

# Image Restoration Using Deep Regulated Convolutional Networks

Peng Liu  
University of Florida, USA  
pliu1@ufl.edu

Xiaoxiao Zhou  
Sun Yat-Sen University, China  
zhouxx5@mail2.sysu.edu.cn

Yangjunyi Li  
University of Florida, USA  
yangjuniyili92@ufl.edu

El Basha Mohammad D  
University of Florida, USA  
mdelbasha@ufl.edu

Ruogu Fang  
University of Florida, USA  
ruogu.fang@bme.ufl.edu

## Abstract

While the depth of convolutional neural networks has attracted substantial attention in the deep learning research, the width of these networks has recently received greater interest [24, 17, 16]. The width of networks, defined as the size of the receptive fields and the density of the channels, has demonstrated crucial importance in low-level vision tasks such as image denoising and restoration [16]. However, the limited generalization ability, due to increased width of networks, creates a bottleneck in designing wider networks. In this paper we propose Deep Regulated Convolution Network (RC-Net), a deep network composed of regulated sub-network blocks cascaded by skip-connections, to overcome this bottleneck. Specifically, the Regulated Convolution block (RC-block), featured by a combination of large and small convolution filters, balances the effectiveness of prominent feature extraction and the generalization ability of the network. RC-Nets have several compelling advantages: they embrace diversified features through large-small filter combinations, alleviate the hazy boundary and blurred details in image denoising and super-resolution problems, and stabilize the learning process. Our proposed RC-Nets outperform state-of-the-art approaches with large performance gains in various image restoration tasks while demonstrating promising generalization ability. The code is available at <https://github.com/cswin/RC-Nets>.

## 1. Introduction

Image restoration aims at recovering a high-quality image from a corrupted one. In particular, image denoising and single image super-resolution (SR) are two of the most important tasks. Recent works in both image denoising and SR consist mainly of learning-based methods [4, 14, 27, 18, 16], which learn mapping between

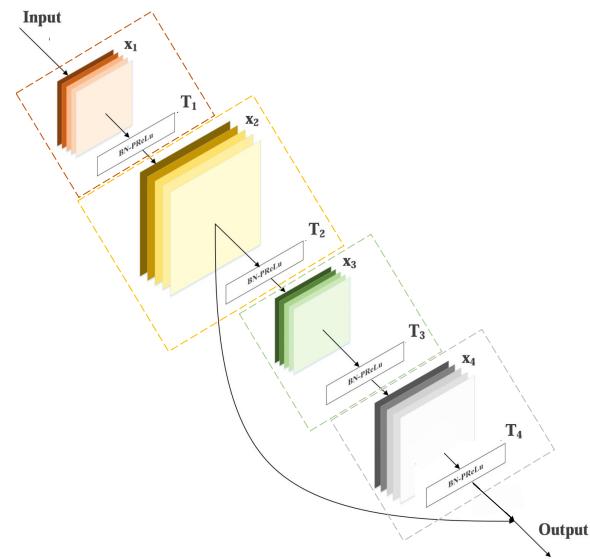


Figure 1. A regulated convolution block with 4 composite units (dotted boxes). The first and third group-squares present  $1 \times 1$  convolution; the second and last ones indicate large and small convolution, respectively. The large convolution is regulated by the small one.

spaces of the corrupted images and the high-quality images. Among them, Wider Inference Networks (WIN) [16] have demonstrated substantial performance gain in additive white Gaussian noise denoising. By adopting large reception fields (filters) and dense channels in convolutional neural networks (CNNs), WIN can accurately exploit prominent image features to infer high-quality images.

However, wider networks such as WIN suffer from low generalization ability. For example, WIN only achieves superior performance in additive Gaussian noise removal with mediocre performance on image SR. To overcome this critical constraint, we investigated the network architecture and found three inherent limitations that constrain its general-

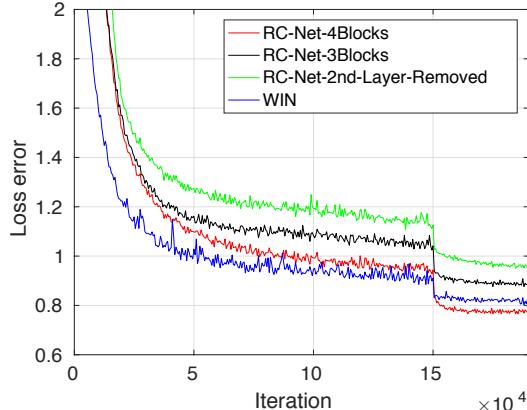


Figure 2. Comparison of validation error of RC-Nets with different number of blocks, a RC-Net having the 2<sup>nd</sup> layer removed, and WIN during training.

ization capability.

First, large convolution filters introduce bias. While large convolution filters (e.g.  $128 \times 7 \times 7$ ) improve the performance of extracting prominent local features (corners and edge/color conjunctions) [25], they simultaneously introduce bias in the network to learn specific features of similar pixel distributions [16]. The introduced “bias” can boost performance significantly in one single task such as Gaussian denoising, but will degrade the performance in other tasks, such as image SR, deblurring and inpainting. This is one of the reasons that most learning-based image restoration methods [14, 27, 18] use smaller convolution filters (e.g.  $3 \times 3$ ).

Second, large convolution filters lead to significant performance fluctuations in learning (see Fig. 1). The larger the convolution filters, the greater the feature variance will be, especially when noisy level is high in the training images. Small convolution filters (e.g.  $3 \times 3$ ) tend to capture subtle features favoring shared weights [21]. This instability issue limits the filter size in learning-based image restoration methods [18].

Third, large convolution filters require expensive computation. As convolution operations dominate the computation times in CNNs [7], small filter sizes allow networks go deeper without significantly increasing the computational cost. Most learning-based image restoration methods [14, 27] are designed to be very deep (20 to 30 layers) by embedding  $3 \times 3$  filters only, as the generalization capability relies on the network’s depth.

From the three limitations above, our focus is concentrating on how to balance between the generalizability and performance.

In this work, we investigate a scalable network structure for image restoration with both generalizability and high performance. Inspired by WIN [16], we propose a novel cascaded scalable architecture, called Deep Regulated Con-

volution Network (RC-Net) (Fig. 3), which comprises of regulated sub-network blocks (Fig. ) cascaded by shortcut connections [10], to handle both image denoising and SR.

Our proposed RC-Net addresses the aforementioned three limitations. The key contribution is to take advantages of fusing both large and small filters, by regulating the large convolution filters by small filters. This combination of varying filter sizes can both extract the prominent image features and improve the network’s generalizability at reasonable computational cost.

We present comprehensive experiments on both image denoising and SR to evaluate our networks. We show that 1) Our deep regulated convolutional networks can increase accuracy via regulated convolution, producing results substantially better than previous networks (DnCNN [27], RED-Net [18], and WIN [16]) on image denoising; 2) Our deep regulated convolutional networks can be generalized to deal with single image SR problems, producing impressive results of high-resolution images that are more appealing for human vision.

## 2. Related Work

**Deep Learning for Image Restoration.** Recently, deep learning based methods have shown impressive performance on both high-level [15, 21, 10] and low-level [27, 14, 18, 4] vision research fields, compared to the non-CNN based models [2, 6, 22, 12]. “Deeper is better” has been considered as a design criteria of building convolutional neural networks (CNN). The preference of deep and thin CNNs stem comes from the success of deep networks in high-level vision [15]. Deep learning based approaches boost the performance with the cost of increasing complexity and network depth.

Despite of having high capability of nonlinear representation, deep neural networks usually encounter potential problems including gradient vanishing, over-fitting, degradation, and high inference cost. Diverse techniques have been utilized to cope with these obstacles; RED-NET [18] employs a number of skip-connections (shortcuts) to link convolution layers with mirrored deconvolution layers; VDSR [14] adopts very small ( $3 \times 3$ ) convolution filters to restrict the amount of parameters and exploits a skip-connection to carry the input to the end layer and reconstruct residuals, both of which contribute to speeding up training and reducing the degradation issue; DnCNN accelerates training by assembling batch normalization layers (BN) [13] and  $3 \times 3$  filters in every convolution layer, produces the residual between high-quality and corrupted images.

**Effective Feature Extraction and Learning.** WIN [16] is built with a competitive shallow network. which is naturally superior in terms of training efficiency. WIN is

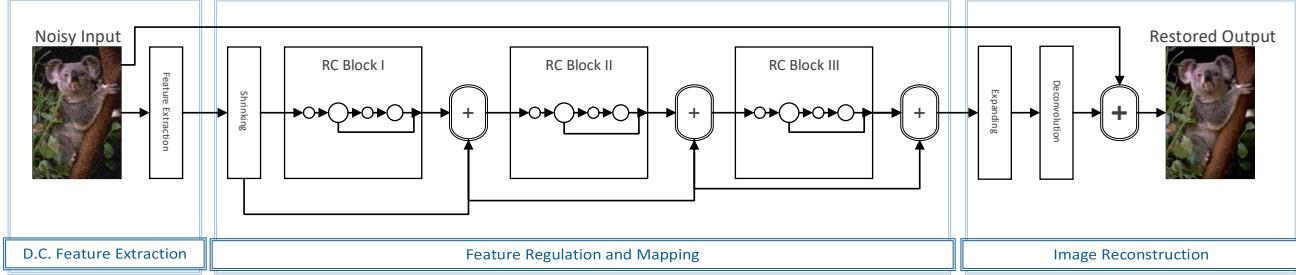


Figure 3. A deep RC-Net with four RC-blocks. In RC-blocks, the largest, medium and smallest size of circles denote a composite unit using large, small, and  $1 \times 1$  convolution filters, respectively. The  $\oplus$  represents a summation computing and indicates residual learning.

stacked with 4 composite layers of three consecutive operations: convolution (Conv) layer with size  $128 \times 7 \times 7$ , batch normalization (BN) Layer [13], and rectified linear unit (ReLU) Layer [20]. In addition, a skip-connection linking the input with the end layer allows the network to predict a residual image.

With the following three key contributions, WIN offers a performance breakthrough in the Gaussian image denoising task: 1) Utilizing highly dense convolution layers composed of large filters to extract prominent local features; 2) Embedding BN layers to learn the mean and variance of feature maps associated with the pixel-distribution of training images. BN is especially helpful when pixel distributions are similar due to the same type of corruption. BN also performs as a memory storage to keep the shareable mean and variance; 3) Linking input to the end layer with a skip-connection can lead to residual learning and carrying shareable information from corrupted images to the end layers, both of which contribute to improving the accuracy of the loss-error calculation via comparing with the ground-truth images.

However, WIN also suffers from a number of limitations despite of the substantial performance margin. WIN is not able to be extended to other sub-problems of image restoration, such as SR. This is because the difference between low-resolution (LR) images and high-resolution (HR) images is subtle and not merely Gaussian distribution. Furthermore, there are fewer similarities shared by the pixel-level distribution features among the LR images. In this case, in addition to the disadvantages mentioned in Section 1 on large filters, BN layers cannot help but restrain the network's generalizability due to data normalization. This is especially important to SR because the mapping between a subtle residual from the normalized data is no longer easier than mapping from the original data.

Convolution is the primary operation in CNNs. The number and size of filters determine the type of features extracted and the computational cost. Herein, design of the convolution filter structures is a key component of developing deep CNN models to optimize the performance.

However, most deep models primarily adopt  $3 \times 3$  convolution filters and interpolate using  $1 \times 1$  convolution filters to reduce the number of feature-maps, such as bottleneck blocks used in ResNet [10] and DenseNet [11]. To challenge the current trend, we propose to regulate the large filters with smaller ones to achieve a balance between performance generalizability, which remarkably outperform the state-of-the-art using only small filters and bottleneck connections.

### 3. Regulating Convolution Nets

Consider a corrupted image  $x_0$  is passed through a convolutional network. The network intends to learn a mapping function  $F$  between the corrupted image  $x_0$  and a noise-free image  $y$ . The convolutional network contains  $L$  convolution layers (Conv), each of which implements a feature extraction transition:

$$y_l = \text{Conv}(x_l, f_l, n_l, c_l) \quad (1)$$

where  $l$  indexes the layer,  $x_l$  denotes the  $l$ 's input, and  $f_l$ ,  $n_l$ , and  $c_l$  represent the filter size, filter number, and channel number, respectively.  $y_l$  are the feature-maps extracted from  $x_l$  by  $\text{Conv}(\cdot)$ .

As the top and bottom layers have different functional attentions [25], the network can be decomposed into three parts (see Fig. 3): densely convolved feature extraction, feature regulation and mapping, and image reconstruction.

**Densely convolved feature extraction.** We use a considerable amount of large filters in the first two [25] convolutional layers *densely convolutional feature-extraction layers* to extract diverse and representative features for feature mapping and spatial transformation. This part is similar to WIN, but WIN adopts dense convolutions in all layers and does not distinguish between the hierarchical characteristics of the bottom-up layers. Based on the empirical results of WIN, we define densely convolved features extracted from the  $l^{th}$  layer as:

$$y_l = \text{Conv}(x_l, f_l, n_l, c_l)_{f \geq 7 \times 7, n \geq 128} \quad (2)$$

**Composite unit.** Motivated by [10, 27, 5], we combine a convolution (Conv) layer, a batch normalization (BN) layer [13], and a Parametric Rectified Linear Unit (PReLU) layer [8] as a composite unit in our proposed RC-Net, except for the last layer, which is a single deconvolutional layer. Following [11], we define the mapping of the composite unit as a composite function  $T(\cdot)$ :

$$T(\cdot) = PReLU(BN(Conv(\cdot))) \quad (3)$$

$$T_l(x_l) = PReLU(BN(y_{l-1})) \quad (4)$$

**Feature regulation and mapping.** We divide the network into multiple regulated convolution *RC-blocks*, which are cascaded to perform feature extraction, mapping, and transformation. We assume a RC-Net contains  $m$  RC-blocks, each of which comprises of four composite units; large and small convolution filters and two other  $1 \times 1$  convolution filters for reducing the feature map dimension and regulating features extracted from the previous composite or RC-blocks. Note, the  $1 \times 1$  convolution filters cannot change but only exert combing effect to the features generated from the preceding composite.

The key to a RC-block is to use smaller convolution filters to regulate the feature outputs from the proceeding larger filters. This regulation processing aims to balance the feature extraction so that the features generated from the larger filters can fuse information from both the larger receptive fields and the smaller ones for finer details.

To ensure a RC-block having sufficient representative capability for feature mapping, we also suggest the composite unit embedding larger filters in each RC-block to adopt  $n \geq \frac{n^d}{2}$  filters with size of  $f \geq f^d \times f^d$ , where  $n^d$  and  $f^d$  denote the filter number and the filter size in densely convolutional feature-extraction layers, respectively.

**Residual learning.** Without residual learning, traditional convolutional networks pass the output of the  $l^{th}$  layer as the input to the  $(l+1)^{th}$  layer [15]. If  $l$  represents a composite unit, there is a transition:  $y_l = T_l(y_{l-1})$ . Residual learning [10] can adopt a skip-connection (shortcut) to perform an identity mapping:

$$y_l = T_l(y_{l-1}) + y_{l-1} \quad (5)$$

RC-Net adopts residual learning in three different locations. Within each RC-block, there is a skip-connection linking the output of larger filters to the end of the block to fuse the information generated from the smaller filters and forms residual learning as:

$$y_{l_{fs}} = T_{l_{fs}}(y_{l_{f_l}}) + y_{l_{f_l}} \quad (6)$$

where  $l_{fs}$  and  $l_{f_l}$  denote the layer using small and large filters, respectively. The second residual learning is performed by another skip-connection, which connects the outputs of two adjacent RC-blocks or connects the first RC-block and the composite unit that precedes it. Motivated by [14], the third residual learning is achieved by adding the corrupted image to the output of the last layer, which is a deconvolutional layer that converts the preceding feature-maps to an image.

**Scale controlling.** To make RC-Nets more compact, motivated by [5], we introduce two  $1 \times 1$  composite units as transition layers, refereed as “Shrinking” and “Expanding”, which are shown on Figure . After densely convolutional feature-extraction layers, we reduce the number of feature-maps by “Shrinking”, in which, inspired by [17], we suggest to adopt  $n \geq \frac{n^d}{2}$  filters to remain sufficient features for following RC-blocks, where  $n^d$  denotes the filter number in densely convolutional feature-extraction layers.

After feature regulation and mapping, we use another transition layers to expand feature-maps so that there are sufficient various features that can be provided for image reconstruction.

**Image reconstruction.** RC-Nets reconstruct a noise-free image  $y$  by adding the input (corrupted) image  $x_0$  to the one generated from the RC-Net’s last layer: a deconvolutional layer, which can be denoted as:  $Deconv(\cdot)$ . Thus we can define the image reconstruction as:

$$y = x_0 + Deconv(x, f, n, c)_{s,n=1} \quad (7)$$

where  $s, n = 1$  represents one small filters used to inverse the preceding feature-maps finely.

## 4. Experiments and Results

We perform experiments on both image denoising and super-resolution.

### 4.1. Datasets

**Image Denoising.** The training data for image denoising consists of 91 images from Yang et. al. [23] and an additional 200 images from the Berkeley Segmentation Dataset BSD200. We add Gaussian noise at four different levels ( $\sigma = 10, 30, 50, 70$ ) to train the RC model separately for each noise level. Noise is added using the *randn* function in MATLAB. We train other comparing methods using the same dataset of 291 images. To keep consistent with the original paper [27], DnCNN follows [1] to use 400 images of size  $180 \times 180$  for training.

To evaluate the denoising performance of each method, we use the BSD200 test dataset and Set12. Due to various

Table 1. Comparison of structure detail of RC-Net; RC-3-Blocks which removes one RC-Block from RC-Net; RC-2nd-Layer-Removed which removes Composite unit(2) from RC-Net; WIN consists of three parts as shown in table. number of parameters calculate the total parameter number of network.Comparison is based on image denoising performance, average PSNR (dB) / SSIM / Run Time (s), are evaluated on the BSD200-test with noise level  $\sigma = 50$ .

Layers	Output Size	RC-Net-4- Blocks	RC-Net-3- Blocks	RC-2nd-Layer-Removed	WIN
Composite unit (1)	128	$7 \times 7$ conv, stride 1			
Composite unit (2)	128	$7 \times 7$ conv, stride 1	$7 \times 7$ conv, stride 1	removed	$\left[ \begin{array}{l} \text{Conv + BN + ReLU} \\ 1 \times 128 \times 7 \times 7 \end{array} \right] \times 1$
Transition (1)	64	$1 \times 1$ conv, stride 1			
RC-blocks	64	$\left[ \begin{array}{l} 1 \times 1 \text{ conv} \\ 7 \times 7 \text{ conv} \\ 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{array} \right] \times 4$	$\left[ \begin{array}{l} 1 \times 1 \text{ conv} \\ 7 \times 7 \text{ conv} \\ 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{array} \right] \times 3$	$\left[ \begin{array}{l} 1 \times 1 \text{ conv} \\ 7 \times 7 \text{ conv} \\ 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{array} \right] \times 4$	$\left[ \begin{array}{l} \text{Conv + BN + ReLU} \\ 128 \times 128 \times 7 \times 7 \end{array} \right] \times 3$
Transition (2)	128	$1 \times 1$ conv, stride 1			
Deconvolution	1	$3 \times 3$ deconv, stride 1			

Comparison					
# of parameters		1.8M	1.6M	1.0M	2.4M
PSNR (dB)		32.48	30.59	31.57	31.79
SSIM		0.9112	0.8575	0.8933	0.8831
Time (s)		0.5199	0.6091	0.6417	0.2667

versions of Set12, we resized images of the 12 standard image to be  $481 \times 321$ , as the same size as the majority of images in the BSD200 test dataset. Both datasets are used in the image denoising experiment in Table 2 and Table 3.

**Single Image Super-Resolution.** We use the same dataset of 291 images at three different scale factors ( $\times 2$ ,  $\times 3$ ,  $\times 4$ ) to train the RC-Net model separately. WIN and VDSR also share the same training dataset “291”. SR-CNN [4] uses a large ImageNet dataset [3] as in its original paper [4]. For evaluation, two datasets are used: BSD200 [19] from the Berkeley Segmentation Dataset [19] and Set14 [26].

## 4.2. Training

Our RC-Net has four RC-blocks for all datasets. RC-Net has a depth of 21 layers, where 20 of them are composite units (see Section 3). The remaining layer is a deconvolutional layer (Deconv). To avoid the “dead features” [25] issue in ReLU [9], we adopt the Parametric Rectified Linear Unit (PReLU) [9] for the activation function after each batch normalization layer (BN) [13].

All models adopt Stochastic Gradient Descent (SGD) with mini-batch size of 64 for image denoising and 128 for SR. For model optimization, we introduced the BN layer to reduce the internal covariate shift leading to an accelerated learning speed. For weight initialization, we follow the method described in [9]. This is especially suitable for the

networks adopting ReLU or PReLU.

All experiments are trained over 50 epochs (5,000 iterations per epoch) totaling 250,000 iterations. Learning rate is initially set to 0.1 and is divided by 10 after every  $15 \times 10^4$  iterations. Weight decay is set to 0.0001 and momentum to 0.9. We follow the practice in [27, 18, 14] for both image denoising and SR. Data-augmentation is performed to increase the size and variance of training samples. A  $41 \times 41$  crop with stride 14 is randomly sampled from an image or a horizontally flipped copy to generate image patches for additional training samples. All training is processed on 8 GeForce GTX TITAN Xp GPUs.

## 4.3. Image Denoising

Both quantitative and qualitative comparisons of various denoising and SR methods with our proposed RC-Net are provided. In Table 2 and Table 3, we provide quantitative evaluations of five different methods: BM3D [2], RED-Net [18], DnCNN [27], WIN [16], and RC-Net. PSNR, SSIM, and run time are used as evaluation criteria.

From the results, our proposed RC-Net method significantly outperforms BM3D, RED-Net, and DnCNN on both datasets. When compared with WIN, our RC-Net method has relatively similar performance when noise level is 30. For example, on the dataset BSD200, our RC-Net method generates a PSNR of 33.57 dB while WIN generates 33.62 dB. While RC-Net and WIN have extremely close PSNR, RC-Net still outperforms WIN in terms of SSIM. With the

Table 2. The average results of PSNR (dB) / SSIM / Run Time (seconds) of different methods on the BSD200-test [19] (200 images). All methods are applied on several noise level ( $\sigma = 10, 30, 50, 70$ ). The best results are highlighted in bold.

$\sigma$	PSNR (dB) / SSIM / Time (s)				
	BM3D [2]	RED-Net [18]	DnCNN [27]	WIN [16]	RC-Net
10	34.02 / 0.9182 / 1.01	32.96 / 0.8963 / 60.73	34.60 / 0.9283 / 13.49	35.83 / 0.9494 / 28.72	<b>36.36 / 0.9541 / 32.55</b>
30	28.57 / 0.7823 / 1.23	29.05 / 0.8049 / 60.27	29.13 / 0.8060 / 13.48	<b>33.62 / 0.9193 / 31.94</b>	33.57 / 0.9271 / 33.52
50	26.44 / 0.7028 / 2.02	26.88 / 0.7230 / 66.62	26.99 / 0.7289 / 12.55	31.79 / 0.8831 / 23.77	<b>32.48 / 0.9112 / 33.52</b>
70	25.23 / 0.6522 / 1.98	26.66 / 0.7108 / 66.99	25.65 / 0.6709 / 13.42	30.34 / 0.8362 / 23.90	<b>31.17 / 0.8795 / 32.81</b>

Table 3. The average PSNR(dB) / SSIM / Run Time (seconds) of different methods on the resized 12 standard testset (12 images). All methods are applied on several noise level ( $\sigma = 10, 30, 50, 70$ ). The best results are highlighted in bold.

$\sigma$	PSNR (dB) / SSIM / Time (s)				
	BM3D [2]	RED-Net [18]	DnCNN [27]	WIN [16]	RC-Net
10	34.28 / 0.9208 / 0.91	30.48 / 0.8610 / 61.89	33.30 / 0.9096 / 12.79	37.24 / 0.9546 / 12.03	<b>38.54 / 0.9627 / 29.56</b>
30	29.09 / 0.8287 / 0.99	30.48 / 0.8610 / 62.71	28.19 / 0.8151 / 12.52	34.89 / 0.9162 / 12.77	<b>35.18 / 0.9409 / 29.50</b>
50	25.23 / 0.6522 / 1.98	28.03 / 0.7988 / 63.66	25.85 / 0.7522 / 12.48	32.99 / 0.8825 / 12.37	<b>34.55 / 0.9270 / 30.44</b>
70	25.20 / 0.7149 / 1.75	27.95 / 0.7950 / 63.20	23.75 / 0.6890 / 12.41	30.91 / 0.8263 / 11.75	<b>32.87 / 0.8961 / 30.43</b>

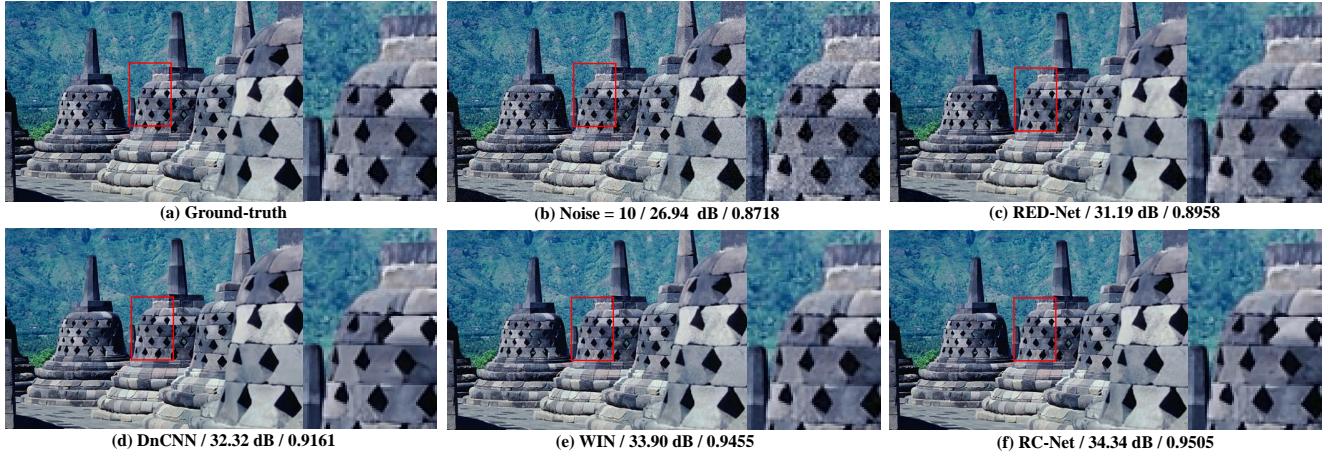


Figure 4. Visual results of one image from BSD200-test with  $\sigma = 10$  along with PSNR(dB) / SSIM.

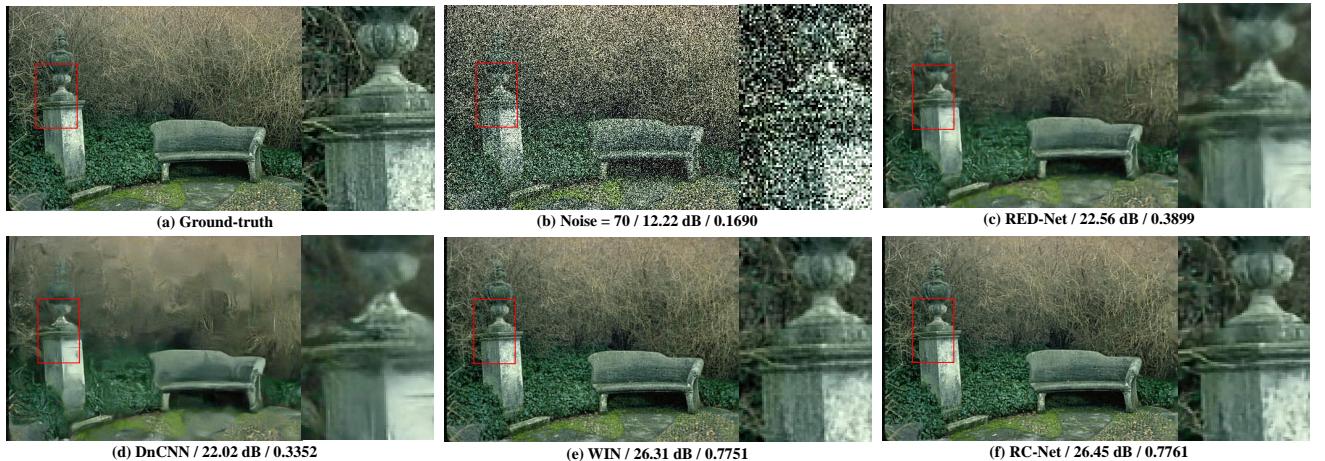


Figure 5. Visual results of one image from BSD200-test with  $\sigma = 70$  along with PSNR(dB) / SSIM.

increased noise level, the denoising results of RC-Net surpasses WIN remarkably. For example, on Set12, RC-Net generates an average PSNR of 32.87 dB, while WIN produces a PSNR of 30.34 dB.

We illustrate visual comparisons of four different methods: RED-Net, DnCNN, WIN, and RC-Net in Figure 4 and Figure 5 with sample images from BSD200 dataset. In Figure 4, our RC-Net method excels in maintaining the detailed particle structure on the building’s surface while other methods fail to keep this important texture information. Figure 5 is a challenging image due to a great quantity of plants with the same color and the high noise level. In this challenging case, our method still excels in restoring the pillar’s texture features, as well as clear contours and details of the chair and branches.

The visual results show a well match with quantitative results. Our method has improved performance compared to the state-of-the-art image denoising approaches, especially when the noise level is high. Our RC-Net gains the best SSIM term among all test datasets as well as the best human visual image result.

#### 4.4. Super-Resolution

To demonstrate the generalizability of RC-Net, we show illustrations of single image super-resolution using our RC-Net and other state-of-the-art approaches (A+ [22], SR-CNN [4], VDSR [14], and DnCNN [27]) in Figure 6 and Figure 7. When we scale an image with  $2\times$  factor as shown in Figure 6, it is obvious that our method obtains superior eyes texture and boundary of baboon compared to the state-of-the-art methods. In Figure 7, our method has demonstrated notable capability to maintain the image details in SR by capturing the wrinkle and eyebrow texture while the comparing state-of-the-art methods, which tend to lose the texture information and result in a blurred image.

Furthermore, most existing methods use multiple networks to handle different scale factors in image SR, while one general network that can perform SR at various scale factors is preferred. Figure 8 exhibits the ability of a single RC network (RC-blind) to perform SR at different scale factors. We compare the performance of recovered images at multiple scales using RC-blind (top) with VDSR (bottom). From Figure 8, we can observe that while both methods can handle different scale factors, our proposed RC-Net outperforms VDSR in terms of clarity and sharpness.

## 5. Discussion

Ostensibly, the specific structure design on RC-Nets (Table 3) that differs from other deep convolutional networks merely in the cascaded RC-Blocks, which include large-small filter combination for regulation (Figure ). However, this small modification leads to a remarkable image restoration performance of the networks.

**Diversified Feature Representation.** The most direct consequence of introducing RC-Blocks is efficiency in diversified feature extraction . Table 3 shows the comparison between RC-Nets and WIN, which is a simplified structure of RC-Net without RC-blocks but more parameters. It is apparent that RC-Nets outperforms WIN even with with a much smaller parameter number (1.8 M vs. 2.4 M). Even though the performance degrades a little after we remove one RC-block, RC-3-Blocks Nets with only 1.6 M parameters still outperforms state-of-the-art methods and similar to WIN. We also further remove one Composite Unit (Table 3 and shows that with a minor degradation of performance, the parameters number decrease from 1.8 M to 1.0 M. These variations and comparisons demonstrate the effectiveness of our RC-Block in feature utilization and the substantial potential for a more compact network.

**Handling Training Fluctuation.** There is connection between filter sizes and loss-error curve as shown in Figure 1. WIN shows strong fluctuating even in platform period. However, RC-Nets substantially improve the stability in the loss error during the training process and outperforms WIN after around  $15 \times 10^4$  iterations when the entire dataset has been traversed.

**Enhanced Generalizability.** Cascaded RC-Blocks overcomes the limitation in generalizability of merely using dense convolutional layers in WIN. RC-Nets show superb visual results in single image super-resolution (Figure 6 and Figure 7). The collaborative effect large-small filter combination in RC-Blocks encourages feature extraction with detail preservation as both dominant and detailed features are collected by large and small filters. The overall performance of RC-Nets in both image restoration indicates the enhanced generalization ability with cascaded and specially RC-Blocks. It also implies the great potential of our proposed RC-Nets in solving broader image restoration tasks.

## 6. Conclusion

We proposed a new convolutional network architecture, referred as Deep Regulated Convolutional Network (RC-Net), for image restoration. It introduces large-small filter combinations (RC-Block) where small convolution filters regulate the features extracted by the large ones. In our experiments, we demonstrated that RC-Nets achieved state-of-the-art performance across several highly competitive datasets for multiple image restoration tasks. Moreover, RC-Nets need substantially fewer parameters and less computational cost to outperform state-of-the-art approaches with large margins, with appealing performance in structural information preservation and generating results appealing for the human visual systems. Owing to the regu-

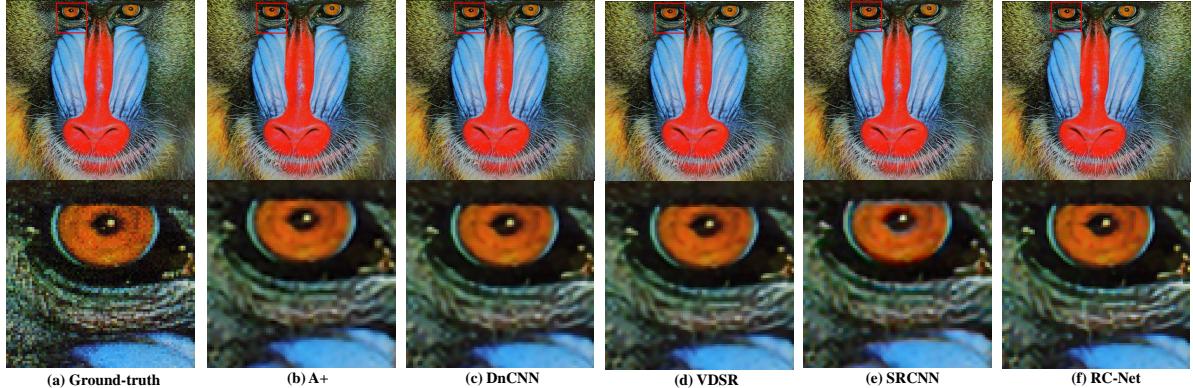


Figure 6. Visual results of one image from Set14 with scale factor x2 along with PSNR(dB) / SSIM.

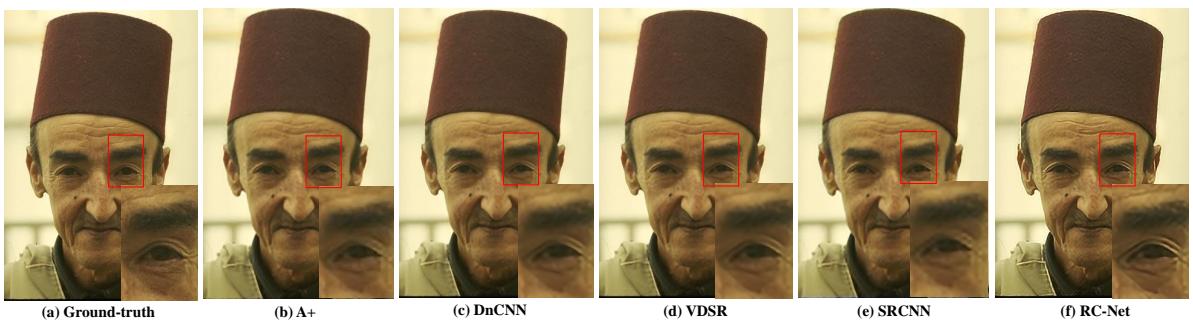


Figure 7. Visual results of one image from B100 with scale factor x3 along with PSNR(dB) / SSIM.

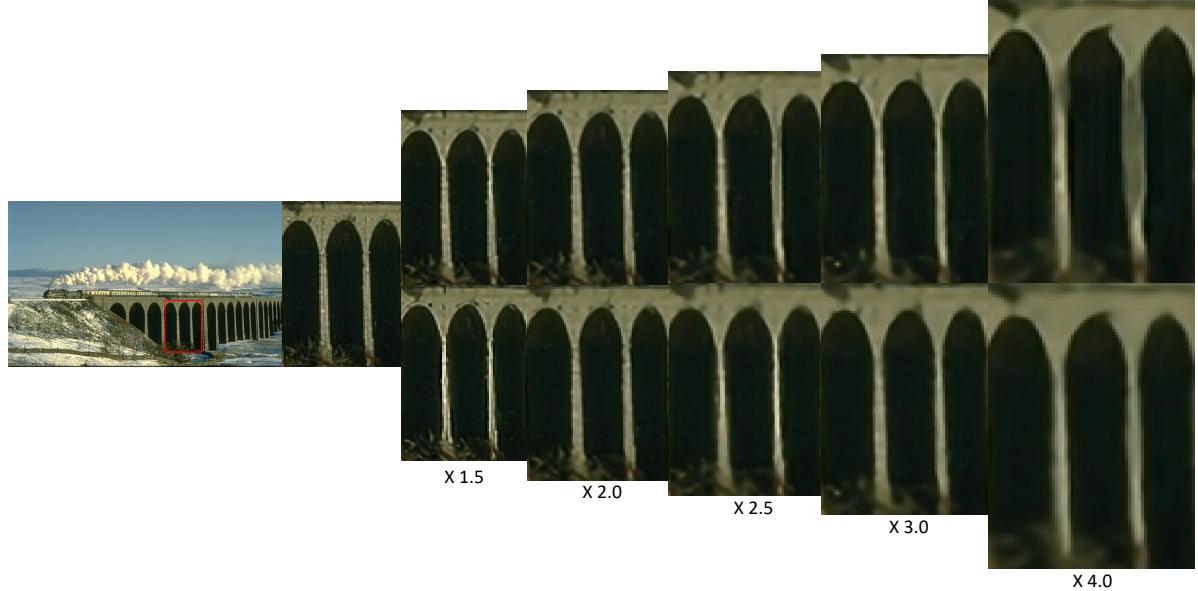


Figure 8. Visual results of single network for multiple scale factors. ROIs are selected and enlarged on the right. **Top:** Restoration results within RC-blind network, RC-blind is trained on scale factor =  $\times 2$ ,  $\times 3$ , and  $\times 4$ . **Bottom:** SR results generated by VDSR

lated convolutions, RC-Nets can balance the feature extraction and generalizability by fusing information from both the representative features extracted from the larger receptive fields and the smaller textures obtained from finer de-

tails.

The advantages of RC-Net lie in the balance between large and small convolutional filters, a simple combination that naturally integrates diversified feature extraction, sta-

bility in the training process, and enhanced generalizability. This design allows regularized diversified feature representation throughout the networks, which consequently yielding more efficient and compact models with highly competitive image restoration performance. Due to their efficient and compact model for feature representation, we plan to investigate the generalization ability of RC-Nets to various computer vision tasks that build on convolutional neural networks.

## References

- [1] 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*, 39(6):1256–1272, 2017. [4](#)
- [2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. BM3D image denoising with shape-adaptive principal component analysis. In *SPARS’09-Signal Processing with Adaptive Sparse Structured Representations*, 2009. [2, 5, 6](#)
- [3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. [5](#)
- [4] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pages 184–199. Springer, 2014. [1, 2, 5, 7](#)
- [5] C. Dong, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision*, pages 391–407. Springer, 2016. [4](#)
- [6] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2862–2869, 2014. [2](#)
- [7] K. He and J. Sun. Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5353–5360, 2015. [2](#)
- [8] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. [4](#)
- [9] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. [5](#)
- [10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [2, 3, 4](#)
- [11] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016. [3, 4](#)
- [12] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015. [2](#)
- [13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. [2, 3, 4, 5](#)
- [14] J. Kim, J. Kwon Lee, and K. Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016. [1, 2, 4, 5, 7](#)
- [15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. [2, 4](#)
- [16] P. Liu and R. Fang. Wide inference network for image denoising. *arXiv preprint arXiv:1707.05414*, abs/1707.05414, 2017. [1, 2, 5, 6](#)
- [17] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. *arXiv preprint arXiv:1709.02540*, 2017. [1, 4](#)
- [18] X.-J. Mao, C. Shen, and Y.-B. Yang. Image restoration using convolutional auto-encoders with symmetric skip connections. *arXiv preprint arXiv:1606.08921*, 2016. [1, 2, 5, 6](#)
- [19] 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*, volume 2, pages 416–423, July 2001. [5, 6](#)
- [20] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. [3](#)
- [21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. [2](#)
- [22] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Asian Conference on Computer Vision*, pages 111–126. Springer, 2014. [2, 7](#)
- [23] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010. [4](#)
- [24] S. Zagoruyko and N. Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016. [1](#)
- [25] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014. [2, 3, 5](#)
- [26] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010. [5](#)
- [27] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *arXiv preprint arXiv:1608.03981*, 2016. [1, 2, 4, 5, 6, 7](#)

## A. Shortcomings of using Batch Normalization

A RC-Net using batch normalization (BN) does not perform as well in Super-resolution (SR) as it does in the Gaussian denoising task. On one hand, following the feature extraction in the regulated convolution layers, BN is able to enhance the high-frequency features (lines, edges, corners, etc.) by summarizing mean and variance of feature-maps during training into two learned parameters. The enhanced features benefit both image denoising and single image SR. On the other hand, for Gaussian denoising, because the noisy images fed into networks follow Gaussian pixel-distribution, BN normalizes the information flow and keeps the distribution of each layer's inputs the same as the original input (noisy images) during training. Nevertheless, for SR, the information processed by BN is shifted and largely different from the original input (low-resolution images). This gives rise to negative effects on SR when the predicated high-resolution image is generated from the summation of the shifted signals and the original one through a skip-connection linking the input to the end of the last layer. The high-frequency features still can be maintained but the low-frequency features are latently lost during shifting and normalization. This is the reason why the visual results from RC-Nets for SR are better to the human vision system but PSNR and SSIM are less than the most competitive state-of-the-art methods.

We may overcome this drawback in three potential ways: (1) Follow the work of DnCNN by directly predicting the residual between the corrupted image and the noise-free one. In this case, we remove the input-to-end skip-connection and BN layers in the first and last layers. (2) Simply remove all BN layers from RC-Nets; (3) Dynamically route the training and testing to a specific part of a RC-Net based on the property of an input. (1) and (2) are simpler and may impact the denoising performance. (3) needs an additional learning process to allow the network knows how and where to route the information. We consider (3) as a future work and implemented (2). The results for SR (see **B**) demonstrate that BN indeed has the negative effects on SR, and that RC-Nets without BN achieve competitive PSNR and SSIM with state-of-the-art methods. *Code and models are available at <https://github.com/cswin/RC-Nets>.*

## B. More Results

More visual results are shown below to demonstrate the competitive performance of RC-Nets for image restoration. We illustrate image results on datasets BSD200-test for image denoising with noise level  $\sigma = 30$  (Figure 9) and 50 (Figure 10). While RC-Nets outperform all competing methods, they perform exceptionally well in complex images and at high noise level. Our method reconstructs well

the shape and edges of left eye in Figure 9 as presented in the ROI. Similarly, in Figure 10, contours are clean and visually discernible.

Figure 11 and Figure 12 are visual results of single image SR with scale factor  $\times 3$ . We removed all BN layers of a RC-Net and call this network as “RC-Net without BN”, while the original RC-Net keeping all BN layers is called “RC-Net with BN”. Figure 11 shows an image from dataset Set5. The appealing human visual results showing sharp and bright hair recovered by RC-Net with BN. However, removing BN layers leads a better quantitative evaluation in terms of PSNR and SSIM, and comparable performance to the state-of-the-art methods VDSR and DnCNN. We illustrate one more SR result from the dataset BSD200-test shown in Figure 12. Simiarly, RC-Net with BN restores sharp edges, while RC-Net without BN network performs better in terms of PSNR and SSIM.

## C. More Discussions

**The effect of Batch Normalization layer.** Normally, BN layers are introduced into the network for reducing internal covariate shift and then accelerating training speed. However, BN layers are preset as a regulator in our RC-Nets by potentially storing and enhancing high-frequency features. This property helps with restoring images to appear visually close to ground-truth. RC-Nets with BN layer performs remarkably well in the denoising task. In the SR task, RC-Nets with BN provides nice human visual results while RC-Nets without BN achieve excellent quantitative numbers.

**The properties of large-small filter combination.** Indeed, the large-small filter combination introduced in our RC-Nets fundamentally strengthens our model in image restoration. High-frequency features collected by large filters provide major features while details collected by small filters help to regulate the learned feature maps. After removing BN layers, the competitive PSNR and SSIM values demonstrate that our large-small filter combination can enhance RC-Nets' generalization ability both visually and quantitatively.

Figure 9. **Image Denoising:** Visual results of one image from BSD200-test with  $\sigma = 30$  along with PSNR(dB) / SSIM. Compares to state-of-the-art method, our RC-Net well recovers facial features of kids, especially the eyes are almost identical to the ground truth while state-of-the-art methods lose contours and details because of the high noise level.

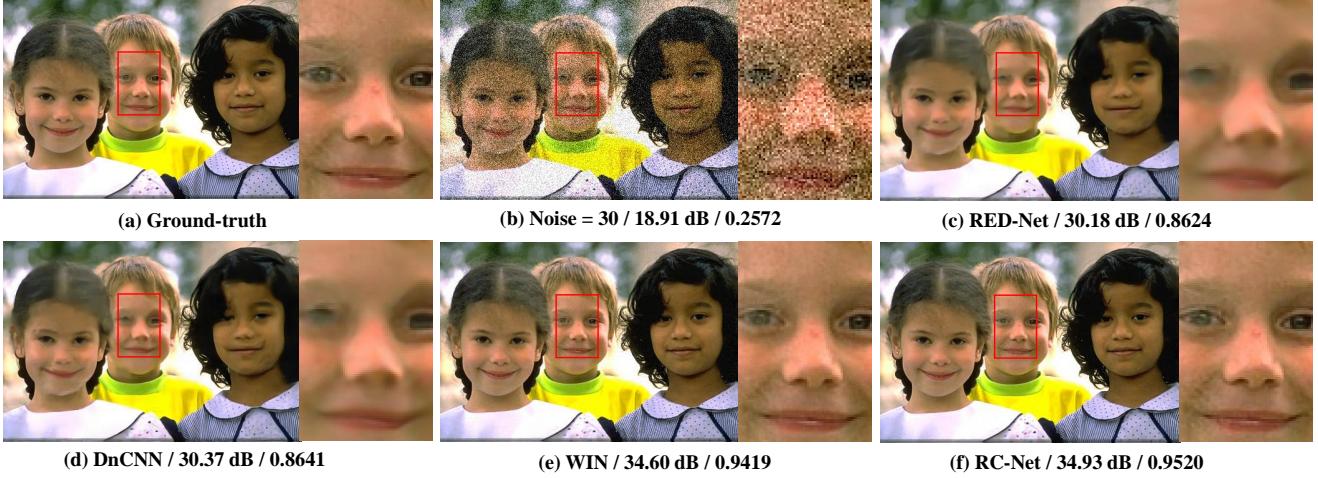


Figure 10. **Image Denoising:** Visual results of one image from BSD200-test with noise level of  $\sigma = 50$  along with PSNR(dB) / SSIM. Horn and ears of rhinoceros are accurately reproduced using our RC-Net method. Bushes and woods in the background are discernible instead of the blurred restoration from other state-of-the-art methods.

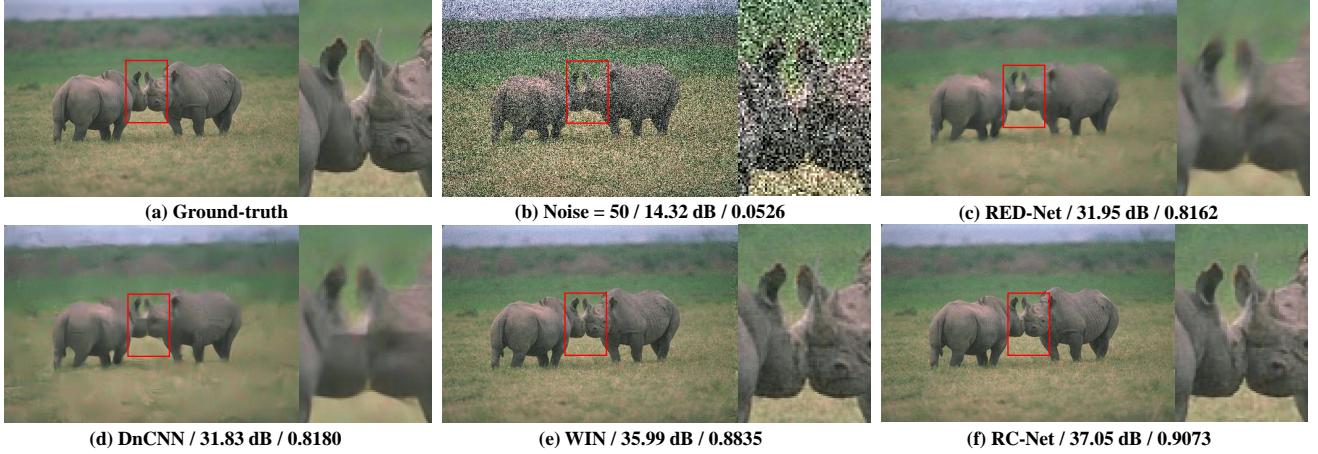
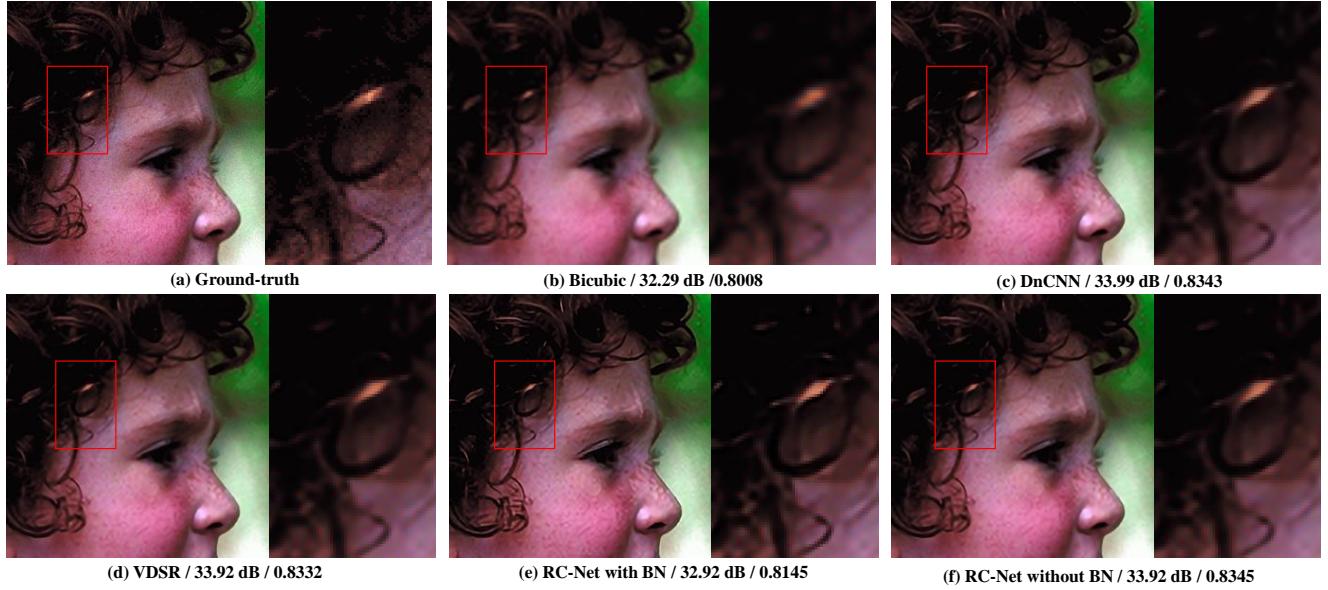


Table 4. **Super-resolution:** Average PSNR / SSIM / Run Time (seconds on GPU) for scale factor  $\times 2$ ,  $\times 3$  and  $\times 4$  on datasets Set5, Set14 and B100. Red color indicates the best performance and blue color indicates the second best performance.

PSNR (dB) / SSIM / Time (s)-GPU						
Dataset	scale	SRCNN	VDSR	RC-Net with BN	RC-Net without BN	
Set5	$\times 2$	36.55 / 0.9542 / 3.37	37.60 / 0.9591 / 0.13	35.48 / 0.9515 / 0.68	37.42 / 0.9586 / 0.46	
	$\times 3$	32.67 / 0.9086 / 3.35	33.54 / 0.9207 / 0.14	27.47 / 0.7851 / 0.68	33.43 / 0.9191 / 0.44	
	$\times 4$	30.37 / 0.8620 / 3.30	30.99 / 0.8800 / 0.15	26.06 / 0.7117 / 0.73	31.01 / 0.8775 / 0.47	
B100	$\times 2$	31.24 / 0.8881 / 3.81	31.89 / 0.8956 / 0.17	30.61 / 0.8880 / 1.09	31.86 / 0.8959 / 0.52	
	$\times 3$	28.40 / 0.7869 / 3.74	28.76 / 0.7973 / 0.17	27.47 / 0.7851 / 1.14	28.76 / 0.7966 / 0.49	
	$\times 4$	26.88 / 0.7114 / 3.32	27.18 / 0.7245 / 0.17	26.06 / 0.7117 / 1.07	27.21 / 0.7242 / 0.53	

**Figure 11. Super-resolution:** Visual results of one image from Set 5 with a scale factor of  $\times 3$  along with PSNR(dB) / SSIM. ROIs show the improved human visual results using RC-Net with BN by restoring sharp and clean contour of the hair. When comparing PSNR and SSIM, RC-Net without BN *outperforms* VDSR.



**Figure 12. Super-resolution:** Visual results of one image from B100 (100 images from BSD200-test dataset) with scale factor of  $\times 3$  along with PSNR(dB) / SSIM. Similarly, RC-Net with BN outperforms other methods in restoring sharp details. RC-Net without BN performs excellently on generating image with improved PSNR and SSIM.

