Real-World Superresolution by Using Deep Degradation Learning

Rui Zhao[1(✉)], Junhong Chen[2], and Zhen Zhang[3]

[1] Institute of Special Environments Physical Sciences, Harbin Institute of Technology (Shenzhen), Shenzhen, China
zhaorui2020@hit.edu.cn
[2] Sun Yat-sen University, Guangzhou, China
caily3@mail2.sysu.edu.cn
[3] Shanghai Institute of Aerospace Electronic Technology, Shanghai, China

Abstract. Most current deep convolutional neural networks can achieve excellent results on a single image superresolution and are trained using corresponding high-resolution (HR) images and low-resolution (LR) images. Conversely, their superresolution performance in real-world superresolution tests is reduced because these methods create paired LR images by simply interpolating and downsampling HR images, which is very different from natural degradation. In this article, we design a new unsupervised framework conditioned by degradation representations of real-world hyperresolution problems. The approach presented in this paper consists of three stages: we first learn the implicit degradation representation from real-world LR images and then acquire LR images by shrinking the network, which will share similar degradation with real-world images. Finally, we make paired data of the generated real LR images and HR images for training the SR network. Our approach can obtain better results than the recent SR approach on the NTIRE2020 real-world SR challenge Track1 dataset.

Keywords: Super resolution · Contrastive learning · Image degradation

1 Introduction

Image superresolution is a problem that attempts to obtain a higher resolution image from a low-resolution quality image. In the last few years, DNN-based methods have achieved remarkable results of impressive visual quality, which mainly concentrate on building complicated network architectures to enhance various metrics in existing datasets. Most methods use simple interpolation operations to downsample HR images to construct paired training data. Despite the effectiveness and convenience of this operation, these methods have not considered the uncertainty of real-world degradation. Some approaches model the degradation by the ideal downsampling method:

$$Y_{lr} = (X_{hr} \otimes k) \downarrow_s + n \tag{1}$$

The Support Plan for Core Technology Research and Engineering Verification of Development and Reform Commission of Shenzhen Municipality (number 202100036).

Y. Wang et al. (Eds.): ICPCSEE 2022, CCIS 1628, pp. 209–218, 2022.
https://doi.org/10.1007/978-981-19-5194-7_16

where Y_{lr} and X_{hr} indicate the LR image and HR image, respectively $X_{hr} \otimes k$ means the HR image is blurred with the kernel k, $\downarrow_s$ means the image will be downsampled by scale factor s times, and the parameter n indicates the natural noise. According to this degradation model, we can easily obtain paired training data. Alternatively, it is also difficult to predict the real-world complicated blur kernel. Some recent studies have proposed GAN-based kernel estimation methods to generate realistic images. Based on paired data, the goal of the SR model is to minimize the average cost of images in the dataset:

$$\arg\min_{F} \frac{1}{N} \sum_{i=1}^{N} Loss_{sr}(F(y_i), x_i) \tag{2}$$

where $F(.)$ refers to an SR model and $Loss_{sr}$ is a total loss function. If we valid the SISR model with the same downsampling dataset, the results are not unexpected. Once testing the model with the real-world image, the SR images are of poor quality. It is clear that the image downscaled by the bicubic operation does not have the same degradation as the real-world images. Therefore, creating an LR image similar to real-world degradation is a very useful problem that must be addressed. To this end, enlightened by the popularity of contrastive learning in computer vision, we promote a novel unsupervised-based superresolution approach that uses degradation representations to assist the downsampling network in generating realistic LR images and constructing paired input data for the superresolution network. Specifically, we assume that the images in the same dataset have approximate degradation. Consequently, the degradation of a real-world low resolution image will be approximate to other images in a dataset and far from high resolution images, as shown in Fig. 1.

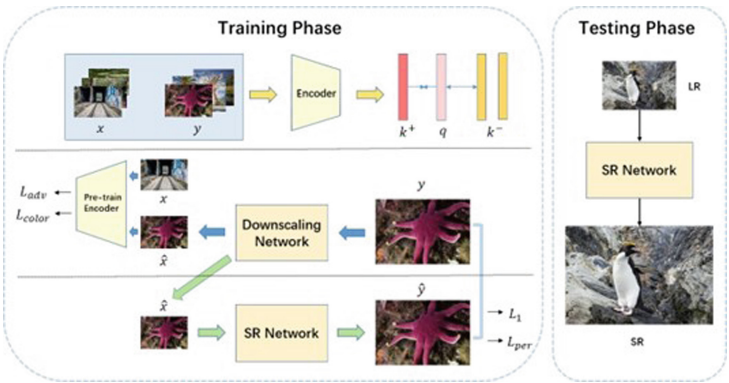

Fig. 1. The overview map of our proposed method in this paper. First, we train the degradation learning network, as shown by the yellow arrows. Given a trained encoder, the downscaling network is trained by employing adversarial and color losses, depicted by blue arrows. The SR network is optimized to obtain the high-resolution images in the third step, assisted by paired data $(\hat{x}, y)$ created in our downscaling network. (Color figure online)

Additionally, we develop an LR image generator to create realistic and paired LR images, which are the input of the superresolution network. By this means, our model

is available for real-world degradation patterns instead of interpolations (e.g., nearest neighbor). To test the validity of our approach, we perform many experiments on the NTIRE2020 real-world SR challenge Track1 dataset. The results of experiments show that our method outperforms most of the current methods. Finally, we perform an ablation study to show the significance of the degradation learning module and downsampling network.

2 Related Work

2.1 Real-World Superresolution

In most previous superresolution studies, datasets with paired images are usually obtained by downsampling HR images with fixed operations. The SRCNN [1] was the first to apply a convolutional neural network to superresolution, and then various LR-to-HR reconstruction networks were developed to enhance SR performance. However, these models can only achieve good results on clean datasets because the model has not been trained with blurry or noisy image data. This is obviously different from real-world images, which often carry serious noise and blur. Cai et al. [2, 3] collected paired photos from the real world with a special camera directly. However, making such a dataset requires considerable manpower and material resources. To solve the problem of real-world superresolution, some research attempts to solve the SR problem without using paired training datasets. Lugmayr et al. [4] designed a downsample network to generate images with degradation and then used them to train an upsample network. Yuan et al. [5] developed a cycle framework to train degradation and superresolution networks concurrently. Since these aforementioned methods regard degradation as input, a growing number of studies have started relying on predicting degradation for real-world SR. Hence, incorrect degradation estimation can lead to poor SR performance with respect to fidelity. To address this issue, Gu et al. [6] optimized the estimated degradation by iteratively comparing the SR image with the ground truth.

2.2 Contrastive Learning

Currently, there are two unsupervised representation learning methods, generative learning and contrastive learning. Generative learning methods usually rely on autoencoding of images and conduct representation learning to minimize the similarity of the output images and the ground truth images in the pixel features. As a result, most of them require expensive calculation costs. Instead, contrastive learning aims to make the output representations closer to the positive images and farther away from those negative ones. Chen et al. [7] proposed a novel framework, named SimCLR, which extracts representations using a variety of data augmentations and contrastive learning. After that, He et al. [8] developed MoCo and MoCo v2, using a momentum encoder and a memory bank to maintain consistent representations. In this paper, images in datasets that seem to take the same degradation are considered positive counterparts, and contrastive learning will learn to draw content-invariance degradation features.

3 PurPosed Method

3.1 Overview of the Unsupervised Framework

Our real-world SR framework component is shown in Fig. 1, which contains three parts: a degradation learning network, an LR image generator and a reconstruction model. The degradation learning network aims to train an encoder that can extract degradation features from real-world images and assist generators in obtaining more realistic LR images. Motivated by kernelGAN [9], our LR image generator is a linear network without any nonlinear activation, which downscales images only by convolution and subsampling. The generated LR image will be extracted representation by the degradation encoder to enable the degradation feature in it to be as similar as the real world. Specifically, given real-world images x as LR and unpaired high-resolution images y as input, the first stage of our framework is to obtain the real-world degradation representation q, k+, k− from two groups of images by contrastive learning, as the yellow arrow pointed out in Fig. 1. Once the initial degradation learning is finished, we use a linear network to generate paired LR images $\hat{x}$ by downsampling y. Then, we encode $\hat{x}$ to ensure that it should have the same degradation as x. This process is marked by the blue arrow in Fig. 1. Our final goal is to use the generated samples that are used to train SR models. The constructed method tries to recover corresponding HR images $\hat{y}$ and make it approximate y, as shown by the green arrow in Fig. 1. In the testing phase, we only use the SR network to obtain the SR image and evaluate their quality by calculating the PSNR and SSIM. Different from previous work [10], our real-world degradation is learned from the LR and HR datasets and the LR image generated by our downscaling network instead of a fixed operator.

3.2 Degradation Model

The degradation model is the second part, which learns to obtain more realistic LR images by an unsupervised method. We extract the degradation representations from LR images using a contrastive learning framework. We assume that the degradation representation in real-world domain images is similar and is distinguished from high-resolution domain images. We randomly select patches from real-world images as the query patch, and the positive patches come from the same dataset. Other patches extracted from HR images are regarded as the negative patches. Then, all the query, positive and negative patches are encoded as degradation representations by a convolutional network model, which is shown in Fig. 2(a). SimCLR [11] pointed out that the representations need to go through a three-layer fully connected network to obtain q, k^+, and k^-. Contrastive learning aims to make q and k^+ more similar and keep q away from k^-. Following MoCo [8], we use InfoNCE loss to measure the similarity, which can be formulated as follows:

$$\text{Loss}_q = -\log \frac{e^{q \cdot k^+/\tau}}{\sum_{\text{i}=0}^{k} e^{q \cdot k_1^-/\tau}} \tag{3}$$

in which K is the number of negative samples, $\cdot$ is the dot product and τ is a hyper-parameter. Previous contrastive learning methods [6] mentioned that a large number of

negative samples is essential for the model to obtain a good representation. A queue with negative samples is maintained for learning in the degradation model. During the training phase, we first randomly extract N patches from real-world datasets and divide two patches per group. Then, these N patches are encoded into q_i, k_i^+ by the degradation encoder. For negative samples, we also encode the HR images into k_i^- to update the queue, where i denotes the position of the i-th image. We define the total loss as follows:

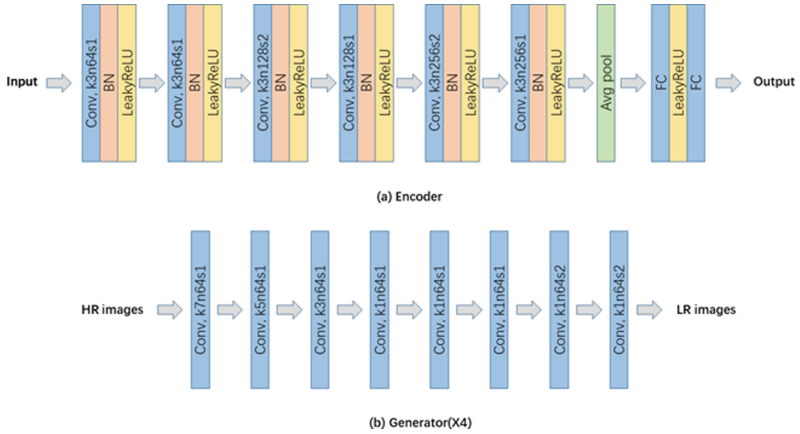

Fig. 2. Architecture of Encoder and Generator Network. The 'k', 'n' and 's' in each layer indicate the kernel size, number of channels and stride size, respectively.

$$Loss_{deg} = \sum_{i=0}^{N/2} -log \frac{e^{q_t - k_i^+/\tau}}{\sum_{j=0}^{k} e^{q_i - k_t^-/\tau}} \tag{4}$$

where S is the size of the queue. Figure 2(a) shows the architecture of the degradation encoder. We adapt a similar encoder as the work of DASR [10]. Specifically, we assemble the convolutional layer with 3 filters, batch normalization (BN) [12] and LeakyReLU [13] layers with a negative slope of 0.1. Note that the average pooling output is embedding. The final multilayer perceptron consists of two fully connected networks and one LeakyReLU layer. The characters 'k', 'n' and 's' indicate the parameters of kernel size, numbers of different channels and the size of each stride. For instance, k7n128s2 means that there are 128 filters in the convolutional layer, the kernel size is 7, and the stride is 2. For the LR image generator, inspired by kernelGAN [9], we also design a linear model that does not contain any activation layer that is more in line with the degradation equation. This is consistent with the degradation model equation mentioned before, since downscaling by blur kernel is a linear operation applied to LR images. The generator architecture is shown in Fig. 2(b). There are eight convolutional layers with 64 channels each. The first three kernel sizes are 7, 5, and 3, and the rest are 1. The last two layers refer to the downscaling operator, whose scale factor is 4. The whole network can be regarded as a single convolutional layer with a 15×15 receptive field. The reason why

we design a multilayer network instead of one convolutional layer is that gradient-based optimization is more efficient for deep linear networks than only one layer, as kernel-GAN [9] mentioned. We fine tune the pretrained encoder as a discriminator to ensure that the generated LR image can obtain the same degradation as the real-world image. The color loss and the adversarial loss aim to maintain the basic structure information of the original image.

3.3 Reconstruction Model

Based on SRGAN, we implement an SR model trained by paired data $\hat{x}, y$. The network adopts the architecture of the generator network in SRGAN, and the resolution of the SR image will be enlarged 4 times. We apply pixel loss and perceptual loss [14] during training. The pixel loss uses the L1 distance, which is calculated as:

$$Loss_1 = \frac{1}{S^2 WH} \sum_{i=0}^{sW} \sum_{j=0}^{sH} \left\| X_{i,j}^{hr} - F\left(Y_{i,j}^{lr}\right) \right\| \tag{5}$$

in which s is the scale factor, W, H is the width and height of the HR image and $I_{i,j}^{HR}$ describes the pixel value of the image. This is the most widely used loss function for image SR. However, there is the problem that solutions of the L1 regularized method often tend to lack high-frequency content, so we add a perceptual loss to obtain a sharper texture. The perceptual loss uses the inactive features of VGG-19 [15], which benefits the image vision quality:

$$Loss_{per} = \frac{1}{W_{i,j} H_{i,j}} \sum_{m=0}^{W_{i,j}} \sum_{n=0}^{H_{i,j}} \left(\phi_{i,j}\left(I^{HR}\right)_{m,n} - \phi_{i,j}\left(F\left(I^{LR}\right)\right)_{m,n} \right)^2 \tag{6}$$

where $\phi_{i,j}$, $W_{i,j}$, and $H_{i,j}$ indicates the feature map created by the j-convolution before the i-maxpooling of the VGG-19 network. The total loss can be calculated as the weighted sum of these two different losses as follows:

$$Loss_{\text{total}} = \lambda_1 \cdot Loss_1 + \lambda_2 \cdot Loss_{per} \tag{7}$$

where λ_1 and λ_2 are set as 1 and 0.1.

4 Experiments

4.1 Training Data

We train the proposed model in the NTIRE-2020 dataset. There are 2650 degraded images that can be regarded as real-world images, 800 high-resolution images, and 100 paired validation images in Track 1. The 2650 degraded images are not paired with the 800 high-quality images and do not exist in the same image. We randomly select 800 images from the degraded image and 800 unpaired high-resolution images as training data during the degradation learning phase.

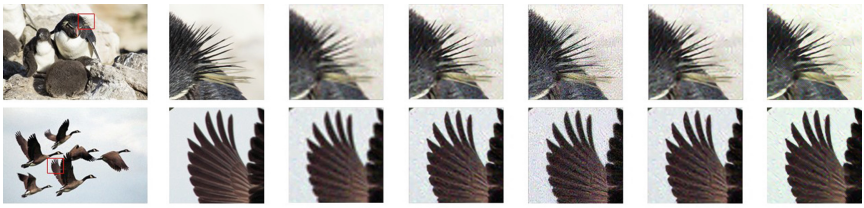

Fig. 3. Qualitative comparison with state-of-the-art blind methods on the NTIRE 2020 Real World SR challenge Track 1 validation set (SR scale $\times 4$).

4.2 Training Details

As in Fig. 1, we split our training phase of the superresolution process into three parts. We first train the degradation learning network. During training, 64 h patches of size 192×192 and size 48×45 are images that are cropped from high-resolution images. Sixty-four LR patches of size 48×48 are randomly cropped from low-resolution images. In detail, we set τ and S in Eq. 4 to 0.06 and 8192. The model is optimized by the Adam optimizer, in which $\beta_1 = 0.5$ and $\beta_2 = 0.999$ without weight decay for 600 epochs. The learning rate is initialized as 1×10^{-3}. It will decrease 0.1 every 500 epochs. Given a trained encoder, we train the downscaling network with the same optimizer parameters for 100 epochs. Then, we train the reconstruction model and pretrained dowscaling network using generated paired LR-HR images for 300 epochs. The learning rates of the downscaling network and SR model are initialized to 1×10^{-4} and decreased by 0.5 every 100 epochs. The minibatch size of all networks is set to 32. Note that the patch size of the image and the network layers depend on the scale parameter, which is 4 in our experiments. We implement the proposed method on NVIDIA TITAN XP GPUs in the PyTorch platform, and it takes approximately two days and a half to train our model.

4.3 Training Details

We prove the effectiveness of our approach with current good methods in the same field: SRFBN [16], ESRGAN [17], EDSR [18], Impressionism [19] and DBPN [20]. Table 1 displays different parameters, such as the average PSNR, and SSIM values of the NTIRE2020 real-world SR Track1 validation set with different methods trained with clean LR images downscaled from HR images. Our methods outperform the previous methods. This shows that EDSR and SRFBN do not achieve good performance if the degradation is unknown in the training phase. The unsupervised ESRGAN enhances the noise and degradation, leading to poor quality of the SR image. Note that the Impressionism method makes more effort on so that the PSNR is lower than others. Our approach is much better than those other methods in both PSNR and SSIM, which may be because the model is trained on paired degradation image data. Several subjective results are illustrated in Fig. 3.

To validate the advantages of our model in solving real-world SR tasks, we apply the ESRGAN generator in our SR model, which is named RRDB.

Table 1. Quantitative results for the NTIRE 2020 real world SR challenge Track 1 validation dataset

Methods	PSNR	SSIM
Bicubic	25.48	0.680
EDSR	25.36	0.640
SRFBN	25.37	0.642
ESRGAN (Unsupervised)	19.04	0.242
Impressionism	24.82	0.662
DBPN	24.51	0.701
Ours	**25.50**	**0.738**

Table 2. Quantitative results for the NTIRE 2020 Real World SR challenge Track 1 validation dataset, comparing the ESRGAN (Supervised) and ours (RRDB)

Methods	ESRGAN (Supervised)	Ours (RRDB)
PSNR/SSIM	24.74/0.695	24.98/0.6873

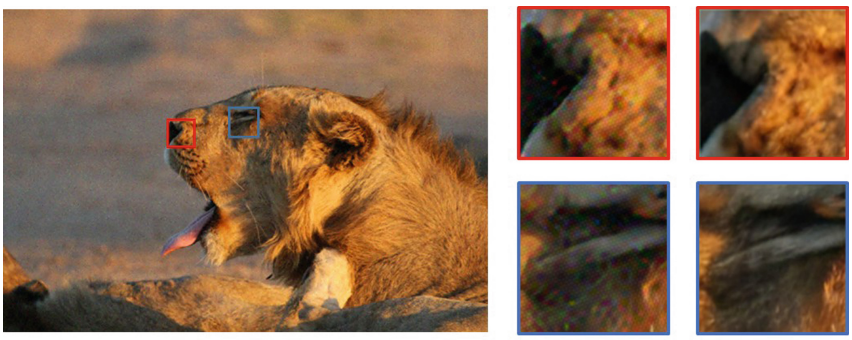

Fig. 4. Qualitative comparison between ESRGAN (Supervised) and our method (RRDB) for Track 1 (SR scale $\times 4$).

Table 2 shows a comparison with the supervised ESRGAN. The supervised ESRGAN is trained with real paired data provided by the NTIRE 2020 official baseline. We change the SR network into the RRDB but keep the same setting of the loss functions. The whole training process is also the same. As shown by the experimental results, our framework outperforms the supervised ESRGAN in PSNR and makes the SSIM value close to it. The quality results are illustrated in Fig. 4.

5 Conclusion

We propose a novel framework assisted by image degradation learning. In contrast to existing methods that downscale HR images to obtain LR images by a fixed operation, we acquire degradation representations of real-world LR images that assist the downsampling network in generating LR images with the consistent domain using contrastive learning. This assists us in obtaining more realistic and paired image data for the later reconstruction network. Experiments on NTIRE2020 datasets show the effectiveness of our approach.

References

1. Dong, C., Loy, C.C., He, K., Tang, X.: Image superresolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. **38**(2), 295–307 (2015)
2. Cai, J., Zeng, H., Yong, H., Cao, Z., Zhang, L.: Toward real-world single image superresolution: a new benchmark and a new model. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3086–3095 (2019)
3. Xu, X., Ma, Y., Sun, W.: Toward real scene superresolution with raw images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1723–1731 (2019)
4. Lugmayr, A., Danelljan, M., Timofte, R.: Unsupervised learning for real-world superresolution. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 3408–3416. IEEE (2019)
5. Yuan, Y., Liu, S., Zhang, J., Zhang, Y., Dong, C., Lin, L.: Unsupervised image superresolution using cycle-in-cycle generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 701–710 (2018)
6. Gu, J., Lu, H., Zuo, W., Dong, C.: Blind superresolution with iterative kernel correction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1604–1613 (2019)
7. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
8. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738 (2020)
9. Bell-Kligler, S., Shocher, A., Irani, M.: Blind superresolution kernel estimation using an internal-GAN. arXiv preprint arXiv:1909.06581 (2019)
10. L. Wang, Y. Wang, X. Dong, Q. Xu, J. Yang, W. An, and Y. Guo, "Unsupervised degradation representation learning for blind superresolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10 581–10 590 (2021)
11. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning. PMLR, pp. 1597–1607 (2020)
12. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. PMLR, pp. 448–456 (2015)
13. Maas, A.L., Hannun, A.Y., Ng, A.Y., et al.: Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the ICML, vol. 30, no. 1, p. 3. Citeseer (2013)

14. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
16. Li, Z., Yang, J., Liu, Z., Yang, X., Jeon, G., Wu, W.: Feedback network for image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3867–3876 (2019)
17. Wang, X., et al.: ESRGAN: enhanced super-resolution generative adversarial networks. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11133, pp. 63–79. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11021-5_5
18. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image superresolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 136–144 (2017)
19. Ji, X., Cao, Y., Tai, Y., Wang, C., Li, J., Huang, F.: Real-world superresolution via kernel estimation and noise injection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 466–467 (2020)
20. Haris, M., Shakhnarovich, G., Ukita, N.: Deep back-projection networks for superresolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1664–1673 (2018)