DeSRA: Detect and Delete the Artifacts of GAN-based Real-World Super-Resolution Models

Liangbin Xie [*] [1 2 3] Xintao Wang [*] [3] Xiangyu Chen [*] [1 2 4] Gen Li [5] Ying Shan [3] Jiantao Zhou [1] Chao Dong [2 4]

Abstract

Image super-resolution (SR) with generative adversarial networks (GAN) has achieved great success in restoring realistic details. However, it is notorious that GAN-based SR models will inevitably produce unpleasant and undesirable artifacts, especially in practical scenarios. Previous works typically suppress artifacts with an extra loss penalty in the training phase. They only work for in-distribution artifact types generated during training. When applied in real-world scenarios, we observe that those improved methods still generate obviously annoying artifacts during inference. In this paper, we analyze the cause and characteristics of the GAN artifacts produced in unseen test data without ground-truths. We then develop a novel method, namely, DeSRA, to **De**tect and then "**De**lete" those **SR A**rtifacts in practice. Specifically, we propose to measure a relative local variance distance from MSE-SR results and GAN-SR results, and locate the problematic areas based on the above distance and semantic-aware thresholds. After detecting the artifact regions, we develop a finetune procedure to improve GAN-based SR models with a few samples, so that they can deal with similar types of artifacts in more unseen real data. Equipped with our DeSRA, we can successfully eliminate artifacts from inference and improve the ability of SR models to be applied in real-world scenarios. The code will be available at `https://github.com/TencentARC/DeSRA`.

[*]Equal contribution [1]State Key Laboratory of Internet of Things for Smart City, University of Macau [2]Shenzhen Key Lab of Computer Vision and Pattern Recognition, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences [3]ARC Lab, Tencent PCG [4] Shanghai Artificial Intelligence Laboratory [5]Platform Technologies, Tencent Online Video. Correspondence to: Chao Dong <chao.dong@siat.ac.cn>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

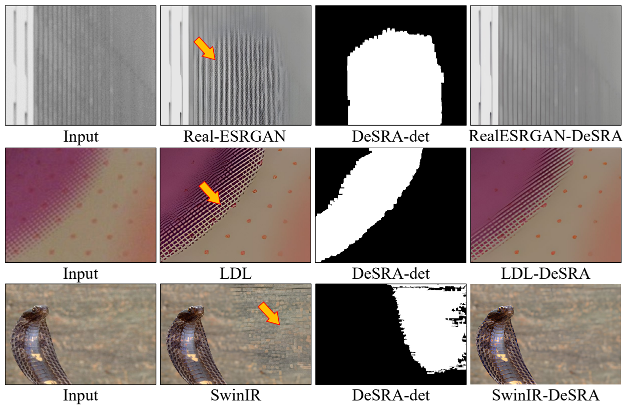


Figure 1. Visualization of GAN-SR artifacts, artifact detection results, and improved GAN-SR results by our proposed DeSRA. Since these GAN-inference artifacts appear on unseen real test data, existing methods that deal with GAN-SR artifacts by improving the training like LDL (Liang et al., 2022b) would still suffer from these artifacts. Our DeSRA can effectively detect the artifact regions and then improve the SR model to eliminate those artifacts and restore visually-pleasant results. **Zoom in for best view**

1. Introduction

Single image super-resolution (SISR) aims to reconstruct high-resolution (HR) images from their low-resolution(LR) observations. Since the pioneering work of SRCNN (Dong et al., 2014), numerous approaches (Lim et al., 2017; Zhang et al., 2018b; Liang et al., 2021) have been developed and made great strides in this field. Among them, GAN-based methods (Ledig et al., 2017; Wang et al., 2018b) have achieved great success in generating realistic SR results with detailed textures. Recently, BSRGAN (Zhang et al., 2021) and Real-ESRGAN (Wang et al., 2021c) extend GAN-based models to real-world applications and obtain promising results, demonstrating their immense potential to restore textures for real-world images. However, it is notorious that GAN-SR methods often generate perceptually unpleasant artifacts, which would seriously affect the user experience. This problem is exacerbated in real-world scenarios, due to the unknown and complex degradation of LR images.

Several works (Ma et al., 2020; Liang et al., 2022b) have

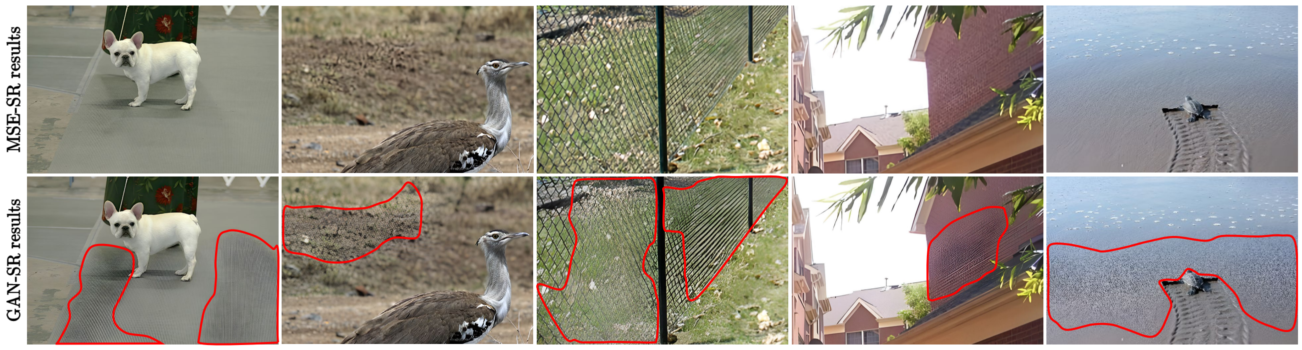


Figure 2. MSE-SR and GAN-SR results of some practical samples. GAN-SR results with artifacts have even worse visual quality than MSE-SR results. The artifacts are complicated with different types and characteristics, and are diverse for different image contents. Regions with red circles are GT of the detection mask.

been proposed to deal with the artifacts generated by GAN-SR models. Typically, LDL (Liang et al., 2022b) proposes to construct a pixel-wise map indicating the probability of each pixel being an artifact by analyzing the type of texture, and then penalizes the artifacts by adding loss during training. Although it indeed improves GAN-SR results, we can still observe obvious visual artifacts when inferencing real-world testing data, as shown in Fig. 1. It is hard to solve these artifacts only by improving the training on existing data pairs, since such artifacts probably do not appear during the training of GAN-SR models.

To better illustrate this problem, we attempt to classify the GAN-SR artifacts according to the different stages they appear. **1)** *GAN-training artifacts* usually arise in the training phase, mainly due to the unstable optimization (Liang et al., 2022b) and the ill-posed property of SR in the in-distribution data. With the presence of ground-truth images, those artifacts could be monitored during training and thus can be mitigated by improving the training, as LDL (Liang et al., 2022b) does. **2)** There is another kind of artifacts that often appears in the real-world unseen data during inference, which we term as *GAN-inference artifacts*. Those artifacts are typically out of training distribution and do not appear in the training phase. Thus, those methods that focus on synthetic images and improve the training procedure (, LDL) cannot solve those artifacts.

Dealing with GAN-inference artifacts is a new and challenging task. There is no ground-truth for real-world testing data with GAN-inference artifacts. Besides, it is hard to simulate these artifacts since they may seldom or even never appear in the training set. In other words, these artifacts are unseen and out of distribution to the models. However, solving this problem is the key to applying GAN-SR models for real-world scenarios, which has great practical value.

There are two steps to resolve the artifacts. The first step is to detect the artifact regions. In actual training of the GAN-SR model, we usually finetune it from the MSE-SR model with GAN training strategy, aiming to add fine details. Since there is no ground-truth of the inference results, we adopt the MSE-based results as the reference, which are easily accessible even for real-world data. We then design a quantitative indicator that calculates the local variance to measure the texture difference between results generated by MSE-based and GAN-based models. After obtaining a pixel-wise distance map, we further introduce semantic-aware adjustment to enlarge the difference in perceptually artifact-sensitive regions (, building, sea) while suppressing the difference in textured regions (, foliage, animal fur). We then filter out detection noises and perform morphological manipulations to generate the final artifact mask.

Based on the detected artifact regions, the second step is to make the pseudo GT and finetune the GAN-SR model. Firstly, we collect a small amount of GAN-based results with artifacts and replace the artifact regions with the MSE-based results according to the binarized detection masks. Then we use the combined results as the pseudo GT to construct training pairs to finetune the model for a very short period of iterations. Experimental results show that our fine-tuning strategy can significantly alleviate GAN-inference artifacts and restore visually-pleasant results on other unseen real-world data.

To summarize, **1)** We make the first attempt to analyze GAN-inference artifacts that usually appear on *unseen test data without ground-truth* during inference. **2)** Based on our analysis, we design a method to effectively detect regions with GAN-inference artifacts. **3)** We further propose a fine-tuning strategy that only requires a small number of artifact images to eliminate the same kinds of artifacts, which bridges the gap of applying SR algorithms to practical scenarios. **4)** Compared to previous work, our method is able to detect unseen artifacts more accurately and alleviate the artifacts produced by the GAN-SR model in real-world test data more effectively.

2. Related Work

MSE-based Super-Resolution. SR methods in this category aim to restore high-fidelity results by minimizing the pixel-wise distance between SR outputs and HR ground-truth like $l1$ and $l2$ distance. Since SRCNN (Dong et al., 2014) successfully applies deep convolution neural networks (CNNs) to the image SR task, numerous deep networks (Dong et al., 2015; 2016; Zhang et al., 2018b; Dai et al., 2019; Niu et al., 2020; Chen et al., 2021; Liang et al., 2021; Li et al., 2021; Chen et al., 2023; Wang et al., 2022) have been proposed to further improve the reconstruction quality. For instance, many methods apply more elaborate convolution module designs, such as residual block (Ledig et al., 2017; Lim et al., 2017) and dense block (Wang et al., 2018b; Zhang et al., 2018c). At the same time, many works have been proposed for the Blind SR task (Zhang et al., 2018a; Gu et al., 2019; Huang et al., 2020; Wang et al., 2021a; Xie et al., 2021b) and video SR task (Huang et al., 2017; Chan et al., 2021; 2022a; Liang et al., 2022a; Shi et al., 2022; Lin et al., 2022). Recently, several Transformer-based networks (Liang et al., 2021; Chen et al., 2023) are proposed and refresh the state-of-the-art performance. However, due to the ill-posedness of the SR problem, optimizing the pixel-wise distance unavoidably results in smooth reconstructions that lack fine details.

GAN-based Super-Resolution. To improve the perceptual quality of SR results, GAN-based methods are proposed to introduce generative adversarial learning for SR task (Ledig et al., 2017; Wang et al., 2018b;a; Zhang et al., 2019; Fuoli et al., 2021; Rad et al., 2019; Ma et al., 2020; Zhang et al., 2020; Mou et al., 2022). SRGAN uses SRResNet generator and perceptual loss (Johnson et al., 2016) to train the network. ESRGAN further improves the visual quality by adopting Residual-in-Residual Dense Block as the backbone to enhance the generator. To extend the GAN-SR model to real-world applications, BSRGAN (Zhang et al., 2021) and Real-ESRGAN (Wang et al., 2021c) design practical degradation models. For real-world video scenarios, RealBasicVSR (Chan et al., 2022b) and FastRealVSR (Xie et al., 2022) also incorporate practical degradation models. Despite the success, GAN-SR models often suffer from severe perceptually-unpleasant artifacts. SPSR (Ma et al., 2020) proposes to alleviate the structural distortion by introducing a gradient guidance branch. LDL (Liang et al., 2022b) constructs a pixel-wise map that represents the probability of each pixel being artifact and penalizes the artifacts by introducing extra loss during training. Nonetheless, these methods would still result in artifacts in actual inference.

3. Methodology

Preliminary: GAN-SR models aim to learn a generative network G parameterized by θ_{GAN} that estimates a high-resolution image $\hat{y}$ for a given low-resolution x image as:

$$\hat{y} = G(x; \theta_{GAN}). \tag{1}$$

To optimize the network parameters, a weighted combination of three sorts of losses is adopted in most GAN-SR methods (Ledig et al., 2017; Wang et al., 2018b; 2021c) as the loss function:

$$\mathcal{L}_{GAN} = \lambda_1 \mathcal{L}_{recons} + \lambda_2 \mathcal{L}_{percep} + \lambda_3 \mathcal{L}_{adv}, \tag{2}$$

where $\mathcal{L}_{recons}$ represents the pixel-wise reconstruction loss such as l_1 or l_2 distance, $\mathcal{L}_{percep}$ is the perceptual loss (Johnson et al., 2016) calculating the feature distance and $\mathcal{L}_{adv}$ denotes the adversarial loss (Ledig et al., 2017). Due to the instability GAN training, in the training of a GAN-SR model, a MSE-SR model is generally trained first only using $\mathcal{L}_{recons}$ to obtain θ_{MSE}, and then the GAN-SR model is finetuned based on the pretrained θ_{MSE} using $\mathcal{L}_{GAN}$ to get the final θ_{GAN}.

3.1. Analyze GAN Artifacts Introduced in Inference

Unlike MSE-based optimizations that naturally tend to produce over-smooth reconstruction results, GAN-based models can generate fine details benefiting from adversarial training. However, GAN-SR models often introduce severe perceptually-unpleasant artifacts that seriously affect the visual quality of restored images, especially in real-world scenarios. In some cases, the GAN-SR artifacts would make the results even worse than those generated by the MSE-based model, as shown in Fig. 2. Besides, these artifacts are complicated, with many types and characteristics, and are diverse for different image content.

Essentially, methods for dealing with GAN-SR artifacts are all aimed at improving the results obtained in the inference stage. Nevertheless, the types of artifacts that can be addressed are limited for existing methods, since they deal with the artifacts only by *improving the training process*. For instance, LDL (Liang et al., 2022b) processes the GAN-SR artifacts by adding penalty loss to problematic regions and improving the learning strategy. It works for artifact types generated during the **training phase**, which exist in the in-distribution data of the training set. We name those artifacts as **GAN-training artifacts**. However, some cases of artifacts generated during the **inference phase** are out-of-distribution, namely, **GAN-inference artifacts**. They usually appear in unseen data without reference. Dealing with GAN-training artifacts would lead to better recovery of training data, but the capability of the model to process out-of-distribution data can only rely on its limited generalization ability. For real-world applications, how to solve more general GAN-inference artifacts is much more important. These artifacts are hard to synthesize during training, and thus can not be resolved by only improving the training.

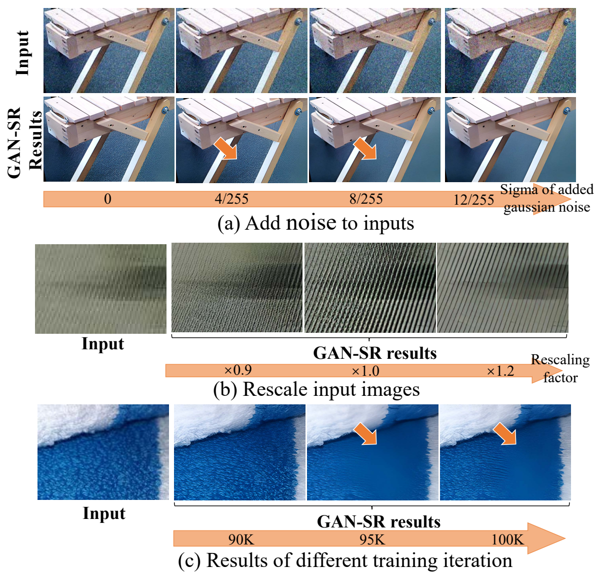


Figure 3. Observations about GAN-inference artifacts. a) Adding imperceptible Gaussian noise or b) slightly rescaling the image can alleviate the artifacts. c) Models of different iterations result in artifacts with different severity. **Zoom in for best view**

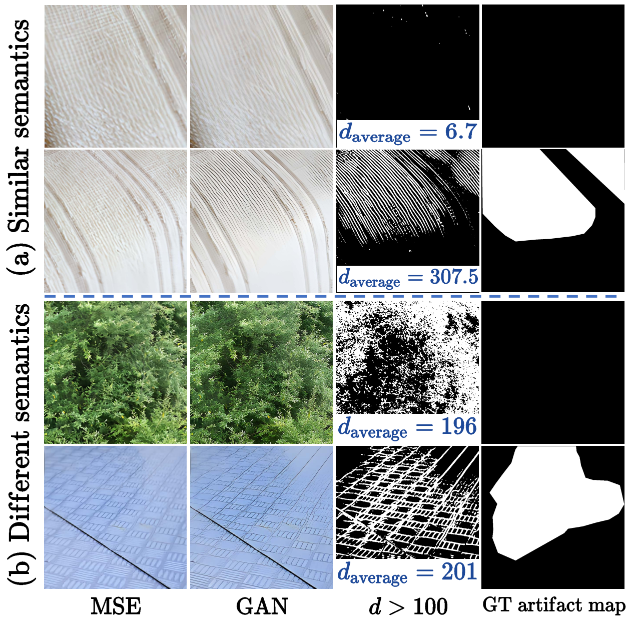

(a) Similar semantics
(b) Different semantics
$d_{\text{average}} = 6.7$
$d_{\text{average}} = 307.5$
$d_{\text{average}} = 196$
$d_{\text{average}} = 201$
MSE
GAN
$d > 100$
GT artifact map

Figure 4. (a) For patches with similar semantics, too large texture difference d from MSE-SR results usually indicates GAN artifacts. (b) Patches with similar d have different visual quality due to their different underlying semantics. Tree regions do not have artifacts while building regions have. Thus, it is hard to measure texture differences with absolute distance. **Zoom in for best view**

In this work, we focus on processing the GAN-inference artifacts, as those artifacts have a largely negative impact on real-world applications, and solving them has great practical value. Due to the complexity and diversity of these kinds of artifacts, it is challenging to address all of them at once. We, therefore, deal with GAN-inference artifacts with the following two characteristics. **1)** The artifacts do not appear in the pretrained MSE-SR model (*i.e.*, the generator G with parameters θ_{MSE}). **2)** The artifacts are obvious and have a large area, which can be observed at the first glance. Some practical examples containing such artifacts are shown in Fig. 2. For the former characteristic, we want to ensure that the artifacts are caused by GANs while the corresponding MSE-SR results are good references for test data to distinguish the artifacts. For the latter feature, we want to address those artifacts that have a great impact on visual quality.

Before introducing the methods for addressing the artifacts, we first give a glimpse of the causes of GAN-inference artifacts. We found that manipulations that would slightly change the degradations, such as adding imperceptible Gaussian noise or rescaling the image, could eliminate the artifacts. As shown in Fig. 3, by modulating the adding noise from $\sigma = 0$ to $\sigma = 12/255$, the artifacts are alleviated gradually. A similar phenomenon appears when we rescale the input by setting the upscaling factor from $\times 0.9$ to $\times 1.2$. We conjecture that these operations can change the distribution of the original image, and may make the degradation of the real image close to the simulated degradations. Besides, models of different training iterations also result in artifacts with different severity, as shown in Fig. 3. It reflects that the unstable training of GAN is also the cause of these artifacts.

3.2. Automatically Detect GAN-inference Artifacts

At first, we want to automatically detect the regions with obvious artifacts according to some quantitative values in the inference phase before processing these artifacts. *Due to the lack of ground-truth images, we choose the MSE-SR results as the reference to evaluate the artifacts generated by the GAN-SR model.* Its rationale lies in that the presentation of GAN artifacts is usually caused by too many unwanted high-frequency 'details'. In other words, we introduce GAN training to generate fine details, but we do not want the generated content by GAN to deviate too much from MSE-SR results. Note that MSE-SR results are easy to access even for unseen test data, as we usually finetune the MSE-SR models to obtain GAN-SR models.

Relative difference of local variance between MSE-SR and GAN-SR patches. Based on the above analysis, we propose to design a quantitative indicator to measure the difference between patches from MSE-SR and GAN-SR results as a basis for judging the artifacts. We adopt the

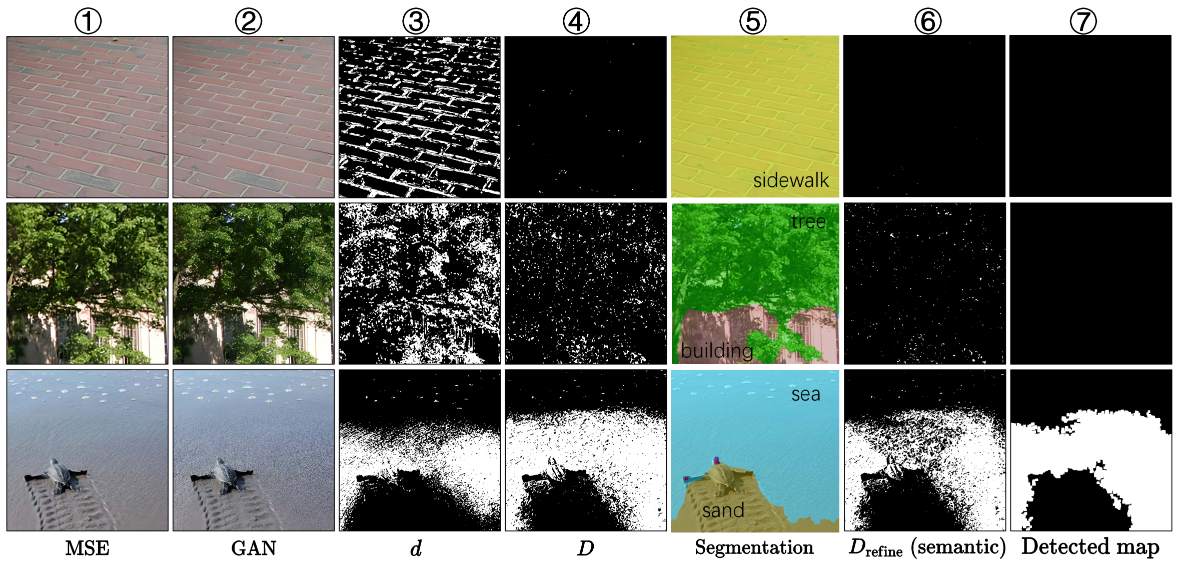


Figure 5. Visualization on the artifact detection pipeline. 172 MSE-SR results. 173 GAN-SR results. 174 Results of directly applying texture difference d. 175 Results of D can roughly indicate artifacts but with some noises. 176 Segmentation map. 177 Semantic-adjusted D with suppressed detection noise. 178 Final detection map. **Zoom in for best view**

standard deviation of pixel intensities within a local region P to indicate the complexity of local texture as:

$$\sigma(i,j) = sd(P(i-\frac{n-1}{2}:i+\frac{n-1}{2}, j-\frac{n-1}{2}:j+\frac{n-1}{2})), \tag{3}$$

where $\sigma(i,j)$ indicates the local standard deviation at (i,j), $sd(\cdot)$ represents the standard deviation operator, and n denotes the local window size and is set to 11. Then we calculate the difference between standard deviations of two patches to measure the texture difference d as:

$$d(x,y) = (\sigma_x - \sigma_y)^2. \tag{4}$$

In our case, x refers to GAN-SR patches, while y denotes MSE-SR patches. As shown in Fig. 4 (a), for patches with similar semantics, too large texture difference d from MSE-SR results usually indicates GAN artifacts. However, d measures the **absolute** difference between patches, which is also related to the texture complexity itself. As depicted in Fig. 4 (b), patches with similar d have different visual quality due to their different underlying semantics. Tree regions do not have artifacts while building regions have. Thus, we want the texture difference indicator to be a **relative** value independent of their original texture variation (, the scale of σ), so we further divide d by the product of σ_x and σ_y as:

$$d'(x,y) = \frac{(\sigma_x - \sigma_y)^2}{2\sigma_x\sigma_y}. \tag{5}$$

To facilitate subsequent operations for the distance map, we hope to normalize the distance d' in the range of $[0,1]$. Inspired by SSIM (Wang et al., 2004), we adopt a similar transformation:

$$d''(x,y) = \frac{1}{1+\frac{(\sigma_x-\sigma_y)^2}{2\sigma_x\sigma_y}} = \frac{2\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2}. \tag{6}$$

A constant C is introduced to stabilize the division with a weak denominator. The final quantitative indicator can be written as:

$$D = \frac{2\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2+C}. \tag{7}$$

We derive this formula step by step according to our actual needs in artifact detection, and each step has its practical meaning in our GAN-inference artifact detection. As shown in 3rd and 4th column of Fig. 5, we can observe that the generated map based on d covers most of the regions with high-frequency difference between MSE-SR and GAN-SR results, but cannot distinguish these artifacts. The relative and normalized texture difference D is successfully used to produce the artifacts map.

Semantic-aware adjustment. After obtaining the distance map, we can exploit it to determine the regions that need to be addressed. However, it is not enough to only use the difference in texture complexity as a basis for judgment, because the perceptual tolerance rate of different semantic regions is different. For example, fine details in areas with complicated textures are difficult to perceive as artifacts like foliage, hair, and *etc*, while large pixel-wise differences in areas with smooth or regular textures, such as sea, sky, and buildings, are sensitive to human perception and easy to be seen as artifacts, as shown in 1st and 2nd column of

Fig. 5. Hence, it is required to adjust the artifact map D based on the underlying semantics. We choose the SegFormer (Xie et al., 2021a) as the segmentation model to distinguish different regions. Specifically, the SegFormer is trained on ADE20K, which covers most semantic concepts of the world. To determine the reasonable adjustment weight for each class, we calculate pixel-wise D values in each class of all images in the training set. For each class, we sort all the D values in descending order and set the D value in the 85$\%$ percentile as the adjustment weight:

$$A_k = P_{85}(D_k), k \in \{1, 2, ..., K\}, \tag{8}$$

where A_k is the adjustment weight for the k^{th} class, D_k is the D value of all pixels identified as the k^{th} class, and P_{85} is the 85^{th} percentile operation. The value of K is 150. For each image, the refined detected map based on segmentation M is computed as:

$$M(i,j) = \begin{cases} 0, & D(i,j)/A_k \geq threshold; \\ 1, & D(i,j)/A_k < threshold. \end{cases} \tag{9}$$

where $D(i,j)$ is the D value of pixel (i,j) and $threshold$ is a hyper-parameter to control whether the current pixel is artifact or not. We empirically set the *threshold* to 0.7.

We additionally perform morphological manipulations to obtain the final detected map, as shown in the 6^th column of Fig. 5. Concretely, we first perform erosion using a 5×5 all-ones matrix. Then we implement dilation using the matrix to join disparate regions. Next, we fill the hole in the map by using a 3×3 all-ones matrix. Finally, we filter out discrete small regions as the detection noise.

3.3. Improve GAN-SR Models with Fine-tuning

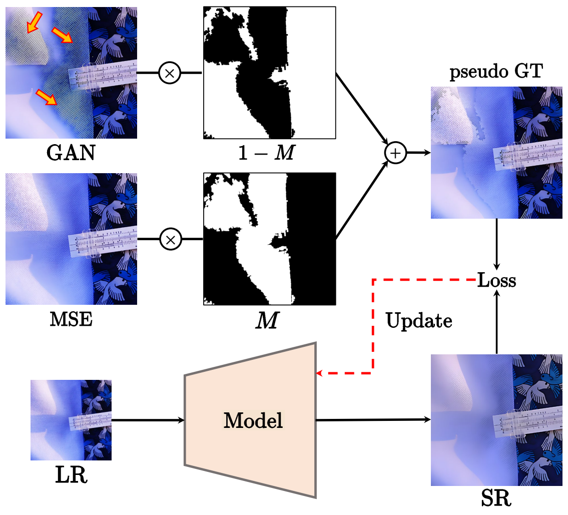


Figure 6. The procedure of our method to finetune the GAN-SR model. **Zoom in for best view**

The detection of GAN-inference artifacts itself is of great practical value. We hope to further improve the GAN-SR model based on the detection results. Note that we aim to solve the GAN-inference artifacts for unseen real data, so there is no ground-truth for the inference results with artifacts. In practice, "weak restoration without artifacts is even better than strong restoration with artifacts". Thus, we exploit the MSE-SR results as the restoration reference.

As illustrated in Fig. 6, we use MSE-SR results to replace the regions where artifacts were detected in GAN-SR results. The merged images serve as the pseudo GT. This process is formulated as:

$$\widetilde{y} = M \cdot y_{MSE} + (1 - M) \cdot y_{GAN}, \tag{10}$$

where $\widetilde{y}$ indicates the generated pseudo GT, y_{MSE} and y_{GAN} are MSE-SR and GAN-SR results, $(\cdot)$ represents the element-wise product, and M is the detected artifact map. We then use a small amount of data to generate the data pairs $(x, \widetilde{y})$ from real data to finetune the model, where x represents the LR data. We only need to finetune the model for a few iterations (about $1K$ iterations are enough in our experiments) and the updated model would produce perceptually-pleasant results without obvious artifacts. Moreover, it does not influence other fine details in regions without artifacts. It can effectively suppress similar kinds of artifacts in more real testing data. The working mechanism behind this approach is that the finetuning process narrows the gap between the distribution of synthetic data and real data to alleviate the GAN-inference artifacts.

4. Experiments

4.1. Experiment Setup

We exploit two state-of-the-art GAN-SR models, Real-ESRGAN (Wang et al., 2021c) and LDL (Liang et al., 2022b), to validate the effectiveness of our method. We use the officially released model for each method to detect the GAN-inference artifacts. For finetuning, the training HR patch size is set to 256. The models are trained with 4 NVIDIA A100 GPUs with a total batch size of 48. We finetune the model only for 1000 iterations and the learning rate is 1e-4.

Dataset. Although several real-world super-resolution datasets (Cai et al., 2019; Wang et al., 2021b) are proposed, they assume camera-specific degradations and is far from practical scenarios. Therefore, we construct a GAN-SR artifacts dataset. Considering the diversity of both image content and degradations, we use the validation set of ImageNet-1K (Deng et al., 2009) as the real-world LR data. Then we choose 200 representative images with GAN-inference artifacts for each method to construct this GAN-SR artifact dataset. Since there is no ground-truth map for artifact regions to evaluate the algorithm, we manually label the artifact area using labelme (Wada). This is the first dataset constructed for GAN-inference artifact detection. For the finetuning process, we further divide the dataset by using 50 pairs for training and 150 pairs for validation.

Evaluation. Due to the lack of ground-truth for real-world LR data, the classic metrics such as PSNR, SSIM cannot be adopted. We also test NIQE (Mittal et al., 2012) and MANIQA (Yang et al., 2022), and observe that these two metrics do not always match perceptual visual quality (Lugmayr et al., 2020) (see Section A.8). Thus, we consider three metrics to evaluate the detection results, including 1) **Intersection over Union (IoU)** of the detected artifact area and the ground-truth artifact area, 2) **Precision** of the detection results and 3) **Recall** of the detection results.

When using A and B to represent the detected artifact area and the ground-truth artifact area for a specific region z, IoU is given by:

$$\text{IoU} = \frac{A \cap B}{A \cup B}. \tag{11}$$

We can calculate IoU for each image, and we use the average IoU on the validation set to evaluate the detection algorithm. A higher IoU means better detection accuracy.

We then define the set of regions with detected artifacts as S and the set of correct samples T is defined as:

$$\text{T} = \{z \in S \mid \frac{A \cap B}{A} > p\}. \tag{12}$$

The metric Precision $= N_T/N_S$ indicates the number of correctly detected regions (N_T) out of the total number of detected regions (N_S). We define the set of the ground-truth regions as G, and the set of detected GT artifact regions R is computed by:

$$\text{R} = \{z \in G \mid \frac{A \cap B}{B} > p\}. \tag{13}$$

The metric Recall $= N_R/N_G$ represents the number of detected GT artifact regions (N_R) out of the total number of GT artifact regions (N_G). p is a threshold and we empirically set it to 0.5.

4.2. Artifact Detection Results

We conduct experiments based on Real-ESRGAN (Wang et al., 2021c) and LDL (Liang et al., 2022b) to validate GAN-inference artifact detection results. We compare our DeSRA-det described in Sec. 3.2 with detection based on NIQE (Mittal et al., 2012), PAL4Inpainting (Zhang et al., 2022), and the modified detection protocol in LDL (Mittal et al., 2012).

Since there is no reference image for the unseen data in the inference phase, we choose the non-reference index NIQE (Mittal et al., 2012) to detect the artifacts for comparing the detection scheme without using MSE-SR results. A similar sliding window mechanism is adopted to compute the pixel-wise map for measuring the local texture and we select the best-performing threshold for filtering the noise to obtain the final detected map. PAL4Inpainting (Zhang et al., 2022) is a newly proposed perceptual artifacts localization method originally for inpainting. We also include it for completeness. As the artifact detection scheme in LDL (Mittal et al., 2012) is designed for GAN-training artifacts with ground-truth images on synthetic data, it cannot be directly applied to solve GAN-inference artifacts without GT. Thus, we use MSE-SR results to replace GT and set a group of threshold $\{0.001, 0.005, 0.01\}$ for the LDL scheme.

Tab. 1 shows the artifact detection results based on Real-ESRGAN. Our method obtains the best IoU and Precision that far outperforms other schemes. Note that LDL with threshold=0.001 obtains the highest Recall. It is because this scheme treats most areas as artifacts, and thus such detection results are almost meaningless. Similar conclusions can be drawn from Tab. 2 for artifact detection results based on LDL. The visual comparison is presented in Fig. 7. The detection results obtained by our approach have significantly higher accuracy than other schemes.

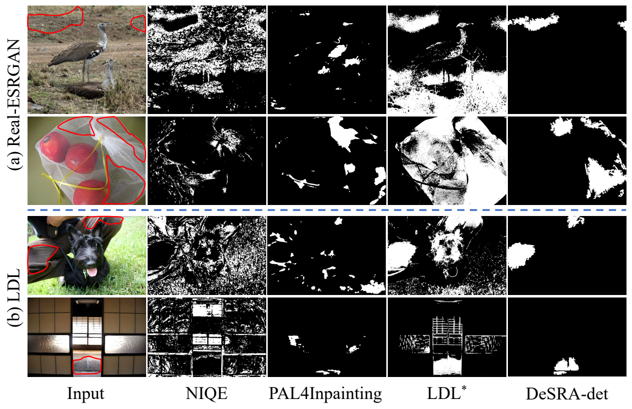


Figure 7. Visual comparison of different methods on artifact detection results. Regions with red circles are GT of detection mask.**Zoom in for best view**

Table 1. Artifact detection results based on Real-ESRGAN (Wang et al., 2021c). LDL* represents the modified detection method in LDL (Liang et al., 2022b).

Method	IoU (↑)	Precision	Recall
NIQE	2.9	0.0494	0.1054
PAL4Inpainting	8.4	0.0855	0.0992
LDL*(threshold=0.01)	29.9	0.3504	0.3485
LDL*(threshold=0.005)	36.2	0.2618	0.5442
LDL*(threshold=0.001)	35.3	0.1410	**0.8391**
DeSRA-det (ours)	**51.1**	**0.7055**	0.6081

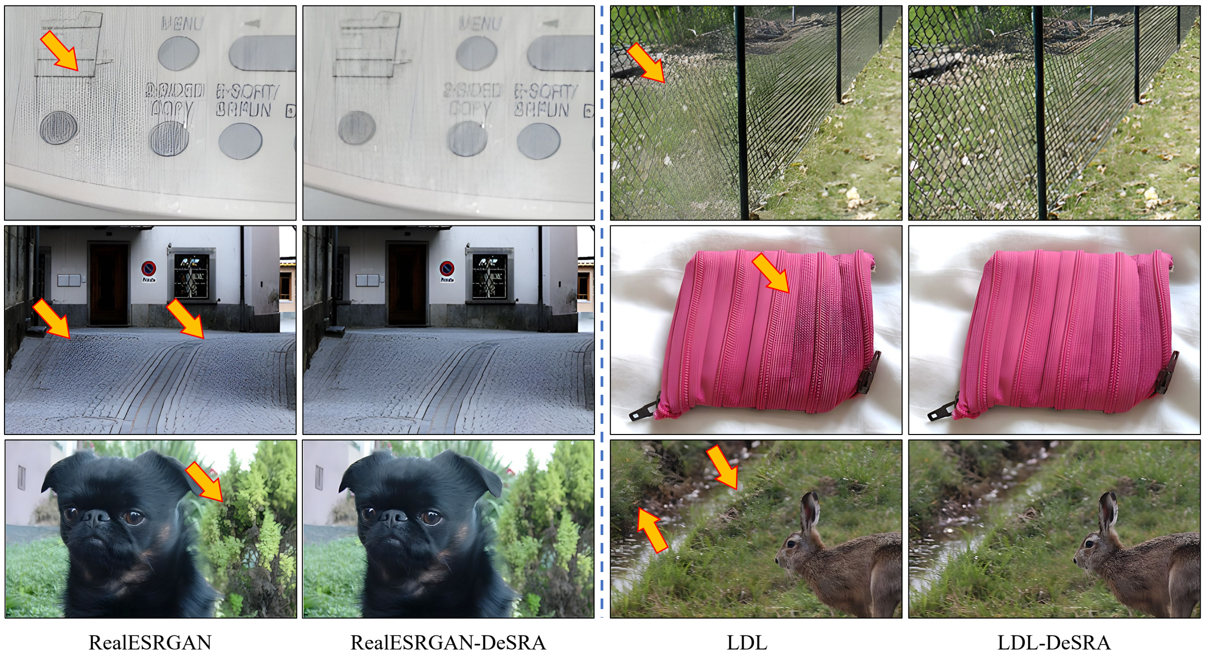


Figure 8. Visual comparison of results generated from original GAN-SR models and the improved GAN-SR models by using our DeSRA. Artifacts are obviously alleviated for results produced by the improved GAN-SR models. **Zoom in for best view**

Table 2. Artifact detection results based on LDL (Liang et al., 2022b). LDL* represents the modified detection method in LDL (Liang et al., 2022b).

Method	IoU (↑)	Precision	Recall
NIQE	2.6	0.0236	0.1770
PAL4Inpainting	8.8	0.0699	0.1337
LDL*(threshold=0.01)	32.7	0.3070	0.4110
LDL*(threshold=0.005)	36.7	0.2100	0.5770
LDL*(threshold=0.001)	31.1	0.1003	**0.8659**
DeSRA-det(ours)	**44.5**	**0.6087**	0.5335

Table 3. Artifact detection results of GAN-SR models with and without using DeSRA finetuning.

Method	IoU (↓)	Removal rate	Addition rate
Real-ESRGAN	51.1	-	-
Real-ESRGAN-DeSRA	**12.9**	75.43%	0%
LDL	44.5	-	-
LDL-DeSRA	**13.9**	74.97%	0%

4.3. Improved GAN-SR Results

We finetune the model to alleviate the GAN-inference artifacts based on the detected artifacts map, as described in Sec. 3.3. Note that this process has a very small training cost (*i.e.*, 50 training pairs with 1000 iterations). We compare the artifact detection results before and after using our DeSRA finetuning strategy to verify the effectiveness of improving the GAN-SR model to alleviate GAN-inference artifacts. The condition for judging the removal of artifacts is $A \cap B = 0$, and the condition for judging the introduction of new artifacts is $A \cup B > B$. As depicted in Tab. 3, after the application of our DeSRA, IoU decreases from 51.1 to 12.9 on Real-ESRGAN and from 44.5 to 13.9 on LDL, illustrating that the detected area of artifacts is greatly reduced. The removal rate is 75.43% and 74.97%, showing that three-quarters of the artifacts on unseen test data can be completely removed after finetuning. Besides, our method does not introduce new additional artifacts, as the addition rate is 0. We provide the visual comparison between results with and without using our method to improve GAN-SR models, as shown in Fig. 8. Results generated by the improved GAN-SR models have greatly better visual quality without obvious GAN-SR artifacts compared to the original inference results. All these experimental results demonstrate the effectiveness of our method for alleviating the artifacts and improving the GAN-SR model.

4.4. User Study

To further verify the effectiveness of our DeSRA finetuning strategy, we perform two user studies. The first is the comparison of the results generated by the original GAN-

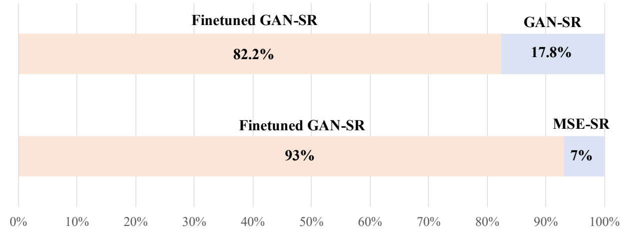


Figure 9. The results of user studies, comparing the results generated by the finetuned GAN-SR models with GAN-SR models and MSE-SR models. **Zoom in for best view**

SR models and the finetuned GAN-SR models. For this experiment, the focus of comparison is on whether there are obvious artifacts. We produce a total of 20 sets of images, each containing the output results of the GAN-SR model and finetuned GAN-SR model. These images are randomly shuffled. A total of 15 people participate in the user study and select the image they think has fewer artifacts for each set. The final statistical results are shown in Fig. 9. 82.23% of participants think that the results generated by fine-tuned GAN-SR models have fewer artifacts. It can be seen that our method largely removes the artifacts generated by the original model.

The second is the comparison of the results of the finetuned GAN-SR models and the original MSE-SR models. This experiment is conducted to compare whether the results generated by the model have more details. We produce a total of 20 sets of images, each containing the output results of the MSE-SR model and finetuned GAN-SR model. These images are randomly shuffled. A total of 15 people participate in the user study and select the image they think has more details for each set. The final statistical results are shown in Fig. 9. 93% of participants think that the results generated by fine-tuned GAN-SR models have more details. It can be seen that the finetuned GAN-SR model still generates more detailed results than the MSE-SR model.

4.5. Ablation Study

We first conduct the ablation study on three key designs of our artifact detection method, including relative difference (RD) (*i.e.,* from d to d'), normalization (*i.e.*, from d' to D) and semantic-aware threshold. As shown in Tab. 4, without using the relative difference suffers the lowest Precision and the full Recall. It is because the detection based on absolute difference would treat most areas as artifacts. The detection scheme without normalization also results in low IoU, Precision, and Recall, since the thresholds for each sample probably have a different scale. Using the semantic-aware threshold can improve the artifact detection results, because the sensitivity of human perception to different semantics is different. All these results demonstrate the necessity of the three designs in our artifact detection method.

Table 4. Ablation study on the key designs for artifact detection.

RD	Normalization	Semantics	IoU	Precision	Recall
	✓	✓	16.1	0.0245	1.0000
✓		✓	3.6	0.2247	0.2869
✓	✓		46.2	0.6627	0.5496
✓	✓	✓	51.1	0.7055	0.6081

Table 5. Influence of the threshold on the detection results.

Threshold	IoU	Precision	Recall	Precision×Recall
0.9	54.3	0.3255	0.9223	0.3002
0.8	56.6	0.5158	0.8123	0.4190
0.7	51.1	0.7055	0.6081	**0.4290**
0.6	38.4	0.8343	0.3351	0.2796

We also conduct an ablation study for the threshold to explore its impact on artifact detection results. The threshold is used to control whether the pixel is the artifact or not for generating the detected map, as described in Equ. 14. Usually, a precision-recall curve shows the trade-off between precision and recall for different thresholds, and a high area under the curve represents both high recall and high precision. For simplicity, we directly use "Precision×Recall" to measure the performance of detection results to select the best threshold. As depicted in Tab. 5, the highest Precision×Recall is obtained when the threshold is set to 0.7. Thus, we select 0.7 as the default setting in our methods.

5. Conclusion

In this work, we analyze GAN artifacts introduced in the inference phase and propose a systematic approach to detect and delete these artifacts. We first measure the relative local variance distance from MSE-based and GAN-based results, and then locate the problematic areas based on the distance map and semantic regions. After detecting the regions with artifacts, we use the MSE-based results as the pseudo ground-truth to finetune the model. By using only a small amount of data, the finetuned model can successfully eliminate artifacts from the inference. Experimental results show the superiority of our approach for detecting and deleting the artifacts and we significantly improve the ability of the GAN-SR model in real-world applications.

6. Acknowledgements

This work was supported in part by Macau Science and Technology Development Fund under SKLIOTSC-2021-2023, 0072/2020/AMJ, and 0022/2022/A1; in part by Natural Science Foundation of China under 61971476, and 62276251; the Joint Lab of CAS -HK; in part by the Youth Innovation Promotion Association of Chinese Academy of Sciences (No. 2020356).

References

Cai, J., Zeng, H., Yong, H., Cao, Z., and Zhang, L. Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3086–3095, 2019.

Chan, K. C., Wang, X., Yu, K., Dong, C., and Loy, C. C. Basicvsr: The search for essential components in video super-resolution and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4947–4956, 2021.

Chan, K. C., Zhou, S., Xu, X., and Loy, C. C. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5972–5981, 2022a.

Chan, K. C., Zhou, S., Xu, X., and Loy, C. C. Investigating tradeoffs in real-world video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5962–5971, 2022b.

Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., and Gao, W. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12299–12310, 2021.

Chen, X., Wang, X., Zhou, J., Qiao, Y., and Dong, C. Activating more pixels in image super-resolution transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22367–22377, June 2023.

Dai, T., Cai, J., Zhang, Y., Xia, S.-T., and Zhang, L. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11065–11074, 2019.

De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009.

Dong, C., Loy, C. C., He, K., and Tang, X. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pp. 184–199. Springer, 2014.

Dong, C., Loy, C. C., He, K., and Tang, X. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

Dong, C., Loy, C. C., and Tang, X. Accelerating the super-resolution convolutional neural network. In *European conference on computer vision*, pp. 391–407. Springer, 2016.

Fuoli, D., Van Gool, L., and Timofte, R. Fourier space losses for efficient perceptual image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2360–2369, 2021.

Gu, J., Lu, H., Zuo, W., and Dong, C. Blind super-resolution with iterative kernel correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Huang, Y., Wang, W., and Wang, L. Video super-resolution via bidirectional recurrent convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):1015–1028, 2017.

Huang, Y., Li, S., Wang, L., Tan, T., et al. Unfolding the alternating optimization for blind super resolution. *Advances in Neural Information Processing Systems*, 33: 5632–5643, 2020.

Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pp. 694–711. Springer, 2016.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690, 2017.

Li, W., Lu, X., Lu, J., Zhang, X., and Jia, J. On efficient transformer and image pre-training for low-level vision, 2021.

Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., and Timofte, R. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1833–1844, 2021.

Liang, J., Cao, J., Fan, Y., Zhang, K., Ranjan, R., Li, Y., Timofte, R., and Van Gool, L. Vrt: A video restoration transformer. *arXiv preprint arXiv:2201.12288*, 2022a.

Liang, J., Zeng, H., and Zhang, L. Details or artifacts: A locally discriminative learning approach to realistic image super-resolution. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pp. 5657–5666, 2022b.

Lim, B., Son, S., Kim, H., Nah, S., and Mu Lee, K. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 136–144, 2017.

Lin, L., Wang, X., Qi, Z., and Shan, Y. Accelerating the training of video super-resolution. *arXiv preprint arXiv:2205.05069*, 2022.

Lugmayr, A., Danelljan, M., and Timofte, R. Ntire 2020 challenge on real-world image super-resolution: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 494–495, 2020.

Ma, C., Rao, Y., Cheng, Y., Chen, C., Lu, J., and Zhou, J. Structure-preserving super resolution with gradient guidance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7769–7778, 2020.

Mittal, A., Soundararajan, R., and Bovik, A. C. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012.

Mou, C., Wu, Y., Wang, X., Dong, C., Zhang, J., and Shan, Y. Metric learning based interactive modulation for real-world super-resolution. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pp. 723–740. Springer, 2022.

Niu, B., Wen, W., Ren, W., Zhang, X., Yang, L., Wang, S., Zhang, K., Cao, X., and Shen, H. Single image super-resolution via a holistic attention network. In *European conference on computer vision*, pp. 191–207. Springer, 2020.

Rad, M. S., Bozorgtabar, B., Marti, U.-V., Basler, M., Ekenel, H. K., and Thiran, J.-P. Srobb: Targeted perceptual loss for single image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2710–2719, 2019.

Shi, S., Gu, J., Xie, L., Wang, X., Yang, Y., and Dong, C. Rethinking alignment in video super-resolution transformers. *arXiv preprint arXiv:2207.08494*, 2022.

Wada, K. Labelme: Image Polygonal Annotation with Python. URL https://github.com/wkentaro/labelme.

Wang, L., Wang, Y., Dong, X., Xu, Q., Yang, J., An, W., and Guo, Y. Unsupervised degradation representation learning for blind super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10581–10590, June 2021a.

Wang, T., Xie, J., Sun, W., Yan, Q., and Chen, Q. Dual-camera super-resolution with aligned attention modules. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2001–2010, October 2021b.

Wang, X., Yu, K., Dong, C., and Loy, C. C. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 606–615, 2018a.

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., and Change Loy, C. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pp. 0–0, 2018b.

Wang, X., Xie, L., Dong, C., and Shan, Y. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1905–1914, 2021c.

Wang, X., Dong, C., and Shan, Y. Repsr: Training efficient vgg-style super-resolution networks with structural re-parameterization and batch normalization. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 2556–2564, 2022.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021a.

Xie, L., Wang, X., Dong, C., Qi, Z., and Shan, Y. Finding discriminative filters for specific degradations in blind super-resolution. *Advances in Neural Information Processing Systems*, 34:51–61, 2021b.

Xie, L., Wang, X., Shi, S., Gu, J., Dong, C., and Shan, Y. Mitigating artifacts in real-world video super-resolution models. *arXiv preprint arXiv:2212.07339*, 2022.

Yang, S., Wu, T., Shi, S., Lao, S., Gong, Y., Cao, M., Wang, J., and Yang, Y. Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1191–1200, 2022.

Zhang, K., Zuo, W., and Zhang, L. Learning a single convolutional super-resolution network for multiple degradations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3262–3271, 2018a.

Zhang, K., Gool, L. V., and Timofte, R. Deep unfolding network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3217–3226, 2020.

Zhang, K., Liang, J., Van Gool, L., and Timofte, R. Designing a practical degradation model for deep blind image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4791–4800, 2021.

Zhang, L., Zhou, Y., Barnes, C., Amirghodsi, S., Lin, Z., Shechtman, E., and Shi, J. Perceptual artifacts localization for inpainting. In *European Conference on Computer Vision*, pp. 146–164. Springer, 2022.

Zhang, W., Liu, Y., Dong, C., and Qiao, Y. Ranksrgan: Generative adversarial networks with ranker for image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3096–3105, 2019.

Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., and Fu, Y. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 286–301, 2018b.

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., and Fu, Y. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2472–2481, 2018c.

A. Appendix

In this appendix, we provide the following materials:

1. More discussions about our work. Refer to Section A.1 in the appendix.
2. More details of GAN-inference artifacts detection pipeline (referring to Section 3.2 in the main paper). Refer to Section A.2 in the appendix.
3. More visual results of GAN-SR artifacts (referring to Section 3.1 in the main paper). Refer to Section A.3 in the appendix.
4. Visual results of GT detection mask labeled by labelme (referring to Section 4.1 in the main paper). Refer to Section A.4 in the appendix.
5. More visual comparisons of different methods on artifact detection results (referring to Section 4.2 in the main paper). Refer to Section A.5 in this supplementary material.
6. Artifact detection results based on SwinIR (referring to Section 4.2 and Section 4.3 in the main paper). Refer to Section A.6 in the appendix.
7. More visual comparisons of results generated from original GAN-SR models and the improved GAN-SR models by using our DeSRA (referring to Section 4.2 in the main paper). Refer to Section A.7 in the appendix.
8. The unreliable of NIQE (Mittal et al., 2012) and MANIQA (Yang et al., 2022) metrics on evaluating the performance of artifacts removal (referring to Tab. 3 in the main paper). Refer to Section A.8 in the appendix.

A.1. More Discussions about Our Work

Discussion 1: why do we introduce the concept of GAN-inference artifacts?

Compared with the previous work, the focus of this work is different and orthogonal. Previous works focus on improving the realness of SR results or mitigating the artifacts generated in the training phase. In real-world scenarios without ground-truth, if one algorithm can restore sharp or real textures but may also generate obviously artifacts, this algorithm is still limited in practical usage since it greatly affects the user experience. For practical application, obviously annoying artifacts are intolerable and the weak restoration results without artifacts are more acceptable by users than the strong restoration results with artifacts. Therefore, dealing with the artifacts that are generated during the inference phase, called GAN-inference artifacts, are of great value for real-world applications. Besides, some cases of artifacts generated during the inference phase are out-of-distribution, so how to alleviate the GAN-inference artifacts is challenging and needs more attention.

Discussion 2: why do we use MSE-SR results as the reference?

We admit that adopting the MSE-SR results as the reference is not optimal to distinguish the GAN-inference artifacts. However, **1)** For real-world testing data, there is no ground-truth. **2)** Detecting the GAN-inference artifacts perfectly is a challenging task. From our experiments, it can be observed that when we adopt the MSE-SR results as the reference to detect the artifacts, there are many overlap areas between our detected artifact map and the GT artifact map. The quantitative and qualitative results illustrate that choosing MSE-SR results as the reference is effective for detecting the GAN-inference artifacts. Deleting the GAN-inference artifacts is a challenging task and this work is the first attempt. We believe there exist other better choices and elegant algorithms to distinguish the GAN-inference artifacts, which needs further exploration.

Discussion 3: why do not we adopt PSNR, SSIM, NIQE . . . metrics?

1) The GAN-inference artifacts appear on unseen real test data, in this circumstance, the corresponding ground-truth images are absent. Therefore, PSNR and SSIM metrics can not be adopted to evaluate the performance. **2)** We test some no-reference metrics (e.g, NIQE and MANIQA), and observe that these no-reference IQA metrics do not always match perceptual visual quality (Lugmayr et al., 2020) (see Section A.8). 3) The focus of this work is on detecting and alleviating GAN-inference artifacts. Motivated by the binary classification task, we adopt three metrics (, IOU, precision, and recall) to evaluate the performance.

Discussion 4: why do we assume that GAN-Artifact is usually a large area?

The GAN-inference artifacts are complicated and diverse, which appear in both large areas and small areas. Previous works focus on dealing with GAN-training artifacts and ignore the GAN-inference artifacts. When applied in real-world scenarios, those methods still generate obviously annoying artifacts during inference. Dealing with GAN-inference artifacts is a challenging task and there need several steps to resolve this problem. This work is the first attempt, and we only consider the artifacts that are obvious and have a large area, since this kind of artifact has a great impact on human perception. We hope that more researchers will pay attention to solving the GAN-inference artifacts, and the following works can deal with the GAN-inference artifacts that have a small area.

Discussion 5: semantic segmentation.

We admit that the detection results based on semantic segmentation are not entirely accurate, while it can get roughly accurate results to help distinguish artifacts and guide us for further processing of these artifacts, and the lost precision has a limited impact on practical applications.

Discussion 6: online continual learning.

Our method can provide a new paradigm combined with continual learning (De Lange et al., 2021) to address the artifacts that appear in the inference stage online. For example, for an online SR system that processes real-world data, we can use our detection pipeline to detect whether the results have GAN-inference artifacts. We can then use the images with detected artifacts to quickly finetune the SR model, so that it can deal with similar kinds of artifacts until the system encounter a new kind of GAN-inference artifacts. Continual learning is widely studied on high-level vision tasks, but has not been applied to SR. Our approach and application scenes naturally introduce continual learning to SR. We hope to investigate this problem in the future, since it can greatly advance the application of GAN-SR methods in practical scenarios.

A.2. Details of GAN-inference artifacts detection pipeline

In this section, we first describe the details of GAN-inference artifacts detection pipeline. Then, we provide more details about calculating the adjustment weights.

Overall pipeline of detecting GAN-inference artifacts. The pipeline of detecting GAN-inference artifacts is shown in Fig. 10 (a). For a GAN-SR and MSE-SR result, we first calculate the indicator D according to equation 7 in the main paper. Then we generate the segmentation map of MSE-SR result by adopting SegFormer. The segmentation map will be converted into semantic-aware adjustment weight A according to the calculated adjustment weights of each semantic class (Fig. 10 (b)). By combining A, D and setting $threshold$, we can obtain the refined detected map M:

$$M(i,j) = \begin{cases} 0, & D(i,j)/A_k \geq threshold; \\ 1, & D(i,j)/A_k < threshold, \end{cases} \tag{14}$$

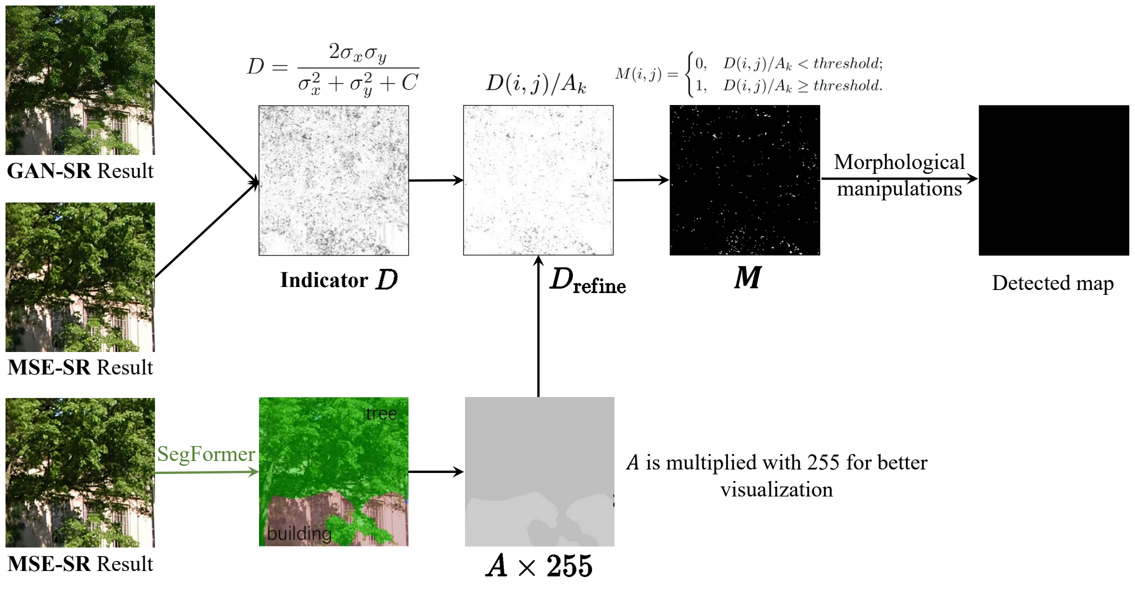
where $D(i,j)$ is the D value of pixel (i,j) and $threshold$ is empirically set to 0.7. At last, we perform morphological manipulations to obtain the final detected map. Concretely, we first perform erosion using a 5×5 all-ones matrix. Then we implement dilation using the matrix to join disparate regions. Next, we fill the hole in the map by using a 3×3 all-ones matrix. Finally, we filter out discrete small regions as the detection noise.

Note that the visualization results of indicator D and D_{refine} in Fig. 10 (a) are different from Fig. 5 in the main paper. Here we show their original values. In the main paper, for better understanding, we show their corresponding binary maps by comparing their original values with the threshold 0.7. Values that are smaller than the threshold 0.7 are set to 1.

Details of calculating adjustment weights. The calculation of adjustment weights for each semantic class is illustrated in Fig. 10 (b). We first generate the corresponding low-resolution version by adopting the degradation model used in the training phase on the DIV2K training dataset. Then, we generate the MSE-SR and GAN-SR results for each distorted image. After that, we calculate pixel-wise indicator D between the MSE-SR and GAN-SR results. To distinguish D of each semantic class, we choose SegFormer (Xie et al., 2021a) as the segmentation model, and obtain the segmentation map of MSE-SR results. By incorporating the segmentation map and indicator D, we get pixel-wise D values in each class of DIV2K. For each class, we sort all the D values in descending order and set the D value in the 85% percentile as the adjustment weight:

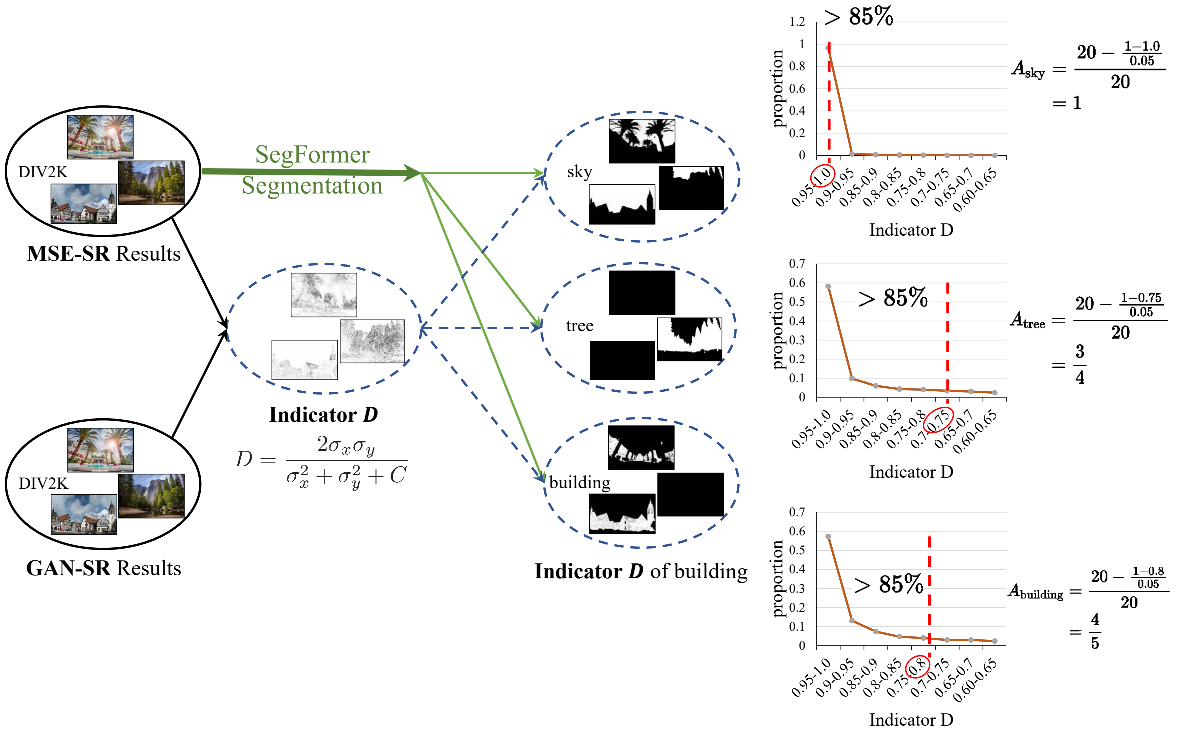
$$A_k = P_{85}(D_k), k \in \{1, 2, ..., K\}, \tag{15}$$

where A_k is the adjustment weight for the k^{th} class, D_k denotes the D value of all pixels identified as the k^{th} class, and P_{85} is the 85^{th} percentile operation. For example, the values of A_{sky}, A_{tree} and A_{building} are 1, 0.75 and 0.80, respectively.



GAN-SR Result
MSE-SR Result
MSE-SR Result
$D = \frac{2\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2+C}$
Indicator D
$D(i,j)/A_k$
D_{refine}
$M(i,j) = \begin{cases} 0, & D(i,j)/A_k < threshold; \\ 1, & D(i,j)/A_k \geq threshold. \end{cases}$
M
Morphological manipulations
Detected map
SegFormer
tree
building
$A \times 255$
A is multiplied with 255 for better visualization

(a) Pipeline of detecting GAN-inference artifacts



DIV2K
MSE-SR Results
DIV2K
GAN-SR Results
SegFormer Segmentation
Indicator D
$D = \frac{2\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2+C}$
sky
tree
building
Indicator D of building
$> 85\%$
proportion
Indicator D
0.95-1.0
0.9-0.95
0.85-0.9
0.8-0.85
0.75-0.8
0.7-0.75
0.65-0.7
0.60-0.65
$A_{\text{sky}} = \frac{20 - \frac{1-1.0}{0.05}}{20} = 1$
$A_{\text{tree}} = \frac{20 - \frac{1-0.75}{0.05}}{20} = \frac{3}{4}$
$A_{\text{building}} = \frac{20 - \frac{1-0.8}{0.05}}{20} = \frac{4}{5}$

(b) Calculation of adjustment weights for each semantic class

Figure 10. The overall pipeline of detecting GAN-inference artifacts and the calculation of adjustment weights. Note that to meet the max size requirement for the camera-ready version, we applied some compression to the image, which might have affected the quality displayed in the original picture. For a better understanding of our paper, please refer to the images in our arXiv paper.

A.3. More Visual Results of GAN-SR Artifacts

In real-world scenarios, GAN-SR models often introduce severe perceptually-unpleasant artifacts that seriously affect the visual quality of restored images. As depicted in Fig. 11, in some cases, the GAN-SR artifacts would make the results even worse than those generated by the MSE-based model.

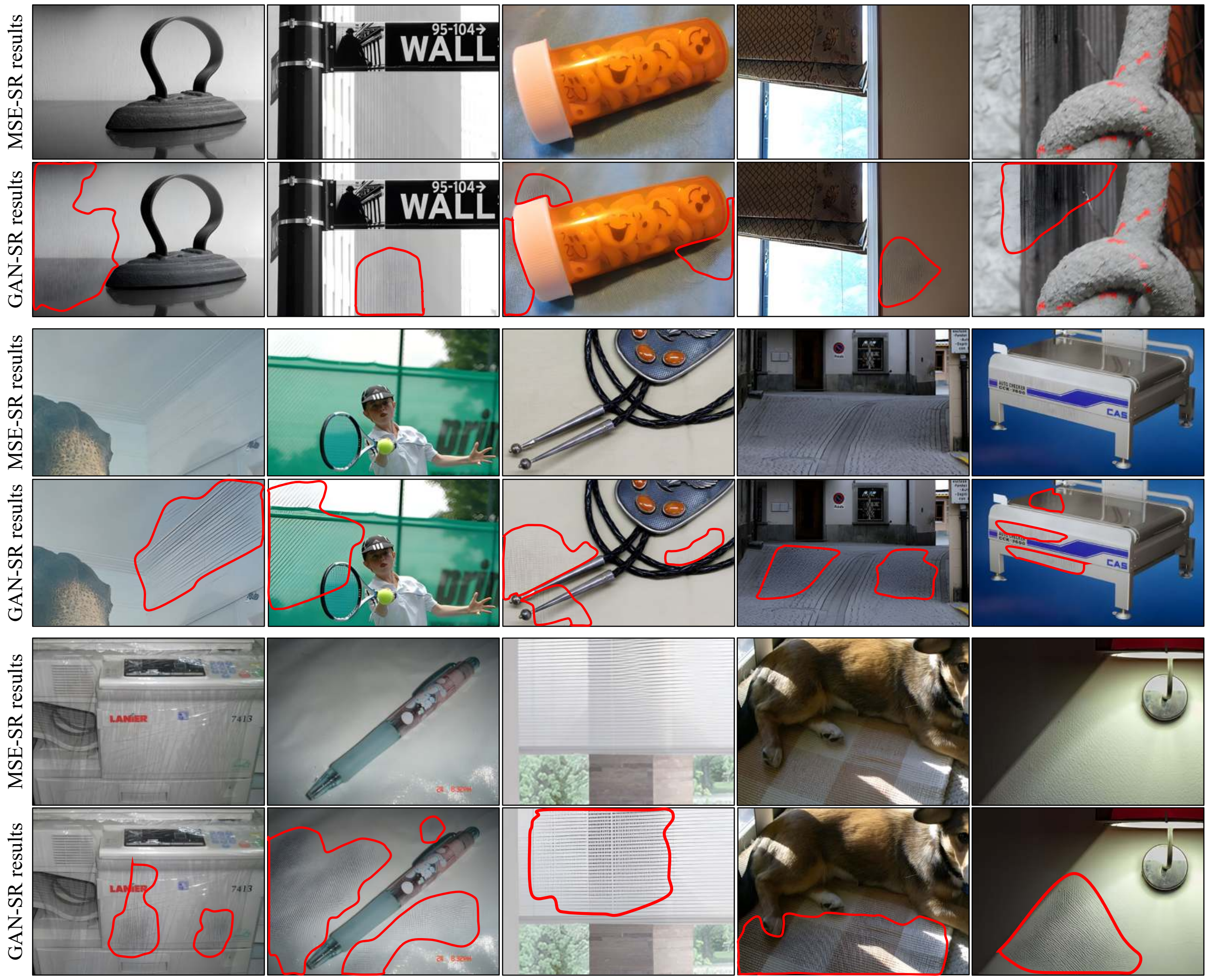


Figure 11. MSE-SR and GAN-SR results of some practical samples. GAN-SR results with artifacts have even worse visual quality than MSE-SR results. The artifacts are complicated with different types and characteristics, and are diverse for different image contents. Regions with red circles are GT of the detection mask. Note that to meet the max size requirement for the camera-ready version, we apply some compression to the image, which might have affected the quality displayed in the original picture. For a better understanding of our paper, please refer to the images in our arXiv paper.

A.4. Visual Results of GT Detection Mask

For Real-ESRGAN (Wang et al., 2021c), LDL (Liang et al., 2022b) and SwinIR (Liang et al., 2021), we construct their independent GAN-SR artifacts datasets. Each dataset contains 200 representative images with GAN-inference artifacts. Since there is no ground-truth map for artifact regions to evaluate the algorithm, we manually label the artifact areas using labelme (Wada) and generate a binary map to indicate the artifact region, as shown in Fig. 12.

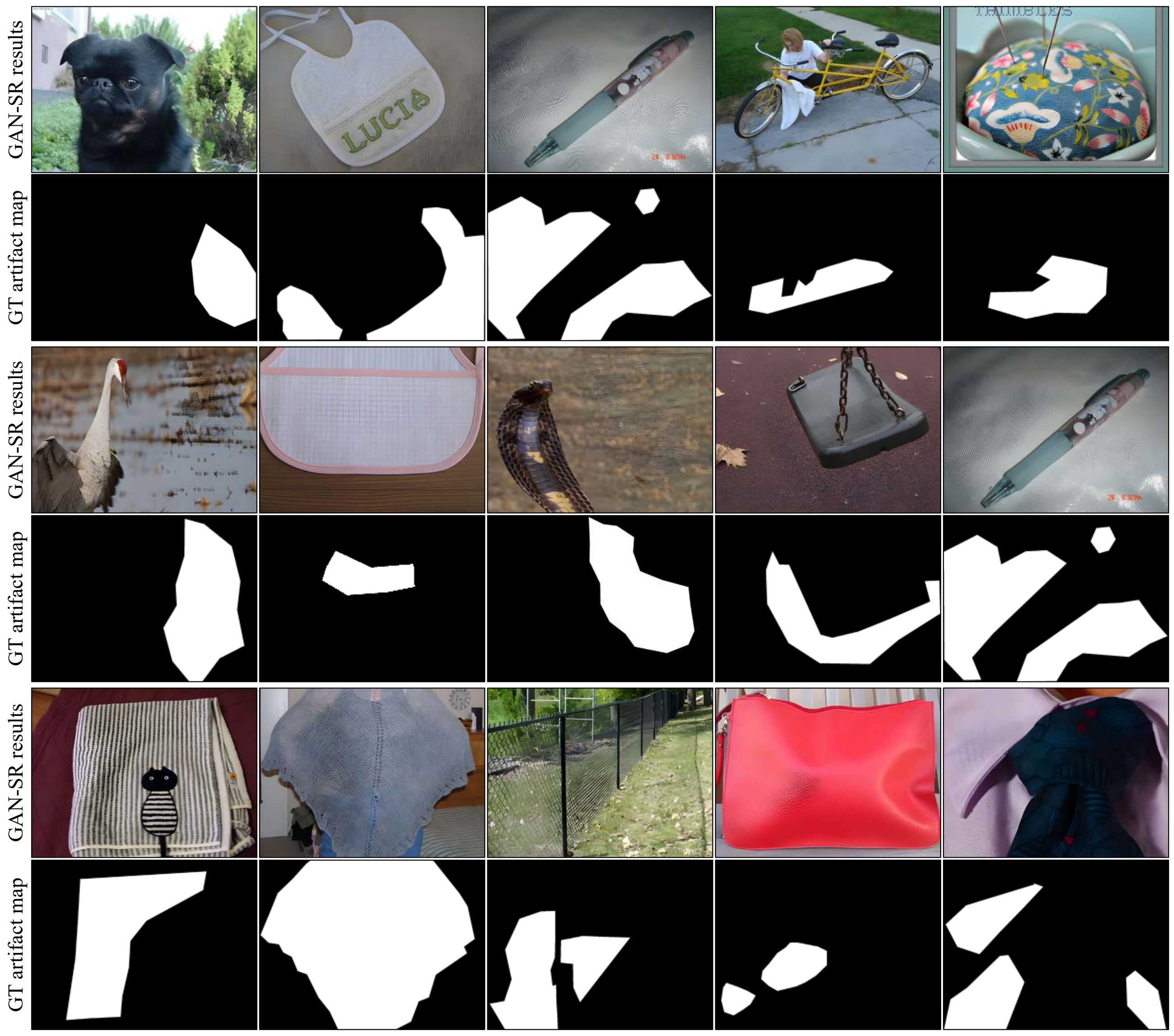


Figure 12. Visualization of GT artifact map. For GAN-SR results with artifacts, we manually generate their corresponding binary artifact map with labelme. The white regions of GT artifact map indicate the artifact regions in GAN-SR results. Note that to meet the max size requirement for the camera-ready version, we apply some compression to the image, which might have affected the quality displayed in the original picture. For a better understanding of our paper, please refer to the images in our arXiv paper.

A.5. More Visual Comparisons of Different Methods on Artifact Detection Results

For the GAN-inference artifacts generated by Real-ESRGAN (Wang et al., 2021c), LDL (Liang et al., 2022b) and SwinIR (Liang et al., 2021), we compare different methods on artifact detection results. The visual comparison is presented in Fig. 13. The detection results obtained by our approach have significantly higher accuracy than other schemes.

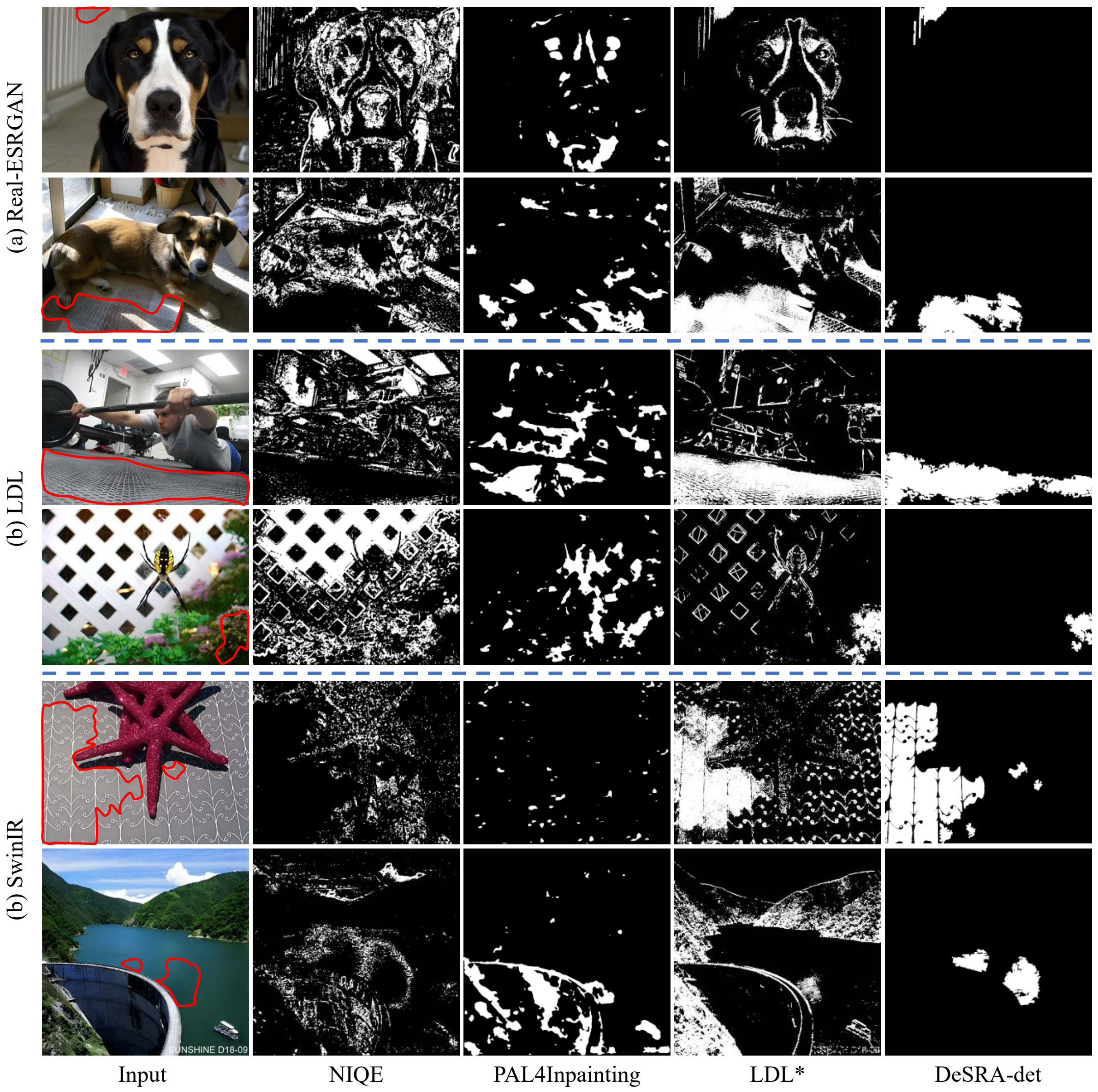


Figure 13. Visual comparison of different methods on artifact detection results. Regions with red circles are GT of the detection mask. Note that to meet the max size requirement for the camera-ready version, we apply some compression to the image, which might have affected the quality displayed in the original picture. For a better understanding of our paper, please refer to the images in our arXiv paper.

A.6. Artifact Detection Results based on SwinIR

To validate the effectiveness of our proposed GAN-inference artifact detection algorithm and fine-tuning strategy, we further conduct experiments based on SwinIR. Due to the lack of the official-released pretrained weight of the discriminator, we retrain SwinIR using the officially released codes[1] in real setting and obtain the corresponding MSE-SR and GAN-SR models. For the GAN-inference artifacts generated by SwinIR, the artifact detection results are shown in Tab. 6. We can observe that our method obtains the best IoU and Precision that far outperforms other schemes.

After obtaining the detected artifacts map, we finetune SwinIR with 1000 iterations to alleviate the GAN-inference artifacts. As depicted in Tab. 7, after the application of our DeSRA, IoU decreases from 57.9 to 21.8, illustrating that the detected area of artifacts is greatly reduced. The removal rate is 61.35%, showing that three-fifths of the artifacts on unseen test data can be completely removed after fine-tuning. Besides, our method does not introduce new additional artifacts, as the addition rate is 0.

Table 6. Artifact detection results based on SwinIR (Liang et al., 2021). LDL* represents the modified detection method in LDL (Liang et al., 2022b).

Method	IoU (↑)	Precision	Recall
NIQE (Mittal et al., 2012)	2.3	0.0227	0.0668
PAL4Inpainting (Zhang et al., 2022)	6.3	0.0547	0.1277
LDL*(threshold=0.01) (Liang et al., 2022b)	11.0	0.3039	0.1176
LDL*(threshold=0.005)	19.4	0.2377	0.2647
LDL*(threshold=0.001)	29.2	0.1473	0.7380
LDL*(threshold=0.0001)	29.2	0.1473	0.7380
DeSRA-det (ours)	**57.9**	**0.7600**	**0.7412**

Table 7. Artifact detection results of SwinIR (Liang et al., 2021) with and without using DeSRA finetuning.

Method	IoU (↓)	Removal rate	Addition rate
SwinIR (Liang et al., 2021)	57.9	-	-
SwinIR-DeSRA	**21.8**	61.35%	0%

A.7. More Visual Comparisons between the Original GAN-SR Models and the Improved GAN-SR Models with DeSRA

We provide the visual comparison between results with and without using our method to improve GAN-SR models, as shown in Fig. 14, Fig. 15 and Fig. 16. We can observe that results generated by the improved GAN-SR models have greatly better visual quality without obvious GAN-SR artifacts compared to the original inference results. All these experimental results demonstrate the effectiveness of our method for alleviating the artifacts and improving the GAN-SR model (Real-ESRGAN, LDL, and SwinIR).

[1]https://github.com/JingyunLiang/SwinIR

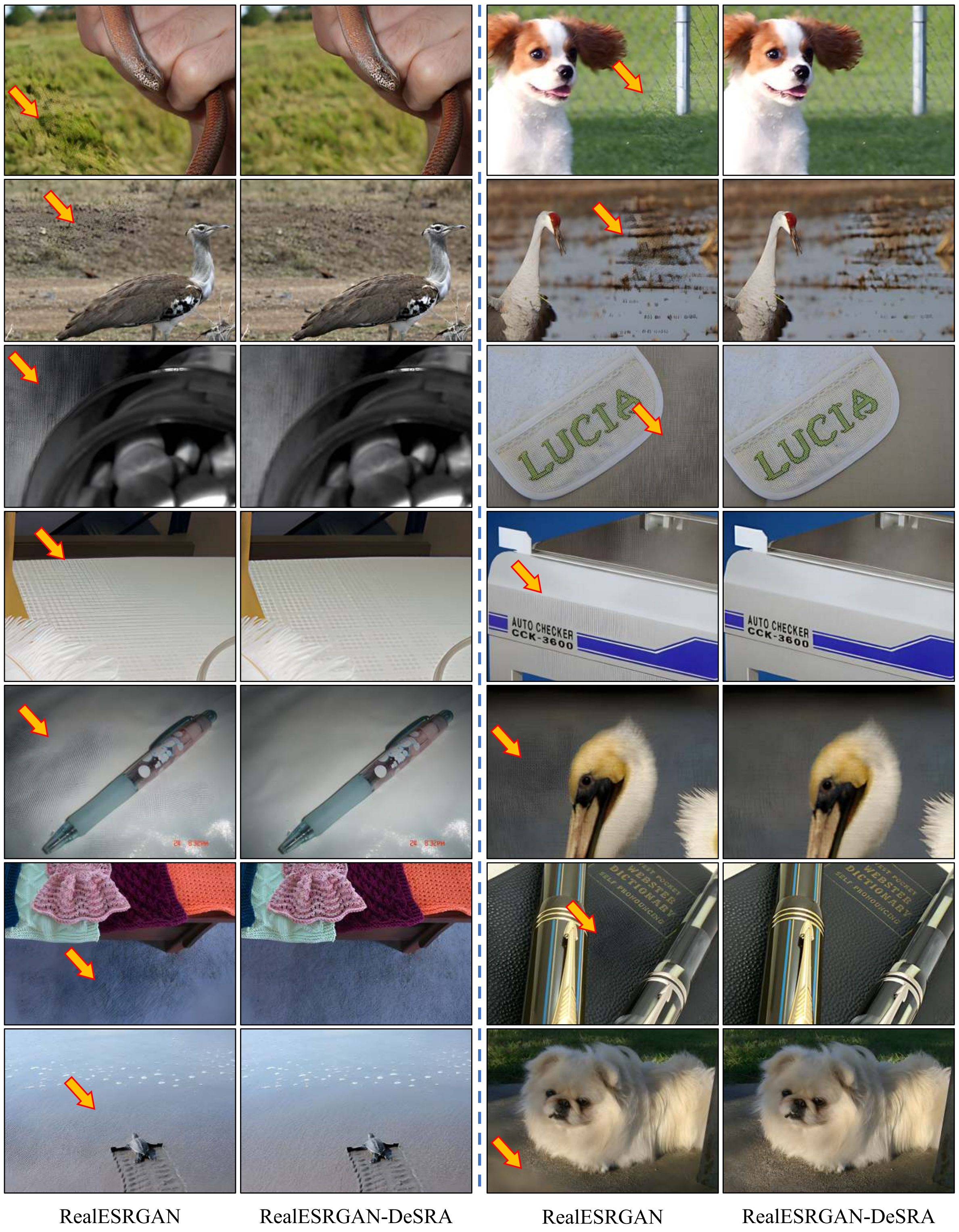


Figure 14. Visual comparison of results generated from RealESRGAN and RealESRGAN-DeSRA. Artifacts are obviously alleviated for results produced by RealESRGAN-DeSRA. Note that to meet the max size requirement for the camera-ready version, we apply some compression to the image, which might have affected the quality displayed in the original picture. For a better understanding of our paper, please refer to the images in our arXiv paper.

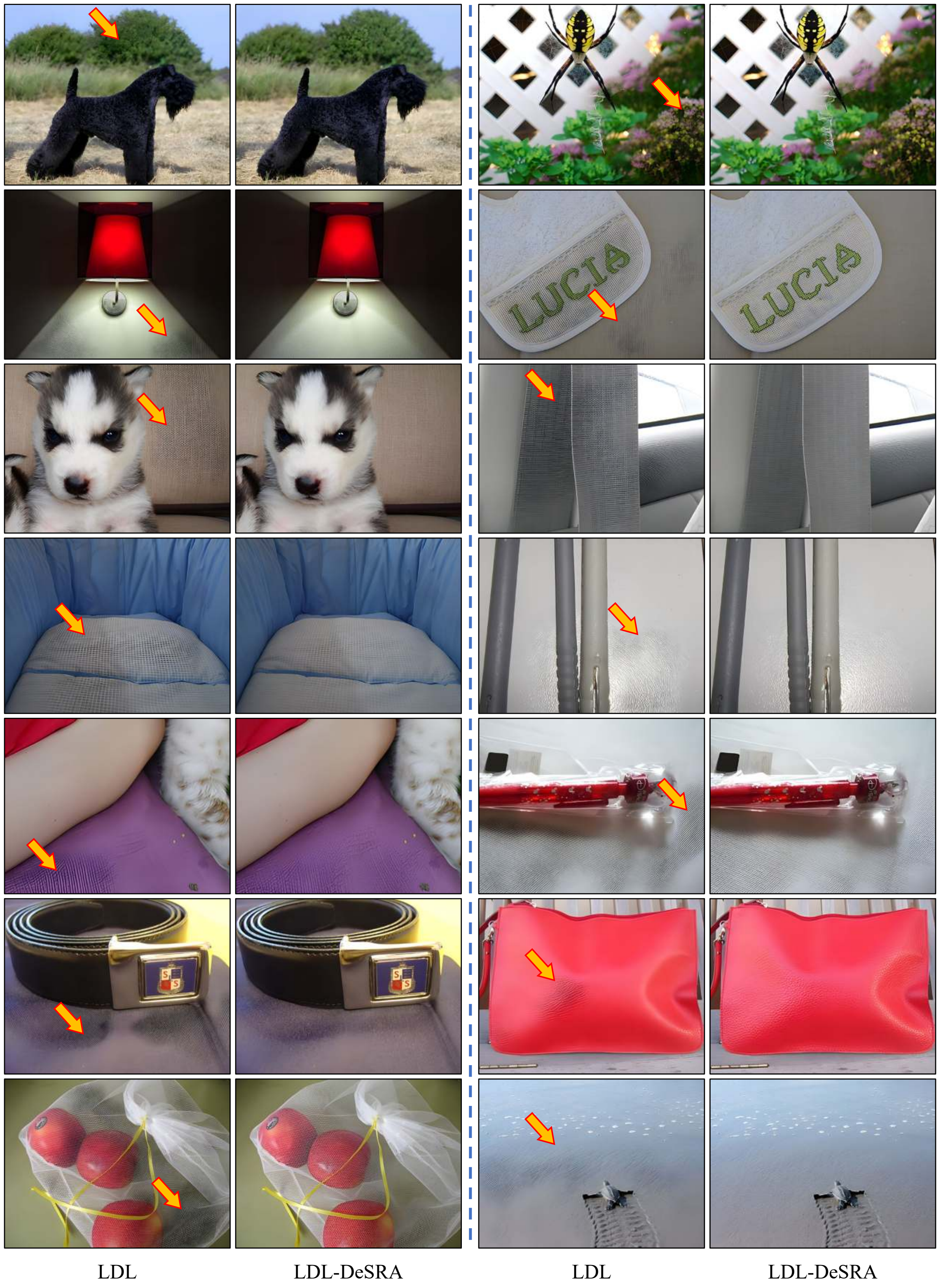


Figure 15. Visual comparison of results generated from LDL and LDL-DeSRA. Artifacts are obviously alleviated for results produced by LDL-DeSRA. Note that to meet the max size requirement for the camera-ready version, we apply some compression to the image, which might have affected the quality displayed in the original picture. For a better understanding of our paper, please refer to the images in our arXiv paper.

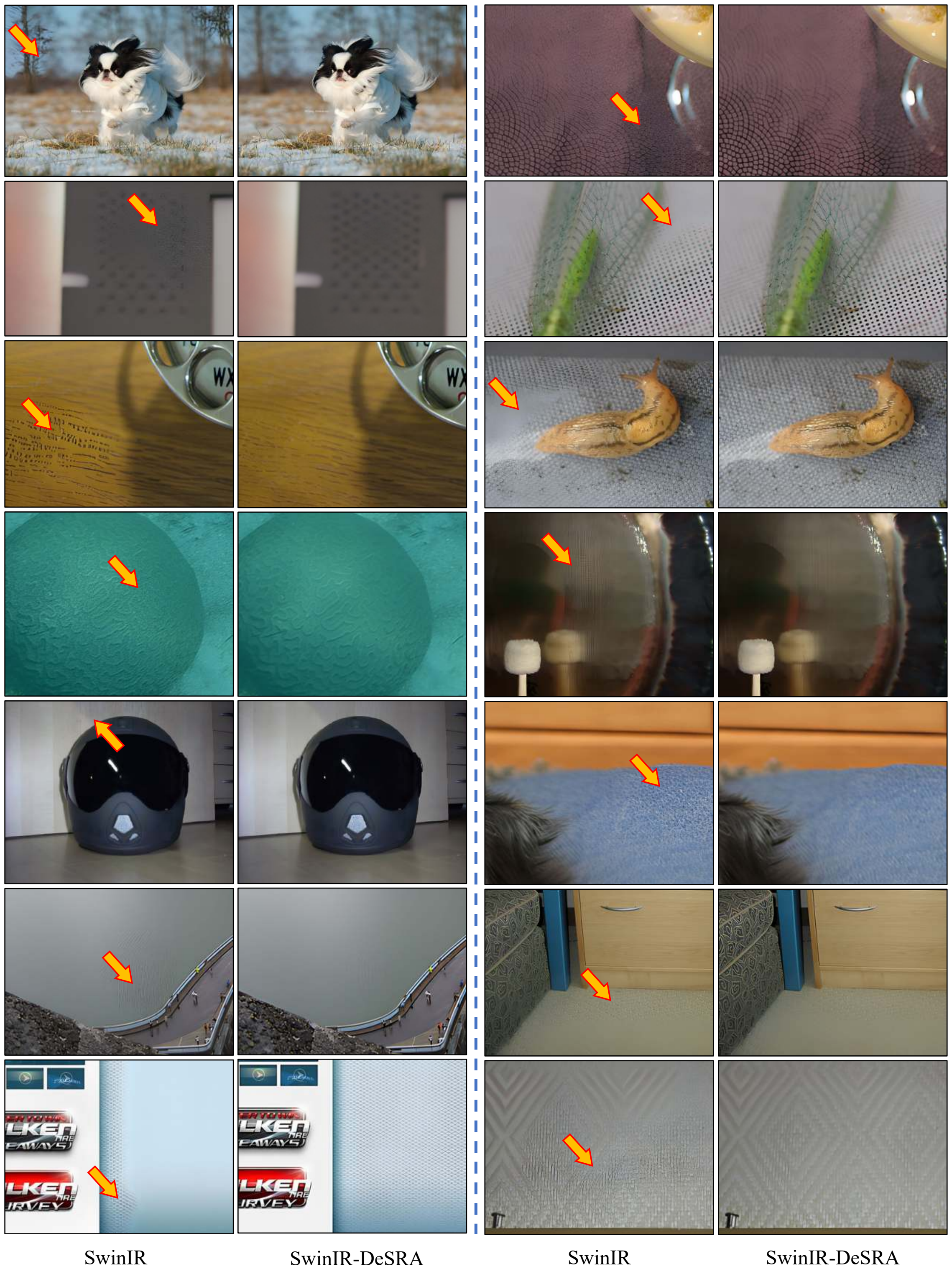


Figure 16. Visual comparison of results generated from SwinIR and SwinIR-DeSRA. Artifacts are obviously alleviated for results produced by SwinIR-DeSRA. Note that to meet the max size requirement for the camera-ready version, we apply some compression to the image, which might have affected the quality displayed in the original picture. For a better understanding of our paper, please refer to the images in our arXiv paper.

A.8. Unrealiable of NIQE and MANIQA Metrics

In Tab. 3 of the main paper and Tab. 7 in this supplementary material, we adopt IoU, Removal rate, and Addition rate metrics to evaluate the performance of improved GAN-SR models with DeSRA. Although NIQE (Mittal et al., 2012) is the commonly-used metric in GAN-SR works, we observe that this metric cannot well reflect the performance of the improved GAN-SR models. As illustrated in Fig. 17, it can be obviously observed that the images in the second column have better visual results with fewer artifacts than the images in the first column. However, the values of NIQE (lower is better) and MANIQA (Yang et al., 2022) (higher is better) show the opposite results. MANIQA is the champion of the NTIRE 2022 Perceptual Image Quality Assessment Challenge. Therefore, we do not adopt these two metrics to evaluate the performance.

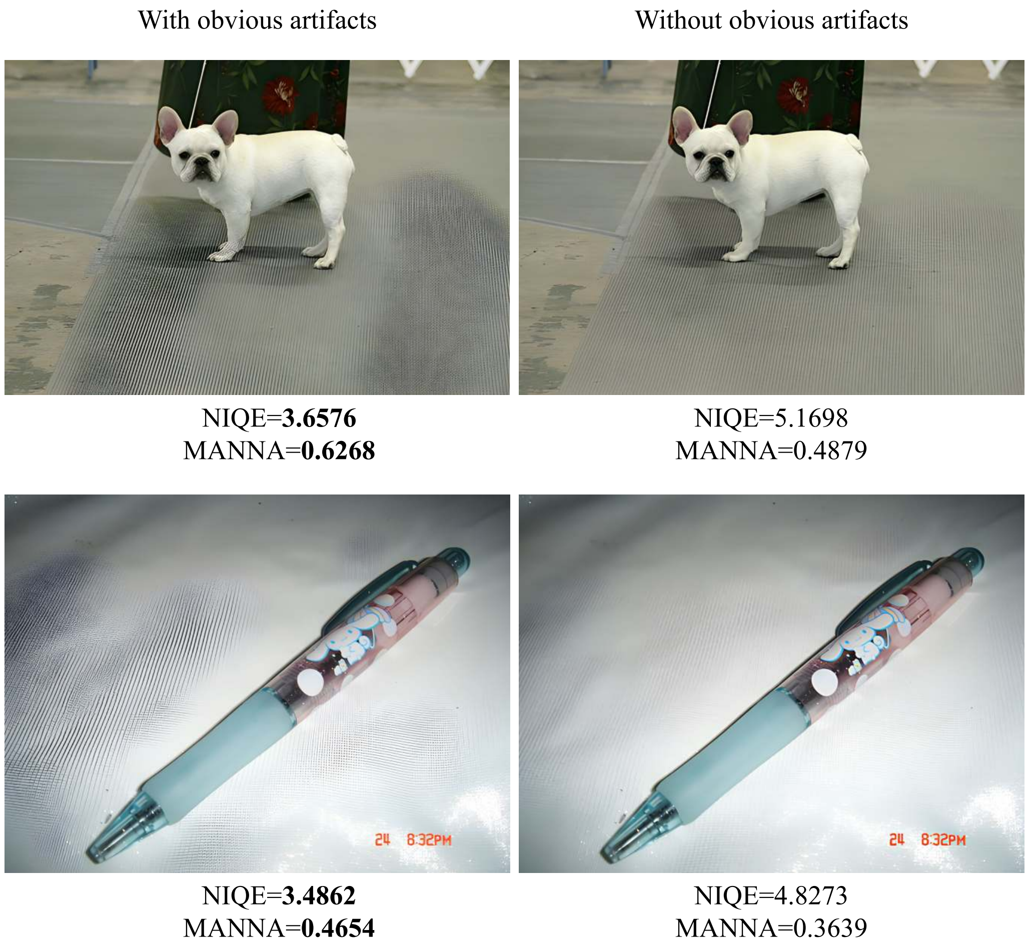


Figure 17. Evaluation of images with or without obvious artifacts on NIQE and MANNA metrics. Both of these two metrics cannot reflect the effects of artifact removal. Note that to meet the max size requirement for the camera-ready version, we apply some compression to the image, which might have affected the quality displayed in the original picture. For a better understanding of our paper, please refer to the images in our arXiv paper.