Details or Artifacts: A Locally Discriminative Learning Approach to Realistic Image Super-Resolution

Jie Liang[1]*, **Hui Zeng**[2]* and **Lei Zhang**[1]†
[1]The HongKong Polytechnic University, [2]OPPO Research
{*liang27jie, cshzeng*}*@gmail.com*; *cslzhang@comp.polyu.edu.hk*

Abstract

Single image super-resolution (SISR) with generative adversarial networks (GAN) has recently attracted increasing attention due to its potentials to generate rich details. However, the training of GAN is unstable, and it often introduces many perceptually unpleasant artifacts along with the generated details. In this paper, we demonstrate that it is possible to train a GAN-based SISR model which can stably generate perceptually realistic details while inhibiting visual artifacts. Based on the observation that the local statistics (e.g., residual variance) of artifact areas are often different from the areas of perceptually friendly details, we develop a framework to discriminate between GAN-generated artifacts and realistic details, and consequently generate an artifact map to regularize and stabilize the model training process. Our proposed locally discriminative learning (LDL) method is simple yet effective, which can be easily plugged in off-the-shelf SISR methods and boost their performance. Experiments demonstrate that LDL outperforms the state-of-the-art GAN based SISR methods, achieving not only higher reconstruction accuracy but also superior perceptual quality on both synthetic and real-world datasets. Codes and models are available at https://github.com/csjliang/LDL.

1. Introduction

Single image super-resolution (SISR) [6, 13, 14, 19, 20, 30–33, 38, 40, 42, 45, 47, 48], which aims to reconstruct a high-resolution (HR) image from its low-resolution (LR) observation, is one hot yet challenging research topic in low-level computer vision. It has become prevalent to train deep neural networks (DNNs) for SISR, while many DNN-based SISR models [2, 6, 27, 37, 48] are trained with pixel-wise ℓ_1 and ℓ_2 losses, and/or local window based metrics (such as SSIM [41]). It is well-known that though high PSNR and SSIM indices might be induced by these losses, they can hardly produce rich image details [4, 20].

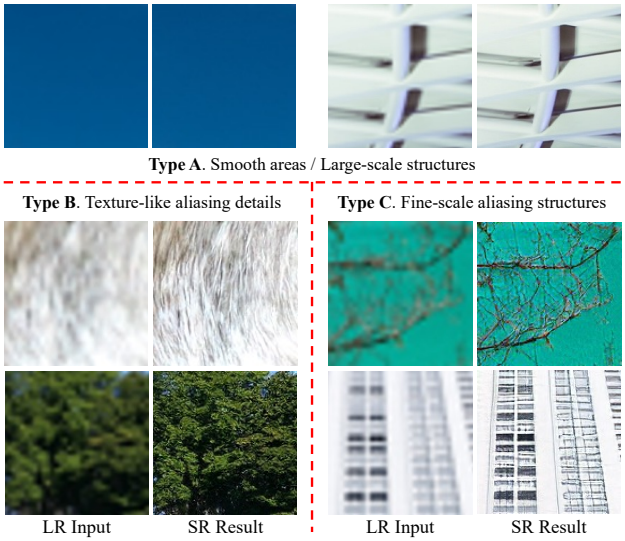


Figure 1. Three representative types of SISR regions generated by ESRGAN [40]. For each example, the left is an LR patch and the right is its GAN-SR result. Type A patches represent regions that are easy to super-resolve, *e.g.*, smooth and large-scale structural areas, where the main structures are preserved in the LR input. In contrast, patches of type B and type C are with fine-scale details, which are hard to be faithfully restored due to the signal aliasing in the LR inputs. The results of texture-like type B patches are perceptually realistic despite the pixel-wise differences to the ground-truth, since the patterns are naturally irregular with weak priors for observers. However, the results of type C patches exhibit perceptually unpleasant visual artifacts since the overshoot pixels and distorted structures are sensitive to human perception.

With the rapid development of generative adversarial networks (GAN) [9, 15], GAN-based SISR (GAN-SR for short) has recently attracted significant attention for its potentials to recover sharp images with rich details [20, 30, 32, 40, 44]. Though great progresses have been achieved, adversarial training is unstable and often introduces unpleasant visual artifacts [20, 44]. As users are mostly expecting

*Equal contribution.
†This work is supported by the Hong Kong RGC RIF grant (R5001-18).

rich and realistic details in SISR results [5, 12, 28], how to inhibit the visual artifacts of GAN-SR without affecting the realistic details becomes a key issue. Unfortunately, details and artifacts are often entangled in high-frequency components of images. As a result, optimizing one of them often harms the other under existing frameworks [4, 20, 25, 40].

In order to address the above mentioned challenges, we investigate in-depth the GAN-SR methods and categorize their results into three typical types of regions, as illustrated in Figure 1. Specifically, type A patches (*e.g.*, flat sky, long edges) are easy to reconstruct since they are smooth or contain only large-scale structures. In contrast, it is difficult to produce high-fidelity SISR results for patches of type B and type C because they have much fine-scale details and suffer from signal aliasing in the degradation process, where most high-frequency components are lost. Fortunately, for texture-like type B patches (*e.g.*, animal fur, tree leaves in distance), the pixels are randomly distributed so that the differences between SR results and ground truth are insensitive to human perception. Therefore, rich details generated by GAN-SR methods can lead to better perceptual quality in these regions. However, patches of type C (*e.g.*, thin twigs, dense windows in the building) contain many fine-scale regular structures or sharp transitions among adjacent pixels. The distorted structures and overshoot pixels generated by GAN-SR methods can be easily perceived by observers as unpleasant artifacts.

Based on the above analysis, we can see that to get perceptually realistic SISR results, the visual artifacts in type C regions should be inhibited, while the realistic details generated in type A and type B regions should be preserved. To achieve this goal, we analyze the local statistic of the three types of GAN-SR regions, and find that the local variance of residuals between SISR results and ground truth HR images can serve as an effective feature to distinguish unpleasant artifacts from realistic details. Accordingly, we construct a pixel-wise map indicating the probability of each pixel being artifacts based on the local and patch-level residual variances. We further refine the discrimination map via a model ensemble strategy to encourage stable and accurate optimization direction toward high-fidelity reconstruction. Based on the refined map, we design a Locally Discriminative Learning (LDL) framework to penalize the artifacts without affecting realistic details.

To sum up, in this paper we first analyze the GAN-SR results and the instability of model training. We then propose to explicitly discriminate visual artifacts from realistic details, and design an LDL framework to regularize the adversarial training. Our method is simple yet effective, and it can be easily plugged into off-the-shelf GAN-SR methods. It provides a novel way to suppress the artifacts in GAN-SR while generating rich realistic details. We conduct extensive experiments on synthetic and real-world SISR tasks, and LDL demonstrates clear improvements against the state-of-the-arts both quantitatively and qualitatively.

2. Related work

Since the pioneer work of SRCNN [6], which firstly introduces a three-layer convolutional neural network (CNN) for SISR, a number of CNN based SISR models have been proposed, which can be roughly divided into signal fidelity-oriented ones [2, 27, 37, 48] and perceptual quality-oriented ones [14, 20, 22, 30, 32, 40], depending on the losses and training strategies employed by them.

Signal fidelity-oriented SISR methods. SISR methods in this category adopt the pixel-wise distance measures (such as ℓ_2 and ℓ_1 losses) and local structural similarity measures (such as SSIM [41]) to optimize the signal fidelity between the SISR outputs and the HR ground-truth. Since SRCNN [6], researchers have made remarkable progresses by stacking more convolution layers [18, 19] and designing more complex building blocks [23, 34] and connections [20, 36, 48]. For instance, benefited from the very deep network, effective residual connections, and channel attentions, RCAN [47] achieves superior performance on reconstruction accuracy (*e.g.*, PSNR). However, due to the ill-posedness of the SISR problem, optimizing the pixel-wise losses tends to find a blurry result that is the average of many possible solutions [4, 30, 32]. The SSIM loss can preserve better the image local structures but it is hard to reproduce fine details.

Perceptual quality-oriented SISR methods. To improve the perceptual quality of SISR images, Johnson *et al*. [14] proposed a perceptual loss by calculating the distance between HR and SISR results in the VGG feature space. To tackle the difficulties of signal fidelity-oriented methods in reproducing image details, most recent works have resorted to using the GAN techniques [9] for their capability to generate desired images by discriminating between image distributions [8, 29, 39]. For example, Ledig *et al*. [20] proposed SRGAN with adversarial training on top of the SRResNet generator. To improve the visual quality, Wang *et al*. [40] proposed the ESRGAN by introducing the Residual-in-Residual Dense Block (RRDB) along with other improvements on adversarial training and perceptual loss. RRDB has been employed as a standard backbone in many state-of-the-art GAN-SR methods [25, 38, 45].

Zhang *et al*. [44] proposed a trainable unfolding network, termed USRGAN, which integrates the merits of traditional model-based methods and CNN-based ones. Ma *et al*. [25] introduced a gradient guidance via an additional branch in the network. By alleviating the structural distortion and inconsistency problem, the proposed SPSR method achieves leading performance among GAN-SR methods on synthetic data. Nonetheless, one key issue of all existing GAN-SR works lies in that they will produce many unpleasant visual

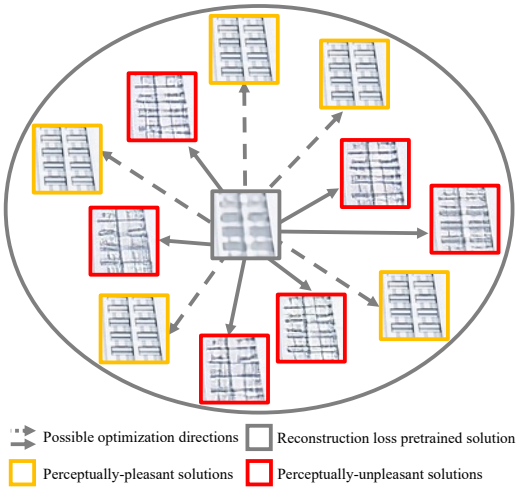


Figure 2. An illustration of possible optimization directions of GAN-SR models. The patch in the center is obtained by a pre-trained SISR model using ℓ_1-loss, while the patches in red and yellow boxes are possible GAN-SR results by adversarial losses.

artifacts due to the instability of adversarial training.

Remarks. As indicated in [4], both signal fidelity- and perceptual quality-oriented SISR methods fall in a perception-distortion trade-off; that is, improving either the perceptual quality or signal fidelity will affect the other under the existing training strategies. Empirical experiences also tell us that inhibiting the artifacts can limit the generation of details. In this paper, we propose to regularize the adversarial training by explicitly discriminating the artifacts from realistic details, which effectively addresses the dilemma. Recent researches, *e.g.*, BSRGAN [45] and RealESRGAN [38], have also recognized the significance of the real-world image SR task. As a plug-and-play module, our method can also be easily extended to such challenging task. The experimental results demonstrated its high generalization performance in generating realistic details while inhibiting artifacts.

3. Methodology

3.1. GAN-SR induced visual artifacts

Most of the existing GAN-SR methods [20, 40] are trained using a weighted combination of three losses:

$$\mathcal{L}_{\text{GAN}} = \lambda_1 \mathcal{L}_{\text{recons}} + \lambda_2 \mathcal{L}_{\text{percep}} + \lambda_3 \mathcal{L}_{\text{adv}}, \qquad (1)$$

where $\mathcal{L}_{\text{recons}}$ indicates the pixel-wise reconstruction loss such as ℓ_1 and ℓ_2 distances, $\mathcal{L}_{\text{percep}}$ is the perceptual loss [14, 20] measuring the feature distance in VGG feature space and $\mathcal{L}_{\text{adv}}$ denotes the adversarial loss [9, 40]. λ_1, λ_2 and λ_3 are balancing parameters, which are usually set to $0.01, 1, 0.005$, respectively, as in ESRGAN [40].

According to the pioneer work of SRGAN [20], using only the $\mathcal{L}_{\text{recons}}$ loss will result in a blurred average of all possible HR images, while the $\mathcal{L}_{\text{adv}}$ loss can push the SISR solution away from the blurred average, generating more details. Unfortunately, GAN-SR models also generate many perceptually-unpleasant artifacts in addition to the details. An intuitive illustration is shown in Figure 2. Since SISR is an ill-posed task, one LR input corresponds to many possible HR counterparts scattering in the high-dimensional image space. Starting from the blurry solution (the center patch in Figure 2) generated by an SISR model pre-trained using only the $\mathcal{L}_{\text{recons}}$ loss, the $\mathcal{L}_{\text{GAN}}$ loss can update it along many possible directions, some yielding perceptually pleasant results (in yellow boxes) and some producing unpleasant ones (in red boxes). This leads to an unstable optimization process that may generate artifacts along with details.

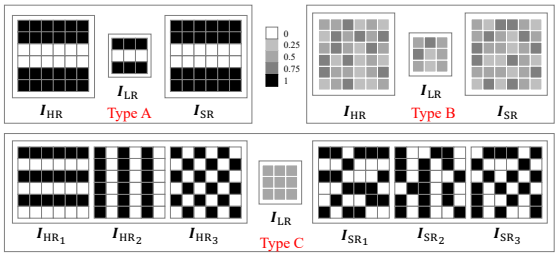


Figure 3. Toy examples of the GAN-SR results on three types of regions. The LR patches are obtained by applying 2×2 average pooling with stride 2 on the HR patches. The large-scale structure in type A patch can be well reproduced with good fidelity and perceptual quality. Though the pixels in texture-like type B patch are not faithfully reconstructed, the perceptual quality of the reconstructed patch is not bad due to the random distribution of pixels in HR patch. However, for those type C patches, visually unpleasant artifacts are perceived in the GAN-SR results since the fine-scale yet regular structures are destroyed.

The above situation can vary among different image regions, as discussed in Figure 1. To better understand how GAN-SR generates visual artifacts in different areas of an image, in Figure 3 we show toy examples of the three types of patches. We see that for type A patch, the large-scale structure is preserved in its LR version and the HR patch can be easily reproduced with good fidelity and perceptual quality. For the texture-like type B patch, though it is not pixel-wise faithfully reconstructed, the perceptual quality of the GAN-SR output is not bad. This is mainly because the pixels in texture-like patches are often randomly distributed in a relatively small range so that human eyes are hard to perceive the pixel-wise difference. In contrast, type C patches have regular and sharp transitions, while the local patterns are lost in the LR patch after degradation. The largely varied and even contradictory HR targets lead to unstable adversarial training, and the irregular and unnatural patterns in the GAN-SR results can be easily perceived by observers as artifacts.

In Figure 4, we further investigate the training stability of GAN-SR methods on different patches, including the

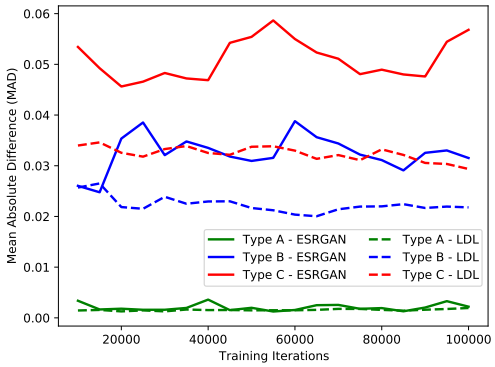


Figure 4. The stability on the training of different patches by ESRGAN [40] and our LDL. The patches of flat sky (type A), animal fur (type B) and thin twigs (type C) in Figure 1 are used here. The mean absolute differences (MAD) of intermediate GAN-SR results between iterations k and k+5000 are plotted.

flat sky (type A), animal fur (type B) and thin twigs (type C) in Figure 1. We calculate the mean absolute difference (MAD) of the intermediate GAN-SR outputs at two different iterations, *i.e.*, MAD=$|\boldsymbol{I}_{\text{SR}}^{(k)} - \boldsymbol{I}_{\text{SR}}^{(k+p)}|$, where $\boldsymbol{I}_{\text{SR}}^{(k)}$ is the GAN-SR result at iteration k, and we set p to 5000. The curves of MAD vs. k for ESRGAN [40] are plotted as solid lines. As can be seen, the training process of type A patch is stable (small value and variation of MAD). Type B shows larger variation, indicating higher uncertainty during optimization. Type C has the largest variation and instability, implying that many possible GAN-SR solutions of type C are available in a large space, as illustrated in Figure 2.

3.2. Discriminating artifacts from realistic details

According to the investigations in Section 3.1, we should inhibit the generation of artifacts in type C patches while preserving the realistic details in type A and B patches. To achieve this challenging goal, we carefully design a pixel-wise map to discriminate artifacts from realistic details, as well as a learning strategy to stabilize the training of GAN-SR models. The whole procedure of map generation is illustrated in Figure 5 using three patches.

Discrimination of artifacts. Suppose that the resolution of a full-color SISR image $\boldsymbol{I}_{\text{SR}}$ is $H \times W \times 3$, our goal is to find a pixel-wise map $\boldsymbol{M} \in \mathbb{R}^{H \times W \times 1}$, where $\boldsymbol{M}(i,j) \in [0,1]$ indicates the probability of $\boldsymbol{I}_{\text{SR}}(i,j)$ being an artifact pixel. Considering that both the artifacts and details belong to high-frequency image components, we first calculate the residual between ground truth image $\boldsymbol{I}_{\text{HR}}$ and SISR result $\boldsymbol{I}_{\text{SR}}$ to extract high-frequency components:

$$\boldsymbol{R} = \boldsymbol{I}_{\text{HR}} - \boldsymbol{I}_{\text{SR}}. \tag{2}$$

As shown in the 3^{rd} column of Figure 5, most pixels in the smooth type A patch have very small residuals. Both type B and type C patches have large residuals, while the distribution of residuals in patch B is much more random. Based on the observation that artifacts usually consist of overshoot pixel values, we propose to calculate the local variance of the residual map $\boldsymbol{R}$ as the primary map to indicate artifact pixels:

$$\boldsymbol{M}(i,j) = var(\boldsymbol{R}(i-\frac{n-1}{2} : i+\frac{n-1}{2}, j-\frac{n-1}{2} : j+\frac{n-1}{2}), \tag{3}$$

where var represents the variance operator and n denotes the local window size. We empirically set $n = 7$.

As shown in the 4^{th} column of Figure 5, the primary map $\boldsymbol{M}$ can effectively detect the artifact pixels in patch C. However, since the local variance is calculated with a very small receptive field, it is unstable to discriminate artifacts from edges and textures. Some pixels in patches A and B will also have large response, causing wrong punishment on the generation of realistic details. To address this issue, we further calculate a stable patch-level variance σ from the whole residual map $\boldsymbol{R}$ as follows:

$$\sigma = (var(\boldsymbol{R}))^{\frac{1}{a}}, \tag{4}$$

where $(\cdot)^{\frac{1}{a}}$ scales the global variance $var(\boldsymbol{R})$ to an appropriate scale. We fix a to 5 throughout our experiments. In general, type A patches have smaller σ values than type B and type C patches, while type C patches have the largest σ values. By using σ to scale the primary map $\boldsymbol{M}$ as $\sigma \cdot \boldsymbol{M}$, a more reliable artifact map can be obtained. As shown in the 5^{th} column of Figure 5, the over-punishment issue on patches A and B is mostly addressed, while the artifacts in patch C are still identified.

Stabilization and refinement. Although the map $\sigma \cdot \boldsymbol{M}$ can discriminate the artifacts in different types of patches, it may still over-penalize the realistic details in patch C, and slightly penalize the generation of high-fidelity details in patches A and B, especially at the early training stages. To alleviate this problem, we further stabilize the training process and refine the artifact map.

Specifically, denote by Ψ the GAN-SR model optimized via gradient decent on-the-fly, we use the exponential moving average (EMA) technique to temporally ensemble a more stable model Ψ_{EMA} from Ψ as:

$$\Psi_{\text{EMA}}^{(k)} = \alpha \cdot \Psi_{\text{EMA}}^{(k-1)} + (1-\alpha) \cdot \Psi^{(k)}, \tag{5}$$

where α is the weighting parameter. Compared to Ψ, Ψ_{EMA} is more reliable to alleviate the generation of random artifacts. As in prior arts of EMA [16, 17], we set $\alpha = 0.999$.

With Ψ_{EMA}, we can further refine the artifact map $\sigma \cdot \boldsymbol{M}$ to alleviate penalty on generation of realistic details during optimization. Denote by $\boldsymbol{I}_{\text{SR}_1} = \Psi(\boldsymbol{I}_{\text{LR}})$ and $\boldsymbol{I}_{\text{SR}_2} = \Psi_{\text{EMA}}(\boldsymbol{I}_{\text{LR}})$ the outputs of two GAN-SR models. Usually, the output of the ensemble model, *i.e.*, $\boldsymbol{I}_{\text{SR}_2}$, has few artifacts, while $\boldsymbol{I}_{\text{SR}_1}$ may contain more details and artifacts simultaneously. We then calculate two residuals map

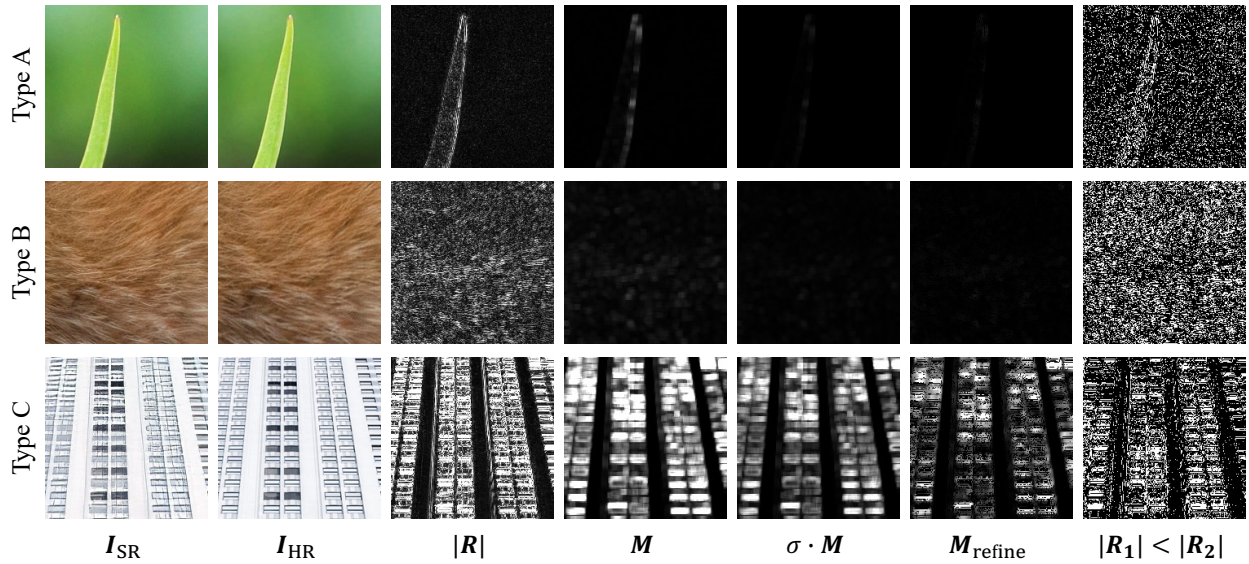


Figure 5. Visualization on the generation process of artifact map. $I_{SR}, I_{HR}, |R|, M, \sigma$ and M_{refine} indicate the SISR output of a GAN-SR method, the ground truth patch, the absolute value of the residual between I_{SR} and I_{HR}, the primary map calculated by Eq. (3), the scaling factor computed by Eq. (4), and the refined map by Eq. (6), respectively. In the 5^{th} column, the σ values for type A, B and C patches are $0.25, 0.39, 0.67$, respectively. The last column shows the locations where $|R_1| < |R_2|$ with white pixels.

$R_1 = I_{HR} - I_{SR_1}$ and $R_2 = I_{HR} - I_{SR_2}$, and refine the artifact map $\sigma \cdot M$ by:

$$M_{refine}(i,j) = \begin{cases} 0, & \text{if } |R_1(i,j)| < |R_2(i,j)|; \\ \sigma \cdot M(i,j), & \text{if } |R_1(i,j)| \geq |R_2(i,j)|. \end{cases} \quad (6)$$

That is, the refined map M_{refine} will only penalize the pixels where $|R_1(i,j)| \geq |R_2(i,j)|$. At locations where the residuals of I_{SR_1} are smaller than I_{SR_2}, the model Ψ is updated towards the correct direction and should not be penalized. The refined map M_{refine} and the location map $|R_1| < |R_2|$ are shown in the last two columns of Figure 5. We see that the locations of fine textures and desirable edges are removed from the refined artifact map so that the penalty can be imposed more precisely on the artifact pixels.

3.3. Loss and learning strategy

Given the refined artifact map M_{refine}, we propose an artifact discrimination loss $\mathcal{L}_{artif}$ as follows:

$$\mathcal{L}_{artif} = \|M_{refine} \cdot (I_{HR} - I_{SR_1})\|_1. \quad (7)$$

The loss $\mathcal{L}_{artif}$ can be easily introduced to the existing GAN-SR models and the final loss function is:

$$\mathcal{L}_{LDL} = \mathcal{L}_{GAN} + \beta \mathcal{L}_{artif}, \quad (8)$$

where $\mathcal{L}_{GAN}$ is defined in Eq. (1) and β is a weighting parameter. We simply fix $\beta = 1$ in all our experiments.

The pipeline of the proposed locally discriminative learning (LDL) method is shown in Figure 6. The input I_{LR} is fed into two models, *i.e.*, Ψ and Ψ_{EMA}, to output I_{SR_1} and I_{SR_2}, respectively. The artifact map M_{refine} is then constructed using the ground-truth image I_{HR}, as well as I_{SR_1} and I_{SR_2}. After that, the loss $\mathcal{L}_{artif}$ is calculated based on I_{HR}, I_{SR_1} and M_{refine}. Finally, the model Ψ is optimized using $\mathcal{L}_{LDL}$, and the parameters of Ψ are temporally ensembled to Ψ_{EMA}. This process is iterated until converge.

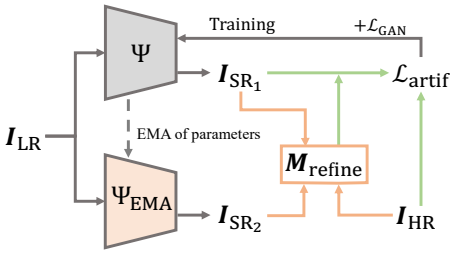


Figure 6. Overall learning pipeline of the proposed LDL method.

With the proposed LDL, we train the same RRDB backbone [40] and plot the MAD curves of intermediate GAN-SR outputs in Figure 4 using dash lines. As can be seen, our LDL method has much better stability than ESRGAN in model learning, especially for type B and type C patches, resulting in much smaller MAD and MAD variations.

4. Experimental results

4.1. Experiment setup

Backbones and compared methods. We validate the effectiveness of the proposed LDL method on top of three representative backbone networks, *i.e.*, SRResNet [20], RRDB [40] and SwinIR [21], resulting in SRResNet+LDL, RRDB+LDL and SwinIR+LDL. SRResNet is a light-weight network, and we compare SRResNet+LDL

Table 1. Quantitative comparison between GAN-SR methods and the proposed LDL. Three groups of comparisons are made based on the employed backbone networks: SRResNet-like backbone for the first 3 columns, RRDB backbone for the middle 5, and SwinIR backbone for the last 2. The best results of each group are highlighted in **bold**. ↑ and ↓ mean that the larger or smaller score is better, respectively.

Metrics	Benchmark	SFTGAN [39]	SRGAN [20]	SRResNet [20]+LDL	ESRGAN [40]	USRGAN [44]	SPSR [25]	RRDB [40]+LDL	RRDB [40]+LDL	SwinIR [21]+$\mathcal{L}_{\text{GAN}}$	SwinIR [21]+LDL
Training Dataset		ImageNet + OST	DIV2K	DIV2K	DF2K + OST	DF2K	DIV2K	DIV2K	DF2K	DF2K	DF2K
LPIPS ↓	Set5	0.0800	**0.0753**	0.0759	0.0758	0.0795	**0.0647**	0.0670	0.0691	0.0656	**0.0655**
	Set14	0.1313	0.1327	**0.1303**	0.1241	0.1347	0.1207	0.1207	**0.1132**	0.1160	**0.1091**
	Manga109	0.0716	0.0707	**0.0673**	0.0649	0.0630	0.0672	0.0553	**0.0544**	0.0542	**0.0469**
	General100	0.0947	0.0964	**0.0898**	0.0879	0.0937	0.0862	**0.0790**	0.0796	0.0796	**0.0740**
	Urban100	0.1343	0.1439	**0.1330**	0.1229	0.1330	0.1184	0.1096	**0.1084**	0.1077	**0.1021**
	DIV2K100	0.1331	0.1257	**0.1172**	0.1154	0.1325	0.1099	0.1011	**0.0999**	0.1038	**0.0944**
DISTS ↓	Set5	0.1085	**0.1003**	0.1010	0.0949	0.1045	0.0921	**0.0917**	0.0919	0.0930	**0.0899**
	Set14	0.1133	0.1067	**0.1016**	0.0951	0.0997	0.0920	0.0935	**0.0866**	0.0930	**0.0869**
	Manga109	0.0646	0.0557	**0.0523**	0.0471	0.0471	0.0463	0.0404	**0.0355**	0.0365	**0.0315**
	General100	0.0992	0.0982	**0.0939**	0.0874	0.0931	0.0884	0.0827	**0.0801**	0.0835	**0.0794**
	Urban100	0.1062	0.1081	**0.0989**	0.0880	0.0975	0.0849	0.0822	**0.0793**	0.0835	**0.0800**
	DIV2K100	0.0736	0.0663	**0.0624**	0.0593	0.0645	0.0546	0.0528	**0.0526**	0.0531	**0.0507**
FID ↓	Set5	39.261	31.507	**27.542**	27.215	37.006	30.904	25.288	**24.803**	35.401	**27.955**
	Set14	60.493	63.945	**52.080**	54.933	55.635	53.867	49.577	**43.454**	48.910	**46.057**
	Manga109	21.464	**11.948**	12.652	11.552	10.658	10.662	**9.855**	10.161	9.703	**8.680**
	General100	36.845	33.868	**32.737**	29.843	32.959	30.159	27.506	**27.211**	27.557	**25.304**
	Urban100	**21.370**	22.162	21.512	20.345	21.555	18.672	17.758	**16.351**	17.555	**16.282**
	DIV2K100	18.183	**13.922**	14.823	13.557	14.031	13.754	12.145	**12.121**	12.736	**12.075**
PSNR ↑	Set5	30.057	29.920	**30.527**	30.438	30.910	30.397	30.985	**31.033**	30.873	**31.028**
	Set14	26.743	26.839	**27.278**	26.594	27.405	26.860	**27.491**	27.228	27.282	**27.526**
	Manga109	28.167	28.110	**28.664**	28.413	28.753	28.561	29.407	**29.620**	29.345	**30.143**
	General100	29.159	29.327	**29.775**	29.425	30.001	29.424	30.232	**30.289**	30.104	**30.441**
	Urban100	24.338	24.410	**24.745**	24.365	24.891	24.804	**25.498**	25.459	25.736	**26.231**
	DIV2K100	28.085	28.165	**28.602**	28.175	28.787	28.182	**28.951**	28.819	28.784	**29.117**
SSIM ↑	Set5	0.8483	0.8478	**0.8570**	0.8523	0.8657	0.8443	**0.8626**	0.8611	**0.8655**	0.8611
	Set14	0.7175	0.7252	**0.7366**	0.7144	0.7486	0.7254	**0.7476**	0.7358	0.7407	**0.7478**
	Manga109	0.8562	0.8632	**0.8702**	0.8595	0.8717	0.8590	**0.8746**	0.8734	0.8796	**0.8880**
	General100	0.8060	0.8074	**0.8164**	0.8095	0.8241	0.8091	0.8277	**0.8280**	0.8305	**0.8347**
	Urban100	0.7235	0.7302	**0.7409**	0.7341	0.7503	0.7474	**0.7673**	0.7661	0.7786	**0.7918**
	DIV2K100	0.7707	0.7745	**0.7855**	0.7759	0.7941	0.7720	**0.7951**	0.7897	0.7911	**0.8011**

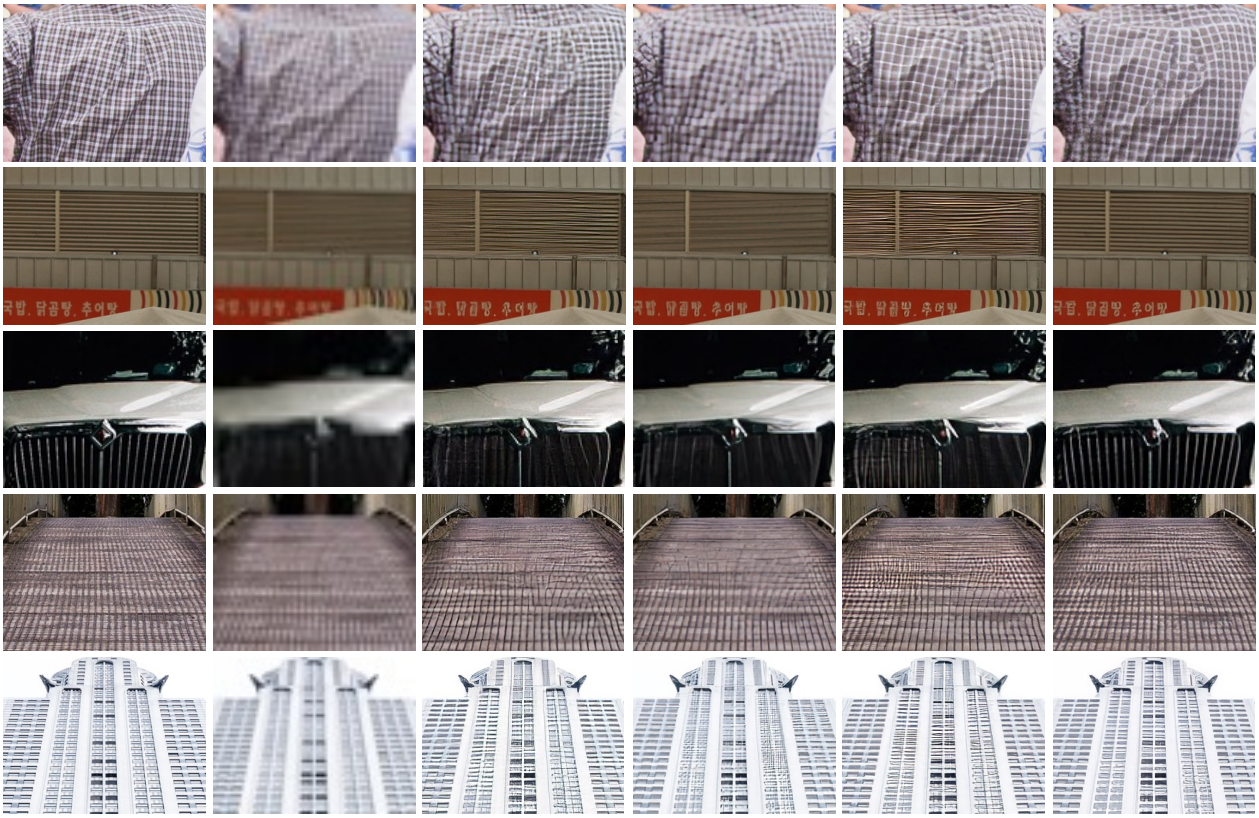
against SRGAN [20] and SFTGAN [39], which have comparable number of parameters. RRDB is widely used in recent GAN-SR methods [25, 40, 44] for its competitive performance. We compare RRDB+LDL against ESRGAN [40], USRGAN [44] and SPSR [25], which all use RRDB as backbone. Very recently, SwinIR has reported excellent SISR performance by using the Swin Transformer architecture [24]. We also train SwinIR with the $\mathcal{L}_{\text{LDL}}$ and $\mathcal{L}_{\text{GAN}}$ (SwinIR+$\mathcal{L}_{\text{GAN}}$) losses, respectively, and compare their performance. We further validate LDL for real-world SISR by applying LDL to RealESRGAN [38], and compare the obtained RealESRGAN+LDL model with both RealESRGAN and BSRGAN [45] models.

Training datasets and settings. Following prior arts [20, 25, 40], we conduct experiments with a scaling factor of $4\times$ on both synthetic (downsampled using MATLAB bicubic kernel) and real-world experiments. We also report $2\times$ GAN-SR results on synthetic data in the supplementary materials. We use the same data augmentation, discriminator and optimizer settings as in ESRGAN [40]. We train our model on either DIV2K [1] (800 images) or DF2K (3450 images) dataset [23, 35], and the resolution of HR patches is 128×128. We implement the experiments on 4 NVIDIA GTX 2080Ti GPUs with PyTorch and the batch size is 16 per GPU. We initialize the generator with a pre-trained fidelity-oriented model, and calculate the perceptual loss as in [38] for both synthetic and real-world settings. The learning rate is $1e^{-4}$ and the number of training iteration is $300k$.

Evaluation benchmarks and metrics. We employ 6 benchmarks for evaluation, including Set5 [3], Set14 [43], Manga109 [26], General100 [7], Urban100 [11] and DIV2K100 [1]. We compare the GAN-SR results in terms of both perceptual quality and reconstruction accuracy. For the former, we employ LPIPS [46], DISTS [5] and FID [10] as metrics. LPIPS and DISTS have been validated effective on evaluating GAN-SR results [12], and FID is widely used to evaluate the image perceptual quality in image generation tasks [16]. For the latter, we compute PSNR and SSIM indices on the Y channel in the YCbCr space.

4.2. Comparison with state-of-the-arts

Quantitative comparison. Table 1 compares quantitatively the state-of-the-art GAN-SR methods and our LDL. We can see that our proposed LDL scheme improves both the perceptual quality (LPIPS, DISTS, FID) and reconstruc-



(a) HR (b) Bicubic (c) ESRGAN (d) USRGAN (e) SPSR (f) RRDB+LDL (DF2K)

Figure 7. Visual comparison (better zoom-in on screen) to state-of-the-art GAN-SR methods that use RRDB [40] as backbone, including ESRGAN [40], USRGAN [44], SPSR [25] and our RRDB+LDL. As can be seen, our method has clear advantages in reconstructing realistic details and inhibiting artifacts. More visual comparisons can be found in the supplementary materials.

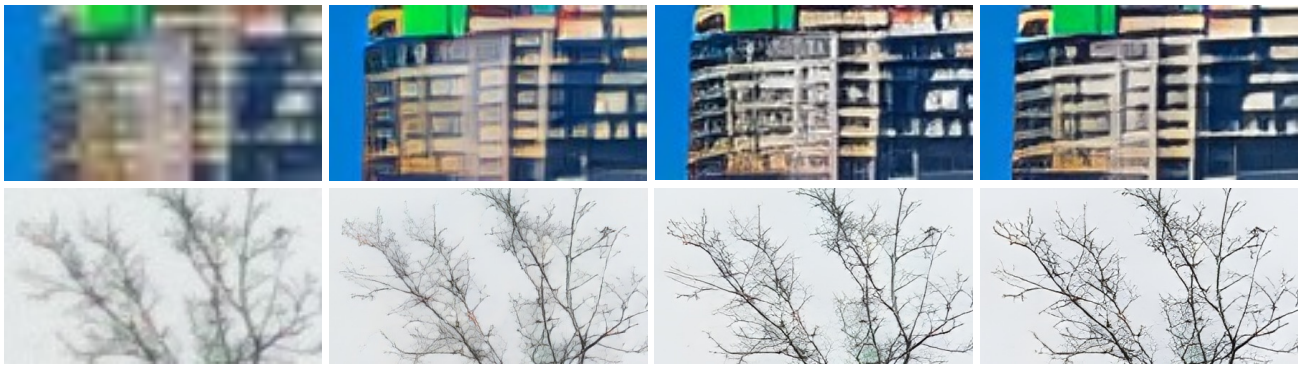
tion accuracy (PSNR, SSIM) on most benchmarks under all the three backbones, *i.e.*, SRResNet, RRDB and SwinIR.

Specifically, for the three light-weight models, SRResNet+LDL outperforms SFTGAN and SRGAN on most benchmarks in terms of those perceptual quality metrics LPIPS, DISTS and FID, and it outperforms SFTGAN and SRGAN on all benchmarks in terms reconstruction accuracy, *e.g.*, PSNR +0.3 $\sim$ 0.5dB and SSIM +0.01 over the second best method, respectively.

For the CNN based backbone RRDB, we train the GAN-SR models on DIV2K and DF2K, respectively, to be consistent with the employed competing models. We can see that among the three competing methods, SPSR performs the best in terms of perceptual quality metrics since it benefits from the additional network branch to restore the gradient map of images. By explicitly discriminating artifacts and regularizing the adversarial training, LDL achieves improvements against SPSR, *e.g.*, LPIPS from 0.1099 to 0.1011 (about 8%) on DIV2K validation set. USRGAN achieves the best reconstruction accuracy among the three competing methods since it integrates learning-based and model-based strategies. Compared to the USRGAN, LDL not only achieves much better reconstruction accuracy on all benchmarks, but also improves the perceptual indexes. This validates that LDL can simultaneously inhibit the visual artifacts and generate more details with high-fidelity.

For the transformer-based backbone SwinIR, we see that SwinIR+$\mathcal{L}_{\text{GAN}}$ outperforms the CNN based methods on most benchmarks in terms of both perceptual quality and reconstruction accuracy, demonstrating the potentials of transformer-based architecture for GAN-SR. As expected, SwinIR+LDL further improves SwinIR+$\mathcal{L}_{\text{GAN}}$ on most benchmarks, demonstrating the generalization capacity of LDL on different network architectures.

Qualitative comparison. Figure 7 presents some visual comparisons among the GAN-SR methods using the RRDB backbone. Similar conclusions to the quantitative comparisons can be drawn. LDL generates much less visual artifacts compared to ESRGAN, USRGAN and SPSR, especially on regions with fine-scale aliasing structures. In addition, by regularizing the adversarial training process, LDL is able to reconstruct more details with high fidelity, such as the areas with regular patterns (*e.g.*, the lines on windows and the grid on bridge). These improvements make LDL a



(a) Bicubic (b) BSRGAN (c) RealESRGAN (d) RealESRGAN+LDL

Figure 8. Visual comparison (better zoom-in on screen) to state-of-the-art real-world SISR methods, including BSRGAN [45] and RealESRGAN [38] . The training setting of RealESRGAN+LDL is the same as RealESRGAN except for the proposed $\mathcal{L}_{\text{LDL}}$ loss. More visual comparisons of different backbones can be found in the supplementary materials.

Table 2. Ablation study on the different components of the proposed LDL method. Results are obtained by RRDB+LDL trained on DF2K and evaluated on DIV2K validation set. ✓ denotes that the corresponding operation is used.

#	$\boldsymbol{M}$	$\sigma \cdot \boldsymbol{M}$	$\boldsymbol{M}_{\text{refine}}$	Ψ_{EMA}	LPIPS	PSNR
1					0.1154	28.175
2	✓				0.1020	28.740
3		✓			0.1006	28.678
4			✓		0.1001	28.761
5			✓	✓	0.0999	28.819

practical GAN-SR solution for image quality enhancement.

4.3. Applications to real-world SISR

To demonstrate the generalization capability of the proposed LDL, we also apply it to the real-world SISR task. Compared to SISR on synthetic LR images, SISR on real-world LR images faces unknown and much more complicated degradation [45]. We introduce the $\mathcal{L}_{\text{artif}}$ loss to the RealESRGAN method [38] and keep all other settings unchanged to train our RealESRGAN+LDL model. Since there is no ground-truth, we show qualitative comparisons with RealESRGAN and BSRGAN in Figure 8. As can be seen in the area of dense windows, RealESRGAN introduces unpleasant artifacts, while BSRGAN produces relatively smooth structures. In contrast, our LDL suppresses the generation of artifacts and encourages sharp details. In the area of twigs, the proposed LDL improves the generation of fine details, benefiting from the explicit and accurate discrimination between artifacts and realistic details.

4.4. Ablation study

We conduct ablation studies to investigate the roles of major components in our LDL method, including the primary artifact map $\boldsymbol{M}$, the globally scaled map $\sigma \cdot \boldsymbol{M}$ in Eq. (4), the refined map $\boldsymbol{M}_{\text{refine}}$ in Eq. (6) and the EMA model Ψ_{EMA}. Results are reported in Table 2. #1 gives the baseline performance when none of the above operations is used. By introducing $\boldsymbol{M}$ in #2, we can observe a clear performance gain in both perceptual quality and reconstruction accuracy. This demonstrates the effectiveness of explicitly discriminating and penalizing the visual artifacts in GAN-SR. The usage of $\sigma \cdot \boldsymbol{M}$ in #3 and $\boldsymbol{M}_{\text{refine}}$ in #4 each further improves the performance. Finally, by using the stable EMA model Ψ_{EMA} during testing in #5, we achieve more performance gain as expected.

4.5. Limitations

Although the proposed LDL is effective in improving both the perceptual quality and reconstruction accuracy of SISR outputs, it still has some limitations in discriminating the visual artifacts in regions suffering from heavy aliasing. Take the last row of Figure 7 for example, there still remain some artifacts around the dense windows in our result. In this paper, we discussed how the artifacts are generated by GAN-SR methods and proposed a simple attempt to tackle this problem, while we believe there exist more effective designs for artifacts discrimination and details generation.

5. Conclusion

In this paper, we analyzed how the visual artifacts were generated in the GAN-based SISR methods, and proposed a locally discriminative learning (LDL) strategy to address this issue. A framework to discriminate visual artifacts from realistic details during the GAN-SR model training process was carefully designed, and an artifact map was generated to explicitly penalize the artifacts without sacrificing the realistic details. The proposed LDL method can be easily plugged into different off-the-shelf GAN-SR models for both synthetic and real-world SISR tasks. Extensive experiments on the widely used datasets demonstrated that LDL outperforms the existing GAN-SR methods both quantitatively and qualitatively.

References

[1] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, 2017. 6

[2] Saeed Anwar and Nick Barnes. Densely residual laplacian super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 2

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 6

[4] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *CVPR*, 2018. 1, 2, 3

[5] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *arXiv preprint arXiv:2004.07728*, 2020. 2, 6

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

[7] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, 2016. 6

[8] Dario Fuoli, Luc Van Gool, and Radu Timofte. Fourier space losses for efficient perceptual image super-resolution. In *ICCV*, 2021. 2

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014. 1, 2, 3

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017. 6

[11] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015. 6

[12] Gu Jinjin, Cai Haoming, Chen Haoyu, Ye Xiaoxing, Jimmy S Ren, and Dong Chao. PIPAL: a large-scale image quality assessment dataset for perceptual image restoration. In *ECCV*, 2020. 2, 6

[13] Younghyun Jo, Seoung Wug Oh, Peter Vajda, and Seon Joo Kim. Tackling the ill-posedness of super-resolution through adaptive target generation. In *CVPR*, 2021. 1

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 1, 2, 3

[15] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. *arXiv preprint arXiv:1807.00734*, 2018. 1

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 4, 6

[17] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 4

[18] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 2

[19] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, 2016. 1, 2

[20] 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*, 2017. 1, 2, 3, 5, 6

[21] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using swin transformer. In *ICCVW*, 2021. 5, 6

[22] Jingyun Liang, Andreas Lugmayr, Kai Zhang, Martin Danelljan, Luc Van Gool, and Radu Timofte. Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling. In *ICCV*, 2021. 2

[23] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017. 2, 6

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 6

[25] Cheng Ma, Yongming Rao, Yean Cheng, Ce Chen, Jiwen Lu, and Jie Zhou. Structure-preserving super resolution with gradient guidance. In *CVPR*, 2020. 2, 6, 7

[26] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, 2017. 6

[27] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In *ECCV*, 2020. 1, 2

[28] Ekta Prashnani, Hong Cai, Yasamin Mostofi, and Pradeep Sen. PieAPP: Perceptual image-error assessment through pairwise preference. In *CVPR*, 2018. 2

[29] 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*, 2019. 2

[30] Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *ICCV*, 2017. 1, 2

[31] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016. 1

[32] 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*, 2019. 1, 2

[33] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Gradient profile prior and its applications in image super-resolution and enhancement. *IEEE Transactions on Image Processing*, 20(6):1529–1542, 2010. 1

[34] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *ICCV*, 2017. 2

[35] 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*, 2017. 6

[36] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *CVPR*, 2017. 2

[37] Longguang Wang, Yingqian Wang, Zaiping Lin, Jungang Yang, Wei An, and Yulan Guo. Learning a single network for scale-arbitrary super-resolution. In *ICCV*, 2021. 1, 2

[38] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. In *ICCVW*, 2021. 1, 2, 3, 6, 8

[39] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*, 2018. 2, 6

[40] 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*, 2018. 1, 2, 3, 4, 5, 6, 7

[41] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 1, 2

[42] Qing Yan, Yi Xu, Xiaokang Yang, and Truong Q Nguyen. Single image super resolution based on gradient profile sharpness. *IEEE Transactions on Image Processing*, 24(10):3187–3202, 2015. 1

[43] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, 2010. 6

[44] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *CVPR*, 2020. 1, 2, 6, 7

[45] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *ICCV*, 2021. 1, 2, 3, 6, 8

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

[47] 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*, 2018. 1, 2

[48] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, 2018. 1, 2