Real-World Super-Resolution via Kernel Estimation and Noise Injection

Xiaozhong Ji[1,2] Yun Cao[2] Ying Tai[2*] Chengjie Wang[2] Jilin Li[2] Feiyue Huang[2]
[1]Nanjing University, [2]Tencent Youtu Lab
shawn_ji@163.com, {yuncao, yingtai, jasoncjwang, jerolinli, garyhuang}@tencent.com
https://github.com/jixiaozhong/RealSR

Abstract

Recent state-of-the-art super-resolution methods have achieved impressive performance on ideal datasets regardless of blur and noise. However, these methods always fail in real-world image super-resolution, since most of them adopt simple bicubic downsampling from high-quality images to construct Low-Resolution (LR) and High-Resolution (HR) pairs for training which may lose track of frequency-related details. To address this issue, we focus on designing a novel degradation framework for real-world images by estimating various blur kernels as well as real noise distributions. Based on our novel degradation framework, we can acquire LR images sharing a common domain with real-world images. Then, we propose a real-world super-resolution model aiming at better perception. Extensive experiments on synthetic noise data and real-world images demonstrate that our method outperforms the state-of-the-art methods, resulting in lower noise and better visual quality. In addition, our method is the winner of NTIRE 2020 *Challenge on both tracks of Real-World Super-Resolution, which significantly outperforms other competitors by large margins.*

1. Introduction

Super-Resolution (SR) task is to increase the resolution of low-quality images, and enhance its clarity [2]. In recent years, deep learning-based methods [9, 35, 7, 34, 20, 19, 23] have achieved remarkable results with respect to fidelity performance, which mainly focuses on designing network structures to further improve the performance of specific datasets. Most of them use fixed bicubic operation for downsampling to construct training data pairs. Similarly, in test phase, the input image downsampled by bicubic kernel is feed to the designed network. Subsequently, the generated results will be compared with Ground Truth (GT) to calculate PSNR, SSIM and other metrics.

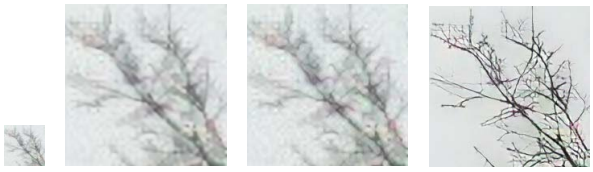

Figure 1. **Visualization comparison among EDSR, ZSSR, and our RealSR** on a real-world low-resolution image.

Despite the improvement of fidelity, a problem ignored by these methods is that downsampling with the ideal bicubic is unreasonable. Previous methods construct data by ideal downsampling method:

$$\mathbf{I}_{LR} = \mathbf{I}_{HR}\downarrow_s, \tag{1}$$

where $\mathbf{I}_{LR}$ and $\mathbf{I}_{HR}$ indicate the LR and HR image, respectively, and s denotes the scale factor. This makes it easy to obtain paired data for training models. However, with such a known and fixed downsampling kernel, the degraded images may lose high-frequency details but make the low-frequency content more clear. Based on such constructed paired data, SR model $f(\cdot)$ is trained to minimize the average error of n images:

$$\arg\min_f \Sigma \, \|f(\mathbf{I}^i_{LR}) - \mathbf{I}^i_{HR}\|, i \in \{1, 2 \cdots n\}. \tag{2}$$

If testing on the same downsampling dataset, the generated results are as expected. However, once we directly test on the original image, the results are very blurry with lots of noise. The main reason is that the bicubic downsampled image does not belong to the same domain as the original image. Due to the domain gap, these methods produce unpleasant artifacts and fail on real-world images. For example, EDSR/ZSSR produce unsatisfied result of a real image in Figure 1. Therefore, the key problem of real-world super-resolution is to introduce an accurate degradation method to ensure the generated Low-Resolution (LR) image and the original image with the same domain attributes.

We first analyze the impact of different kernels on the downsampled image [42, 49, 11]. Before our analysis, we

[*] This work was done when Xiaozhong is an intern at Tencent Youtu Lab. Ying Tai is the corresponding author.

define the original real images as the source domain $\mathcal{X}$, and the clean High-Resolution (HR) images as the target domain $\mathcal{Y}$. We found blur kernels with different degrees directly affect the blur of the downsampled images. Bicubic can be regarded as an ideal way of downsampling because it retains the information from $\mathcal{X}$ as much as possible. However, the frequency of these downsampled images has changed to another domain $\mathcal{X}^{'}$. When training on $\{\mathcal{X}^{'}, \mathcal{Y}\}$, the model will try to recover all the details due to all information is important in the domain $\mathcal{X}^{'}$. The model works well on $\mathbf{I}_{LR}$ but usually fails on $\mathbf{I}_{src} \in \mathcal{X}$, which is an unprocessed real image. Another problem is the downsampled image has almost no noise, while real-world images in $\mathcal{X}$ usually have a lot. Mere estimation of the blurry kernel cannot accurately model the degradation process.

In this paper, we propose a novel Realistic degradation framework for Super-Resolution (RealSR), which contains kernel estimation and noise injection to preserve the original domain attributes. On one hand, we first use the existing kernel estimation method [3] to generate more realistic LR images. On the other hand, we propose a simple and effective method to directly collect noise from the original image and add it to the downsampled image. Further, we introduce the patch discriminator [17] for RealSR to avoid generated artifacts. To verify the effectiveness of the proposed method, we conduct experiments on synthetic dataset and real dataset. The experimental results show that our method produces clearer and cleaner results compared with state-of-the-art methods. Finally, we conduct ablation experiments to verify the effectiveness of the kernel estimation, noise injection, and the patch discriminator for SR generator, respectively. We also participate in the NTIRE 2020 Challenge on Real-World Super-Resolution, and outperform other competitors by large margins on both tracks.

In summary, our overall contribution is three-fold:

- We propose a novel degradation framework RealSR under real-world setting, which provides realistic images for super-resolution learning.
- By estimating the kernel and noise, we explore the specific degradation of blurry and noisy images.
- We demonstrate that the proposed RealSR achieves state-of-the-art results in terms of visual quality.

2. Related Work

Super-Resolution Recently, many Convolutional Neural Networks (CNN)-based SR networks [23, 30, 13, 14, 22, 31, 41] achieve strong performance on bicubic downsampling images. Among them, the representative is EDSR [23], which uses a deep residual network for training SR model. Zhang *et al*. [46] propose a residual in residual structure to form very deep network which achieves better performance than EDSR. Dai *et al*. [8] propose a second-order channel attention module to adaptively rescale the channel-wise features by using second-order feature statistics for more discriminative representations. Haris *et al*. [12] propose deep back-projection networks to exploit iterative up- and down-sampling layers, providing an error feedback mechanism for projection errors at each stage. Although the authors have achieved good performance with respect to fidelity, the generated images have poor visual effects and appear blurry. To address this issue, some researchers enhanced realistic texture via spatial feature transform [47, 48, 38]. Soh *et al*. propose a natural manifold discrimination to classify HR images with blurry or noisy images, which is used to supervise the quality of the generated images. Furthermore, some Generative Adversarial Networks (GAN)-based methods [21, 44, 39] pay more attention to visual effects, introducing adversarial losses and perceptual losses.

However, these SR models trained on the data generated by bicubic kernel can only work well on clean HR data, because the model has never seen blurry/noisy data during training. This is inconsistent with real-world needs, and real LR images often carry noise and blur. To address this conflict, Xu *et al*. [40, 5, 4, 45] collect raw photo pairs directly from nature scene with particular camera equipment. But collecting such paired data requires strict conditions and a lot of manual costs. In this paper, we focus on the training strategy of SR networks in real data by analyzing the degradation in real images.

Real-World Super-Resolution To overcome these challenges of real-world super-resolution, recent work [42, 49] combined with denoising or deblurring have been proposed. These methods are trained on the artificially constructed blurry and noise-added data, which further enhanced the robustness of the SR model. However, these explicit modeling methods need sufficient prior about blur/noise, therefore the scope of application is limited.

Recently, a series of real-world super-resolution challenges [25, 26] have attracted many participants. Many novel methods are proposed to solve this problem. For instance, Fritsche *et al*. [10] propose the DSGAN model to generate degraded images. Lugmayr *et al*. [24] propose an unsupervised learning method for real-world super-resolution. ZSSR [32] abandon the training process on big data and train a small model for each test image so that specific models pay more attention to the internal information of the image. But the price paid is that the time for inference is greatly increased, which is difficult to apply to the real scene. Different from these methods, we explicitly estimate the kernel degradation in real images, which is very important for generating clear and sharp results.

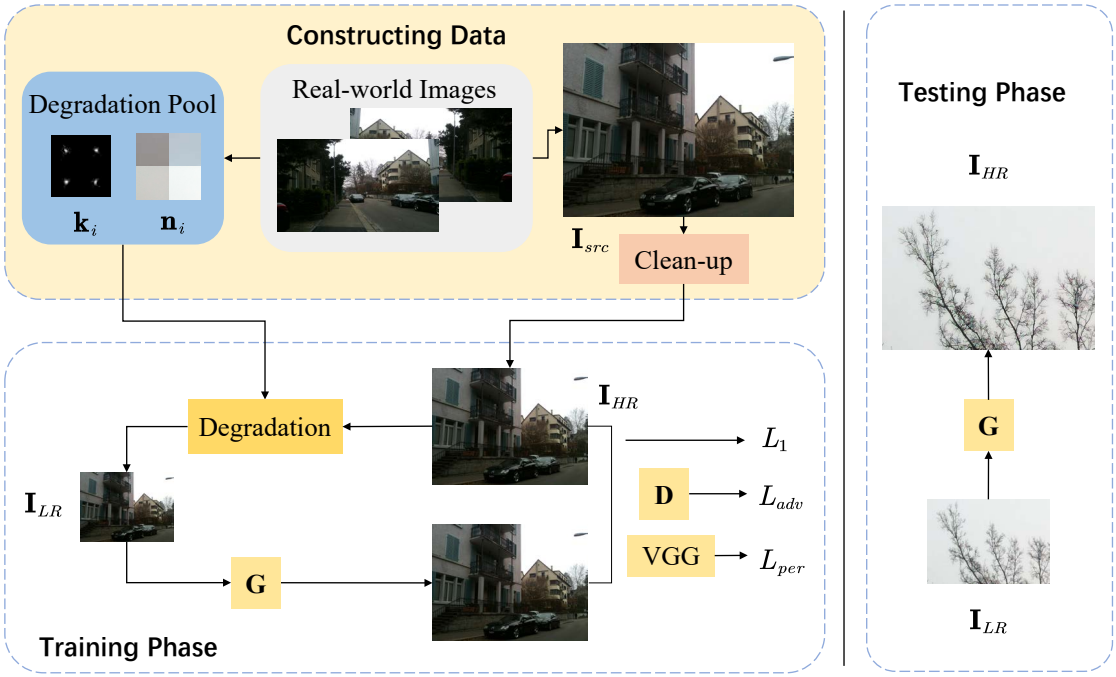

Figure 2. **Framework of our proposed RealSR method**. The degradation pool provides diverse blur kernels and noise distributions for constructing realistic low-resolution images. During training phase, the SR model is optimized to reconstruct high-resolution images.

3. The Proposed RealSR

In this section, we introduce the proposed degradation method as shown in Figure 2. Our method is mainly divided into two stages. The first stage is to estimate the degradation from real data and generate realistic LR images. The second stage is to train the SR model based on the constructed data.

3.1. Realistic Degradation for Super-Resolution

Here, we introduce a novel method of real image degradation based on kernel estimation and noise injection. Assume that the LR image is obtained by the following degradation method:

$$\mathbf{I}_{LR} = (\mathbf{I}_{HR} * \mathbf{k})\downarrow_s + \mathbf{n}, \tag{3}$$

where $\mathbf{k}$ and $\mathbf{n}$ indicate blurry kernel and noise, respectively. $\mathbf{I}_{HR}$ is unknown, indicating that $\mathbf{k}$ and $\mathbf{n}$ are also unknown. In order to estimate the degradation method more accurately, we explicitly estimate the kernel and noise from the image. After getting the estimated kernel and noise patch, we build a degradation pool, which is used to degrade clean HR images into blurry and noisy images, thereby generating image pairs for training SR models. To describe our method concisely, we formalize this data-constructing pipeline as an algorithm shown in Algorithm 1.

Algorithm 1 Realistic Degradation of our RealSR

Input: Real images set $\mathcal{X}$, HR images set $\mathcal{Y}$, downsampling scale factor s
Output: Realistic paired images $\{\mathbf{I}_{LR}, \mathbf{I}_{HR}\}$

```
Initialize kernel pool K = ∅
Initialize noise pool N = ∅
for all I_src such that I_src ∈ X do
    Estimate k from I_src by solving Eqn. 4
    Add k to K
    Crop n from I_src
    if n meet Eqn. 7 then
        Add n to N
    end if
end for
for all I_HR such that I_HR ∈ Y do
    Randomly select k_i ∈ K, n_j ∈ N
    Generate I_LR with k_i and n_j
end for
return {I_LR, I_HR}
```

3.2. Kernel Estimation and Downsampling

We use a kernel estimation algorithm to explicitly estimate kernels from real images. Inspired by KernelGAN [3], we adopt a similar kernel estimation method and set appropriate parameters based on real images. The generator of KernelGAN is a linear model without any activation layers,

therefore the parameters of all layers can be combined into a fixed kernel. The estimated kernel needs to meet the following constraints:

$$\begin{aligned}\arg\min_{\mathbf{k}} \|(\mathbf{I}_{src} * \mathbf{k})\downarrow_s - \mathbf{I}_{src}\downarrow_s\|_1 + |1 - \Sigma\, \mathbf{k}_{i,j}| \\ +|\Sigma\, \mathbf{k}_{i,j} \cdot \mathbf{m}_{i,j}| + |1 - D((\mathbf{I}_{src} * \mathbf{k})\downarrow_s)|.\end{aligned} \tag{4}$$

$(\mathbf{I}_{src} * \mathbf{k})\downarrow_s$ is downsampled LR image with kernel $\mathbf{k}$, and $\mathbf{I}_{src}\downarrow_s$ is downsampled image with ideal kernel, therefore to minimize this error is to encourage the downsampled image to preserve important low-frequency information of the source image. What's more, the second term of the above formula is to constrain $\mathbf{k}$ to sum to 1, and the third term is to penalty boundaries of $\mathbf{k}$. Finally, the discriminator $D(\cdot)$ is to ensure the consistency of source domain .

Clean-Up To get more HR images, we try to generate *noise-free* images from $\mathcal{X}$. Specifically, we adopt bicubic downsampling on the real image in the source domain to remove noise and make the image sharper. Let $\mathbf{I}_{src} \in \mathcal{X}$ be an image from real source images set, and $\mathbf{k}_{bic}$ be the ideal bicubic kernel. Then the image is downsampled with a clean-up scale factor sc.

$$\mathbf{I}_{HR} = (\mathbf{I}_{src} * \mathbf{k}_{bic})\downarrow_{sc}. \tag{5}$$

Degradation with Blur Kernels We regard the images after downsampling as clean HR images. Then we perform degradation to these HR images by randomly selecting a blur kernel from the degradation pool. The downsampling process is cross-correlation operations followed by sampling with stride s, which can be formulated as:

$$\mathbf{I}_D = (\mathbf{I}_{HR} * \mathbf{k}_i)\downarrow_s, i \in \{1, 2 \cdots m\}, \tag{6}$$

where $\mathbf{I}_D$ denotes the downsampled image, and $\mathbf{k}_i$ refers to the selected specific blur kernel from $\{\mathbf{k}_1, \mathbf{k}_2 \cdots \mathbf{k}_m\}$.

3.3. Noise Injection

For noisy images, we explicitly inject noise into the downsampled images to generate realistic LR images. Since the high-frequency information is lost during the downsampling process, the degraded noise distribution changes at the same time. In order to make the degraded image have a similar noise distribution to the source image, we directly collect noise patches from the source dataset $\mathcal{X}$. We observe that patches with richer content have a larger variance. Based on this observation and inspired by [6, 49], we design a filtering rule to collect patches with their *variance* in a certain range. Simply but effectively, we decouple noise and content by the following rule:

$$\sigma(\mathbf{n}_i) < v, \tag{7}$$

where $\sigma(\cdot)$ denotes the function to calculate variance, and v is the max value of variance.

Degradation with Noise Injection Assume that a series of noise patches $\{\mathbf{n}_1, \mathbf{n}_2 \cdots \mathbf{n}_l\}$ are collected and added into the degradation pool. The noise injection process is performed by randomly cropping patches from the noise pool. Similarly, we formalize this process as:

$$\mathbf{I}_{LR} = \mathbf{I}_D + \mathbf{n}_i, i \in \{1, 2 \cdots l\}, \tag{8}$$

where $\mathbf{n}_i$ is a cropped noise patch from the noise pool consisting of $\{\mathbf{k}_1, \mathbf{k}_2 \cdots \mathbf{k}_l\}$. In detail, we adopt an online noise injection method that the content and the noise are combined during training phase. This makes the noise more diverse and regularizes the SR model to distinguish content with noise. After the degradation with blur kernels and injecting noise, we obtain $\mathbf{I}_{LR} \in \mathcal{X}$.

3.4. Super-Resolution Model

Based on ESRGAN [39], we implement an SR model and train it on constructed paired data $\{\mathbf{I}_{LR}, \mathbf{I}_{HR}\} \in \{\mathcal{X}, \mathcal{Y}\}$. The generator adopts RRDB [39] structure, and the resolution of the generated image will be enlarged for 4 times. Several losses are applied to training includes pixel loss, perceptual loss [18], and adversarial loss. The pixel loss L_1 uses L1 distance. Perceptual loss L_{per} uses the inactive features of VGG-19 [33], which helps to enhance the visual effect of low-frequency features such as edges. Adversarial loss L_{adv} is used to enhance the texture details of the generated image to make it look more realistic. The final loss function is the weighted sum of these three losses:

$$L_{total} = \lambda_1 \cdot L_1 + \lambda_{per} \cdot L_{per} + \lambda_{adv} \cdot L_{adv}, \tag{9}$$

where λ_1, λ_{per}, and λ_{adv} are set as 0.01, 1, and 0.005 empirically.

3.5. Patch Discriminator in RealSR

In addition, we observe that the ESRGAN [39] discriminator may introduce many artifacts. Different from default ESRGAN setting, we use patch discriminator [17, 50] instead of VGG-128 [33] because of two conveniences: 1) VGG-128 limits the size of the generated image to 128, making multi-scale training inconvenient. 2) VGG-128 contains a deeper network and its fixed fully connected layers make the discriminator pay more attention to global features and ignore local features. In contrast, we use a patch discriminator with fully convolution structure, which has a fixed receptive field. For example, a three-layer network corresponds to a 70×70 patch. That is, each output value by the discriminator is only related to the patch of local fixed area. The patch losses will be fed back to the generator to optimize the gradient of local details. Note that the final error is the average of all local errors to guarantee global consistency.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
EDSR	**25.31**	0.6383	0.5784
ESRGAN	19.06	0.2423	0.7552
ZSSR	25.13	0.6268	0.6160
K-ZSSR	18.46	0.3826	0.7307
Ours	24.82	**0.6619**	**0.2270**

Table 1. **Quantitative results on DF2K dataset** compared with EDSR, ESRGAN, ZSSR, and K-ZSSR. Note that 'Ours' refers to the proposed RealSR. ↑ and ↓ mean higher or lower is desired.

4. Experiments

4.1. Datasets

DF2K The DF2K dataset merges the DIV2K [36] and Flikr2K [1] datasets, and contains a total of $3,450$ images. These images are artificially added with Gaussian noise to simulate sensor noise. The validation set contains 100 images with corresponding ground truth, therefore the metrics based on reference can be calculated.

DPED The DPED [15] dataset contains $5,614$ images taken by the iPhone3 camera. The images in this dataset are unprocessed real images, which are more challenging containing noise, blur, dark light and other low-quality problems. The 100 images in validation set are cropped from original real images. Since there is no corresponding ground truth, we can only provide a visual comparison.

4.2. Evaluation Metrics

For the case of synthetic data, we calculate PSNR, SSIM and LPIPS [43] of results generated by different methods. Among them, PSNR and SSIM are commonly-used evaluation metrics for image restoration. These two metrics pay more attention to the fidelity of the image rather than visual quality. In contrast, LPIPS pays more attention to whether the visual features of images are similar or not. It uses pre-trained Alexnet to extract image features, and then calculates the distance between the two features. Therefore, the smaller the LPIPS is, the closer the generated image is to the ground truth.

4.3. Evaluation on Corrupted Images

First, we compare our RealSR with state-of-the-art SR methods on corrupted DF2K dataset. We evaluate performance on validation set which consists of 100 images. After generating results by these methods, we calculate PSNR, SSIM, and LPIPS according to ground truth. Due to the fact that LPIPS better reflects visual quality, we mainly focus on this metric. The comparing methods include EDSR [23], ESRGAN [39], ZSSR [32], K-ZSSR. We evaluate the EDSR and ESRGAN method using the pre-trained model released by the authors. Since ZSSR doesn't need a training process, we simply run its test code on the validation images. Specifically, K-ZSSR is a combination of KernelGAN [3] and ZSSR. The estimated kernel by KernelGAN is used for downsampling image patches during ZSSR training while ZSSR adopts default bicubic for degradation.

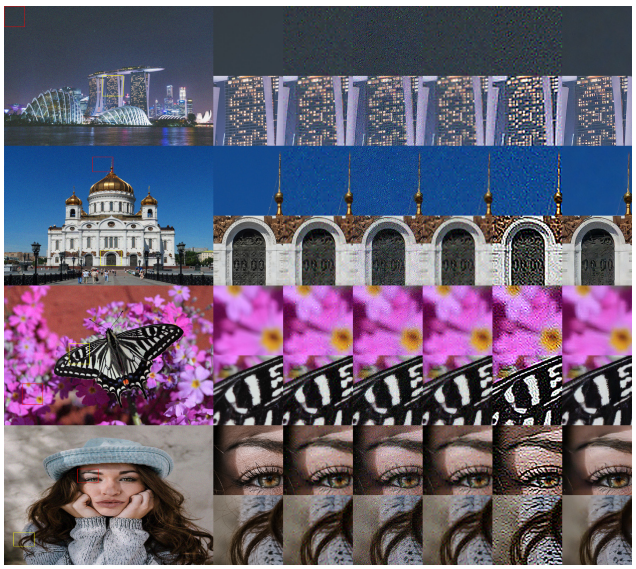

Figure 3. **Qualitative results on DF2K dataset** compared with EDSR, ESRGAN, ZSSR, and K-ZSSR. GT denotes the original HR ground truth image. The red and yellow area is cropped from different results and enlarged for visual convenient.

Quantitative Results on DF2K As shown in Table 1, our RealSR achieves the best LPIPS performance, indicating our results are much closer to the ground truth in terms of visual characteristics. Note that our method is lower in PSNR than EDSR, and this is because we use perceptual loss that pays more attention to visual quality. Generally, the PSNR and LPIPS metrics are not positively correlated, and even show the opposite relationship within a certain range.

Qualitative Results on DF2K From Figure 3, we see the local details of different methods on the same image, where our RealSR produces much less noise. On one hand, compared with EDSR and ZSSR, our results are clearer with richer texture details. On the other hand, compared with ESRGAN and K-ZSSR, our results have *almost no artifacts*, which is benefit from the accurate degradation estimated from real noise distribution. In particular, K-ZSSR uses a more blurry kernel than bicubic, therefore the image used for training has almost no noise, which leads to many artifacts when feeding with noisy images. The SR model mistakes the noise as the content of input image during test.

Figure 4. **Qualitative results on DPED dataset** compared with EDSR, ESRGAN, ZSSR, and K-ZSSR. The red and yellow area is cropped from different results and enlarged for visual convenient.

4.4. Evaluation on Real-World Images

The most concerning problem of our proposed method is real-world super-resolution, so we evaluate our RealSR on the DPED dataset, in which the photos suffer from degradation problems such as blur, noise, *etc*. Just like the problems encountered in SR training of real images, no ground truth that can be referred to in the validation stage. Therefore, we only show the results of visual comparison. In order to make the details clearer, we enlarge the local area.

Qualitative Results on DPED As shown in Figure 4, the EDSR, ESRGAN, and ZSSR methods do not correctly distinguish the noise from the branches and the sky, leading to blurry results. In our results, the trunk and branches are clearer, and the dividing line between the object and the background is sharper. Regarding K-ZSSR, due to wrong processing of the noise, the result produces unnecessary texture details. If we zoom in, this result is unacceptable and cannot be considered as an HR image. When dealing with some solid backgrounds, the advantages of our method are more obvious. As can be seen from the third image, the noise under the eaves has been eliminated, leaving only the important low-frequency features.

Compared with existing methods, our RealSR produces few noise and artifacts, indicating that the noise estimated

Team	PSNR↑	SSIM↑	LPIPS↓	MOS↓
Impressionism (ours), *winner*	24.67 (16)	0.683 (13)	**0.232 (1)**	**2.195**
Samsung-SLSI-MSL	25.59 (12)	0.727 (9)	0.252 (2)	2.425
BOE-IOT-AIBD	26.71 (4)	0.761 (4)	0.280 (4)	2.495
MSMers	23.20 (18)	0.651 (17)	0.272 (3)	2.530
KU-ISPL	26.23 (6)	0.747 (7)	0.327 (8)	2.695
InnoPeak-SR	26.54 (5)	0.746 (8)	0.302 (5)	2.740
ITS425	27.08 (2)	**0.779 (1)**	0.325 (6)	2.770
MLP-SR	24.87 (15)	0.681 (14)	0.325 (7)	2.905
Webbzhou	26.10 (9)	0.764 (3)	0.341 (9)	-
SR-DL	25.67 (11)	0.718 (10)	0.364 (10)	-
TeamAY	**27.09 (1)**	0.773 (2)	0.369 (11)	-
BIGFEATURE-CAMERA	26.18 (7)	0.750 (6)	0.372 (12)	-
BMIPL-UNIST-YH-1	26.73 (3)	0.752 (5)	0.379 (13)	-
SVNIT1-A	21.22 (19)	0.576 (19)	0.397 (14)	-
KU-ISPL2	25.27 (14)	0.680 (15)	0.460 (15)	-
SuperT	25.79 (10)	0.699 (12)	0.469 (16)	-
GDUT-wp	26.11 (8)	0.706 (11)	0.496 (17)	-
SVNIT1-B	24.21 (17)	0.617 (18)	0.562 (18)	-
SVNIT2	25.39 (13)	0.674 (16)	0.615 (19)	-
Bicubic	25.48 (-)	0.680 (-)	0.612 (-)	3.050
ESRGAN Supervised	24.74 (-)	0.695 (-)	0.207 (-)	2.300

Table 2. **Quantitative results for the NTIRE** 2020 **Challenge on Real-World Image Super-Resolution: Track** 1. The number in () indicates ranking of each metric. Note that the Mean Opinion Score (MOS) metric is measured according to human study.

by noise injection is closer to the real noise. Our RealSR results are more clear with no ambiguity compared with EDSR, ESRGAN and ZSSR. This reason is that their methods are all trained on bicubic data without estimating blurry kernel from real images. In addition, we use perceptual loss that pays more attention to the visual characteristics of the image. Compared with EDSR using pixel-loss, our results have more clear details. What's more, the cost of training a new ZSSR or K-ZSSR model is much higher than inference, while our method costs only forward time during inference.

4.5. NTIRE 2020 Challenge

Our RealSR is the winner of NTIRE 2020 *Challenge on both tracks of Real-World Super-Resolution [26]*, where Track 1 is synthetic corrupted data via image processing artifacts and Track 2 is real data of smartphone images. The data provided by each track includes two domains. One is source domain dataset containing noise and blur, and the other is defined clean HR target dataset. The task is to enlarge the resolution of LR image by 4 times, and keep the clarity and sharpness of the generated SR image consistent with the given target dataset. Since there is no given pair of data for training, participants need to use these two sets of images to construct training data. We applied the proposed method and achieved the best results on both tracks as shown in Tables 2 and 3. Note that the final decision is based on human study, *i.e.*, Mean Opinion Score (MOS) for Track 1 and Mean Opinion Rank (MOR) for Track 2 [26]. Our method outperforms other approaches by a large margin, and generates SR images with superior sharpness and clarity.

4.6. Ablation Study

In order to further verify the necessity of estimating kernel, injecting noise during the degradation process, and the patch discriminator during SR training, we conduct ablation experiments on the DPED dataset. We first introduce the settings of each experiment.

- Bicubic: Under this setting, we adopt bicubic kernel to downsample HR images, and then directly use these paired data to train SR model. Without kernel estimation and noise injection, this setting keeps other parameters as default, which can be understood as fine-tuning ESRGAN on the real dataset to verify its robustness.

- Noise: This setting is to add noise injection on the basis of bicubic. Because the kernel estimation method is not used, this setting can be observed to verify the validity of the kernel estimation when compared with the proposed complete method.

- Kernel: This setting only uses the kernel estimation method, but no explicit noise is added, so it can be used to observe the effect of noise injection on the result.

- VGG-128: As discussed in Section 3.5, this setting uses the default VGG-128 discriminator.

- Patch: This setting uses a lighter patch discriminator, which is compared with the previous four settings to verify our conclusion.

Next, we demonstrate three comparative analysis to verify the effectiveness of the three proposed components.

Effect of the Kernel Estimation It can be seen from Figure 5 that the generated results 'Patch' are more clear compared with 'Noise'. This proves that the kernel estimation is important to SR training, which helps SR models produce sharper edges.

Effect of the Noise Injection In this comparative experiment, we set noise injection as an option to verify if noise injection is necessary. It can be seen from Figure 5 that without explicit noise injection, the results of 'Kernel' have a lot of artifacts, which are very similar to the ESRGAN results trained on clean data. The injected noise is consistent with the original noise distribution, thus ensuring SR models robust to noise during testing.

Effect of the Patch Discriminator On real data, we use patch discriminator to replace VGG-128. Comparing 'Patch' with 'VGG-128', we show that the VGG-128 discriminator with excessively large receptive field will cause

Team	NIQE↓	BRISQUE↓	PIQE↓	NRQM↑	PI↓	IQA-Rank↓	MOR↓
Impressionism (ours), *winner*	**5.00 (1)**	**24.4 (1)**	17.6 (2)	**6.50 (1)**	**4.25 (1)**	**3.958**	**1.54 (1)**
AITA-Noah-A	5.63 (4)	33.8 (5)	29.7 (8)	4.23 (8)	5.70 (6)	7.720	3.04 (2)
ITS425	8.95 (18)	52.5 (18)	88.6 (18)	3.08 (18)	7.94 (18)	14.984	3.30 (3)
AITA-Noah-B	8.18 (17)	50.1 (12)	88.0 (17)	3.23 (15)	7.47 (17)	13.386	3.57 (4)
Webbzhou	7.88 (15)	51.1 (15)	87.8 (16)	3.27 (14)	7.30 (15)	12.612	4.44 (5)
Relbmag-Eht	5.58 (3)	33.1 (3)	**12.5 (1)**	6.22 (2)	4.68 (2)	4.060	-
MSMers	5.43 (2)	38.2 (7)	20.5 (3)	5.22 (5)	5.10 (3)	5.420	-
MLP-SR	6.45 (8)	30.6 (2)	29.0 (6)	6.12 (3)	5.17 (4)	5.926	-
SR-DL	6.11 (5)	33.5 (4)	29.4 (7)	5.24 (4)	5.43 (5)	6.272	-
InnoPeak-SR	7.42 (13)	39.3 (8)	21.5 (4)	5.12 (6)	6.15 (9)	7.716	-
QCAM	6.21 (6)	44.2 (9)	49.6 (9)	4.10 (10)	6.05 (8)	8.304	-
SuperT	6.94 (10)	50.2 (13)	75.1 (11)	4.23 (9)	6.35 (10)	9.612	-
KU-ISPL	6.79 (9)	45.1 (10)	61.6 (10)	3.60 (13)	6.59 (12)	10.152	-
BMIPL-UNIST-YH-1	7.03 (12)	50.2 (14)	81.5 (13)	3.70 (12)	6.66 (13)	12.218	-
BIGFEATURE-CAMERA	7.45 (14)	49.2 (11)	87.1 (14)	3.23 (16)	7.11 (14)	13.784	-
Bicubic	7.97 (16)	52.0 (17)	87.2 (15)	3.16 (17)	7.40 (16)	14.532	6.04 (6)
RRDB	7.01 (11)	51.3 (16)	76.0 (12)	4.06 (11)	6.48 (11)	10.042	6.06 (7)

Table 3. **Quantitative results for NTIRE** 2020 **Challenge on Real-World Image Super-Resolution: Track** 2. The number in () indicates ranking of each metric. Several no-reference based image quality assessment (IQA) is used to provide computed evaluation. The NIQE [29], BRISQUE [28], and PIQE [37] metric is calculated using their corresponding MATLAB implementations. NRQM [27] is a learned IQA score. Moreover, PI [16] and IQA-Rank indicate summary of the other computed IQA metrics. Note that the final ranking is based on Mean Opinion Rank (MOR).

LR Bicubic Noise Kernel VGG-128 Patch

Figure 5. **Qualitative results on DPED dataset compared with 'Bicubic', 'Noise', 'Kernel', 'VGG-**128**' and 'Patch'**. The red and yellow area is cropped from different results and enlarged for visual convenient.

unreal textures, which are partially in conflict with the original image. In contrast, the patch discriminator restores important edge features, and avoids unpleasant artifacts thus generating more realistic details.

5. Conclusion

In this paper, we propose a novel degradation framework RealSR based on kernel estimation and noise injection. By using different combinations of degradation (*e.g.*, blur and noise), we acquire LR images that share a common domain with real images. With those domain-consistent data, we then train a real image super-resolution GAN with a patch discriminator, which can produce HR results with better perception. Experiments on both synthetic noise data and real-world images show our RealSR outperforms the state-of-the-art methods, resulting in lower noise and better visual quality. Furthermore, our RealSR is also the winner of NTIRE 2020 Challenge on both tracks of Real-World Super-Resolution, which significantly outperforms other approaches by large margins in human perception.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017.

[2] Simon Baker and Takeo Kanade. Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, 2002.

[3] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. In *Advances in Neural Information Processing Systems*, pages 284–293, 2019.

[4] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3086–3095, 2019.

[5] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Camera lens super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1652–1660, 2019.

[6] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2018.

[7] Yu Chen, Ying Tai, Xiaoming Liu, Chunhua Shen, and Jian Yang. Fsrnet: End-to-end learning face super-resolution with facial priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[8] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11065–11074, 2019.

[9] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[10] Manuel Fritsche, Shuhang Gu, and Radu Timofte. Frequency separation for real-world super-resolution. *arXiv preprint arXiv:1911.07850*, 2019.

[11] Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. Blind super-resolution with iterative kernel correction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1604–1613, 2019.

[12] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018.

[13] Xiangyu He, Zitao Mo, Peisong Wang, Yang Liu, Mingyuan Yang, and Jian Cheng. Ode-inspired network design for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1732–1741, 2019.

[14] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: a magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1575–1584, 2019.

[15] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Dslr-quality photos on mobile devices with deep convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3277–3285, 2017.

[16] Andrey Ignatov, Radu Timofte, Thang Van Vu, Tung Minh Luu, Trung X Pham, Cao Van Nguyen, Yongwoo Kim, Jae-Seok Choi, Munchurl Kim, Jie Huang, et al. Pirm challenge on perceptual image enhancement on smartphones: Report. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[19] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.

[20] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017.

[21] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[22] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3867–3876, 2019.

[23] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.

[24] Andreas Lugmayr, Martin Danelljan, and Radu Timofte. Unsupervised learning for real-world super-resolution. In *ICCV Workshops*, 2019.

[25] Andreas Lugmayr, Martin Danelljan, Radu Timofte, et al. Aim 2019 challenge on real-world image super-resolution: Methods and results. In *ICCV Workshops*, 2019.

[26] Andreas Lugmayr, Martin Danelljan, Radu Timofte, et al. Ntire 2020 challenge on real-world image super-resolution: Methods and results. *CVPR Workshops*, 2020.

[27] Chao Ma, Chih-Yuan Yang, Xiaokang Yang, and Ming-Hsuan Yang. Learning a no-reference quality metric for single-image super-resolution. *Computer Vision and Image Understanding*, 158:1–16, 2017.

[28] A Mittal, AK Moorthy, and AC Bovik. Referenceless image spatial quality evaluation engine. In *45th Asilomar Conference on Signals, Systems and Computers*, volume 38, pages 53–54, 2011.

[29] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2012.

[30] Jinshan Pan, Sifei Liu, Deqing Sun, Jiawei Zhang, Yang Liu, Jimmy Ren, Zechao Li, Jinhui Tang, Huchuan Lu, Yu-Wing Tai, et al. Learning dual convolutional neural networks for low-level vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3070–3079, 2018.

[31] Yajun Qiu, Ruxin Wang, Dapeng Tao, and Jun Cheng. Embedded block residual network: A recursive restoration model for single-image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4180–4189, 2019.

[32] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3118–3126, 2018.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[34] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[35] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017.

[36] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017.

[37] N Venkatanath, D Praneeth, Maruthi Chandrasekhar Bh, Sumohana S Channappayya, and Swarup S Medasani. Blind image quality evaluation using perception based features. In *2015 Twenty First National Conference on Communications (NCC)*, pages 1–6. IEEE, 2015.

[38] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 606–615, 2018.

[39] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[40] Xiangyu Xu, Yongrui Ma, and Wenxiu Sun. Towards real scene super-resolution with raw images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1723–1731, 2019.

[41] Xi Yin, Ying Tai, Yuge Huang, and Xiaoming Liu. Fan: Feature adaptation network for surveillance face recognition and normalization. *arXiv:1911.11680v1*, 2019.

[42] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Deep plug-and-play super-resolution for arbitrary blur kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1671–1681, 2019.

[43] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[44] Wenlong Zhang, Yihao Liu, Chao Dong, and Yu Qiao. Ranksrgan: Generative adversarial networks with ranker for image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3096–3105, 2019.

[45] Xuaner Zhang, Qifeng Chen, Ren Ng, and Vladlen Koltun. Zoom to learn, learn to zoom. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3762–3770, 2019.

[46] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 286–301, 2018.

[47] Zhifei Zhang, Zhaowen Wang, Zhe Lin, and Hairong Qi. Image super-resolution by neural texture transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7982–7991, 2019.

[48] Haitian Zheng, Mengqi Ji, Haoqian Wang, Yebin Liu, and Lu Fang. Crossnet: An end-to-end reference-based super resolution network using cross-scale warping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 88–104, 2018.

[49] Ruofan Zhou and Sabine Susstrunk. Kernel modeling super-resolution on real low-resolution images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2433–2443, 2019.

[50] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.