Fine-grained Attention and Feature-sharing Generative Adversarial Networks for Single Image Super-Resolution

Yitong Yan, Chuangchuang Liu, Changyou Chen, Xianfang Sun, Longcun Jin*, *Member, IEEE,* Xinyi Peng, and Xiang Zhou

Abstract—Traditional super-resolution (SR) methods by minimize the mean square error usually produce images with over-smoothed and blurry edges, due to the lack of high-frequency details. In this paper, we propose two novel techniques within the generative adversarial network framework to encourage generation of photo-realistic images for image super-resolution. Firstly, instead of producing a single score to discriminate real and fake images, we propose a variant, called Fine-grained Attention Generative Adversarial Network (FASRGAN), to discriminate each pixel of real and fake images. FASRGAN adopts a UNet-like network as the discriminator with two outputs: an image score and an image score map. The score map has the same spatial size as the HR/SR images, serving as the fine-grained attention to represent the degree of reconstruction difficulty for each pixel. Secondly, instead of using different networks for the generator and the discriminator, we introduce a feature-sharing variant (denoted as Fs-SRGAN) for both the generator and the discriminator. The sharing mechanism can maintain model express power while making the model more compact, and thus can improve the ability of producing high-quality images. Quantitative and visual comparisons with state-of-the-art methods on benchmark datasets demonstrate the superiority of our methods. We further apply our super-resolution images for object recognition, which further demonstrates the effectiveness of our proposed method. The code is available at https://github.com/Rainyfish/FASRGAN-and-Fs-SRGAN.

Index Terms—Fine-grained attention, feature-sharing, generative adversarial network, image super-resolution.

I. Introduction

SINGLE image super-resolution (SISR), which aims to recover a high-resolution (HR) image from its low-solution (LR) version, has been an active research topic in computer graphic and vision for decades. SISR has also attracted increasing attention in both academia and industry, with applications in various fields such as medical imaging, security surveillance, object recognition and so on. However, SISR is a typically ill-posed problem due to the irreversible image degradation process, *i.e.*, multiple HR images can be generated from one single LR image. Learning the mapping between HR and LR images plays an important role in addressing this problem.

Recently, deep convolution neural networks (CNNs) have been shown great success in many vision tasks, such as image classification, object detection, and image restoration. Dong *et al.* [1] first proposed a three-layer CNN for single image super-resolution (SRCNN) to directly learn the mapping from LR to HR images. Since then the CNN-based methods [2] have been dominant for the SR problem because they greatly improved the reconstruction performance. Kumar *et al.* [3] tapped into the ability of polynomial neural networks to hierarchically learn refinements of a function that maps LR to HR patches. VDSR [4] obtained remarkable performance by increasing the depth of the network to 20, proving the importance of the network depth for detecting effective features of images. EDSR [5] removed unnecessary batch normalization layer in the ResNet [6] architecture and widened the channels, significantly improving the performance. RCAN [7] applied residual in residual structure to construct a very deep network and used a channel attention mechanism to adaptively rescale features.

The aforementioned methods use the optimization idea of minimizing the mean squared error (MSE) between the recovered SR image and the corresponding HR image. Such methods are designed to maximize the peak signal-to-noise ratio (PSNR). However, they typically produce over-smoothed edges and lack tiny textures. To produce photo-realistic SR images, Ledig *et al.*[8] first used the generative adversarial network (GAN) [9] to match the underlying distributions of HR and SR images. ESRGAN [10] further extended the generator network and used the Relativistic Discriminator [11] to produce more photo-realistic images. However, as shown in Fig.1, the discriminator in these GAN-based methods only outputs a score of the whole input SR/HR image, which is a coarse way to guide the generator. Furthermore, the generator and discriminator of these works are considered to be independent, while we believe there should be significant information to be shared. For example, the lower-level parts of the two networks both aim at extracting tiny features such as corners and edges, which we believe should be correlated.

To address these limitations, we propose two novel tech-

Y. Yan and C. Liu contribute equally in this work and share the first authorship.

*Corresponding author: Longcun Jin (lcjin@scut.edu.cn).

Y. Yan, C. Liu, L. J and X. Peng were with the School of Software Engineering, South China University of Technology, Guangzhou, China (e-mail: seyannis_yan@mail.scut.edu.cn; selcc@mail.scut.edu.cn; lcjin@scut.edu.cn; adxypeng@scut.edu.cn).

C. Chen was with the Department of Computer Science and Engineering, University at Buffalo, State University of New York, NY, USA (e-mail: changyou@buffalo.edu).

X. Sun was with the School of Computer Science and Informatics Cardiff University, UK (e-mail: sunx2@cardiff.ac.uk).

X. Zhou was with the School of Data Science and Department of Mathematics at College of Science, City University of Hong Kong, Hong Kong, China (e-mail: Xiang.Zhou@cityu.edu.hk).

niques based on the GAN framework for image super-resolution, a fine-grained attention mechanism for the discriminator and a feature-sharing network component for both the generator and the discriminator. Specifically, we use a UNet-like [12] discriminator (Fig.2) to introduce a fine-grained attention in the GAN (FASRGAN). Our discriminator produces two outputs, a score of the whole input image and a fine-grained score map of every pixel in the image. The score map shares the same spatial size as the input image, and measures the degree of differences at each pixel between the generated and the true distributions. To produce better visual quality images, we incorporate the score map into the loss function with an attention mechanism to make the generator pay more attention on the hard parts of an image, instead of treating all parts equally. In addition, we propose a feature-sharing mechanism (Fig.3) to align the low-level feature extraction of both the generator and the discriminator (Fs-SRGAN). This novel structure can significantly reduce the number of parameters and improve the performance.

Overall, our main contributions are three-fold:

- We propose a novel UNet-like discriminator to generate a single score for the whole image and a pixel-wise score map of the input image. We further incorporate the score map into the loss function with an attention mechanism to define the generator. This attention mechanism makes the generator focus on hard parts of an image for generation.
- We introduce a feature-sharing mechanism to define the shared low-level feature extraction for the generator and the discriminator. This reduces the number of model parameters and helps the generator and the discriminator extract more effective features.
- The proposed two components are general, and can be applied to other GAN-based SR models. Extensive experiments on benchmark datasets illustrate the superiority of our proposed methods compared with current state-of-the-art methods.

The remainder of the paper is organized as follows. Section II describes related works. The proposed GAN-based methods are presented in Section III. Experimental results are discussed in Section IV. Finally, the conclusions are drawn in Section V.

II. Related Work

Traditional SISR methods are exemplar or dictionary based. However, these methods are limited by the size of datasets or dictionaries, and are usually time-consuming. These shortcomings can be greatly alleviated by the recent CNN-based methods [2].

In their pioneer work, Dong *et al.* [1] applied convolutional neural networks with three layers for SISR, namely SRCNN, to learn a mapping from LR to HR images in an end-to-end manner. Kim *et al.* [4] increases the depth of the network to 20, achieving great improvement in accuracy compared to SRCNN. Instead of using the interpolated LR images as the inputs of network, FSRCNN [13] extracted features from the origin LR images and upscaled the spatial size by upsampling layers at the tail of the network. This architecture is widely used in the subsequent SR methods. Various advanced upsampling structures have been proposed recently, for instance, deconvolutional layer [14] , sub-pixel convolution [15], and EUSR [16]. LapSRN [17] progressively reconstructed an HR image with increasing scales of an input image by the Laplacian pyramid structure. Lim *et al.* [5] proposed a very large network (EDSR) and its multi-scale version (MDSR), which removed the unnecessary batch normalization layer in the ResNet [6] and greatly improved super-resolution performance. D-DBPN [18] introduced an error-correcting feedback mechanism to learn relationships between LR features and SR features. ZSSR [19] used a unsupervised method to learn the mapping between HR images and LR images. DIP [20] showed that the structure of a generator network can capture a large amount of low-level image statistics before any learning is performed, which can be used as a handcrafted prior with excellent results in super-resolution and other standard inverse problems. To address the real-world LR image problem, Fritsche *et al.* [21] proposed to separate the low and high image frequencies and treat them in different ways during training. Adversarial training is used to modify only the high, not the low frequencies. RDN [22] combined dense and residual connections to make full use of information of LR images. Different from RDN, MS-RHDN [23] proposed multi-scale residual hierarchical dense networks to extract multi-scale and hierarchical feature maps. Yang *et al.* [24] proposed a deep recurrent fusion network (DRFN) for SR with large-scale factors, which used transposed convolution to jointly extract and upsample raw features from the input and used multi-level fusion for reconstruction. SeaNet [25] proposed a Soft-edge assisted Network to reconstruct the high-quality SR images with the help of image soft-edge. Zhang *et al.* [26] proposed an adaptive importance learning scheme to enhance the performance of the lightweight SISR network architecture. RCAN [7] applied channel-attention mechanism to adaptively rescale channel-wise features. SAN [27] further proposed a second-order channel attention (SOCA) module to rescale the features instead of global average pooling.

The aforementioned methods aim to achieve high PSNR and SSIM [28] values. However, these criteria usually cause heavy over-smoothed edges and artifacts. Images generated by these MSE-based SR methods lose various high-frequency details and have a bad perceptual quality. To generate more photo-realistic images, Ledig *et al.* [8] firstly introduced generative adversarial network into image super-resolution, called SRGAN. SRGAN combined a perceptual loss and an adversarial loss to improve the reality of generated images. But some visually implausible artifacts still could be found in some generated images. To reduce the artifacts, EnhanceNet [29] combined a pixel-wise loss in the image space, a perceptual loss in the feature space, a texture matching loss and an adversarial loss. The contextual loss [30] was a kind of perceptual loss to make the generated images as similar as possible to ground-truth images. Yan *et al.* [31] firstly trained a novel full-reference image quality assessment (FR-IQA) approach for SISR, then employed the proposed loss function (SR-IQA) to train their SR network which contains their proposed highway unit. In addition, they also integrated SR-IQA loss to the GAN-based SR method to achieve better results for both accuracy

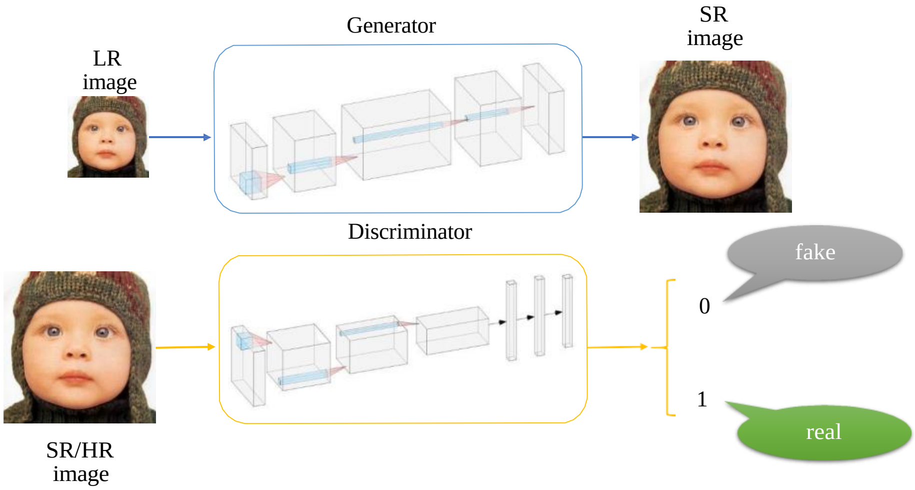

Fig. 1. The architecture of GAN-based Super-Resolution method. The generator aims to reconstruct photo-realistic SR images, while the discriminator distinguishes the SR image from the ground-truth HR image.

and perceptual quality. Based on SRGAN, ESRGAN [10] *i*) substituted the standard residual block with a residual-in-residual dense block, *ii*) removed batch normalization layers, *iii*) utilized VGG feature before activated as perceptual loss, and *iv*) replaced the standard discriminator with Relativistic Discriminator proposed in RaGAN [11]. Noteworthily, ESRGAN won the first place in the 2018 PIRM Challenge on Perceptual Image Super-Resolution [32], which pursued the high perceptual-quality images. RankSRGAN [33] firstly trained a ranker to learn the behavior of perceptual metrics and then introduced a rank-content loss to optimize the perceptual quality.

III. Proposed Methods

A. Overview

Our methods aim to reconstruct a high-resolution image $I^{\mathrm{SR}} \in R^{Wr \times Hr \times C}$ from a low-resolution image $I^{\mathrm{LR}} \in R^{W \times H \times C}$, where W and H are the width and height of the LR image, r is the upscaling factor, and C is the number of channels of the color space. This section details our two strategies within the GAN framework for image super-resolution in order: FASRGAN and Fs-SRGAN. Specifically, we propose a fine-grained attention mechanism in FASRGAN to make the generator focus on the difficult parts of image reconstruction based on the score map from the UNet-like discriminator, instead of treating each part equally. We further propose a feature-sharing mechanism in Fs-SRGAN by sharing the low-level feature extractor of the generator and the discriminator. Both networks update the gradient of the shared low-level feature extractor in the training phase, which could make the model more compact while keeping enough representation power. These two strategies contribute to the overall perceptual quality for SR, respectively. For simplicity, we use the same network architecture as ESRGAN [10] for the generator to generate the SR images from the input LR images.

B. Fine-grained Attention Generator Adversarial Networks

Our proposed fine-grained attention GAN (FASRGAN) designs a specific discriminator for SISR. As discussed above and shown in Fig.1, the discriminator in a standard GAN-based SR model outputs a score of the whole input SR/HR image. This can be considered as a coarse way to judge an input image and cannot discriminate local features of inputs. To tackle this problem, the proposed FASRGAN defines a UNet-like discriminator contained two outputs, corresponding to a score of the whole image and a fine-grained score map. The score map has the same size as the input image and is used for pixel-wise discrimination. The proposed discriminator is illustrated in Fig. 2.

1) A UNet-like Discriminator: The UNet-like discriminator with two outputs can be divided into two parts: an encoder and a decoder.

Encoder. Similar to the standard discriminator D in ESRGAN, the encoder part of the proposed UNet-like discriminator uses a standard max-pooling layer with a stride of 2 to reduce the spatial size of a feature map and increase receptive fields. At the same time, the number of channels is increased for improving representative ability. At the end of the encoder, two fully connected layers are added to output a score, measuring the overall probability of an input image x being real or fake. We further enhance the discriminator based on the Relativistic GAN [11], which has also been used in ESRGAN [10]. The loss function is defined as:

$$\begin{aligned} L^{D}_{adv} &= \mathbb{E}_{x_r}[\log(1 - D_{Ra}(x_r, x_f))] \\ &\quad + \mathbb{E}_{x_f}[\log(D_{Ra}(x_f, x_r))] \\ &= \mathbb{E}_{x_r}[\log(1 - \sigma(C(x_r) - \mathbb{E}_{x_f}[C(x_f)]))] \\ &\quad + \mathbb{E}_{x_f}[\log(\sigma(C(x_f) - \mathbb{E}_{x_r}[C(x_r)]))], \end{aligned} \tag{1}$$

where x_r and x_f stand for the ground-truth image and the generated SR image, respectively. $D_{Ra}(\cdot)$ refers to the function of the relativistic discriminator, which tries to predict the probability that a real image x_r is more realistic than a fake one x_f; $C(x)$ is the discriminator output before sigmoid

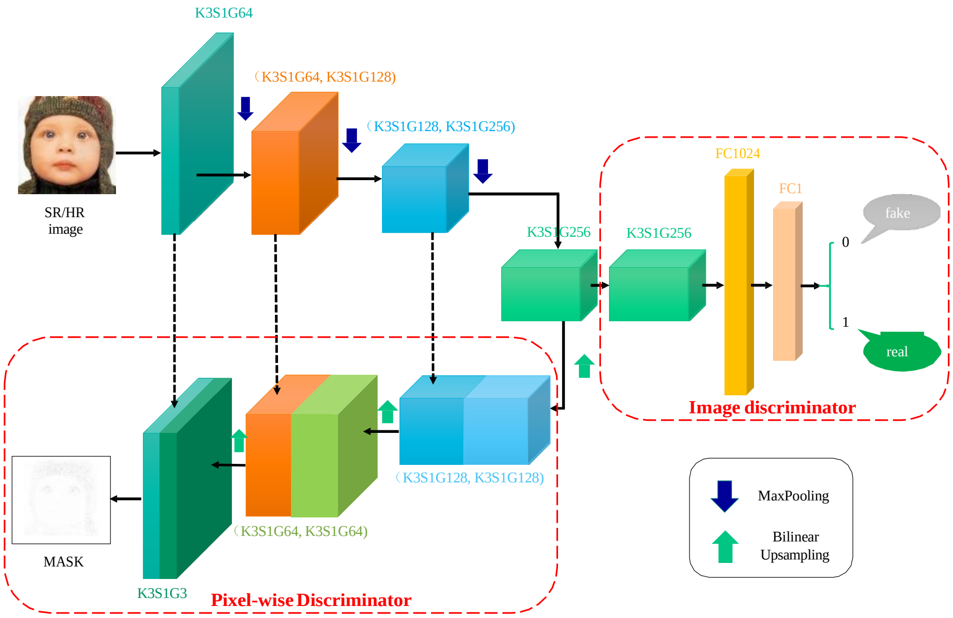

Fig. 2. The discriminator architecture of FASRGAN, where K, S, G represent the kernel size, the stride, and the filter number of the Conv layer, respectively. FC stands for fully connected layer. The mask is a score map among $[0, 1]$, donating the difficulty of reconstruction of each pixel in the image.

function and σ is the sigmoid function.

Decoder. We exploit an upsampling layer to gradually extend the spatial size of feature maps, followed by two convolutional layers to extract more information. To make full use of features, we concatenate the previous outputs from the encoder, which have the same spatial size as current ones. As shown in Fig. 2, the feature maps at the end of the decoder have the same spatial size as input images. Finally, we use the sigmoid function to produce a score map $M \in R^{Wr \times Hr \times C}$ that provides pixel-wise discrimination between real and fake pixels of an input image. The fine-grained adversarial loss function L_M^D for the discriminator is defined as:

$$L_M^D = \frac{1}{Wr \times Hr \times C} \times \sum_{c=1}^{C} \sum_{w=1}^{Wr} \sum_{h=1}^{Hr} \{\log(1 - M_r(w,h,c)) + \log(M_f(w,h,c))\}, \tag{2}$$

where M_r and M_f represent the score maps of the HR image and the generated SR image, respectively. Finally, the loss function for the discriminator is defined as: $L^D = L_{adv}^D + L_M^D$.

2) Generator Objective Function: In the GAN-based SR methods, the generator is generally used to generate the SR images from the LR images. ESRGAN [10] introduced Residual-in-Residual Dense Block (RRDB) without batch normalization as the basic network building unit, which is of higher capacity and easier to train compared with the ResBlock in SRGAN [8]. In this paper, we also adopt RRDB to construct our generator for a fair comparison with ESRGAN. The generator is trained by several losses, defined as following:

Content Loss. Following [5, 17, 22, 34], we use an L_1 loss function to constrain the content of a generated SR image to be close to the HR image. The loss is defined in Eq.3.

$$L_1 = \frac{1}{Wr \times Hr \times C} \times \sum_{w=1}^{Wr} \sum_{h=1}^{Hr} \sum_{c=1}^{C} \| F_\theta^G(I_i^{LR})(w,h,c) - I_i^{HR}(w,h,c) \|_1, \tag{3}$$

where $F_\theta^G(\cdot)$ represents the function of the generator, θ is the parameters of the generator and I_i means the i-th image.

Perceptual Loss. The perceptual loss [35] aims to make the SR image close to the corresponding HR image based on high-level features extracted from a pre-trained network. Similar to [8, 10], we consider both the SR and HR images as the input to the pre-trained VGG19 and extract the VGG19-54 layer features. The perceptual loss is defined as:

$$L_{percep} = \| F_\theta^{VGG}(G(I_i^{LR})) - F_\theta^{VGG}(I_i^{HR}) \|_1, \tag{4}$$

where $F_\theta^{VGG}(\cdot)$ is the function of VGG and I_i is the i-th image, $G(\cdot)$ is the function of the generator.

Adversarial Loss. The discriminator contains two outputs, a whole estimation of the entire image and the pixel-wise fine-grained estimations of an input image. The total adversarial loss function for the generator is defined as:

$$L_{adv}^G = L_{entire}^G + L_{fine}^G, \tag{5}$$

As shown in Eq.2, the discriminator tries to distinguish the real and fake image in a fine-grained way, while the generator aims to fool the discriminator. Thus the fine-grained adversarial loss function L_{fine}^G for the generator is the symmetrical form of

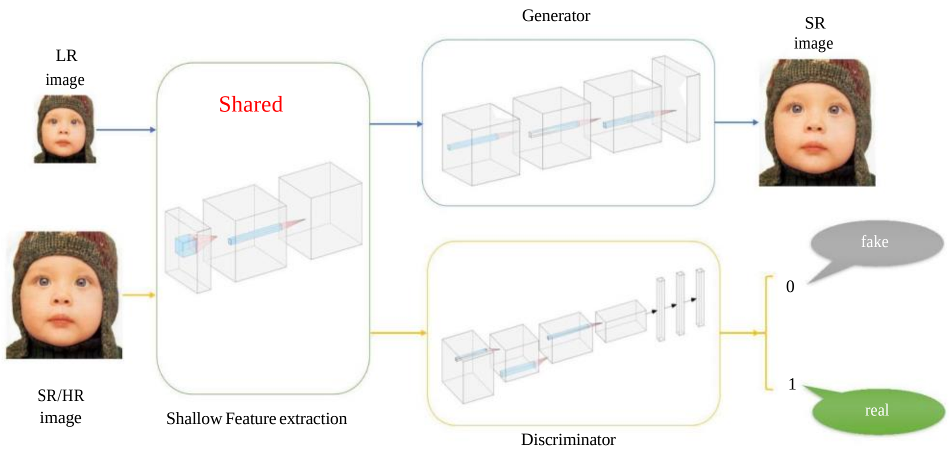

Fig. 3. The illustration of our Feature-sharing Generator Adversarial Networks (Fs-SRGAN). The input sizes of the generator and the discriminator are different. We use a fully Convolutional Neural Network with invariant size of the feature map so that the different input sizes do not matter.

Eq.2:

$$L^G_{fine} = \frac{1}{Wr \times Hr \times C} \times \sum_{c=1}^{C} \sum_{w=1}^{Wr} \sum_{h=1}^{Hr} \{\log(M_r(w,h,c)) + \log(1 - M_f(w,h,c))\}, \tag{6}$$

L^G_{entire} is also the symmetrical form of Eq.1 and defined as:

$$L^G_{entire} = \mathbb{E}[\log(\sigma(C(x_r) - \mathbb{E}[C(x_f)]))] + \mathbb{E}[\log(1 - \sigma(C(x_f) - \mathbb{E}[C(x_r)]))]. \tag{7}$$

Fine-grained Attention Loss Function. The score map generated by the UNet-like discriminator is represented as pixel-wise discrimination scores of an input image, with values M (w, h, c) among $[0, 1]$. A higher score means the corresponding pixel of the input image is closer to that of the ground-truth image. In this manner, the score map can indicate which parts of an image are more difficult to generate and which parts are easier. For instance, the structure background part of an image is sometimes simpler, and thus it would expect the discriminator reflects this to the generator when updating the generator. In other words, the part with lower scores (more difficult to generate) should receive more attention when updating the generator. As a result, we incorporate the score map as the fine-grained attention mechanism into a L_1 loss function, constituting a weighted attention loss function:

$$L_{attention} = \frac{1}{Wr \times Hr \times C} \sum_{w=1}^{Wr} \sum_{h=1}^{Hr} \sum_{c=1}^{C} (1 - M_f(w,h,c)) \times \| F^G_\theta(I^{LR}_i)(w,h,c) - I^{HR}_i(w,h,c) \|_1, \tag{8}$$

where $M_f(w, h, c)$ is the score map of the generated image given by the discriminator. Instead of treating every pixel of an image equally, $L_{attention}$ contributes to pay more attention in the hard-to-recovered part of an image, such as the textures with rich semantic information.

Combining the above losses with different weights, the total loss of the generator is:

$$L^G = L_{percep} + \lambda_1 L^G_{adv} + \lambda_2 L_{attention} + \lambda_3 L_1, \tag{9}$$

where λ_1, λ_2, λ_3 are the coefficients to balance different loss terms.

C. Feature-sharing Generator Adversarial Networks

In the standard GANs, the generator and the discriminator are usually defined as two independent networks. Based on the observation that the low-level parts of the two networks always extract low-level textures such as edges and corners, we propose a new network structure (Fs-SRGAN) to enable low-level feature sharing between the generator and the discriminator. This can reduce the number of parameters and help both networks extract more effective features. Consequently, our Fs-SRGAN contains three parts: a shared feature extractor, a generator, and a discriminator, as shown in Fig. 3.

1) Shared Feature Extractor: We first use a share feature extractor to transform an input image from color space to feature space, before extracting low-level feature maps. The feature-sharing mechanism allows the generator and the discriminator to jointly optimize the low-level feature extractor. Similar to FASRGAN, we adopt RRDB, the basic block of ESRGAN [10], as the basic structure. The shared feature extractor contains E RRDBs to extract helpful feature maps for both the generator and the discriminator, described as following:

$$H_{shared} = F_{shared}(x), \tag{10}$$

where H_{shared} is the low-level feature maps extracted by the shared part, F_{shared} represents the function of the shared feature extractor, and x is the input. For the generator, the input is an LR image, while for the discriminator it is a SR image or a HR image. Considering the input sizes of the generator and the discriminator are different, we apply a fully Convolutional Neural Network with invariant size of feature maps to extract features so that the different input sizes do not matter.

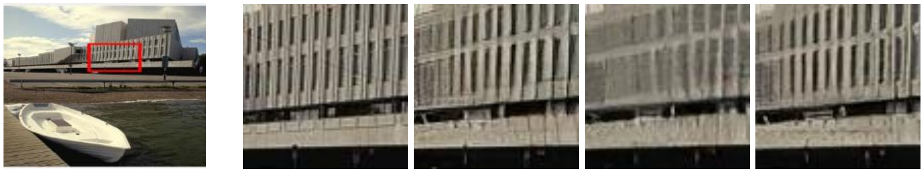

Fig. 4. A visual comparison between the state-of-the-art perceptual image SR methods and FASRGAN (ours) for 4× SR.

2) The Generator and the Discriminator: The rest parts of the generator and the discriminator are the same as those in standard GAN-based methods, except that the inputs are feature maps instead of images as shown in Fig.3.

Generator. The generator contains three parts: low-level feature extraction, deep feature extraction, and reconstruction, which are used for transforming the input image to the feature space from the color space and extracting low-level information, extracting high-level semantic features and reconstructing SR image, respectively. Note that the generator in our Fs-SRGAN only contains the latter two parts. Similar to the shared low-level feature extraction, we adopt RRDB as the basic part of deep feature extraction, except that more RRDBs are used to increase the depth of the network with the purpose of extracting more high-frequency feature for reconstruction.

The reconstruction part utilizes an upsampling layer to upscale the high-level feature maps and a Conv layer to reconstruct an SR image. The loss function of the generator is same as that of ESRGAN [10], which includes perceptual loss, adversarial loss, and pixel-based loss:

$$L^G = L_{percep} + \lambda_1 L^G_{adv} + \lambda_2 L_1, \tag{11}$$

where λ_1, λ_2 are the coefficients to balance different loss terms, L_{percep} and L_1 are defined in Eq.4 and Eq.3, respectively, L^G_{adv} is the adversarial loss with the same definition as L^G_{entire} in Eq.7.

Discriminator. Because the discriminator is a classification network that distinguishes the input as SR or HR image, we apply a structure similar to the VGG network as the discriminator. To reduce information loss, we substitute the pooling layer (used in the encoder of the UNet-discriminator) for a Conv layer with a stride of 2 to decrease the size of feature map. At the tail of the discriminator, we use a Conv layer to transform the feature map into a one-dimensional vector, then use two fully connected layers to output the classification score s among $[0, 1]$. The value of s closer to 1 means more real, otherwise more fake. The loss function of the discriminator is same as L^D_{adv} defined in Eq.1.

IV. Experimental Results

In this section, we first describe our model training details, then provide quantitative and visual comparisons with several state-of-the-art methods on benchmark datasets for our two proposed novel methods, FASRGAN and Fs-SRGAN. We further combine the fine-grained attention and the feature-sharing mechanisms into one single model, termed FA+Fs-SRGAN.

A. Training Details

In training, we use the training set from DIV2K [36] as the training set to train our models. The LR images are obtained by bicubic downsampling (BI) from the source high-resolution images. Images are augmented by rotating and flipping. We also randomly crop 48×48 patches from LR images as the input of the network. Our networks are optimized with the ADAM optimizer [37]. The hyper-parameters β_1 and β_2 in the ADAM optimizer are set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The batch size is set to 16. The generator is pre-trained by the L_1 loss function, followed by generator and the discriminator training with the corresponding loss functions. Following [5, 15, 17, 22, 38], the initial learning rate is set to 1×10^{-4}, and then decays to half every 2×10^{5} iterations. In FASRGAN, the coefficients in Eq.9 are set as $\lambda_1 = 5e\text{-}3$, $\lambda_2 = 1e\text{-}2$ and $\lambda_3 = 1e\text{-}2$. Similar to ESRGAN [10], the number of RRDBs in the generator is set as 23. In Fs-SRGAN, we set the number of RRDBs in the shared feature extractor as $E = 1$, and in the deep feature extractor as 16. The coefficients in Eq.11 are set as $\lambda_1 = 5e\text{-}3$ and $\lambda_2 = 1e\text{-}2$. In FA+Fs-SRGAN, the number of RRDBs in the share part is set as 2, while in the deep feature extraction part is 15. The discriminator and the coefficients of the loss function are the same as those of FASRGAN. We implement our models with the PyTorch [39] framework on two NVIDIA GeForce RTX 2080Ti GPUs.

B. Datasets and Evaluation Metrics

In the testing phase, we use seven standard benchmark datasets to evaluate the performance of our proposed methods: Set5 [40], Set14 [41], BSD100 [42], Urban100 [43], Manga109 [44], DIV2K validation [36], PIRM validation and test dataset[45]. Blau *et al.* [46] proved mathematically that perceptual quality is not always improved with the increase of PSNR value and there is a trade-off between average distortion and perceptual quality. Hence, we not only use PSNR and SSIM [28] to measure the reconstruction accuracy, but also adopt the learned perceptual image patch similarity (LPIPS) [47] and perceptual index (PI) [45] to evaluate the perceptual quality of SR images. LPIPS firstly adopts a pre-trained network F to extract patches y, y_0 from the reference and target images x, x_0. The network F computes the activations of the image patches, each is scaled by a learned weight w_l and then sums up the L_2 distances across all layers. Finally, it computes a perceptual real/fake prediction as follows:

$$d(x, x_0) = \sum_{l} \frac{1}{H_l W_l} \sum_{h,w} \| w_l \odot (y_l^{hw} - \hat{y}_{0l}^{hw}) \|_2^2, \tag{12}$$

TABLE I
QUANTITATIVE RESULTS WITH THE BICUBIC DEGRADATION MODEL FOR 4× SR. BEST AND SECOND BEST RESULTS ARE HIGHLIGHTED AND UNDERLINED, RESPECTIVELY.

Dataset	Metric	EnhanceNet [29]	SRGAN [8]	ESRGAN [10]	RankSRGAN [33]	FASRGAN (ours)	Fs-SRGAN (ours)	FA+Fs-SRGAN (ours)
Set5	PSNR	28.57	29.91	**30.46**	29.73	30.15	30.28	30.19
	SSIM	0.8102	0.8510	0.8515	0.8398	0.8450	**0.8588**	0.8571
	PI	**2.8466**	3.4322	3.5463	2.9867	3.1685	3.9143	3.7455
	LPIPS	0.0488	0.0389	0.0350	0.0348	**0.0325**	0.0330	0.0344
Urban100	PSNR	23.54	24.39	24.36	24.49	24.51	24.55	**24.67**
	SSIM	0.6933	0.7309	0.7341	0.7319	0.7380	**0.7509**	0.7466
	PI	3.4543	3.4814	3.7312	**3.3253**	3.5173	3.5940	3.5819
	LPIPS	0.0777	0.0693	0.0591	0.0667	**0.0588**	0.0591	0.0625
BSD100	PSNR	24.94	25.50	25.32	25.51	25.41	25.61	**25.87**
	SSIM	0.6266	0.6528	0.6514	0.6530	0.6523	0.6726	**0.6747**
	PI	2.8467	2.3054	2.4150	**2.0768**	2.2783	2.4056	2.3749
	LPIPS	0.0982	0.0887	0.0798	0.0850	**0.0796**	0.0801	0.0855
DIV2K val	PSNR	27.28	28.16	28.17	28.10	28.15	28.15	**28.23**
	SSIM	0.7460	0.7753	0.7759	0.7710	0.7768	**0.7903**	0.7891
	PI	3.4953	3.1619	3.2572	**3.0130**	3.1034	3.3303	3.3092
	LPIPS	0.0753	0.0605	0.0550	0.0576	**0.0539**	0.0542	0.0576
PIRM val	PSNR	25.47	25.61	25.18	25.65	25.38	25.75	**26.00**
	SSIM	0.6569	0.6757	0.6596	0.6726	0.6648	0.6907	**0.6934**
	PI	2.6762	2.2254	2.5548	**2.0183**	2.2476	2.3311	2.2482
	LPIPS	0.0838	0.0718	0.0714	0.0675	0.0685	**0.0651**	0.0677

where $y_l, y_{0l} \in \mathrm{R}^{H_l \times W_l \times C_l}$ represent the reference or target

are regarded as the reference images, the SR images generated by our methods or the compared methods as the target images. We use the public codes and pre-trained network (AlexNet from version 0.0) for evaluation. While PI is based on the

NIQE [49]: PI=$\frac{1}{2}$((10-Ma)+NIQE). PSNR and SSIM are calculated on the luminance channel in the YCbCr color space. We also use LPIPS and root mean square error (RMSE) to measure the trade-off between perceptual quality and reconstruction accuracy. Using LPIPS/RMSE rather than LPIPS/PSNR to evaluate the trade-off is for better observation, where lower LPIPS/RMSE means a better result. Higher PSNR/SSIM and lower RMSE mean better results in reconstruction accuracy, while lower scores of LPIPS/PI imply that the images are more similar to the HR images.

As shown in Fig. 4, the SR image of our FASRGAN has less artifacts than that of ESRGAN [10] and is clearer than that of RankSRGAN [33]. But the PI value of the SR image produced by RankSRGAN [33] is lower than that of our FASRGAN, and even lower than that of the original HR image. In terms of LPIPS, our method attains the lowest value, which is more consistent with human observation. Hence, we use LPIPS as our first perceptual quality metric and PI as the second one.

C. Quantitative Comparisons

We present the quantitative comparisons between our methods and the state-of-the-art perceptual image SR methods on several benchmark datasets. As shown in Table I, in most cases, RankSRGAN [33] obtains the lowest PI values among these methods, benefiting from using the loss function with the newly added ranker to optimize the generator. However, our FASRGAN obtains the best LPIPS on most datasets, and both the LPIPS and PI values are better than that of ESRGAN [10], whose structure of the generator is the same as ours. It demonstrates that the fine-grained attention in our FASRGAN can transfer more information to the generator to produce better results. With less RRDBs in the generator, our Fs-SRGAN obtains best SSIM results and comparable, sometimes even better LPIPS results than those of ESRGAN [10] and FASRGAN. In other words, our Fs-SRGAN extracts features more effectively and efficiently, benefiting from the feature-sharing mechanism. The combined model FA+Fs-SRGAN obtains the highest PSNR except for Set5, indicating that it can recover more contents in the SR images.

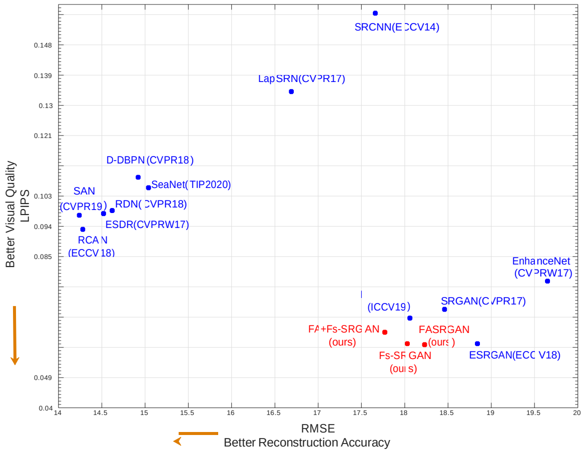

Fig. 5. The trade-off of RMSE and LPIPS on Urban100 of our methods and the state-of-the-art methods for 4× super-resolution.

We also compare our methods with the state-of-the-art methods on the trade-off between reconstruction accuracy and visual quality. The results are shown in Fig. 5. Methods in the top-left part are almost MSE-based with low RMSE and high LPIPS scores due to the over-smoothed edges and lack of high-frequency details. The bottom-right category includes the

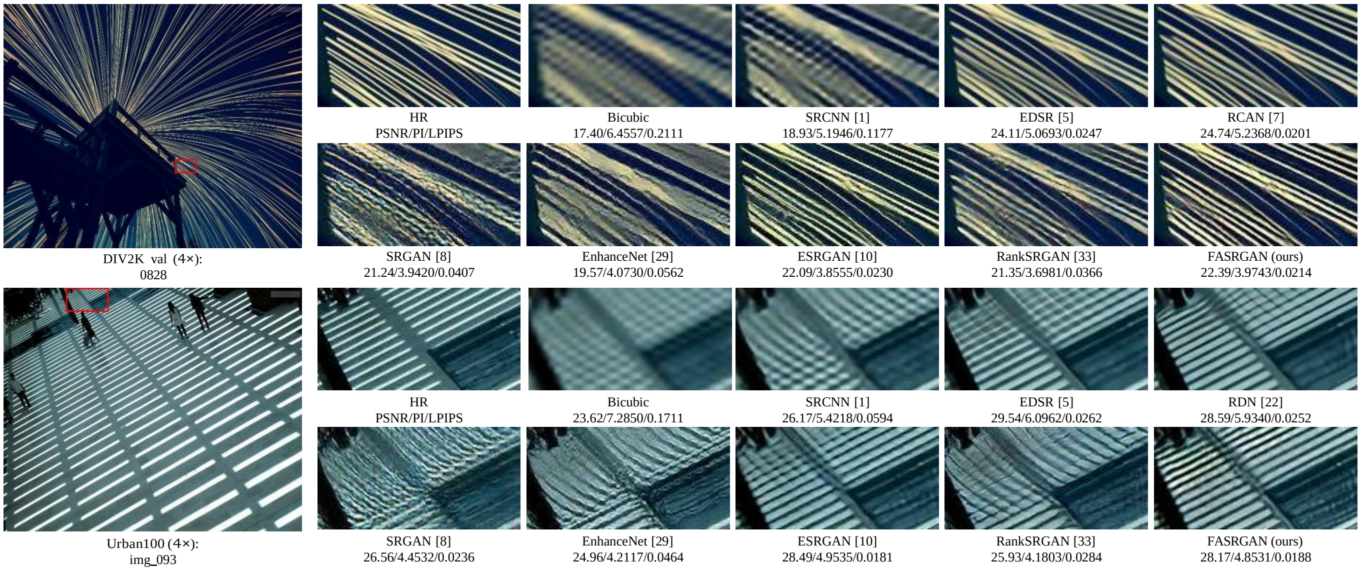

Fig. 6. The visual comparisons between FASRGAN and the state-of-the-art SR methods for 4× super-resolution.

GAN-based methods, such as SRGAN [8], EnhanceNet [29], ESRGAN [10], RankSRGAN [33], and our methods. These methods usually gain high-visual quality images even if their RMSE values are larger than those of the MSE-based methods. Our FASRGAN gets better visual quality and reconstruction accuracy compared with EnhanceNet, SRGAN and ESRGAN, and lower LPIPS than RankSRGAN. Our Fs-SRGAN attains comparable LPIPS with ESRGAN but lower RMSE, and better visual quality and reconstruction accuracy than RankSRGAN. The combined model FA+Fs-SRGAN obtains the lowest RMSE among the GAN-based methods. These demonstrate the effectiveness of our fine-grained attention and feature-sharing mechanism.

To further demonstrate the effectiveness of our FASRGAN and Fs-SRGAN, we conduct a user study to calculate the Mean Opinion Score (MOS) [50] against the state-of-the-art SR methods, i.e. SRGAN [8], ESRGAN [10] and RankSRGAN [33]. Ten candidates are shown with a side-by-side comparison of the generated SR image and the corresponding ground-truth. They are then asked to evaluate the difference of the two images on a 5-level scale defined as: 0 - ‘almost identical’, 1 - ‘mostly similar’, 2 - ‘similar’, 3 - ‘somewhat similar’ and 4 - ‘mostly different’. We randomly select 10 images from PIRM val dataset [45], and invite 10 participants to give a score on each image according to the 5-level scale. For a better comparison, one small patch from the image is zoomed in. The average scores of all images are considered as the final results. As suggested in Table II, our FASRGAN and Fs-SRGAN achieve better performance than all the compared methods, proving the effectiveness of our proposed fine-grained attention and feature-sharing mechanism.

D. Qualitative Results

We compare our final models on several public benchmark datasets with the state-of-the-art MSE-based methods: SRCNN [1], EDSR [5], RDN [22], RCAN [7], and GAN-based approaches: SRGAN [8], EnhanceNet [29], ESRGAN [10], RankSRGAN [33].

TABLE II
THE COMPARISON OF LPIPS AND MOS BETWEEN OUR METHODS AND THE STATE-OF-THE-ART METHODS ON PIRM VAL, WHERE THE LOWER VALUES MEAN MORE SIMILAR WITH THE HR IMAGE. THE LPIPS IS TESTED ON THE WHOLE DATASET, WHILE MOS IS CALCULATED ON 10 RANDOMLY SELECTED IMAGES.

Methods	PIRM Val	
	LPIPS	Mos
SRGAN [8]	0.0718	1.98
ESRGAN [10]	0.0714	1.88
RankSRGAN [33]	0.0675	1.84
FASRGAN (ours)	0.0685	1.46
Fs-SRGAN (ours)	0.0651	1.46

1) Visual Comparisons of FASRGAN: Some representative quality results are presented in Fig. 6. PSNR , PI and LPIPS are also provided for reference.

As shown in Fig. 6, our proposed FASRGAN outperforms previous methods by a large margin. Images generated by FASRGAN contain more fine-grained textures and details. For example, the cropped parts of image ‘0828’ are full of textures. All the compared MSE-based methods suffer from heavy blurry artifacts, failing to recover the structure and the gap of the stripes. SRGAN, EnhanceNet, ESRGAN and RankSRGAN generate high-frequency noise and wrong textures; while our FASRGAN can reduce noise and recover them more correctly, producing more faithful results and being closer to the HR images. For image ‘img 093’ in Urban100, the cropped parts of the images generated by the compared methods contain heavily blurry artifacts and lines with wrong directions. Although the LPIPS of ESRGAN is a little lower than our FASRGAN, our FASRGAN can alleviate the artifacts better and recover zebra crossing in the right direction. More results can be seen in the supplemental material. These comparisons demonstrate the strong ability of FASRGAN for producing more photo-realistic and high-quality SR images.

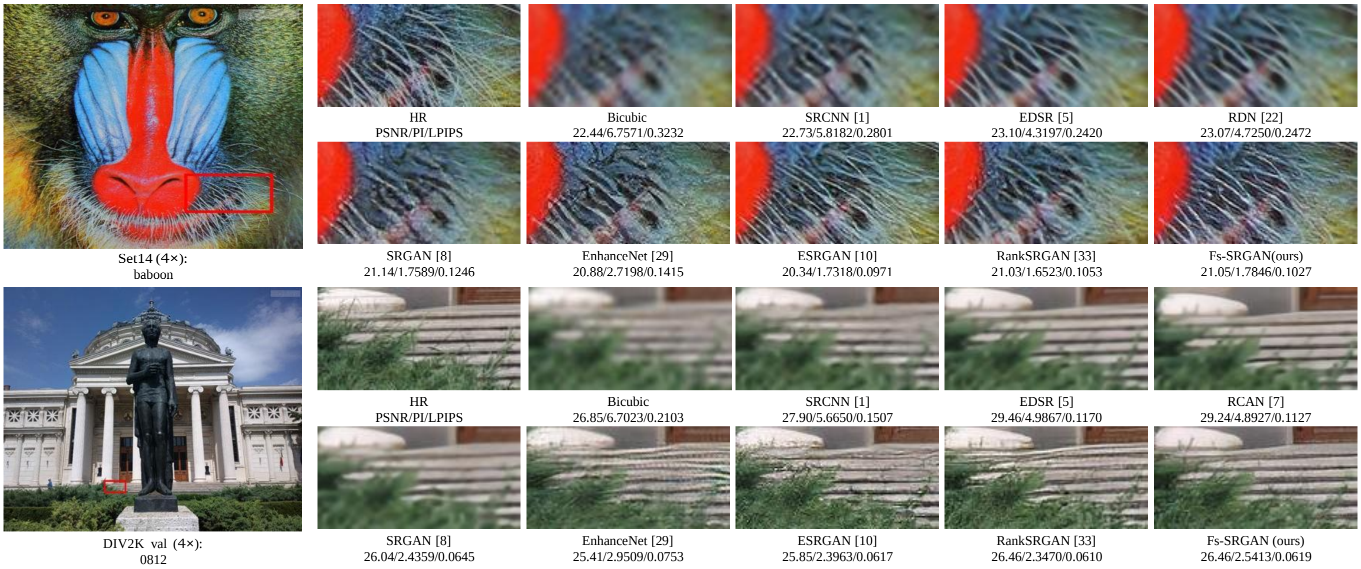

Fig. 7. The visual comparisons between Fs-SRGAN and the state-of-the-art SR methods for 4×.

2) *Visual Comparisons of Fs-SRGAN:* We also compare our Fs-SRGAN with state-of-the-art methods in Fig. 7. Our Fs-SRGAN obtains better performance than other methods in producing SR images, in terms of sharpness and details. For image ‘baboon’, the cropped parts of the images generated by the MSE-based methods are over-smoothed. Previous GAN-based methods not only fail to produce clear whiskers but also introduce lots of unpleasing noise. Despite having lower LPIPS value, ESRGAN generates too many whiskers, which have not appeared in the original HR image. While our Fs-SRGAN produces more correct whiskers. For image ‘0812’, MSE-based methods still suffer from heavy blurry artifacts and generate unnatural results. GAN-based methods cannot maintain the structures of the stairs or the train tracks and introduce artifacts. Our proposed Fs-SRGAN outperforms the compared methods, reducing the artifacts and recovering the correct textures. More results can be seen in the supplemental material. These also indicate that the shared low-level feature extractor of the generator and the discriminator is beneficial.

3) *Visual Comparisons of FA+Fs-SRGAN:* We further present the visual results of our FA+Fs-SRGAN compared with ESRGAN [10], FASRGAN and Fs-SRGAN. As shown in Fig. 8, for image ‘57’ and ’OhWareraRettouSeitokai’, the results from FA+Fs-SRGAN are better than those of FASRGAN, and contain more correct textures. The PSNR and LPIPS values are both the best for FA+Fs-SRGAN. More results can be seen in the supplemental material. These results illustrate that the combined method can restore more contents for the SR images and obtains comparable or even better visual results compared with FASRGAN and Fs-SRGAN.

E. Model Analysis

This section compares the sizes and the time complexity of the generators between our methods and ESRGAN [10], which use RRDB as the basic block to construct the generators. We are not comparing our methods with SRGAN [8] and RankSRGAN [33] in these aspects as clearly they use 16 Resblocks to build their generators, so their parameters and inference time are less than ours.

In the aspect of the numbers of parameters, both ESRGAN and our FASRGAN have 23 RRDBs and 16.7M parameters, while our Fs-SRGAN and FA+Fs-SRGAN have 17 RRDBs and 12.46M parameters in their generators.

In the aspect of inference time, we run our models and the public official test code and model from ESRGAN on Urban100 using a machine with 4.2GHz Inter i7 CPU (32G RAM) and Nvidia RTX 2080 platform. We conduct five times of inference on Urban100 and take the mean as the inference time.

Our Fs-SRGAN and FA+Fs-SRGAN run much faster than ESRGAN, where Fs-SRGAN has the average time of 0.1377 seconds and FA+Fs-SRGAN 0.1364 seconds, while ESRGAN has 0.3573 seconds. Even our FASRGAN runs a little faster than ESRGAN, with average time 0.3160 seconds. From Table I we can see that our Fs-SRGAN has comparable or even better results than ESRGAN, which demonstrates the efficiency of our feature-sharing mechanism.

Fig. 9 plots the curves of PI values in the training process of our proposed methods on Set14. We observe that the training process of FASRGAN is more stable and the PI value is the lowest. The average PI value of Fs-SRGAN is higher than FASRGAN. As described in [10], the deep model has a stronger representation capacity to capture semantic information and reduce unpleasing noises. And as mentioned above, Fs-SRGAN contains fewer RRDBs than FASRGAN. Hence, we speculate that compared with FASRGAN, Fs-SRGAN with fewer RRDBs captures less information for reconstruction but brings more noises, causing higher PI values. FA+Fs-SRGAN, which combines the fine-grained attention mechanism into Fs-SRGAN, obtains the lower PI values than Fs-SRGAN, which demonstrates the effectiveness of our fine-grained attention mechanism. However, the training of the FA+Fs-SRGAN is not stable, which is the concern we need to focus on in our future work.

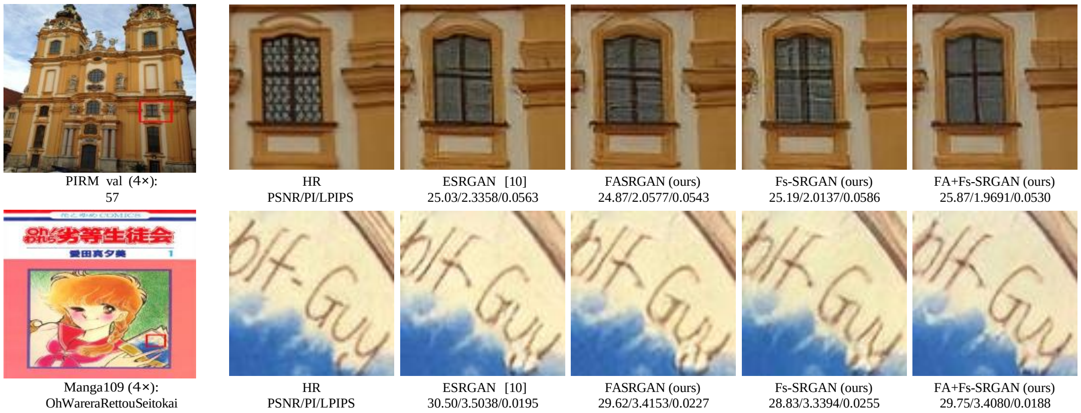

Fig. 8. A visual results of FA+Fs-SRGAN for x4 magnification.

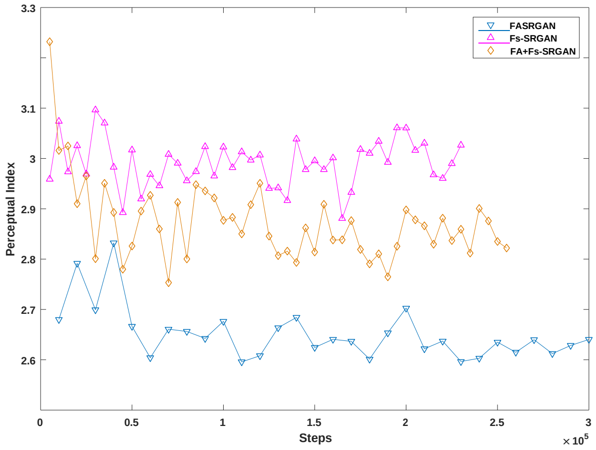

Fig. 9. The changes of average PI on Set14 during the training process for 4× super-resolution.

TABLE III
THE RESULT OF OBJECT RECOGNITION BETWEEN OUR METHODS AND THE STATE-OF-THE-ART METHODS FOR 4× SR. THE BASELINE USES THE ORIGINAL HR IMAGE AS THE INPUT OF RESNET-50 MODEL.

Evaluation	Top-1 error	Top-5 error
Bicubic	0.526	0.277
SRCNN [1]	0.464	0.230
FSRCNN [13]	0.488	0.252
SRGAN [8]	0.410	0.191
EnhanceNet [29]	0.454	0.224
ESRGAN [10]	0.334	0.132
RankSRGAN [33]	0.342	0.136
Fs-SRGAN (ours)	0.338	0.136
FA+Fs-SRGAN (ours)	0.337	0.134
FASRGAN (ours)	0.323	0.124
Baseline	0.241	0.071

F. Object Recognition Performance

To further demonstrate the quality of our generated SR images, we treat them as a pre-processing step for object recognition.We use the same setting as EnhanceNet and evaluate the object recognition performance with the generated images by our methods and other state-of-the-art methods: SRCNN [1], FSRCNN [13], SRGAN [8], EnhanceNet [29], ESRGAN [10], RankSRGAN [33].

We use the pre-trained ResNet-50 on imageNet as an evaluation model and fetch the first 1000 images in ImageNet CLS-LOC validation dataset for evaluation. The test images are first down-sampled by bicubic and then upscaled by our methods and the compared methods. These SR images are then used as inputs to the ResNet-50 model to calculate their Top-1 and Top-5 errors for evaluation. As shown in Table III, both two methods we proposed and the variant FA+Fs-SRGAN achieve considerable accuracy compared to the state-of-the-art methods. Among these three methods, FASRGAN achieves the lowest Top-1 and Top-5 errors, Fs-SRGAN and FA+Fs-SRGAN obtain comparable results with ESRGAN, demonstrating the effectiveness of both the fine-grained attention and the feature-sharing mechanisms.

TABLE IV
THE ABLATION STUDY OF FINE-GRAINED ATTENTION (FA) AND FEATURE-SHARING (FS) MECHANISMS FOR 4× SUPER-RESOLUTION.

Model		FA mechanism		Fs mechanism	
		w/o FA	FASRGAN	w/o Fs	Fs-SRGAN
PIRM Test	PSNR	25.04	25.26	25.44	25.69
	SSIM	0.6454	0.6523	0.6626	0.6785
	PI	2.4251	2.1160	2.1420	2.2279
	LPIPS	0.0751	0.0718	0.0731	0.0695

G. Ablation Study

To study the effects of the two mechanisms in the proposed methods, we conduct ablation experiments by removing the mechanisms and test the differences, respectively. The quantitative results are illustrated in Table IV, overall visual comparisons are presented in Fig. 10, Fig. 11 and the supplemental material. A detailed discussion is provided as follows.

1) Removing the Fine-grained Attention Mechanism: We first remove the fine-grained attention (FA) mechanism in FASRGAN. The attention item is removed from the loss functions of the generator in the model without FA. The coefficients of Eq.9 are set as $\lambda_1 = 5e$-3, $\lambda_2 = 0$ and $\lambda_3 = 1e$-2. The fine-grained adversarial loss functions, L_M^D and L_{fine}^G are also removed. The generators of FASRGAN and the model without FA have the same parameters, and the difference lies in the loss function in training.

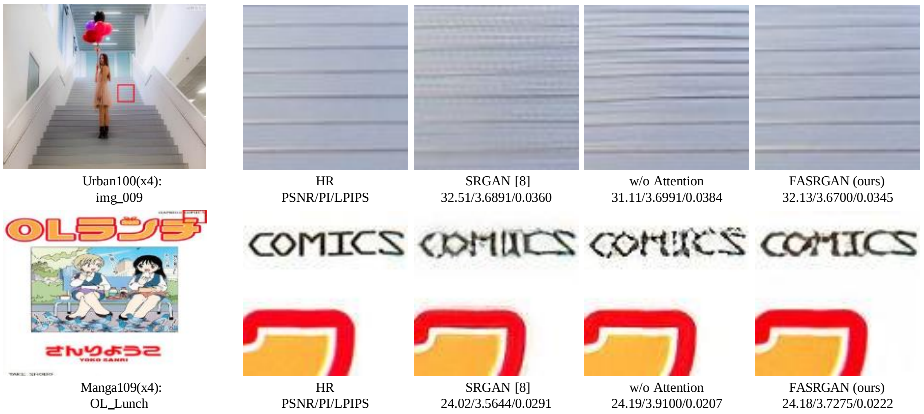

Fig. 10. The visual results of ablation study of FASRGAN 4× SR.

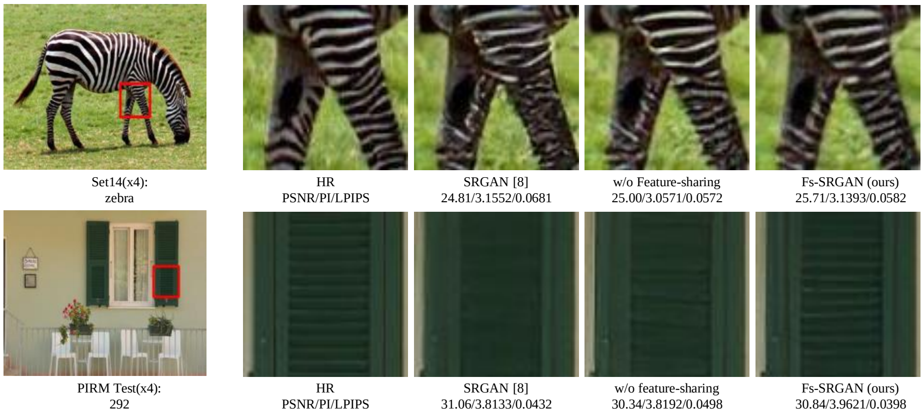

Fig. 11. The visual results of ablation study of Fs-SRGAN for 4× SR.

From Table IV we can see that FASRGAN surpasses the model without FA in all metrics. An obvious performance decrease can be observed in Fig. 10. For image ‘img 009’, the model without FA mechanism introduces some unnatural noises and undesired edges, while FASRGAN can maintain the structure and produce high-quality SR images. For image ‘OL_Lunch’, the result from the model without FA mechanism contains more artifacts and noise and the letters cannot be well recognized, while FASRGAN reduces the artifacts and noises, whose result looks closer to the original HR images. The visual analysis indicates the effectiveness of the FA mechanism in removing unpleasant and unnatural artifacts.

2) Removing the Feature-sharing Mechanism: We remove the feature-sharing (Fs) mechanism, so that the generator and discriminator extract their low-level features separately, but the loss function keeps the same as that of Fs-SRGAN. The discriminator and the generator in our Fs-SRGAN use a shared RRDB to extract low-level features, while in the case the Fs mechanism is removed, different RRDBs are used to extract low-level features for them individually.

Table IV shows that Fs-SRGAN has lower LPIPS and higher PSNR/SSIM than the model without Fs mechanism. Fig. 11 presents the results of the model without Fs mechanism and Fs-SRGAN. We can observe that Fs-SRGAN outperforms the model without Fs mechanism by a large margin. The removal of Fs mechanism tends to introduce unpleasant artifacts. For image ‘zebra’, by employing the Fs mechanism, Fs-SRGAN can alleviate heavy artifacts and noises, recovering the strips of legs more clearly and correctly. For image ‘292’, our Fs-SRGAN generates more textures of the pane. The above results illustrate the effectiveness of our Fs mechanism.

3) Feature Visualization of Fine-Grained Attention and Feature-sharing Mechanisms: To further verify the effectiveness of our proposed fine-grained attention and feature-sharing mechanisms, we present feature visualizations of the first RRDB of the generator (FASRGAN) and the shared feature extraction part (Fs-SRGAN) in Fig.12. FASRGAN reduces the noises in the image ‘img 063’ and extracts more texture information compared with the model without attention mechanism. The feature maps from Fs-SRGAN also contain more helpful textures, showing that our proposed methods help the networks extract more useful information.

4) The Block Number of the Feature-sharing Part: To further study the effect of the depth of the shared feature extractor in Fs-SRGAN, we vary the number of RRDBs in both the shared low-level feature extractor and the deep feature extractor, keeping the total number of RRDBs in the generator unchanged. As shown in Fig. 13, increasing the number of the shared feature extractor leads to performance reduction and increases the burden of the discriminator due to more parameters which makes the model difficult to train. Among them, E1G16 obtains the best results in both visual quality and reconstruction accuracy.

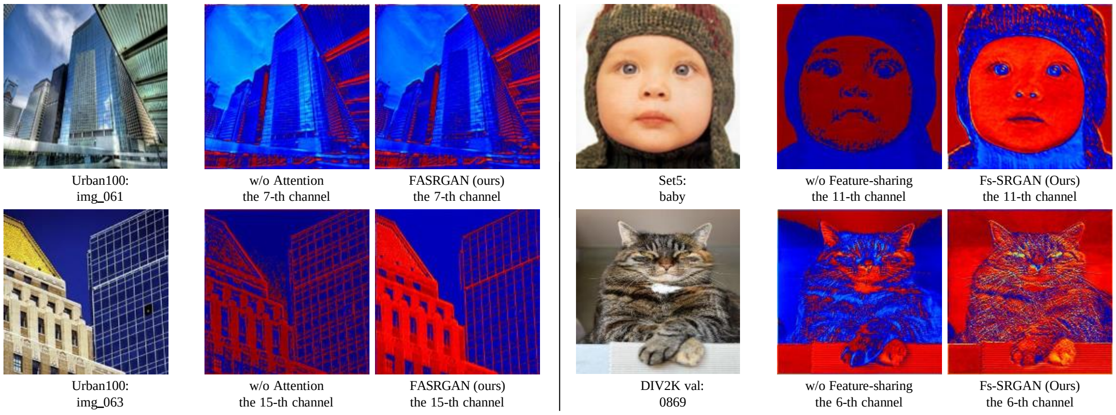

Fig. 12. The feature visualization of the first RRDB of the generators in FASRGAN and the model w/o attention, and of the shared low-level feature extractor in Fs-SRGAN and the first RRDBs of the generator in the model w/o feature-sharing.

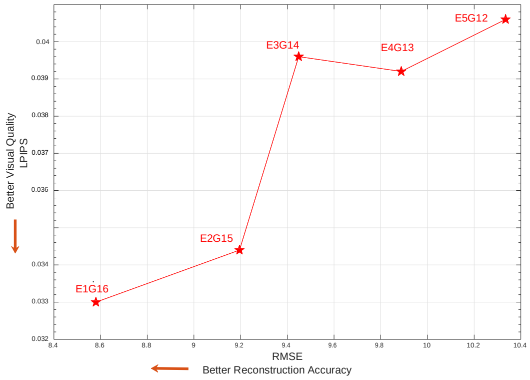

Fig. 13. The change of the number of RRDBs in shared low-level feature extractor (E) in Fs-SRGAN. G represents the number of RRDBs in the deep feature extraction part. The test is conducted on Set5 for 4× super-resolution.

TABLE V
THE ABLATION STUDY OF THE COEFFICIENT λ_2 OF $L_{attention}$ IN EQ.8

Dataset	Metric	$\lambda_2 = 0.05$	$\lambda_2 = 0.01$	$\lambda_2 = 0.005$
Urban100	PSNR	24.35	24.51	24.31
	SSIM	0.7359	0.7380	0.7364
	PI	3.5091	3.5173	3.5284
	LPIPS	0.0608	0.0588	0.0613
DIV2K val	PSNR	28.16	28.15	28.19
	SSIM	0.7771	0.7903	0.7803
	PI	3.1826	3.3303	3.2378
	LPIPS	0.0547	0.0542	0.0560

5) Coefficient of the Fine-Grained Attention Loss in the FASRGAN Generator: We also conduct an ablation study to verify the influences of different coefficient λ_2 of the fine-grained attention loss $L_{attention}$ in the generator of FASRGAN. We set λ_2 as 0.05, 0.01 and 0.005, while the other settings are kept the same. As shown in Table V, the model with $\lambda_2 = 0.01$ has the best performance in SSIM and LPIPS on Urban100 and DIV2K val, and achieves comparable results in PSNR and PI. The visual results are shown in the supplemental material. When λ_2 is set too small, the fine-grained feedback from the discriminator has less impact on the generator. And when λ_2 is set too large, the training is unstable for both the generator and discriminator and hard to converge. These results indicate that $\lambda_2 = 0.01$ is a good setting in practice, which is used in our FASRGAN.

TABLE VI
THE EFFECT OF FINE-GRAINED ATTENTION (FA) AND FEATURE-SHARING (FS) MECHANISMS ON SRGAN.

Dataset	Metric	SRGAN [8]	SRGAN_FA	SRGAN_Fs
	PSNR	29.91	29.61	29.66
	SSIM	0.8510	0.8437	0.8541
	PI	3.4322	3.0651	3.4440
	LPIPS	0.0389	0.0341	0.0368
Set14	PSNR	26.56	26.11	26.27
	SSIM	0.7093	0.6977	0.7179
	PI	2.8549	2.7550	2.7705
	LPIPS	0.0696	0.0692	0.0669
Urban100	PSNR	24.39	24.00	24.04
	SSIM	0.7309	0.7205	0.7331
	PI	3.4814	3.4252	3.4818
	LPIPS	0.0693	0.0688	0.0691
BSD100	PSNR	25.50	25.35	25.46
	SSIM	0.6528	0.6506	0.6650
	PI	2.3054	2.2503	2.3348
	LPIPS	0.0887	0.0856	0.0876

6) The Fine-grained Attention and Feature-sharing Mechanisms in SRGAN: To verify whether our proposed fine-grained attention (FA) and feature-sharing mechanisms can improve the performance in other GAN-based SR models, we incorporate these two mechanisms into SRGAN [8], denoting as SRGAN FA and SRGAN Fs respectively. The generator in SRGAN FA is the same as that of SRGAN [8], and the discriminator adopts our proposed UNet-like structure. We use a convolution layer and a residual block (RB) as the shared low-level feature extraction part for the generator and discriminator in SRGAN Fs. The rest part of the generator and the discriminator are similar with that of SRGAN [8], except that the number of RB in the deep feature extraction is 13. Hence, the parameter for SRGAN Fs is 1.48K, and 1.554K for SRGAN and SRGAN FA. As shown in Table VI, SRGAN FA achieves the best performance in terms of PI and LPIPS on most of the test dataset. SRGAN Fs also outperforms SRGAN in SSIM and LPIPS. These results demonstrate that our proposed FA and Fs mechanisms can be well adapted to the SRGAN model.

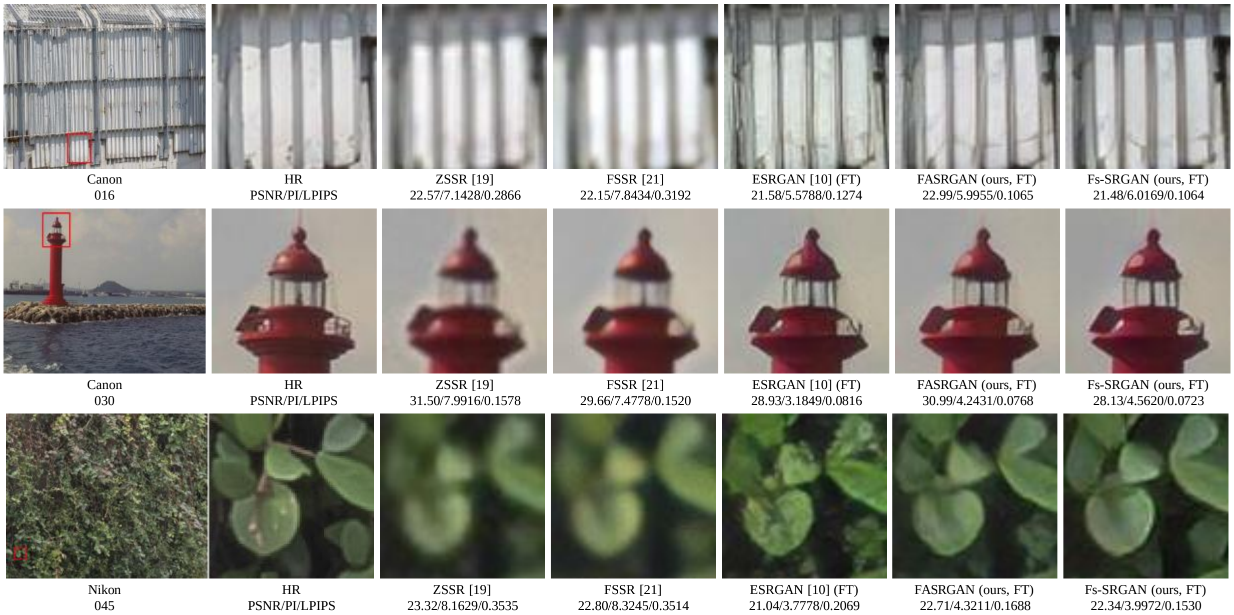

Fig. 14. The visual comparisons between our proposed methods and compared methods on RealSR(V3) test set for 4×. FT represents the model has been fine-tuned on RealSR (v3) training set.

TABLE VII
THE QUANTITATIVE COMPARISON OF OUR METHODS AND OTHER SR METHODS ON REALSR FOR 4× MAGNIFICATION. FT REPRESENTS THE MODEL FINE-TUNES ON REALSR TRAINING DATASET.

Model	RealSR (V3) Test			
	PSNR	SSIM	PI	LPIPS
SRCNN [1]	27.69	0.7808	7.8679	0.2290
RCAN [7]	27.65	0.7803	7.8519	0.2311
ESRGAN [10]	27.57	0.7748	7.4819	0.2215
RankSRGAN [33]	27.56	0.7701	7.0521	0.2100
FASRGAN (ours)	27.57	0.7732	7.1178	0.2111
Fs-SRGAN (ours)	27.47	0.7744	7.3112	0.2151
ZSSR [19]	27.56	0.7719	7.4666	0.2069
FSSR [21]	26.68	0.7773	7.0811	0.1978
ESRGAN [10] (FT)	26.67	0.7378	4.2885	0.1134
FASRGAN (ours, FT)	27.57	0.7809	5.0006	0.1063
Fs-SRGAN (ours, FT)	25.82	0.7663	4.9929	0.1121

H. Results On the Real-World Dataset

We also benchmark our proposed methods on a publicly available real-world dataset to test the robustness. We adopt the test set from RealSR(V3) [51] as the dataset and PSNR/SSIM/PI/LPIPS as evaluation metrics. As shown in the top part of Table VII, our FASRGAN and Fs-SRGAN obtain better PI and LPIPS than ESRGAN and comparable results with RankSRGAN, demonstrating that our proposed models have better robustness on real-world LR images.

In addition, we used the training set from RealSR(V3) to fine-tune ESRGAN and our proposed methods. Both of them have run about 150k iterations, where the learning rate is initially set as 10^{-4} and decays a half every 50k iterations. We test the fine-tuned (FT) models on the test set, and also compare them with ZSSR [19] and the work from Fritsche *et al.* [21] proposed for the AIM 2019 Challenge on Real World, denoted as FSSR. ZSSR is the first unsupervised SR method for non-ideal LR images. The codes and models of FSSR are publicly available, and we adopt the model TDSR of AIM for comparison. As shown in the bottom part of Table VII, our FASRGAN and Fs-SRGAN still obtain better results in LPIPS, indicating that our models are robust on the real-world images.

Visual results of the fine-tuned models and the compared methods are also presented in Fig. 14. We can observe that the results generated by ZSSR and FSSR are heavily blurred, which brings a bad visual effect. The fine-tuned results from ESRGAN contain some artifacts and noises, resulting in unpleasing observation. While our fine-tuned FASRGAN and Fs-SRGAN reduce the artifacts and produce more pleasing results, demonstrating the robustness of our proposed models.

V. CONCLUSION

We propose two GAN-based models, FASRGAN and Fs-SRGAN, for SISR to overcome the limitations of existing methods. FASRAGN introduces a fine-grained attention mechanism into the GAN framework, where the discriminator has two outputs: a score for measuring the quality of the whole input and a fine-grained attention estimation for the input. The fine-grained attention delivers a fine-grained supervisor to the generator to ensure generation of pixel-wise photo-realistic images. The Fs-SRGAN shares the low-level feature extractor of the generator and the discriminator, reducing the number of parameters and improving the reconstruction performance. These two mechanisms are general and could be applied to other GAN-based SR models. Comparisons with other state-of-the-art methods on benchmark datasets demonstrate the effectiveness of our proposed methods.

REFERENCES

[1] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE*

transactions on pattern analysis and machine intelligence, vol. 38, no. 2, pp. 295–307, 2015.

[2] Z. Wang, J. Chen, and S. C. H. Hoi, "Deep learning for image super-resolution: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[3] N. Kumar and A. Sethi, "Fast learning-based single image super-resolution," *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1504–1515, Aug 2016.

[4] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1646–1654.

[5] 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.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 286–301.

[8] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *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.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[10] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *European Conference on Computer Vision*. Springer, 2018, pp. 63–79.

[11] A. Jolicoeur-Martineau, "The relativistic discriminator: a key element missing from standard GAN," in *International Conference on Learning Representations*, 2019, pp. 1–26.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[13] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *European conference on computer vision*. Springer, 2016, pp. 391–407.

[14] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on computer vision and pattern recognition*. IEEE, 2010, pp. 2528–2535.

[15] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[16] J.-H. Kim and J.-S. Lee, "Deep residual network with enhanced upscaling module for super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 800–808.

[17] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5835–5843.

[18] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1664–1673.

[19] A. Shocher, N. Cohen, and M. Irani, "'Zero-shot' super-resolution using deep internal learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3118–3126.

[20] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[21] M. Fritsche, S. Gu, and R. Timofte, "Frequency separation for real-world super-resolution," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 2019, pp. 3599–3608.

[22] 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.

[23] C. Liu, X. Sun, C. Chen, P. L. Rosin, Y. Yan, L. Jin, and X. Peng, "Multi-scale residual hierarchical dense networks for single image super-resolution," *IEEE Access*, vol. 7, pp. 60 572–60 583, 2019.

[24] X. Yang, H. Mei, J. Zhang, K. Xu, B. Yin, Q. Zhang, and X. Wei, "Drfn: Deep recurrent fusion network for single-image super-resolution with large factors," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 328–337, 2018.

[25] F. Fang, J. Li, and T. Zeng, "Soft-edge assisted network for single image super-resolution," *IEEE Transactions on Image Processing*, vol. 29, pp. 4656–4668, 2020.

[26] L. Zhang, P. Wang, C. Shen, L. Liu, W. Wei, Y. Zhang, and A. Van Den Hengel, "Adaptive importance learning for improving lightweight image super-resolution network," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 479–499, 2020.

[27] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 11 065–11 074.

[28] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli

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

[29] M. S. Sajjadi, B. Scholkopf, and M. Hirsch, “Enhancenet: Single image super-resolution through automated texture synthesis,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4491–4500.

[30] R. Mechrez, I. Talmi, F. Shama, and L. Zelnik-Manor, “Maintaining natural image statistics with the contextual loss,” in *Asian Conference on Computer Vision*. Springer, 2018, pp. 427–443.

[31] B. Yan, B. Bare, C. Ma, K. Li, and W. Tan, “Deep objective quality assessment driven single image super-resolution,” *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2957–2971, 2019.

[32] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor, “The 2018 PIRM challenge on perceptual image super-resolution,” in *European Conference on Computer Vision*, 2018, pp. 334–355.

[33] W. Zhang, Y. Liu, C. Dong, and Y. Qiao, “Ranksrgan: Generative adversarial networks with ranker for image super-resolution,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3096–3105.

[34] W. Lai, J. Huang, N. Ahuja, and M. Yang, “Fast and accurate image super-resolution with deep Laplacian pyramid networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–14, Aug 2018.

[35] 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.

[36] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, “NTIRE 2017 challenge on single image super-resolution: Methods and results,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 114–125.

[37] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations*, 2015, pp. 1–15.

[38] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, “Residual non-local attention networks for image restoration,” in *International Conference on Learning Representations*, 2019, pp. 1–18.

[39] 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,” in *Neural Information Processing Systems Workshop*, 2017, pp. 1–4.

[40] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. Alberi Morel, “Low-complexity single-image super-resolution based on nonnegative neighbor embedding,” in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2012, pp. 135.1–135.10.

[41] R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” in *Proceedings of the 7th International Conference on Curves and Surfaces*. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 711–730.

[42] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2. IEEE, 2001, pp. 416–423.

[43] 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.

[44] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, “Sketch-based manga retrieval using manga109 dataset,” *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21 811–21 838, 2017.

[45] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor, “The 2018 pirm challenge on perceptual image super-resolution,” in *The European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[46] Y. Blau and T. Michaeli, “The perception-distortion tradeoff,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6228–6237.

[47] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595.

[48] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a “completely blind” image quality analyzer,” *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2013.

[49] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, “Learning a no-reference quality metric for single-image super-resolution,” *Computer Vision and Image Understanding*, vol. 158, pp. 1–16, 2017.

[50] A. Lugmayr, M. Danelljan, R. Timofte, M. Fritsche, S. Gu, K. Purohit, P. Kandula, M. Suin, A. Rajagoapalan, N. H. Joon *et al.*, “Aim 2019 challenge on real-world image super-resolution: Methods and results,” in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 2019, pp. 3575–3583.

[51] J. Cai, H. Zeng, H. Yong, Z. Cao, and L. Zhang, “Toward real-world single image super-resolution: A new benchmark and a new model,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3086–3095.