2022 18th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) | 978-1-6654-6382-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/AVSS56176.2022.9959415

Real Image Super-Resolution using GAN through modeling of LR and HR process

Rao Muhammad Umer,
Institute of AI for Health (AIH),
Helmholtz Munich, Germany.
engr.raoumer943@gmail.com

Christian Micheloni,
Department of Mathematics and Computer Science,
University of Udine, Italy.
christian.micheloni@uniud.it

Abstract

The current existing deep image super-resolution methods usually assume that a Low Resolution (LR) image is bicubicly downscaled of a High Resolution (HR) image. However, such an ideal bicubic downsampling process is different from the real LR degradations, which usually come from complicated combinations of different degradation processes, such as camera blur, sensor noise, sharpening artifacts, JPEG compression, and further image editing, and several times image transmission over the internet and unpredictable noises. It leads to the highly ill-posed nature of the inverse upscaling problem. To address these issues, we propose a GAN-based SR approach with learnable adaptive sinusoidal nonlinearities incorporated in LR and SR models by directly learn degradation distributions and then synthesize paired LR/HR training data to train the generalized SR model to real image degradations. We demonstrate the effectiveness of our proposed approach in quantitative and qualitative experiments.

1. Introduction

Single image super-resolution (SISR) aims to restore the high-resolution (HR) image from its low-resolution (LR) image counterpart. SISR problem is a fundamental low-level vision and image processing problem with various practical applications in *e.g.*, satellite imaging, medical imaging, astronomy, remote sensing, surveillance, image compression, environment and climate change monitoring, mobile photography, image / video enhancement, and security and surveillance imaging, etc. With the increasing amount of HR images / videos data on the internet, there is a great demand for storing, transferring, and sharing such large sized data with low cost of storage and bandwidth resources. Moreover, the HR images are usually downscaled to easily fit into display screens with different resolution while retaining visually plausible information. The downscaled LR counterpart of the HR can efficiently utilize lower bandwidth, storage save, and easily fit to various digital displays. However, some details are lost and sometimes visible artifacts appear when users downscale and upscale the digital contents.

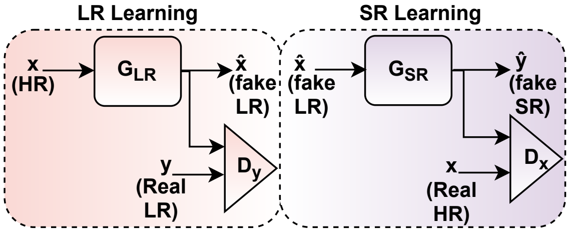

Figure 1: The structure of our proposed real-world SR approach setup. In the LR Learning part, we train the LR generator network $\mathbf{G}_{LR}$ in a GAN framework, where our goal is to learn the real LR $(\mathbf{y})$ corruptions/degradations. Then, we use the synthesized paired LR/HR data by the $\mathbf{G}_{LR}$ model to train the generalized SR model $\mathbf{G}_{SR}$ in the SR Learning part. Both the $\mathbf{G}_{LR}$ and $\mathbf{G}_{SR}$ generators utilize the modified residual structure (refer to the sections 4 and 5 for more details).

Mathematically, SISR is described as a linear forward observation model [19, 21] with the following image degradation process:

$$\mathbf{y} = (\mathbf{H} \otimes \tilde{\mathbf{x}}) \downarrow_s + \eta, \tag{1}$$

where, $\mathbf{y}$ is an observed LR image, $\mathbf{H}$ is a *down-sampling operator* (unknown) that convolves $(\otimes)$ with a latent HR image $\tilde{\mathbf{x}}$ and resizes it by a scaling factor s, and η is considered as an i.i.d additive white Gaussian noise (AWGN) of variance σ^2, *i.e.*, $\eta \sim \mathcal{N}\left(0, \sigma^2\right)$. However, in real-world settings, η also accounts for all possible errors during the image acquisition process that include inherent sensor noise, stochastic noise, compression artifacts, and the possible mismatch between the forward observation model and the camera device. The operator $\mathbf{H}$ is usually ill-

978-1-6654-6382-9/22/$31.00 ©2022 IEEE

conditioned or singular due to the presence of unknown noise realization (η) that turn the SISR to a highly ill-posed nature of inverse problems. Since, due to ill-posed nature, there are many possible solutions thus regularization is required to select the most plausible ones.

Recently, numerous works have been addressed towards the task of SISR [7, 14, 32, 33, 30, 13, 34, 24, 20, 18, 35, 12, 5] and real-world SISR [9, 28, 16, 3, 19, 21, 25]. Most of the SISR methods assume usually bicubic downsampling process, which is different from the real LR degradations. The real-world SISR methods try to solve the problem by utilizing data distribution learning using the GAN [4] framework. However, they do not generalize well to the real complex degradation, which usually come from the complicated degradation processes, *i.e.*, sensor noise, camera blur, sharping artifacts, JPEG compression, and further image editing, and several times image transmission over the internet. In the most recent works [27, 31], the authors aim to restore general real-world LR images by synthesizing training pairs with a more practical degradation process. As the real-world degradation space is much larger/complex, the synthetic modeling also becomes challenging. Moreover, the generators (*i.e.*, LR/HR) require a more powerful capability to model the complex training data, while the gradients needs to be more accurate for local detail enhancement with some sophisticated nonlinearities inside the network.

In this work, we proposed the GAN-based real image SR approach that solves the problem by modeling the LR/HR process with adaptive sinusoidal activitions (*i.e.*, better represent the complicated signals) and thus synthesize the more realistic paired LR/HR data to train the generalized SR model for the real SR task. The structure of our proposed real-world SR approach setup is shown in Fig. 1. In the LR learning, we train the LR network ($\mathbf{G}_{LR}$) with modified residual structure (*i.e.*, incorporating the sinusodial nonlinearities) in a GAN-framework [4] to generate the realistic LR images as the corruptions/degradations of the real LR images ($\mathbf{y}$). After that, we use the synthesized paired LR/HR data to train the generalized SR model in the SR Learning part. The SR network ($\mathbf{G}_{SR}$) is trained in a GAN-framework [4] with the modified residual structure to super-resolve the LR images.

We evaluate our proposed SR method on the Real-World Super-resolution (RWSR) dataset [17] to show the effectiveness of our approach through the quantitative and qualitative experiments. We summarize our contributions in three fold as:

1. We propose an end-to-end deep SRResCSinGAN for the real-world SR task. Instead of using traditional bicubic downsampling or the existing deep LR degradation methods, we synthesize the paired training data with a more practical image corruptions/degradations by modeling the LR/HR process.
2. By exploiting the sinusoidal non-linearities, we employ the modified residual network structure incorporated in both LR and SR learning stages, which better models the underlying complex signals *i.e.*, real LR and HR process.
3. Our proposed approach achieve better quantitative and visual performance in terms of PSNR/SSIM/LPIPS (refer to Tables 1 and 2).

2. Related Work

2.1. Real World SISR methods

Recently, numerous works [7, 14, 32, 33, 30, 13, 34, 24, 20, 18, 35, 12, 5] have addressed the task of SISR using deep CNNs for their powerful feature representation capabilities. The SISR methods mostly rely on the PSNR-based metric by optimizing the $\mathcal{L}_1/\mathcal{L}_2$ losses with blurry results in a supervised way, while they do not preserve the visual quality with respect to human perception. Moreover, the above-mentioned methods are deeper or wider CNN networks to learn non-linear mapping from LR to HR with the ideal bicubic downsampling, while neglecting the real-world settings.

For the real image SR task, several attempts [9, 28, 16, 3, 19, 21, 25] have done to solve for realistic LR degradation. However, the real SR methods still suffer unpleasant artifacts and challenging for learning fine-grained corruptions/degradations with unpaired data. Our approach takes into account the real-world settings by increasing its applicability in practical scenarios.

2.2. Blind / Non-Blind degradation models

Classical degradation model (refer to Eq. (1)) is mostly used in the blind / non-blind deep SISR methods. The common choice, in the existing SISR degradation models, usually consist of a sequence of blur kernel (*i.e.*, Gaussian/motion), downsampling (*i.e.*, bicubic, bilinear, nearest-neighbor), and noise addition (*i.e.*, AWGN). In the existing deep SISR methods [27, 31], they attempt to explicit model the real-world degradation to super-resolve the real LR images. But, yet the real-world degradations are too complex to be explicitly modeled. Therefore, implicit modeling using GAN framework within the network is a suitable choice to synthesize more practical degradations.

3. Proposed Method

3.1. Problem Formulation

By referencing to the Eq. (1), the recovery of $\mathbf{x}$ from $\mathbf{y}$ mostly relies on the variational approach for combining

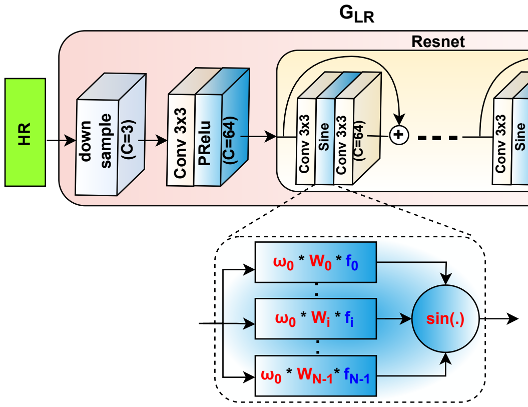

Figure 2: The generator architecture of proposed LR learning stage. The C denotes the output feature channels. Inside the *Sine* layer, ω_0 (hyperparameter) is the scalar frequency factor, $\mathbf{W}$ are the learnable sine weights, and f are the *Conv* layer feature maps (refer to section 3.2 for more details of *Sine* layer).

the observation and prior knowledge, and is given by the following objective function:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2} \|\mathbf{y} - (\mathbf{H} \otimes \mathbf{x}) \downarrow_s \|_2^2 + \lambda \phi(\mathbf{x}), \quad (2)$$

where, $\frac{1}{2\sigma^2}\|\mathbf{y} - (\mathbf{H} \otimes \mathbf{x}) \downarrow_s \|_2^2$ is the data fidelity (also known as log-likelihood) term that measures the closeness of the solution to the observations, $\phi(\mathbf{x})$ is the regularization term that encodes the image prior knowledge, and λ is the trade-off parameter that governs the compromise between the data fidelity and the regularizer term. Interestingly, the variational approach has a direct link to the Bayesian approach and the derived solutions can be described either as penalized maximum likelihood or as maximum a posteriori (MAP) estimates [1, 2]. Thanks to the recent advances of deep learning, the regularizer (*i.e.*, $\phi(\mathbf{x})$) is employed by the SRResCGAN [19] generator structure that is inspired by a powerful image regularization and large-scale optimization techniques to solve the real-world SISR task.

3.2. Residual Network (Resnet) with adaptive Sinusoidal non-linearities

Over the past decades, numerous works have investigated a variety of possible activation functions, such as sigmoid, ReLU, Tanh, PReLU, RBF, and many more to model the natural images. The preferred choice that has emerged over the years is the ReLU activation unit due to promoting sparsity of the feature maps and the faster training of very deep networks. The continuous and piecewise linear functions have proven as a universal approximation of complex signals such as natural images. Recent works have demonstrated the potential to robustly outperform ReLU and other non-linearities by using alternative activation functions for image reconstruction / restoration tasks, such as deep spline activations [26] and periodic nonlinearities like sinusoidal [22]. Motivated by the continuous and differentiable periodic nonlinearities (*i.e.*, sinusoidal) that are capable of representing complex and fine details of signals better than the ReLU and others, we exploit the sinusoidal nonlinearities incorporated in the modified structure of deep residual network (*Resnet*).

We describe the overall explicit compositional structure of the L-layer deep residual network (*Resnet*) with the following formulation:

$$\mathbf{f}_{resnet}(\mathbf{x}) = \left(\left(f_L \circ \sigma_L \circ f_{L-1} \right) (\mathbf{x}_{L-1}) + \mathbf{x}_{L-1} \right) \circ \cdots \circ \\ \left(\left(f_2 \circ \sigma_1 \circ f_1 \right) (\mathbf{x}) + \mathbf{x} \right), \quad (3)$$

Here, f is the affine transformation (*i.e.*, *Conv* layer) defined by the weight matrix $\mathbf{W}$ and the biases $\mathbf{b}$ applied to the input as:

$$f(\mathbf{x}) = \mathbf{W} * \mathbf{x} + \mathbf{b} \quad (4)$$

And, followed by the sine nonlinearity [22] σ applied to the resulting vector f as:

$$\sigma(f) = sin\left(\omega_0 . \mathbf{W} f\right) \quad (5)$$

where, ω_0 is the scalar frequency factor, which is a hyperparameter. The derivative of the sine is a cosine (*i.e.*, the phase-shifted sine) for the backpropagation. The weights of the *Sine* layer are updated during the training via the stochastic gradient descent steps by minimization of the loss function. To initialize the weights ($\mathbf{W}$) of the *Sine* layer, we use the same initialization technique as done in [22], where we draw the weights with $\mathbf{W}_i \sim \mathcal{U}(-\sqrt{6/n}, \sqrt{6/n})$ which ensures that the input to each sine activation is normal distributed with a unit standard deviation.

4. LR Learning Model

In the LR learning phase, we train the model ($\mathbf{G}_{LR}$) in a GAN framework as shown in Fig. 2. In the next sections

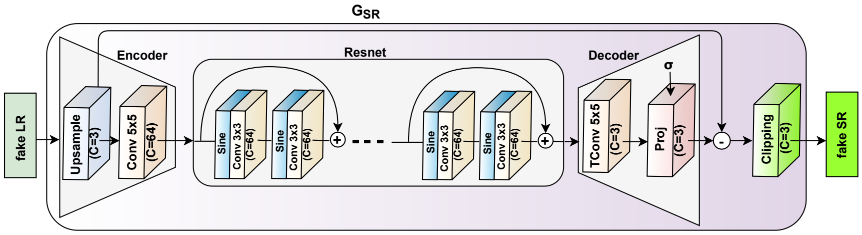

Figure 3: The generator architecture of proposed SR learning stage. The C denotes the output feature channels. The *Sine* layer denotes sinusoidal nonlinearities (refer to Fig. 2 and section 3.2 for more details). The σ is the trainable projection layer parameter.

4.1, 4.2, and 4.3, we describe the network architectures, training losses, and other training details.

4.1. Network Architectures

The modified LR generator network ($\mathbf{G}_{LR}$) (as shown in Fig. 2) consists of 8 *Resnet* blocks. Each residual block contains the *Sine* (*i.e.*, sinusoidal nonlinearities) layer that is sandwiched between two *Conv* layers. All *Conv* layers have 3×3 kernel support with 64 feature maps. Finally, *sigmoid* nonlinearity is applied on the output of the $\mathbf{G}_{LR}$ network. While, the LR discriminator network ($\mathbf{D}_{\mathbf{y}}$) consists of a three-layer convolutional network that operates on a patch level [6, 11]. All *Conv* layers have 5×5 kernel support with feature maps from 64 to 256 and also applied Batch Normalization and Leaky ReLU (LReLU) activations after each *Conv* layer except the last *Conv* layer that maps 256 to 1 features. It is trained to discriminate the real LR images ($\mathbf{y}$) from the fake LR images ($\hat{\mathbf{x}}$) generated by the $\mathbf{G}_{LR}$.

4.2. Network Losses

To learn the degradation/corruptions from the LR domain ($\mathbf{y}$) images, we train the modified network $\mathbf{G}_{LR}$ in a GAN framework [4] with the following loss functions:

$$\mathcal{L}_{\mathbf{G}_{LR}} = \mathcal{L}_{color} + 0.005 \cdot \mathcal{L}_{tex} + 0.01 \cdot \mathcal{L}_{per} \qquad (6)$$

where, these loss functions are defined as follows:
Color loss ($\mathcal{L}_{color}$): It is basically the $\mathcal{L}_1$ loss which focuses on the low frequencies of the image.

$$\mathcal{L}_{color} = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{w}_{\mathrm{L}} * \mathbf{G}_{LR}(\mathbf{x}^{(i)}) - \mathbf{w}_{\mathrm{L}} * \mathbf{x}^{(i)} \downarrow_s\right\|_1 \quad (7)$$

Here, $\mathbf{w}_{\mathrm{L}}$ is the low-pass filter, N is the mini-batch size, and $\downarrow_s$ is the downscaling factor.
Texture loss ($\mathcal{L}_{tex}$): It focuses on the high frequencies of the image.

$$\mathcal{L}_{tex} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{mean}\left(\log \mathbf{D}_{\mathbf{y}}\left(\mathbf{w}_{\mathrm{H}} * \mathbf{G}_{LR}(\mathbf{x}^{(i)})\right)\right) \quad (8)$$

Here, $\mathbf{w}_{\mathrm{H}}$ is the high-pass filter.
Perceptual loss ($\mathcal{L}_{\mathrm{per}}$): It focuses on the perceptual quality of the output image and is defined as:

$$\mathcal{L}_{\mathrm{per}} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{\mathrm{VGG}} = \frac{1}{N}\sum_{i=1}^{N}\|\phi(\mathbf{G}_{LR}(\mathbf{x}^{(i)})) - \phi(\mathbf{x}^{(i)} \downarrow_s)\|_1 \quad (9)$$

where, ϕ is the feature extracted from the pretrained VGG network as done in DSGAN [3].

4.3. Training description

We train the $\mathbf{G}_{LR}$ network with image patches 512×512, which are bicubically downsampled with MATLAB *imresize* function. We randomly crop the LR domain images ($\mathbf{y}$) by 128×128 as done in [3]. We set the $\omega_0 = 30$ for the *Sine* layer. We train the network for 300 epochs with a batch size of 16 using Adam optimizer [8] with parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ without weight decay for both generator and discriminator to minimize the loss in (6). The learning rate is initially set to 2.10^{-4} for the first 150 epochs and then linearly decayed to zero after the remaining (*i.e.*, 150) epochs as done in [3].

5. SR Learning Model

In the SR learning phase, we train the model ($\mathbf{G}_{SR}$) in a GAN framework as shown in Fig. 3. In the next sections 5.1, 5.2, and 5.3, we describe the network architectures, training losses, and other training details.

5.1. Network Architectures

We use the SR generator $\mathbf{G}_{SR}$ network (as shown in Fig. 3) which is an *Encoder-Resnet-Decoder* like structure as done in SRResCGAN [19] with the modified *Resnet* structure by incorporating the sinusoidal nonlinearities. In the $\mathbf{G}_{\mathbf{SR}}$ network, both *Encoder* and *Decoder* layers have 64 convolutional feature maps of 5×5 kernel size with $C \times H \times W$ tensors, where C is the number of channels of the input image. Inside the *Encoder*, LR image is upsampled by the Bicubic kernel with *Upsample* layer. *Resnet* consists of 5 residual blocks with two Pre-activation *Conv* layers, each of 64 feature maps with kernel support 3×3, and the preactivition is the *Sine* layer with 64 output feature channels. The trainable projection layer [10] inside the *Decoder* computes the proximal map with the estimated noise standard deviation σ and handles the data fidelity and prior terms. The noise realization is estimated in the intermediate *Resnet* that is sandwiched between *Encoder* and *Decoder*. The estimated residual image after *Decoder* is subtracted from the LR input image. Finally, the clipping layer incorporates our prior knowledge about the valid range of image intensities and enforces the pixel values of the reconstructed image to lie in the range $[0, 255]$. The reflection padding is also used before all *Conv* layers to ensure slowly varying changes at the boundaries of the input images.

The SR discriminator network ($\mathbf{D_x}$) is trained to discriminate the real HR images ($\mathbf{x}$) from the fake HR images ($\hat{\mathbf{y}}$) generated by the $\mathbf{G}_{SR}$. The raw discriminator network contains 10 convolutional layers with kernels that support 3×3 and 4×4 of increasing feature maps from 64 to 512 followed by Batch Normalization and leaky ReLU as done in SRGAN [9].

5.2. Network Losses

To learn the image super-resolution for the HR domain ($\mathbf{x}$) images, we train the modified network $\mathbf{G}_{SR}$ in a GAN framework with the following loss functions:

$$\mathcal{L}_{G_{SR}} = \mathcal{L}_{\text{per}} + \mathcal{L}_{\text{GAN}} + \mathcal{L}_{tv} + 10 \cdot \mathcal{L}_1 \tag{10}$$

where, these loss functions are defined as follows:
Perceptual loss ($\mathcal{L}_{\text{per}}$): It focuses on the perceptual quality of the output image and is defined as:

$$\mathcal{L}_{\text{per}} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{\text{VGG}} = \frac{1}{N}\sum_{i=1}^{N} \|\phi(\mathbf{G}_{SR}(\hat{\mathbf{x}}^{(i)})) - \phi(\mathbf{x}^{(i)})\|_1 \tag{11}$$

where, ϕ is the feature extracted from the pretrained VGG-19 network as done in ESRGAN [28].
GAN loss ($\mathcal{L}_{\text{GAN}}$): It focuses on the high frequencies of the output image and it is defined as:

$$\mathbf{D_x}(\mathbf{x}, \hat{\mathbf{y}})(C) = \sigma(C(\mathbf{x}) - \mathbb{E}[C(\hat{\mathbf{y}})]) \tag{12}$$

Here, C is the raw discriminator output and σ is the sigmoid function. By using the relativistic discriminator [28], we have:

$$\begin{aligned}\mathcal{L}_{\text{GAN}} = \mathcal{L}_{\text{RaGAN}} = & -\mathbb{E}_{\mathbf{x}}\left[\log\left(1 - \mathbf{D_x}(\mathbf{x}, \mathbf{G}_{SR}(\hat{\mathbf{x}}))\right)\right] \\ & -\mathbb{E}_{\hat{\mathbf{y}}}\left[\log\left(\mathbf{D_x}(\mathbf{G}_{SR}(\hat{\mathbf{x}}), \mathbf{x})\right)\right]\end{aligned} \tag{13}$$

where, $\mathbb{E}_{\mathbf{x}}$ and $\mathbb{E}_{\hat{\mathbf{y}}}$ represent the operations of taking average for all real HR ($\mathbf{x}$) images and fake HR ($\hat{\mathbf{y}}$) images in the mini-batches, respectively.
TV (total-variation) loss ($\mathcal{L}_{tv}$): It focuses to minimize the gradient discrepancy and produces sharpness in the output SR image, and it is defined as:

$$\begin{aligned}\mathcal{L}_{tv} = & \frac{1}{N}\sum_{i=1}^{N}\Big(\|\nabla_h \mathbf{G}_{SR}(\hat{\mathbf{x}}^{(i)}) - \nabla_h(\mathbf{x}^{(i)})\|_1 + \\ & \Big\|\nabla_v \mathbf{G}_{SR}(\hat{\mathbf{x}}^{(i)}) - \nabla_v(\mathbf{x}^{(i)})\|_1\Big)\end{aligned} \tag{14}$$

Here, ∇_h and ∇_v denote the operators calculating the image directional derivatives in the horizontal and vertical directions, respectively.
Content loss ($\mathcal{L}_1$): It is defined as:

$$\mathcal{L}_1 = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{G}_{SR}(\hat{\mathbf{x}}^{(i)}) - \mathbf{x}^{(i)}\|_1 \tag{15}$$

where, N represents the size of mini-batch.

5.3. Training description

During the training phase, we set the input LR patch size as 32×32. We train the network for 51000 training iterations with a batch size of 16 using Adam optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ without weight decay for both generator and discriminator to minimize the loss in (10). We set the $\omega_0 = 30$ for the *Sine* layer. The learning rate is initially set to 10^{-4} and then multiplies by 0.5 after 5K, 10K, 20K, and 30K iterations. The projection layer parameter σ is estimated according to [15] from the input LR image. We initialize the projection layer parameter α on a log-scale values from $\alpha_{max} = 2$ to $\alpha_{min} = 1$ and then further fine-tune during the training via back-propagation.

6. Experimental Results

6.1. Training data preparation

We use the LR domain images ($\mathbf{y}$: 2650 HR images) that are corrupted with unknown degradation, e.g., sensor noise, compression artifacts, sharpening artifacts, etc., and HR domain images ($\mathbf{x}$: 800 clean HR images), provided in the NTIRE-2020 Real-World Super-resolution (RWSR) Challenge [17]. The LR domain images contain synthetic visible corruptions that are similar to the induced corruptions by the current camera devices. We use the LR and HR domain data to train the $\mathbf{G}_{LR}$ network to learn the domain degradation/corruptions. Then, we train the $\mathbf{G}_{SR}$ network by synthesizing the realistic LR/HR paired training data.

6.2. Data augmentation

We take the LR/HR patches due to the network training efficiency and we also assume that the patch based degrada-

Dataset (LR/HR pairs)	SR methods	#Params	PSNR↑	SSIM↑	LPIPS↓
			sensor noise ($\sigma = 8$)		
Bicubic	EDSR [14]	$43M$	24.48	0.53	0.6800
Bicubic	ESRGAN [28]	$16.7M$	17.39	0.19	0.9400
CycleGAN [16]	ESRGAN-FT [16]	$16.7M$	22.42	0.55	0.3645
DSGAN [3]	ESRGAN-FS [3]	$16.7M$	22.52	0.52	0.3300
DSGAN [3]	SRResCGAN [19]	$380K$	25.46	0.67	0.3604
DSSinGAN (ours)	SRResCSinGAN (ours)	$380K$	25.50	0.69	0.3750
			JPEG compression (quality=30)		
Bicubic	EDSR [14]	$43M$	23.75	0.62	0.5400
Bicubic	ESRGAN [28]	$16.7M$	22.43	0.58	0.5300
CycleGAN [16]	ESRGAN-FT [16]	$16.7M$	22.80	0.57	0.3729
DSGAN [3]	ESRGAN-FS [3]	$16.7M$	20.39	0.50	0.4200
DSGAN [3]	SRResCGAN [19]	$380K$	23.34	0.59	0.4431
DSSinGAN (ours)	SRResCSinGAN (ours)	$380K$	23.70	0.63	0.4258
			unknown (validset) [17]		
DSGAN [3]	SRResCGAN [19]	$380K$	25.05	0.67	0.3357
DSSinGAN (ours)	SRResCSinGAN (ours)	$380K$	25.58	0.69	0.3610
DSSinGAN (ours)	SRResCSinGAN+ (ours)	$380K$	25.89	0.71	0.3769

Table 1: ×4 SR quantitative results comparison of our method over the DIV2K validation-set (100 images) with added two known degradations *i.e.*, sensor noise ($\sigma = 8$) and JPEG compression ($quality = 30$) artifacts. Bottom section: ×4 SR results comparison with the unknown corruptions in the RWSR challenge series (validation-set) [17]. The arrows indicate if high ↑ or low ↓ values are desired. The best performance is shown in red and the second best performance is shown in blue.

tion is same as in the whole image. We augment the training data with random vertical and horizontal flipping, and 90° rotations. Moreover, we also consider another effective data augmentation technique, called mixture of augmentation (MOA) [29] strategy. In the MOA, a data augmentation (DA) method, among *i.e.*, Blend, RGB permutation, Mixup, Cutout, Cutmix, Cutmixup, and CutBlur is randomly selected then applied on the inputs. This MOA technique encourages the network to acquire more generalization power by partially blocking or corrupting the training sample.

6.3. Evaluation metrics

We evaluate the trained model under the Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), and LPIPS metrics. The PSNR and SSIM are distortion-based measures that correlate poorly with actual perceived similarity, while LPIPS better correlates with human perception than the distortion-based/handcrafted measures. As LPIPS is based on the features of pretrained neural networks, so we use it for the quantitative evaluation with features of AlexNet. The quantitative SR results are evaluated on the RGB color space. To further enhance the fidelity, we use a self-ensemble strategy [23] (denoted as SRResCSinGAN+) at the test time, where the LR inputs are flipped/rotated and the SR results are aligned and averaged for enhanced prediction.

6.4. Comparison with state-of-the-art SR methods

We compare our method with other state-of-art SR methods including EDSR [14], ESRGAN [28], ESRGAN-FT [16], and ESRGAN-FS [3] and SRResCGAN [19], whose source codes are available online. The two degradation settings (*i.e.*, sensor noise, JPEG compression) have been considered under the same experimental situations for all methods. We run all original source codes and trained models by the default parameters settings for comparison.

The EDSR is trained without perceptual loss (only $\mathcal{L}_1$) by a deep SR residual network using bicubic supervision. The ESRGAN is trained with the $\mathcal{L}_{\text{perceptual}}$, $\mathcal{L}_{\text{GAN}}$, and $\mathcal{L}_1$ by a deep SR network using bicubic supervision. The ESRGAN-FT and ESRGAN-FS apply the same SR architecture and perceptual losses as in the ESRGAN using the two known degradation supervisions. The SRResCGAN is trained with the similar losses combination as done in the ESRGAN using the two known degradation supervisions. We train the proposed SRResCSinGAN with the similar losses combination as done in the ESRGAN and SRResCGAN with the modified Resnet structure by the sine nonlinearities.

Table 1 shows the quantitative results comparison of our method over the DIV2K validation-set (100 images) with two known degradations (*i.e.*, sensor noise, JPEG compression) as well as the unknown degradation in the RWSR challenge dataset [17]. In the case of sensor noise, our method has better PSNR/SSIM values compared to all existing SR methods, while we have comparable LPIPS value. Since these are the contradicted measures (PSNR/SSIM vs. LPIPS), our objective is to achieve a good PSNR/SSIM score, while getting the satisfactory LPIPS value. In the case of jpeg compression artifacts, our proposed method has better PSNR/SSIM values except the EDSR, which is slightly better PSNR, but low LPIPS value and it has a very deep network with $43M$ parameters, while our model has only $380K$ parameters. Finally, in the case of unknown corruptions, our method has better SR results in terms of

PSNR and SSIM, while we have comparable LPIPS value with others.

Regarding the visual quality, Fig. 4 shows the qualitative comparison of our method with the other SR methods on ×4 upscaling factor (validation-set). In contrast to the existing state-of-art methods, our proposed method produces excellent SR results that are reflected in the PSNR/SSIM/LPIPS values, as well as the visual quality of the reconstructed images with almost no visible corruptions.

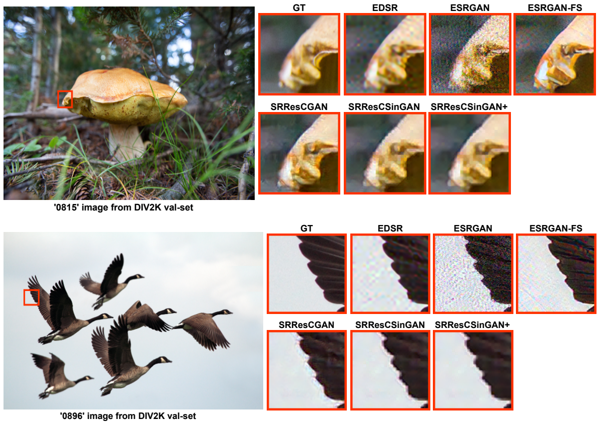

Figure 4: Visual SR comparison of our method with the other state-of-art methods on the DIV2K validation set at the ×4 upscaling factor.

Dataset (LR/HR pairs)	SR method	PSNR↑	SSIM↑	LPIPS↓
Bicubic	SRResCGAN	24.13	0.57	0.4853
Bicubic	SRResCSinGAN	24.78	0.62	0.4365
DSGAN	SRResCGAN	25.05	0.67	0.3357
DSGAN	SRResCSinGAN	25.53	0.69	0.3792
DSSinGAN	SRResCSinGAN	25.58	0.69	0.3610
DSSinGAN	SRResCSinGAN+	25.89	0.71	0.3769

Table 2: The quantitative SR results comparison of our method with others over the DIV2K validation set (100 images) with unknown degradation for our ablation study. The arrows indicate if high ↑ or low ↓ values are desired. The best performance is shown in red and the second best performance is shown in blue.

6.5. Ablation Study

For our ablation study, we generated different LR/HR pair data to train the SR models. We reached the better PSNR/SSIM score, while achieving good LPIPS for its better visual correlation with human perception. Table 2 shows the quantitative results of our method over the DIV2K validation-set (100 images) with unknown degradation [17].

In the top section of the table, we trained the SRResCGAN [19] method with and without sine nonlinearities with the bicubic downsampled data (refer to section 5 for the SR learning training). The SRResCGAN with sine nonlinearities (*i.e.*, SRResCSinGAN) has achieved better results in terms of PSNR, SSIM, and LPIPS.

In the middle section, we generated the LR data from the DSGAN [3] as done in SRResCGAN [19] and then trained the two variants of our SR model with the generated LR/HR pairs. The SRResCSinGAN has better SR results in terms of PSNR and SSIM, while satisfactory LPIPS value compared to the SRResCGAN.

In the bottom section, we generated the LR data from the DSGAN with sine nonlinearities (denoted as DSSinGAN, refer to section 4 for the LR learning) and then finally train our proposed SRResCSinGAN method with the generated LR/HR pairs. The SRResCSinGAN has better PSNR and LPIPS values, while the same SSIM value. To further enhance the performance, we used the self-ensemble strategy [23] at the test time, denoted as SRResCSinGAN+. It suggests that better generation of the LR images instead of the traditional bicubic downscaling gives the better performance gain and also incooperating the sinusoidal nonlinearites instead of ReLU/PReLU activation in the resnet structure gives the improvement in the reconstruction quality.

7. Conclusion

We proposed a deep SRResCSinGAN method for real image super-resolution by following the real-world settings. The proposed method solves the real image SR problem by implicitly modeling the degradation process within the network. The proposed approach first synthesize the realistic paired training data with a more practical corruptions/degradations, instead of using the traditional bicubic downsampling or the existing deep learning based methods. Secondly, the proposed approach use the synthesized LR/HR paired data to train the generalized SR model to super-resolve the real LR images. The proposed approach incorporate the sinusoidal nonlinearities in the LR and HR model process to better representing the underlying complex signals in natural images. Our method achieves better SR results in terms of PSNR/SSIM values and comparable LPIPS values as well as visual quality compared to the existing state-of-art methods.

References

[1] M. Bertero and P. Boccacci. *Introduction to inverse problems in imaging*. CRC press, 1998. 3

[2] M. Figueiredo, J. M. Bioucas-Dias, and R. D. Nowak. Majorization–minimization algorithms for wavelet-based image restoration. *IEEE Transactions on Image processing*, 16(12):2980–2991, 2007. 3

[3] M. Fritsche, S. Gu, and R. Timofte. Frequency separation for real-world super-resolution. *ICCV workshops*, 2019. 2, 4, 6, 7

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems (NIPS)*, pages 2672–2680, 2014. 2, 4

[5] Y. Gou, B. Li, Z. Liu, S. Yang, and X. Peng. Clearer: Multi-scale neural architecture search for image restoration. *NeurIPS*, 2020. 2

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, pages 1125–1134, 2017. 4

[7] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. *CVPR*, pages 1646–1654, 2016. 2

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, pages 1–15, 2015. 4

[9] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 4681–4690, 2017. 2, 5

[10] S. Lefkimmiatis. Universal denoising networks: A novel cnn architecture for image denoising. *CVPR*, pages 3204–3213, 2018. 5

[11] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. *ECCV*, pages 702–716, 2016. 4

[12] W. Li, K. Zhou, L. Qi, N. Jiang, J. Lu, and J. Jia. Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. *NeurIPS*, 2020. 2

[13] Y. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu. Feedback network for image super-resolution. *CVPR*, 2019. 2

[14] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. *CVPRW*, pages 1132–1140, 2017. 2, 6

[15] X. Liu, M. Tanaka, and M. Okutomi. Single-image noise level estimation for blind denoising. *IEEE Transactions on Image Processing (TIP)*, pages 5226–5237, 2013. 5

[16] A. Lugmayr, M. Danelljan, and R. Timofte. Unsupervised learning for real-world super-resolution. *ICCV workshops*, 2019. 2, 6

[17] A. Lugmayr, M. Danelljan, and R. Timofte. 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 (CVPR) Workshops*, June 2020. 2, 5, 6, 7

[18] Z. Luo, Y. Huang, S. Li, L. Wang, and T. Tan. Unfolding the alternating optimization for blind super resolution. *NeurIPS*, 2020. 2

[19] R. Muhammad Umer, G. Luca Foresti, and C. Micheloni. Deep generative adversarial residual convolutional networks for real-world super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 438–439, 2020. 1, 2, 3, 4, 6, 7

[20] R. Muhammad Umer, G. Luca Foresti, and C. Micheloni. Deep iterative residual convolutional network for single image super-resolution. In *Proceedings of the International Conference of Pattern Recognition (ICPR)*, January 2021. 2

[21] R. Muhammad Umer and C. Micheloni. Deep cyclic generative adversarial residual convolutional networks for real image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, August 2020. 1, 2

[22] V. Sitzmann, J. N. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020. 3

[23] R. Timofte, R. Rothe, and L. Van Gool. Seven ways to improve example-based single image super resolution. In *CVPR*, pages 1865–1873, 2016. 6, 7

[24] R. M. Umer, G. L. Foresti, and C. Micheloni. Deep super-resolution network for single image super-resolution with realistic degradations. In *ICDSC*, pages 21:1–21:7, September 2019. 2

[25] R. M. Umer, A. Munir, and C. Micheloni. A deep residual star generative adversarial network for multi-domain image super-resolution. In *6th International Conference on Smart and Sustainable Technologies (SpliTech)*, 2021. 2

[26] M. Unser. A representer theorem for deep neural networks. *Journal of Machine Learning Research*, pages 1–30, 2019. 3

[27] X. Wang, L. Xie, C. Dong, and Y. Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *ICCV*, pages 1905–1914, 2021. 2

[28] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy. ESRGAN: Enhanced super-resolution generative adversarial networks. In *ECCV*, 2018. 2, 5, 6

[29] J. Yoo, N. Ahn, and K.-A. Sohn. Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8375–8384, 2020. 6

[30] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *CVPRW*, pages 701–710, 2018. 2

[31] K. Zhang, J. Liang, L. Van Gool, and R. Timofte. Designing a practical degradation model for deep blind image super-resolution. In *CVPR*, pages 4791–4800, 2021. 2

[32] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2808–2817, 2017. 2

[33] K. Zhang, W. Zuo, and L. Zhang. Learning a single convolutional super-resolution network for multiple degradations. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3262–3271, 2018. 2

[34] K. Zhang, W. Zuo, and L. Zhang. Deep plug-and-play super-resolution for arbitrary blur kernels. In *CVPR*, pages 1671–1681, 2019. 2

[35] S. Zhou, J. Zhang, W. Zuo, and C. C. Loy. Cross-scale internal graph neural network for image super-resolution. *NeurIPS*, 2020. 2