Human Guided Ground-truth Generation for Realistic Image Super-resolution

Du Chen[1]*, Jie Liang[1,2]*, Xindong Zhang[1,2], Ming Liu[1,3], Hui Zeng[2] and Lei Zhang[1,2]†
[1]The Hong Kong Polytechnic University, [2]OPPO Research Institute, [3]Harbin Institute of Technology
{csdud.chen, c-ming.liu}@connet.polyu.hk, {liang27jie, cshzeng}@gmail.com
{csxdzhang, cslzhang}@comp.polyu.edu.hk

Abstract

How to generate the ground-truth (GT) image is a critical issue for training realistic image super-resolution (Real-ISR) models. Existing methods mostly take a set of high-resolution (HR) images as GTs and apply various degradations to simulate their low-resolution (LR) counterparts. Though great progress has been achieved, such an LR-HR pair generation scheme has several limitations. First, the perceptual quality of HR images may not be high enough, limiting the quality of Real-ISR outputs. Second, existing schemes do not consider much human perception in GT generation, and the trained models tend to produce over-smoothed results or unpleasant artifacts. With the above considerations, we propose a human guided GT generation scheme. We first elaborately train multiple image enhancement models to improve the perceptual quality of HR images, and enable one LR image having multiple HR counterparts. Human subjects are then involved to annotate the high quality regions among the enhanced HR images as GTs, and label the regions with unpleasant artifacts as negative samples. A human guided GT image dataset with both positive and negative samples is then constructed, and a loss function is proposed to train the Real-ISR models. Experiments show that the Real-ISR models trained on our dataset can produce perceptually more realistic results with less artifacts. Dataset and codes can be found at https://github.com/ChrisDud0257/HGGT.

1. Introduction

Owing to the rapid development of deep learning techniques [14, 18, 19, 22, 44], the recent years have witnessed the great progress in image super-resolution (ISR) [2, 8–10, 12, 13, 23, 26–29, 31–33, 35, 45, 46, 48, 51, 52, 54, 56], which aims at generating a high-resolution (HR) version of the low-resolution (LR) input. Most of the ISR models (*e.g.*, CNN [37, 38] or transformer [5, 29] based ones) are trained on a large amount of LR-HR image pairs, while the generation of LR-HR image pairs is critical to the real-world performance of ISR models.

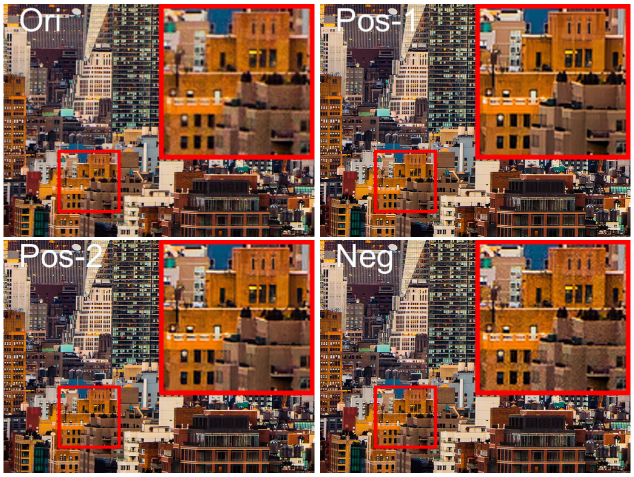


Figure 1. From left to right and top to bottom: one original HR image (Ori) in the DIV2K [1] dataset, two of its enhanced positive versions (Pos-1 and Pos-2) and one negative version (Neg). The positive versions generally have clearer details and better perceptual quality, while the negative version has some unpleasant visual artifacts. **Please zoom in for better observation.**

Most of the existing ISR methods take the HR images (or after some sharpening operations [46]) as ground-truths (GTs), and use them to synthesize the LR images to build the LR-HR training pairs. In the early stage, bicubic downsampling is commonly used to synthesize the LR images from their HR counterparts [8, 9, 23, 33, 42, 56]. However, the ISR models trained on such HR-LR pairs can hardly generalize to real-world images whose degradation process is much more complex. Therefore, some researchers proposed to collect HR-LR image pairs by using long-short camera focal lengths [3, 4]. While such a degradation process is more reasonable than bicubic downsampling, it only covers a small subspace of possible image degradations. Recently, researchers [12, 20, 30, 32, 34, 46, 50, 51, 59] have proposed

*Equal contribution.
†Corresponding author. This work is supported by the Hong Kong RGC RIF grant (R5001-18) and the PolyU-OPPO Joint Innovation Lab.

to shuffle or combine different degradation factors, such as Gaussian/Poisson noise, (an-)isotropic blur kernel, downsampling/upsampling, JPEG compression and so on, to synthesize LR-HR image pairs, largely improving the generalization capability of ISR models to real-world images.

Though great progress has been achieved, existing LR-HR training pair generation schemes have several limitations. First, the original HR images are used as the GTs to supervise the ISR model training. However, the perceptual quality of HR images may not be high enough (Fig. 1 shows an example), limiting the performance of the trained ISR models. Second, existing schemes do not consider much human perception in GT generation, and the trained ISR models tend to produce over-smoothed results. When the adversarial losses [27, 40, 48] are used to improve the ISR details, many unpleasant artifacts can be introduced.

In order to tackle the aforementioned challenges, we propose a human guided GT data generation strategy to train perceptually more realistic ISR (Real-ISR) models. First, we elaborately train multiple image enhancement models to improve the perceptual quality of HR images. Meanwhile, one LR image can have multiple enhanced HR counterparts instead of only one. Second, to discriminate the visual quality between the original and enhanced images, human subjects are introduced to annotate the regions in enhanced HR images as "Positive", "Similar" or "Negative" samples, which represent better, similar or worse perceptual quality compared with the original HR image. Consequently, a human guided multiple-GT image dataset is constructed, which has both positive and negative samples.

With the help of human annotation information in our dataset, positive and negative LR-GT training pairs can be generated (examples of the positive and negative GTs can be seen in Fig. 1), and a new loss function is proposed to train the Real-ISR models. Extensive experiments are conducted to validate the effectiveness and advantages of the proposed GT image generation strategy. With the same backbone, the Real-ISR models trained on our dataset can produce more perceptually realistic details with less artifacts than models trained on the current datasets.

2. Related Work

According to how the LR-HR image pairs are created, the existing ISR methods can be categorized into three major groups: simple degradation based, long-short focal length based, and complex degradation based methods.

Simple Degradation based Training Pairs. Starting from SRCNN [8, 9], most of the deep learning based ISR methods synthesize the LR images from their HR counterparts by bicubic downsampling or direct downsampling after Gaussian smoothing. By using such a simple degradation model to generate a large amount of training data, researchers focus more on the ISR network module design, such as residual [23]/dense [58] connection, channel-attention [6, 17, 56], multiple receptive field [16, 28] or self-attention [5, 29, 54]. The fidelity based measures, such as PSNR and SSIM [49], are used to evaluate and compare the performance of different ISR methods. Later on, many works [27, 31, 35, 39–41, 47, 48] have been developed to adopt the Generative Adversarial Network (GAN) [11] techniques to train Real-ISR models so as to produce photo-realistic textures and details.

Long-short Focal Length based Training Pairs. Instead of synthesizing LR-HR pairs using simple degradation operators, researchers have also tried to use long-short camera focal length to collect real-world LR-HR pairs. The representative works include CameraSR [4] and RealSR [3]. The former builds a dataset using DSLR and mobile phone cameras to model degradation between the image resolution and field-of-view. The latter utilizes different focal lengths of the DSLR camera to shot the same scene at different resolutions, and employs an image registration method to crop and align the LR-HR image pairs. Nonetheless, ISR models trained on those datasets might fail when applied to images from different resources (*e.g.*, different degradation, different focal length and cameras).

Complex Degradation based Training Pairs. The image degradation in real-world scenarios can be too complex to model using a simple operator. To enable the Real-ISR models having higher generalization capability, BSRGAN [51] and Real-ESRGAN [46] have been proposed to synthesize LR-HR training pairs with more complex image degradations. They employ a set of degradation factors, such as different types of noise, blur kernels, scaling factors, JPEG compression, *etc*., to enlarge the degradation space. BSRGAN [51] shuffles and combines different degradations, while Real-ESRGAN [46] employs a two-stage synthesis progress. In DASR [32], Liang *et al*. partitioned the complex degradation space into different levels, and proposed a degradation adaptive method for Real-ISR.

Other Training Pairs. Beside the above three groups of ISR methods, MCinCGAN [57] and Pseudo-SR [36] utilize unpaired training images to do unsupervised learning. They utilize one or more discriminators to tell the HR GT from the unpaired SR output. AdaTarget [21] employs a transformation CNN block to generate a training-friendly GT from the original GT during the training progress. Nevertheless, the quality of the generated training-friendly GT might not have a good perception quality.

3. Human Guided Ground-truth Generation

3.1. Overview

As discussed in Section 2, almost all existing methods [8, 9, 15, 25, 34, 37, 48, 55] directly take the HR images as the GT to construct the training pairs. Unfortunately, the

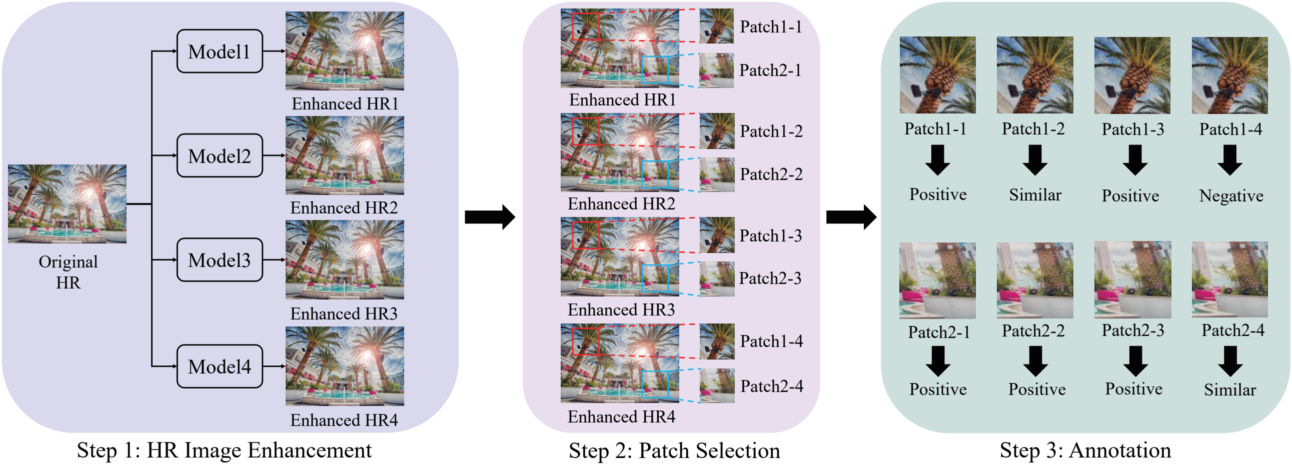


Figure 2. Illustration of our human guided ground-truth (GT) generation process. We first train four image enhancement models to enhance the original high-resolution (HR) image, and then extract the patches which have rich textural and structural details while having certain differences between the original and enhanced versions. Finally, human subjects are involved to annotate the extract patches as "Positive", "Similar" and "Negative" samples.

perceptual quality of many HR images may not be good enough to serve as GTs, limiting the performance trained Real-ISR models. Therefore, we propose to enhance the quality of HR images so that they can better serve as GTs. In particular, human guidance can be introduced in the GT generation process so that perceptually more realistic Real-ISR models can be trained.

As illustrated in Fig. 2, the proposed human guided GT generation method has three steps. First, we elaborately train multiple image enhancement models to improve the perceptual quality of HR images. Second, those patches which have enough textural and structural details and have certain differences between the enhanced version and the original version are extracted. Third, human subjects are introduced to discriminate the visual quality between the enhanced patches and the original patch, and label them as "Positive" (*i.e.*, better quality), "Similar" (*i.e.*, similar quality) or "Negative" (*i.e.*, worse quality) samples. In the following subsections, we describe these three steps in detail.

3.2. Design of the Enhancement Models

In order to generate visually more pleasing GTs from the original HR image, we train multiple image enhancement models and apply them to the HR image. To this end, the commonly used DF2K-OST dataset (including 800 high quality images from DIV2K [1], 2650 high quality images from Flickr2K [43] and 10,324 images from OST [47]) is employed. The original images are denoted by I^H, and the low quality ones, denoted by I^L, are degraded from I^H by using the following degradation model [46, 51]:

$$I^L = [(I^H \otimes K)_R + V]_J, \tag{1}$$

where K means isotropic/an-isotropic blur kernel, R means resize operation, V is Gaussian/Poisson noise and J denotes JPEG compression. With (I^L, I^H) as training pairs, we can train enhancement models. Note that before inputting the low-quality image I^L into the model, we resize it to the size of I^H since here we are training enhancement models, where the input and output have the same size.

Considering that the quality of HR image to be further enhanced is generally not bad, we deliberately control the degradation settings in Eq. (1) within weak to middle levels. Otherwise, the learned models can over-enhance the HR images and generate many artifacts. Since the major issues of real world images are noise corruption and blurring, we employ two degradation settings, one focusing on processing slightly high noise and the other focusing on dealing with slightly strong blur. The detailed degradation settings can be found in the **supplementary file**.

We select one CNN-based network RCAN [56] and one transformer-based network ELAN [54] as the backbones of our enhancer. RCAN [56] adopts deep residual learning together with channel-attention [18], while ELAN [54] employs a multi-scale self-attention [44] block to extract long-range independence. We remove the up-sampling layer in those models since the input and output share the same size in our case. We choose both CNN and transformer as backbones because though transformers have stronger capability in restoring large scale structures and repetitive patterns, CNN can better characterize some small scale and local image details. With the two different degradation settings and two different backbones, we train four image enhancement models with L_1, perceptual and adversarial losses. The UNet discriminator [46] is used in adversarial training.

3.3. Patch Selection and Annotation

We apply the trained four enhancement models to 1,600 HR images collected from three representative resources: 1) 800 images from the DIV2K [1] dataset; 2) 400 images from Internet which could be used for free, such as Pixabay (https://pixabay.com) and Unsplash (https://unsplash.com); 3) 400 images shot by us using mobile phones. Note that though those HR images have high resolution (2K∼4K), they could contain certain noise, blurred details or other real-world degradations, as we shown in Fig. 1. It is expected that their perceptual quality can be improved by our enhancement models so that they can better serve as GTs in Real-ISR model training.

After applying the four enhancement models to the 1,600 HR images, we obtain 6,400 enhanced images. However, it is inappropriate to directly take them as GTs. On one hand, many regions in these images are smooth and less informative. On the other hand, there is no guarantee that the enhancement models can always produce perceptually better outputs in all regions. Therefore, we extract patches from those enhanced images and invite human volunteers to label them. In specific, we randomly crop $512*512$ patches from each image with the overlapping area less than $1/2$ of patch area. We then filter out the patches that have large smooth background regions according to the quantity of details and textures, which is measured by the standard deviation (std) of the patch in image domain and the std of high-frequency components in a Laplacian pyramid. At last, we remove the patches on which the difference between the original version and the enhanced version is small (*i.e.*, no much enhancement). The patch selection process avoids the cost of annotating flat patches, and can speed up the training process since flat patches have small gradients. Finally, we select 20,193 groups of patches of $512*512$ size, each group having one original HR patch and 4 enhanced patches.

We then invite 60 volunteers with different background to annotate the quality of enhanced patches by comparing them with the original HR patch. A software program, whose interface is shown in the **supplementary file**, is developed for this purpose. The original patch is positioned at the left side of the screen, while the four enhanced versions are located on the right side in random order. Those patches whose perceptual quality is better than the original one are labeled as "Positive", and the patches with worse perceptual quality are labeled as "Negative". In case the quality is tied, the enhanced patch will be labeled as "Similar". Before annotating, all volunteers are briefly trained to ensure that they will focus on the image perceptual quality (*e.g.*, sharpness, noise, details, artifacts, *etc*.) but not on the image content.

3.4. Statistics of the Annotated Dataset

We invite 60 volunteers to annotate the 20,193 patch groups, each consisting of an original HR patch and 4 enhanced patches. Each group is annotated by 3 different volunteers, and each volunteer is assigned with about 1,010 groups to annotate. In total, we obtain 20,193 groups of annotations, and $20,193 \times 4 \times 3 = 242,316$ annotated patches. The average annotation time for one group is 22.79s.

Table 1. The distribution of annotations in our dataset. There are 20,193 groups of patches, while each group consists of an original HR patch and 4 enhanced patches. Each enhanced patch is annotated by 3 different volunteers, resulting in a total of $20,193 \times 4 \times 3 = 242,316$ annotations.

Label	Enhance Model				Total
	1	2	3	4	
Positive	42362	39031	47251	47398	176042
Similar	14623	17615	10259	8407	50904
Negative	3594	3933	3069	4774	15370
Total	60579	60579	60579	60579	242316

Distribution of the patch annotations. Tab. 1 shows the distribution of "Positive", "Similar" and "Negative" labels for each enhancement model, as well as the overall distribution. We see that there are overall 176,042 "Positive" (72.65%), 50,904 "Similar" (21.00%) and 15,370 "Negative" (6.35%) patches. Such statistics imply that our enhancement models improve the visual quality of HR patches in most cases, but there are indeed some bad cases.

Distribution of the final patch labels. For each of the $20,193 \times 4 = 80,772$ enhanced patches, we have three annotations from three different volunteers. We take the majority as the final label of the patch, *i.e.*, if one patch has two or three same annotations, it will be labeled by that annotation. In case the three annotations are different from each other (*i.e.*, one "Positive", one "Similar" and one "Negative"), the final label is marked as "Similar". Tab. 2 shows the distribution of the final labels of the enhanced patches. We can see that finally there are 63,583 "Positive" (78.72%), 14,675 "Similar" (18.17%) and 2,514 "Negative" (3.11%) patches. Most of the final labels are "Positive" ones, and only a small portion (3.11%) are "Negative" ones. The maximum divergence of "Positive" labels is 3,329 (5.24%) between Model 2 and Model 3. The examples of "Positive", "Similar" and "Negative" patches can be found in the **supplementary file**.

Distribution of the number of final "Positive" patches per group. For each group of patches, there can be $0 \sim 4$ final "Positive" samples. Tab. 3 shows the distribution of the number of final "Positive" patches per group. One can see that among the 20,193 groups, 11,413 (56.52%) groups have 4 "Positive" patches, 3,901 (19.32%) have 3 "Positive" patches, 2,616 (12.95%) have 2 "Positive" patches, 996 (4.93%) have 1 "Positive" patch, and 1,267 (6.28%) have none. We will use those "Positive" patches as "Positive" GTs, and those "Negative" patches as "Negative" GTs

Table 2. The distribution of final patch labels in our dataset. There are $20,193 \times 4 = 80,772$ enhanced patches, each having three annotations. We take the majority annotation label as the final label of each patch.

Final Label	Enhance Model 1	2	3	4	Total
Positive	15250	13907	17236	17190	63583
Similar	4379	5635	2517	2144	14675
Negative	564	651	440	859	2514
Total	20193	20193	20193	20193	80772

Table 3. The distribution of the number ($0 \sim 4$) of final "Positive" patches per group in our dataset.

"Positive" Count	0	1	2	3	4	Total
Groups count	1267	996	2616	3901	11413	20193

to train Real-ISR models. The patches with "Similar" labels are not employed in our training progress.

4. Training Strategies

As described in Sec. 3, for an original HR patch, denoted by I^H, we may have several (less than 4) positive GTs, denoted by I^{Pos}, and several negative GTs, denoted by I^{Neg}. To construct the positive or negative LR-GT pairs for Real-ISR model training, we apply the degradation model in Eq. 1 to I^H and obtain the corresponding LR image, denoted by I^L. (The setting of degradation parameters will be discussed in Sec. 5.1). In total, there are 63,583 positive LR-GT pairs (I^L, I^{Pos}) and 2,514 negative LR-GT pairs (I^L, I^{Neg}). Note that in our dataset, one LR image may correspond to multiple positive GTs or negative GTs.

Training with positive pairs only. By removing those groups that do not have any positive GT from the 20,193 training groups, we have 18,926 groups with $1 \sim 4$ GTs, and 63,583 positive LR-GT pairs to train Real-ISR models. As in previous works [46, 51], we employ the L_1 loss, perceptual loss L_p and GAN loss L_{GAN} to train the model. Since one LR image I^L may have multiple positive GTs, each time we randomly choose one positive GT to calculate the L_1, L_p and L_{GAN} losses of the corresponding LR image I^L, and update the discriminator and generator networks. The overall training loss is as follows:

$$L_{Total} = \alpha L_1 + \beta L_p + \gamma L_{adv}, \tag{2}$$

where α, β and γ are balance parameters.

Training with both positive and negative pairs. By filtering out those groups that only contain "Similar" GTs, we obtain 19,272 groups that have at least one "Positive" or "Negative" GT, totally 63,583 positive LR-GT pairs and 2,514 negative LR-GT pairs. When training with the positive GTs, we adopt the same strategy as described above. For each negative LR-GT pair, we introduce a negative loss, denoted by L_{neg}, to update the model.

It is observed that most of the negative GTs have over-sharpened details, strong noise or false details (example images are provided in the **supplementary file**). Inspired by LDL [31], we build a map $\boldsymbol{M}^{Neg}$ to indicate the local residual variation of a negative GT, which is defined as $\boldsymbol{M}^{Neg}_{i,j} = var(\boldsymbol{R}^{Neg}_{i,j}(3,3))^a$, where $\boldsymbol{R}^{Neg} = |I^{Neg} - I^H|$ is the residual between the original HR image and the negative GT, $\boldsymbol{R}^{Neg}_{i,j}(3,3)$ is a local 3×3 window of $\boldsymbol{R}^{Neg}$ centered at (i,j), var denotes the variance operation and a is the scaling factor (we set a to $\frac{3}{4}$ in our experiments).

Similarly, we can build a residual variation map $\boldsymbol{M}^{Pos}_{i,j} = var(\boldsymbol{R}^{Pos}_{i,j}(3,3))^a$ for the positive GT, where $\boldsymbol{R}^{Pos} = |I^{Pos} - I^H|$. At location (i,j), if the negative residual variation is higher than the positive one, we identify this pixel at I^{Neg} as a truly negative pixel, which should be used to update the model. Therefore, we first define an indication map $\boldsymbol{M}^{Ind}_{i,j}$:

$$\boldsymbol{M}^{Ind}_{i,j} = \begin{cases} 0, & \boldsymbol{M}^{Neg}_{i,j} <= \boldsymbol{M}^{Pos}_{i,j} \\ \boldsymbol{M}^{Neg}_{i,j}, & \boldsymbol{M}^{Neg}_{i,j} > \boldsymbol{M}^{Pos}_{i,j} \end{cases} \tag{3}$$

and then define the negative loss L_{neg} as follows:

$$L_{neg} = ||\boldsymbol{M}^{Ind} \odot (I^{Neg} - I^{SR})||_1, \tag{4}$$

where $\odot$ means dot product.

Finally, the overall training loss is defined as:

$$L_{Total} = \alpha L_1 + \beta L_p + \gamma L_{adv} - \delta L_{neg}, \tag{5}$$

where α, β, γ and δ are balance parameters.

5. Experimental Results

5.1. Experiment Setup

To validate the effectiveness of our human guided GT (HGGT) dataset and the role of negative GTs, we perform two sets of experiments. First, in Sec. 5.2, we train several representative Real-ISR models, such as Real-ESRGAN [46], BSRGAN [51], AdaTarget [21] and LDL [31] on the DF2K-OST dataset and our HGGT dataset, and compare their performance. Second, in Sec. 5.3, we train two commonly used Real-ISR backbones (RRDB [46, 48] and SwinIR [29]) on our dataset by using only the postive GTs and using both the positive and negative GTs.

Implementation details. Before training a model on our dataset, we first pre-train it on the DF2K-OST dataset by using the pixel-wise ℓ_1 loss to get a stable initialization. Since the original degradation settings in Real-ESRGAN [46] and

BSRGAN [51] is too strong to use in practical ISR applications, we adopt a single-stage degradation process, including blur, noise, down-sampling and JPEG compression with moderate intensities. Detailed settings and visual examples are provided in the **supplementary file**. For the two backbones, RRDB and SwinIR, we utilize the UNet discriminator [46] for adversarial training, resulting in a RRDB-GAN model and a SwinIR-GAN model.

We conduct Real-ISR experiments with scaling factor 4 in this paper. We randomly crop training patches of size $256 * 256$ from the GT images, and resize the corresponding regions in the LR images to $64 * 64$. The batch size is set to 12 for RRDB backbone and 8 for SwinIR backbone to save GPU memory. We train our model on one NVIDIA RTX 3090 GPU for 300K iterations using the Adam [24] optimizer. The initial learning rate is set to $1e-4$, and we halve it after 200K iterations for RRDB backbone, and 200K, 250K, 275K and 287.5K iterations for SwinIR backbone. The balance parameters α, β, γ and δ in Eq. 5 are set to 1, 1, 0.1 and 300, respectively. δ is set much larger than others because the number of negative GTs is much smaller than positive ones.

Testing set. To evaluate the performance of Real-ISR models trained on our dataset quantitatively, we construct a test set using the same steps as in the construction of our training set. In specific, 100 patch groups with at least 2 'Positive' GTs are constructed. The input LR patches are generated by using the same degradation process as in the training process. The LR patches together with their GTs are used to quantitatively evaluate the Real-ISR models. We denote this dataset as *Test-100*.

Evaluation protocol. For the quantitative evaluation on *Test-100*, we adopt the commonly used PSNR, SSIM [49] LPIPS [53] and DISTS [7] as quality metrics. Since in *Test-100* one LR image has at least 2 positive GTs, we average the PSNR/SSIM/LPIPS/DISTS scores respectively over the multiple positive GTs as the final scores. For the qualitative evaluation, we invite 12 volunteers to perform subjective assessment, and report the count of preferred Real-ISR models as the user study results.

5.2. DF2K-OST Dataset vs. Our HGGT Dataset

We first evaluate the effectiveness of the proposed dataset by training representative Real-ISR models respectively on the DF2K-OST dataset and the positive GTs of our HGGT dataset. Four state-of-the-art Real-ISR models are employed, including Real-ESRGAN [46], BSRGAN [51], AdaTarget [21] and LDL [31]. For Real-ESRGAN and BSRGAN, we adjust the degradation parameters so that the quality of synthesized training LR images is comparable to the LR images in our test set. For AdaTarget and LDL, we use the single-stage degradation as explained in Sec. 5.1, and employ the loss functions in the original papers. All models are firstly pre-trained on DF2K-OST with ℓ_1 loss. The UNet discriminator [46] is used for adversarial training in our experiments. Quantitative comparison are reported in Table 4 and visual comparisons are shown in Figure 3.

As shown in Table 4, training on our HGGT dataset leads to much better LPIPS/DISTS scores against the DF2K-OST dataset. Specifically, the LPIPS/DISTS scores are significantly improved by 10.14%/12.40%, 16.30%/15.27%, 17.45%/18.91% and 19.23%/21.99%, respectively, for Real-ESRGAN, BSRGAN, LDL and AdaTarget-GAN. This indicates a clear advantage of perceptual quality brought by our dataset. Some visual examples are shown in Figure 3. One can see that the models trained on our positive GTs can produce perceptually more pleasing results against the models trained on DF2K-OST. The reconstructed images by our dataset have sharper textures and richer details. This is because the original GTs in the DF2K-OST dataset have mixed visual qualities, where a large number of local patches are smooth. In comparison, in our HGGT dataset, the perceptual quality of positive GTs is much enhanced, and the smooth or artifactual patches are mannually removed. These improvements on the training data bring clear advantages to the trained Real-ISR models. More visual results are put in the **supplementary file**.

As a common problem of GAN-based models, the superior perceptual quality sacrifices the pixel-wise fidelity which is depicted by PSNR and SSIM. This trade-off, which is mainly caused by the ill-posed nature of the image restoration tasks, has been discussed in previous researches [51]. It is well-known that the pixel-wise metrics do not correlate well to the visual quality [27,40,41]. In addition, in our proposed HGGT dataset, the perceptual quality of GT is improved by using GAN-based enhancement models so that the pixel-wise correlations may not be well-preserved in the data. However, human observers generally prefer the enhanced images in our annotation process, while the perceptually more pleasing results demonstrate the significance of our proposed HGGT dataset in improving the upper bound of the Real-ISR tasks.

The main goal of the proposed HGGT dataset is to improve the perceptual quality of Real-ISR outputs by introducing human perceptions into the training pair generation. We perform a user study to validate the effectiveness of our strategy by inviting 12 volunteers to evaluate the Real-ISR results on the *Test-100* dataset. For each of the four Real-ISR methods, *i.e.*, Real-ESRGAN, BSRGAN, AdaTarget-GAN and LDL, the two models trained on the DF2K-OST dataset and the positive GTs of our HGGT dataset are compared. Each time, the Real-ISR results of the two models on the same LR input are shown to the volunteers in random order, and the volunteers are asked to chose the perceptually better one based on their evaluation. The statistics of the user study are shown in Fig. 5. It should be noted the vol-

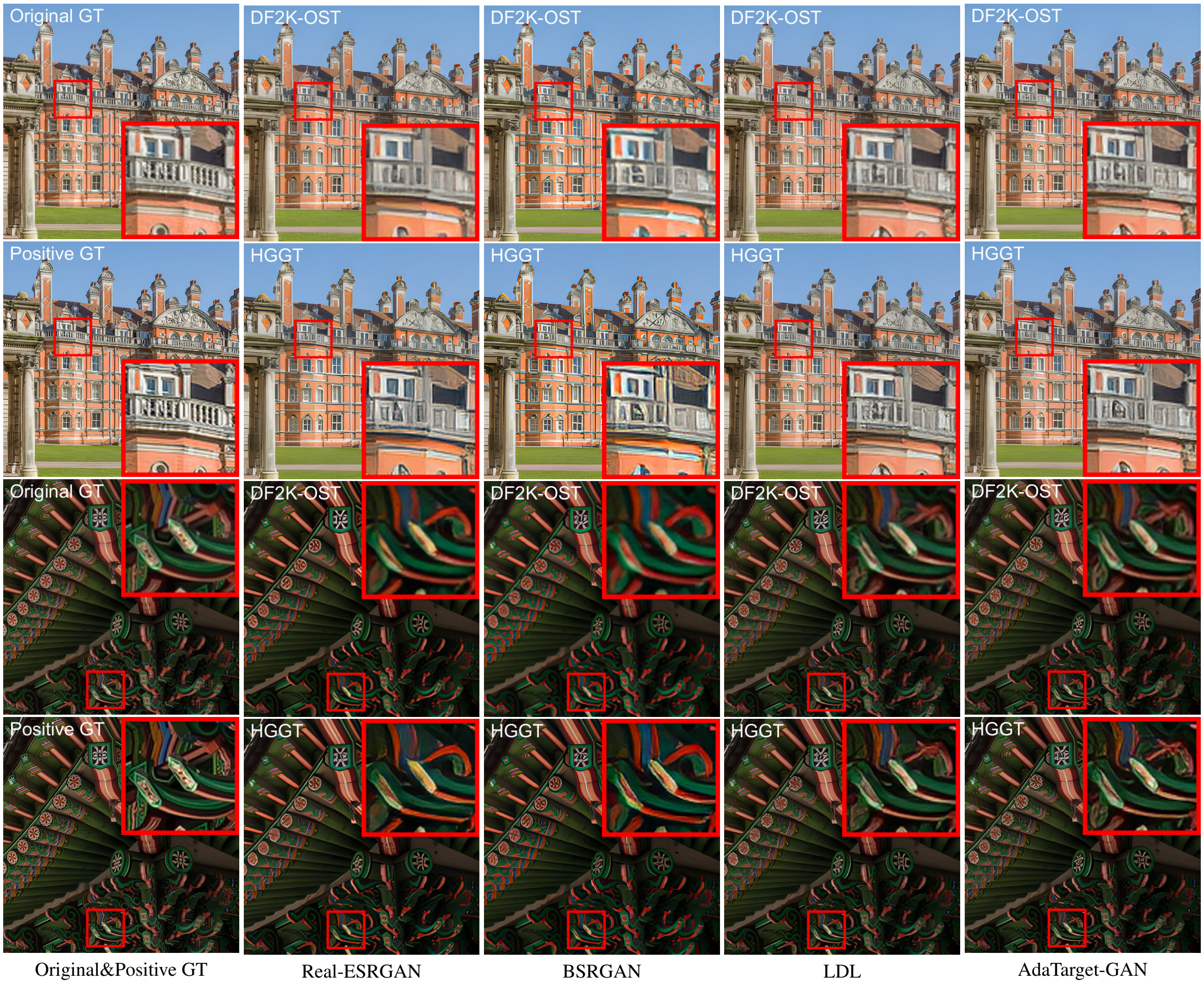


Figure 3. Visual comparison of state-of-the-art models trained on the DF2K-OST and our proposed HGGT datasets. The 1st and 3rd rows show the results of models trained on DF2K-OST, while the 2nd and 4th rows show the results of models trained on ours positive GTs. The left column shows the original GT and the positive GT in our dataset. **Please zoom in for better observation**.

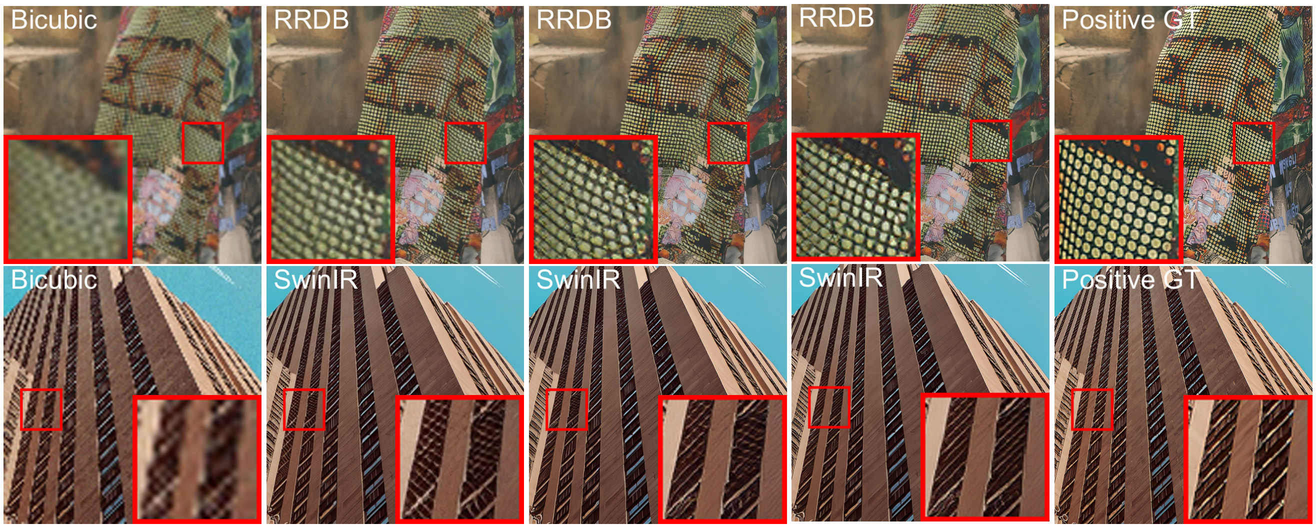
unteers invited in this user study do not participate in the annotation process of our dataset.

As shown in Fig. 5, the majority of participants (more than 80% for all tests) prefer the models trained on our HGGT dataset. This validates the effectiveness of the proposed approach and the dataset, which can be plug-and-play to most of the existing Real-ISR methods and improve their performance by a large margin. For the images where models trained on DF2K-OST are selected, we observe that they mostly contain much flat and smooth regions, and the results of the two models are actually very close.

5.3. The Effectiveness of Negative GTs

We then evaluate the effectiveness of the negative GTs in our HGGT dataset. We first train the baseline model on the original HR images that are used to build our dataset. Then, we train the models on positive GTs only by using Eq. (2), as illustrated in Section 4. Finally, we train the models on both positive and negative GTs by using Eq. (5). The CNN-based RRDB and transformer-based SwinIR backbones are used to train Real-ISR models. Due to the limit of space, quantitative comparisons of the trained models are reported in the **supplementary file**.

Visual comparisons are shown in Figure 4, which provides more intuitive evidences on the effectiveness of the annotated negative GTs. As shown in the second column, the models trained on original HR images yield blurry details and irregular patterns, especially on the area with dense textures. This is mainly caused by the low and mixed quality of the original HR images. In contrast, training on our



Bicubic | Trained on Ori HR | Trained on Pos GT | Trained on Pos+Neg GT | Positive GT

Figure 4. Visualizations of RRDB-GAN ans SwinIR-GAN models trained on the original HR (Ori HR) patches, positive GTs (Pos GT) only, and both positive and negative GTs (Pos+Neg GT) in our HGGT dataset. The top and bottom rows show the results of RRDB-GAN and SwinIR-GAN, respectively. From left to right are the results of bicubic interpolation and the models trained on the Ori HR, Pos GT, Pos+Neg GT, respectively. **Please zoom in for better observation**.

Table 4. The quantitative results of different Real-ISR models trained on DF2K-OST and our HGGT datasets on *Test-100*.

Method	Train Dataset	PSNR/SSIM/LPIPS/DISTS
Real-ESRGAN	DF2K-OST	21.9797/0.6173/0.2593/0.1806
	Positive GT	21.5379/0.6078/0.2330/0.1582
BSRGAN	DF2K-OST	21.7083/0.6092/0.2865/0.1880
	Positive GT	20.9037/0.5898/0.2398/0.1593
LDL	DF2K-OST	22.4724/0.6394/0.2304/0.1676
	Positive GT	22.0190/0.6325/0.1902/0.1359
AdaTarget-GAN	DF2K-OST	22.3944/0.6360/0.2335/0.1687
	Positive GT	21.9216/0.6301/0.1886/0.1316

positive GTs can produce much sharper textures and richer details, whereas there remain some false details and visual artifacts (see the windows of the building). Further, training on both positive and negative GTs leads to a more balanced visual performance. Some over-enhanced local pixels can be suppressed, while the textures remain sharp and regular. This is owing to the effective annotation of negative samples, which bring useful human perception guidance into the data for model training. More visual results can be found in the **supplementary file**.

6. Conclusion

In this paper, we elaborately designed a human guided ground-truth (GT) generation method for realistic image super-resolution (Real-ISR). We first trained four image enhancement models to improve the perceptual quality of original high resolution images, and then extracted structural and textural patches from the enhanced images. Finally, human subjects were invited to annotate the perceptual quality of extracted patches as positive and negative GTs, resulting in the human guided ground-truth (HGGT) dataset. The sharper textures and richer details in the positive GTs could largely improve the performance of trained Real-ISR models, while the negative GTs could provide further guidance for the model to avoid generating visual artifacts. Extensive experiments validated the effectiveness of the proposed HGGT dataset and the training strategies.

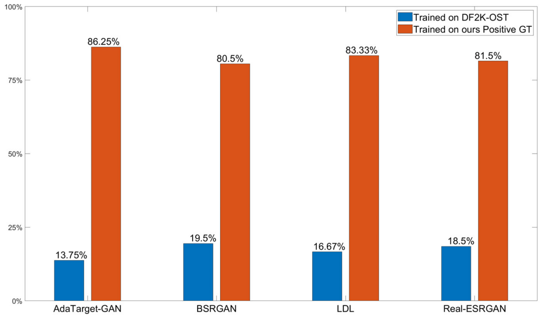


Figure 5. User study results on the Real-ISR models trained on the DF2K-OST dataset (the blue bar) and the positive GTs in our HGGT dataset (the red bar).

Acknowledgement. We thank Dr. Lida LI for providing support in GPU server configuration and the many people participating in data collection and annotation.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, pages 126–135, 2017. 1, 3, 4

[2] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. *NeurIPS*, 32, 2019. 1

[3] 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 *ICCV*, pages 3086–3095, 2019. 1, 2

[4] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Camera lens super-resolution. In *CVPR*, pages 1652–1660, 2019. 1, 2

[5] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, pages 12299–12310, 2021. 1, 2

[6] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *CVPR*, pages 11065–11074, 2019. 2

[7] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE TPAMI*, 44(5):2567–2581, 2020. 6

[8] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, pages 184–199. Springer, 2014. 1, 2

[9] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE TPAMI*, 38(2):295–307, 2015. 1, 2

[10] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, pages 391–407. Springer, 2016. 1

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[12] Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. Blind super-resolution with iterative kernel correction. In *CVPR*, pages 1604–1613, 2019. 1

[13] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Deep back-projectinetworks for single image super-resolution. *IEEE TPAMI*, 43(12):4323–4337, 2020. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[15] Xiangyu He, Zitao Mo, Peisong Wang, Yang Liu, Mingyuan Yang, and Jian Cheng. Ode-inspired network design for single image super-resolution. In *CVPR*, pages 1732–1741, 2019. 2

[16] Zewei He, Yanpeng Cao, Lei Du, Baobei Xu, Jiangxin Yang, Yanlong Cao, Siliang Tang, and Yueting Zhuang. Mrfn: Multi-receptive-field network for fast and accurate single image super-resolution. *IEEE TMM*, 22(4):1042–1054, 2019. 2

[17] Zewei He, Du Chen, Yanpeng Cao, Jiangxin Yang, Yanlong Cao, Xin Li, Siliang Tang, Yueting Zhuang, and Zheming Lu. Single image super-resolution based on progressive fusion of orientation-aware features. *PR*, page 109038, 2022. 2

[18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 1, 3

[19] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017. 1

[20] Yan Huang, Shang Li, Liang Wang, Tieniu Tan, et al. Unfolding the alternating optimization for blind super resolution. *NeurIPS*, 33:5632–5643, 2020. 1

[21] Younghyun Jo, Seoung Wug Oh, Peter Vajda, and Seon Joo Kim. Tackling the ill-posedness of super-resolution through adaptive target generation. In *CVPR*, pages 16236–16245, 2021. 2, 5, 6

[22] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016. 1

[23] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pages 1646–1654, 2016. 1, 2

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[25] Xiangtao Kong, Hengyuan Zhao, Yu Qiao, and Chao Dong. Classsr: A general framework to accelerate super-resolution networks by data characteristic. In *CVPR*, pages 12016–12025, 2021. 2

[26] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, pages 624–632, 2017. 1

[27] 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 *CVPR*, pages 4681–4690, 2017. 1, 2, 6

[28] Juncheng Li, Faming Fang, Kangfu Mei, and Guixu Zhang. Multi-scale residual network for image super-resolution. In *ECCV*, pages 517–532, 2018. 1, 2

[29] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCVW*, pages 1833–1844, 2021. 1, 2, 5

[30] Jingyun Liang, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Mutual affine network for spatially variant kernel estimation in blind image super-resolution. In *ICCV*, pages 4096–4105, 2021. 1

[31] Jie Liang, Hui Zeng, and Lei Zhang. Details or artifacts: A locally discriminative learning approach to realistic image super-resolution. In *CVPR*, pages 5657–5666, 2022. 1, 2, 5, 6

[32] Jie Liang, Hui Zeng, and Lei Zhang. Efficient and degradation-adaptive network for real-world image super-resolution. In *ECCV*, pages 574–591, 2022. 1, 2

[33] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, pages 136–144, 2017. 1

[34] Zhengxiong Luo, Yan Huang, Shang Li, Liang Wang, and Tieniu Tan. Learning the degradation distribution for blind image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6063–6072, 2022. 1, 2

[35] Cheng Ma, Yongming Rao, Yean Cheng, Ce Chen, Jiwen Lu, and Jie Zhou. Structure-preserving super resolution with gradient guidance. In *CVPR*, pages 7769–7778, 2020. 1, 2

[36] Shunta Maeda. Unpaired image super-resolution using pseudo-supervision. In *CVPR*, pages 291–300, 2020. 2

[37] Salma Abdel Magid, Yulun Zhang, Donglai Wei, Won-Dong Jang, Zudi Lin, Yun Fu, and Hanspeter Pfister. Dynamic high-pass filtering and multi-spectral attention for image super-resolution. In *ICCV*, pages 4288–4297, 2021. 1, 2

[38] Yiqun Mei, Yuchen Fan, and Yuqian Zhou. Image super-resolution with non-local sparse attention. In *CVPR*, pages 3517–3526, 2021. 1

[39] Mohammad Saeed Rad, Behzad Bozorgtabar, Urs-Viktor Marti, Max Basler, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Srobb: Targeted perceptual loss for single image super-resolution. In *ICCV*, pages 2710–2719, 2019. 2

[40] Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *ICCV*, pages 4491–4500, 2017. 2, 6

[41] Jae Woong Soh, Gu Yong Park, Junho Jo, and Nam Ik Cho. Natural and realistic single image super-resolution with explicit natural manifold discrimination. In *CVPR*, pages 8122–8131, 2019. 2, 6

[42] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *CVPR*, pages 3147–3155, 2017. 1

[43] 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 *CVPRW*, pages 114–125, 2017. 3

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 1, 3

[45] Longguang Wang, Yingqian Wang, Xiaoyu Dong, Qingyu Xu, Jungang Yang, Wei An, and Yulan Guo. Unsupervised degradation representation learning for blind super-resolution. In *CVPR*, pages 10581–10590, 2021. 1

[46] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *ICCVW*, pages 1905–1914, 2021. 1, 2, 3, 5, 6

[47] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*, pages 606–615, 2018. 2, 3

[48] 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 *ECCVW*, pages 0–0, 2018. 1, 2, 5

[49] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 2, 6

[50] Zongsheng Yue, Qian Zhao, Jianwen Xie, Lei Zhang, Deyu Meng, and Kwan-Yee K Wong. Blind image super-resolution with elaborate degradation modeling on noise and kernel. In *CVPR*, pages 2128–2138, 2022. 1

[51] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *ICCV*, pages 4791–4800, 2021. 1, 2, 3, 5, 6

[52] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *CVPR*, pages 3262–3271, 2018. 1

[53] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 6

[54] Xindong Zhang, Hui Zeng, Shi Guo, and Lei Zhang. Efficient long-range attention network for image super-resolution. In *ECCV*, pages 649–667, 2022. 1, 2, 3

[55] Xindong Zhang, Hui Zeng, and Lei Zhang. Edge-oriented convolution block for real-time super resolution on mobile devices. In *ACM MM*, pages 4034–4043, 2021. 2

[56] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, pages 286–301, 2018. 1, 2, 3

[57] Yongbing Zhang, Siyuan Liu, Chao Dong, Xinfeng Zhang, and Yuan Yuan. Multiple cycle-in-cycle generative adversarial networks for unsupervised image super-resolution. *IEEE TIP*, 29:1101–1112, 2019. 2

[58] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, pages 2472–2481, 2018. 2

[59] Hongyi Zheng, Hongwei Yong, and Lei Zhang. Unfolded deep kernel estimation for blind image super-resolution. In *ECCV*, pages 502–518, 2022. 1