Residual Dense Generative Adversarial Network for Single Image Super-Resolution

Jiahao Meng
School of Science
Beijing University of Posts and Telecommunications
Beijing, P.R. China
Email: jiahaomeng@bupt.edu.cn

Zekuan Yu
Department of Biomedical Engineering
College of Engineering
Peking University
Beijing, P.R. China
Email: yuzekuan518@163.com

Tianping Shuai
School of Science
Beijing University of Posts and Telecommunications
Beijing, P.R. China
Email: tpshuai@bupt.edu.cn

***Abstract*—Model-based very deep Convolutional Neural Networks (CNN) have achieved great success in Single Image Super-Resolution (SISR) work. However, most of the super-resolution models based on deep convolution networks can not fully utilize the hierarchical features of the original low-resolution images. In order to improve the quality of the high-frequency details of the reconstructed super-resolution image, we proposes a super-resolution method for Residual Dense Generative Adversarial Networks (RDGAN). We use the Generative Adversarial Networks (GAN) as our main model structure and the residual-dense block as the basic building blocks of the generator, which makes the network pay more attention to the extraction of low-resolution image hierarchical features. Then, we fully exploit the hierarchical features from all the convolutional layers. Finally, we use perceptual loss as our loss function to get finer texture details and more realistic photo effects. Experiments show that our method can achieve significant improvement in the quality of high-frequency detail reconstruction at high magnification.**

Keywords–CNN; Single Image Super-Resolution; Generative Adversarial Networks.

I. INTRODUCTION

The task of estimating a High-Resolution (HR) image from its Low-Resolution (LR) counterpart is called Single Image Super-Resolution (SISR) [1], which has received significant attention and progress in recent years. Super-Resolution (SR) has direct applications in computer vision, such as image/video enhancement, medical image processing [2][3], face recognition [4] and image generation [5].

Image SR is an ill-posed problem. The ill-posed character of the under-constrained SR problem is especially pronounced for high upscaling factors. Recently, a large number of SISR methods have been proposed to solve this underdetermined problem, including interpolation-based [3][6], reconstruction methods [7], and learning-based methods [8][9]. Most CNN-based methods [10][11] attempt to minimize pixel-wise the Mean Square Error (MSE) between the ground truth image and the reconstructed HR image. This strategy calculates the pixel-wise image difference and maximizes the Peak Signal-to-Noise Ratio (PSNR), which is a common measurement for evaluating the SR algorithm. In these cases, the high-frequency details of some sharp edges and textures in the SR image are still blurred and smooth in appearance, which is significantly different from the ground truth image.

In order to solve these drawbacks, Ledig *et al.* [12] proposed a GAN-based network. This enhances the invariance of the pixel field change. However, for a very deep network, only using Residual Networks (ResNets) and jump connections can not fully utilize the LR image information. Inspired by Zhang *et al.* [13], we use Residual Dense Block (RDB) as the basic component of our generator and we use Local Residual Learning (LRL) in order to make full use of the hierarchical features of LR. In this paper, we proposes a deep learning SISR method, which uses enhance Residual Dense Generative Adversarial Network (RDGAN) to improve the reconstruction quality of high-frequency edges and textures in the SR images. At the same time, we minimize the perceptual loss so that the generated images have photo realistic textures.

The rest of this paper is organized as follows. Section II addresses the related works in the literature. Section III describes the method. Section IV describes the experiments. We conclude the paper in Section V.

II. RELATED WORKS

In order to solve the Single Image Super-Resolution problem, early algorithms [14][15] have been mainly based on sampling interpolation techniques, but these methods show considerable limitations in predicting the texture details of the image.

Recently, the CNN-based [16] approaches have shown excellent performance. Dong *et al.* proposed Super-Resolution Convolutional Neural Networks (SRCNN) [10], which train a 3 layer deep fully convolutional network end-to-end to achieve excellent SR performance. Kim *et al.* [17] used a very deep CNN network (20 weight layers) to achieve better performance and visual effects. In particular, they showed skip-connection and recursive convolution alleviate the burden of carrying identity information in the super-resolution network. In [18], Lim *et al.* propose the Enhanced Deep Residual Networks (EDSR) with better performance than SRResNet. Johnson *et al.* [19] proposed perceptual loss functions based on high-level features extracted from pretrained networks, which can reconstruct finer details compared to the per-pixel loss. Recently, Generative adversarial networks [20] have shown excellent results in many computer vision problems including SISR. Ledig *et al.* [12] used GAN to get photo-realistic natural images, which have better visually implausible performance than any other state-of-the-art methods. The authors propose a perceptual loss function constructed by both an adversarial loss and a perceptual content loss based on high-level features

 ISBN: 978-1-61208-716-0

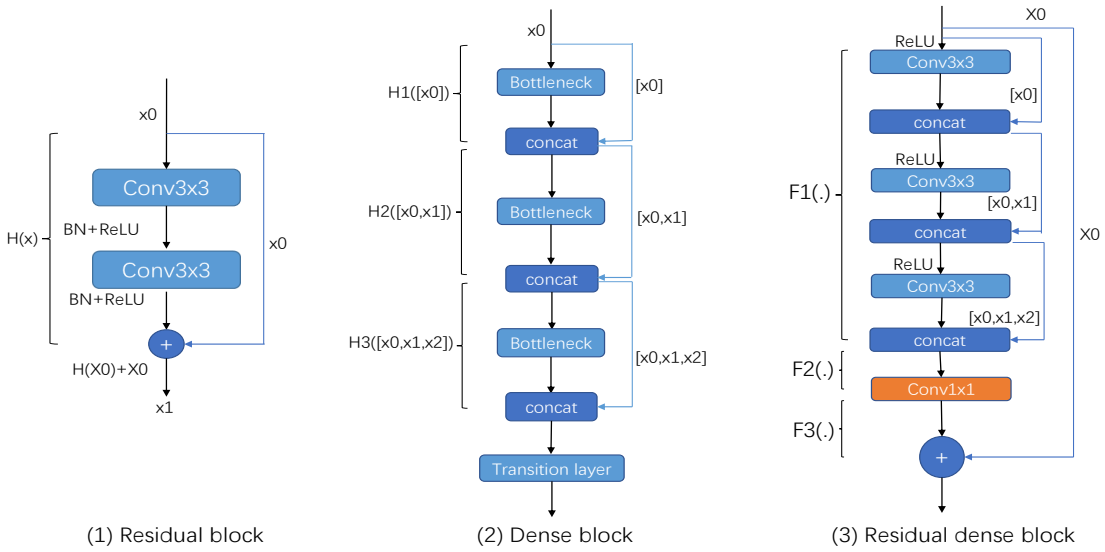

Figure 1. Examples of residual block, dense block and residual dense block. ”+” means element-wise summation operation. ”concat” means concatenation operation. ”bottleneck” in dense block which produces k feature-maps.

extracted from pre-trained Visual Geometry Group (VGG) networks.

III. METHOD

A. Network selection

Many research works [21][22] show networks that perform satisfactory in image generation, classification, and feature extraction and they are equally superior in image super-resolution. Among them, GANs, ResNets and DenseNets are successfully applied to image super-resolution tasks [23][24].

GANs: Following Goodfellow *et al.*, we define a discriminator network D_{θ_D} that is optimized in an alternating manner with the generator network G_{θ_G} to solve the adversarial minimum-maximum problem:

$$\min_{\theta_G}\max_{\theta_D}\mathbb{E}_{I^{HR}\sim p_{train}(I^{HR})}[logD_{\theta_D}(I^{HR})]+ \mathbb{E}_{I^{LR}\sim p_G(I^{LR})}[log(1-D_{\theta_D}(G_{\theta_G}(I^{LR}))] \quad (1)$$

where θ_G and θ_D represent the parameters of the generator and the discriminator, I^{HR} and I^{LR} represent the ground truth image and the low resolution image. The general idea is that it allows people to train a generative model G, the purpose of which is to fool the discriminator D that can distinguish between the real image and the generated image. With this approach, our generator can learn to create solutions that are highly similar to real images. This encourages perceptually superior solutions residing in the subspace, the manifold, of natural images.

ResNets: The main idea is to use a residual learning framework to ease the training of very deep networks. Let a single image x_0 go through a L-layer convolutional network. Each layer corresponds to a non-linear transformation $H_\ell(\cdot)$, where ℓ represents the index of the layer. Let x_ℓ be the output of ℓ-th layer. The traditional convolutional network generally uses the output of the ℓth layer as the input of the (ℓ+1)-th layer, which can be expressed as: $x_{\ell+1} = H_{\ell+1}(x_\ell)$. Unlike traditional CNNs, ResNets implements a residual block that sums up the identity mapping of the input to output of a layer, where the output can be depicted as: $x_{\ell+1} = H_{\ell+1}(x_\ell) + x_\ell$. This process eases the convergence during training. The structure of ResNets is shown in Figure 1(1).

DenseNets: The obvious difference between DenseNets and ResNets is that ResNets is a summation, while DenseNets is a concatenation. DenseNets enhances the transmission efficiency of information and gradients in the network. Each layer can directly get the gradient from the loss function and directly get the input signal, so that it can train deeper networks. The dense connection was introduced among memory blocks and dense blocks. Consequently, the feature maps of all previous layers are treated as separate inputs by connecting them to a single tensor $[x_0, x_1, ..., x_\ell]$, while their own feature maps are passed as input to all subsequent layers. Layer ℓ+1 receives the feature maps of all previous layers and can be expressed as: $x_{\ell+1} = H_{\ell+1}([x_0, x_1, ..., x_\ell])$. Figure 1(2) shows an example of dense block construction.

RDBs mainly integrates the residual blocks and the dense blocks. The structure difference is obvious in Figure 1(3). Let F_{l-1} and F_l represent the input and output of the ℓ-th RDB, respectively, and they all have G0 feature maps. In our experiment, we set G0 to 128. In the ℓ-th RDB, the output of the c-th convolutional layer can be formulated as:

$$F_{\ell,c} = \sigma(W_{\ell,c}[F_{\ell-1}, F_{\ell,1}, ..., F_{\ell,c-1}]) \quad (2)$$

where σ represents the Rectified Linear Unit (ReLU) activation function. $W_{\ell,c}$ is the weight of the c-th convolutional layer. For convenience, we ignore the bias term. In our work, we set the number of convolution layers in each RDB to 9. We use these layers to extract continuous memory. Then, connect all the feature maps extracted earlier. Inspired by MemNet [25], we introduce a 1×1 convolutional layer to adaptively control

 ISBN: 978-1-61208-716-0

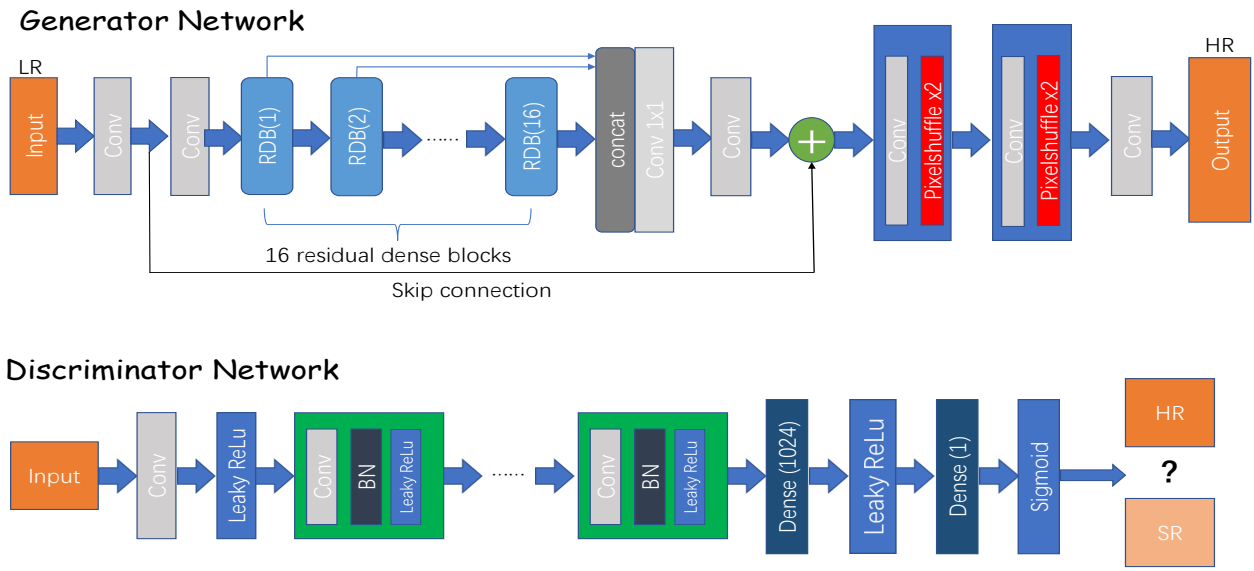

Figure 2. Architecture of Generator and Discriminator Network

the output information. Finally, the number of feature maps becomes G0. This step can be expressed as:

$$F_{\ell,m} = H_{LFF}([F_{\ell-1}, F_{\ell,1}, ..., F_{\ell,9}]) \quad (3)$$

where H_{LFF} represents a function of the 1*1 convolution layer. Finally, use the principle of residuals to achieve local residual learning. The output of the ℓ-th RDB can be expressed as:

$$F_\ell = F_{\ell-1} + F_{\ell,m} \quad (4)$$

B. Basic network architecture

The entire network structure of the generator is presented in Table I. We set up sixteen RDBs and each RDB is set as described above. In order to prevent the loss of LR images detail, we removed the pooling layer and the BN layer, then connected the outputs of all RDBs, with 1×1 convolution kernels to fuse feature maps and add residuals connect to retain more details. More details can be seen in Figure 2. This model can accept the input of LR images of any size, obtain the SR image of a given scaling factor α through the whole generator, and upgrade the image quality through continuous optimization of the generator and discriminator.

C. Loss function

The loss function we use is the same as Ledig *et al.* [12], combining pixel-wise loss and vgg19 loss [19] based on the high-level features extracted from the pre-trained 19 layer VGG networks. Given the high resolution ground truth image I^{HR}, the corresponding low resolution image I^{LR} and the image I^{SR} generated by our network, the loss function can be defined as follows:

$$L_{percep}(I^{HR}, I^{SR}) = \lambda_M \times L_M(I^{HR}, I^{SR}) + \lambda_V \times L_V(I^{HR}, I^{SR}) \quad (5)$$

where $L_M(I^{HR}, I^{SR})$ is pixel-wise loss and $L_V(I^{HR}, I^{SR})$ is vgg19 loss. λ_M and λ_V are scaling hyperparameters. In our work, we set λ_M to 1 and λ_V to 0.006.

TABLE I. ARCHITECTURE DETAILS FOR 4× RDGAN GENERATOR. NOTE THAT EACH "CONV" LAYER SHOWN IN THE TABLE CORRESPONDS THE SEQUENCE RELU-CONV.

Layers	Output size	Residual DenseNet	Feature-maps
Convolution	W×H	3×3 conv	64
RDB(1-16)	W×H	Bottleneck × 9	64
Concat	W×H	Connect	1024(64×16)
Convolution	W×H	1×1 conv	64
Convolution	W×H	3×3 conv	64
Summation	W×H	3×3 conv	64
Convolution	W×H	3×3 conv	256
Upscale	2W×2H	PixelShuffle	64
Convolution	2W×2H	3×3 conv	256
Upscale	4W×4H	PixelShuffle	64
Convolution	4W×4H	3×3 conv	3

pixel-wise loss: It is the Euclidean distance between the generated image I^{SR} and the ground truth image I^{HR}. Pixel-wise loss is defined as follows:

$$L_M(I^{HR}, I^{SR}) = \frac{1}{SWH}\|I^{HR} - I^{SR}\|^2 \quad (6)$$

where SWH is the size of the target image. This loss is added to achieve smoother textures from the ground truth image.

vgg19 loss: It is the Euclidean distance between the feature maps generated by the loss network. When given the pre-training network ϕ and a series of convolutional layers C and the feature map of each convolutional layer on C is $S_i \times W_i \times H_i$, we can define vgg19 loss as follows:

$$L_V(I^{HR}, I^{SR}) = \sum_{i\in C} \frac{1}{S_i \times W_i \times H_i}\|\phi_i(HR) - \phi_i(SR)\| \quad (7)$$

where $S_i \times W_i \times H_i$ represent the size of the respective feature map in the VGG networks.

adversarial loss: In addition to the perceptual loss described above, we also add the adversarial loss to the perceptual loss. This encourages our network to preserve more textures on

 ISBN: 978-1-61208-716-0

TABLE II. QUANTIFIED PERFORMANCE OF DIFFERENT SUPER-RESOLVED METHODS ON BENCHMARK DATA, WHICH IS MEASURED BY (PSNR [DB], SSIM). [4× UPSCALING].

Set5	Bicubic	Aplus [26]	SRCNN [10]	VDSR [17]	DRCN [27]	SRGAN [12]	**RDGAN**
PSNR	28.42	30.28	30.07	31.35	31.53	29.40	**31.70**
SSIM	0.8104	0.8603	0.8627	0.8838	0.8854	0.8472	**0.8903**
Set14							
PSNR	25.99	27.32	27.18	28.01	28.02	26.02	**28.13**
SSIM	0.7027	0.7491	0.7503	0.7674	0.7670	0.7397	**0.7872**
BSD100							
PSNR	25.96	26.82	26.68	27.27	27.23	25.18	**27.39**
SSIM	0.6675	0.7087	0.7101	0.7251	0.7233	0.6688	**0.7290**
Urban100							
PSNR	23.14	24.32	24.52	25.18	25.14	-	**25.68**
SSIM	0.6577	0.7183	0.7221	0.7524	0.7510	-	**0.7712**

natural images. It optimizes parameters by minimizing the generative loss L_{GAN} defined based on $D_{\theta_D}(G_{\theta_G}(I^{LR}))$, which means the probability of the discriminator that the reconstructed images $G_{\theta_G}(I^{LR})$ is a natural HR image:

$$L_{GAN} = \sum_{n=1}^{N} -logD_{\theta_D}(G_{\theta_G}(I^{LR})) \tag{8}$$

Finally, our loss function can be expressed as:

$$\begin{aligned} L(I^{HR}, I^{SR}) &= L_{percep}(I^{HR}, I^{SR}) + \\ &\quad \lambda_{GAN} \times L_{GAN}(I^{HR}, I^{SR}) \end{aligned} \tag{9}$$

We set λ_{GAN} = 0.001. After doing this, we get the images with more natural textures and more realistic details.

IV. EXPERIMENTS

A. Training Details

The train and validation datasets were sampled from DIV2K datasets [28]. DIV2K datasets were obtained from [29][30]. The train dataset has 800 images and the validation dataset has 100 images. We obtained the LR images by downsampling the HR images using bicubic kernel with downsampling factor r=4. This corresponds to a 16× reduction in image pixels. We test the performance on four standard benchmark datasets: Set5 [31], Set14 [32], BSD100 [33], Urban100 [34].

All the experiments were implemented by means of Python 3.6 and PyTorch [35] on a NVIDIA 1080Ti GPU. For training, we use the Red-Green-Blue (RGB) input patches of size 128×128 from LR images with the corresponding HR patches. Note that we can apply the generator model to images of arbitrary size as it is fully convolutional. We train our model with the ADAM optimizer [36] by setting β_1=0.9. The learning rate was initially set to 0.0001 and decreased by a factor of 10 after 50 epoches. We alternate updates to the generator and discriminator network, which is equivalent to k=1 as used in Goodfellow *et al.* [20]. A mini-batch size of 5 was set during the training. It takes about one days to train RDGAN.

B. Evaluation on benchmark datasets

We train all models with 400 epochs. The training process stopped after no improvements of the loss was observed after 350 epoches. We present the quantitative evaluation results of our RDGAN on public benchmark datasets in Table II. We compare the proposed method with the state-of-the-art methods including Aplus [26], SRCNN [11], SRGAN [12], VDSR [17] and DRCN [27]. For comparison, the SR results are evaluated with PSNR and Structural Similarity (SSIM) [37] on Y channel (i.e., luminance) of transformed YCbCr space. Our RDGAN shows significant improvement compared to other models. We also provide the qualitative results in Figure 3. We can see that the method we propose produces relatively sharper edges, while other models may produce ambiguous results.

V. CONCLUSION

In this work, we proposed a very deep Residual Dense Generative Adversarial Network (RDGAN) for Single Image Super-Resolution, where RDBs are used as basic modules for the generator network. By using the new generator network architecture, we maintain the accuracy of the reconstructed image while maintaining the visual quality of the super-resolution image. In terms of the loss function, we retain the confrontation loss, which makes the generated image retain full detail and more realistic in terms of visual perception. We evaluated our method on a large number of datasets and the results show that our RDGAN can achieve good results in Single Image Super-Resolution.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (NSFC), grant (NO. 11571044, 11471052, 11671052). Tianping Shuai is the corresponding author of this paper.

REFERENCES

[1] D. G. S. B. M. Irani, "Super-resolution from a single image," in Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan, 2009, pp. 349–356.

[2] W. Shi et al., "Cardiac image super-resolution with global correspondence using multi-atlas patchmatch," in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2013, pp. 9–16.

[3] T. M. Lehmann, C. Gonner, and K. Spitzer, "Survey: Interpolation methods in medical image processing," IEEE transactions on medical imaging, vol. 18, no. 11, 1999, pp. 1049–1075.

[4] F. Juefei-Xu and M. Savvides, "Single face image super-resolution via solo dictionary learning," in 2015 IEEE International Conference on Image Processing (ICIP). IEEE, 2015, pp. 2239–2243.

[5] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," arXiv preprint arXiv:1710.10196, 2017.

 ISBN: 978-1-61208-716-0

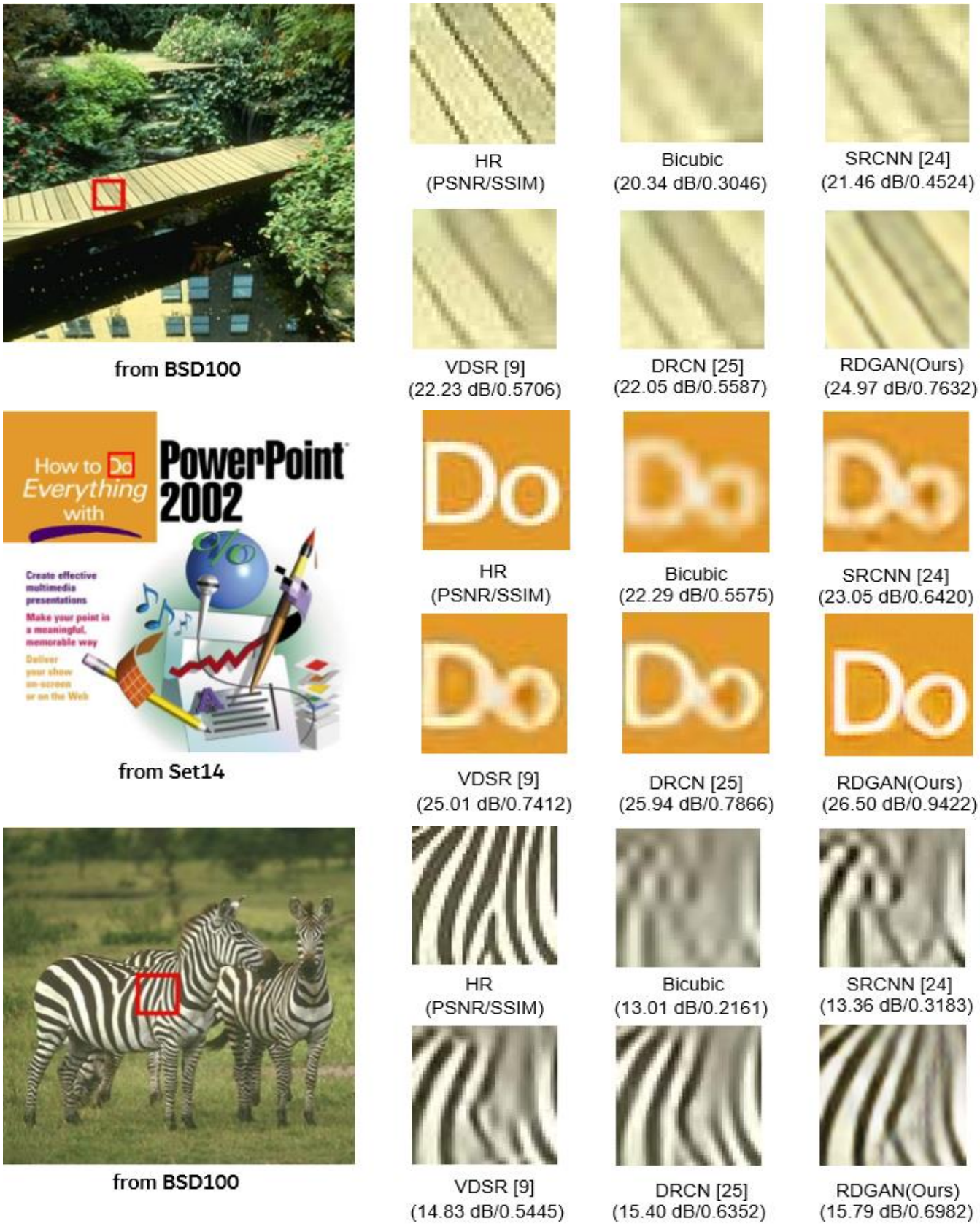

Figure 3. Qualitative comparison of our RDGAN with other methods on 4 super-resolution

 ISBN: 978-1-61208-716-0

[6] R. Keys, “Cubic convolution interpolation for digital image processing,” IEEE transactions on acoustics, speech, and signal processing, vol. 29, no. 6, 1981, pp. 1153–1160.

[7] H. A. Aly and E. Dubois, “Image up-sampling using total-variation regularization with a new observation model,” IEEE Transactions on Image Processing, vol. 14, no. 10, 2005, pp. 1647–1659.

[8] S. Schulter, C. Leistner, and H. Bischof, “Fast and accurate image upscaling with super-resolution forests,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3791–3799.

[9] T. Tong, G. Li, X. Liu, and Q. Gao, “Image super-resolution using dense skip connections,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4799–4807.

[10] 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. Springer, 2014, pp. 184–199.

[11] ——, “Image super-resolution using deep convolutional networks,” IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 2, 2016, pp. 295–307.

[12] C. Ledig et al., “Photo-realistic single image super-resolution using a generative adversarial network,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.

[13] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, “Residual dense network for image super-resolution,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2472–2481.

[14] R. Keys, “Cubic convolution interpolation for digital image processing,” IEEE transactions on acoustics, speech, and signal processing, vol. 29, no. 6, 1981, pp. 1153–1160.

[15] H. Chang, D.-Y. Yeung, and Y. Xiong, “Super-resolution through neighbor embedding,” in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., vol. 1. IEEE, 2004, pp. I–I.

[16] K. Hayat, “Multimedia super-resolution via deep learning: A survey,” Digital Signal Processing, 2018.

[17] 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, 2016, pp. 1646–1654.

[18] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, “Enhanced deep residual networks for single image super-resolution,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 136–144.

[19] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in European Conference on Computer Vision. Springer, 2016, pp. 694–711.

[20] I. Goodfellow et al., “Generative adversarial nets,” in Advances in neural information processing systems, 2014, pp. 2672–2680.

[21] J. Kim, J. K. Lee, and K. M. Lee, “Deeply-recursive convolutional network for image super-resolution,” 2015, pp. 1637–1645.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

[23] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, 2015, pp. 91–99.

[24] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” IEEE transactions on pattern analysis and machine intelligence, vol. 40, no. 4, 2018, pp. 834–848.

[25] Y. Tai, J. Yang, X. Liu, and C. Xu, “Memnet: A persistent memory network for image restoration,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 4539–4547.

[26] R. Timofte, V. De Smet, and L. Van Gool, “A+: Adjusted anchored neighborhood regression for fast super-resolution,” in Asian conference on computer vision. Springer, 2014, pp. 111–126.

[27] J. Kim, J. Kwon Lee, and K. Mu Lee, “Deeply-recursive convolutional network for image super-resolution,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1637–1645.

[28] E. Agustsson and R. Timofte, “Ntire 2017 challenge on single image super-resolution: Dataset and study,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 126–135.

[29] A. Ignatov, R. Timofte et al., “Pirm challenge on perceptual image enhancement on smartphones: report,” in European Conference on Computer Vision (ECCV) Workshops, January 2019.

[30] E. Agustsson and R. Timofte, “Ntire 2017 challenge on single image super-resolution: Dataset and study,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 126–135.

[31] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, “Low-complexity single-image super-resolution based on nonnegative neighbor embedding,” 2012.

[32] R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” in International conference on curves and surfaces. Springer, 2010, pp. 711–730.

[33] D. Martin et al., “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics.” Iccv Vancouver:, 2001.

[34] 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, 2015, pp. 5197–5206.

[35] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017.

[36] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[37] Z. Wang et al., “Image quality assessment: from error visibility to structural similarity,” IEEE transactions on image processing, vol. 13, no. 4, 2004, pp. 600–612.

 ISBN: 978-1-61208-716-0