Unsupervised Image Super-Resolution using Cycle-in-Cycle Generative Adversarial Networks

Yuan Yuan\textsuperscript{1,2,*} Siyuan Liu\textsuperscript{1,3,4} Jiawei Zhang\textsuperscript{1} Yongbing Zhang\textsuperscript{3} Chao Dong\textsuperscript{1} Liang Lin\textsuperscript{1}

\textsuperscript{1}Sensetime Research
\textsuperscript{2}Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University
\textsuperscript{3}Graduate School at Shenzhen, Tsinghua University, Shenzhen
\textsuperscript{4}Department of Automation, Tsinghua University, Beijing

Abstract

We consider the single image super-resolution problem in a more general case that the low-/high-resolution pairs and the down-sampling process are unavailable. Different from traditional super-resolution formulation, the low-resolution input is further degraded by noise and blurring. This complicated setting makes supervised learning and accurate kernel estimation impossible. To solve this problem, we resort to unsupervised learning without paired data, inspired by the recent successful image-to-image translation applications. With generative adversarial networks (GAN) as the basic component, we propose a Cycle-in-Cycle network structure to tackle the problem within three steps. First, the noisy and blurry input is mapped to a noise-free low-resolution space. Then the intermediate image is up-sampled with a pre-trained deep model. Finally, we fine-tune the two modules in an end-to-end manner to get the high-resolution output. Experiments on NTIRE2018 datasets demonstrate that the proposed unsupervised method achieves comparable results as the state-of-the-art supervised models.

1. Introduction

Recent deep learning based super-resolution (SR) methods have achieved significant improvement either on PSNR values \cite{8,12,13,16,17,25,28,30} or on visual quality \cite{16,23}. These methods require supervised learning on high-resolution (HR) and low-resolution (LR) image pairs. However, their common assumption that the downscaling factor is known and the input image is noise-free hinders them from practical usages. In real-world scenarios, the SR problem often have the following properties: 1) HR datasets are unavailable, 2) downscaling method is unknown, 3) input LR images are noisy and blurry. This problem is extremely difficult if the input images suffer from different kinds of degradation. For an easier case, in this study, we assume that input images are degraded with the same processing which is complex and unavailable.

Under the above circumstances, models learned from synthetic data tend to generate similar results as traditional
methods [13,30] or even simple interpolation. In Fig. 1 we show the results of bicubic interpolation and the state-of-the-art deep learning model—EDSR [17] with a noisy input. This is mainly due to the data bias between training and testing images. Detailed survey and analysis of deep learning based methods on real data can be found in [13].

As an alternative choice, blind SR [7,19,29] deal with the real-world data by estimating the down-sampling kernel from internal or external similar patches. However, when the input is noisy, the down-sampling kernel cannot be accurately estimated, and the inverse mapping results are accompanied by amplified noises. There are also works attempting at restoring LR images with additive Gaussian noises [34]. But real-world noises may neither be additive nor follow the standard Gaussian distribution, causing noise estimation infeasible. More generally, LR images may suffer from complex noises, blurry and non-uniform down-sampling kernels, which fail almost all existing blind SR methods.

Inspired by the development of unsupervised learning in image-to-image translation, such as CycleGAN [35] or WESPE [9], we intend to investigate unsupervised strategies to overcome this obstacle. In CycleGAN, images are translated between different domains with unpaired training data. They assume that the input image is of the same size as the output image, with only the difference on styles. However, in SR, output images are several times larger than the inputs, making the direct application of CycleGAN impossible. Further, using a bicubic-upsampled image as the input also could not obtain satisfactory results. SR problem is specific as it requires high quality output but not just a different style.

After exploring several training strategies, we find an effective Cycle-in-Cycle structure, named CinCGAN, which could achieve superior results. The whole pipeline consists of two CycleGANs, while the second GAN covers the first one (See Fig. 2). The first CycleGAN maps the LR image to the clean and bicubic-downsampled LR space. This module ensures that the LR input is fairly denoised/deblurred. We then stack another well-trained deep model with bicubic-downsampling assumption to up-sample the intermediate result to the desired size. Finally, we fine-tune the whole network using adversarial learning in an end-to-end manner. We conduct experiments on the NTIRE2018 Super-Resolution Challenge dataset, and show that the proposed Cycle-in-Cycle structure is much stable at training and achieves competitive performance as supervised deep learning methods.

The contributions of this work are three-folds: 1) We study a more general super-resolution problem, where the high-resolution ground truth, down-sampling kernel and degradation function are unavailable. 2) We explore several unsupervised training strategies under the above assumption, and show that super-resolution task is different from conventional image-to-image translation. 3) We propose a Cycle-in-Cycle structure that could achieve comparable results as supervised CNN networks.

2. Related work

2.1. Image Super-Resolution

Single image super-resolution (SISR) has been widely studied for decades. Early approaches either rely on natural image statistics [33,13] or pre-defined models [10,5,26]. Later, mapping functions between LR images and HR images are investigated, such as sparse coding based SR methods [30,32].

Recently, deep convolution neural networks (CNN) have shown explosive popularity and powerful capability to improve the quality of SR results. Ever since Dong [3] first proposed using CNN for SR and achieved the state-of-the-art performance, plenty of CNN architectures have been studied for SISR. Inspired by the VGG [24] networks used for ImageNet classification, Kim et al. [12] present a very deep network (VDSR) that learns a residual image. For accelerating the speed of SR, FSRCNN [1] and ESPCN [23] extract feature maps at the low-resolution space and up-sample the image at the last layer by transposed convolution and sub-pixel convolution, respectively. All the above mentioned CNN based SR methods aim at minimizing the mean-square error (MSE) between the reconstructed HR image and the ground truth. Based on the observation that minimizing MSE will make the SR results overly smooth, SRGAN [16] combines an adversarial loss [6] and a perceptual loss [24,11] as the final objective function, and generates visually pleasing images which contain more high frequency details than the MSE-loss based methods. The champion of NTIRE2017 Super-Resolution Challenge [27], EDSR [17], employs deeper and wider networks to achieve the state-of-the-art performance by removing the unnecessary modules in SRResNet [16].

2.2. Blind Image Super-Resolution

Although a lot of works focus on SR problems with known degradation/downsampling kernels, little works try to solve blind SR—the degradation operation from HR images to LR images are unavailable. Estimating the degradation/blur kernel is an essential step for blind SR. Wang et al. [29] propose a probabilistic framework combined with the image co-occurrence prior to estimate the unknown point spread function (PSF) parameters. According to the property that small image patches will re-appear in natu-
nal images, Michaeli and Irani \cite{19} present a method that is able to estimate the optimal blur kernel. Another relevant work \cite{21} introduces a convolution consistency constraint and $\ell_2$-norm regularization \cite{22} to guide the blur kernel estimation process, achieving state-of-the-art blind SR performance.

In this work, we investigate how deep learning can be beneficial for addressing blind SR problems.

### 2.3. Unsupervised Learning

Existing supervised deep learning methods cannot handle blind SR without LR-HR image pairs. In real-world scenarios, where paired data is unavailable, it is essential to find a way to realize unsupervised learning. Recent work on GAN \cite{6} provides a feasible solution, which includes a generator and a discriminator. The generator tries to generate fake images to fool the discriminator, while the discriminator aims at distinguishing the generated results from real data. GAN is widely used to solve the unsupervised learning problems. DualGAN \cite{31} and CycleGAN \cite{35} are two works about image-to-image translation using unsupervised learning, and both of them present an interesting network structure that contains a pair of forward and inverse generators. The forward generator maps domain X to domain Y, while the inverse generator maps the output back to domain X to maintain cycle consistency. Ignatov et al. \cite{9} use the similar architecture to design a weakly supervised photo enhancer (WESPE) that translates ordinary photos to DSLR-quality images.

Different from the proposed method, both DualGAN \cite{31} and CycleGAN \cite{35} deal with input and output images of the same size, while SR requires the output images several times larger than the inputs. Utilizing the property of cycle consistency, we present a Cycle-in-Cycle structure named CinCGAN as shown in Fig. 2. Given an input image $x$ to a “clean” LR image set $Y$, where images are noise-free and down-sampled from HR images $Z$ with bicubic kernel. In other words, we deblur and denoise the input images at low resolution. Second, we adopt an existing SR model to super-resolve the intermediate results to the desired resolution. In the end, we combine and fine-tune these two models simultaneously to get the final HR images.

Under the guidance of the above pipeline, we propose a Cycle-in-Cycle structure named CinCGAN as shown in Fig. 2. To be specific, we adopt two coupled CycleGANs to learn the mapping from $X$ to $Y$ and $Y$ to $Z$, respectively. Unpaired images $x_i \in X$, $y_j \in Y$ and $z_j \in Z$ are used for training where $y_j$ is down-sampled from $z_j$ with bicubic kernel. Details are given in the following.

### 3.1. LR Image Restoration

The framework of the first CycleGAN that maps an LR image $x$ to a clean LR image $y$ is shown as $\text{LR} \rightarrow \text{clean LR}$ in Fig. 2. Given an input image $x$, the generator $G_1$ learns to generate an image $\hat{y}$ that looks similar to the clean LR $y$, so as to fool the discriminator $D_1$. Meanwhile, $D_1$ learns to distinguish the generated sample $G_1(x)$ from the real sample $y$. To stabilize the training procedure, we use the least square loss \cite{13} instead of the negative log-likelihood used in \cite{6}. The generator-adversarial loss is:

$$L_{GAN}^{LR} = \frac{1}{N} \sum_{i}^N \|D_1(G_1(x_i)) - 1\|_2,$$

\footnote{For simplicity, we omit the subscript $i$ and $j$ in the following.}
In summary, the final objective loss for the \textit{LR$\rightarrow$clean LR} model is a weighted sum of the four losses:

$$L_{\text{total}}^{LR} = L_{\text{GAN}}^{LR} + w_1 L_{\text{cyc}}^{LR} + w_2 L_{\text{idt}}^{LR} + w_3 L_{\text{TV}}^{LR}$$

where $w_1, w_2, w_3$ are the weights of different losses.

### 3.2. Jointly Restoration and Super-Resolution

We then investigate how to super-resolve the intermediate image $\hat{y}$ to the desired size. Recently, the enhanced deep residual network – EDSR [17] has won the first prize in the NTIRE 2017 challenge on single image super-resolution [1]. For simplicity, we directly adopt EDSR as the \textit{SR} network stacked after $G_1$. Similarly, we use a discriminator $D_2$ for adversarial training both $G_1$ and \textit{SR} networks. We also utilize another generator $G_3$ to ensure cycle consistency between $x$ and the reconstructed $x''$. The GAN loss, cycle loss and TV loss for the \textit{LR$\rightarrow$HR} network are formulated as follows:

$$L_{\text{GAN}}^{HR} = \frac{1}{N} \sum_{i} \|D_2(SR(G_1(x_i))) - 1\|_2,$$

$$L_{\text{cyc}}^{HR} = \frac{1}{N} \sum_{i} \|G_3(SR(G_1(x_i))) - x_i\|_2,$$

$$L_{\text{TV}}^{HR} = \frac{1}{N} \sum_{i} \|\nabla_h \text{SR}(G_1(x_i))\|_2 + \|\nabla_w \text{SR}(G_1(x_i))\|_2.$$

For the identity loss, instead of maintaining the tint consistency between input and output, we consider ensuring the \textit{SR} network can generate adequate quality of super-resolved images. We define a new identity loss as:

$$L_{\text{idt}}^{HR} = \sum_{i} \|\text{SR}(z') - z\|_2.$$
where $z'$ is down-sampled from $z$ with bicubic kernel. This $\mathcal{L}_{\text{idt}}^{HR}$ makes the SR network does not betray its original ambition, such that the produced $\tilde{z}$ can be reasonable SR results.

To sum up, the total loss for fine-tuning the LR to HR networks is

$$
\mathcal{L}_{\text{total}}^{HR} = \mathcal{L}_{GAN}^{HR} + \lambda_1 \mathcal{L}_{cyc}^{HR} + \lambda_2 \mathcal{L}_{\text{idt}}^{HR} + \lambda_3 \mathcal{L}_{TV}^{HR}
$$

where $\lambda_1, \lambda_2, \lambda_3$, for $i = 1, 2, 3$, are weights of each loss.

### 3.3. Network Architecture

The architecture of generators $G_1, G_2, G_3$ and discriminators $D_1, D_2$ are shown in Fig. 3. We adapt similar architecture as the work of Zhu et al. [35], which has shown impressive results for unpaired image-to-image translation. Here, “conv” means convolution layer, where a Leaky ReLU layer with negative slope 0.2 is added right after except for the last convolution layer (we omit it for simplicity). “BN” means a batch normalization layer. The number after symbols $k, n$ and $s$ represents kernel size, number of filters and stride size, respectively. For example, \textbf{k3n64s1} refers to the convolution layer that contains 64 filters, of which the spatial size is 3 and stride is 1.

For the generators $G_1$ and $G_2$, we use 3 convolution layers at the head and tail, and 6 residual blocks in the middle. The generator $G_3$ shares the same architecture as $G_1$ and $G_2$, except for the 2-nd and 3-nd convolution layers in discriminator $D_1$, where the stride is set to 2 to perform down-sampling. As to the discriminator, we use a $70 \times 70$ PatchGAN for $D_2$. Since we up-sample LR images with a scale of $\times 4$, the size of input images is usually less than 70 (we use $32 \times 32$ LR images and $128 \times 128$ HR images for training). Hence, we modify the stride of the first three convolution layers as 1 for discriminator $D_1$, such that the respective field of $D_1$ is reduced to $16 \times 16$.

### 4. Experiments

In this section, we first introduce the dataset and details we used for training. We then evaluate the performance of the proposed CinCGAN model by comparing with several state-of-the-art SISR methods. Finally, we perform ablation study to validate the advantages of CinCGAN.
4.1. Training data

We take the track 2 dataset from the NTIRE2018 Super-Resolution Challenge for training. The challenge aims to restore a HR image given a degraded LR image. They provide a high-quality image dataset, DIV2K \[^1\], which contains 800 training images and 100 validation images. The DIV2K dataset contains almost all kinds of natural scenarios: buildings (indoor and outdoor), forest, lakes, animals, people, etc. The track 2 dataset is degraded from DIV2K dataset, with down-sampling, blurring, pixel shifting, and noises. Although the parameters of the degradation operators are fixed for all images, the blur kernels are randomly generated and their resulting pixel shifts vary from image to image. Hence, the degradation kernels of images in the track 2 dataset are unknown and diverse.

Since our purpose is to unsupervised train a network without paired LR-HR data, we take the first 400 images (numbered from 1 to 400) from the training LR set as input images \( X \), and the other 400 images (numbered from 401 to 800) from the HR set as demanding HR images \( Z \). The intermediate clean LR images \( Y \) are directly bicubic down-sampled from \( Z \). Similar to \[^4\] [^24], we augment data with 90 degree rotation and flipping. Our experiments are performed with a scaling factor of \( \times 4 \). We randomly crop \( X \) and \( Y \) with size \( 32 \times 32 \) and crop \( Z \) with size \( 128 \times 128 \). We conduct testing on the provided 100 validation images. Note that, although DIV2K contains paired training dataset, we do not use paired data for supervised training.

4.2. Training details

We divide our training process into two steps. We first train the model \( G_1, G_2 \) and \( D_1 \) for mapping LR images to clean LR images (shown as \( LR\rightarrow clean\ LR \) in Fig. 2). The three parameters in (5) are set to be \( w_1 = 10, w_2 = 5 \) and \( w_3 = 0.5 \), respectively. We train our model with Adam optimizer \[^14\] by setting \( \beta_1 = 0.5, \beta_2 = 0.999 \) and \( \epsilon = 10^{-8} \), without weight decay. Learning rate is initialized as \( 2 \times 10^{-4} \) and then decreased by a factor of 2 every 40000 iterations. The weights of filters in each layer are initialized using a normal distribution and the batch size is set as 16. We train the model over 400000 iterations, until it converges.

We then jointly fine-tune the LR to HR model (shown as \( LR\rightarrow HR \) in Fig. 2). We initialize our SR network by publicly available EDSR models. \[^3\] We set parameters in (10) as \( \lambda_1 = 10, \lambda_2 = 5 \) and \( \lambda_3 = 2 \). The optimizer is set almost the same as training the \( LR\rightarrow clean\ LR \) model, except for we initialize learning rate with \( 10^{-4} \). As to the weight of identity loss \( L_{idt} \) in (5), we set \( w_2 = 1. \) At each iteration, we update (5) and (10) in turn. We first train \( G_1 \) and \( G_2 \) to update the \( LR\rightarrow clean\ LR \) network. We then train \( G_1, SR \) and \( G_3 \) simultaneously to update the \( LR\rightarrow HR \) network.

We implement the proposed networks with PyTorch and train them on a Nvidia Tesla K80 GPU. It takes about 1 day to pre-train the \( LR\rightarrow clean\ LR \) model and about 2 days to jointly fine-tune the \( LR\rightarrow HR \) model.

4.3. Results

We compare the performance of the proposed CinCGAN model with several state-of-the-art SISR methods: FSRCNN \[^3\], EDSR \[^17\] and SRGAN \[^16\]. We use the publicly available FSRCNN and EDSR models which are trained with paired LR and HR images, where the inputs are clean LR images down-sampled from HR images. To make the results more comparable, we also fine-tune EDSR and SRGAN (labelled as EDSR \(^+\) and SRGAN \(^+\) respectively) with the paired track 2 dataset. To emphasize the effectiveness of CinCGAN structure, we also try to first denoise the input LR images and then super-resolve the denoised images for comparison. BM3D \[^2\] is one of the state-of-the-art image denoising approach, which is an efficient and powerful denoiser. Hence, we pre-process the test LR images with BM3D first, and then super-resolve it using EDSR (labelled as BM3D+EDSR).

Table 1 shows the average PSNR and SSIM values of the restored test images. It shows that FSRCNN and EDSR cannot work well if the blur and noises are unknown in the training process. After fine-tuning by paired track 2 dataset, EDSR \(^+\) and SRGAN \(^+\) improve their results and our method can work comparably against SRGAN \(^+\) in terms of PSNR and SSIM without paired training data. Although BM3D can remove noise, it also over-smooth the input images. The PSNR and SSIM values of BM3D+EDSR are lower than the proposed method. Several subjective results are illustrated in Fig. 4.

4.4. Ablation Study

To validate the advantages of the proposed CinCGAN model for the unsupervised SISR problem, we design some other network structures for comparison.

Structure 1 The first frame structure is to restore LR images \( X \) to HR images \( Z \) using only one CycleGAN, i.e. denoise, deblur and super-resolve the LR images at the same time. The structure of the model is shown in Fig. 6(a), where we set an LR image \( x \) as input to the SR network directly. Correspondingly, we only minimize the total loss \( L_{total} \) (with replacing \( SR(G_1(\cdot)) \) as \( SR(\cdot) \) in Eq. (6)). However, during the training procedure, we found that the result \( \tilde{z} \) are always unstable and there are a lot of undesired artifacts, as shown in Fig. 6(a). It is hard for a single network to simultaneously denoise, deblur and up-sample the degraded images. Similar to \[^4\] [^24], we augment data with 90 degree rotation and flipping.
Figure 4. Super-resolution results of “0801”, “0816” and “0853” (DIV2K) with scale factor ×4. EDSR+ and SRGAN+ are trained on paired NTIRE2018 track 2 dataset. BM3D+EDSR means using BM3D for denoising first and then using EDSR for super-resolution. The proposed CinCGAN model shows comparable results with SRGAN+ and is better than BM3D+EDSR method.

Table 1. Quantitative evaluation on NTIRE 2018 track 2 dataset of the proposed CinCGAN model, in terms of PSNR and SSIM.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>22.85</td>
<td>22.79</td>
<td>22.67</td>
<td>25.77</td>
<td>24.33</td>
<td>22.88</td>
<td>24.33</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.65</td>
<td>0.61</td>
<td>0.62</td>
<td>0.71</td>
<td>0.67</td>
<td>0.68</td>
<td>0.69</td>
</tr>
</tbody>
</table>

images, especially when the degradation kernels are different from image to image and with unsupervised learning.

**Structure 2** We remove $D_2$ and $G_3$ from the proposed CinCGAN model for our second experiment. We map the input LR images to a set of clean LR images using the same $LR\rightarrow clean\ LR$ networks shown in Fig. 2, we then super-resolve the converted LR images directly using the $SR$ network. The whole structure is shown in Fig. 5(b). The corresponding result is illustrated in Fig. 6(b). As we can see, some negligible noise in the resulted clean LR images is
Figure 5. Experiments for validating the advantages of the proposed structure. (a) Structure 1: transform the LR images $x$ to HR images $z$ directly with one CycleGAN model; (b) Structure 2: remove $D_2$ and $G_3$ from the proposed CinCGAN model; (c) Structure 3: remove $D_1$ and $G_2$ from the proposed CinCGAN model.

Figure 6. Super-resolution results of “0829” (DIV2K) with scale factor $\times 4$, for each frame structure as described in Fig. 5.

magnified and now is visible in the super-resolved images, which affects the visual quality.

**Structure 3** Our third experiment is performed by removing $D_1$ and $G_2$ from the proposed CinCGAN model, as shown in Fig. 5(c). We use one CycleGAN for the LR to HR model, where we take $G_1 + SR$ as the forward network and $G_3$ as the inverse network. $D_2$ is used for distinguishing $\tilde{z}$ from $z$. We load the pre-trained $G_1$ (in the $LR\rightarrow$clean $LR$ networks) and the downloaded EDSR models for initialization. Experimental results on Fig. 5(c) show that the resulting $\tilde{z}$ are still noisy. Since without the $L_{cyc}$ and $L_{GAN}$ constraints on $G_1$ network ($L_{idt}^{LR}$ and $L_{tv}^{LR}$ are still used for this model), $G_1$ is unable to denoise and deblur. The whole model becomes similar to Structure 1.

**Proposed Method** We then propose our final solution as shown in Fig. 2, jointly fine-tune LR to HR networks with CinCGAN. We sequentially update the $LR\rightarrow$ clean $LR$ and the $LR\rightarrow$HR models. With the two constraint $L_{total}^{LR}$ and $L_{total}^{HR}$, the $G_1$ network can denoise and deblur the degraded input image $x$, while the $SR$ network can up-sample as well as further restore the resulted intermediate image $\tilde{y}$. The final resulted SR image is shown in Fig. 5(d), which shows the best visual result comparing with other three structures.

### 5. Conclusions

We investigate the single image super-resolution problem with a more general assumption: the low-/high-resolution image pairs and the down-sampling process are unavailable. Inspired by the recent successful image-to-image translation applications, we resort to the unsupervised learning methods to solve this problem. Using generative adversarial networks (GAN), the proposed method contains two CycleGANs, where the second GAN covers the first one. The solution pipeline consists of three steps. First, we map the input LR images to the clean and bicubic-downsampled LR space with the first CycleGAN. We then stack another well-trained deep model with bicubic-downsampling assumption to up-sample the intermediate result to the desired size. Finally, we fine-tune the two modules in an end-to-end manner to get the high-resolution out. Experimental results demonstrate that the proposed unsupervised method achieves comparable results as the state-of-the-art supervised models.

**Acknowledgement.** This work is supported by SenseTime Group Limited and in part by the Projects of National Science Foundations of China (61571254), Guangdong Special Support plan (2015TQ01X16), and Shenzhen Fundamental Research fund (JCYJ20160513103916577).
References


