Signal Processing 166 (2020) 107251

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Single image super-resolution reconstruction based on multi-scale feature mapping adversarial network

Dengwen Zhou, Ran Duan*, Lijuan Zhao, Xiaoliang Chai

School of Control and Computer Engineering, North China Electric Power University, Peking, China

ARTICLE INFO

Article history:
Received 5 October 2018
Revised 13 June 2019
Accepted 5 August 2019
Available online 5 August 2019

Keywords:
Super-resolution
Generative Adversarial Network
Image restoration
Deep learning
Perceptual loss

ABSTRACT

Single image super-resolution (SISR) aims to reconstruct a high-resolution image from a degraded low-resolution image. In recent years, the super-resolution methods based on convolutional neural network (CNN) have achieved promising performance on SISR task, indicating that CNN is a viable approach to image super-resolution reconstruction. The one limitation of the current SISR methods is that many methods use the pixel-wise loss. It is well known that the pixel-wise loss cannot well recover high-frequency details even if the high peak signal-to-noise ratio (PSNR) can be obtained. Some other methods purely focus on restoring more details, which resulted in poor PSNR score and high-frequency noise. In this paper, we proposed a multi-component loss function based on pixel-wise loss, perceptual loss and adversarial loss for a multi-scale feature mapping generator network for SISR image reconstruction model. We evaluated our method on commonly used benchmarks and compared it with other SISR methods. The results showed that our method could achieve the better balance between the high-frequency detail and stable spatial structure generation.

1. Introduction

Single image Super-Resolution (SISR) aims to reconstruct its high-resolution (HR) counterpart from a single low-resolution (LR) image, where the HR image has more pixels and visual information than the original LR one. SISR has many practical applications such as video surveillance, remote sensing image observation, astronomical image processing and medical imaging [1].

A kind of SISR methods [2,3] that generate HR reconstruction images by learning the mapping relation between LR and HR images have been widely studied. In these methods, the prior knowledge that guides the reconstruction process is not artificially defined but acquired through learning [4]. In recent years, deep learning has been widely applied in the SISR task. Dong et al. [5] first propose Super-Resolution Convolutional Neural Network (SRCNN) for the SISR reconstruction, and show good performance. However, the SRCNN model exhibits limitations. First, the SRCNN has the shallow network structure that consists of only three convolutional layers for feature extraction, mapping and reconstruction, respectively. The shallow network structure takes small receptive field, resulting it difficult for SRCNN to learn the features of large range. The size of the receptive field affects the amount of contextual information that can be used to infer the details of the HR image [6]. The larger receptive field provides more information for network to reconstruct images, while the information in smaller receptive field is insufficient for the network to restore the image details. Besides, the shallow network has fewer non-linear activation layers than the deep network and is more difficult to learn and model a complex mapping function from the massive training data and abundant information in images [7]. Second, the SRCNN uses the pixel-wise loss (Mean Squared Error, MSE) to optimize the network, which results in the reconstructed images blurry, especially when the upscaling factor is large. The MSE loss can obtain high Peak Signal to Noise Ratio (PSNR), but fails to capture the high-frequency details of the images [8].

We build an encoder-decoder network with 23 layers based on the algorithm process of feature extraction, mapping and reconstruction of SRCNN to solve SISR problems. The proposed network provides larger receptive field and extracts more complex features from the input image. We design the multi-scale feature mapping module to learn the mapping relation between LR and HR features, where the LR features extracted by different convolutional layers in the encoder are mapped to HR features and processed by corresponding convolutional layers in decoder. The module enables the decoder to use mapped HR features with the receptive fields of different sizes in the reconstruction process, further increasing the amount of information for decoder to reconstruct HR images. Compared with the single convolutional layer in SRCNN, it can increase the utilization of the information of the LR image.

* Corresponding author.
E-mail address: 1162227075@ncepu.edu.cn (R. Duan).

https://doi.org/10.1016/j.sigpro.2019.107251
0165-1684/

Furthermore, we construct a generative adversarial framework to define a novel loss function consisting of MSE loss, perceptual loss and adversarial loss. The proposed multi-scale feature mapping network is optimized by the loss function to better restore the low-frequency content, sharp edges and high-frequency textures of HR images and achieve a better balance of the performance between quantitative score and visual effect.

Our main contributions include:

- We designed a multi-scale feature mapping network for the end-to-end transformation between LR images and HR images. Our network can learn the abundant image information very well, and make good use of the multi-scale features, also significantly improved the performance of SISR reconstruction, compared with some other state-of-the-art SISR methods.
- We defined a joint loss function consisting of MSE loss, perceptual loss and adversarial loss to optimize the multi-scale feature mapping network. The complementarity and antagonism between these loss functions could achieve a good balance between perceptual quality and pixel-level accuracy.

We evaluated our method on the standard benchmark datasets and compared with the state-of-the-art SISR methods. Our method behaved well in both the subjective evaluation and the objective quantification.

2. Related works

In recent years, deep learning has shown the promising ability in the field of computer vision. How to improve the SISR technique with deep learning has become a hot topic. Dong et al. [5] propose the seminal SISR model (SRCNN) based on convolution neural network (CNN). They build a CNN to learn the end-to-end mapping between the LR and HR images. Their model includes three convolutional layers to sequentially extract features from the LR images, learns the mapping relation between the LR and HR image features, and reconstructs HR images from HR features. On the basis of SRCNN, Dong et al. [9] further propose an accelerating super-resolution convolution neural network (FSRCNN) using the hourglass structure. This model can generate higher quality images, also has fewer parameters and lower computation cost than SRCNN, which makes it can work in real time even on a regular CPU. Similarly, Kim et al. [6] improve SRCNN by deepening the network. Their network has 20 convolutional layers, and can extract more complicated features. The deeper network has lower parameter convergence speed and easy to cause problems such as exploding and vanishing gradients. To solve these problems, Kim et al. use residual learning [10] to accelerate convergence. They also limit dynamic range of gradients by clipping strategy, which can effectively resolve the exploding gradients.

Although the above three methods are different in their network structures, they all minimize the mean squared error (MSE) between the reconstructed image and the ground truth, and thus maximize the peak signal-to-noise ratio (PSNR), which is the most common measure to evaluate SISR algorithms. However, it is well known that MSE cannot well capture the high frequency details [8]. The images generated by these MSE-based SISR algorithms almost always tend to be over-smoothed. This problem not only appears in MSE-based SISR algorithms, but the other based pixel-wise loss functions (e.g. L1 loss) are also tend to over-smooth images.

One possible reason that the pixel-wise loss results in over-smoothed images is perhaps that multi-modal distribution cannot be built [11]. Dahl et al. [11] use a different construction method, named PixelCNN, to build dependencies between different pixels to solve this problem. In addition, inspired by the perceptual loss proposed by the Gatys et al. [12] in the style transfer, Johnson et al. [13] firstly use the perceptual loss in image SISR reconstruction and successfully generate high perceptive quality images with sharp edges. They design an image transformation network containing deconvolution layers and residual blocks to transform LR images into HR images, and use a pretrained network (VGG-16) as the loss network to define the perceptual loss function. Different from the pixel-wise loss, the perceptual loss calculates the difference between the reconstructed images and the ground truth in the feature space instead of the pixel space, so it helps to recover the perceptual information.

Generative Adversarial Networks [14–16] (GANs) have been found to have excellent performance in generating plausible-looking natural images with high perceptual quality. Ledig et al. [17] present a generative adversarial framework (SRGAN) for SISR reconstruction. They build a generator network to transform the LR image to its corresponding HR image and a discriminator network to distinguish the generated images from the real HR images. In training process, the generator manages to generate plausible images to fool the discriminator which is trained to learn the differences between the generated HR images and the true HR images. The final goal of their approach is to obtain the super-resolved images with high visual quality which the discriminator cannot distinguish from the true HR images. Compared to the MSE loss, the adversarial loss based on a differentiable discriminator can obtain more suitable and accurate error measurements between the generated images and the true HR images [17].

We can get the pros and cons of three frequently-used loss functions from the previous works. The MSE loss directly shortens the pixel-wise distance between different images. While the perceptual loss based on the features extracted by a classification network, such as VGG, is better at reconstructing sharp edges and fine details, but it fails to recover the local texture details. The Ledig et al.'s work [17] shows that the GAN procedure encourages the reconstruction to be closer to the natural image manifold than the pixel-wise based solutions. However, Ledig et al.'s results seem that the high-frequency textures are overly reconstructed and even incorrectly produced in some smooth areas. The different reconstructed images using above three SISR methods based on different loss functions are exemplified with the corresponding PSNR in Fig. 1.

3. Method

We propose an SISR framework which consists of a generator (G) and a discriminator (D). The generator tries to transform an LR image I^{LR} to the photo-realistic HR image I^{SR}. The discriminator tries to learn the difference between the HR image I^{HR} and the reconstructed image I^{SR} and to provide the probability of which the input image is an HR image. Our goal is to train the generator network to generate images as similar as possible to the true HR images and thus the discriminator fails to distinguish the generated images from the ground truth. Mathematically, it can be described as:

$$l_{GAN}(G,D) = \min_{G}\max_{D} \mathrm{E}_{I^{HR}\sim p_{train}(I^{HR})}\left[\log D\left(I^{HR}\right)\right] + \mathrm{E}_{I^{HR}\sim p_{G}(I^{LR})}\left[\log\left(1-D\left(G\left(I^{LR}\right)\right)\right)\right] \quad (1)$$

It is an adversarial min–max problem. In the training process, the discriminator (D) is optimized by minimizing the loss function, and the generator (G) is optimized by maximizing the loss function.

The overall framework we propose is illustrated in Fig. 2. Our method is summarized in Algorithm 1.

The SISR framework we propose is similar to the SRGAN model which also consists of a generator and a discriminator, but the

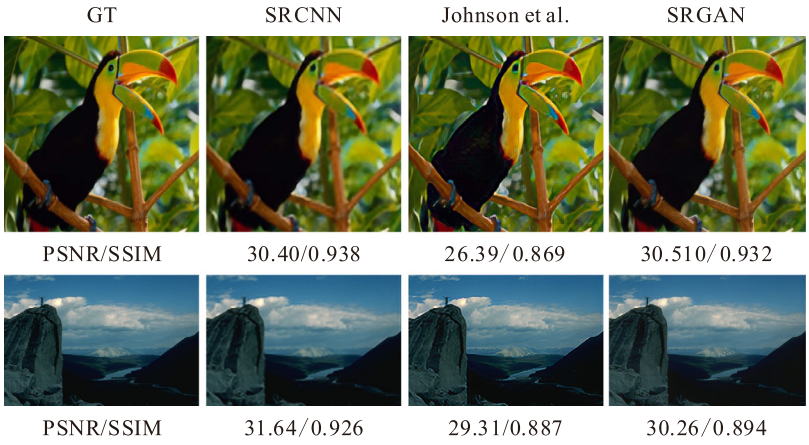

Fig. 1. The images generated by the CNN models with the different loss functions. SRCNN [5] which is based on the MSE-based loss gets high PSNR and SSIM scores but the reconstructed HR images is blurrier. The results of Johnson et al. [13] show that the perceptual loss can better restore clear lines and edges but not the high-frequency textures. While the SRGAN's [17] reconstructed HR images have sharper and more realistic textures than the above two, but produces sharp noise in some smooth areas.

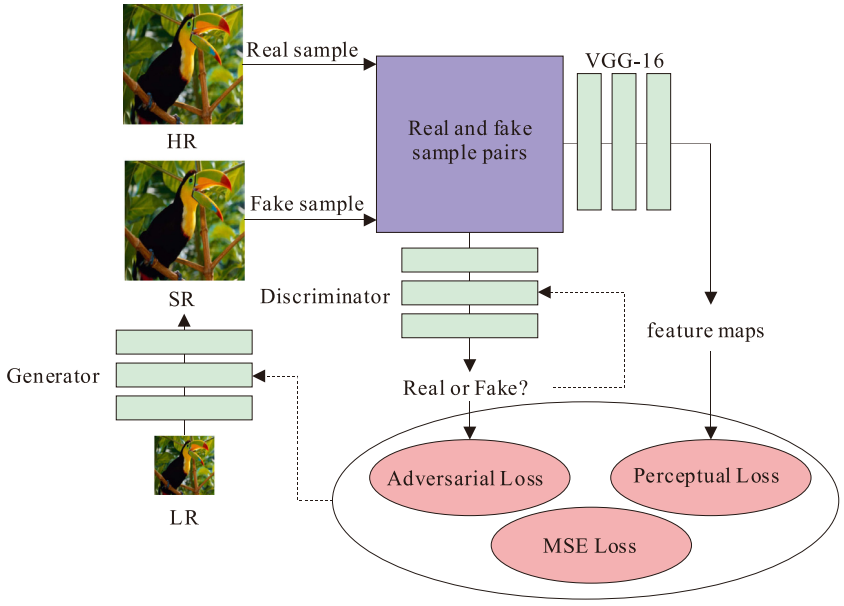

Fig. 2. The structure of our network framework. Similar to SRGAN, our framework consists of a generator and a discriminator. The generator reconstructs photo-realistic HR images from input LR images. The discriminator is trained to differentiate between the generated HR images and the ground truth. A pretrained image classification network (VGG-16) is used to extract feature maps for perceptual loss.

Algorithm 1 Our method for SISR.

Step 1: Input the LR image I^{LR} to the generator network and output the reconstructed image I^{SR}
Step 2: Label the original HR image I^{HR} and the reconstructed image I^{SR} as real and fake respectively, then use them to train the discriminator
Step 3: Calculate the MSE loss using Eq. (3)
Step 4: Input the I^{HR} and I^{SR} to the pre-trained VGG-16 to calculate the perceptual loss according to Eq. (4)
Step 5: Input the I^{SR} to discriminator to get the adversarial loss based on Eq. (5)
Step 6: Combine the MSE loss, perceptual loss and adversarial loss following Eq. (2) to train the generator network
Step 7: Repeatedly execute Step 1–5 until the rival between generator and discriminator achieves a balance

network structures of both the generator and the discriminator are different from SRGAN. We build an encoder-decoder network as our generator network to implement an end-to-end mapping between the LR and HR images. The generator of SRGAN (SRResNet) simply employs the ResNet architecture from He et al.'s work [10], which gets good performance by using multiple residual blocks with the same structure. Different from SRResNet, our generator is built similar to SRCNN, which follows the algorithm process of the Sparse coding based super resolution (ScSR) [2,3]. First, a deep encoder network is built to extract LR image features. The feature maps have larger receptive field by increasing the number of convolutional layers and by using the max-pooling layers. Second, the multi-scale feature mapping module is used to learn the mapping relation between the LR and HR features extracted from

different convolutional layers in encoder network, and the mapped HR features are concatenated to the input of the corresponding convolutional layers in decoder network. Both the low-level features from the shallow layers and the high-level features from the deeper layers can be used for image reconstruction, which helps the decoder use more features and information to restore details when reconstructing the HR image. In addition to the generator, our discriminator is also different from SRGAN. We build our discriminator based on the fully convolutional network (FCN) and train it by following the patchGAN proposed by Isola et al. [16]. The FCN has fewer parameters than the full-connection network used in SRGAN's discriminator, while the training strategy based on patchGAN [16] reduces the amount of calculation. Besides, our discriminator running on small image patches encourages local texture matching and prediction, which helps generate more realistic and natural images [18]. More details of the generator and discriminator are described in Section 3.1 and Section 3.2.

SRGAN [17] constraints the content deviation of the generated images by using the perceptual loss [13]. However, Johnson et al.'s work [13] has shown that perceptual loss based on the high-level features is good at restoring high-frequency information rather than the low-frequency content. The previous works based on GANs [17,19] confirm that it is beneficial to combine the GAN objective with a pixel-wise loss, such as the MSE-based loss, so we add the pixel-wise loss (MSE) into SRGAN's loss function as our generator's objective function. Finally, the total objective loss function for our generator is built by the perceptual loss, the adversarial loss and the MSE-based content loss. The MSE-based content loss ensures the accuracy of the global low frequency information. The adversarial loss is mainly used to estimate the textures and the perceptual loss is mainly used to enhance image edges. More details about the individual loss functions are described in Section 3.3.

3.1. Generator network

The SRCNN proposed by Dong et al. [5] only has three layers. Each layer performs features extraction, non-linear mapping and reconstruction, respectively. This shallow network cannot extract hierarchical image features. An intuition is that deepening the network can perhaps improve the result. However, Dong et al. [5] fail to obtain better performance using a deeper model. They conclude that deepening networks has little help to improve performance. Whereas, Kim et al. [6] argue that increasing depth significantly boosts performance by their VDSR model with 20 weight layers. This model solves the vanishing/exploding gradients problem by an adjustable gradient clipping and speeds up the convergence by the residual-learning.

Inspired by the work of Kim et al. [6] on VDSR, we designed an encoder-decoder generator network with 23 convolutional layers that was deeper than SRCNN. The deeper network has larger receptive field and also has higher non-linearities, which can provide more contextual information and model more complex mapping relations [6]. The symmetric encoder-decoder structure combined with the multi-scale feature mapping module can map the multi-scale features extracted by the encoder to the HR features to reconstruct HR images. Besides, the skip-connection used for information transfer between the encoder, the mapping module and the decoder can accelerate the convergence of the parameters.

3.1.1. Encoder-decoder network

The proposed encoder-decoder network is similar to the U-net [21]. The encoder consisting of 7 convolutional layers and 2 max-pooling layers extracts the LR feature maps with multiple sizes by cascading different numbers of the convolutional layers. We use the small convolutional kernels (3×3) except the first layer to reduce the computation load of the network. We can enlarge the receptive field by increasing the number of convolutional layers [7]. Each max-pooling layer in the encoder downsamples the feature maps by a factor of 2. The downsampling can further expand the receptive field and reduce the sizes of the feature maps, and thus reduce the computational complexity. In order to compensate for the loss of information incurred by downsampling, the skip-connections are used to transfer the un-downsampled low-level feature maps to the decoder for image reconstruction. We also double the numbers of the convolution filters following max-pooling layers to compensate for the lost information. The decoder has 7 convolutional layers and 2 sub-pixel convolutional layers to reconstruct HR images using the multi-scale feature maps extracted from the encoder. The sizes of all the convolutional kernels except the last layer in the decoder are 3×3 which are symmetrical with the encoder. The sub-pixel convolutional layers upsample the feature maps shrunk by the max-pooling layers in the encoder to restore the original size of the HR images. More details of the network structure were illustrated in Fig. 3.

3.1.2. Multi-scale feature mapping module (MSFM)

Our multi-scale feature mapping module is made up of three similar but independent sub-modules, which work on the different feature maps extracted by the encoder, respectively. MSFM has two main effects: Firstly, each sub-module uses 2 sub-pixel convolutional layers to upsample the LR feature maps to the same size as the HR feature maps. On the other hand, MSFM learns the mapping relation between the multi-scale LR features and the HR features. We use skip-connections [10] to transfer the outputs of the convolutional layers in the encoder to MSFM, and concatenates MSFM's outputs with the inputs of the corresponding layers in the decoder.

3.1.3. Batch normalization

Although the batch normalization layer gives good performance in image classification task, Lim et al. [22]'s HR image reconstruction model hasn't used the batch normalization layers (EDSR). The reason is maybe that the batch normalization removes the range flexibility of the network and deteriorates the performance of the HR image reconstruction model [22]. Our generator network also removes the batch normalization layer. The utilization of memory will also be greatly reduced because the batch normalization layer has the same number of parameters as the convolutional layer.

3.1.4. Sub-pixel convolutional layer

In our decoder and multi-scale feature mapping module, we choose the sub-pixel convolutional layer instead of the deconvolution layer to upsample the feature maps because Odena et al. [23] have demonstrated that the deconvolution layers in the neural network tend to produce the checkerboard artifacts at the pixel level in the generated image. Odena et al. also suggest some methods as a substitute of the deconvolution to prevent this problem, such as the sub-pixel convolutional layer designed by Shi et al. [20] or the resize-convolution layer [23]. In this paper, we use the former one.

3.2. Discriminator network

Our discriminator network is different from the traditional GAN [14]. Since the generated image only needs to be labeled as a real or a fake image, we can regard the discriminator as a binary classifier which can make feature extraction and linear classification. A traditional discriminator takes the whole image as its input and uses the fully connected layers for classification, which needs a high computational cost and lacks of flexibility because the number of the discriminator parameters increases with the size of the input image. We design a fully convolutional network

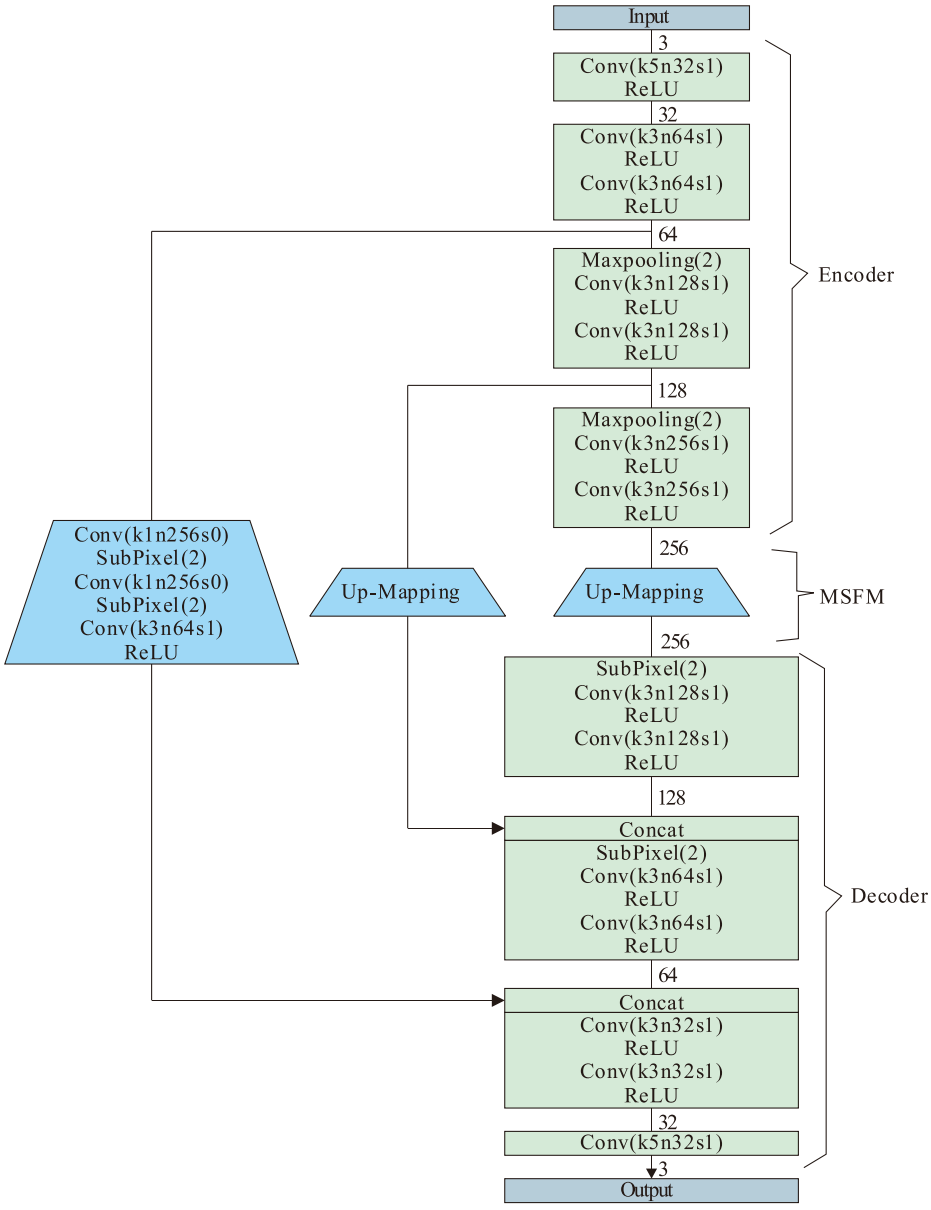

Fig. 3. The structure of our generator network. The convolutional kernel size (k), stride (s) and the number of feature maps (n) are also indicated for each convolutional layer.

based on patchGAN [16] as the discriminator to discriminate the image patches, as shown in Fig. 4.

The proposed discriminator network is composed of convolutional layers, batch-normalization layers, activation layers and max-pooling layers alternately. Except for the first convolutional layer with kernel size 5×5 and stride 2, all the other convolutional layers' kernel sizes and strides are set to 3×3 and 1, respectively. The max-pooling layers perform downsampling operations by a factor of 2 to enlarge the receptive field. The batch-normalization layers are used to improve the generalization ability of the discriminator network [24]. Inspired by Isola et al.'s work [16], our discriminator manages to determine whether each $N \times N$ patch instead of the whole input image is an authentic HR image patch. The final score which is used to decide on the authenticity is obtained by averaging the output responses of all the patches. Isola et al. [16] demonstrate that the high-quality results can still be generated when N is much smaller than the full size of the input image. This is a great benefit because a small patch makes the discriminator have lower computational complexity, run faster than the whole input image, and further can be applied on arbitrarily large images.

Such a discriminator effectively models the image as a Markov random field, assuming the independence between the pixels separated by more than a patch diameter [25]. This relation is also the common assumption in the models of texture [26] and style [12,27]. The PatchGAN can therefore be understood as a form of texture/style loss.

3.3. Loss function

Our ultimate objective is to train a generator to generate realistic and accurate HR images, the loss function l_G defined for the generator G is critical for the performance of the framework. We improved on Ledig et al.'s work by combining the MSE-based loss, perceptual loss and adversarial loss together to jointly optimize a solution, a better balance was obtained between the excessive sharpening and the excessive smooth. We formulate our genera-

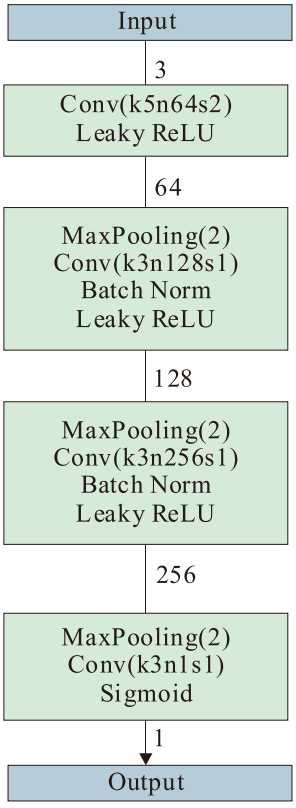

Fig. 4. The structure of our discriminator network. The convolutional kernel size (k), stride (s) and the number of feature maps (n) are also indicated for each convolutional layer.

tor's loss function as the weighted sum of these three losses as:

$$l_G = l_{\text{Pixel}} + W_P l_{\text{Perceptual}} + W_A l_{\text{Adversarial}} \tag{2}$$

where the l_{Pixel}, $l_{\text{Perceptual}}$ and $l_{\text{Adversarial}}$ represent the MSE loss, perceptual loss and adversarial loss, respectively. The W_P and W_A are the weights of the perceptual loss and the adversarial loss, respectively. The optimal WP=0.1 and WA=0.001 were found using cross-validation. In the following we will describe the details of these three loss functions.

3.3.1. MSE loss

Suppose the super-resolution factor is f and I^{HR} is the target HR image of the size $H \times W \times C$, where H is the height, W is the width and C is the color channel of the image, I^{LR} is the LR image of the size $H/f \times W/f \times C$. The MSE-based loss is as following:

$$l_{\text{Pixel}} = \frac{1}{CHW} \sum_{x=1}^{H} \sum_{y=1}^{W} \sum_{z=1}^{C} \left(I^{\text{HR}}_{x,y,z} - G\left(I^{\text{LR}}\right)_{x,y,z} \right)^2 \tag{3}$$

where the $G(I)$ is the mapping function learned by the generator between the LR and HR images.

It can be seen from Eq. (3) that the MSE loss calculates the pixel-wise Euclidean distance between the generated and the target images, which is more sensitive to abnormal pixels with large numerical difference. Therefore, it is difficult to capture the perceptual error in images and generate images with high-frequency details for those MSE-based models, but it restores the low-frequency content of the image well.

3.3.2. Perceptual loss

We used the 16-layer VGG network [28] pretrained on the ImageNet[1] [29] dataset to construct the perceptual loss function:

$$l_{\text{Perceptual}} = \frac{1}{CHW} \sum_{z=1}^{C} \sum_{x=1}^{H} \sum_{y=1}^{W} \left(\phi_{i,j}\left(I^{\text{HR}}\right)_{x,y,z} - \phi_{i,j}\left(G\left(I^{\text{LR}}\right)\right)_{x,y,z} \right)^2 \tag{4}$$

where the $\phi_{i,j}(X)$ is the activations of the jth layer in front of the ith max-pooling layer of the network. Here we take $i=4$ and $j=3$ which were determined by cross-validation.

The perceptual loss can be regarded as the difference in the feature domain of the images. For VGG [28], the low-level feature maps contain the multifarious and detailed information, while the high-level feature maps preserve more abstract structure due to the larger receptive field and smaller size [13]. Johnson et al. [13] use the perceptual loss to resolve the SISR image reconstruction problem. Their method can produce sharp edges in the resulted images, but fails to achieve a satisfactory effect on the restoration of the local high-frequency textures.

We consider that there are two factors leading to this problem. Firstly, the VGG network mainly serves for the image classification task, it extracts the feature maps, and enhances the spatial structure and semantic information of the images, but weakens the local detailed textures which are not very important for image classification. Besides, Johnson et al. [13] take the Euclidean distance between the feature representations as the perceptual loss, the high-frequency textures are further smoothed out. We tried to add adversarial loss to compensate for the restoration of the high-frequency texture, while the perceptual loss was only used to restore the edge information of the images.

3.3.3. Adversarial loss

Based on the adversarial mechanism and the Eq. (1), the expression of the adversarial loss $l_{\text{Adversarial}}$ is defined as:

$$l_{\text{Adversarial}} = -\log\left(\frac{1}{H'W'} \sum_{x=1}^{H'} \sum_{y=1}^{W'} D(G(I^{\text{LR}}))_{x,y} \right) \tag{5}$$

where the $D(G(I^{\text{LR}}))$ is the probability matrix of the size $H' \times W'$ output by the discriminator, here we use -log $D(G(I^{\text{LR}}))$ instead of log $(1 - D(G(I^{\text{LR}})))$ to facilitate the calculation of the gradient [14].

The adversarial loss based model (SRGAN) proposed by Ledig et al. [17] has shown its superior performance in recovering the photo-realistic textures from the heavily down-sampled images. However, using the adversarial loss to restore the high-frequency textures sometimes behaves too aggressively, and thus reduces the accuracy of the basic structural information. In more serious cases, SRGAN even damages the local shape and produces noise in the smooth areas [18]. Sajjadi et al. [18] analyze this problem and propose that using texture loss [25,26] generates more natural high-frequency texture. Inspired by Isola et al. [16], we train the discriminator following the patchGAN to learn the local texture distribution of images. The adversarial loss defined by this discriminator can not only generate realistic images but also be used as a form of texture loss for the restoration of image texture [16]. In addition, the MSE loss can also effectively reduce the high-frequency noise in the smooth region because of its sensitivity to abnormal values.

4. Experiments

Our experiments were performed on widely used benchmark datasets, and the scale factor of ×4 was used in all the experiments. The objective results of the SISR reconstruction were evaluated using the common peak signal to noise ratio (PSNR) [30] and the structural similarity (SSIM) [31] measures.

[1] http://image-net.org/download-images.

Table 1
The average PSNRs(dB) and SSIMs compared with Bicubic, SRCNN and FSRCNN on Set5, Set14 and BSD100, the results of the SRCNN and FSRCNN trained on both the 91-Images and the DIV2K datasets are showed for comparison.

Datasets	Bicubic		SRCNN (91-Images)	SRCNN (DIV2K)	FSRCNN (91-Images)	FSRCNN (DIV2K)	Ours (MSE) (DIV2K)
Set5	×2	33.65/0.930	36.66/0.954	36.84/0.956	37.00/0.955	37.13/0.957	**37.51/0.958**
	×3	30.39/0.868	32.75/0.909	33.01/0.914	33.16/0.914	33.24/0.918	**33.62/0.920**
	×4	28.43/0.810	30.49/0.863	30.56/0.866	30.71/0.865	30.81/0.869	**31.32/0.881**
Set14	×2	30.34/0.870	32.42/0.906	32.55/0.905	32.64/0.908	32.73/0.910	**33.02/0.911**
	×3	27.55/0.774	29.28/0.820	29.39/0.821	29.43/0.824	29.51/0.826	**29.73/0.829**
	×4	25.77/0.703	27.50/0.751	27.57/0.758	27.60/0.753	27.63/0.761	**27.78/0.773**
BSD100	×2	29.56/0.844	31.36/0.887	31.44/0.890	31.51/0.890	31.59/0.891	**31.89/0.895**
	×3	27.21/0.738	28.41/0.786	28.47/0.787	28.52/0.789	28.58/0.791	**28.74/0.796**
	×4	25.98/0.671	26.91/0.712	27.03/0.712	26.97/0.713	27.09/0.713	**27.40/0.736**
Average	×2	31.18/0.881	33.48/0.916	33.61/0.917	33.72/0.918	33.82/0.919	**34.14/0.921**
	×3	28.38/0.793	30.15/0.838	30.29/0.841	30.37/0.842	30.44/0.845	**30.70/0.848**
	×4	26.73/0.728	28.30/0.775	28.39/0.779	28.43/0.777	28.51/0.781	**28.83/0.797**

Fig. 5. The reconstruction results of ×2 upsampling compared with Bicubic Interpolation, SRCNN and FSRCNN.

4.1. Training details

The 91 images are generally used for training in the previous SISR image reconstruction methods, while it is too small for a deep network. We used the DIV2K dataset[2] provided by NTIRE 2017 super-resolution challenge[3] [32] as the training dataset which consists of 1000 diverse 2K-resolution RGB images, where 800 images are for training, 100 for validation and 100 for testing purposes. We augmented the training dataset by rotating the 800 images (0°, 90°, 180° and 270°) and clipping them to patches with a stride of 200, finally we obtained an augmented training dataset with 329,184 sub-images.

We trained both the generator G and the discriminator D using ×4 super-resolution. In each training batch, we randomly selected 8 image patches as the HR patches, each patch with the size of 400 × 400. For the 8 images of each batch, we obtained 16 sub-patches of size 128 × 128 from each image by random clipping and encapsulated the $8 \times 16 = 128$ sub-patches into a batch as the input. We obtained the LR patches by blurring the HR patches with a Gaussian kernel of standard deviation $\sigma = 1.0$ and down-sampling them using the bicubic kernel with the specified down-sampling factor. The parameters are initialized by the "MSRA" method [33]. We trained the model for 200k iterations using Adam [34] with an initial learning rate of 1×10^{-4} without weight decay or dropout. The learning rate of the network decreased by 0.8 times per 20k iterations. Our implementation used PyTorch on a single NVIDIA GTX Titan X GPU.

4.2. Performance evaluations

4.2.1. Baselines

Extensive experiments have been carried out to verify the effectiveness of the proposed method. We choose the 4 representative methods for comparisons: SRCNN [5], FSRCNN [9], Johnson et al. [13] and SRGAN [17]. The methods under consideration used the same training dataset DIV2K [32]. When trained SRCNN and FSRCNN, we initialized the network parameters with the pre-trained models[4,5] provided by Dong et al., and other settings in the training process remained unchanged. We implemented the Johnson et al.'s method according to the supplementary material[6] of [13] and obtained very similar results. The source code of the SRGAN method was obtained from GitHub.[7]

4.2.2. Evaluations of the network

We trained the multi-scale feature mapping network using the MSE loss, and compared it with the other representative SISR methods SRCNN [5], FSRCNN [9] and Bicubic interpolation. All the models were evaluated on the Set5, Set14 and BSD100 datasets, and the PSNRs (dB) and SSIMs were calculated on the brightness channel of all images. The quantitative results are shown in Table 1. It can be seen that the average PSNRs and SSIMs of our MSE-based models on the scale factors of 2, 3 and 4 are all the highest compared with the other methods.

Figs. 5–7 shows the visual results of all the methods on the scale factor of 2, 3 and 4, respectively. The images restored by our multi-scale feature mapping network have clearer contours and edges. The reason perhaps is that the features extracted using our network have larger receptive field, and the skip-connections also make full use of the features.

4.2.3. Evaluations of the loss functions

We analyzed the effects of the MSE loss, the perceptual loss, and the adversarial loss on the quality of the reconstructed images, and proposed a novel loss function (EPA) contained above 3 loss functions. Fig. 8 showed the comparison of the resulted HR images generated using different loss functions on the same network. The images which generated using the MSE loss lacked clear edges and textures and seemed overly smooth. Combining the perceptual loss with adversarial loss (PA) generated clear images having rich details and textures, but some smooth areas were also affected by

[2] https://data.vision.ee.ethz.ch/cvl/DIV2K/.
[3] http://www.vision.ee.ethz.ch/ntire17/.
[4] http://mmlab.ie.cuhk.edu.hk/projects/SRCNN.html.
[5] http://mmlab.ie.cuhk.edu.hk/projects/FSRCNN.html.
[6] https://link.springer.com/chapter/10.1007/978-3-319-46475-6_43#SupplementaryMaterial.
[7] https://github.com/leftthomas/SRGAN.

Fig. 6. The reconstruction results of ×3 upsampling compared with Bicubic Interpolation, SRCNN and FSRCNN.

Fig. 7. The reconstruction results of ×4 upsampling compared with Bicubic Interpolation, SRCNN and FSRCNN.

Fig. 8. Comparison of multi-scale feature mapping networks based on different loss functions.

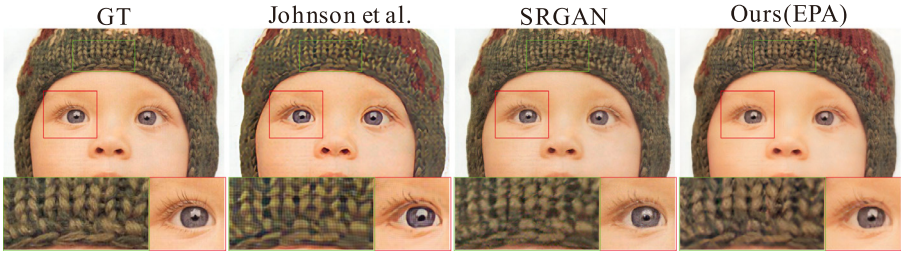

Fig. 9. The reconstructed *baby* image in Set5 using Ours, the Johnson et al. and SRGAN for ×4 upscaling. We used the colorized boxes to highlight a small region containing the rich details. The zoomed regions were shown directly below. From the magnifying images, we could see the advantage of our method compared with other methods. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 10. The reconstruction of *Lenna* image in Set14. From the red boxes and the corresponding zoomed counterpart, we could see that our method generated less noise on the brim of the hat than SRGAN. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

sharp textures (e.g. the black speckled noises in the smooth area of the man's face). The proposed loss function (EPA) could restore more clearly contour and natural texture and reduce the noise in images effectively.

4.2.4. Evaluations of the final model

In order to verify the performance of the final model using the EPA loss, we compared our method with Johnson et al.'s method [13] and the SRGAN [17] in Figs. 9–11. Considering the reconstructed image baby from Set5 shown in Fig. 9, our method reconstructed the clearer and more realistic texture of the wool on the baby's hat, while Johnson et al.'s result was not natural (more like a stylized image). SRGAN seemed to generate a photo-realistic image, but the wool seemed not real. From Fig. 10, we further demonstrated that our method had significantly better results. We could see the clearer comparison from Fig. 11. The SRGAN attached excessive importance to the restoration of the high-frequency details instead caused the clearly visible artifacts. Our method achieved the better balance between the high perceptual quality and the information accuracy than SRGAN and Johnson et al.'s method.

The PSNRs (dB) and SSIMs on the brightness channel on the Set5, Set14 and BSD100 datasets for ×4 super-resolution are shown in Tables 2 and 3. It could be seen from Table 2 that our method had higher average PSNR and SSIM scores than Johnson et al. and SRGAN because the MSE-loss was added as part of the whole loss function to optimize the generator. Table 3 showed a fairer comparison that the Johnson et al.'s model and SRGAN were trained by using the same dataset as our method.

Fig. 11. The reconstructed *comic* image in Set14. More details were shown on the zoomed images in the second row. Our method had significantly better results.

Table 2
The average PSNRs(dB) and SSIMs compared with Bicubic, Johnson et al. [13]. and SRGAN[17] on Set5, Set14 and BSD100.

Datasets	Bicubic (PSNR/SSIM)	Johnson et al. (PSNR/SSIM)	SRGAN (PSNR/SSIM)	Ours(EPA) (PSNR/SSIM)
Set5	28.43/0.810	27.09/0.768	29.40/0.847	**29.67/0.887**
Set14	25.77/0.703	24.99/0.673	26.02/0.739	**26.63/0.801**
BSD100	25.98/0.671	24.95/0.631	25.16/0.668	**26.16/0.774**
Average	26.73/0.728	25.67/0.690	26.86/0.751	**27.49/0.821**

Table 3
The comparison of the average PSNRs and SSIMs with SRGAN [17] and Johnson et al.'s model [13] trained on the DIV2K training dataset.

Datasets	Bicubic (PSNR/SSIM)	Johnson et al.(DIV2K) (PSNR/SSIM)	SRGAN(DIV2K) (PSNR/SSIM)	Ours(EPA) (PSNR/SSIM)
Set5	28.43/0.810	28.08/0.794	29.17/0.838	**29.67/0.887**
Set14	25.77/0.703	25.53/0.736	25.79/0.742	**26.63/0.801**
BSD100	25.98/0.671	24.65/0.628	25.05/0.661	**26.16/0.774**
Average	26.73/0.728	26.08/0.719	26.67/0.747	**27.49/0.821**

5. Conclusion

In this paper, we presented some limitations of the previous SISR method in terms of the network structure and the loss function. To solve these problems, we proposed a framework based on GAN, which consisted of a generator network with multi-scale feature mapping module and a discriminator using the patchGAN. We combined the MSE-based loss, the perceptual loss and the adversarial loss for the generator network. Our method achieved the highest mean PSNR and SSIM scores on the commonly used benchmarks among the methods under comparison. Our results also had the best visual quality according to the generated $\times 4$ upscaling images on the test images. Our method obtained the best balance between the visual quality and the accuracy of the pixel level.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by Beijing Natural Science Foundation (grant no. 4162056).

References

[1] L.W. Yue, H.F. Shen, J. Li, et al., Image super-resolution: the techniques, applications, and future, Signal Process. 128 (2016) 389–408.
[2] J. Yang, J. Wright, T.S. Huang, et al., Image super-resolution as sparse representation of raw image patches, in: IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, IEEE, 2008, pp. 1–8.
[3] J. Yang, J. Wright, T.S. Huang, et al., Image super-resolution via sparse representation, IEEE Trans. Image Process. 19 (11) (2010) 2861–2873.
[4] X. Sun, X.-G. Li, J.-F. Li, et al., Review on deep learning based image super-resolution restoration algorithms, Acta Autom. Sinica 43 (5) (2017) 697–709.
[5] C. Dong, C.L. Chen, K. He, et al., Image super-resolution using deep convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2016) 295–307.
[6] J. Kim, J.K. Lee, K.M. Lee, Accurate image super-resolution using very deep convolutional networks, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Las Vegas, USA, 2016, pp. 1646–1654.
[7] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations, ICLR, Canada, 2015.
[8] Z. Wang, A.C. Bovik, H.R. Sheikh, et al., Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[9] C. Dong, L. Chen C, X. Tang, Accelerating the super-resolution convolutional neural network, in: European Conference on Computer Vision, Amsterdam, the Netherlands, ECCV, 2016, pp. 391–407.
[10] K.M. He, X. Zhang, S. Ren, et al., Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, IEEE, 2016, pp. 770–778.
[11] R. Dahl, M. Norouzi, J. Shlens, et al., Pixel recursive super resolution, in: IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, USA, IEEE, 2017, pp. 5439–5448.
[12] L.A. Gatys, A.S. Ecker, M. Bethge, Image style transfer using convolutional neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, IEEE, 2016, pp. 2414–2423.
[13] J. Johnson, A. Alahi, F.F. Li, Perceptual losses for real-time style transfer and super-resolution, in: European Conference on Computer Vision, Amsterdam, the Netherlands, ECCV, 2016, pp. 694–711.
[14] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., Generative adversarial nets, in: International Conference on Neural Information Processing Systems, Montreal, Canada, NIPS, 2014, pp. 2672–2680.
[15] S. Reed, Z. Akata, X. Yan, et al., Generative adversarial text to image synthesis, in: International Conference on International Conference on Machine Learning, New York City, NY, USA, ICML, 2016, pp. 1060–1069.
[16] P. Isola, J.Y. Zhu, T. Zhou, et al., Image-to-image translation with conditional adversarial networks, in: IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, USA, IEEE, 2017, pp. 5967–5976.
[17] C. Ledig, Z. Wang, W. Shi, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, USA, IEEE, 2017, pp. 105–114.
[18] M.S. Sajjadi, B. Scholkopf, M. Hirsch, et al., EnhanceNet: single image super-res-

olution through automated texture synthesis, in: IEEE International Conference on Computer Vision, Italy, IEEE, 2017, pp. 4501–4510.
[19] D. Pathak, P. Krahenbuhl, J. Donahue, et al., Context encoders: feature learning by inpainting, in: IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, IEEE, 2016, pp. 2536–2544.
[20] W. Shi, J. Caballero, F. Huszar, et al., Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, IEEE, 2016, pp. 1874–1883.
[21] O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Cham: MICCAI, 2015, pp. 234–241.
[22] B. Lim, S. Son, H. Kim, et al., Enhanced deep residual networks for single image super-resolution, in: IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, USA, IEEE, 2017, pp. 1132–1140.
[23] A. Odena, V. Dumoulin, C. Olah, Deconvolution and Checkerboard Artifacts, Distill, 2016 http://doi.org/10.23915/distill.00.
[24] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, Lille, France, ICML, 2015, pp. 448–456.
[25] C. Li, M. Wand, Precomputed real-time texture synthesis with Markovian generative adversarial networks, in: European Conference on Computer Vision, Amsterdam, the Netherlands, ECCV, 2016, pp. 702–716.
[26] L.A. Gatys, A.S. Ecker, M. Bethge, et al., Texture synthesis using convolutional neural networks, in: International Conference on Neural Information Processing Systems, Montreal, Canada, NIPS, 2015, pp. 262–270.
[27] C. Li, M. Wand, Combining Markov random fields and convolutional neural networks for image synthesis, in: IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, IEEE, 2016, pp. 2479–2486.
[28] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations, ICLR, 2015.
[29] O. Russakovsky, J. Deng, H. Su, et al., ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.
[30] Q. Huynh-Thu, M. Ghanbari, Scope of validity of PSNR in image/video quality assessment, Electron. Lett. 44 (13) (2008) 800–801.
[31] Z. Wang, A.C. Bovik, H.R. Sheikh, et al., Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[32] E. Agustsson, R. Timofte, NTIRE 2017 challenge on single image super-resolution: dataset and study, in: IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, USA, IEEE, 2017, pp. 1122–1131.
[33] K. He, X. Zhang, S. Ren, et al., Delving deep into rectifiers: surpassing human-level performance on imagenet classification, in: International Conference on Computer Vision, 2015, pp. 1026–1034.
[34] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, International Conference on Learning Representations, ICLR, 2015.