



neural networks (CNN). The reasons of using CNN are three-fold. First, CNN with very deep architecture [26] is effective in increasing the capacity and flexibility for exploiting image characteristics. Second, considerable advances have been achieved on regularization and learning methods for training CNN, including Rectifier Linear Unit (ReLU) [27], batch normalization [28] and residual learning [29]. These methods can be adopted in CNN to speed up the training process and improve the denoising performance. Third, CNN is well-suited for parallel computation on modern powerful GPU, which can be exploited to improve the run time performance.

We refer to the proposed denoising convolutional neural network as DnCNN. Rather than directly outputting the denoised image  $\hat{x}$ , the proposed DnCNN is designed to predict the residual image  $\hat{v}$ , i.e., the difference between the noisy observation and the latent clean image. In other words, the proposed DnCNN implicitly removes the latent clean image with the operations in the hidden layers. The batch normalization technique is further introduced to stabilize and enhance the training performance of DnCNN. It turns out that residual learning and batch normalization can benefit from each other, and their integration is effective in speeding up the training and boosting the denoising performance.

While this paper aims to design a more effective Gaussian denoiser, we observe that when  $v$  is the difference between the ground truth high resolution image and the bicubic upsampling of the low resolution image, the image degradation model for Gaussian denoising can be converted to a single image super-resolution (SISR) problem; analogously, the JPEG image deblocking problem can be modeled by the same image degradation model by taking  $v$  as the difference between the original image and the compressed image. In this sense, SISR and JPEG image deblocking can be treated as two special cases of a “general” image denoising problem, though in SISR and JPEG deblocking the noise  $v$  is much different from AWGN. It is natural to ask whether it is possible to train a single CNN model to handle such general image denoising problem? By analyzing the connection between DnCNN and TNRD [19], we propose to extend DnCNN for handling several general image denoising tasks, including Gaussian denoising, SISR and JPEG image deblocking.

Extensive experiments show that, our DnCNN trained with a certain noise level can yield better Gaussian denoising results than state-of-the-art methods such as BM3D [2], WNNM [15] and TNRD [19]. For Gaussian denoising with unknown noise level (i.e., blind Gaussian denoising), DnCNN with a single model can still outperform BM3D [2] and TNRD [19] trained for a specific noise level. The DnCNN can also obtain promising results when being extended to several general image denoising tasks. Moreover, we show the effectiveness of training only a single DnCNN model for three general image denoising tasks, i.e., blind Gaussian denoising, SISR with multiple upscaling factors, and JPEG deblocking with different quality factors.

The contributions of this work are summarized as follows:

- 1) We propose an end-to-end trainable deep CNN for Gaussian denoising. In contrast to the existing deep

neural network-based methods which directly estimate the latent clean image, the network adopts the residual learning strategy to remove the latent clean image from noisy observation.

- 2) We find that residual learning and batch normalization can greatly benefit the CNN learning as they can not only speed up the training but also boost the denoising performance. For Gaussian denoising with a certain noise level, DnCNN outperforms state-of-the-art methods in terms of both quantitative metrics and visual quality.
- 3) Our DnCNN can be easily extended to handle general image denoising tasks. We can train a single DnCNN model for blind Gaussian denoising, and achieve better performance than the competing methods trained for a specific noise level. Moreover, it is promising to solve three general image denoising tasks, i.e., blind Gaussian denoising, SISR, and JPEG deblocking, with only a single DnCNN model.

The remainder of the paper is organized as follows. Section II provides a brief survey of related work. Section III first presents the proposed DnCNN model, and then extends it to general image denoising. In Section IV, extensive experiments are conducted to evaluate DnCNNs. Finally, several concluding remarks are given in Section V.

## II. RELATED WORK

### A. Deep Neural Networks for Image Denoising

There have been several attempts to handle the denoising problem by deep neural networks. Kingma and Ba [30] proposed to use convolutional neural networks (CNNs) for image denoising and claimed that CNNs have similar or even better representation power than the MRF model. In [31], the multi-layer perceptron (MLP) was successfully applied for image denoising. In [32], stacked sparse denoising auto-encoders were adopted to handle Gaussian noise removal and achieved comparable results to K-SVD [6]. In [19], a trainable nonlinear reaction diffusion (TNRD) model was proposed and it can be expressed as a feed-forward deep network by unfolding a fixed number of gradient descent inference steps. Among the above deep neural networks based methods, MLP and TNRD can achieve promising performance and are able to compete with BM3D. However, for MLP [31] and TNRD [19], a specific model is trained for a certain noise level. To the best of our knowledge, it remains uninvestigated to develop CNN for general image denoising.

### B. Residual Learning and Batch Normalization

Recently, driven by the easy access to large-scale dataset and the advances in deep learning methods, the convolutional neural networks have shown great success in handling various vision tasks. The representative achievements in training CNN models include Rectified Linear Unit (ReLU) [27], tradeoff between depth and width [26], [33], parameter initialization [34], gradient-based optimization algorithms [35]–[37], batch normalization [28] and residual learning [29]. Other factors, such as the efficient training implementation on modern







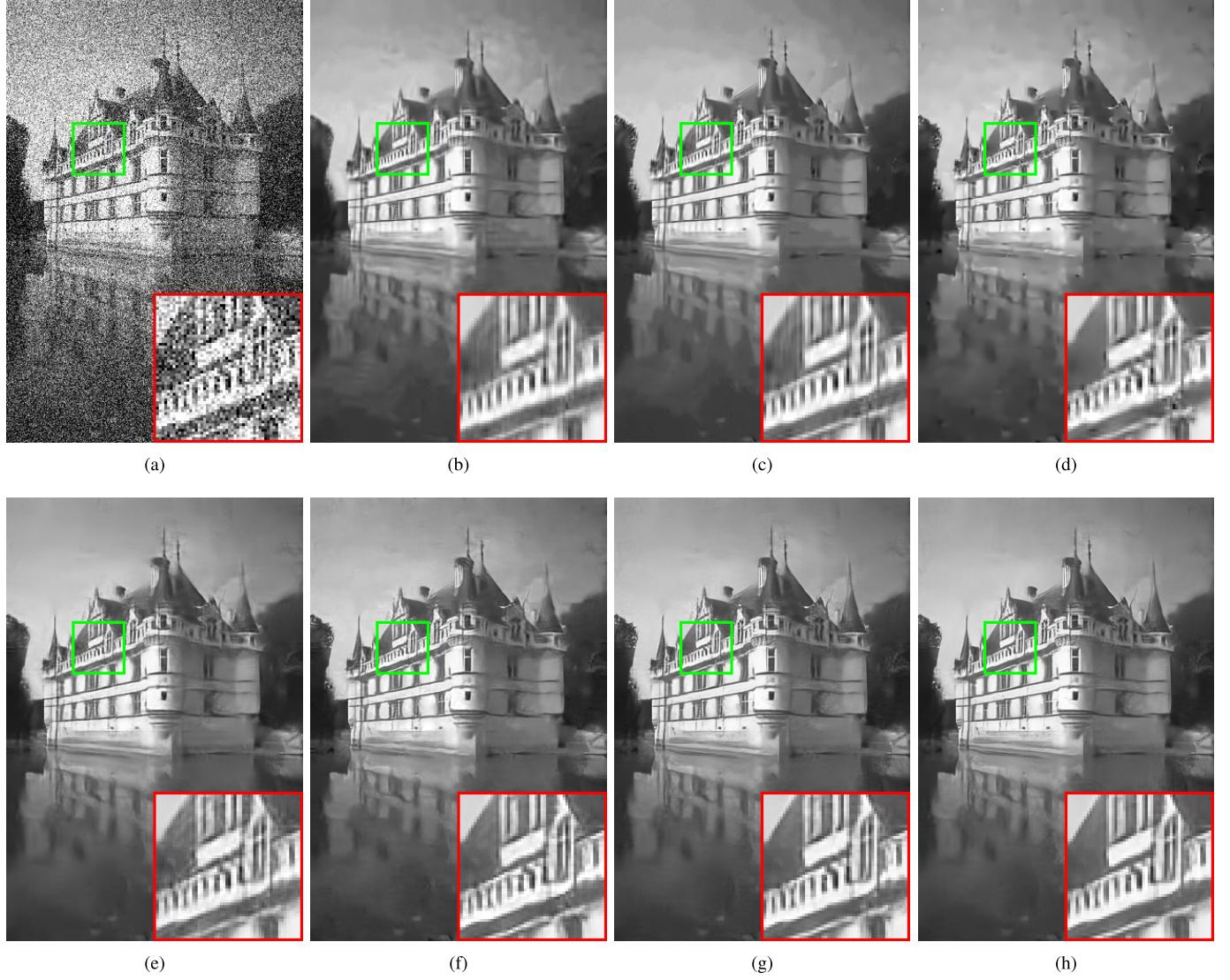


Fig. 4. Denoising results of one image from BSD68 with noise level 50. (a) Noisy / 14.76dB. (b) BM3D / 26.21dB. (c) WNNM / 26.51dB. (d) EPLL / 26.36dB. (e) MLP / 26.54dB. (f) TNRD / 26.59dB. (g) DnCNN-S / 26.90dB. (h) DnCNN-B / 26.92dB.

point  $\mathbf{y}$ , which is given by

$$\mathbf{x}_1 = \mathbf{y} - \alpha \lambda \sum_{k=1}^K (\bar{\mathbf{f}}_k * \phi_k(\mathbf{f}_k * \mathbf{y})) - \alpha \left. \frac{\partial \Psi(\mathbf{z})}{\partial \mathbf{z}} \right|_{\mathbf{z}=\mathbf{0}}, \quad (3)$$

where  $\bar{\mathbf{f}}_k$  is the adjoint filter of  $\mathbf{f}_k$  (i.e.,  $\bar{\mathbf{f}}_k$  is obtained by rotating 180 degrees the filter  $\mathbf{f}_k$ ),  $\alpha$  corresponds to the stepsize and  $\rho'_k(\cdot) = \phi_k(\cdot)$ . For Gaussian denoising, we have  $\left. \frac{\partial \Psi(\mathbf{z})}{\partial \mathbf{z}} \right|_{\mathbf{z}=\mathbf{0}} = \mathbf{0}$ , and Eqn. (3) is equivalent to the following expression

$$\mathbf{v}_1 = \mathbf{y} - \mathbf{x}_1 = \alpha \lambda \sum_{k=1}^K (\bar{\mathbf{f}}_k * \phi_k(\mathbf{f}_k * \mathbf{y})), \quad (4)$$

where  $\mathbf{v}_1$  is the estimated residual of  $\mathbf{x}$  with respect to  $\mathbf{y}$ .

Since the influence function  $\phi_k(\cdot)$  can be regarded as pointwise nonlinearity applied to convolution feature maps, Eqn. (4) actually is a two-layer feed-forward CNN. As can be seen from Fig. 1, the proposed CNN architecture further generalizes one-stage TNRD from three aspects: (i) replacing the influence

function with ReLU to ease CNN training; (ii) increasing the CNN depth to improve the capacity in modeling image characteristics; (iii) incorporating with batch normalization to boost the performance. The connection with one-stage TNRD provides insights in explaining the use of residual learning for CNN-based image restoration. Most of the parameters in Eqn. (4) are derived from the analysis prior term of Eqn. (2). In this sense, most of the parameters in DnCNN are representing the image priors.

It is interesting to point out that, even the noise is not Gaussian distributed (or the noise level of Gaussian is unknown), we still can utilize Eqn. (3) to obtain  $\mathbf{v}_1$  if we have

$$\left. \frac{\partial \Psi(\mathbf{z})}{\partial \mathbf{z}} \right|_{\mathbf{z}=\mathbf{0}} = \mathbf{0}. \quad (5)$$

Note that Eqn. (5) holds for many types of noise distributions, e.g., generalized Gaussian distribution. It is natural to assume that it also holds for the noise caused by SISR and JPEG compression. It is possible to train a single CNN model



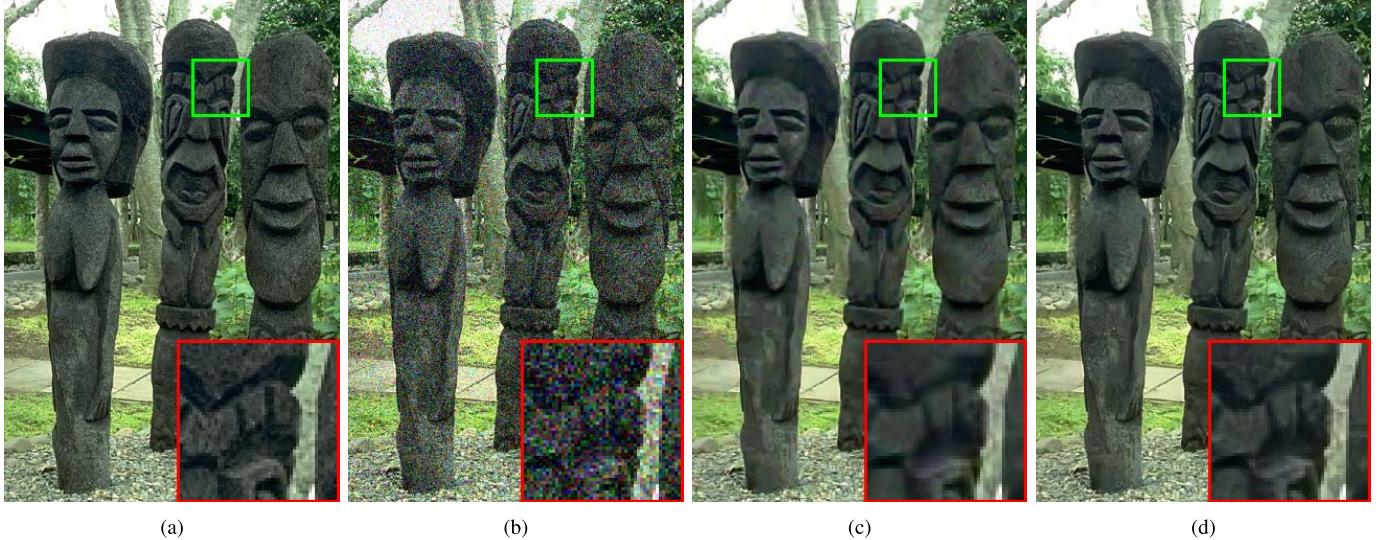


Fig. 6. Color image denoising results of one image from the DSD68 dataset with noise level 35. (a) Ground-truth. (b) Noisy / 17.25dB. (c) CBM3D / 25.93dB. (d) CDnCNN-B / 26.58dB.

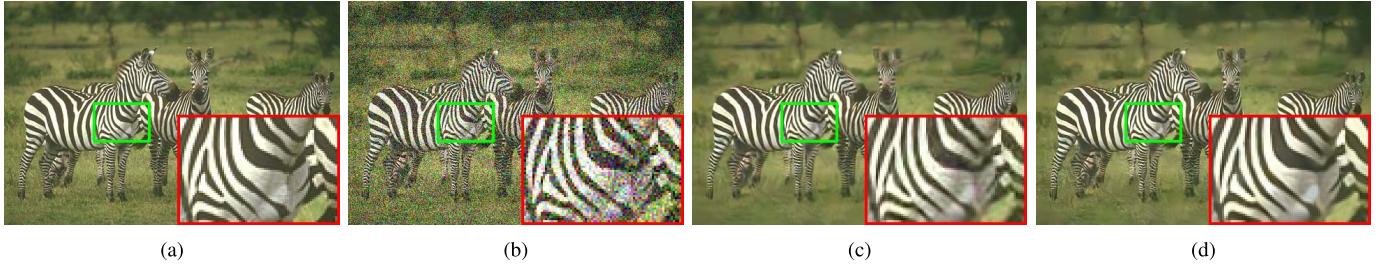


Fig. 7. Color image denoising results of one image from the DSD68 dataset with noise level 45. (a) Ground-truth. (b) Noisy / 15.07dB. (c) CBM3D / 26.97dB. (d) CDnCNN-B / 27.87dB.

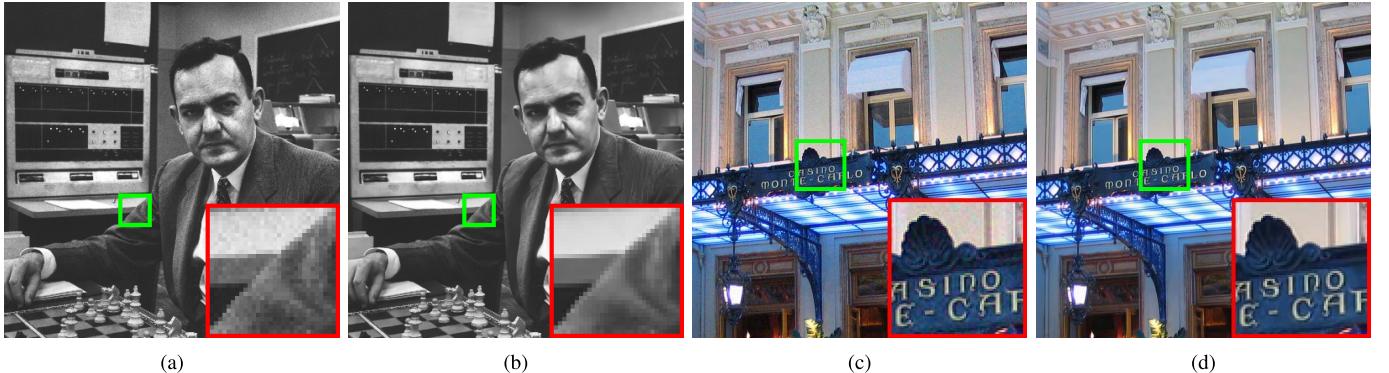


Fig. 8. Gaussian denoising results of two real images by DnCNN-B and CDnCNN-B models, respectively. (a) Noisy. (b) Result by DnCNN-B. (c) Noisy. (d) Result by CDnCNN-B.

DnCNN would also perform robustly in predicting residual image by gradually removing the latent clean image in the hidden layers.

#### E. Extension to General Image Denoising

Like MLP, CSF and TNRD, all of the existing discriminative Gaussian denoising methods train a specific model for a fixed noise level [19], [31]. When applied to Gaussian

denoising with unknown noise, one common way is to first estimate the noise level, and then use the model trained with the corresponding noise level. This makes the denoising results affected by the accuracy of noise estimation. In addition, those methods cannot be applied to the cases with non-Gaussian noise distribution, e.g., SISR and JPEG deblocking.

Our analyses in Section III-D have shown the potential of DnCNN in general image denoising. To demonstrate it, we first extend our DnCNN for Gaussian denoising with unknown



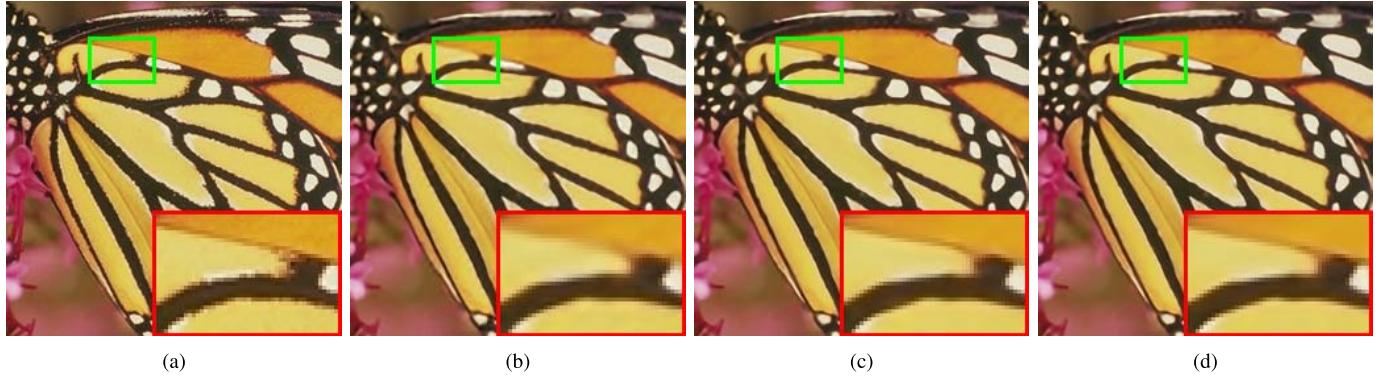


Fig. 10. Single image super-resolution results of “butterfly” from Set5 dataset with upscaling factor 3. (a) Ground-truth. (b) TNRD / 28.91dB. (c) VDSR / 29.95dB. (d) DnCNN-3 / 30.02dB.

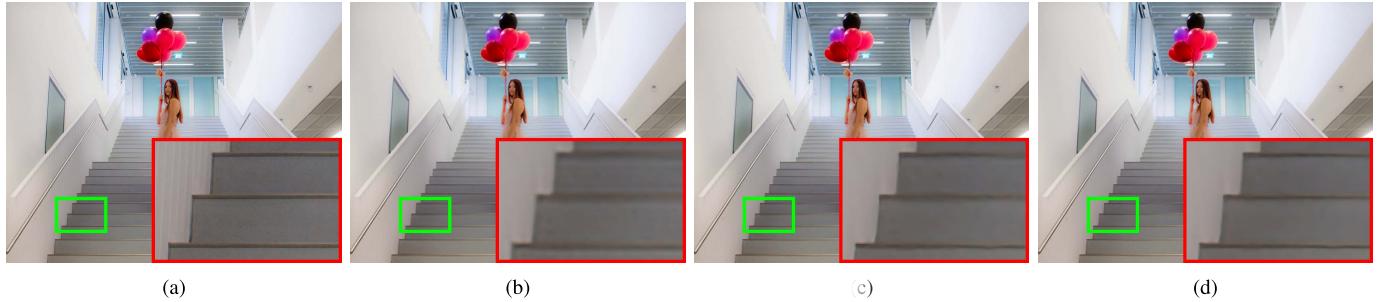


Fig. 11. Single image super-resolution results of one image from Urban100 dataset with upscaling factor 4. (a) Ground-truth. (b) TNRD / 32.00dB. (c) VDSR / 32.58dB. (d) DnCNN-3 / 32.73dB.

DnCNN-S and DnCNN-B can achieve the best PSNR results than the competing methods. Compared to the benchmark BM3D, the methods MLP and TNRD have a notable PSNR gain of about 0.35dB. According to [41] and [45], few methods can outperform BM3D by more than 0.3dB on average. In contrast, our DnCNN-S model outperforms BM3D by 0.6dB on all the three noise levels. Particularly, even with a single model without known noise level, our DnCNN-B can still outperform the competing methods which is trained for the known specific noise level. It should be noted that both DnCNN-S and DnCNN-B outperform BM3D by about 0.6dB when  $\sigma = 50$ , which is very close to the estimated PSNR bound over BM3D (0.7dB) in [45].

Table III lists the PSNR results of different methods on the 12 test images in Fig. 3. The best PSNR result for each image with each noise level is highlighted in bold. It can be seen that the proposed DnCNN-S yields the highest PSNR on most of the images. Specifically, DnCNN-S outperforms the competing methods by 0.2dB to 0.6dB on most of the images and fails to achieve the best results on only two images “House” and “Barbara”, which are dominated by repetitive structures. This result is consistent with the findings in [46]: non-local similarity based methods are usually better on images with regular and repetitive structures whereas discriminative training based methods generally produce better results on images with irregular textures. Actually, this is intuitively reasonable because images with regular and repetitive structures meet well with the non-local similarity prior; conversely, images

with irregular textures would weaken the advantages of such specific prior, thus leading to poor results.

Figs. 4-5 illustrate the visual results of different methods. It can be seen that BM3D, WNNM, EPLL and MLP tend to produce over-smooth edges and textures. While preserving sharp edges and fine details, TNRD is likely to generate artifacts in the smooth region. In contrast, DnCNN-S and DnCNN-B can not only recover sharp edges and fine details but also yield visually pleasant results in the smooth region.

For color image denoising, the visual comparisons between CDnCNN-B and the benchmark CBM3D are shown in Figs. 6-7. One can see that CBM3D generates false color artifacts in some regions whereas CDnCNN-B can recover images with more natural color. In addition, CDnCNN-B can generate images with more details and sharper edges than CBM3D.

Fig. 8 shows two real image denoising examples by our DnCNN-B and CDnCNN-B models. Note that our DnCNN-B is trained for blind Gaussian denoising. However, as discussed in Sec. III-D, DnCNN-B can work well on real noisy images when the noise is additive white Gaussian-like or roughly satisfies the assumption in Eqn. (5). From Fig. 8, one can see that our models can recover visually pleasant results while preserving image details. The results indicate the feasibility of deploying our method for some practical image denoising applications.

Fig. 9 shows the average PSNR improvement over BM3D/CBM3D with respect to different noise levels

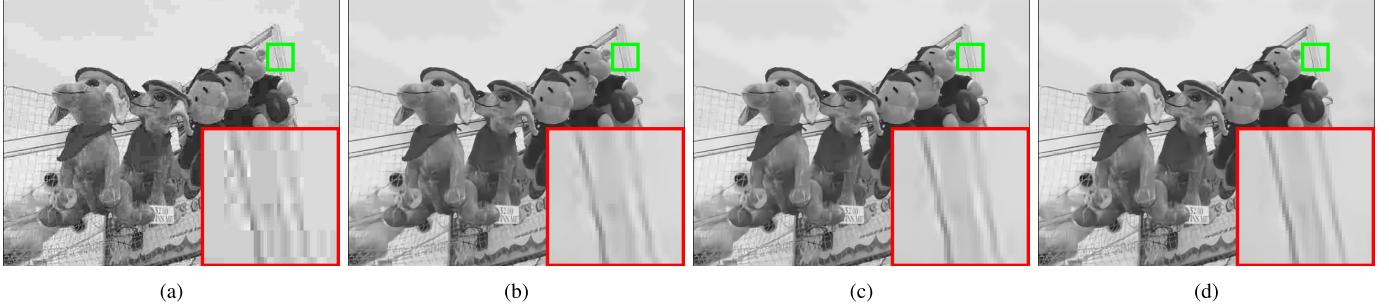


Fig. 12. JPEG image deblocking results of “Carnivaldolls” from LIVE1 dataset with quality factor 10. (a) JPEG / 28.10dB. (b) AR-CNN / 28.85dB. (c) TNRD/ 29.54dB. (d) DnCNN-3 / 29.70dB.

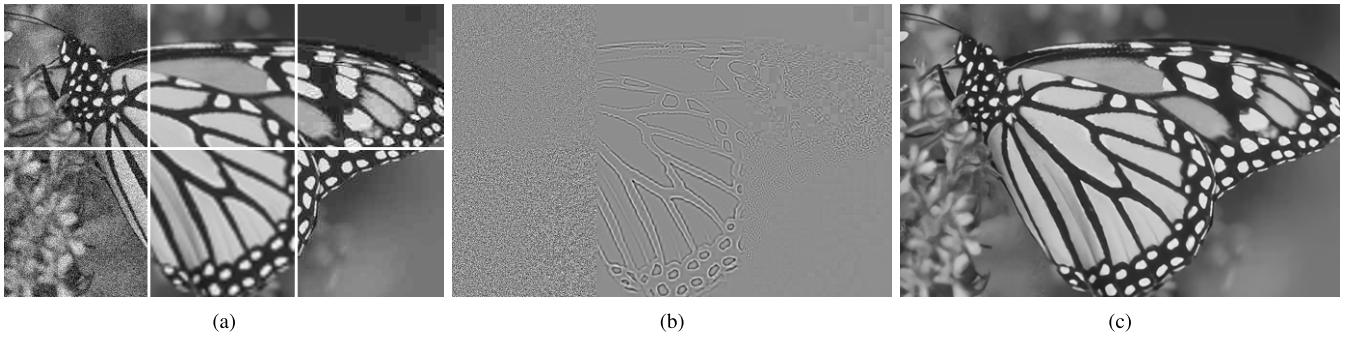


Fig. 13. An example to show the capacity of our proposed model for three different tasks. The input image is composed by noisy images with noise level 15 (upper left) and 25 (lower left), bicubically interpolated low-resolution images with upscaling factor 2 (upper middle) and 3 (lower middle), JPEG images with quality factor 10 (upper right) and 30 (lower right). Note that the white lines in the input image are just used for distinguishing the six regions, and the residual image is normalized into the range of [0, 1] for visualization. Even the input image is corrupted with different distortions in different regions, the restored image looks natural and does not have obvious artifacts. (a) Input image. (b) Output residual image. (c) Restored image.

by DnCNN-B/CDnCNN-B model. It can be seen that our DnCNN-B/CDnCNN-B models consistently outperform BM3D/CBM3D by a large margin on a wide range of noise levels. This experimental result demonstrates the feasibility of training a single DnCNN-B model for handling blind Gaussian denoising within a wide range of noise levels.

#### D. Run Time

In addition to visual quality, another important aspect for an image restoration method is the testing speed. Table IV shows the run times of different methods for denoising images of sizes  $256 \times 256$ ,  $512 \times 512$  and  $1024 \times 1024$  with noise level 25. Since CSF, TNRD and our DnCNN methods are well-suited for parallel computation on GPU, we also give the corresponding run times on GPU. We use the Nvidia cuDNN-v5 deep learning library to accelerate the GPU computation of the proposed DnCNN. As in [19], we do not count the memory transfer time between CPU and GPU. It can be seen that the proposed DnCNN can have a relatively high speed on CPU and it is faster than two discriminative models, MLP and CSF. Though it is slower than BM3D and TNRD, by taking the image quality improvement into consideration, our DnCNN is still very competitive in CPU implementation. For the GPU time, the proposed DnCNN achieves very appealing computational efficiency, e.g., it can denoise an image of size  $512 \times 512$  in 60ms with unknown noise level, which is a distinct advantage over TNRD.

#### E. Experiments on Learning a Single Model for Three General Image Denoising Tasks

In order to further show the capacity of the proposed DnCNN model, a single DnCNN-3 model is trained for three general image denoising tasks, including blind Gaussian denoising, SISR and JPEG image deblocking. To the best of our knowledge, none of the existing methods have been reported for handling these three tasks with only a single model. Therefore, for each task, we compare DnCNN-3 with the specific state-of-the-art methods. In the following, we describe the compared methods and the test dataset for each task:

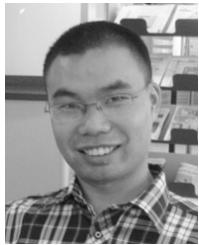
- For Gaussian denoising, we use the state-of-the-art BM3D and TNRD for comparison. The BSD68 dataset are used for testing the performance. For BM3D and TNRD, we assume that the noise level is known.
- For SISR, we consider two state-of-the-art methods, i.e., TNRD and VDSR [42]. TNRD trained a specific model for each upscaling factor while VDSR [42] trained a single model for all the three upscaling factors (i.e., 2, 3 and 4). We adopt the four testing datasets (i.e., Set5 and Set14, BSD100 and Urban100 [47]) used in [42].
- For JPEG image deblocking, our DnCNN-3 is compared with two state-of-the-art methods, i.e., AR-CNN [48] and TNRD [19]. The AR-CNN method trained four specific models for the JPEG quality factors 10, 20, 30 and 40, respectively. For TNRD, three models for JPEG quality







**Wangmeng Zuo** (M'09–SM'14) received the Ph.D. degree in computer application technology from the Harbin Institute of Technology, Harbin, China, in 2007. From 2004 to 2006, he was a Research Assistant with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. From 2009 to 2010, he was a Visiting Professor with Microsoft Research Asia. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology. He has published over 60 papers in top-tier academic journals and conferences. His current research interests include image enhancement and restoration, weakly supervised learning, visual tracking, and image classification. He has served as a Tutorial Organizer in ECCV 2016, an Associate Editor of the *IET Biometrics*, and the Guest Editor of *Neurocomputing, Pattern Recognition*, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.



**Yunjin Chen** received the B.Sc. degree in applied physics from the Nanjing University of Aeronautics and Astronautics, China, in 2007, the M.Sc. degree in optical engineering from the National University of Defense Technology, China, in 2010, and the Ph.D. degree in computer science from the Graz University of Technology, Austria, in 2015. Since 2015, he serves as a Scientific Researcher with the Military of China. His current research interests include learning image prior model for low-level computer vision problems and convex optimization.



**Deyu Meng** received the B.Sc., M.Sc., and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2001, 2004, and 2008, respectively. He is currently a Professor with the Institute for Information and System Sciences, School of Mathematics and Statistics, Xi'an Jiaotong University. From 2012 to 2014, he took his two-year sabbatical leave in Carnegie Mellon University. His current research interests include self-paced learning, noise modeling, and tensor sparsity.



**Lei Zhang** (M'04–SM'14) received the B.Sc. degree from the Shenyang Institute of Aeronautical Engineering, Shenyang, China, in 1995, and the M.Sc. and Ph.D. degrees in control theory and engineering from Northwestern Polytechnical University, Xi'an, China, in 1998 and 2001, respectively. From 2001 to 2002, he was a Research Associate with the Department of Computing, The Hong Kong Polytechnic University. From 2003 to 2006 he was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, McMaster University, Canada. In 2006, he joined the Department of Computing, The Hong Kong Polytechnic University, as an Assistant Professor, where he has been a Full Professor, since 2015. He has published over 200 papers in his research areas. His research interests include computer vision, pattern recognition, image and video processing, and biometrics. During 2016, his publications have been cited over 20,000 times in the literature. He is an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, the *SIAM Journal of Imaging Sciences*, and *Image and Vision Computing*. He is a Web of Science Highly Cited Researcher selected by Thomson Reuters.