio of dilation is fixed to 2 for each dilated convolution layer. Experimental results demonstrate that further improvements of the boosting unit can be achieved, i.e., DDN as shown in Fig. 3(b).

Path-widening fusion (DDFN). As described in Sec. 4, we further propose the path-widening fusion which aggregates the concatenated features of preceding layers using a 1×1 convolution in the dense block, as shown in Fig. 2(d). This fusion can further promote the denoising performance, i.e., DDFN as shown in Fig. 3(b).

5.3 Investigation of Framework

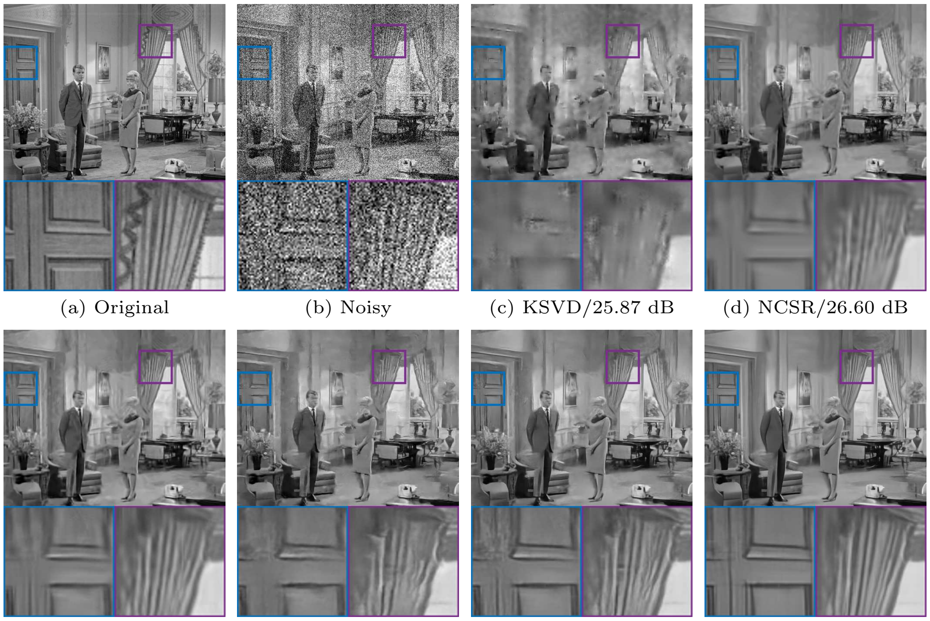
Ablation of subtraction and training scheme. As described in Sec. 3.2, the proposed DBF no longer needs an subtraction of $\hat{x}^n$ as in SOS to guarantee the iterability. We design an ablation experiment based on a 3-stage DBF. Experimental results demonstrate a better performance (+0.12 dB) without the subtraction. As for the training scheme, we consider both joint and greedy training as described in Sec. 3.2. Evaluated on a 3-stage DBF, we find joint training and greedy training give competitive performance.

Boosting - the deeper, the better. We investigate the performance by increasing the stage number of DBF. Experimental results demonstrate the capacity of our DBF in term of the extension in depth, as can be observed from Fig. 3(c). Specifically, a 5-stage DBF brings 0.30 dB gain compared with a single stage one.

DDFN - the wider, the better. Besides the exploration of depth, we also investigate the contribution of width by doubling the number of filters in each layer of DDFN (deriving models with the symbol "W"). Experimental results in Fig. 3(c) demonstrate that widening can further enhance the performance.

Table 3. Comparisons of mean PSNR (dB) between our DBF and seven representative learning-based models. We evaluate the results on two widely used benchmarks (i.e., "Set12" and "BSD68"). Best results reported in the corresponding papers are presented

Dataset	σ	MLP [10]	CSF [8]	GCRF [33]	TNRD [11]	NLNet [34]	DeepAM [35]	DnCNN [12]	DDFN/-x5W	
Set12	15	-	32.32	-	32.50	-	-	32.86	32.73	**32.98**
	25	30.03	29.84	-	30.06	-	30.40	30.44	30.33	**30.60**
	50	26.78	-	-	26.81	-	-	27.18	27.10	**27.46**
BSD68	15	-	31.24	31.43	31.42	31.52	31.68	31.73	31.66	**31.83**
	25	28.96	28.74	28.89	28.92	29.03	29.21	29.23	29.16	**29.35**
	50	26.03	-	-	25.97	26.07	26.24	26.23	26.19	**26.42**



(a) Original (b) Noisy (c) KSVD/25.87 dB (d) NCSR/26.60 dB

(e) BM3D/26.72 dB (f) TNRD/26.81 dB (g) DnCNN/27.18 dB (h) Ours/27.46 dB

Fig. 4. Visual comparisons of the image "Couple" from the "Set12" dataset at $\sigma = 50$

5.4 Comparison with State-of-the-Art Methods

Comparison with the SOS algorithm. We adopt four classical models [1], [2], [5], [7] and their corresponding SOS [17] variants for comparison. As shown in Table 2, our boosting unit DDFN has a clear advantage over these classic models, e.g., +0.46 dB when $\sigma = 25$ and +0.42 dB when $\sigma = 50$ than BM3D [7]) on the "Set5" dataset. With the proposed DBF, our DDFN-x5W achieves notable improvements over BM3D-SOS [17], i.e., +0.77 dB (σ=25) and +0.92 dB (σ=50).

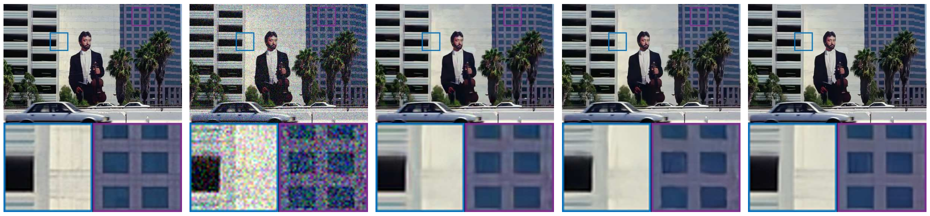
Comparison with other learning-based models. We adopt seven representative models for comparison: MLP [10], CSF [8], GCRF [33], TNRD [11], NLNet [34], DeepAM [35] and DnCNN [12]. The restoration results of our DDFN and DDFN-x5W are listed in Table 3 to compare with them. Specifi-

Table 4. Comparisons for the blind Gaussian denoising (grey-level). We evaluate the mean PSNR (dB) on the "Set12" dataset

Models / σ	5	10	15	20	25	30	35	40	45	50
BM3D [7]	38.03	34.38	32.37	31.01	29.97	29.13	28.40	27.65	27.19	26.72
DnCNN-B [12]	37.70	34.53	32.68	31.38	30.36	29.53	28.83	28.22	27.69	27.21
DDFN-B	37.31	34.30	32.45	31.15	30.13	29.30	28.60	27.98	27.44	26.96
DDFN-x5W-B	**38.14**	**34.76**	**32.86**	**31.55**	**30.54**	**29.71**	**29.02**	**28.41**	**27.87**	**27.44**

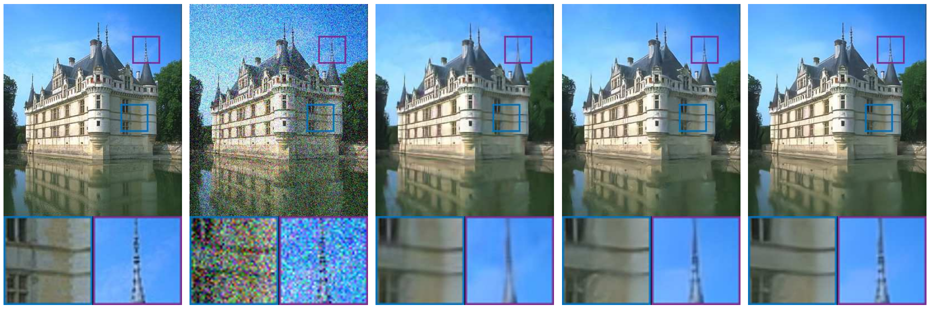
Table 5. Comparisons for the blind color image denoising. We evaluate the mean PSNR (dB) on the color version of "BSD68" dataset. A color variant of DDFN-x3W is adopted as the representative to compared with CBM3D [36] and DnCNN-B [12]

Models / σ	5	10	15	20	25	30	35	40	45	50
CBM3D [36]	40.24	35.91	33.52	31.91	30.71	29.73	28.89	28.09	27.84	27.38
DnCNN-B [12]	40.10	36.12	33.89	32.37	31.23	30.32	29.58	28.95	28.40	27.92
DDFN-x3W-B	**40.47**	**36.42**	**34.17**	**32.65**	**31.52**	**30.62**	**29.88**	**29.26**	**28.72**	**28.26**



(a) Original (b) Noisy (c) CBM3D/28.86 dB (d) CDnCNN/29.46 dB (e) Ours/29.92 dB

Fig. 5. Visual comparisons of an image from the "BSD68" dataset at $\sigma = 35$



(a) Original (b) Noisy (c) CBM3D/28.80 dB (d) CDnCNN/29.29 dB (e) Ours/29.66 dB

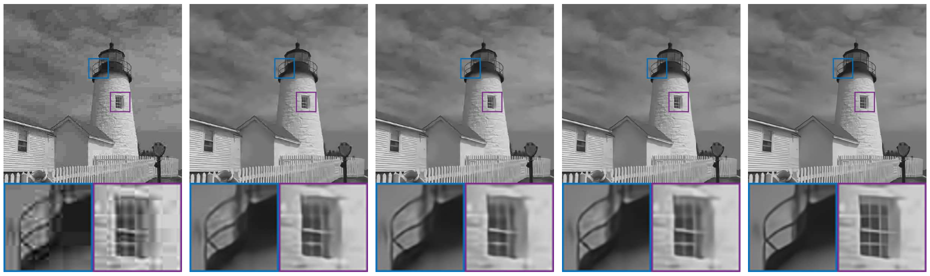
Fig. 6. Visual comparisons of an image from the "BSD68" dataset at $\sigma = 45$

cally, DDFN-x5W achieves a superior performance than TNRD [11] (+0.65 dB) and DnCNN [12] (+0.28 dB) on the "Set12" dataset when $\sigma = 50$.

Comparison for blind Gaussian denoising. Following the settings of training proposed in [12], we re-train our models to derive the DDFN-B and DDFN-x5W-B for blind Gaussian denoising. We adopt the BM3D [7] and the

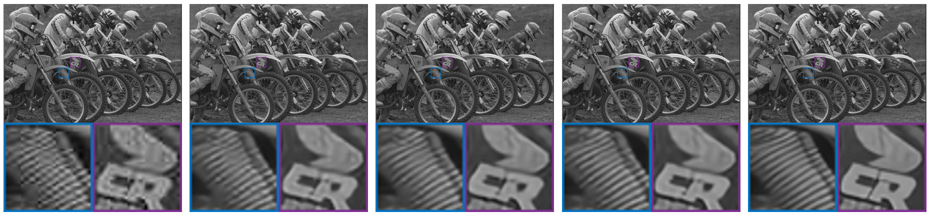
Table 6. Compression of the JPEG image deblocking. We evaluate the mean PSNR (dB) and SSIM on the "Classic5" and "LIVE1" datasets in terms of four quality factors (QF). All methods are implemented using officially available codes

Dataset	QF	AR-CNN [30]	TNRD [11]	DnCNN [12]	DDFN-x3W
		PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM
Classic5	10	29.03 / 0.7929	29.28 / 0.7992	29.40 / 0.8026	**29.55 / 0.8086**
	20	31.15 / 0.8517	31.47 / 0.8576	31.63 / 0.8610	**31.70 / 0.8636**
	30	32.51 / 0.8806	32.78 / 0.8837	32.91 / 0.8861	**33.03 / 0.8881**
	40	33.34 / 0.8953	-	33.77 / 0.9003	**33.90 / 0.9023**
LIVE1	10	28.96 / 0.8076	29.15 / 0.8111	29.19 / 0.8123	**29.39 / 0.8186**
	20	31.29 / 0.8733	31.46 / 0.8769	31.59 / 0.8802	**31.76 / 0.8839**
	30	32.67 / 0.9043	32.84 / 0.9059	32.98 / 0.9090	**33.19 / 0.9117**
	40	33.63 / 0.9198	-	33.96 / 0.9247	**34.20 / 0.9273**



(a) JPEG (b) AR-CNN/29.63 dB (c) TNRD/29.75 dB (d) DnCNN/29.82 dB (e) Ours/30.12 dB

Fig. 7. Visual comparisons of an image from the "LIVE1" dataset at QF=10



(a) JPEG (b) AR-CNN/29.84 dB (c) TNRD/30.11 dB (d) DnCNN/30.28 dB (e) Ours/30.56 dB

Fig. 8. Visual comparisons of an image from the "LIVE1" dataset at QF=20

variant of DnCNN (i.e., DnCNN-B [12]) for comparison. Experimental results listed in Table 4 demonstrate the superiority of our model within a wide range of noise levels. Specifically, when the noise level is small (i.e., $\sigma = 5$), our proposed DDFN-x5W-B has a clear advantage (+0.44 dB) over DnCNN-B. We also evaluate the performance on the task of blind color image denoising. Experimental results listed in Table 5 demonstrate the advantage of our proposed DBF.

Comparison for JPEG image deblocking. We also evaluate our model on the task of image deblocking. Three representative models: AR-CNN [30], TNRD [11], and DnCNN [12] are adopted for comparison. Experimental results listed in Table 6 demonstrate the superiority of our model over existing ones.

Table 7. Comparison of runtime (s) for Gaussian image denoising on the "Set12" dataset with respect to different resolutions. All methods are implemented using officially available codes

Resolution	K-SVD	EPLL	BM3D	MLP	CSF	TNRD	DnCNN	Ours
256 × 256	71.886	19.352	0.369	1.211	1.006	1.125	0.061	**0.051**
512 × 512	56.689	90.184	1.561	3.486	2.491	2.238	0.212	**0.156**

Running time. Although the cascaded structure of our proposed DBF involves more computation than a single stage one (which is the inevitable cost of boosting), it is still quite efficient. Detailed results are listed in Table 7.

Visual comparison. To evaluate the perceptual quality of restoration, we show a few examples including grey-level denoising (Fig. 4), blind color image denoising (Figs. 5-6), and image deblocking (Figs. 7-8). As can be seen, our model performs better than the competitors in both the smooth and edge regions.

6 Conclusions

In this paper, we propose the DBF which first integrates the boosting algorithm with deep learning for image denoising. To fully exploit the potential of this framework, we elaborate the lightweight yet efficient DDFN as the boosting unit. By introducing the dense connection, we address the vanishing of gradients during training. Based on the densely connected structure, we further propose the path-widening fusion cooperated with the dilated convolution to optimize the DDFN for efficiency. Compared with the existing models, our DDFN-based DBF achieves the state-of-the-art performance in both non-blind and blind image denoising on widely used benchmarks.

Besides the scenario of image denoising, the proposed DBF can be readily generalized to other image restoration tasks, e.g., image deblocking, as demonstrated in this paper. Also, the idea of path-widening fusion is demonstrated to be useful in the task of spectral reconstruction from RGB images [37]. We believe the proposed method could inspire even more low-level vision applications.

Acknowledgements. We acknowledge funding from National Key R&D Program of China under Grant 2017YFA0700800, and Natural Science Foundation of China under Grants 61671419 and 61425026.

References

1. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: IEEE International Conference on Computer Vision (ICCV). (2011) 479–486
2. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 2. (June 2005) 60–65 vol. 2

3. Buades, A., Coll, B., Morel, J.: Nonlocal image and movie denoising. International Journal of Computer Vision (IJCV) **76**(2) (2008) 123–139
4. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: IEEE International Conference on Computer Vision (ICCV). (2009) 2272–2279
5. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image Processing **15**(12) (Dec 2006) 3736–3745
6. Dong, W., Zhang, L., Shi, G., Li, X.: Nonlocally centralized sparse representation for image restoration. IEEE Transactions on Image Processing **22**(4) (April 2013) 1620–1630
7. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on Image Processing **16**(8) (Aug 2007) 2080–2095
8. Schmidt, U., Roth, S.: Shrinkage fields for effective image restoration. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2014) 2774–2781
9. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2014) 2862–2869
10. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: Can plain neural networks compete with bm3d? In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2012) 2392–2399
11. Chen, Y., Pock, T.: Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. IEEE Transactions on Pattern Analysis & Machine Intelligence **39**(6) (2017) 1256
12. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing **26**(7) (July 2017) 3142–3155
13. Remez, T., Litany, O., Giryes, R., Bronstein, A.M.: Deep class-aware image denoising. In: International Conference on Sampling Theory and Applications (SampTA). (July 2017) 138–142
14. Charest, M.R., Elad, M., Milanfar, P.: A general iterative regularization framework for image denoising. In: Conference on Information Sciences and Systems. (March 2006) 452–457
15. Milanfar, P.: A tour of modern image filtering: New insights and methods, both practical and theoretical. IEEE Signal Processing Magazine **30**(1) (2013) 106–128
16. Talebi, H., Zhu, X., Milanfar, P.: How to saif-ly boost denoising performance. IEEE Transactions on Image Processing **22**(4) (April 2013) 1470–1485
17. Romano, Y., Elad, M.: Boosting of image denoising algorithms. Siam Journal on Imaging Sciences **8**(2) (2015) 1187–1219
18. Huang, G., Liu, Z., v. d. Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017) 2261–2269
19. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: International Conference on Learning Representations (ICLR). (2016)
20. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning (ICML). (2015) 448–456

21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2016) 770–778
22. Buades, A., B, C., Morel, J.M.: A review of image denoising algorithms, with a new one. Siam Journal on Multiscale Modeling & Simulation **4**(2) (2005) 490–530
23. Han, S., Meng, Z., Khan, A.S., Tong, Y.: Incremental boosting convolutional neural network for facial action unit recognition. In Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R., eds.: Advances in Neural Information Processing Systems (NIPS). Curran Associates, Inc. (2016) 109–117
24. Moghimi, M., Saberian, M., Yang, J., Li, L.J., Vasconcelos, N., Belongie, S.: Boosted convolutional neural networks. In: British Machine Vision Conference (BMVC). (2016) 24.1–24.13
25. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: International Conference on Machine Learning (ICML). (2010) 807–814
26. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2016) 2818–2826
27. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2016) 1646–1654
28. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017) 636–644
29. Roth, S., Black, M.J.: Fields of experts: a framework for learning image priors. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 2. (June 2005) 860–867 vol. 2
30. Dong, C., Deng, Y., Loy, C.C., Tang, X.: Compression artifacts reduction by a deep convolutional network. In: IEEE International Conference on Computer Vision (ICCV). (Dec 2015) 576–584
31. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. Computer Science (2014)
32. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: IEEE International Conference on Computer Vision (ICCV). (Dec 2015) 1026–1034
33. Vemulapalli, R., Tuzel, O., Liu, M.Y.: Deep gaussian conditional random field network: A model-based deep network for discriminative denoising. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2016) 4801–4809
34. Lefkimmiatis, S.: Non-local color image denoising with convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017) 5882–5891
35. Kim, Y., Jung, H., Min, D., Sohn, K.: Deeply aggregated alternating minimization for image restoration. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017) 284–292
36. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space. In: IEEE International Conference on Image Processing (ICIP). Volume 1. (Sept 2007) I – 313–I – 316
37. Shi, Z., Chen, C., Xiong, Z., Liu, D., Wu, F.: Hscnn+: Advanced cnn-based hyperspectral recovery from rgb images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. (June 2018)