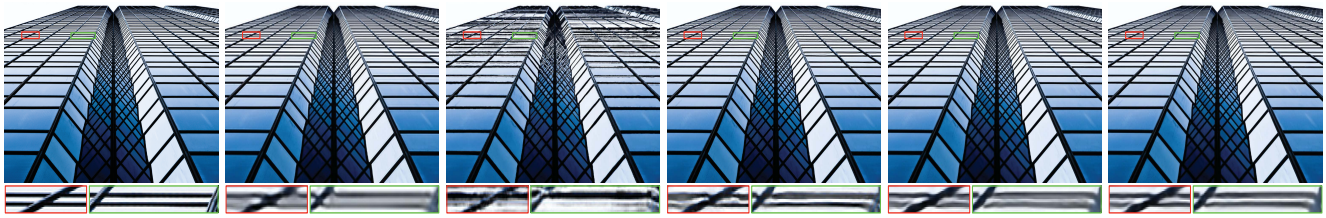
Learning Dual Convolutional Neural Networks for Low-Level Vision

Jinshan Pan[1] Sifei Liu[2] Deqing Sun[2] Jiawei Zhang[3] Yang Liu[4] Jimmy Ren[5]
Zechao Li[1] Jinhui Tang[1] Huchuan Lu[4] Yu-Wing Tai[6] Ming-Hsuan Yang[7]
[1]Nanjing University of Science and Technology [2]NVIDIA [3]City University of Hong Kong
[4]Dalian University of Technology [5]SenseTime Research [6]Tencent Youtu Lab [7]UC Merced



(a) GT (b) SRCNN [5] (c) VDSR with nearest (d) VDSR with bilinear (e) VDSR with bicubic (f) Ours

Figure 1. Visual comparisons of super-resolution results by the VDSR method [17] ($\times 4$) with structures recovered by different methods, *i.e.*, nearest neighbor, bilinear, and bicubic upsampling. Residual learning algorithms usually take upsampled image as the base structures and learn the details, the difference between the upsampled and ground truth images. However, residual learning cannot correct low-frequency errors in the structures, *e.g.*, the structure obtained by nearest neighbor interpolation in (c). In contrast, our algorithm is motivated by the decomposition of a signal into structures and details, which involves both structure and detail learning and thus leads to better results.

Abstract

In this paper, we propose a general dual convolutional neural network (DualCNN) for low-level vision problems, e.g., super-resolution, edge-preserving filtering, deraining and dehazing. These problems usually involve the estimation of two components of the target signals: structures and details. Motivated by this, our proposed DualCNN consists of two parallel branches, which respectively recovers the structures and details in an end-to-end manner. The recovered structures and details can generate the target signals according to the formation model for each particular application. The DualCNN is a flexible framework for low-level vision tasks and can be easily incorporated into existing CNNs. Experimental results show that the DualCNN can be effectively applied to numerous low-level vision tasks with favorable performance against the state-of-the-art methods.

1. Introduction

Motivated by the success of deep learning in high-level vision tasks [19, 11, 13, 31, 32], numerous deep models have been developed for low-level vision tasks, *e.g.*, image super-resolution [6, 7, 5, 17, 18, 22], inpainting [27, 24], noise removal [4, 15, 35], image filtering [38, 24], image deraining [8, 40], and dehazing [28, 2]. Although achieving impressive performance, the network architectures of these models strongly resemble those developed for high-level classification tasks.

Existing methods are based on either plain neural networks or residual learning networks. As demonstrated in [1, 27], plain neural networks cannot outperform state-of-the-art traditional approaches on a number of low-level vision problems, e.g., super-resolution [34]. Low-level vision tasks usually involve the estimation of two components, low-frequency structures and high-frequency details. It is challenging for a single network to learn both components simultaneously. As a result, going deeper with plain neural networks does not always lead to better performance [6].

Residual learning has been shown to be an effective approach to achieve performance gain with a deeper network. The residual learning algorithms (e.g., [17]) assume that the main structure is given and mainly focus on estimating the residual (details) using a deep network. These methods work well when the recovered structures are perfect or near perfect. However, when the main structure is not well recovered, these methods do not perform well, because the final result is a combination of the structures and details. Figure 1 shows the image super-resolution results by the VDSR method [17] with structures recovered by different methods. The residual network cannot correct low-frequency errors in the structures (Figure 1(b)).

To address this issue, we propose a dual convolutional neural network (DualCNN) that can jointly estimate the

structures and details. A DualCNN consists of two branches, one shallow sub-network to estimate the structures and one deep sub-network to estimate the details. The modular design of a DualCNN makes it a flexible framework for a variety of low-level vision problems. When trained end-to-end, DualCNNs perform favorably against state-of-the-art methods that have been specially designed for each individual task.

2. Related Work

Numerous deep learning methods have been developed for low-level vision tasks. A comprehensive review is beyond the scope of this work and we discuss the most related ones in this section.

Super-resolution. The SRCNN [5] method uses a three-layer plain convolutional neural network (CNN) for super-resolution. As the SRCNN method is less effective in recovering image details, Kim et al. [17] propose the residual learning [17] algorithm based on a deeper network. The VDSR algorithm uses the bicubic interpolation of the low-resolution input as the structure of the high-resolution image and estimates the residual details using a 20-layer CNN. However, if the image structure is not well recovered, the generated result is likely to contain substantial artifacts, as shown in Figure 1.

Noise/artifacts removal. Numerous algorithms based on CNNs have been developed to remove noise/artifacts [4, 15, 35] and unwanted components, e.g., rainy/dirty pixels [8, 40]. These methods are based on plain models, residual learning models or recurrent models. In addition, these methods estimate either the output using one plain network, or details using a residual network. However, plain networks cannot recover fine details [13, 17] and residual networks cannot correct structural errors.

Edge-preserving filtering. For edge-preserving filtering, Xu et al. [38] develop a CNN model to approximate a number of filters. Liu et al. [24] use a hybrid network to approximate a number of edge-preserving filters. These methods aim to preserve the main structures and remove details using a single network, but this imposes a difficult learning task. In this work, we show that it is critical to accurately estimate both the structures and the details for low-level vision tasks.

Image dehazing. In image dehazing, existing CNN-based methods [2, 28] mainly focus on estimating the transmission map from an input. Given an estimated transmission map, the atmospheric light can be computed using the air light model. As such, errors in the transmission maps are propagated into the light estimation process. For more accurate results, it is necessary to jointly estimate the transmission map and atmospheric light in one model, which DualCNNs are designed for.

A common theme is that we need to design a new network for every low-level vision task. In this paper, we show that low-level vision problems usually involve the estimation of two components: structures and details. Thus we develop a single framework, called DualCNN, that can be flexibly applied to a variety of low-level vision problems, including the four tasks discussed above.

3. Proposed Algorithm

As shown in Figure 2, the proposed dual model consists of two branches, Net-S, and output of Net-D, which respectively estimate the structure and detail components of the target signals from the input. Take image super-resolution as an example. Given a low-resolution image, we first use the bicubic upsampled image as the input. Then, our dual network learns the details and structures according to the formulation model of the image decomposition.

Dual composition loss function. Let X, S, and D denote the ground truth label, output of Net-S, and output of Net-D, respectively. The dual composition loss function enforces that the recovered structure S and detail D can generate the ground truth label X using the given formation model:

$$\mathcal{L}_x(S, D) = \|\phi(S) + \varphi(D) - X\|_2^2, \tag{1}$$

where the forms of the functions $\phi(\cdot)$ and $\varphi(\cdot)$ are known and depend on the domain knowledge of each task. For example, the functions $\phi(\cdot)$ and $\varphi(\cdot)$ are identity functions for image decomposition problems (e.g., filtering) and restoration problems (e.g., super-resolution, denoising, and deraining). We will show that $\phi(\cdot)$ and $\varphi(\cdot)$ can take more general forms to deal with specific problems.

3.1. Regularization of the DualCNN Model

The proposed DualCNN model has two branches, which may cause instability if only the composition loss (1) is used. For example, if Net-S and Net-D have the same structure, symmetrical solutions exist. To obtain a stable solution, we use individual loss functions to regularize the two branches respectively. The loss functions for the Net-S and Net-D are defined as

$$\mathcal{L}_s(S) = \|S - S_{gt}\|_2^2, \tag{2}$$

$$\mathcal{L}_d(D) = \|D - D_{gt}\|_2^2, \tag{3}$$

where S_{gt} and D_{gt} are ground truths corresponding to the outputs of Net-S and Net-D. Consequently the overall loss function to train DualCNN is

$$\mathcal{L} = \alpha\mathcal{L}_x + \lambda\mathcal{L}_s + \gamma\mathcal{L}_d, \tag{4}$$

where α, λ and γ are non-negative trade-off weights. Our framework can also use other loss functions, e.g., perceptual loss for style transfer.

We use the SGD method to minimize the loss function (4) and train a DualCNN. In the training stage, the gradients for Net-S and Net-D can be obtained by

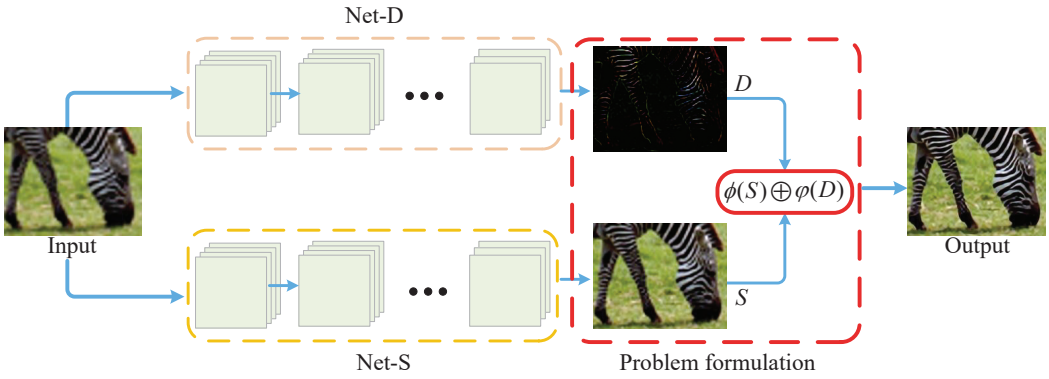


Figure 2. Proposed DualCNN model. It contains two branches, Net-D and Net-S, and a problem formulation module. A DualCNN first estimates the structures and the details and then reconstructs the final results according to the formulation module. The whole network is end-to-end trainable.

$$\frac{\partial \mathcal{L}}{\partial S} = 2\alpha\phi'(S)E + 2\lambda(S - S_{gt}), \tag{5a}$$

$$\frac{\partial \mathcal{L}}{\partial D} = 2\alpha\varphi'(D)E + 2\gamma(D - D_{gt}), \tag{5b}$$

where $E = \phi(S) + \varphi(D) - X$, $\phi'(S)$ and $\varphi'(D)$ are the derivatives with respect to S and D.

In the test stage, we compute the high-quality output X_{est} using the outputs of Net-S and Net-D according to the formation model,

$$X_{est} = \phi(S) + \varphi(D). \tag{6}$$

3.2. Generalization

Aside from image decomposition and restoration problems, the proposed model can handle other low-level vision problems by modifying the composition loss function (1). Here we use image dehazing as an example.

Image dehazing. The image dehazing model can be described using the air light model,

$$I = JD + S(1 - D), \tag{7}$$

where I is the hazy image, J is the haze-free image, S is the atmospheric light, and D is the medium transmission map, which describes the portion of the light that reaches the camera from scene surfaces. With the formulation model (7), we can set $\phi(S) = S(1 - D)$ and $\varphi(D) = JD$ in (1) within the DualCNN framework. As a result, the composition loss function (1) for image dehazing becomes

$$\mathcal{L}_x(S, D) = \|JD + S(1 - D) - I\|_2^2. \tag{8}$$

The other two loss functions (2) and (3) remain the same. In the training phase, we use the same method [28] to generate the atmospheric light S, the transmission map D and construct hazy/haze-free image pairs. The implementation details of the training stage are presented in Section 4.4.

In the test phase, the clear image J_{est} can be reconstructed by the outputs of Net-D and Net-S, i.e.,

$$J_{est} = \frac{I - S}{\max\{D, d_0\}} + S, \tag{9}$$

where d_0 is used to prevent division by zero and a typical value is 0.1 .

4. Experimental Results

We evaluate DualCNNs on several low-level vision tasks including super-resolution, edge-preserving smoothing, deraining and dehazing. The main results are presented in this section and more results can be found in the supplementary material. The trained models are publicly available on the authors' websites.

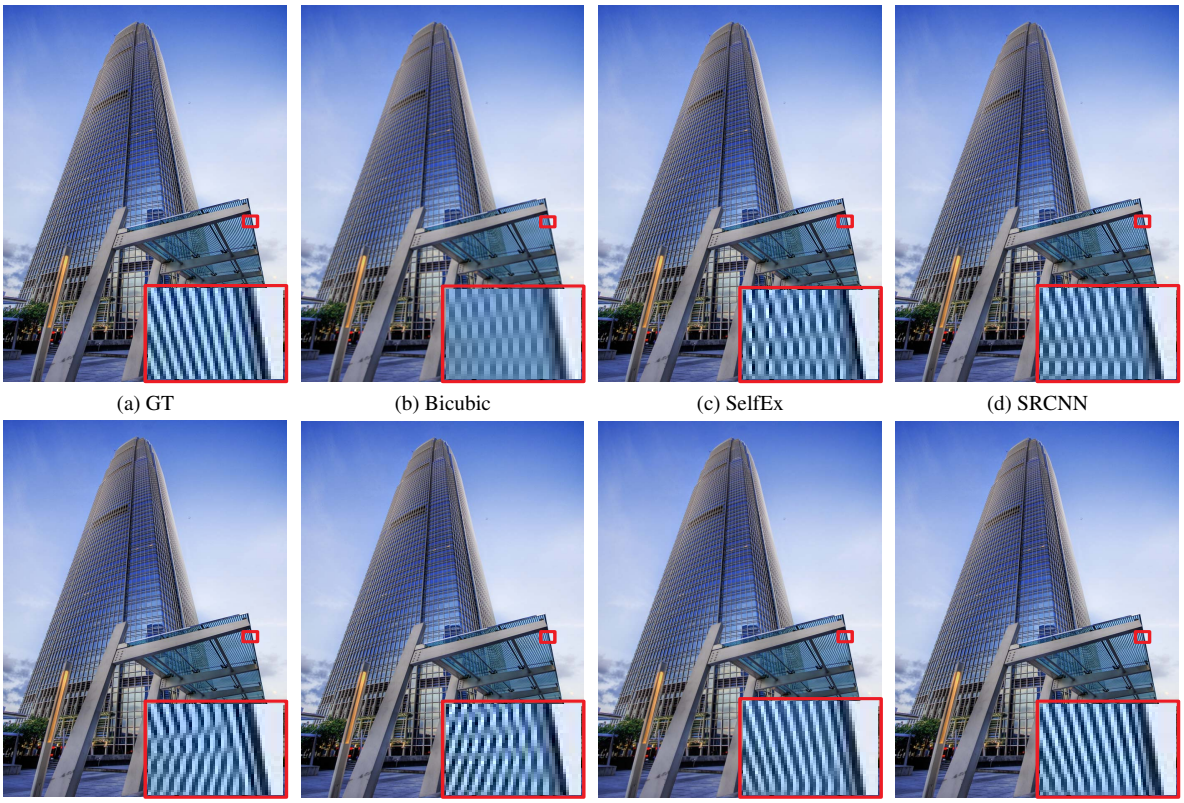
Network parameters. Motivated by the success of SRCNN and VDSR for super-resolution, we use 3 convolution layers followed by the ReLU function for the network Net-S. The filter sizes of each layer are 9×9, 1×1, and 5×5, respectively. The depths of each layer are 64, 32, and 1, respectively. For the network Net-D, we use 20 convolution layers followed by the ReLU function. The filter size of each layer is 3×3 and the depth of each layer is 64. The batch size is set to be 64 and the learning rate is 10^{-4}. Although each branch of the proposed model is similar to SRCNN or VDSR, both our analysis and experimental results show that the proposed model is significantly different from these methods and achieves better results.

4.1. Image Super-resolution

Training data. For image super-resolution, we generate the training data by randomly sampling 250 thousands 41×41 patches from 291 natural images in [25]. We apply the Gaussian filter to each ground truth label X to obtain S_{gt}. The ground truth D_{gt} is the difference between the ground truth label X and the structure S_{gt}.

For this application, we set $\phi(S) = S$ and $\varphi(D) = D$. The weights α, λ and γ in the loss function (4) are set to be 1, 0.001 and 0.01, respectively. To increase the accuracy, we use the pre-trained models of SRCNN and VDSR as the initializations of Net-S and Net-D.

We present quantitative and qualitative comparisons against the state-of-the-art methods including A+ [34], SelfEx [14], SRCNN [5], ESPCN [29], SRGAN [20], and VDSR [17]. Table 1 shows quantitative evaluations on benchmark datasets. Overall, the proposed method performs fa-



(a) GT (b) Bicubic (c) SelfEx (d) SRCNN

(e) VDSR (f) ESPCN (g) SRGAN (h) Ours

Figure 3. Visual comparisons for super-resolution ($\times 2$). The state-of-the-art methods do not preserve the main structures of the images, while the proposed method generates a better result.

Table 1. Quantitative evaluations for the state-of-the-art super-resolution methods on the benchmark datasets (Set5, Set14, Urban100, and BSDS500) in terms of PSNR and SSIM.

Dataset	Scale	Bicubic PSNR/SSIM	A+ PSNR/SSIM	SelfEx PSNR/SSIM	SRCNN PSNR/SSIM	ESPCN PSNR/SSIM	VDSR PSNR/SSIM	SRGAN PSNR/SSIM	Ours PSNR/SSIM
Set5	$\times 2$	33.6924/0.9308	36.575/0.9546	36.5392/0.9537	36.4191/0.9531	36.7315/0.9547	37.6173/0.9596	37.0098/0.9548	**37.7005/0.9600**
	$\times 3$	30.4396/0.8694	32.6866/0.9097	32.6759/0.9099	32.4957/0.9049	32.6880/0.9077	33.7571/0.9229	33.5384/0.9170	**33.8003/0.9234**
	$\times 4$	28.4528/0.8116	30.3471/0.8623	30.3458/0.8636	30.1496/0.8551	30.2730/0.8540	31.4657/**0.8863**	31.3496/0.8797	**31.4778**/0.8860
Set14	$\times 2$	30.2660/0.8687	32.3497/0.9051	32.2657/0.9029	32.2934/0.9040	32.4020/0.9056	33.2037/0.9131	32.6889/0.9049	**33.2334/0.9131**
	$\times 3$	27.5556/0.7731	29.1602/0.8181	29.1743/0.8190	29.0676/0.8147	29.1161/0.8161	29.8720/**0.8319**	29.5351/0.8227	**29.8822**/0.8315
	$\times 4$	26.0089/0.7006	27.3238/0.7481	27.3956/0.7509	27.2283/0.7402	27.1681/0.7401	28.0667/**0.7671**	27.8353/0.7588	**28.0949**/0.7669
Urban100	$\times 2$	26.8621/0.8400	28.5485/0.8782	29.5317/0.8962	29.1009/0.8896	29.2381/0.8920	30.7897/0.9144	29.4390/0.8745	**30.8273/0.9145**
	$\times 3$	24.4375/0.7336	25.7907/0.7878	26.4188/0.8079	25.8549/0.7874	25.9170/0.7897	27.1297/**0.8278**	26.6243/0.8159	**27.1318**/0.8277
	$\times 4$	23.1158/0.6551	24.1890/0.7119	24.7648/0.7361	24.1275/0.7030	24.1534/0.7031	25.1575/**0.7515**	24.1516/0.7298	**25.1722**/0.7510
BSDS500	$\times 2$	29.6393/0.8622	30.8758/0.8929	31.3447/0.9016	31.3319/0.9013	31.4187/0.9030	32.2226/0.9136	31.3938/0.8889	**32.2458/0.9140**
	$\times 3$	27.1875/0.7626	28.1461/0.8024	28.2960/0.8073	28.2233/0.8033	28.2404/0.8048	28.8889/0.8229	28.6354/0.8159	**28.8979/0.8232**
	$\times 4$	25.8953/0.6931	26.6798/0.7324	26.7851/0.7368	26.6595/0.7278	26.6122/0.7278	27.2342/0.7525	26.9104/0.7423	**27.2454/0.7527**

Table 2. Average running time (seconds) of the evaluated methods on the test dataset [17].

Methods	A+	SelfEx	SRCNN	VDSR	Ours
Average running time	0.88	99.04	0.55	4.85	5.19

Table 3. PSNR results for learning various image filters on the test dataset [37].

	Xu et al. [38]	Liu et al. [24]	VDSR [17]	Net-S	Ours
L_0	**32.8**	30.9	31.5	28.0	31.4
WMF	31.4	34.0	38.5	29.2	**39.1**
RTV	32.1	37.1	41.6	32.0	**42.1**

vorably against the state-of-the-art methods. Note that the architecture of one branch in a DualCNN is either similar to SRCNN or VDSR. However, the results generated by a DualCNN have highest average PSNR values, suggest-

(a) Xu et al. (b) Liu et al. (c) Net-D (d) Net-S (e) Ours (f) RTV

Figure 4. Visual comparisons for learning the relative total variation (RTV) image filters. Existing deep learning based methods are not able to remove the details and structures that are supposed to be removed (the green boxes in (a) and (b)). (c) and (d) show the outputs of the two branches of the proposed model. (f) is the result by the original implementation of RTV. Better enlarge and view on a screen.

ing the effectiveness of the proposed dual model. Figure 3 shows some super-resolution results by the evaluated methods. The proposed algorithm can well preserve the main structures than state-of-the-art methods.

Running time. We benchmark the running time of all methods on a machine with an Intel Core i7-7700 CPU and an NVIDIA GTX 1080 GPU. Table 2 shows that the running time of the DualCNN model is comparable to VDSR, which achieves state-of-the-art results on the super-resolution benchmark dataset [17].

4.2. Edge-preserving Filtering

Similar to the methods in [38] and [24], we apply the DualCNN to learn edge preserving image filters including L_0 smoothing [36], relative total variation (RTV) [39], and weighted median filter (WMF) [41]. We generate the training data by randomly sampling 1 million patches (clear/filtered pairs) from 200 natural images in [25]. Each image patch is of 64×64 pixels, and other settings of generating training data are the same as those used in [38].

For this application, as our goal is to learn the filtered image which does not contain rich details, we set weights α, λ, and γ in the loss function (4) to be 1, 10^{-4} and 0, respectively. We further let S_{gt} be the ground truth label X.

We evaluate the proposed DualCNN model against methods [38, 24] using the dataset from [38]. Table 3 summarizes the PSNR results. Note that Xu et al. [38] use image gradients to train their model and the final results are reconstructed by solving a constrained optimization problem. Thus it performs better for approximating L_0 smoothing. However, our method does not need these additional steps and generates high quality filtered images with significant improvements over the state-of-the-art deep learning based methods, particularly on RTV and WMF.

We note that the architecture of Net-D is similar to that of VDSR. As such, we retrain the network of VDSR for these problems. The results in Table 3 suggest that only using residual learning does not always generate high-quality filtered images.

Figure 4 shows the filtering results of approximating RTV [39]. The state-of-the-art methods [38, 24] fail to smooth the structures (e.g., the eyes in the green boxes) that are supposed to be removed using the RTV filter (Figure 4(f)). In addition, the results with only one branch (i.e., Net-S) have lower PSNR values (Table 3) and some remaining tiny structures (Figure 4(d)). In contrast, joint learning structures and details preserves more accurate results and the filtered images are significantly closer to the ground truth.

4.3. Image Deraining

Deraining aims to recover clear contents from rainy images. This process can be regarded as recovering the clear details (rainy streaks) and structures (clear images) from inputs. We evaluate the proposed DualCNN on this task.

To train the proposed DualCNN for image deraining, we generate the training data by randomly sampling 1 million patches (rainly/clear pairs) from the rainy image dataset used in [40]. The size of each image patch used in training stage is 64×64 pixels. Following settings used in learning image filtering, we let S_{gt} be the ground truth label X (i.e., clear image patch). The weights α, λ and γ in the loss function (4) are set to be 1, 0.01, and 0, respectively. We use the test dataset [40] to evaluate the effectiveness of the proposed method.

Table 4 shows the average PSNR values of restored images on the test dataset [40]. Overall, the proposed method generates the results with the highest PSNR values.

Figure 5 shows deraining results from the evaluated methods. The proposed algorithm can accurately estimate both clear details and structures from the input image. The plain CNN-based methods [9], [40] and Net-S all generate results with obvious rainy streaks, demonstrating the advantage of simultaneously recovering structures and details using the DualCNN.

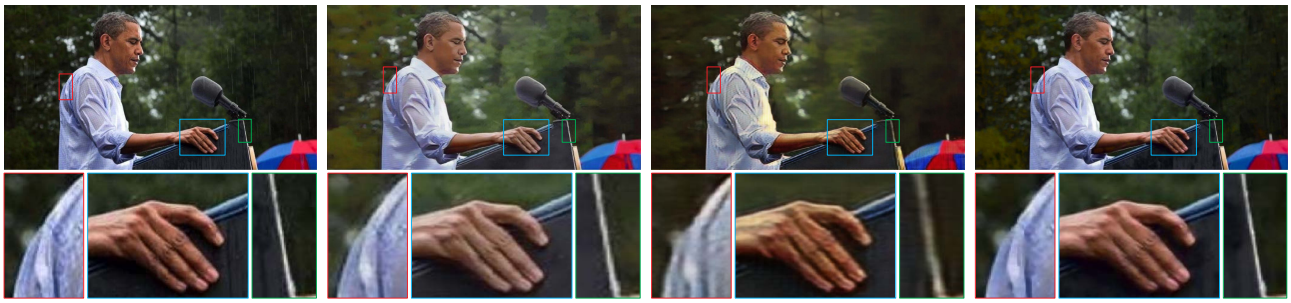
We further evaluate DualCNN using real examples. Figure 6 shows a real example. We note that the algorithm in [10] develops a deep details network for image deraining. The derained images are obtained by extracting details from input. However, this method depends on whether

(a) Input (b) SPM (c) ID-CGAN (d) Net-D (e) Net-S (f) Ours

Figure 5. Visual comparisons for image deraining. The proposed method is able to remove rainy streaks from the input image.

Table 4. Quantitative comparison using the synthetic rainy dataset [40].

Methods	SPM [16]	PRM [3]	CNN [9]	GMM [21]	ID-CGAN [40]	Net-S	Ours
Avg. PSNR	18.88	20.46	19.12	22.27	22.73	22.18	**24.11**



(a) Input (b) ID-CGAN [40] (c) Fu et al. [10] (d) Ours

Figure 6. Visual comparisons of deep learning-based methods for image deraining on real examples. The proposed method is able to remove rainy streaks from the input image and generates much better images with fine details.

Table 5. Quantitative comparison using the synthetic hazy image dataset [28].

Methods	He et al. [12]	Meng et al. [26]	Ren et al. [28]	Ours
Avg. PSNR	15.86	15.06	18.38	**18.85**

the image decomposition method is able to extract details or not. The results shown in Figure 6(c) demonstrate the algorithm in [10] fails to generate clearer images. In contrast, our method generates much clearer results compared to state-of-the-art algorithms.

4.4. Image Dehazing

As discussed in Section 3.2, the proposed method can be applied to the image dehazing. Similar to the method in [28], we synthesize the hazy image dataset using the NYU depth dataset [30] and generate the training data by randomly sampling 1 million patches including hazy/clear pairs (I/J), atmospheric light (S), transmission map (D). The size of each image patch used in training stage is 32×32 pixels. The weights α, λ and γ are set to be 0.1, 0.9, and 0.9, respectively.

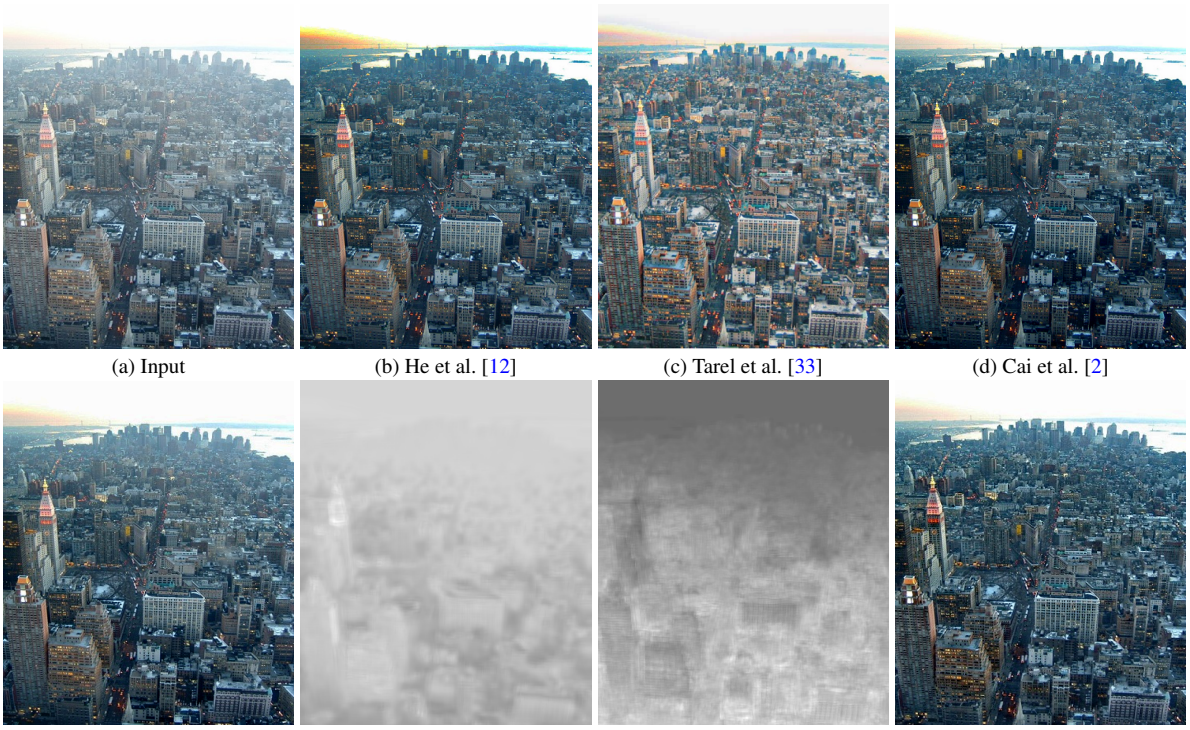
We quantitatively evaluate our method on the synthetic hazy images [28]. As summarized in Table 5, the proposed method performs favorably against the state-of-the-art methods for image dehazing. The dehazed images in Figure 7 show that the proposed method can recover the atmospheric light (Figure 7(e)) and transmission map (Figure 7(f)) well, thereby facilitating to recover the clear image (Figure 7(g)).

5. Analysis and Discussion

In this section, we further analyze the proposed method and compare it with the most related methods.

Effect of the architectures of DualCNN. Lin et al. [23] develop a bilinear model to extract complementary features for fine-grained visual recognition. By contrast, the proposed DualCNN is motivated by the decomposition of a signal into structures and details. More importantly, the formulation of the proposed model facilitates incorporating the domain knowledge of each individual application. Thus, the DualCNN model can be effectively applied to numerous low-level vision problems, e.g., super-resolution, image filtering, deraining, and dehazing.

Numerous deep learning methods have been developed based on a single branch for low-level vision problems, e.g., SRCNN [5] and VDSR [17]. One natural question is why



(a) Input (b) He et al. [12] (c) Tarel et al. [33] (d) Cai et al. [2]

(d) Ren et al. [28] (e) Estimated S (f) Estimated D (g) Ours

Figure 7. Visual comparisons for image dehazing. In contrast to the CNN-based methods [2, 28] which additionally use conventional methods to estimate atmosphere light, the proposed method directly estimates the atmosphere light in (e) and transmission map in (f) and thus leads to comparable results.

(a) Input (b) GT (c) SRCNN (d) VDSR (e) Cascade (f) Ours

Figure 8. Effectiveness of the proposed dual model. (c)-(f) show the comparisons between existing CNNs (including plain net and ResNet) and the proposed net in edge-preserving filtering and image deraining. The plain net (i.e., (c)), ResNet and its deeper version (i.e., (d) and (e)) generate results with significant artifacts. Quantitative evaluations are included in Table 6.

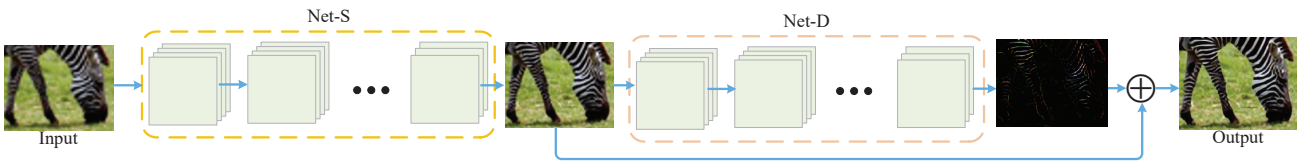


Figure 9. An alternative cascaded architecture that estimates the structure and details sequentially.

Table 6. Quantitative evaluation of different networks on the image filtering [38] and deraining [40] datasets in terms of PSNR.

Different nets	SRCNN	VDSR	Cascade	Ours
Filtering	32.0	41.6	42.0	**42.1**
Deraining	22.3	23.9	23.5	**24.1**

Table 7. Quantitative evaluation of the proposed dual composition loss function on the validation data of image dehazing in terms of PSNR and SSIM.

$(\lambda/\alpha, \gamma/\alpha)$	(0, 0)	(1, 0)	(0, 1)	(9, 9)
Avg. PSNR	21.13	26.27	26.45	26.43
Avg. SSIM	0.7449	0.8987	0.9139	0.9108

deeper architectures do not necessarily lead to better performance. In principle, a sufficiently deep neural network has sufficient capacity to solve any problem given enough training data. However, it is non-trivial to learn very deep CNN models for these problems while ensuring high efficiency and simplicity.

For experimental validation, we use the SRCNN and a deeper model, i.e., VDSR, for image filtering and deraining. The experimental settings are discussed in Section 4.

Sample results using the VDSR model are shown in Figure 8. While the residual learning (i.e., VDSR) approach performs better than the SRCNN, the generated images with the plain CNN model [5] contain blurry boundaries or rainy streaks (Figure 8(d)).

Although the proposed DualCNN consists of two branches, an alternative is to combine the Net-S and Net-D in a cascaded manner as shown in Figure 9. In this cascade model, the first stage estimates the main structure while the second stage estimates details. This network architecture is similar to the ResNet [13]. However, this cascaded architecture does not generate high-quality results compared to the proposed DualCNN (Figure 8(e) and Table 6).

Effect of the loss functions in DualCNN. We evaluate the effects of different loss functions on image dehazing. Table 7 shows that adding two regularization losses $\mathcal{L}_s$ in (2) and $\mathcal{L}_d$ in (3) significantly improves the performance.

Different architectures in DualCNN. We have used different network structures for the two branches of DualCNNs in the experiments in Section 4. It is interesting to test using the same structures for the two branches of a DualCNN. To this end, we set the two branches in a DualCNN using the network structures of SRCNN [5] and train the DualCNN according to the same settings used in the image super-resolution experiment. The trained DualCNN generates the results with higher average PSNR/SSIM values (30.3690/0.8603) than those of SRCNN (30.1496/0.8551) for $\times 4$ upsampling on the "Set5" dataset.

We further quantitatively evaluate the DualCNN when the two branches are the same on image deraining using synthetic rainy dataset [40]. Similar to the image super-resolution experimental settings, the two branches in the DualCNN are set to be the network structures of SRCNN [5] (SDCNN-S) and the network structures of VDSR [17] (SDCNN-D), respectively. Table 8 shows that DualCNN with deeper model generates better results when the architectures of two branches are the same. However, the DualCNN where one branch is SRCNN and the other one is VDSR performs better than SDCNN-D. This is mainly because the main structures of the input images are similar to those of output images. Deeper model used in "net-S" will introduce errors in the learning stage.

Table 8. Quantitative evaluation of two branches in DualCNN using the synthetic rainy dataset [40].

Two branches	SDCNN-S	SDCNN-D	Ours
Avg. PSNR	22.42	23.58	**24.11**

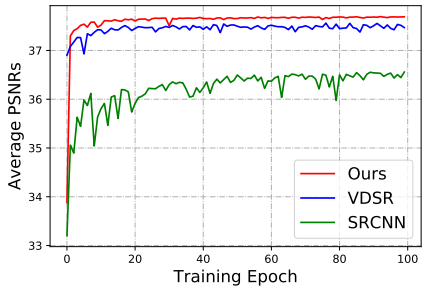


Figure 10. Quantitative evaluation of the convergence property on the super-resolution dataset (Set5, $\times 2$).

Convergence property. We quantitatively evaluate convergence properties of our method on the super-resolution dataset, i.e., Set5. Although the proposed network contains two branches compared to other methods [5, 17], it has the similar convergency property to the SRCNN and VDSR as shown in Figure 10.

6. Conclusion

In this paper, we propose a novel dual convolutional neural network for low-level vision tasks, called DualCNN. From an input signal, the DualCNN recovers both the structure and detail components, which can generate the target signal according to the problem formulation for a specific task. We analyze the effect of the DualCNN and show that it is a generic framework and can be effectively and efficiently applied to numerous low-level vision tasks, including image super-resolution, filtering, image deraining, and image dehazing. Experimental results show that the DualCNN performs favorably against state-of-the-art methods that have been specially designed for each task.

Acknowledgements. This work have been supported in part by the national key research and development program (No. 2016YFB1001001), NSFC (No. 61522203, 61732007 and 61772275), NSF CAREER (No. 1149783), NSF of Jiangsu Province (No. BK20170033), the National Ten Thousand Talent Program of China (Young Top-Notch Talent), and gifts from Adobe, Toyota, Panasonic, Samsung, NEC, Verisk, and Nvidia.

References

[1] H. Burger, C. Schuler, and S. Harmeling. Image denosing: Can plain neural networks compete with BM3D. In *CVPR*, 2012. 1

[2] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE TIP*, 25(11):5187–5198, 2016. 1, 2, 7

[3] Y.-L. Chen and C.-T. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *ICCV*, pages 1968–1975, 2013. 6

[4] C. Dong, Y. Deng, C. C. Loy, and X. Tang. Compression artifacts reduction by a deep convolutional network. In *ICCV*, pages 576–584, 2015. 1, 2

[5] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, pages 184–199, 2014. 1, 2, 3, 6, 8

[6] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE TPAMI*, 38(2):295–307, 2016. 1

[7] C. Dong, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, pages 391–407, 2016. 1

[8] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *ICCV*, pages 633–640, 2013. 1, 2

[9] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Trans. Image Processing*, 26(6):2944–2956, 2017. 5, 6

[10] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In *CVPR*, pages 3855–3863, 2017. 5, 6

[11] R. B. Girshick. Fast R-CNN. In *ICCV*, pages 1440–1448, 2015. 1

[12] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. In *CVPR*, pages 1956–1963, 2009. 6, 7

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 2, 8

[14] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, pages 5197–5206, 2015. 3

[15] V. Jain and H. S. Seung. Natural image denoising with convolutional networks. In *NIPS*, pages 769–776, 2008. 1, 2

[16] L.-W. Kang, C.-W. Lin, and Y.-H. Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE TIP*, 21(4):1742–1755, 2012. 6

[17] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pages 1646–1654, 2016. 1, 2, 3, 4, 5, 6, 8

[18] J. Kim, J. K. Lee, and K. M. Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, pages 1637–1645, 2016. 1

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012. 1

[20] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pages 4681–4690, 2017. 3

[21] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *CVPR*, pages 2736–2744, 2016. 6

[22] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia. Video super-resolution via deep draft-ensemble learning. In *ICCV*, pages 531–539, 2015. 1

[23] T.-Y. Lin, A. Roy Chowdhury, and S. Maji. Bilinear CNN models for fine-grained visual recognition. In *ICCV*, pages 1449–1457, 2015. 6

[24] S. Liu, J. Pan, and M.-H. Yang. Learning recursive filters for low-level vision via a hybrid neural network. In *ECCV*, pages 560–576, 2016. 1, 2, 4, 5

[25] D. R. Martin, C. 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 *ICCV*, pages 416–425, 2001. 3, 5

[26] G. Meng, Y. Wang, J. Duan, S. Xiang, and C. Pan. Efficient image dehazing with boundary constraint and contextual regularization. In *ICCV*, pages 617–624, 2013. 6

[27] J. S. J. Ren, L. Xu, Q. Yan, and W. Sun. Shepard convolutional neural networks. In *NIPS*, pages 901–909, 2015. 1

[28] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. Single image dehazing via multi-scale convolutional neural networks. In *ECCV*, pages 154–169, 2016. 1, 2, 3, 6, 7

[29] W. Shi, J. Caballero, F. Huszar, 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 *CVPR*, pages 1874–1883, 2016. 3

[30] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, pages 746–760, 2012. 6

[31] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NIPS*, pages 1988–1996, 2014. 1

[32] J. Tang, X. Shu, Z. Li, G. Qi, and J. Wang. Generalized deep transfer networks for knowledge propagation in heterogeneous domains. *TOMCCAP*, 12(4s):68:1–68:22, 2016. 1

[33] J. Tarel, N. Hautière, L. Caraffa, A. Cord, H. Halmaoui, and D. Gruyer. Vision enhancement in homogeneous and heterogeneous fog. *IEEE Intell. Transport. Syst. Mag.*, 4(2):6–20, 2012. 7

[34] R. Timofte, V. D. Smet, and L. J. V. Gool. A+: adjusted anchored neighborhood regression for fast super-resolution. In *ACCV*, pages 111–126, 2014. 1, 3

[35] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *NIPS*, pages 350–358, 2012. 1, 2

[36] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via L_0 gradient minimization. *ACM TOG*, 30(6):174:1–174:12, 2011. 5

[37] L. Xu, J. S. J. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *NIPS*, pages 1790–1798, 2014. 4

[38] L. Xu, J. S. J. Ren, Q. Yan, R. Liao, and J. Jia. Deep edge-aware filters. In *ICML*, pages 1669–1678, 2015. 1, 2, 4, 5, 8

[39] L. Xu, Q. Yan, Y. Xia, and J. Jia. Structure extraction from texture via relative total variation. *ACM TOG*, 31(6):139:1–139:10, 2012. 5

[40] H. Zhang, V. Sindagi, and V. M. Patel. Image de-raining using a conditional generative adversarial network. *CoRR*, abs/1701.05957, 2017. 1, 2, 5, 6, 8

[41] Q. Zhang, L. Xu, and J. Jia. 100+ times faster weighted median filter (WMF). In *CVPR*, pages 2830–2837, 2014. 5