Pattern Analysis and Applications (2023) 26:875–888
https://doi.org/10.1007/s10044-023-01150-2

SHORT PAPER

DDNSR: a dual-input degradation network for real-world super-resolution

Yizhi Li[1] · Haixin Chen[1] · Tao Li[1] · Binbing Liu[1]

Received: 28 April 2022 / Accepted: 28 January 2023 / Published online: 18 February 2023

Abstract
Recently, Real-World Super-Resolution has become one of the most popular research fields in the scope of Single Image Super-Resolution, as it focuses on real-world applications. Due to the lack of paired training data, developing real-world super-resolution is considered a more challenging problem. Previous works intended to model the real image degradation process so that paired training images could be obtained. Specifically, some methods attempt to explicitly estimate degradation kernels and noise patterns, while others introduce degradation networks to learn maps from high-resolutions (HRs) to low-resolutions (LRs), which is a more direct and practical way. However, previous degradation networks take only one HR image as an input and therefore can hardly learn the real sensor noise contained in LR samples. In this paper, we propose a novel dual-input degradation network that takes a real LR image as an additional input to better learn the real sensor noise. Furthermore, we propose an effective self-supervised learning method to synchronously train the degradation network along with the reconstruction network. Extensive experiments showed that our dual-input degradation network can better simulate the real degradation process, thereby indicating that the reconstruction network outperforms state-of-the-art methods. Original codes and most of the testing data can be found on our website.

Keywords Real-world super-resolution · Degradation network · Self-supervised learning · Deep learning

1 Introduction

Single Image Super-Resolution (SISR), which aims to reconstruct a high-resolution (HR) image from its low-resolution (LR) counterpart, has wide usage in many real-world applications that require high-quality images. Substantial progress in deep-learning-based SISR algorithms have been made in recent years. In general, researchers first collect a training dataset containing thousands of LR-HR image pairs and then use the typical supervised learning method to train a predesigned deep neural network (for reconstruction). After training, the prior information is learned and stored in the weights of the network. The training LR-HR image pairs should be carefully chosen to achieve better SR performance, as deep learning is known to be a data-driven method. First, a pair of training LR-HR images should be highly matched (in the case of image semantics). Second, the training LR image should be close enough to the real LR input (in real SR applications).

To collect ideal training LR-HR image pairs, the first group of researchers [1–4] applied a predefined degradation algorithm (e.g., bicubic down-sampling with additive Gaussian noise) on HR samples. The LR outputs are close enough to the HR inputs in the case of pixel value, so they can easily achieve a high PSNR[1] value. However, the degradations in real-world LR images are far more complicated than an artificial bicubic down-sampling algorithm. Consequently, those SISR models trained on simulated data can only generate ideal results on these synthetic LR samples, while they have difficulty performing well on real-world LR inputs. Then, Cai et al. [5] used the same camera but different focal lengths for the same scene (with an image pair registration algorithm to correct the unexpected mismatch between image pairs) to collect their training dataset. They captured many real-world LR-HR image pairs; however, there are far more than two categories in which real-world

✉ Binbing Liu
liubinbing@hust.edu.cn

[1] School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan 430074, China

[1] Peak Signal-to-Noise Ratio (PSNR) is a classical metric calculated from the L_2 distance between two images.

Fig. 1 Visual comparison among the ESRGAN [1] (trained on synthetic LR images), Impressionism [2] model (trained on explicitly constructed LR images), and our DDNSR model (trained on realistic LR images generated by our dual-input degradation network) on a real-world LR image of branches and leaves, where the Ground Truth (GT) data are not available (N/A). Our SR result is both clean and detailed, while ESRGAN fails to clean the noise, and the Impressionism model fails to reconstruct the lost details. The Impressionism model tends to generate high-contrast results, which is not the central purpose of super-resolution (contrast enhancement could be done by some simple post-processing algorithms). In the context of Image Super-Resolution, the key technical challenge is restoring lost details. It is notable that these three different models used the same reconstruction network and applied the same training strategy

LR images can be inputted (all images in their dataset were captured by Canon 5D3 and Nikon D810 cameras). In other words, their images are all captured by high-quality cameras, but our real LR inputs (in real-world SR applications) are always obtained from different low-quality sensors (with different kinds of noise).

Real-World Super-Resolution (RealSR), which aims to close the large gap between synthetic LR images and real LR images, has received great attention in recent years. Since real-world fully matched LR-HR image pairs are unavailable, researchers are trying to learn super-resolution from unpaired data. Generally, there are two individual datasets for training: a LR dataset captured by real-world low-quality cameras (for simulating the real degradation process) and a HR dataset captured by high-quality cameras (for supervised training). Furthermore, since supervised learning methods have already achieved impressive performance (such as ESRGAN [4]), researchers usually concentrate on how to obtain their training LR images closer to real cases (using those two individual datasets). In Fig. 1, we compared our model with two state-of-the-art methods. ESRGAN [4] (the champion of the PIRM 2018 SR challenge [6]) was trained on completely synthetic LR images, and the Impressionism model [7] (the champion of the NTIRE 2020 RealSR challenge [8]) explicitly estimated the degradation kernel and collected noisy patches from the LR dataset to construct realistic LR images for fully supervised training. Unlike either of those two, our DDNSR model was directly trained on unpaired data in a self-supervised way. It is notable that all three models used the same reconstruction network (RRDB-net proposed in [4]) and applied the same training strategy (such as loss function and learning rate). Thus, the SR results are very different, as deep learning methods are known to be very sensitive to training data.

To obtain our training LR images that are similar to real cases, we have to estimate two main pieces of information in real LR samples: the real down-sampling map and the real noise pattern. Then, we apply them to our HR samples to obtain training LR-HR image pairs. According to this, state-of-the-art methods can be divided into two parts.

By explicit estimation Ji et al. [7] intended to explicitly estimate the blur kernel (using a particular kernel-estimation algorithm [9]) and collect the real noisy patch from LR samples. On the one hand, they won the NTIRE 2020 RealSR challenge [8] on both tracks with significant outperformance. On the other hand, it is a rather tricky approach since it is difficult to tell how similar the constructed LR image is to the real LR image. Moreover, it is found that a tiny disturbance could lead to severe global noise [10], which indicates that those explicit construction approaches may not be trustworthy for real-world applications.

Through implicit learning Compared with explicit estimation, a more direct and practical method will be introducing a particular degradation network to implicitly learn a map from HR (clean and detailed) to real LR (usually noisy). Regarding this concept, previous works [11–14] first designed a single-input (only one HR sample is taken as input) CNN-based network, then adapted a cycle-consistency loss (proposed in CycleGAN[2] [15]) to synchronously train the degradation network along with the reconstruction (super-resolution) network. Based on our observation, degradation training and super-resolution training are very different. Due to the mathematical

[2] CycleGAN [15] was designed to solve the Image-to-Image Translation problem, which is very different from Image Super-Resolution. However, their idea of cycle-consistency loss is so inspiring that many Real-SR methods adopt this idea (the implementations are very different).

characteristics of convolution, a single-input CNN-based network used for degradation or reconstruction can easily learn a map from a noisy input to a clean output, but it can hardly learn the map from a clean input to a noisy output (we will show this phenomenon in the appendix). This means that those previous single-input degradation networks can only learn the real down-sampling map on some level but can hardly learn the real noise pattern contained in real LR samples.

In this paper, we proposed a novel dual-input degradation network, taking a real LR image (noisy) as an additional input, to collect the real noise from this noisy sample. We synchronously trained our degradation network along with our reconstruction network (there is no special requirement for the reconstruction network) in a self-supervised way. Experiments show that the dual-input degradation network significantly outperforms single-input degradation networks, therefore leading to much better SR results. Moreover, we found that different types of training HR samples significantly affected the final SR results. Specifically, closer image semantics between training LR-HR image pairs will provide more realistic details to the final SR results. In summary, our contributions are twofold.

1. We evaluated the disadvantages of current state-of-the-art RealSR methods and proposed an effective dual-input degradation network, along with an efficient self-supervised learning method. Experiments showed that our method outperforms state-of-the-art RealSR methods.
2. For the first time, we found and validated that training with semantically similar LR-HR image pairs could help our model restore more lost details. This is a very important feature for designing real-world SR applications.

2 Related works

Since the pioneering work of SRCNN [1], many new network architectures and new training strategies have been proposed to improve the SR performance. For example, Zhang et al. [16] proposed a very deep residual channel attention network (RCAN) to improve the representational ability of CNNs. Ledig et al. [3] introduced a generative adversarial network (GAN) [17] to obtain more visually realistic SR results (rather than higher PSNR/SSIM values). However, these methods are only trained and tested on synthetic LR images but can hardly perform well on real-world LR inputs. To apply our SR model to real-world applications, Cai et al. [5] built a real-world dataset in which paired LR-HR images on the same scene were captured by adjusting the focal length of a digital camera (with a carefully designed image pair registration algorithm). Their method generalizes well to other camera devices, such as Sony a7II and mobile phones (with high-quality sensors). However, real-world LR images are far more complicated than several digital cameras. If we want to apply their method to different SR applications (such as video surveillance), we must collect a completely new dataset using some particular cameras, which is too impractical to meet our needs.

In real-world SR applications, different cases indicate very different degradation processes (down-sampling operations and noise patterns). We urgently need a method to simulate the specific degradation process. Generally, the degradation process can be described as follows:

$$I_{LR} = (I_{HR} \otimes k_{real}) \downarrow_s + n \tag{1}$$

where "$\otimes$" represents convolution, "k_{real}" with "$\downarrow_s$" denotes the real down-sampling operation, and "n" is an additive noise. To achieve Eq. (1), Ji et al. [7] explicitly estimated the blur kernel and noise pattern from real LR samples. It could be very tricky since it is difficult to conclude how close the estimation results and the real LR samples are (the only way is to evaluate the SR results afterward).

Instead, using a particular degradation network to learn the real degradation process could be a more direct and practical method. Specifically, we can estimate the blur kernel and noise pattern through end-to-end learning. This type of method can be considered self-supervised learning[3] method since the degradation outputs will be used for supervised training. To train this degradation network, many previous works adapted a cycle-consistency framework as proposed in CycleGAN [15]. In particular, Lugmayr et al. [11–14] used a bicubic down-sampling to downscale HR images first and then performed the transformation (from clean LR to noisy LR) only in LR space. Since bicubic down-sampling is a predesigned algorithm that is very different from the real case, it may lose too much useful information, therefore increasing the learning burden. In contrast, Zhao et al. [18] proposed a DNSR model, which directly regarded the degradation network and the reconstruction network as the paired "G and F" in CycleGAN [15]. It is a simpler and more efficient method to adapt to the cycle-consistency loss (without the prior bicubic down-sampling process). We used the same outside architecture as DNSR, but the inner details are very different. First, DNSR added the degradation loss and reconstruction loss together, while we divided them. Second, their reconstruction loss was mainly L_1 loss, while ours was adopted from ESRGAN [4] and could be replaced

[3] Self-supervised learning could be considered a form of unsupervised learning. In this paper, we prefer to use the word "self-supervised learning" rather than "unsupervised learning" as used in [11–13], since the super-resolution training will still be supervised.

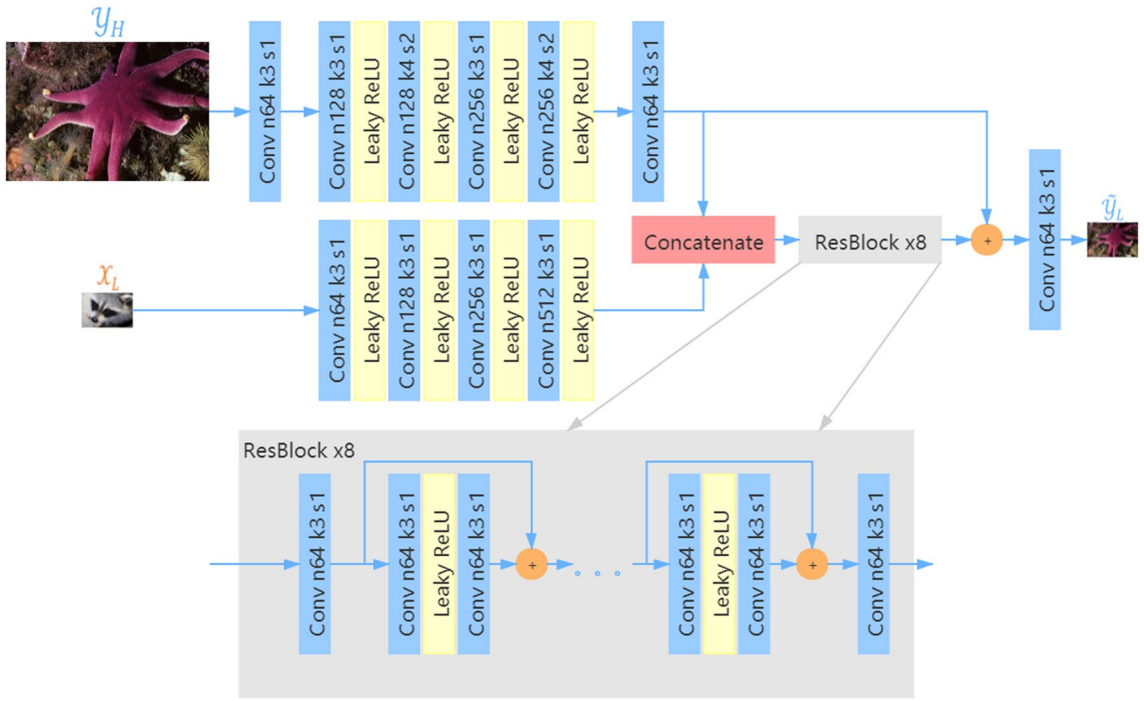

Fig. 2 The architecture of our proposed dual-input degradation network. For the two inputs $\mathcal{Y}_H$ and $\mathcal{X}_L$, we first extract their features using a series of convolution layers; then, we concatenate these two feature matrices and send them to 8 residual blocks. Finally, we add the residual output on the down-sampled $\mathcal{Y}_H$ to obtain the final output $\widetilde{\mathcal{Y}_L}$. Symbols such as $\mathcal{Y}_H$ are described in the first paragraph of Sect. 3

with any type of different super-resolution losses. Third, the most vital difference between previous works and ours is the degradation network architecture. To the best of our knowledge, we are the first to use a dual-input degradation network, which takes both HR and LR samples as inputs, while others take only one HR sample. Experiments show that the additional real LR sample is indispensable for learning the real additive noise (mostly sensor noise). With respect to the DNSR model [18], we name our model "D-DNSR" or "DDNSR", in which the prefix "D" stands for "Dual-input".

NTIRE 2020 Challenge on Real-World Super-Resolution The NTIRE 2020 challenge [8] promoted a benchmark protocol and dataset and probed the current state-of-the-art in the RealSR field. Dozens of different RealSR methods were proposed in this challenge. For a fair comparison, we trained and evaluated our model based on this challenge.

The challenge contains two tracks. Both tracks have the goal of upscaling with factors 4×4 and require training on unpaired images. For the training and testing LR images, Track1 used a synthetic down-sampling operation (bicubic down-sampling) and artificial noise injection (Gaussian noise) applied to HR samples, while Track2 used real LR images captured by an iPhone3 camera. Since there is no persuasive metric to evaluate the perceptual quality of the generated images, the challenge used Mean-Opinion-Score (MOS) for Track1 with ground truth and used Mean-Opinion-Rank (MOR) for Track2 without ground truth data. Both MOS and MOR are evaluated through human eyes. In this paper, we also focus on the direct perceptual quality, some quantitative results (classic PSNR, SSIM, and LPIPS [19]) are shown in the appendix. Moreover, we will mainly focus on the method of the champion team, "Impressionism" [7], as they significantly outperform other competitors by large margins. It is notable that the Impressionism team is the only team that aimed to explicitly estimate the blur kernel in the image for improved source data generation, according to the challenge organizers [8].

3 Proposed method

In this section, we will comprehensively demonstrate our proposed method. For better understanding, we used $\mathcal{X}$ to refer to the source dataset and $\mathcal{Y}$ to refer to the target dataset, with subscripts of $_L$ or $_H$ (e.g., $\mathcal{X}_L$ or $\mathcal{Y}_H$) to describe low resolution and high resolution, respectively. As shown in Fig. 2, we also used superscript $\widetilde{}$ or $\widehat{}$ (e.g., $\widetilde{\mathcal{X}_L}$ or $\widehat{\mathcal{Y}_L}$) to refer to the middle results or final outputs, respectively.

3.1 Degradation process

According to Eq. (1), the ideal degradation process can be divided into two parts:

Real down-sampling operation Lugmayr et al. [11] and Chen et al. [12] used bicubic down-sampling to simulate the real down-sampling operation and only performed the transformation in LR space. This will increase the burden of the transformation since much useful information was lost during the down-sampling process. Ji et al. [7] used explicit kernel estimation [9] to simulate the real down-sampling operation, which will, to some extent, introduce an additional error (caused by the predefined kernel estimation algorithm). In contrast to the above, the real down-sampling operation will be jointly learned inside our degradation network, which is simpler (without the prior down-sampling process) and more effective (less error accumulation).

Real additive noise The additive noise in real LR images is mainly caused by low-quality sensors, which have little relation with the target $\mathcal{Y}_H$. Ji et al. [7] tried to directly collect noisy patches from the source $\mathcal{X}_L$ with a predefined filter algorithm, which may not be reliable since they do not have a clear reference to adjust their filter threshold. Other unsupervised/self-supervised methods [11–13, 18, 20] tried to learn this sensor-noise through a single-input CNN-based network. This indicates that the sensor-noise was completely extracted from $\mathcal{Y}_H$, which will severely increase the learning burden. To overcome these disadvantages, we introduce the source as an additional input $\mathcal{X}_L$ for our degradation network, which aims to efficiently extract the sensor-noise from $\mathcal{X}_L$.

3.2 The dual-input degradation network

Based on the analysis above, we design our degradation network as two parts: real down-sampling operation, achieved by convolution layers with strides set to 1 and 2, alternatively; additive noise, achieved by a noise extraction block, taking down-sampled target $\mathcal{Y}_H$ along with the feature matrix of source $\mathcal{X}_L$ concatenated as input (Fig. 2).

For learning the real LR distribution, the additive-noise will be trickier (from experience), so we used 8 residual blocks to learn, while for the down-sampling part we used only 4 vanilla convolution layers. Moreover, we concatenated these two parts and added the residual output to the down-sampled result. Thus, our proposed degradation network could be considered a dual-input degradation network taking both HR (target) and LR (source) samples as inputs.

3.3 The framework of our self-supervised learning method

As shown in Fig. 3, we take both the target and source as inputs and synchronously train the reconstruction network along with the degradation network. For the reconstruction part, we directly use the RRDB-net proposed in ESRGAN [4]. Furthermore, we use the identical loss function for training, which is:

$$L_{sr} = L_{perceptual} + 0.005L_{adversarial} + 0.01L_1 \tag{2}$$

For the degradation network, the total loss function can be described as:

$$L_{degradation} = \lambda_1 L_{cont} + \lambda_2 L_{adv} + \lambda_3 L_{cycle} \tag{3}$$

where λ_1, λ_2, and λ_3 are trade-off hyperparameters for balancing the three types of loss. In the context of deep-learning, those hyperparameters will be fine-tuned through the training process.

To explain our idea on degradation network training, we describe a feature loss as follows:

$$L_{feature} = \|F_s(x_{real}) - F_s(x_{fake})\|_2^2 \tag{4}$$

where F_s denotes a particular feature extractor with a downscale factor s. The purpose of Eq. (4) is to maintain the image semantics of the generated sample while maintaining space to learn the noise pattern (the downscale factor s leads to a map from s^2 to 1). Based on Eq. (4), we design three parts of the loss function to achieve three different purposes:

Content Loss Since the degradation outputs will be used as supervised inputs for Eq. (2), we must ensure $\mathcal{Y}_L$ matches $\mathcal{Y}_H$ in the case of image semantics. To achieve this, we first used the average pooling to downscale $\mathcal{Y}_H$ and then sent $\mathcal{Y}_L$ with $\widetilde{\mathcal{Y}_L}$ to the feature extractor to calculate $L_{feature}$, as shown in Fig. 3.

A good feature extractor will be able to mainly keep the useful high-frequency information while losing the unnecessary low-frequency part. In the scope of RealSR, we use VGG-34 (the 4th convolution layer's output before the 3rd max-pooling layer, as mentioned in [3], the "VGG-22" below will have the same meaning) from a pretrained VGG19 [21] network as an effective feature extractor. It is notable that this type of feature extractor has a downscale factor $s = 4$.

Adversarial Loss As used in [18, 20], the adversarial loss is used for transferring $\mathcal{Y}_L$ to $\mathcal{X}_L$ in the case of noise distribution. We define our adversarial loss as follows:

$$L_{disc} = -\log(Disc(x_{real})) - \log(1 - Disc(x_{fake})) \tag{5}$$

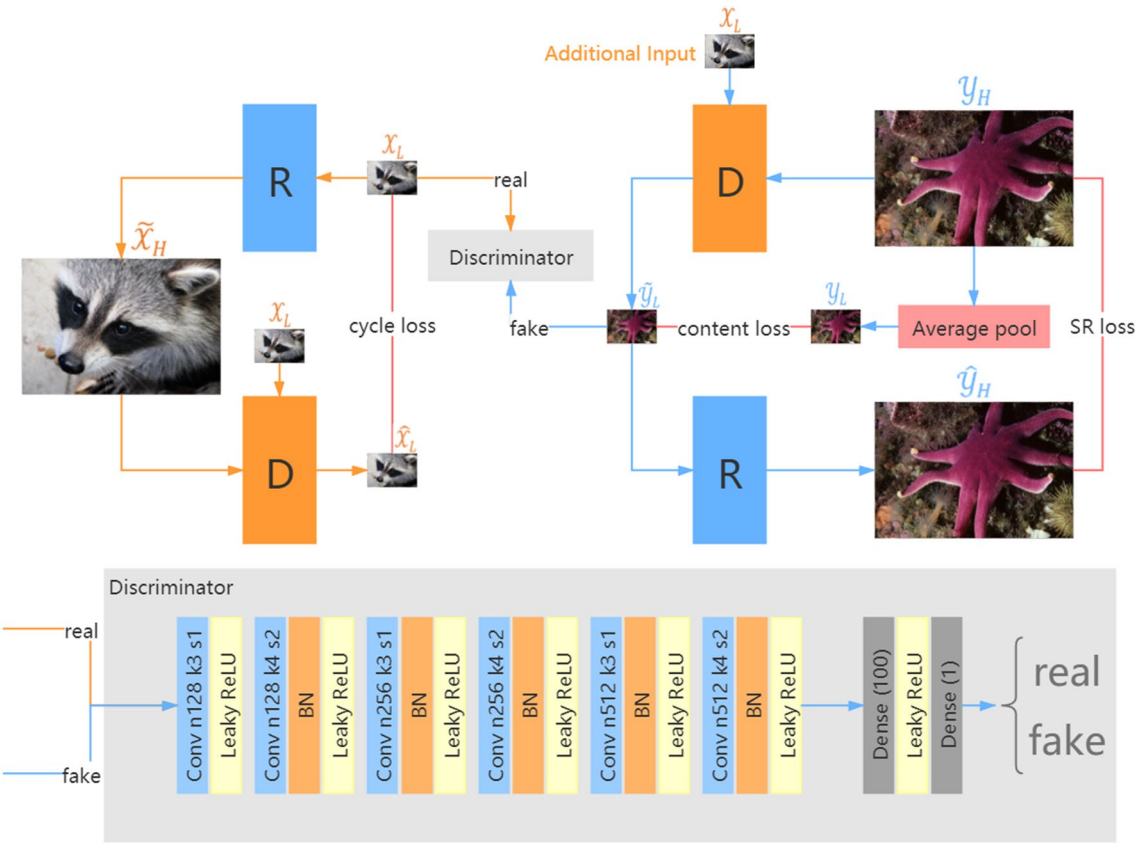

Fig. 3 The framework of our proposed self-supervised learning method, where R and D represent the reconstruction network and degradation network, respectively

$$L_{gen} = -\log(Disc(x_{fake})) \tag{6}$$

Cycle Loss To boost the total training procedure, we add a cycle-consistency loss [15] based on Eq. (4). Our cycle-consistency loss considers "D and R" (degradation network and reconstruction network) instead of "G and F", as mentioned in CycleGAN [15] and other self-supervised/unsupervised methods [11–13]. In the scope of RealSR, we use VGG-22 from a pre-trained VGG19 network as an effective feature extractor. It is notable that this feature extractor has a downscale factor $s = 2$.

4 Experiment

4.1 Implementation details

We trained our model under the TensorFlow framework with a single NVIDIA GeForce RTX 3090 GPU. We adopted the RRDB-net [4] as our reconstruction (super-resolution) network and synchronously trained the reconstruction network along with the degradation network using the same training schedule as proposed in ESRGAN [4]. The initial learning rate is set to 1.0×10^{-4} and then divided by a factor of 2 after 50k, 100k, 200k, and 300k mini-batches of training. The batch size was set to 16, which is also the same as in ESRGAN. For both tracks, the hyperparameters λ_1, λ_2, λ_3 were set to be the same, which are 0.1, 1.0, and 10.0, respectively. To minimize the implementation differences of the reconstruction network, we used a VGG-style [3] discriminator for Track1 and a PatchGAN [9] discriminator for Track2. Thus, our reconstruction setting is identical to the Impressionism model [7].

4.2 Datasets

DIV2K [22], This dataset is a real-world high-quality dataset containing many different types of real-world scenes, along with their low-quality image counterparts achieved by the synthetic degradation procedure. We use $\mathcal{Y}_{DIV2K}$ and $\mathcal{X}_{DIV2K}$ to refer to the HR and LR parts in the DIV2K dataset, respectively.

Flikr2K [23]. Source images on Track1 are obtained from the Flikr2K [23] dataset, which is first down-sampled using a bicubic algorithm and then artificially added with Gaussian noise to simulate real sensor noise. The

Table 1 Comparison of the Training Unpaired Datasets Among the Three Models

	ESRGAN	Impressionism	DDNSR (Ours)
Track1-Target	$\mathcal{Y}_{DIV2K}$	$\mathcal{Y}_{DIV2K} + \mathcal{X}^{clean}_{Flikr2K}$	$\mathcal{Y}_{DIV2K}$
Track1-Source	$\mathcal{X}_{DIV2K}$	$\mathcal{X}_{Flikr2K}$	$\mathcal{X}_{Flikr2K}$
Track2-Target	$\mathcal{Y}_{DIV2K}$	$\mathcal{Y}_{DIV2K} + \mathcal{X}^{clean}_{DPED}$	$\mathcal{Y}_{DIV2K} + \mathcal{X}^{clean}_{DPED}$
Track2-Source	$\mathcal{X}_{DIV2K}$	$\mathcal{X}_{DPED}$	$\mathcal{X}_{DPED}$

semantic information in this dataset is very close to those in DIV2K [22]. We use $\mathcal{X}_{Flikr2K}$ to refer to this dataset.
DPED [24]. Original images in DPED [24] are captured synchronously by four different cameras: iPhone 3GS, BlackBerry Passport, Sony Xperia Z, Canon 70D DSLR, with no preprocessing at all. These images are unmatched and do not have a 4× scale relation. Therefore, they cannot be directly used for supervised training. Moreover, there is a large semantic gap between the DPED dataset and DIV2K dataset, which makes unpaired training very difficult. We use $\mathcal{X}_{DPED}$ and $\mathcal{Y}_{DPED}$ to refer to images captured by iPhone 3GS and Canon 70D DSLR, respectively.
Clean-up Method [7]. To skip the large semantic gap between the target and source, the Impressionism [7] team used a clean-up method to obtain more target images. In particular, they use a bicubic algorithm with a down-scale factor $s = 2$ to down-sample the source images, and directly use the outputs as the target. It could be helpful to some extent. We use $\mathcal{X}^{clean}_{Flick2K}$ and $\mathcal{X}^{clean}_{DPED}$ to refer to these two datasets.

4.3 Models for comparison

We compared our results with two different types of state-of-the-art methods. Table 1 shows the differences in their training unpaired datasets.

Supervised Model Training on Synthetic LR Images ESRGAN [4] was the champion of the PIRM 2018 SR [6] challenge, and we use their pretrained model released by the author as a representative model for all the traditional SR models trained on synthetic LR images.
Supervised Model Training on Explicitly Constructed LR Images The Impressionism team was the champion of the NTIRE 2020 RealSR [8] challenge and was the only team that explicitly constructed LR-HR image pairs for supervised training. Since they outperform other models by large margins, we will mainly focus on their results, using their released pretrained models.
Other self-supervised/unsupervised learning methods Since we did not find any available pretrained model for the NTIRE 2020 RealSR [8] challenge, we only evaluated those methods in the ablation study. All other methods used single-input networks to achieve either transformation [11–13] or degradation [18, 20]. We will compare the effectiveness of the additional input $\mathcal{X}_L$ in Sect. 4.5.

4.4 Perceptual results

Track1 on the Synthetic Corrupted LR Images We compared the perceptual results using the validation dataset containing 100 noisy LR images with a ground truth image as a reference. As shown in Fig. 4, both our DDNSR model and the Impressionism model can successfully generate a high-quality clean image with many reconstructed details, while ESRGAN can hardly clean the noise. The results of the Impressionism model still contain a little noise if we look at it carefully. This might be caused by their clean-up method, which uses a bicubic down-sampling operation to clear the noise in LR images and then takes the "clean" LR images as the target for training. The cleaned images were not completely clean, as the bicubic algorithm is not an ideal clean-up algorithm.
Track2 on Real World LR Images We used the test dataset containing 100 real LR images without any ground truth data. As shown in Fig. 5, both our DDNSR model and the Impressionism model can successfully generate clean images, while ESRGAN can hardly clean the noise. Moreover, the Impressionism model intends to generate sharper results which may easily change the original image semantics, while ours intends to generate detailed images without obviously changing the original semantic information. This may because the Impressionism model used the explicitly constructed LR images, without an effective way to observe how close the constructed LR images and the real LR images are.

4.5 Ablation study

In this subsection, we will investigate the effectiveness of our proposed method, by removing a particular part from the whole method. As shown in Table 2, we focus on the three components of Eq. (3) and the additive noise part in the degradation network. In particular, we trained these independent models in only 100k batches, instead of 300k batches, as mentioned in Sect. 4.1, to clearly show their differences.

Three Parts of the Loss Function To show the effectiveness of our self-supervised learning method, we trained three independent models using Eqs. (2) and (3) while setting one of the three λ values to zero. As shown in Fig. 7, missing any part of these three could lead to a worse perceptual result.
Additional input $\mathcal{X}_L$ As shown in Fig. 2, our degradation network used both $\mathcal{X}_L$ and $\mathcal{Y}_H$ as inputs, while other self-

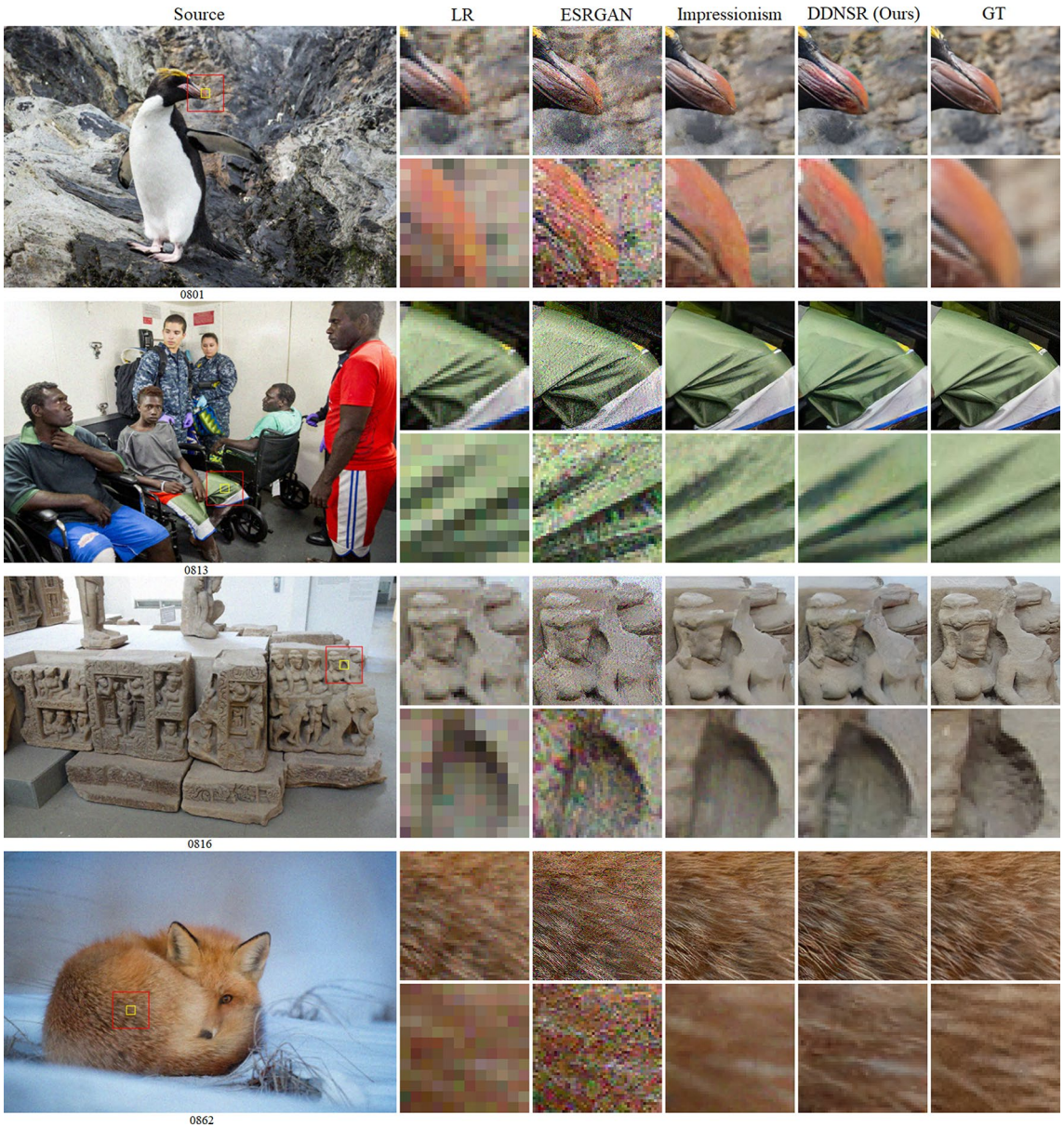

Fig. 4 Perceptual results on Track1 with ground truth. Both our DDNSR model and the Impressionism model can generate clean SR outputs, but the Impressionism's results still contain a little noise. Zooming in can provide more details

supervised/unsupervised learning methods used $\mathcal{Y}_H$ only. The source $\mathcal{X}_L$ is used for sensor-noise collection, which is indispensable for the degradation process learning. For comparison, we set the source $\mathcal{X}_L$ of our degradation network to be an all-zero matrix with the same shape of $\mathcal{X}_L$ to minimize the implementation differences. As shown in Fig. 6, a degradation network without the source $\mathcal{X}_L$ can hardly simulate the sensor noise, therefore leading to poor reconstruction results (Fig. 7), as we mentioned in Sect. 3.1.

4.6 Evaluation on unpaired datasets

According to the experimental results in Sect. 4, our method can successfully generate both clean and detailed SR results without obviously changing their original semantic information. Moreover, we use the same method on both tracks without changing any hyperparameter or network architecture. This means that our method is a robust self-supervised learning method that can effectively learn on different unpaired datasets.

Fig. 5 Perceptual results on Track2 without the ground truth. Both our model and the Impressionism model can generate clean SR outputs. On the one hand, their model intended to generate sharper results, while our model intended to generate detailed results. On the other hand, their model will partially lose the original semantic information, while ours will not. Zooming in can provide more details

While it does not matter which source we use, it does matter how close the source and target are. Generally, the closer the source and target are, the easier our training procedure will be. For Track1, $\mathcal{X}_{Flikr2K}$ and $\mathcal{Y}_{DIV2K}$ are both real-world image datasets containing all kinds of natural scenes with many details. In addition, the down-sampling operation from target to source was a predefined bicubic algorithm, which retained many high-frequency details for reconstruction. Thus, the SR results in Track1 were very impressive (Fig.4).

But for Track2, things will be much more complicated. First, source images were captured by a real mobile device (iPhone3), where too many high-frequency details were lost. Second, there is a large semantic gap between $\mathcal{X}_{DPED}$ and $\mathcal{Y}_{DIV2K}$, which could severely increase the burden of model training. Based on our observation, more than 90% of images in the target $\mathcal{Y}_{DIV2K}$ have no relation at all with LR images in $\mathcal{X}_{DPED}$. Thus, the Impressionism model used the clean-up method, which uses bicubic down-sampled $\mathcal{X}^{clean}_{DPED}$ as an additional target. $\mathcal{X}^{clean}_{DPED}$ can be very close to $\mathcal{X}_{DPED}$, since it is directly obtained from $\mathcal{X}_{DPED}$.

To evaluate the effect of different targets, we trained three independent models on different targets (Sect. 4.2): $\mathcal{X}^{clean}_{DPED}$ (have the exact same content and color), $\mathcal{Y}_{DPED}$ (have a close

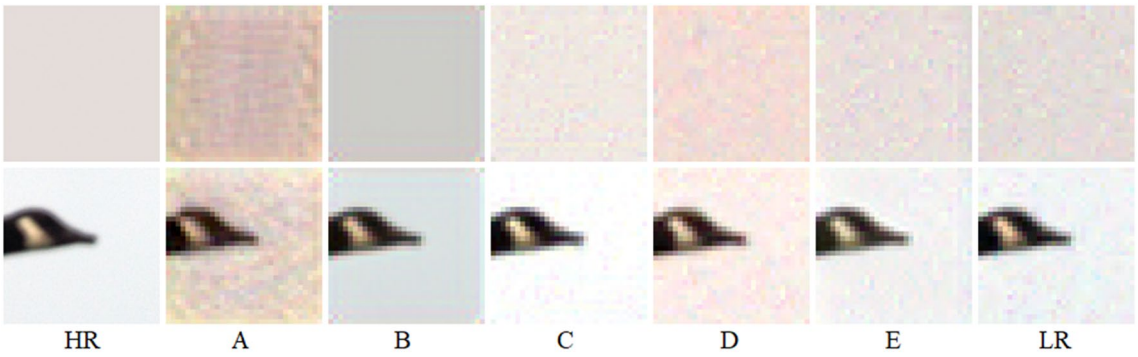

Fig. 6 Degradation results in the ablation study. The degradation network in Model D is considered a single-input network. Their simulated sensor noise is mainly caused by zero-padding, while the results of Model E are mainly from real LR samples. Moreover, missing any part of Eq. (3) will cause worse performance. It is notable that we trained these five different networks on only 100k batches

content, with more details and brighter color) and $\mathcal{Y}_{DIV2K}$ (have completely different semantic information). As shown in Fig. 8, $\mathcal{Y}_{DIV2K}$ and $\mathcal{Y}_{DPED}$ could help with the color reconstruction but to some extent increase the learning burden, therefore failing to reconstruct the lost details. More SR results can be found on our website.

5 Conclusion

In this paper, we proposed an effective dual-input degradation network, along with a self-supervised learning method for Real-World Super-Resolution. Our method could easily address the difficulty of training on unpaired data and therefore is very practical for real-world SR applications. Moreover, we found that different training targets will have a great effect on the reconstruction results so that we must carefully choose the training target for special purposes. In addition, we believe our method could be easily used in other image reconstruction algorithms, such as image denoising and image deblurring (these tasks have the same goal of restoring low-quality images to high-quality images). We leave this as future work.

Table 2 Implementation Differences in the Ablation Study

Model name	Additional input $\mathcal{X}_L$	λ_1	λ_2	λ_3
A	Yes	0.0	1.0	10.0
B	Yes	0.1	0.0	10.0
C	Yes	0.1	1.0	0.0
D	No	0.1	1.0	10.0
E	Yes	0.1	1.0	10.0

Appendix A: Convolution with Zero-padding

In the scope of SISR, deep-learning-based models always use zero padding to ensure that the convolution outputs have the same size as inputs. We consider a single-input CNN-based network G (as used in most existing unsupervised/self-supervised methods) containing n convolution layers, and an input image x of size $h \times w$, with all pixel values set to be the same (e.g., full-white color). We set G with random weights and random bias values; then, we calculate $G(x)$, we will obtain an output with a patch size $(h-n) \times (w-n)$ having the same pixel value (Fig. 9).

This could be a fatal weakness. First, all the hidden outputs of which patch size $(h-n) \times (w-n)$ is positive will contain no noise at all. In addition, zero padding is completely artificial, which is a very low-efficiency way for training. Based on our observation, those single-input CNN-based networks can hardly learn the real sensor noise. We have shown this phenomenon in our ablation study (Sect. 4.5).

Appendix B: Cleaned-up result

As mentioned in Sect. 4.2, the clean-up method was helpful. We adopted this method in Track2 and evaluated it above. On the one hand, this method could help our model reconstruct more details (Fig. 8). On the other hand, it will introduce unnecessary noise if we look at the reconstruction results carefully.

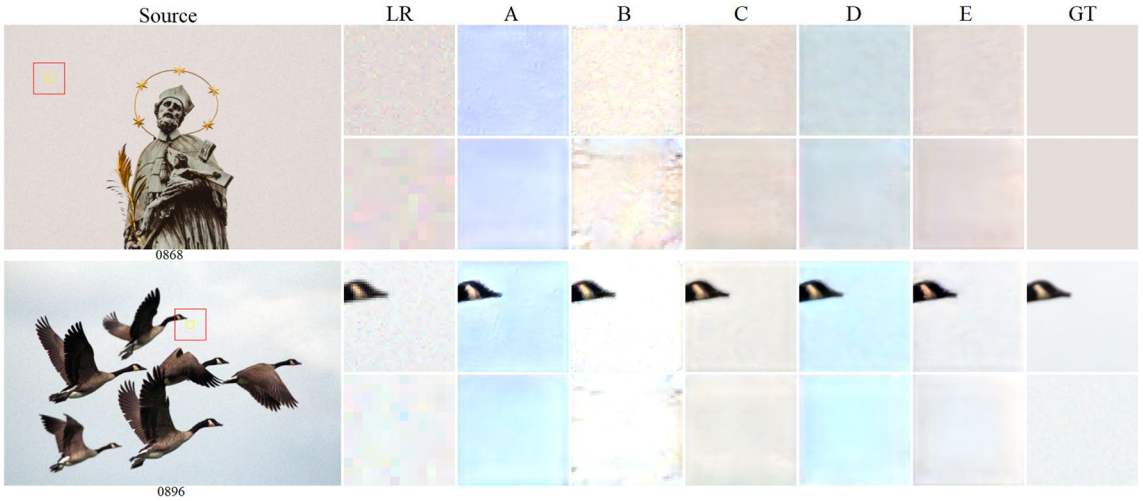

Fig. 7 Super-resolution results in the ablation study. The degradation network in Model D is considered a single-input network, which can hardly learn the real sensor noise. Regarding the results, they can hardly generate ideal SR results. Moreover, missing any part of Eq. (3) will cause worse performance. It is notable that we trained these five different networks on only 100k batches

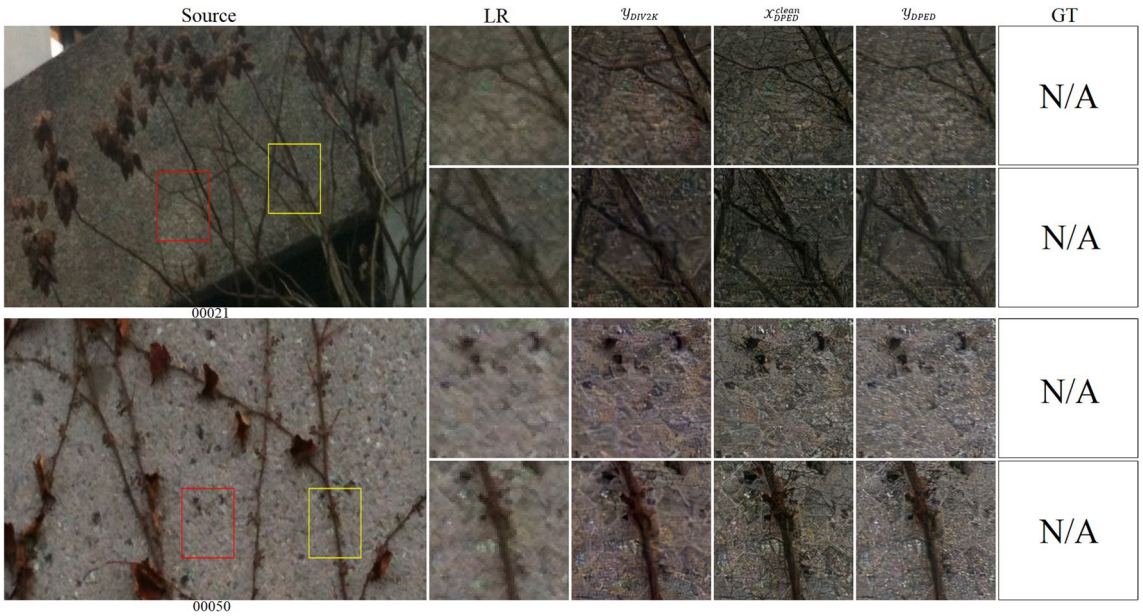

Fig. 8 Visual comparisons among models trained on different targets. Different targets will lead to very different SR results. In particular, $\mathcal{X}^{clean}_{DPED}$ are the closest to the source data, therefore, their SR results are the best in the case of image semantics. $\mathcal{Y}_{DIV2K}$ and $\mathcal{Y}_{DPED}$ are real-world high-quality datasets, which will help with color reconstruction

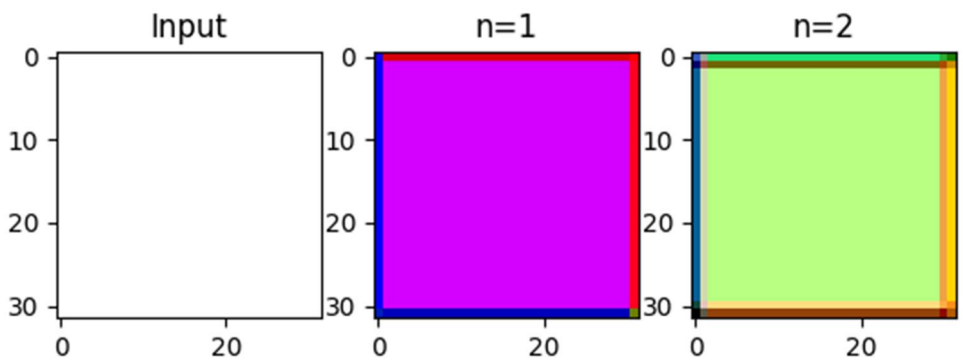

Fig. 9 Results of a single-input CNN-based network G with zero-padding. If G is a shallow network, most of the pixels in the output will have the same value. The input image has a typical size of 32×32, and 'n' is the number of convolution layers

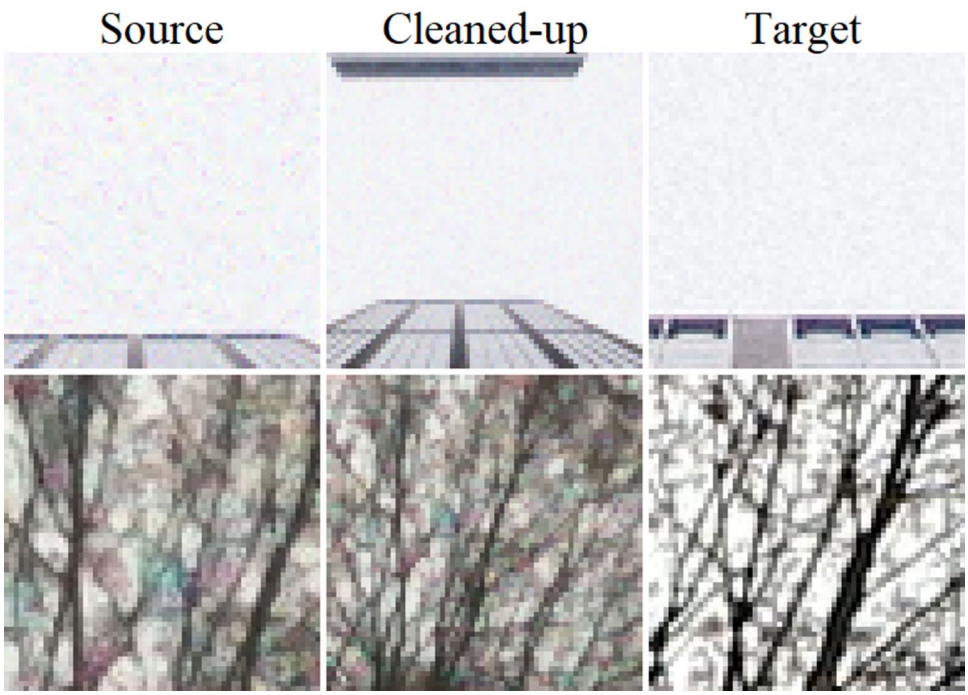

Fig. 10 The perceptual quality of the cleaned-up results are still blurry and noisy. It is notable that images in the same row are cropped from the same scene but are not fully matched

Table 3 PSNR and SSIM Values Calculated on Track1

Model	PSNR↑	SSIM↑	LPIPS↓
ESRGAN	19.06	0.262	0.755
Impressionism	24.77	0.673	0.227
DDNSR (Ours with ESRGAN)	23.29	0.660	0.232
DDNSR (Ours with L_2 loss)	25.13	0.734	0.352

In Fig. 10, we show some of the cleaned-up results compared with the original noisy LR images and the target images. As we can see, the cleaned-up image is not completely clean since the bicubic algorithm is not an ideal clean-up algorithm.

Appendix C: Full-reference metrics of Track1

We used only the L_2 loss as the loss function instead of Eq. (2) for the reconstruction network, trained two independent models, and calculated the PSNR, SSIM, and LPIPS [19] values on the 100 validation image pairs.

As shown in Table 3, L_2 loss could lead to a higher PSNR and SSIM value, which is a common phenomenon that appeared in many other works. In the context of Image Super-Resolution, there is no persuasive measurement of the image quality yet. Thus, we provided these results for reference only.

Appendix D: Full results on both tracks

See Figs. 11, 12.

Fig. 11 Full result on Track1. The left part is the LR image upscaled by a 4 × 4 factor, the right part is the SR result generated by our DDNSR model

Fig. 12 Full results on Track2. The left part is the LR image upscaled by a 4 × 4 factor, the right part is the SR result generated by our DDNSR model

Data availability The data that support the findings of this study are openly available in Codalab at https://competitions.codalab.org/competitions/22220#participate.

Declarations

Conflict of Interest We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

1. Dong C, Loy CC, He K, Tang X (2016) Image Super-Resolution Using Deep Convolutional Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(2): 295–307. https://doi.org/10.1109/TPAMI.2015.2439281. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence
2. Lim B, Son S, Kim H, Nah S, Lee KM (2017) Enhanced Deep Residual Networks for Single Image Super-Resolution. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1132–1140. https://doi.org/10.1109/CVPRW.2017.151. ISSN: 2160-7516
3. Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, Shi W (2017) Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 105–114. https://doi.org/10.1109/CVPR.2017.19. ISSN: 1063-6919
4. Wang X, Yu K, Wu S, Gu J, Liu Y, Dong C, Qiao Y, Loy CC (2019) ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In: Leal-Taixé L, Roth S (eds.) Computer Vision – ECCV 2018 Workshops. Lecture Notes in Computer Science, pp. 63–79. Springer, Cham. https://doi.org/10.1007/978-3-030-11021-5_5
5. Cai J, Zeng H, Yong H, Cao Z, Zhang L (2019) Toward Real-World Single Image Super-Resolution: A New Benchmark and a New Model. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3086–3095. IEEE, Seoul, Korea (South). https://doi.org/10.1109/ICCV.2019.00318. https://ieeexplore.ieee.org/document/9009805/ Accessed 2021-03-14
6. Blau Y, Mechrez R, Timofte R, Michaeli T, Zelnik-Manor L (2019) The 2018 PIRM Challenge on Perceptual Image Super-Resolution. In: Leal-Taixé L, Roth S (eds.) Computer Vision – ECCV 2018 Workshops vol. 11133, pp. 334–355. Springer, Cham. https://doi.org/10.1007/978-3-030-11021-5_21. Series Title: Lecture Notes in Computer Science. http://link.springer.com/10.1007/978-3-030-11021-5_21 Accessed 2021-07-19
7. Ji X, Cao Y, Tai Y, Wang C, Li J, Huang F (2020) Real-World Super-Resolution via Kernel Estimation and Noise Injection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1914–1923. https://doi.org/10.1109/CVPRW50498.2020.00241. ISSN: 2160-7516
8. Lugmayr A, Danelljan M, Timofte R, Ahn N, Bai D, Cai J, Cao Y, Chen J, Cheng K, Chun S, Deng W, El-Khamy M, Ho CM, Ji X, Kheradmand A, Kim G, Ko H, Lee K, Lee J, Li H, Liu Z, Liu Z-S, Liu S, Lu Y, Meng Z, Michelini PN, Micheloni C, Prajapati K, Ren H, Seo YH, Siu W-C, Sohn K-A, Tai Y, Umer RM, Wang S, Wang H, Wu TH, Wu H, Yang B, Yang F, Yoo J, Zhao T, Zhou Y, Zhuo H, Zong Z, Zou X (2020) NTIRE 2020 Challenge on Real-World Image Super-Resolution: Methods and Results. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2058–2076. https://doi.org/10.1109/CVPRW50498.2020.00255. ISSN: 2160-7516
9. Bell-Kligler S, Shocher A, Irani M (2019) Blind super-resolution kernel estimation using an internal-GAN. Advances in Neural Information Processing Systems 32
10. Choi J-H, Zhang H, Kim J-H, Hsieh C-J, Lee J-S (2019) Evaluating Robustness of Deep Image Super-Resolution Against Adversarial Attacks. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 303–311. https://doi.org/10.1109/ICCV.2019.00039. ISSN: 2380-7504
11. Lugmayr A, Danelljan M, Timofte R (2019) Unsupervised Learning for Real-World Super-Resolution. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 3408–3416. https://doi.org/10.1109/ICCVW.2019.00423. ISSN: 2473-9944
12. Chen S, Han Z, Dai E, Jia X, Liu Z, Liu X, Zou X, Xu C, Liu J, Tian Q (2020) Unsupervised Image Super-Resolution with an Indirect Supervised Path. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1924–1933. IEEE, Seattle, WA, USA. https://doi.org/10.1109/CVPRW50498.2020.00242. https://ieeexplore.ieee.org/document/9151023/ Accessed 2021-06-05
13. Kim G, Park J, Lee K, Lee J, Min J, Lee B, Han DK, Ko H (2020) Unsupervised Real-World Super Resolution with Cycle Generative Adversarial Network and Domain Discriminator. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1862–1871. https://doi.org/10.1109/CVPRW50498.2020.00236. ISSN: 2160-7516
14. Sun W, Gong D, Shi Q, van den Hengel A, Zhang Y (2021) Learning to Zoom-In via Learning to Zoom-Out: Real-World Super-Resolution by Generating and Adapting Degradation. IEEE Transactions on Image Processing 30:2947–2962. https://doi.org/10.1109/TIP.2021.3049951. Conference Name: IEEE Transactions on Image Processing
15. Zhu J-Y, Park T, Isola P, Efros AA (2017) Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2242–2251. https://doi.org/10.1109/ICCV.2017.244. ISSN: 2380-7504
16. Zhang Y, Li K, Li K, Wang L, Zhong B, Fu Y (2018) Image Super-Resolution Using Very Deep Residual Channel Attention Networks. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y (eds.) Computer Vision – ECCV 2018 vol. 11211, pp. 294–310. Springer, Cham. https://doi.org/10.1007/978-3-030-01234-2_18. Series Title: Lecture Notes in Computer Science. http://link.springer.com/10.1007/978-3-030-01234-2_18 Accessed 2021-06-06

17. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial networks. Adv Neural Inf Process Syst 3:87
18. Zhao T, Ren W, Zhang C, Ren D, Hu Q (2018) Unsupervised degradation learning for single image super-resolution. arXiv:1812.04240 [cs]. arXiv: 1812.04240. Accessed 15 June 2021
19. Zhang R, Isola P, Efros AA, Shechtman E, Wang O (2018) The unreasonable effectiveness of deep features as a perceptual metric. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, pp 586–595. https://doi.org/10.1109/CVPR.2018.00068. ISSN: 2575-7075
20. Bulat A, Yang J, Tzimiropoulos G (2018) To learn image super-resolution, use a GAN to learn how to do image degradation first. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y (eds.) Computer vision–ECCV 2018. Lecture Notes in Computer Science, pp 187–202. Springer, Cham. https://doi.org/10.1007/978-3-030-01231-1_12
21. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 [cs]. arXiv: 1409.1556. Accessed 05 June 2021
22. Timofte R, Agustsson E, Gool LV, Yang M-H, Zhang L, Lim B, Son S, Kim H, Nah S, Lee KM, Wang X, Tian Y, Yu K, Zhang Y, Wu S, Dong C, Lin L, Qiao Y, Loy CC, Bae W, Yoo J, Han Y, Ye JC, Choi J-S, Kim M, Fan Y, Yu J, Han W, Liu D, Yu H, Wang Z, Shi H, Wang X, Huang TS, Chen Y, Zhang K, Zuo W, Tang Z, Luo L, Li S, Fu M, Cao L, Heng W, Bui G, Le T, Duan Y, Tao D, Wang R, Lin X, Pang J, Xu J, Zhao Y, Xu X, Pan J, Sun D, Zhang Y, Song X, Dai Y, Qin X, Huynh X-P, Guo T, Mousavi HS, Vu TH, Monga V, Cruz C, Egiazarian K, Katkovnik V, Mehta R, Jain AK, Agarwalla A, Praveen CVS, Zhou R, Wen H, Zhu C, Xia Z, Wang Z, Guo Q (2017) NTIRE 2017 challenge on single image super-resolution: methods and results. In: 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW), pp 1110–1121. https://doi.org/10.1109/CVPRW.2017.149. ISSN: 2160-7516
23. Agustsson E, Timofte R (2017) NTIRE 2017 challenge on single image super-resolution: dataset and study. In 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW), pp 1122–1131. https://doi.org/10.1109/CVPRW.2017.150. ISSN: 2160-7516
24. Ignatov A, Kobyshev N, Timofte R, Vanhoey K (2017) DSLR-quality photos on mobile devices with deep convolutional networks. In: 2017 IEEE international conference on computer vision (ICCV), pp 3297–3305. https://doi.org/10.1109/ICCV.2017.355. ISSN: 2380-7504