ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks

Xintao Wang[1], Ke Yu[1], Shixiang Wu[2], Jinjin Gu[3], Yihao Liu[4], Chao Dong[2], Chen Change Loy[5], Yu Qiao[2], Xiaoou Tang[1]

[1]CUHK-SenseTime Joint Lab, The Chinese University of Hong Kong
[2]SIAT-SenseTime Joint Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences [3]The Chinese University of Hong Kong, Shenzhen
[4]University of Chinese Academy of Sciences [5]Nanyang Technological University, Singapore
{wx016,yk017,xtang}@ie.cuhk.edu.hk, {sx.wu,chao.dong,yu.qiao}@siat.ac.cn
liuyihao14@mails.ucas.ac.cn, 115010148@link.cuhk.edu.cn, ccloy@ntu.edu.sg

Abstract. The Super-Resolution Generative Adversarial Network (SRGAN) [1] is a seminal work that is capable of generating realistic textures during single image super-resolution. However, the hallucinated details are often accompanied with unpleasant artifacts. To further enhance the visual quality, we thoroughly study three key components of SRGAN – network architecture, adversarial loss and perceptual loss, and improve each of them to derive an Enhanced SRGAN (ESRGAN). In particular, we introduce the Residual-in-Residual Dense Block (RRDB) without batch normalization as the basic network building unit. Moreover, we borrow the idea from relativistic GAN [2] to let the discriminator predict relative realness instead of the absolute value. Finally, we improve the perceptual loss by using the features before activation, which could provide stronger supervision for brightness consistency and texture recovery. Benefiting from these improvements, the proposed ESRGAN achieves consistently better visual quality with more realistic and natural textures than SRGAN and won the first place in the PIRM2018-SR Challenge[1] [3]. The code is available at https://github.com/xinntao/ESRGAN.

1 Introduction

Single image super-resolution (SISR), as a fundamental low-level vision problem, has attracted increasing attention in the research community and AI companies. SISR aims at recovering a high-resolution (HR) image from a single low-resolution (LR) one. Since the pioneer work of SRCNN proposed by Dong et al. [4], deep convolution neural network (CNN) approaches have brought prosperous development. Various network architecture designs and training strategies have continuously improved the SR performance, especially the Peak Signal-to-Noise Ratio (PSNR) value [5,6,7,1,8,9,10,11,12]. However, these PSNR-oriented approaches tend to output over-smoothed results without sufficient high-frequency details, since the PSNR metric fundamentally disagrees with the subjective evaluation of human observers [1].

[1] We won the first place in region 3 and got the best perceptual index.

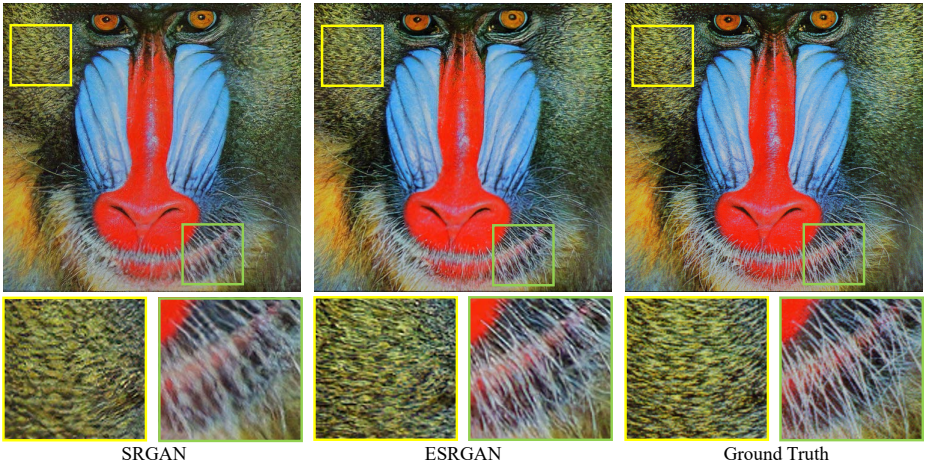
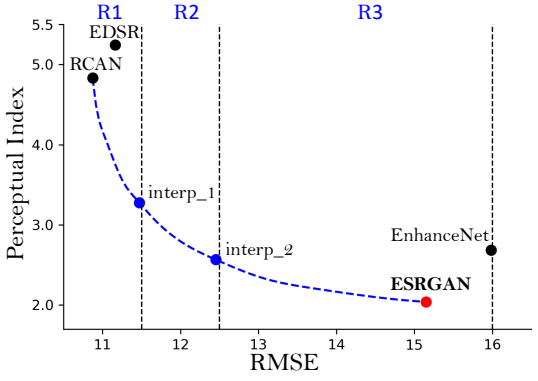


Fig. 1: The super-resolution results of ×4 for SRGAN[2], the proposed ESRGAN and the ground-truth. ESRGAN outperforms SRGAN in sharpness and details.

Several perceptual-driven methods have been proposed to improve the visual quality of SR results. For instance, perceptual loss [13,14] is proposed to optimize super-resolution model in a feature space instead of pixel space. Generative adversarial network [15] is introduced to SR by [1,16] to encourage the network to favor solutions that look more like natural images. The semantic image prior is further incorporated to improve recovered texture details [17]. One of the milestones in the way pursuing visually pleasing results is SRGAN [1]. The basic model is built with residual blocks [18] and optimized using perceptual loss in a GAN framework. With all these techniques, SRGAN significantly improves the overall visual quality of reconstruction over PSNR-oriented methods.

However, there still exists a clear gap between SRGAN results and the ground-truth (GT) images, as shown in Fig. 1. In this study, we revisit the key components of SRGAN and improve the model in three aspects. First, we improve the network structure by introducing the Residual-in-Residual Dense Block (RDDB), which is of higher capacity and easier to train. We also remove Batch Normalization (BN) [19] layers as in [20] and use residual scaling [21,20] and smaller initialization to facilitate training a very deep network. Second, we improve the discriminator using Relativistic average GAN (RaGAN) [2], which learns to judge "whether one image is more realistic than the other" rather than "whether one image is real or fake". Our experiments show that this improvement helps the generator recover more realistic texture details. Third, we propose an improved perceptual loss by using the VGG features *before activation* instead of after activation as in SRGAN. We empirically find that the adjusted perceptual loss provides sharper edges and more visually pleasing results, as will be shown

[2] We use the released results of original SRGAN [1] paper – https://twitter.app.box.com/s/lcue6vlrd01ljkdtdkhmfvk7vtjhetog.



Results on PIRM self val dataset

Method	PI	RMSE
ESRGAN	**2.040**	15.15
interp_2	2.567	12.45
EnhanceNet	2.688	15.99
interp_1	3.279	11.47
RCAN	4.831	10.87
EDSR	5.243	11.16

Fig. 2: Perception-distortion plane on PIRM self validation dataset. We show the baselines of EDSR [20], RCAN [12] and EnhanceNet [16], and the submitted ESRGAN model. The blue dots are produced by image interpolation.

in Sec. 4.4. Extensive experiments show that the enhanced SRGAN, termed ESRGAN, consistently outperforms state-of-the-art methods in both sharpness and details (see Fig. 1 and Fig. 7).

We take a variant of ESRGAN to participate in the PIRM-SR Challenge [3]. This challenge is the first SR competition that evaluates the performance in a perceptual-quality aware manner based on [22], where the authors claim that distortion and perceptual quality are at odds with each other. The perceptual quality is judged by the non-reference measures of Ma's score [23] and NIQE [24], i.e., perceptual index $= \frac{1}{2}((10-\text{Ma})+\text{NIQE})$. A lower perceptual index represents a better perceptual quality.

As shown in Fig. 2, the perception-distortion plane is divided into three regions defined by thresholds on the Root-Mean-Square Error (RMSE), and the algorithm that achieves the lowest perceptual index in each region becomes the regional champion. We mainly focus on region 3 as we aim to bring the perceptual quality to a new high. Thanks to the aforementioned improvements and some other adjustments as discussed in Sec. 4.6, our proposed ESRGAN won the first place in the PIRM-SR Challenge (region 3) with the best perceptual index.

In order to balance the visual quality and RMSE/PSNR, we further propose the network interpolation strategy, which could continuously adjust the reconstruction style and smoothness. Another alternative is image interpolation, which directly interpolates images pixel by pixel. We employ this strategy to participate in region 1 and region 2. The network interpolation and image interpolation strategies and their differences are discussed in Sec. 3.4.

2 Related Work

We focus on deep neural network approaches to solve the SR problem. As a pioneer work, Dong et al. [4,25] propose SRCNN to learn the mapping from LR

to HR images in an end-to-end manner, achieving superior performance against previous works. Later on, the field has witnessed a variety of network architectures, such as a deeper network with residual learning [5], Laplacian pyramid structure [6], residual blocks [1], recursive learning [7,8], densely connected network [9], deep back projection [10] and residual dense network [11]. Specifically, Lim et al. [20] propose EDSR model by removing unnecessary BN layers in the residual block and expanding the model size, which achieves significant improvement. Zhang et al. [11] propose to use effective residual dense block in SR, and they further explore a deeper network with channel attention [12], achieving the state-of-the-art PSNR performance. Besides supervised learning, other methods like reinforcement learning [26] and unsupervised learning [27] are also introduced to solve general image restoration problems.

Several methods have been proposed to stabilize training a very deep model. For instance, residual path is developed to stabilize the training and improve the performance [18,5,12]. Residual scaling is first employed by Szegedy et al. [21] and also used in EDSR. For general deep networks, He et al. [28] propose a robust initialization method for VGG-style networks without BN. To facilitate training a deeper network, we develop a compact and effective residual-in-residual dense block, which also helps to improve the perceptual quality.

Perceptual-driven approaches have also been proposed to improve the visual quality of SR results. Based on the idea of being closer to perceptual similarity [29,14], perceptual loss [13] is proposed to enhance the visual quality by minimizing the error in a feature space instead of pixel space. Contextual loss [30] is developed to generate images with natural image statistics by using an objective that focuses on the feature distribution rather than merely comparing the appearance. Ledig et al. [1] propose SRGAN model that uses perceptual loss and adversarial loss to favor outputs residing on the manifold of natural images. Sajjadi et al. [16] develop a similar approach and further explored the local texture matching loss. Based on these works, Wang et al. [17] propose spatial feature transform to effectively incorporate semantic prior in an image and improve the recovered textures.

Throughout the literature, photo-realism is usually attained by adversarial training with GAN [15]. Recently there are a bunch of works that focus on developing more effective GAN frameworks. WGAN [31] proposes to minimize a reasonable and efficient approximation of Wasserstein distance and regularizes discriminator by weight clipping. Other improved regularization for discriminator includes gradient clipping [32] and spectral normalization [33]. Relativistic discriminator [2] is developed not only to increase the probability that generated data are real, but also to simultaneously decrease the probability that real data are real. In this work, we enhance SRGAN by employing a more effective relativistic average GAN.

SR algorithms are typically evaluated by several widely used distortion measures, e.g., PSNR and SSIM. However, these metrics fundamentally disagree with the subjective evaluation of human observers [1]. Non-reference measures are used for perceptual quality evaluation, including Ma's score [23] and NIQE [24],

both of which are used to calculate the perceptual index in the PIRM-SR Challenge [3]. In a recent study, Blau et al. [22] find that the distortion and perceptual quality are at odds with each other.

3 Proposed Methods

Our main aim is to improve the overall perceptual quality for SR. In this section, we first describe our proposed network architecture and then discuss the improvements from the discriminator and perceptual loss. At last, we describe the network interpolation strategy for balancing perceptual quality and PSNR.

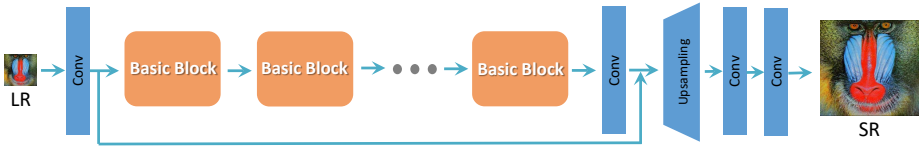


Fig. 3: We employ the basic architecture of SRResNet [1], where most computation is done in the LR feature space. We could select or design "basic blocks" (e.g., residual block [18], dense block [34], RRDB) for better performance.

3.1 Network Architecture

In order to further improve the recovered image quality of SRGAN, we mainly make two modifications to the structure of generator G: 1) remove all BN layers; 2) replace the original basic block with the proposed Residual-in-Residual Dense Block (RRDB), which combines multi-level residual network and dense connections as depicted in Fig. 4.

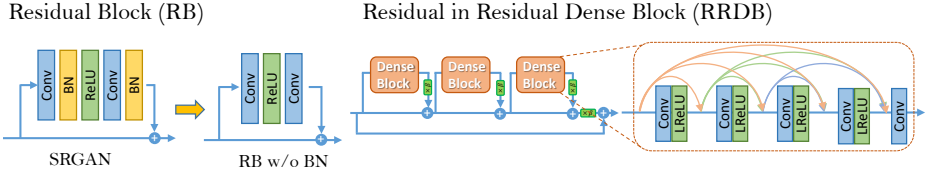


Fig. 4: **Left**: We remove the BN layers in residual block in SRGAN. **Right**: RRDB block is used in our deeper model and β is the residual scaling parameter.

Removing BN layers has proven to increase performance and reduce computational complexity in different PSNR-oriented tasks including SR [20] and deblurring [35]. BN layers normalize the features using mean and variance in a batch during training and use estimated mean and variance of the whole training dataset during testing. When the statistics of training and testing datasets differ a lot, BN layers tend to introduce unpleasant artifacts and limit the generalization ability. We empirically observe that BN layers are more likely to bring

artifacts when the network is deeper and trained under a GAN framework. These artifacts occasionally appear among iterations and different settings, violating the needs for a stable performance over training. We therefore remove BN layers for stable training and consistent performance. Furthermore, removing BN layers helps to improve generalization ability and to reduce computational complexity and memory usage.

We keep the high-level architecture design of SRGAN (see Fig. 3), and use a novel basic block namely RRDB as depicted in Fig. 4. Based on the observation that more layers and connections could always boost performance [20,11,12], the proposed RRDB employs a deeper and more complex structure than the original residual block in SRGAN. Specifically, as shown in Fig. 4, the proposed RRDB has a residual-in-residual structure, where residual learning is used in different levels. A similar network structure is proposed in [36] that also applies a multi-level residual network. However, our RRDB differs from [36] in that we use dense block [34] in the main path as [11], where the network capacity becomes higher benefiting from the dense connections.

In addition to the improved architecture, we also exploit several techniques to facilitate training a very deep network: 1) residual scaling [21,20], i.e., scaling down the residuals by multiplying a constant between 0 and 1 before adding them to the main path to prevent instability; 2) smaller initialization, as we empirically find residual architecture is easier to train when the initial parameter variance becomes smaller. More discussion can be found in the *supplementary material*.

The training details and the effectiveness of the proposed network will be presented in Sec. 4.

3.2 Relativistic Discriminator

Besides the improved structure of generator, we also enhance the discriminator based on the Relativistic GAN [2]. Different from the standard discriminator D in SRGAN, which estimates the probability that one input image x is real and natural, a relativistic discriminator tries to predict the probability that a real image x_r is relatively more realistic than a fake one x_f, as shown in Fig. 5.

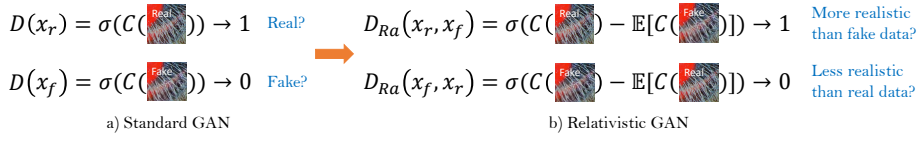


Fig. 5: Difference between standard discriminator and relativistic discriminator.

Specifically, we replace the standard discriminator with the Relativistic average Discriminator RaD [2], denoted as D_{Ra}. The standard discriminator in SRGAN can be expressed as $D(x) = \sigma(C(x))$, where σ is the sigmoid function and $C(x)$ is the non-transformed discriminator output. Then the RaD is formulated as $D_{Ra}(x_r, x_f) = \sigma(C(x_r) - \mathbb{E}_{x_f}[C(x_f)])$, where $\mathbb{E}_{x_f}[\cdot]$ represents the

operation of taking average for all fake data in the mini-batch. The discriminator loss is then defined as:

$$L_D^{Ra} = -\mathbb{E}_{x_r}[\log(D_{Ra}(x_r, x_f))] - \mathbb{E}_{x_f}[\log(1 - D_{Ra}(x_f, x_r))]. \quad (1)$$

The adversarial loss for generator is in a symmetrical form:

$$L_G^{Ra} = -\mathbb{E}_{x_r}[\log(1 - D_{Ra}(x_r, x_f))] - \mathbb{E}_{x_f}[\log(D_{Ra}(x_f, x_r))], \quad (2)$$

where $x_f = G(x_i)$ and x_i stands for the input LR image. It is observed that the adversarial loss for generator contains both x_r and x_f. Therefore, our generator benefits from the gradients from both generated data and real data in adversarial training, while in SRGAN only generated part takes effect. In Sec. 4.4, we will show that this modification of discriminator helps to learn sharper edges and more detailed textures.

3.3 Perceptual Loss

We also develop a more effective perceptual loss L_{percep} by constraining on features before activation rather than after activation as practiced in SRGAN.

Based on the idea of being closer to perceptual similarity [29,14], Johnson et al. [13] propose perceptual loss and it is extended in SRGAN [1]. Perceptual loss is previously defined on the activation layers of a pre-trained deep network, where the distance between two activated features is minimized. Contrary to the convention, we propose to use features before the activation layers, which will overcome two drawbacks of the original design. First, the activated features are very sparse, especially after a very deep network, as depicted in Fig. 6. For example, the average percentage of activated neurons for image 'baboon' after VGG19-54[3] layer is merely 11.17%. The sparse activation provides weak supervision and thus leads to inferior performance. Second, using features after activation also causes inconsistent reconstructed brightness compared with the ground-truth image, which we will show in Sec. 4.4.

Therefore, the total loss for the generator is:

$$L_G = L_{\text{percep}} + \lambda L_G^{Ra} + \eta L_1, \quad (3)$$

where $L_1 = \mathbb{E}_{x_i}||G(x_i) - y||_1$ is the content loss that evaluate the 1-norm distance between recovered image $G(x_i)$ and the ground-truth y, and λ, η are the coefficients to balance different loss terms.

We also explore a variant of perceptual loss in the PIRM-SR Challenge. In contrast to the commonly used perceptual loss that adopts a VGG network trained for image classification, we develop a more suitable perceptual loss for SR – MINC loss. It is based on a fine-tuned VGG network for material recognition [38], which focuses on textures rather than object. Although the gain of perceptual index brought by MINC loss is marginal, we still believe that exploring perceptual loss that focuses on texture is critical for SR.

[3] We use pre-trained 19-layer VGG network[37], where 54 indicates features obtained by the 4^{th} convolution before the 5^{th} maxpooling layer, representing high-level features and similarly, 22 represents low-level features.

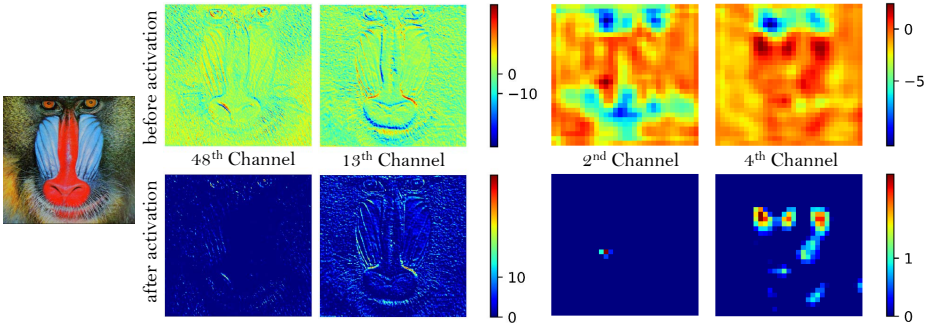


Fig. 6: Representative feature maps before and after activation for image 'baboon'. With the network going deeper, most of the features after activation become inactive while features before activation contains more information.

3.4 Network Interpolation

To remove unpleasant noise in GAN-based methods while maintain a good perceptual quality, we propose a flexible and effective strategy – network interpolation. Specifically, we first train a PSNR-oriented network G_{PSNR} and then obtain a GAN-based network G_{GAN} by fine-tuning. We interpolate all the corresponding parameters of these two networks to derive an interpolated model G_{INTERP}, whose parameters are:

$$\theta_G^{\text{INTERP}} = (1-\alpha)\,\theta_G^{\text{PSNR}} + \alpha\,\theta_G^{\text{GAN}}, \tag{4}$$

where θ_G^{INTERP}, θ_G^{PSNR} and θ_G^{GAN} are the parameters of G_{INTERP}, G_{PSNR} and G_{GAN}, respectively, and $\alpha \in [0, 1]$ is the interpolation parameter.

The proposed network interpolation enjoys two merits. First, the interpolated model is able to produce meaningful results for any feasible α without introducing artifacts. Second, we can continuously balance perceptual quality and fidelity without re-training the model.

We also explore alternative methods to balance the effects of PSNR-oriented and GAN-based methods. For instance, one can directly interpolate their output images (pixel by pixel) rather than the network parameters. However, such an approach fails to achieve a good trade-off between noise and blur, i.e., the interpolated image is either too blurry or noisy with artifacts (see Sec. 4.5). Another method is to tune the weights of content loss and adversarial loss, i.e., the parameter λ and η in Eq. (3). But this approach requires tuning loss weights and fine-tuning the network, and thus it is too costly to achieve continuous control of the image style.

4 Experiments

4.1 Training Details

Following SRGAN [1], all experiments are performed with a scaling factor of $\times 4$ between LR and HR images. We obtain LR images by down-sampling HR

images using the MATLAB bicubic kernel function. The mini-batch size is set to 16. The spatial size of cropped HR patch is 128×128. We observe that training a deeper network benefits from a larger patch size, since an enlarged receptive field helps to capture more semantic information. However, it costs more training time and consumes more computing resources. This phenomenon is also observed in PSNR-oriented methods (see *supplementary material*).

The training process is divided into two stages. First, we train a PSNR-oriented model with the L1 loss. The learning rate is initialized as 2×10^{-4} and decayed by a factor of 2 every 2×10^5 of mini-batch updates. We then employ the trained PSNR-oriented model as an initialization for the generator. The generator is trained using the loss function in Eq. (3) with $\lambda = 5 \times 10^{-3}$ and $\eta = 1 \times 10^{-2}$. The learning rate is set to 1×10^{-4} and halved at $[50k, 100k, 200k, 300k]$ iterations. Pre-training with pixel-wise loss helps GAN-based methods to obtain more visually pleasing results. The reasons are that 1) it can avoid undesired local optima for the generator; 2) after pre-training, the discriminator receives relatively good super-resolved images instead of extreme fake ones (black or noisy images) at the very beginning, which helps it to focus more on texture discrimination.

For optimization, we use Adam [39] with $\beta_1 = 0.9, \beta_2 = 0.999$. We alternately update the generator and discriminator network until the model converges. We use two settings for our generator – one of them contains 16 residual blocks, with a capacity similar to that of SRGAN and the other is a deeper model with 23 RRDB blocks. We implement our models with the PyTorch framework and train them using NVIDIA Titan Xp GPUs.

4.2 Data

For training, we mainly use the DIV2K dataset [40], which is a high-quality (2K resolution) dataset for image restoration tasks. Beyond the training set of DIV2K that contains 800 images, we also seek for other datasets with rich and diverse textures for our training. To this end, we further use the Flickr2K dataset [41] consisting of 2650 2K high-resolution images collected on the Flickr website, and the OutdoorSceneTraining (OST) [17] dataset to enrich our training set. We empirically find that using this large dataset with richer textures helps the generator to produce more natural results, as shown in Fig. 8.

We train our models in RGB channels and augment the training dataset with random horizontal flips and 90 degree rotations. We evaluate our models on widely used benchmark datasets – Set5 [42], Set14 [43], BSD100 [44], Urban100 [45], and the PIRM self-validation dataset that is provided in the PIRM-SR Challenge.

4.3 Qualitative Results

We compare our final models on several public benchmark datasets with state-of-the-art PSNR-oriented methods including SRCNN [4], EDSR [20] and RCAN [12], and also with perceptual-driven approaches including SRGAN [1] and EnhanceNet

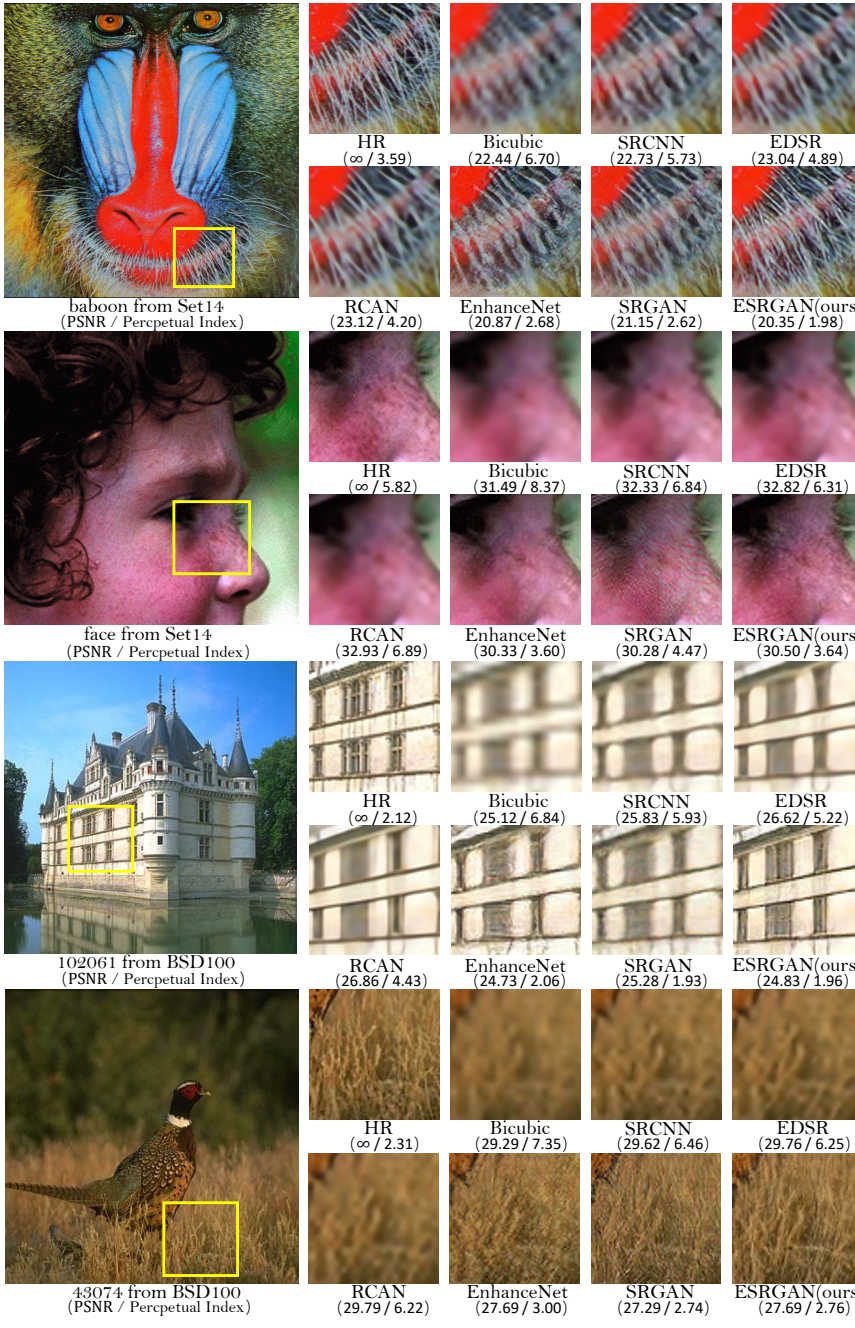


Fig. 7: Qualitative results of ESRGAN. ESRGAN produces more natural textures, e.g., animal fur, building structure and grass texture, and also less unpleasant artifacts, e.g., artifacts in the face by SRGAN.

[16]. Since there is no effective and standard metric for perceptual quality, we present some representative qualitative results in Fig. 7. PSNR (evaluated on the luminance channel in YCbCr color space) and the perceptual index used in the PIRM-SR Challenge are also provided for reference.

It can be observed from Fig. 7 that our proposed ESRGAN outperforms previous approaches in both sharpness and details. For instance, ESRGAN can produce sharper and more natural baboon's whiskers and grass textures (see image 43074) than PSNR-oriented methods, which tend to generate blurry results, and than previous GAN-based methods, whose textures are unnatural and contain unpleasing noise. ESRGAN is capable of generating more detailed structures in building (see image 102061) while other methods either fail to produce enough details (SRGAN) or add undesired textures (EnhanceNet). Moreover, previous GAN-based methods sometimes introduce unpleasant artifacts, e.g., SRGAN adds wrinkles to the face. Our ESRGAN gets rid of these artifacts and produces natural results.

4.4 Ablation Study

In order to study the effects of each component in the proposed ESRGAN, we gradually modify the baseline SRGAN model and compare their differences. The overall visual comparison is illustrated in Fig. 8. Each column represents a model with its configurations shown in the top. The red sign indicates the main improvement compared with the previous model. A detailed discussion is provided as follows.

BN removal. We first remove all BN layers for stable and consistent performance without artifacts. It does not decrease the performance but saves the computational resources and memory usage. For some cases, a slight improvement can be observed from the 2^{nd} and 3^{rd} columns in Fig. 8 (e.g., image 39). Furthermore, we observe that when a network is deeper and more complicated, the model with BN layers is more likely to introduce unpleasant artifacts. The examples can be found in the *supplementary material*.

Before activation in perceptual loss. We first demonstrate that using features before activation can result in more accurate brightness of reconstructed images. To eliminate the influences of textures and color, we filter the image with a Gaussian kernel and plot the histogram of its gray-scale counterpart. Fig. 9a shows the distribution of each brightness value. Using activated features skews the distribution to the left, resulting in a dimmer output while using features before activation leads to a more accurate brightness distribution closer to that of the ground-truth.

We can further observe that using features before activation helps to produce sharper edges and richer textures as shown in Fig. 9b (see bird feather) and Fig. 8 (see the 3^{rd} and 4^{th} columns), since the dense features before activation offer a stronger supervision than that a sparse activation could provide.

RaGAN. RaGAN uses an improved relativistic discriminator, which is shown to benefit learning sharper edges and more detailed textures. For example, in

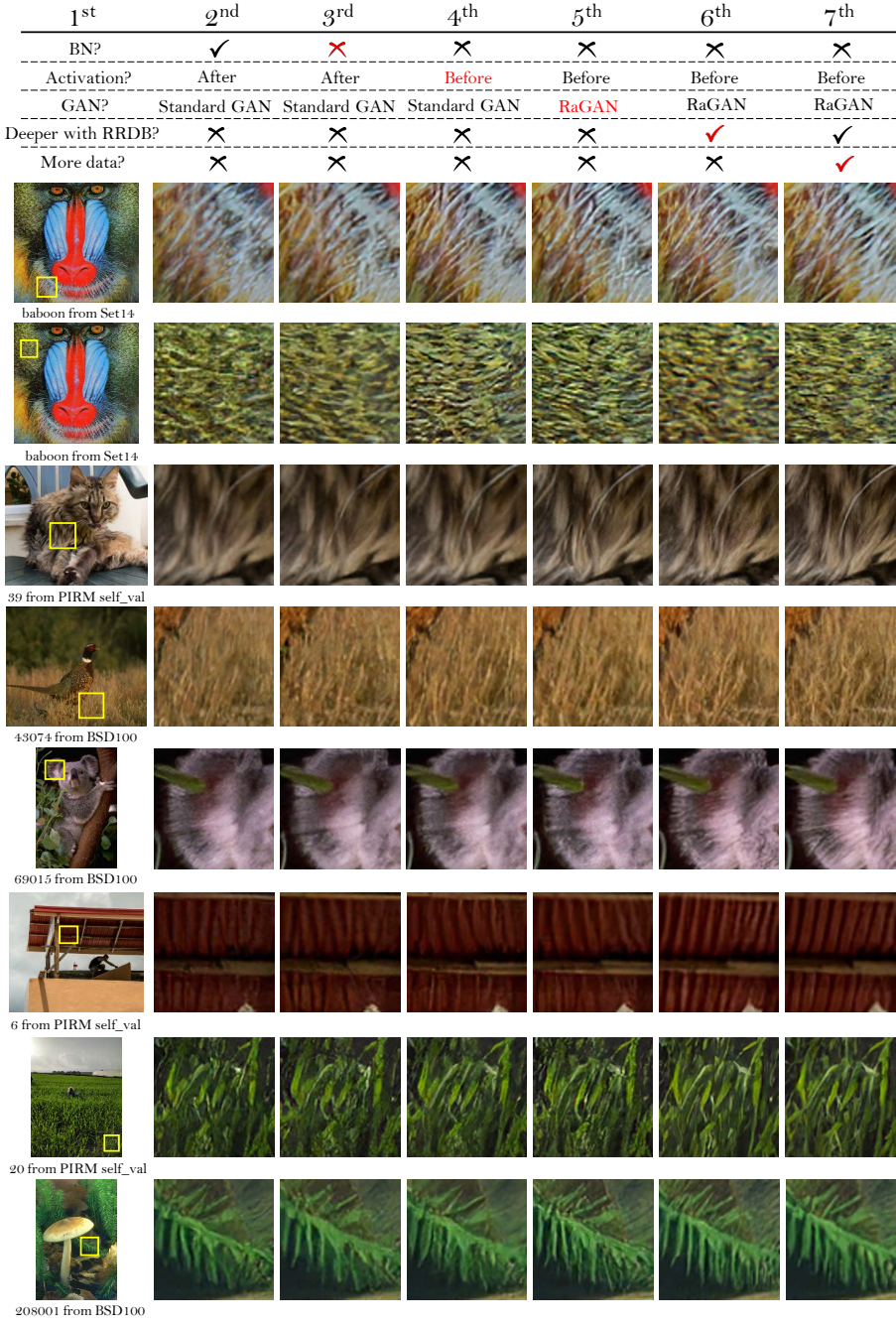
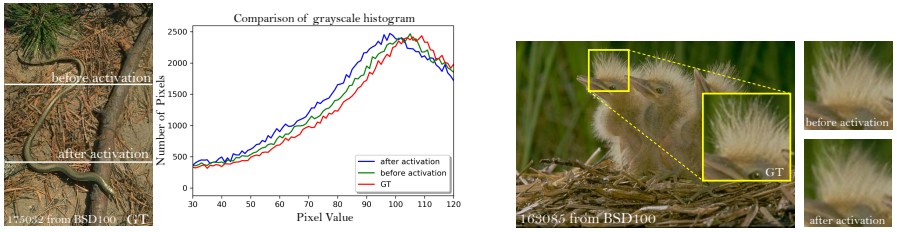


Fig. 8: Overall visual comparisons for showing the effects of each component in ESRGAN. Each column represents a model with its configurations in the top. The red sign indicates the main improvement compared with the previous model.



(a) brightness influence (b) detail influence

Fig. 9: Comparison between before activation and after activation.

the 5^{th} column of Fig. 8, the generated images are sharper with richer textures than those on their left (see the baboon, image 39 and image 43074).

Deeper network with RRDB. Deeper model with the proposed RRDB can further improve the recovered textures, especially for the regular structures like the roof of image 6 in Fig. 8, since the deep model has a strong representation capacity to capture semantic information. Also, we find that a deeper model can reduce unpleasing noises like image 20 in Fig. 8.

In contrast to SRGAN, which claimed that deeper models are increasingly difficult to train, our deeper model shows its superior performance with easy training, thanks to the improvements mentioned above especially the proposed RRDB without BN layers.

4.5 Network Interpolation

We compare the effects of network interpolation and image interpolation strategies in balancing the results of a PSNR-oriented model and GAN-based method. We apply simple linear interpolation on both the schemes. The interpolation parameter α is chosen from 0 to 1 with an interval of 0.2.

As depicted in Fig. 10, the pure GAN-based method produces sharp edges and richer textures but with some unpleasant artifacts, while the pure PSNR-oriented method outputs cartoon-style blurry images. By employing network interpolation, unpleasing artifacts are reduced while the textures are maintained. By contrast, image interpolation fails to remove these artifacts effectively.

Interestingly, it is observed that the network interpolation strategy provides a smooth control of balancing perceptual quality and fidelity in Fig. 10.

4.6 The PIRM-SR Challenge

We take a variant of ESRGAN to participate in the PIRM-SR Challenge [3]. Specifically, we use the proposed ESRGAN with 16 residual blocks and also empirically make some modifications to cater to the perceptual index. 1) The MINC loss is used as a variant of perceptual loss, as discussed in Sec. 3.3. Despite the marginal gain on the perceptual index, we still believe that exploring perceptual loss that focuses on texture is crucial for SR. 2) Pristine dataset [24], which is

Fig. 10: The comparison between network interpolation and image interpolation.

used for learning the perceptual index, is also employed in our training; 3) a high weight of loss L_1 up to $\eta = 10$ is used due to the PSNR constraints; 4) we also use back projection [46] as post-processing, which can improve PSNR and sometimes lower the perceptual index.

For other regions 1 and 2 that require a higher PSNR, we use image interpolation between the results of our ESRGAN and those of a PSNR-oriented method RCAN [12]. The image interpolation scheme achieves a lower perceptual index (lower is better) although we observed more visually pleasing results by using the network interpolation scheme. Our proposed ESRGAN model won the first place in the PIRM-SR Challenge (region 3) with the best perceptual index.

5 Conclusion

We have presented an ESRGAN model that achieves consistently better perceptual quality than previous SR methods. The method won the first place in the PIRM-SR Challenge in terms of the perceptual index. We have formulated a novel architecture containing several RDDB blocks without BN layers. In addition, useful techniques including residual scaling and smaller initialization are employed to facilitate the training of the proposed deep model. We have also introduced the use of relativistic GAN as the discriminator, which learns to judge whether one image is more realistic than another, guiding the generator to recover more detailed textures. Moreover, we have enhanced the perceptual loss by using the features before activation, which offer stronger supervision and thus restore more accurate brightness and realistic textures.

Acknowledgement. This work is supported by SenseTime Group Limited, the General Research Fund sponsored by the Research Grants Council of the Hong Kong SAR (CUHK 14241716, 14224316. 14209217), National Natural Science Foundation of China (U1613211) and Shenzhen Research Program (JCYJ20170818164704758, JCYJ20150925163005055).

References

1. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR. (2017)
2. Jolicoeur-Martineau, A.: The relativistic discriminator: a key element missing from standard gan. arXiv preprint arXiv:1807.00734 (2018)
3. Blau, Y., Mechrez, R., Timofte, R., Michaeli, T., Zelnik-Manor, L.: The pirm challenge on perceptual super resolution. https://www.pirm2018.org/PIRM-SR.html (2018)
4. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: ECCV. (2014)
5. Kim, J., Kwon Lee, J., Mu Lee, K.: Accurate image super-resolution using very deep convolutional networks. In: CVPR. (2016)
6. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR. (2017)
7. Kim, J., Kwon Lee, J., Mu Lee, K.: Deeply-recursive convolutional network for image super-resolution. In: CVPR. (2016)
8. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR. (2017)
9. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration. In: ICCV. (2017)
10. Haris, M., Shakhnarovich, G., Ukita, N.: Deep backprojection networks for super-resolution. In: CVPR. (2018)
11. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: CVPR. (2018)
12. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: ECCV. (2018)
13. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV. (2016)
14. Bruna, J., Sprechmann, P., LeCun, Y.: Super-resolution with deep convolutional sufficient statistics. In: ICLR. (2015)
15. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. (2014)
16. Sajjadi, M.S., Schölkopf, B., Hirsch, M.: Enhancenet: Single image super-resolution through automated texture synthesis. In: ICCV. (2017)
17. Wang, X., Yu, K., Dong, C., Loy, C.C.: Recovering realistic texture in image super-resolution by deep spatial feature transform. In: CVPR. (2018)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)
19. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICMR. (2015)

20. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: CVPRW. (2017)
21. Szegedy, C., Ioffe, S., Vanhoucke, V.: Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv:1602.07261 (2016)
22. Blau, Y., Michaeli, T.: The perception-distortion tradeoff. In: CVPR. (2017)
23. Ma, C., Yang, C.Y., Yang, X., Yang, M.H.: Learning a no-reference quality metric for single-image super-resolution. CVIU **158** (2017) 1–16
24. Mittal, A., Soundararajan, R., Bovik, A.C.: Making a completely blind image quality analyzer. IEEE Signal Process. Lett. **20**(3) (2013) 209–212
25. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. TPAMI **38**(2) (2016) 295–307
26. Yu, K., Dong, C., Lin, L., Loy, C.C.: Crafting a toolchain for image restoration by deep reinforcement learning. In: CVPR. (2018)
27. Yuan, Y., Liu, S., Zhang, J., Zhang, Y., Dong, C., Lin, L.: Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In: CVPRW. (2018)
28. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: ICCV. (2015)
29. Gatys, L., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural networks. In: NIPS. (2015)
30. Mechrez, R., Talmi, I., Shama, F., Zelnik-Manor, L.: Maintaining natural image statistics with the contextual loss. arXiv preprint arXiv:1803.04626 (2018)
31. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
32. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: NIPS. (2017)
33. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
34. Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L.: Densely connected convolutional networks. In: CVPR. (2017)
35. Nah, S., Kim, T.H., Lee, K.M.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: CVPR. (2017)
36. Zhang, K., Sun, M., Han, X., Yuan, X., Guo, L., Liu, T.: Residual networks of residual networks: Multilevel residual networks. IEEE Transactions on Circuits and Systems for Video Technology (2017)
37. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
38. Bell, S., Upchurch, P., Snavely, N., Bala, K.: Material recognition in the wild with the materials in context database. In: CVPR. (2015)
39. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR. (2015)
40. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: CVPRW. (2017)
41. Timofte, R., Agustsson, E., Van Gool, L., Yang, M.H., Zhang, L., Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M., et al.: Ntire 2017 challenge on single image super-resolution: Methods and results. In: CVPRW. (2017)
42. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: BMVC, BMVA press (2012)
43. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: International Conference on Curves and Surfaces, Springer (2010)

44. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV. (2001)
45. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR. (2015)
46. Timofte, R., Rothe, R., Van Gool, L.: Seven ways to improve example-based single image super resolution. In: CVPR. (2016)
47. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: International Conference on Artificial Intelligence and Statistics. (2010)

ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks Supplementary File

Xintao Wang[1], Ke Yu[1], Shixiang Wu[2], Jinjin Gu[3], Yihao Liu[4], Chao Dong[2], Chen Change Loy[5], Yu Qiao[2], Xiaoou Tang[1]

[1]CUHK-SenseTime Joint Lab, The Chinese University of Hong Kong
[2]SIAT-SenseTime Joint Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences [3]The Chinese University of Hong Kong, Shenzhen
[4]University of Chinese Academy of Sciences [5]Nanyang Technological University, Singapore
{wx016,yk017,xtang}@ie.cuhk.edu.hk, {sx.wu,chao.dong,yu.qiao}@siat.ac.cn
liuyihao14@mails.ucas.ac.cn, 115010148@link.cuhk.edu.cn, ccloy@ntu.edu.sg

Abstract. In this supplementary file, we first show more examples of Batch-Normalization (BN) related artifacts in Section 1. Then we introduce several useful techniques that facilitate training very deep models in Section 2. The analysis of the influence of different datasets and training patch size is depicted in Section 3 and Section 4, respectively. Finally, in Section 5, we provide more qualitative results for visual comparison.

1 BN artifacts

We empirically observe that BN layers tend to bring artifacts. These artifacts, namely BN artifacts, occasionally appear among iterations and different settings, violating the needs for a stable performance over training. In this section, we present that the network depth, BN position, training dataset and training loss have impact on the occurrence of BN artifacts and show corresponding visual examples in Fig. 1, 2 and 3.

Table 1: Experimental variants for exploring BN artifacts.

Name	Number of RB	BN position	training dataset	training loss
Exp_base	16	LR space	DIV2K	$L1$
Exp_BNinHR	16	LR and HR space	DIV2K	$L1$
Exp_64RB	64	LR space	DIV2K	$L1$
Exp_skydata	16	LR space	sky data	$L1$
Exp_SRGAN	16	LR space	DIV2K	$VGG + GAN + L1$

To explore BN artifacts, we conduct several experiments as shown in Tab. 1. The baseline is similar to SRResNet [1] with 16 Residual Blocks (RB) and all the BN layers are in the LR space, i.e., before up-sampling layers. The baseline setting is unlikely to introduce BN artifacts in our experiments. However, if the network goes deeper or there is an extra BN layer in HR space (i.e., after up-sampling layers), BN artifacts are more likely to appear (see examples in Fig. 1).

When we replace the training dataset of the baseline with the sky dataset [17], the BN artifacts appear (see examples in Fig. 1). BN layers normalize the features

Exp_64RB
Deeper network with 64 RBs

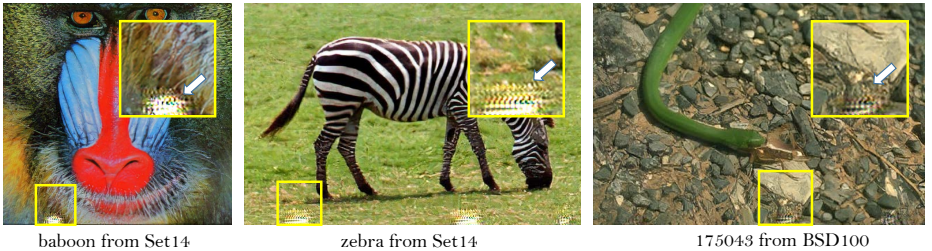
Exp_BNinHR
with BN in HR space

Exp_skydata
training with sky dataset

Fig. 1: Examples of BN artifacts in PSNR-oriented methods. The BN artifacts are more likely to appear in deeper networks, with BN in HR space and using mismatched dataset whose statistics are different from those of testing dataset.

using mean and variance in a batch during training while using estimated mean and variance of the whole training dataset during testing. Therefore, when the statistics of training (e.g., sky dataset) and testing datasets differ a lot, BN layers tend to introduce unpleasant artifacts and limit the generalization ability.

Training in a GAN framework increases the occurrence probability of BN artifacts in our experiments. We employ the same network structure as baseline and replace the $L1$ loss with $VGG + GAN + L1$ loss. The BN artifacts become more likely to appear and the visual examples are shown in Fig. 2.



baboon from Set14

zebra from Set14

175043 from BSD100

Fig. 2: Examples of BN artifacts in models under the GAN framework.

The BN artifacts occasionally appear over training, i.e, the BN artifacts appear, disappear and change on different training iterations, as shown in Fig 3. We therefore remove BN layers for stable training and consistent performance. The reasons behind and potential solutions remain to be further studied.

2 Useful techniques to train a very deep network

Since we remove BN layers for stable training and consistent performance, training a very deep network becomes a problem. Despite the proposed Residual-in-Residual Dense Block (RRDB), which takes advantages of residual learning and more connections, we also find two useful techniques to ease the training of a very deep networks – smaller initialization and residual scaling.

Fig. 3: Evolution of the model Exp_BNinHR (with BN in HR space) during training progress.The BN artifacts occasionally appear over training, resulting in unstable performance.

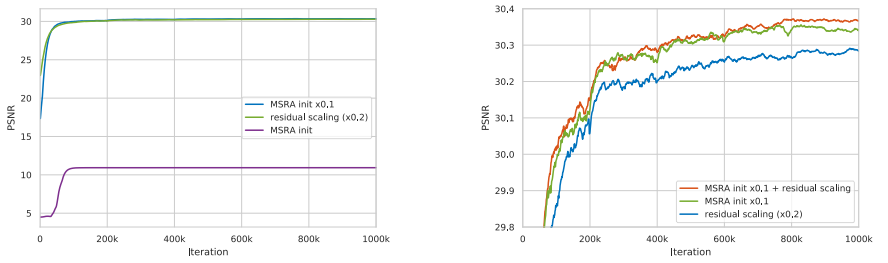
Initialization is important for a very deep network especially without BN layers [47,28]. He et al. [28] propose a robust initialization method, namely MSRA initialization, that is suitable for VGG-style network (plain network without residual connections). The assumption is that a proper initialization method should avoid reducing or magnifying the magnitudes of input signals exponentially. It is worth noting that this assumption no longer holds due to the residual path in ResNet [18], leading to a magnified magnitudes of input signals. This problem is alleviated by normalizing the features with BN layers [19]. For a very deep network containing residual blocks without BN layers, a new initialization method should be applied. We find a smaller initialization than MSRA initialization (multiplying 0.1 for all initialization parameters that calculated by MSRA initialization) works well in our experiments.

Another method for training deeper networks is residual learning, proposed by Szegedy et al. [21] and also used in used in EDSR [20]. It scales down the residuals by multiplying a constant between 0 and 1 before adding them to the main path to prevent instability. In our settings, for each residual block, the residual features after the last convolution layer are multiplied by 0.2. Intuitively, the residual scaling can be interpreted to correct the improper initialization, thus avoiding magnifying the magnitudes of input signals in residual networks.

We use a very deep network containing 64 RBs for experiments. As shown in Fig. 4a, if we simply use MSRA initialization, the network falls into an extremely bad local minimum with poor performance. However, smaller initialization ($\times 0.1$) helps the network to jump out the bad local minimum and achieve good performance. The zoomed curves are shown in Fig. 4b. Smaller initialization achieves a higher PSNR than residual scaling. In addition, we can use both techniques to further obtain a slight improvement.

3 The influence of different datasets

First we show that larger datasets lead to better performance for PSNR-oriented methods. We use a large model, where 23 Residual-in-Residual Blocks (RRDB) are placed before the upsampling layer followed by two convolution layers for reconstruction. The overall comparison of quantitative evaluation can be found in Tab. 2.



(a) (b)

Fig. 4: Smaller initialization and residual scaling benefit the convergence and the performance of very deep networks (PSNR is evaluated on Set5 with RGB channels).

A widely used training dataset is DIV2K [40] that contains 800 images. We also explore other datasets with more diverse scenes – Flickr2K dataset [41] consisting of 2650 2K high-resolution images collected on the Flickr website. It is observed that the merged dataset with DIV2K and Flickr2K, namely DF2K dataset, increases the PSNR performance (see Tab. 2).

Table 2: Quantitative evaluation of state-of-the-art PSNR-oriented SR algorithms: average PSNR/SSIM on Y channel. The best and second best results are **highlighted** and underlined, respectively.

Method with training data		Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
Bicubic	-	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	24.89/0.7866
SRCNN [4]	291	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221	27.58/0.8555
MemNet [9]	291	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630	29.42/0.8942
EDSR [20]	DIV2K	32.46/0.8968	28.80/0.7876	27.71/0.7420	26.64/0.8033	31.02/0.9148
RDN [11]	DIV2K	32.47/0.8990	28.81/0.7871	27.72/0.7419	26.61/0.8028	31.00/0.9151
RCAN [12]	DIV2K	32.63/0.9002	28.87/0.7889	27.77/0.7436	26.82/ 0.8087	31.22/ 0.9173
RRDB(ours)	DIV2K	32.60/0.9002	28.88/0.7896	27.76/ 0.7432	26.73/0.8072	31.16/0.9164
RRDB(ours)	DF2K	**32.73/0.9011**	**28.99/0.7917**	**27.85/0.7455**	**27.03/0.8153**	**31.66/0.9196**

For perceptual-driven methods that focus on texture restoration, we further enrich the training set with OutdoorSceneTraining (OST) [17] dataset with diverse natural textures. We employ the large model with 23 RRDB blocks. A subset of ImageNet containing about $450k$ images is also used for comparison. The qualitative results are shown in Fig. 5. Training with ImageNet introduces new types of artifacts as in image zebra of Fig. 5 while OST dataset benefits the grass restoration.

4 The influence of training patch size

We observe that training a deeper network benefits from a larger patch size, since an enlarged receptive field helps the network to capture more semantic

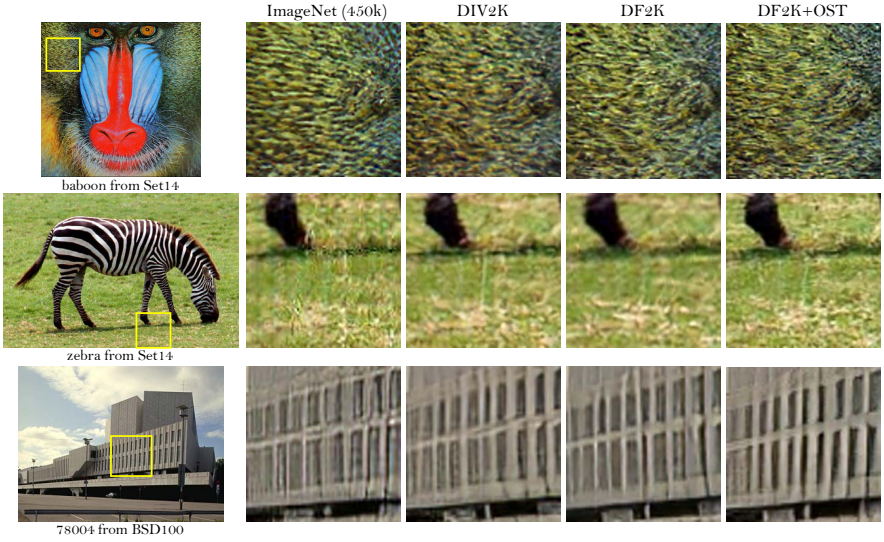
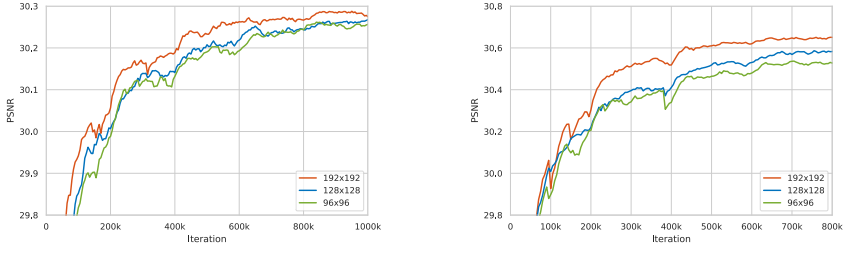


Fig. 5: The influence of different datasets.

information. We try training patch size 96×96, 128×128 and 192×192 on models with 16 RBs and 23 RRDBs (larger model capacity). The training curves (evaluated on Set5 with RGB channels) are shown in Fig. 6.

It is observed that both models benefit from larger training patch size. Moreover, the deeper model achieves more improvement (∼0.12dB) than the shallower one (∼0.04dB) since larger model capacity is capable of taking full advantage of larger training patch size.

However, larger training patch size costs more training time and consumes more computing resources. As a trade-off, we use 192×192 for PSNR-oriented methods and 128×128 for perceptual-driven methods.



(a) 16 Residual Blocks (b) 23 RRDBs

Fig. 6: The influence of training patch size (PSNR is evaluated on Set5 with RGB channels).

5 More qualitative comparison

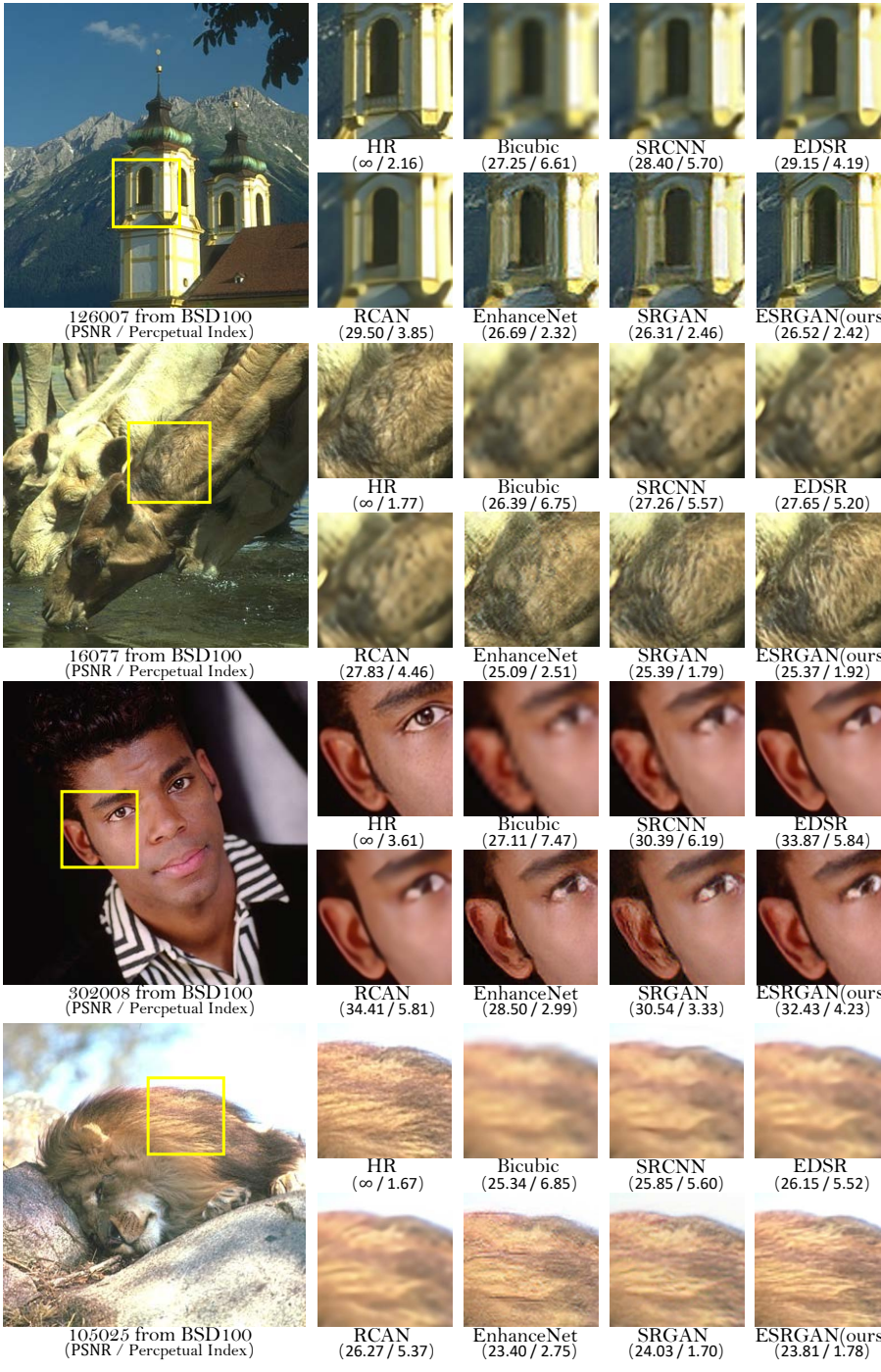


Fig. 7: More qualitative results. PSNR (evaluated on the Y channel) and the perceptual index are also provided for reference.