Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 5217429, 12 pages
https://doi.org/10.1155/2020/5217429

Research Article

GAN-Based Image Super-Resolution with a Novel Quality Loss

Xining Zhu, Lin Zhang, Lijun Zhang, Xiao Liu, Ying Shen, and Shengjie Zhao

School of Software Engineering, Tongji University, Shanghai 201804, China

Correspondence should be addressed to Lin Zhang; cslinzhang@tongji.edu.cn

Received 20 September 2019; Revised 29 December 2019; Accepted 29 January 2020; Published 18 February 2020

Guest Editor: Marco Perez-Cisneros

Single image super-resolution (SISR) has been a very attractive research topic in recent years. Breakthroughs in SISR have been achieved due to deep learning and generative adversarial networks (GANs). However, the generated image still suffers from undesired artifacts. In this paper, we propose a new method named GMGAN for SISR tasks. In this method, to generate images more in line with human vision system (HVS), we design a quality loss by integrating an image quality assessment (IQA) metric named gradient magnitude similarity deviation (GMSD). To our knowledge, it is the first time to truly integrate an IQA metric into SISR. Moreover, to overcome the instability of the original GAN, we use a variant of GANs named improved training of Wasserstein GANs (WGAN-GP). Besides GMGAN, we highlight the importance of training datasets. Experiments show that GMGAN with quality loss and WGAN-GP can generate visually appealing results and set a new state of the art. In addition, large quantity of high-quality training images with rich textures can benefit the results.

1. Introduction

Single image super-resolution (SISR) aims at recovering a high-resolution (HR) image from a low-resolution (LR) one. It is inherently ill posed since for one LR image, there exists multiple HR ones that could generate it. Although there have been many breakthroughs in addressing SISR, there is still one challenge: how to recover photorealistic results with more natural textures and less unpleasant artifacts. To this end, traditional methods and learning-based methods are proposed in succession. It is worth mentioning that a learning-based method constructs an HR image by learning a nonlinear LR-to-HR mapping, conducted by a predefined deep neural network with a sophisticated loss function, which is also the solution we chose.

The mean squared error (MSE) loss is often used as the term of loss function when learning-based methods become popular. Although MSE has clear physical meaning, it is convenient for calculating and favors a high peak signal to noise ratio (PSNR) value, and it tends to generate overly smooth results. Actually, the MSE loss is equivalent to PSNR, which is still a main image quality assessment (IQA) criterion for evaluating the performance of SISR. Unfortunately, PSNR relies only on low-level differences between pixels and is implemented under the assumption of additive Gaussian noise, which may be invalid for SISR. We therefore attempt to alleviate the influence of MSE loss by crafting a novel loss term named quality loss and no longer aim to achieve state-of-the-art PSNR results.

Besides the problem of MSE loss, there are two other key issues that need to be addressed. First, since generative adversarial networks (GANs) have been proved to facilitate generating more visually appealing results, many learning-based methods adopt them. However, the original GAN suffers from training instability. Second, for learning-based methods, training dataset is crucial but easily overlooked. For example, ImageNet [1] is a popular training dataset for SISR. Although it contains large quantity of images, the quality of them is relatively poor for this dataset is built for classification originally, thus not suitable for SISR. Instead, professional datasets containing abundant high-quality images should be the alternative. We attempt to solve the aforementioned issues, and Figure 1 shows a representative result of our proposed solution, gradient map generative adversarial network (GMGAN). It can be seen that the generated details are nearly indistinguishable from the ground truth (refer to Section 4.3 for more result comparisons).

Figure 1: The lower left and lower right are the LR images I_{LR}, the middle left is the HR image I_{HR}, and the middle right is the SR image I_{SR} generated by GMGAN. Compared with I_{LR}, I_{SR} is much sharper. Compared with I_{HR}, I_{SR} is nearly indistinguishable, e.g., in I_{SR}, the cat hairs are lifelike and its eyes are sharp and bright.

The remainder of this paper is organized as follows. Section 2 presents the related work and our contributions. Section 3 introduces our proposed GMGAN, integrating quality loss and WGAN-GP together. Experimental results are presented in Section 4. Section 5 concludes the paper finally.

2. Related Work and Our Contributions

Among the solutions of SISR, early methods are based on prediction [2, 3], edge [4–6], statistic [7–9], sparse dictionary [10], patch recurrence [11, 12], etc. These methods can solve the problem of SISR to some extent. However, with the invention of learning-based methods, traditional methods pale in comparison, whether in performance or efficiency. Therefore, we focus our discussion on learning-based methods.

2.1. Network Architecture for Learning-Based SISR. Our proposed SISR solution, GMGAN (refer to Section 3 for details) is based on a network architecture, and thus we firstly give a brief review about the development of network architecture for learning-based methods in this section.

The research in this area begins with Dong et al.'s pioneer work [13]. In [13], Dong et al. proposed a deep learning method named SRCNN that directly learns an end-to-end mapping between the LR and HR images. The network architecture consists of three layers: the first convolutional layer, extracting a set of feature maps; the second layer, mapping these feature maps nonlinearly to high-resolution patch representations; and the last layer, combining the predictions to produce the final super-resolution image. The drawbacks of this simple model are that the input LR image needs extra interpolation operation and the nonlinear mapping step is computationally costly. To solve the aforementioned problems, Dong et al. proposed FSRCNN [14]. In [14], a deconvolutional layer is adopted to replace the bicubic interpolation. Meanwhile, a shrinking and an expanding layer are added at the beginning and the end of the mapping layers, respectively, to restrict mapping in a low-dimensional feature space. Although SRCNN and FSRCNN have demonstrated great superiority over traditional methods due to the strong capability of CNN to learn useful representations in an end-to-end manner, the layers and complexity of network architectures are limited. In other words, more elaborate architectures with proper depth, width, and topology may lead to much better results. Moreover, some specific requirements could be taken into consideration during the process of designing the network architecture, such as large scale factor [15, 16], unknown downsampling [17], and realistic texture details [18, 19]. The following related work introduction will focus on these two points: (1) more elaborate architectures with proper depth, width, and topology and (2) more task-specific architectures considering special requirements of SISR.

2.1.1. More Elaborate Architectures with Proper Depth, Width, and Topology. Increasing the depth and width of network has been reported to effectively improve the final performance. For instance, VDSR [20] is a 20-layer VGG-net. Different from mapping directly from the bicubic image to its HR version like SRCNN, VDSR learns their residuals by introducing a residual structure and finally adds the learned residuals to the bicubic image to get the final output. Despite achieving improved performance and accelerated convergence speed, VDSR suffers from redundant parameters and training difficulty. DRCN [21] proposed a multisupervised strategy to overcome the drawbacks of VDSR. The strategy can help gradients to flow more smoothly during backpropagation and assist all the intermediate representations to reconstruct the SR image. However, the strategy has two drawbacks as for fusion: (1) the weighted scalars are fixed in the training process, which cannot be modified, and (2) weighting the SR image by using single scalars ignores pixelwise differences. To sum up, these two methods both use a VGG-net, which is a plain architecture. It does improve the performance of the network but stacking layers and extending width roughly cannot improve the performance indefinitely.

To go deeper, with the development of ResNet [22] and DenseNet [23], SRResNet [24] and SRDenseNet [25] were proposed in succession to solve SISR. Based on skip connection, SRResNet is comprised of 16 residual units. Each residual unit consists of convolutions, ReLUs, and batch normalization (BN) layers, where BN is reported to stabilize the training process. Lim et al. proposed EDSR [26] to enhance deep residual networks for SISR. Unlike SRResNet, EDSR removes all BN layers as BN is designed for high-level computer vision problems like classification at first, where inner representations are relatively abstract and insensitive to the shift by BN. However, for low-level vision problems like SISR, there is relatively strong relationship between the input and output images, and such shift could actually harm the final performance. Removing BN layers brings an additional benefit: sufficiently reducing the GPU memory usage. Thus, under limited computational resources, EDSR can contain more residual units, going deeper. Moreover, since different scales may share many intermediate representations, Lee et al. initialized the parameters with a pretrained ×2 network when

training the models for ×3 and ×4. The effectiveness of the pretraining strategy lies in accelerating the training and improving the final performance. ResNet benefits feature reusage, while DenseNet benefits feature exploration. In [25], a set of dense blocks (like residual units) are connected by skip connection. In each block, short paths are built between a layer and every other layer, strengthening the flow of information through dense blocks and alleviating the vanishing-gradient problem. In order to reduce the number of feature maps, a bottleneck layer, which is actually a convolution layer with 1×1 kernel, is used before all the feature maps are directly fed into deconvolution layers. The bottleneck layer can effectively keep the model compactness and improve the computational efficiency, which is also adopted by [27].

2.1.2. More Task-Specific Architectures considering Special Requirements of SISR. To solve more specific tasks, special requirements could be taken into account when designing the network architecture.

One representative challenging task is the big scale factor (e.g., ×8). Lai et al. proposed LapSRN [15] to progressively reconstruct the subband residuals of HR images with a Laplacian pyramidal structure in 3 levels. The architecture has two branches: feature extraction and image reconstruction. At each level, the image reconstruction branch estimates an intermediate SR output; the feature extraction branch outputs a residual image between the raw estimator and corresponding HR image, extracting effective representations for the next level. LapSRN can generate competitive results for big scale.

However, the recursive pyramid results in quadratic growth of computation in the higher pyramidal level, hindering further reducing runtime and expanding the network capability. To perfect the work of LapSRN, a fully progressive approach for SISR [16] adopts an asymmetric pyramidal architecture. Compared with LapSRN, the intermediate SR outputs are neither supervised nor used as base images in subsequent levels to simplify the backward pass and reduce the optimization difficulty. Besides, original convolutions are replaced with dense compression units (DCUs) and more DCUs are put in the lower level to reduce the memory consumption and increase the receptive field. Such architecture results in an asymmetric structure, outperforming the symmetric equivalent in terms of runtime and reconstruction quality. These two methods can tackle the problem of big scale partly, but they do not fully address the mutual dependencies of LR and HR images. Haris et al. proposed DBPN [28], which exploits iterative upsampling and downsampling layers, providing an error feedback mechanism for projection errors at each stage. DBPN can be divided into three parts: initial feature extraction, projection, and reconstruction. Initial LR feature maps are constructed from the input LR image. Then, a sequence of projection units alternates between the construction of LR and HR feature maps. Finally, the feature maps produced in each up-projection unit are concatenated to reconstruct the final SR image.

Apart from big scale, there are many other specific tasks for SISR. For example, to recover natural texture, Wang et al. proposed spatial feature transform (SFT) layer [19], which can be conveniently applied to existing SR networks. Concretely, segmentation probability maps are fed into the condition network firstly to generate intermediate conditions to be shared by all the SFT layers. Then, the shared conditions run through the process of modulating the feature maps by applying affine transformation. Finally, distinct and rich textures are generated for multiple semantic regions in the final SR image. However, the effectiveness of this method highly depends on the image quality and categories of the segmentation probability maps fed into the conditional network.

For our proposed GMGAN, we aim to generate natural and realistic images with less unpleasant artifacts. We adopt one architecture, combining the advantages of ResNet and DenseNet, whose details are presented in Section 3.

2.2. Loss Function: Optimization Objective for Learning-Based SISR. For learning-based methods, there is no doubt that a suitable network architecture can benefit the final results for the most part. It is also worth mentioning the importance of loss function. In fact, it is the loss function that defines how distributions of the generated image and the ground truth get closer to each other, which can be seen as the soul of learning-based methods. Generally, the loss function is a weighted sum of several loss terms.

The most widely used loss term is pixelwise loss. MSE [29] and Charbonnier penalty [15] are its two representatives. As mentioned above, MSE favors a high PSNR value and PSNR is still a widely used metric for evaluating the performance of SISR quantitatively. Nevertheless, models with these per-pixel losses tend to generate unnatural and overly smooth results since it is poorly connected with human vision system (HVS) [29]. To address the shortcomings of pixelwise loss, perceptual loss [24, 30, 31] is proposed to combine or replace with it. Optimizing perceptual loss based on features extracted from a pretrained network can generate high-quality results. Actually, the pretrained network has already learned to encode the semantic and perceptual information we would like to measure in the loss term. Concretely, the network pretrained originally for image classification is fixed to define the loss term, which in our experiments is the 19-layer VGG network [32] pretrained on the ImageNet dataset [1]. Another attempt to address the limitations of the simple pixelwise loss is inspired by the structural similarity index (SSIM) [33]. SSIM takes luminance, contrast, and structure into consideration. It is further extended to MS-SSIM [34], which is a multiscale version of SSIM that weighs SSIM computed at different scales according to the sensitivity of the HVS. Because SSIM and MS-SSIM are differentiable, they can be used as cost functions. Zhao et al. [35] conducted experiments on them and showed their feasibility. Global loss can effectively capture style and texture by comparing statistics collected over the entire image. For example, Gram loss [18] is defined as a matrix based on iterative optimization. For a given target texture image, the output image is generated iteratively by matching statistics extracted from a pretrained

network, which is also a VGG network, to the target texture. This method is slow and only works if a target texture is given.

Aside from these three loss terms, adversarial loss is complementary. Training the network with GANs has recently been substantially improving the perceptual quality of generated images [16, 18, 19, 24]. GAN actually is a minimax two-player game [36]. A generator G can capture the data distribution, while the discriminator D keeps distinguishing whether the sample is from the training dataset or not. Under this powerful mechanism, GANs can generate more visually appealing images without supervised information. However, the original GAN is hard to train due to its internal instability. Wasserstein GANs (WGANs) [37] and improved training of Wasserstein GANs (WGAN-GP) [38] are two alternative solutions. The former uses weight clipping to enforce D to lie within the space of 1-Lipschitz and the latter uses gradient penalty to encourage D to learn smoother decision boundaries. Since gradient penalty has been proven to be easier and faster during the process of training, WGAN-GP is chosen to define the adversarial loss.

Experimental results from different research groups indicate that setting a well-designed combination of the aforementioned loss terms as the loss function can facilitate models to generate realistic results. Therefore, in our SISR approach, GMGAN, an elaborate loss function consisting of weighted loss terms, is designed as the optimization objective (refer to Section 3 for details).

2.3. Our Motivations and Contributions. Through the literature survey, we find that in the field of learning-based SISR, we need to continue to devote efforts to at least three aspects.

First, learning-based SISR methods conventionally use MSE loss to formulate the whole or part of the loss function. However, this per-pixel loss is still stuck at comparing two images coarsely, generating overly smooth results, which are unnatural and unrealistic.

Second, since GAN is a powerful mechanism for models to learn more realistic results, it is prevalent to combine adversarial learning with SISR. However, the original GAN is formidable for training due to its internal instability and tends to suffer from mode collapse.

Third, extrinsic factors do affect the final results, e.g., training dataset. However, these factors might have been ignored in previous works since most attention is attached to network architecture, algorithm, etc.

In this work, we attempt to fill the aforementioned research gaps to some extent. Our major contributions are as follows:

(1) We propose a novel method named gradient map generative adversarial network (GMGAN) to optimize the improved generative model in gradient map space. Experimental results show that GMGAN performs better in the aspect of visual quality.

(2) In GMGAN, we design a quality loss by integrating an IQA metric named gradient magnitude similarity deviation (GMSD) [39]. We chose GMSD from IQA metrics for its meaningful derivative and capability of predicting perceptual quality consistently with human subjective evaluation. In addition, to overcome the flaw of training instability, the original GAN is replaced with WGAN-GP. With gradient penalty, the discriminator is encouraged to learn smoother decision boundaries to train the generator more smoothly and generate better results.

(3) To analyze the effect of different training datasets, we did sufficient experiments on different training datasets. Experimental results show that large quantity of high-quality training images with rich textures is beneficial for the final results.

3. GAN-Based Image Super-Resolution with a Novel Quality Loss

In this section, our proposed approach GMGAN, which integrates the merits of an image quality assessment based (IQA-based) loss function and improved adversarial training, will be presented in detail. GMGAN consists of two major parts: network architecture and loss function. We present GMGAN architecture first and then discuss loss terms thoroughly. In particular, we design a quality loss for the loss function, which is the first truly IQA-based loss term.

3.1. Network Architecture. The architecture of GMGAN is inspired by SRGAN [24], and hence the structure of the discriminator D remains the same as SRGAN. However, to further improve the perceptual quality of reconstructed images, three modifications are made to the generator G.

(1) In order to take advantage of dense connections and multilevel residual network, the original residual blocks are replaced with residual-in-residual dense block (RRDB) blocks [40].

(2) As mentioned in Section 2.2, the original GAN suffers from training instability. To overcome this flaw, the original GAN is replaced with WGAN-GP [38] to balance the training of G and to help generate more realistic results.

(3) To reduce memory usage and computational complexity, batch normalization (BN) layers are removed as suggested in EDSR [26]. The entire network architecture of G is depicted in Figure 2.

As shown in Figure 2, the high-level architecture design of SRGAN [24] is retained and residual blocks in the low-level architecture are replaced with RRDB blocks. Each RRDB block employs a more complex structure than the original residual block. More concretely, residual learning is implemented at different levels, resulting in a residual-in-residual structure. As introduced in Section 2.1.1, dense connection can facilitate feature exploration. In each dense block, by building short paths between a layer and every other layer, the flow of information through the block can be strengthened. In our setting, 23 RRBD blocks are used.

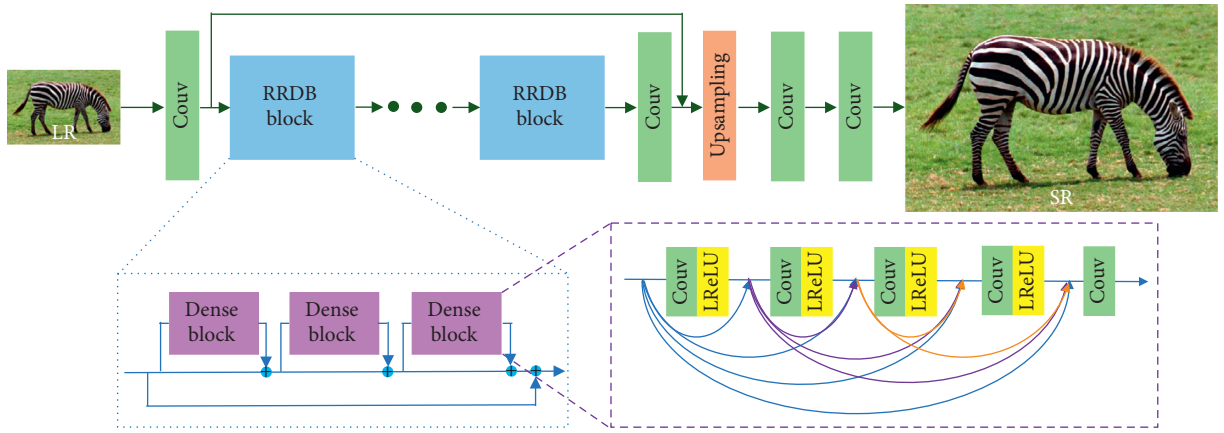

FIGURE 2: The upper is the architecture of the generator G, which employs the basic architecture of SRResNet [24]. Here, the original residual block is replaced with RRDB block. The lower left depicts the internal structure of each RRDB block and the lower right presents the internal structure of each dense block. Notably, the dense block of RRDB is comprised of convolutions and ReLUs without BN layers.

Besides improving the architecture of G, BN layers are removed. Removing BN layers can bring many benefits. First, BN layers would normalize the features to get rid of range flexibility from networks. Second, BN layers consume relatively large computation and GPU memory usage. Hence, removing BN layers can improve the capability of the network, especially under limited computational resources. Third, unpleasant artifacts may be brought due to BN layers when the network is relatively deep and trained with adversarial learning [26]. For the above reasons, BN layers are removed.

3.2. Loss Function. Loss function is the optimization objective for learning-based SISR methods. In this section, all the loss terms of the loss function are listed in detail. The integrant factors are MSE loss l_{MSE}, perceptual loss l_{P}, quality loss l_{Q}, adversarial loss for the generator l_{GA}, and adversarial loss for the discriminator l_{DA}, respectively. In general, the loss function for the generator G is

$$l_{\mathrm{G}} = \alpha l_{\mathrm{MSE}} + \beta l_{\mathrm{P}} + \gamma l_{\mathrm{Q}} + \delta l_{\mathrm{GA}}. \tag{1}$$

The loss function for the discriminator D is

$$l_{\mathrm{D}} = \delta l_{\mathrm{DA}}, \tag{2}$$

where α, β, γ, and δ are the weights for each loss term.

3.2.1. MSE Loss in Image Space. As the most common optimization objective for SISR, the pixelwise MSE loss is calculated as

$$l_{\mathrm{MSE}} = \left\| G_\theta(I_{\mathrm{LR}}) - I_{\mathrm{HR}} \right\|_2^2, \tag{3}$$

where the parameter of the generator is denoted by θ; the generated image, namely, I_{SR}, is denoted by $G_\theta(I_{\mathrm{LR}})$; and the ground truth is denoted by I_{HR}. Although models with MSE loss favor a high PSNR value, the generated results tend to be perceptually unsatisfying with overly smooth textures. Despite the aforementioned shortcomings, this loss term is still kept because MSE has clear physical meaning and helps to maintain color stability.

3.2.2. Perceptual Loss in Feature Space. To compensate the shortcomings of MSE loss and allow the loss function to better measure semantic and perceptual differences between images, we define and optimize a perceptual loss based on high-level features extracted from a pretrained network [30, 41]. The rationality of this loss term lies in that the pretrained network for classification originally has learned to encode the semantic and perceptual information that may be measured in the loss function. Johnson et al. used a 16-layer VGG network [30] pretrained on the ImageNet dataset for SISR. To enhance the performance of the perceptual loss, a 19-layer VGG network is used instead. The perceptual loss is actually the Euclidean distance between feature representations, which is defined as

$$l_{\mathrm{P}} = \left\| \varnothing(G_\theta(I_{\mathrm{LR}})) - \varnothing(I_{\mathrm{HR}}) \right\|_2^2, \tag{4}$$

where $\varnothing$ refers to the 19-layer VGG network. With this loss term, I_{SR} and I_{HR} are encouraged to have similar feature representations rather than to exactly match with each other in a pixelwise manner.

3.2.3. Quality Loss in Gradient Map Space. Inspired by an IQA metric named gradient magnitude similarity deviation (GMSD), we proposed a novel loss term named quality loss. In general, there are two reasons why we chose the metric from [39]. Firstly, full-reference IQA can evaluate the quality of the generated image and is reference-based, which is a prerequisite for being used as a loss function. It is widely accepted that MSE, which is equivalent to PSNR, does not correlate well with human's perception of image quality. To address the limitations of the simple MSE loss, we focus on reference-based metrics with good performance. The visual information fidelity (VIF) [42], which is based on the amount of shared information between the reference and distorted images, can measure the visual information

fidelity. The feature similarity index (FSIM) [43] can measure the dissimilarity between two images based on local phase congruency and gradient magnitude. The metric we chose, GMSD, is characterized by its simplicity and can still predict the perceptual image quality consistently with HVS. Secondly, we cherry picked GMSD from these candidates for its capability of balancing the feasibility and efficiency. Despite the fact that VIF and FSIM can even achieve a better quality prediction performance, their formulations are more complicated and are not differentiable, making their applications for optimization in neural networks infeasible. Thus, for those two reasons, we cherry picked GMSD to define the quality loss.

Generally, GMSD is derived by three steps. Calculate the gradient magnitude similarity (GMS) between I_{HR} and I_{SR} first. Then, adopt an average pooling strategy to predict the perceptual image quality, namely, gradient magnitude similarity mean (GMSM). Lastly, to reflect the overall quality of the global variation of local quality map (LQM), calculate the standard deviation of GMS, namely, GMSD. The flowchart of the complete GMSD calculation is given in Figure 3.

More concretely, first convolve h_x and h_y with the generated image I_{SR}, namely, $G_\theta(I_{LR})$, and the ground truth I_{HR}. Here, h_x and h_y are the Prewitt filters along horizontal and vertical directions. The convolution operation yields the horizontal and vertical gradient images of I_{SR} and I_{HR}. The gradient magnitudes of I_{SR} and I_{HR} at location i, denoted as $m_{SR}(i)$ and $m_{HR}(i)$, are calculated as follows:

$$\begin{aligned} m_{SR}(i) &= \sqrt{\left(I_{SR} \otimes h_x\right)^2(i) + \left(I_{SR} \otimes h_y\right)^2(i)}, \\ m_{HR}(i) &= \sqrt{\left(I_{HR} \otimes h_x\right)^2(i) + \left(I_{HR} \otimes h_y\right)^2(i)}, \end{aligned} \tag{5}$$

where symbol "⊗" denotes the convolution operation. Then, the GMS map is calculated as

$$\mathrm{GMS}(i) = \frac{2m_{SR}(i) \cdot m_{HR}(i) + c}{m_{SR}^2(i) + m_{HR}^2(i) + c}, \tag{6}$$

where c is a positive constant to keep the numerical stability. Note now the LQM of I_{SR} has been acquired, reflecting the local quality of each small patch in I_{SR} in a pixelwise manner. To further estimate the overall quality score from the LQM, an average pooling strategy is applied to the GMS map to obtain GMSM:

$$\mathrm{GMSM} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{GMS}(i), \tag{7}$$

where N refers to the total number of pixels in I_{SR}. Because the average pooling strategy cannot reflect how the local quality degradation varies but the global variation of image local quality degradation can, the standard deviation of the GMS map is calculated to be the final IQA metric:

$$\mathrm{GMSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\mathrm{GMS}(i) - \mathrm{GMSM})^2}. \tag{8}$$

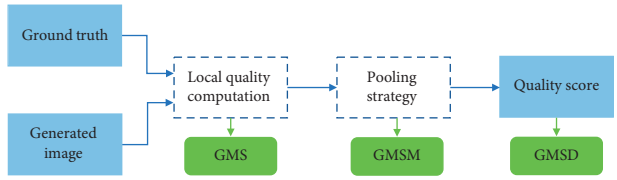

Figure 3: Flowchart of the GMSD calculation.

The final GMSD can reflect the distortion severity of an image, which serves as one of the optimization objectives during the training. Therefore, the quality loss is defined as

$$l_Q = \mathrm{GMSD}\left(G_\theta(I_{LR}), (I_{HR})\right), \tag{9}$$

where GMSD contains the complete calculation of GMSD. Note that a higher GMSD score implies a larger distortion.

In practice, for each pair of the generated image I_{SR} and the ground truth I_{HR}, one GMSD value is calculated from one channel (R, G, or B). After that, the three GMSD values are summed together as the ultimate quality loss.

3.2.4. Adversarial Loss. In SRGAN [24], the adopted generative model is generative adversarial network (GAN) [36] and it suffers from training instability. WGAN [37] leverages the Wasserstein distance to produce a value function, which has better theoretical properties than the original GAN. However, WGAN requires that the discriminator must lie within the space of 1-Lipschitz through weight clipping, resulting in either vanishing or exploding gradients without careful tuning of the clipping threshold. To overcome the flaw of weight clipping, an alternative approach named gradient penalty from WGAN-GP [38] is taken to enforce the Lipschitz constraint. Compared with WGAN, WGAN-GP still measures the Wasserstein distance between two distributions to help decide when to stop the training but penalizes the gradient of the discriminator with respect to its input instead of weight clipping. With gradient penalty, the discriminator is encouraged to learn smoother decision boundaries. Moreover, the training phase can be accelerated, and the quality of generated images can also be improved.

In our setting, the adversarial loss for the generator G is defined as

$$I_{GA} = -\mathbb{E}\left(D\left(G_\theta(I_{LR})\right)\right). \tag{10}$$

The adversarial loss for the discriminator D is defined as

$$I_{DA} = \mathbb{E}\left[D\left(G_\theta(I_{LR})\right)\right] - \mathbb{E}\left[D(I_{HR})\right] + \lambda\mathbb{E}\left(\left\|\nabla_{\widehat{I}} D(\widehat{I}) - 1\right\|_2 - 1\right)^2, \tag{11}$$

where λ refers to the penalty coefficient and $\widehat{I}$ samples between each pair of images from respective ground truth I_{HR} and generated image $G_\theta(I_{LR})$.

4. Experimental Results

4.1. Training Details. All the low-resolution images were downsampled with a scale factor 4x by bicubic interpolation

from high-resolution ones, using the MATLAB bicubic kernel function.

As mentioned in EDSR [26] and ESRGAN [40], to accelerate the speed of convergence and help the discriminator to work more effectively at the beginning of the training, a pretrained model was used. In the pretraining phase, GMGAN was PSNR oriented and its loss function only consisted of the MSE loss. The learning rate was initialized as 2×10^{-4} and halved at every 2×10^5 minibatch updates. In the formal training phase, GMGAN was trained with Adam optimizer [44] by setting $\beta_1 = 0$ and $\beta_2 = 0.9$. The gradient penalty coefficient λ was set to 10. The minibatch was set to 16. The learning rate was initialized as 2×10^{-4} and halved at every 1×10^5 minibatch updates. Our model was trained with the loss function in equations (1) and (2), where $\alpha = 1 \times 10^{-2}$, $\beta = 1$, $\gamma = 1$, and $\delta = 5 \times 10^{-3}$. During training, the generator and discriminator were updated alternatively until GMGAN converged finally.

All the methods involved in our experiments were implemented with PyTorch. The experiments were conducted on an Ubuntu workstation with an NVIDIA Titan Xp GPU.

4.2. Datasets. For training, to analyze the effect of different training datasets, four kinds of datasets were chosen:

(1) Part of ImageNet [1] (short for ImageNet for convenience), consisting of 10000 relatively low-quality images. Note that ImageNet dataset is a benchmark for object category classification and detection on hundreds of categories originally, and thus this dataset is characterized by large number but low quality of contained images.

(2) DF2K dataset, consisting of 3450 relatively high-quality images in total. Here, DF2K is short for DIVIK [45] + Flickr2K [46], which contains 2650 and 800 relatively high-quality images, respectively. Note that DIVIK dataset is a professional dataset for image restoration tasks.

(3) DF2K + OST dataset, consisting of 3902 images totally. Here, 2650 relatively high-quality images are from DF2K dataset and 1342 images of rich textures are from OutdoorSceneTraining (OST) [19] dataset.

(4) DIVIK [45] augmentation (short for augmentation for convenience) dataset, consisting of 20480 relatively high-quality images flipped or rotated from DIVIK dataset. As suggested in [47], to enhance the final performance, we rotated each original image from DIVIK dataset by 90°, 180°, and 270° and then flipped them vertically to get 8 corresponding images including identity without altered content (refer to Figure 4).

For evaluation, experiments were conducted on four public benchmark datasets: Set5 [48], Set14 [49], BSD100 [50], and Urban100 [51]. Set5, Set14, and BSD100 contain natural scenes such as woman, butterfly, and bridge, while Urban100 contains urban scenes with details in different frequency bands such as high-rise buildings, brick wall, and retro church. Because the published results of some state-of-the-art methods do not contain the case of Urban100, Urban100 was adopted only by our proposed GMGAN.

4.3. Benchmark Results. In this section, GMGAN is compared not only with PSNR-oriented methods, including LapSRN [15], EDSR+ [26], and EnhanceNet-E [18], but also with perceptual-driven methods, including SRGAN [24] and EnhanceNet-PAT [18]. Results were all tested on public benchmark datasets, including Set5 [48], Set14 [49], and BSD100 [50].

Traditional IQA criteria like PSNR and SSIM can only reflect part of human perception, and thus we attempt to find an alternative for them: perceptual index (PI). As visual coherence metric [52] is elaborately designed for image inpainting, PI is designed for image super-resolution. PI was firstly introduced in 2018 perceptual image restoration and manipulation (PIRM) challenge, which was a competition for perceptual image super-resolution [53]. The challenge defines perceptual quality as the visual quality of the reconstructed image regardless of its similarity to the ground-truth image. To this end, 2018 PIRM challenge proposed to measure the perceptual quality of the reconstructed image by combining two no-reference image quality metrics: Ma et al. [54] and NIQE [55]. Using this new criterion, the reconstructed result is measured how it looks like a valid natural image without relying on any ground-truth image. PI is defined as

$$\mathrm{PI} = \frac{1}{2}((10 - \mathrm{Ma}) + \mathrm{NIQE}). \tag{12}$$

To meet the requirement of reconstruction accuracy, the full-reference root mean square error (RMSE) distortion was also measured in the challenge, which was the same as PSNR in essence.

Inspired by 2018 PIRM challenge, in our experiments, PI was used as the main criterion, providing PSNR and SSIM as well for reference. Quantitive evaluation results of GMGAN on public benchmark datasets are provided in Table 1. For PSNR and SSIM, a higher score means a better quality. On the contrary, a lower PI score implies a better quality. As summarized in Table 1, in terms of PI, GMGAN performs the best or the second best, which indicates that GMGAN can obtain comparatively the best perceptual quality among these methods.

Besides quantitive evaluation, qualitative results are also provided in Figure 5. Observing from these four representatives, GMGAN outperforms previous methods, reflecting in generating more natural textures and suffering from less undesired artifacts. For example, GMGAN can generate more realistic and sharper ship's outline (refer to 219090 from BSD100) than the PSNR-oriented approaches such as EDSR+ and EnhanceNet-E, whose textures are inevitably overly smooth. Besides, GMGAN is able to generate more natural elephant nose (refer to 296059 from BSD100) than the perceptual-driven approaches such as SRGAN and EnhanceNet-PAT, whose

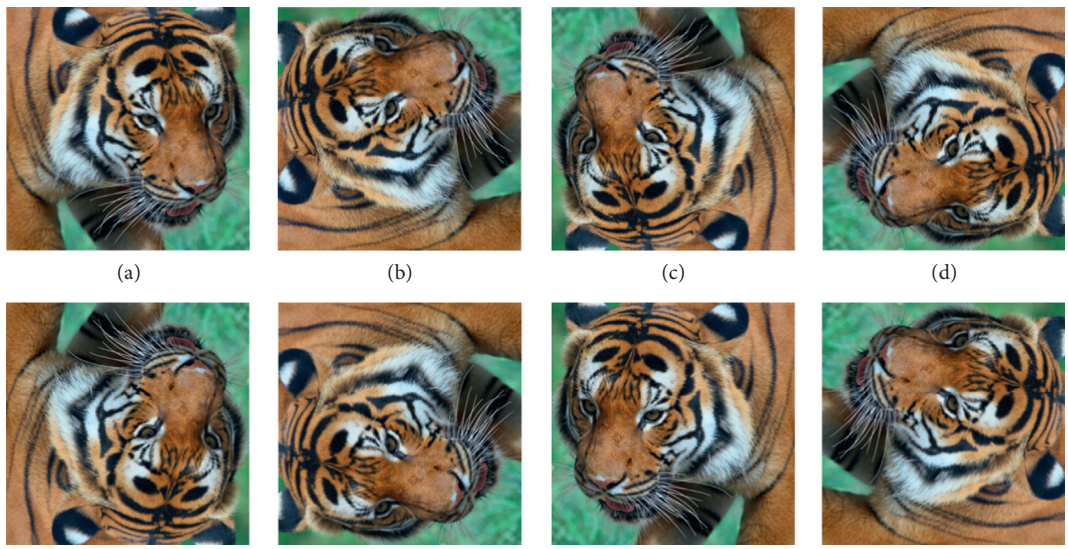

Figure 4: Augmentation of DIVIK dataset by rotation and flipping. (a) Original. (b) Rotated 90°. (c) Rotated 180°. (d) Rotated 270°. (e) Flipped. (f) 90° and flipped. (g) 180° and flipped. (h) 270° and flipped.

Table 1: Public benchmark test results (PI/SSIM/PSNR(dB)).

Datasets	Set5	Set14	BSD100
Bicubic	7.33/0.8111/28.42	6.97/0.7163/26.10	6.94/0.6681/25.96
LapSRN	6.48/0.8866/31.54	5.96/0.7720/28.19	5.81/0.7264/27.32
EDSR+	5.99/0.9003/32.62	5.50/0.7903/28.94	5.39/0.7439/27.79
EnhanceNet-E	6.05/0.8889/31.74	5.25/0.7774/28.42	5.49/0.7324/27.50
EnhanceNet-PAT	**2.93**/0.8103/28.56	3.02/0.6782/25.77	2.91/0.6267/24.93
SRGAN	3.35/0.8356/32.05	*2.88*/0.6958/28.49	*2.35*/0.6426/27.58
GMGAN	*3.25*/0.8447/30.02	**2.77**/0.7055/26.37	**2.29**/0.6592/25.46

Bold indicates the best and italics indicate the second best performance in terms of PI. PSNR and SSIM are provided for reference. All comparison results are acquired from published papers. Notably, GMGAN is trained on augmentation dataset.

textures are relatively unnatural. In addition, GMGAN can generate results more like valid natural images. For example, the wing's texture generated by GMGAN is in the same direction, while by other methods tends to be in mixed directions, which contradicts the real scene (refer to 8023 from BSD100). Moreover, GMGAN suffers from less unpleasant artifacts. Refering to 101087 from BSD100, the hula skirt's texture generated by GMGAN is clear without undesired artifacts, while other approaches, such as EnhanceNet-PAT, tend to introduce artifacts.

In order to examine the contribution of the quality loss, we compared SRGAN, GMGAN ($l_{\mathrm{MSE}} + l_{\mathrm{P}} + l_{\mathrm{GA}}$), and GMGAN on Set5, Set14, and BSD100, and the results are summarized in Table 2. In Table 2, GMGAN denotes the proposed model that considers all loss terms described in Section 3.2 including quality loss, while GMGAN ($l_{\mathrm{MSE}} + l_{\mathrm{P}} + l_{\mathrm{GA}}$) denotes the variant that only contains MSE loss, perceptual loss, and adversarial loss. In both of these two cases, the architecture has been improved based on SRGAN. It is observed that the performance of GMGAN is the best in terms of PI among all competitors. By comparing GMGAN with GMGAN ($l_{\mathrm{MSE}} + l_{\mathrm{P}} + l_{\mathrm{GA}}$), the effectiveness of the quality loss can be corroborated.

4.4. Effect of Different Training Datasets. External factors (e.g., training dataset) tend to be ignored in learning-based SISR approaches. In fact, they are comparatively important as internal factors. To analyze how different training datasets affect final results, we did sufficient experiments on different training datasets, which have been introduced in Section 4.2.

Qualitative and quantitive results of different training datasets are presented in Figure 6 and Table 3, respectively. As observed in Figure 6, three factors beneficial for the final results can be inferred:

(1) An extended dataset with equal image quality can increase the final performance. For instance, augmentation dataset outperforms DF2K dataset. For example, the hay's outline of 175032 is clearer when it is dealt with the model trained on augmentation.

(2) Priority should be given to image quality over quantity. For instance, DF2K dataset outperforms ImageNet dataset. For example, the branch's texture of bridge is more natural when it is restored by the model trained on DF2K.

(3) Dataset with richer and diverse textures is beneficial for generating more realistic results. For instance, the

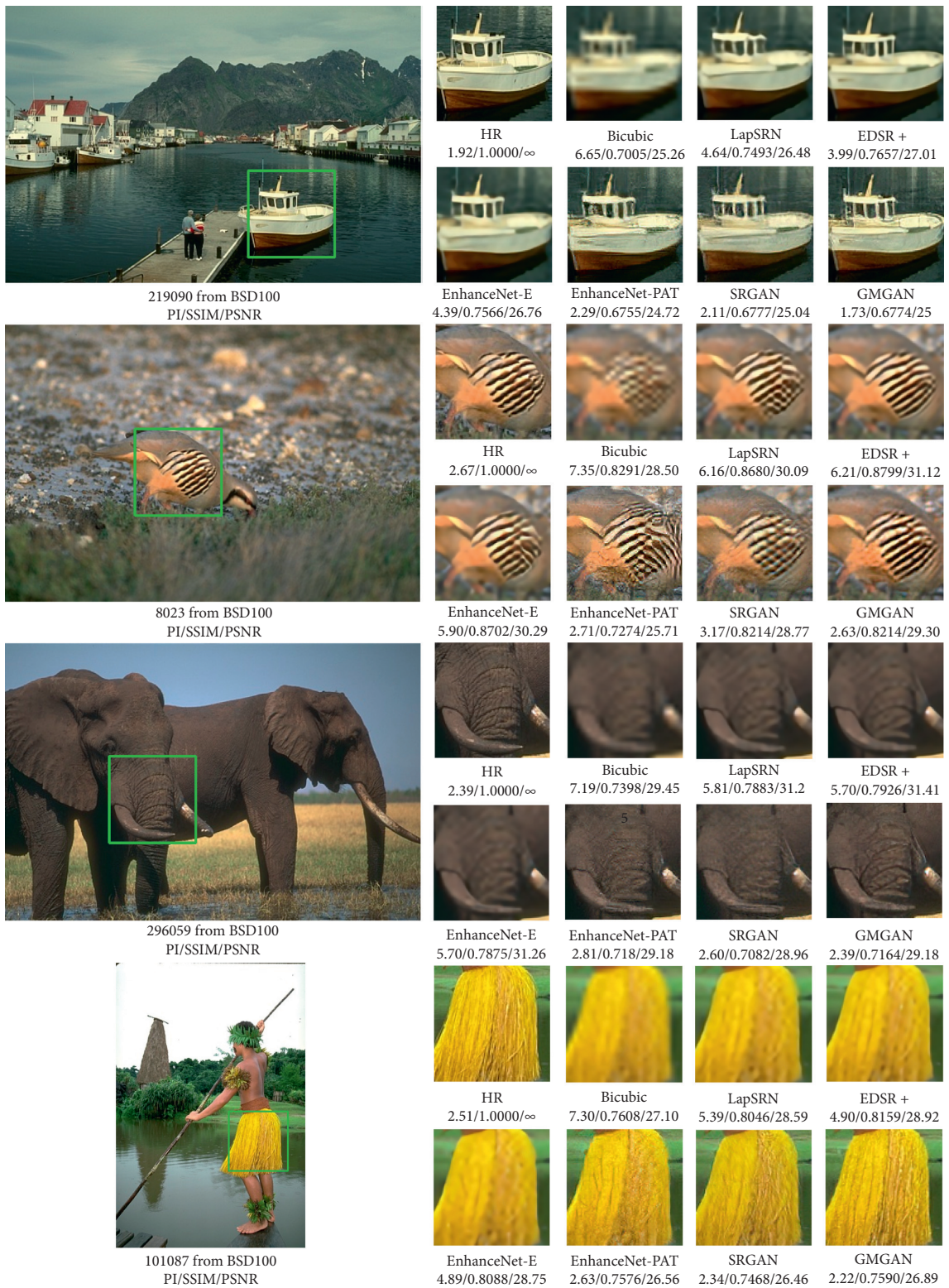

Figure 5: Visual comparisons for 4x SR on samples from BSD100 (PI/SSIM/PSNR(dB)). In general, GMGAN performs well not only in PI but also in visual effects.

Table 2: Performances of SRGAN, GMGAN ($l_{MSE} + l_P + l_{GA}$), and GMGAN (PI/SSIM/PSNR(dB)).

Datasets	Set5	Set14	BSD100
SRGAN	3.35/0.8356/32.05	2.88/0.6958/28.49	2.35/0.6426/27.58
GMGAN ($l_{MSE} + l_P + l_{GA}$)	*3.30*/0.8491/31.41	*2.87*/0.7113/26.40	*2.31*/0.6651/25.51
GMGAN	**3.25**/0.8447/30.02	**2.77**/0.7055/26.37	**2.29**/0.6592/25.46

Bold indicates the best and italics indicate the second best performance in terms of PI. PSNR and SSIM are provided for reference.

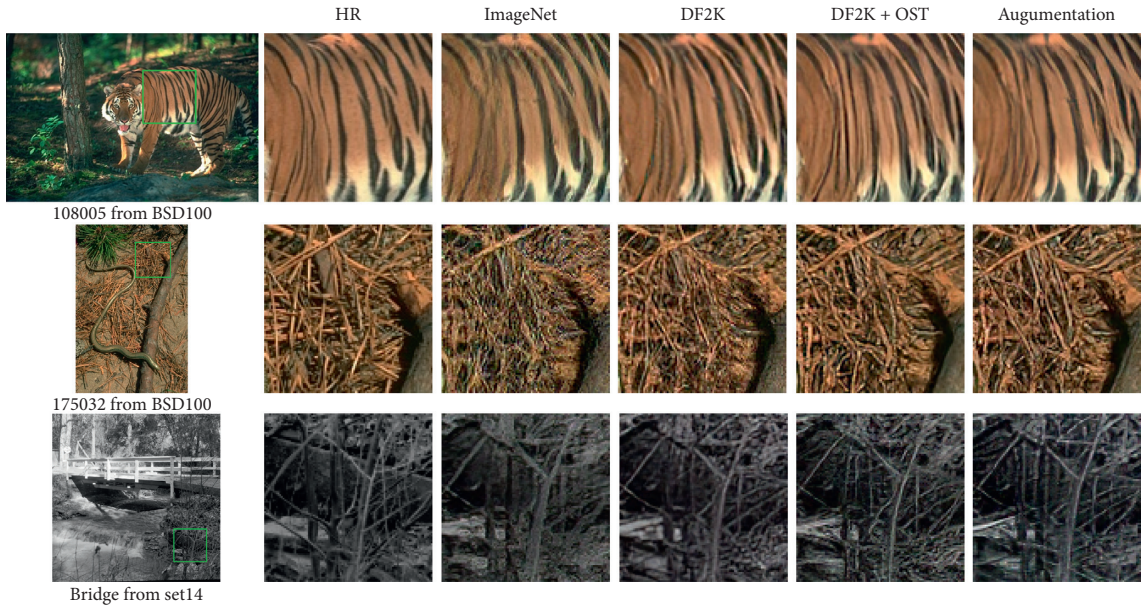

Figure 6: Visual comparisons on four different training datasets.

Table 3: Test results (PI/SSIM/PSNR(dB)) when using four different training datasets.

Datasets	Set5	Set14	BSD100
ImageNet	3.85/0.8510/30.12	2.97/0.6898/26.51	2.40/0.6470/25.31
DF2K	3.70/0.8447/30.40	*2.87*/0.7055/25.96	*2.34*/0.6601/25.56
DF2K + OST	*3.43*/0.8497/30.31	2.96/0.7015/26.60	2.37/0.6590/25.70
Augmentation	**3.25**/0.8342/30.02	**2.77**/0.7021/26.37	**2.29**/0.6593/25.46

Bold indicates the best and italics indicate the second best performance in terms of PI. PSNR and SSIM are provided for reference.

tiger's texture of 108005 is the most realistic when it is restored by the model trained on DF2K + OST. We tested the models trained on four different datasets on Set5, Set14, and BSD100 in terms of PI, SSIM, and PSNR, and the results are presented in Table 3. Similar inferences can be drawn from Table 3. Note that although adding images with rich textures to the training dataset can facilitate generating more richly textured results, it may damage the performance of PI. We blame it to the relatively low quality of OST dataset.

4.5. Failure Case of GMGAN. GMGAN currently is not perfect. When dealing with images with strong repetitive structures, GMGAN cannot yield satisfactory results. Figure 7 presents a failure case, which was tested on img_092 from Urban100 dataset, compared with its corresponding I_{HR}. The reconstructed lines of each block should be perpendicular to the lines of its surrounding blocks. However, our result yields diagonal lines in part of blocks, especially in smaller blocks. It may be due to the imperfection of the network architecture. In the future, we may adopt more appropriate network architectures (e.g., DBPN [28]) for a sequence of iterative up- and down-projection units, providing an error feedback mechanism for projection errors to guide each back-projection stage, to solve this issue.

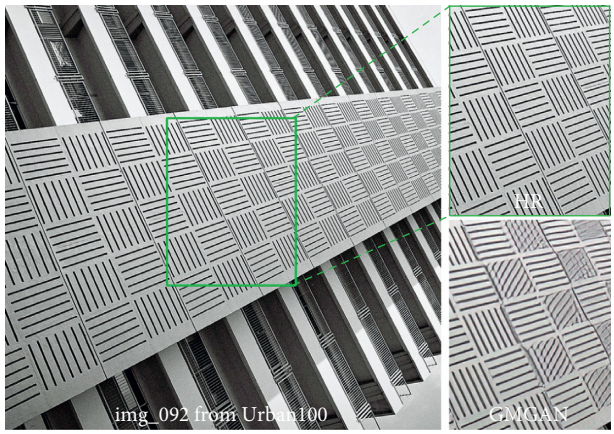

Figure 7: A failure case of GMGAN.

5. Conclusion and Future Work

Reconstructing visually appealing super-resolution image has been a crucial issue in SISR. To tackle it, in this paper, we propose a method named GMGAN, integrating an IQA-based loss function and improved adversarial training. Besides, we also analyze the effect of different training datasets. Extensive experiments indicate that GMGAN can generate more photorealistic results with less unpleasant artifacts. Moreover, large quantity of high-quality training

images with rich textures can benefit the final results. In the future, we would continue to improve the architecture of the network to make GMGAN capable of tackling images with strong repetitive structures.

Data Availability

The data used in this work are all publicly available on the Internet.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under grant nos. 61973235, 61672380, and 61936014 and in part by the Natural Science Foundation of Shanghai under grant no. 19ZR1461300.

References

[1] O. Russakovsky, J. Deng, H. Su et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[2] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 3, pp. 231–239, 1991.

[3] C. E. Duchon, "Lanczos filtering in one and two dimensions," *Journal of Applied Meteorology*, vol. 18, no. 8, pp. 1016–1022, 1979.

[4] R. Fattal, "Image upsampling via imposed edge statistics," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 95, 2007.

[5] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, vol. 10, no. 10, pp. 1521–1527, 2001.

[6] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Transactions on Graphics*, vol. 30, pp. 1–11, 2011.

[7] Q. Shan, Z. Li, J. Jia, and C. K. Tang, "Fast image/video upsampling," *ACM Transactions on Graphics*, vol. 27, no. 5, pp. 1–7, 2008.

[8] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1127–1133, 2010.

[9] Z. Xiong, X. Sun, and F. Wu, "Robust web image/video super-resolution," *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 2017–2028, 2010.

[10] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1838–1857, 2011.

[11] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[12] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision*, pp. 349–356, IEEE, Kyoto, Japan, September 2009.

[13] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proceedings of the 2014 European Conference on Computer Vision*, pp. 184–199, Zurich, Switzerland, September 2014.

[14] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proceedings of the 2016 European Conference on Computer Vision*, pp. 391–407, Amsterdam, Netherlands, October 2016.

[15] W. S. Lai, J. B. Huang, N. Ahuja, and M. H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5835–5843, IEEE, Honolulu, HI, USA, July 2017.

[16] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers, "A fully progressive approach to single-image super-resolution," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 864–873, IEEE, Salt Lake City, UT, USA, June 2018.

[17] A. Shocher, N. Cohen, and M. Irani, ""Zero-shot" super-resolution using deep internal learning," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3118–3126, IEEE, Salt Lake City, UT, USA, June 2018.

[18] M. S. M. Sajjadi, B. Schölkopf, and M. Hirsch, "Enhancenet: single image super-resolution through automated texture synthesis," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4501–4510, IEEE, Venice, Italy, October 2017.

[19] X. Wang, K. Yu, C. Dong, and C. C. Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 606–615, IEEE, Salt Lake City, UT, USA, June 2018.

[20] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1646–1654, IEEE, Las Vegas, NV, USA, June 2016.

[21] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1637–1645, IEEE, Las Vegas, NV, USA, June 2016.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, IEEE, Las Vegas, NV, USA, June 2016.

[23] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, IEEE, Honolulu, HI, USA, July 2017.

[24] C. Ledig, L. Theis, F. Huszár et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 105–114, IEEE, Honolulu, HI, USA, July 2017.

[25] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4799–4807, IEEE, Venice, Italy, October 2017.

[26] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in

Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1132–1140, IEEE, Honolulu, HI, USA, July 2017.

[27] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2472–2481, IEEE, Salt Lake City, UT, USA, June 2018.

[28] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1664–1673, IEEE, Salt Lake City, UT, USA, June 2018.

[29] Z. Wang and A. C. Bovik, "Mean squared error: love it or leave it? a new look at signal fidelity measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009.

[30] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proceedings of the 2016 European Conference on Computer Vision*, pp. 694–711, Amsterdam, Netherlands, October 2016.

[31] J. Bruna, P. Sprechmann, and Y. LeCun, "Super-resolution with deep convolutional sufficient statistics," 2015, https://arxiv.org/abs/1511.05666.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[34] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proceedings of the 37th Asilomar Conference on Signals, Systems & Computers 2003*, vol. 2, pp. 1398–1402, IEEE, Pacific Grove, CA, USA, November 2003.

[35] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2016.

[36] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the 2014 Conference on Neural Information Processing Systems*, pp. 2672–2680, Montreal, Canada, December 2014.

[37] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 2017 International Conference on Machine Learning*, pp. 214–223, Ho Chi Minh City, Vietnam, January 2017.

[38] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein gans," in *Proceedings of the 2017 Conference on Neural Information Processing Systems*, pp. 5767–5777, Long Beach, CA, USA, December 2017.

[39] W. Xue, L. Zhang, X. Mou, and A. C. Bovik, "Gradient magnitude similarity deviation: a highly efficient perceptual image quality index," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 684–695, 2014.

[40] X. Wang, K. Yu, S. Wu et al., "ESRGAN: enhanced super-resolution generative adversarial networks," in *Proceedings of the 2018 European Conference on Computer Vision*, pp. 63–79, Munich, Germany, September 2018.

[41] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5188–5196, Boston, MA, USA, June 2015.

[42] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 430–444, 2006.

[43] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, 2011.

[44] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[45] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: dataset and study," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1122–1131, IEEE, Honolulu, HI, USA, July 2017.

[46] R. Timofte, E. Agustsson, L. V. Gool et al., "NTIRE 2017 challenge on single image super-resolution: methods and results," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1110–1121, IEEE, Honolulu, HI, USA, July 2017.

[47] R. Timofte, R. Rothe, and L. V. Gool, "Seven ways to improve example-based single image super resolution," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1865–1873, IEEE, Las Vegas, NV, USA, June 2016.

[48] M. Bevilacqua, A. Roumy, C. Guillemot, and M. A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proceedings of the 2012 British Machine Vision Conference*, pp. 135:1–135:10, Surrey, UK, September 2012.

[49] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proceedings of the 2010 International Conference on Curves and Surfaces*, pp. 711–730, Avignon, France, June 2010.

[50] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the 8th IEEE International Conference on Computer Vision*, pp. 416–423, IEEE, Vancouver, Canada, July 2001.

[51] J. B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5197–5206, IEEE, Boston, MA, USA, June 2015.

[52] T. T. Dang, A. Beghdadi, and M. C. Larabi, "Visual coherence metric for evaluation of color image restoration," in *Proceedings of the 2013 Colour and Visual Computing Symposium (CVCS)*, pp. 1–6, IEEE, Gjovik, Norway, September 2013.

[53] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor, "The 2018 PIRM challenge on perceptual image super-resolution," in *Proceedings of the 2018 European Conference on Computer Vision*, pp. 334–355, Munich, Germany, September 2018.

[54] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, "Learning a no-reference quality metric for single-image super-resolution," *Computer Vision and Image Understanding*, vol. 158, pp. 1–16, 2017.

[55] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2013.