Real-world-like Single Image Super Resolution Training: Final Report

Tiancheng Si
The University of Edinburgh

Abstract

The single-image super-resolution (SISR) task aims at recovering a high-resolution (HR) image from a low-resolution (LR) image. Although recent deep-learning based models have achieved significant progress, most of them are failed to perform well on real-world LR cases, since they are trained and tested on paired LR-HR dataset where LR images are manually synthesised from specific degradation methods. However, paired real-world LR-HR images are often insufficient in practices. In this project, we propose a novel framework CSR-GAN: we use an image translation model CycleGAN (Zhu et al., 2017) to learn the mapping between real-world LR and synthetic LR images, and use the trained model to generate real-like LR images. Then, we use these data to train the supervised baseline model SRGAN (Ledig et al., 2017). Our goal is to train a network that can have a better performance on real-world test images from unknown degradation. We demonstrate the effectiveness of our approach in quantitative and qualitative experiments.

1. Introduction

Single-image super-resolution (SISR) is a classical problem in the field of computer vision, which aims to recover high-resolution (HR) image based on low-resolution (LR) image. It has a broad application in tasks such as medical image, satellite imaging and security surveillance (Tai et al., 2017). With the development of deep learning (DL), many DL-based SISR algorithms have achieved great performance, such as SRCNN (Dong et al., 2015) and SRGAN (Ledig et al., 2017). Most DL-based SISR models are trained in a supervised way with paired LR-HR images to learn the mapping from the distribution of LR to HR directly. Traditionally, LR image y is down-sampled from its corresponding clean HR image x, shown below:

$$y = (x \otimes k) \downarrow_s + n \tag{1}$$

where k denoted as a blur kernel such as bicubic or bilinear kernel, $\downarrow_s$ denotes a downsampling operation with scaling factor s such as 2×, 4× or 8×, and n is noise such as Gaussian noise.

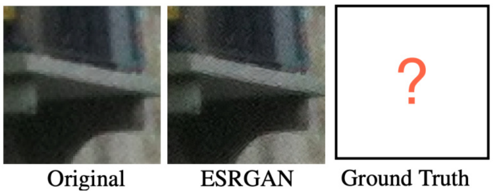

Figure 1. ESRGAN fails to generalize sensor noise and other artifacts in natural images (Lugmayr et al., 2019)

However, these DL-based algorithms have some limitations. These specific degradation methods cannot represent the real degradation processes, since real-world LR images have resulted from various factors, such as blurring, compression artefacts, colour and sensor noise (Bulat et al., 2018). Whereas the widely used strategy for most SR models is that images are downscaled beforehand (such as bicubic interpolation) to generate corresponding LR-HR training pairs. As a consequence, the artefact LR image is too “clean" and almost noise-free, which causes the model has never seen noise from the training data.

Hence, these methods are incapable of generalizing noise or other artefacts from the unseen images that are downsampled from unknown degradation process. The performance of models which are trained by a specific filter will drop drastically when they meet LR images from unknown degradation (Yang et al., 2019). An example of the poor SR performance on real-world images is shown in Figure 1.

The failure of supervised-learning based models on the testing stage mainly caused by the insufficient HR and real LR pairs. Though many researchers try to collect dataset with real-world LR-HR pairs (Cai et al., 2019; Zhang et al., 2019), this process is expensive and complicated since it requires many pre and post-processing procedures.

To address the aforementioned issue, Chen *et al.* (Chen et al., 2020) proposed a novel method, which uses an unsupervised image translation network CycleGAN (Zhu et al., 2017) to learn the mapping between real-world LR and synthetic LR images. Their proposed network manages to generate the corresponding real-LR-like images based on the input synthetic LR image, and is demonstrated to be well performed in NTIRE 2020 Real World Super-Resolution Challenge.

In this project, we aim at addressing the insufficient real-LR image problem in training the SISR model, and improving the performance of the model at the testing stage with unseen real-world LR images. Inspired by Chen *et* *al.* (2020), we propose a new SISR network **CSR-GAN** (Cycle Super-Resolution Generative Adversarial Network), which consists of two networks: SRGAN (Ledig et al., 2017) and CycleGAN (Zhu et al., 2017). Different to Chen *et al.*'s work which uses a CNN (Convolutional Neural Network) as the super-resolved model, we set SRGAN as our baseline model, which is the major part of the joint network. We choose SRGAN since the adversarial ability of generator and discriminator in the GAN framework has been proved to perform well in SISR task (Ledig et al., 2017), compare to CNN network (Dong et al., 2014). For the transfer network, we use CycleGAN as it has good performance on image-to-image translation task, such as grey-scale to colour, sketches to colour painting (Zhu et al., 2017), and translation between real-LR to approximately LR as demonstrated in Chen *et al.*'s work (2020).

Our main objective is to reduce the gap of the LR domain from training to the testing stage with SRGAN. To achieve this, we first use CycleGAN to learn distribution from real-world LR to synthetic LR. Then, the finetuned CycleGAN will be embedded into the SRGAN so that the network can generate real-world-like LR images. These LR images along with their HR counterpart will be used as pair for training the SRGAN network. After the new round training, the CSR-GAN network is capable of the SISR task for unseen real-world LR testing images. The detail of SRGAN and CycleGAN, and how they form our new CSR-GAN will be discussed in section 3.

Our contributions can be summarized as follows:

- We proposed an approach that uses unsupervised transition network CycleGAN to model real-world degradation process and generates real-LR-like images for training.
- We used real-LR-like images and HR images as data to train supervised-learning based network SRGAN for better super-resolution result on unknown cases.

The rest of the proposal is organized as follow: section 2 describes the dataset used in this project, as well as how the results are evaluated. Section 3 provides a detailed discussion of the baseline model SRGAN, transition network CycleGAN. The experiments and results are introduced in section 4, 5 and 6. Then, a review of published works is demonstrated in section 7. Finally, we present a brief summarise in section 8.

2. Data and task

2.1. Dataset

We use two different datasets to train our proposed network CSR-GAN. Firstly, for CycleGAN training, we use the publicly available dataset DIV2K (Agustsson & Timofte, 2017) with 1000 images (800 training images, 100 validation images and 100 testing images), which contain nearly all kinds of natural scenarios, including building, forest, lakes, animals and people. DIV2K dataset has both HR and LR images, where the degradation of LR images is unknown. Here, the task for CycleGAN is to learn the mapping between the paired data. Please note that we do not use the paired image for CycleGAN training. We use random real-world LR in the original training set and bicubic images to compose our unpaired training set. We use the original validation set for comparison and quantitative evaluation.

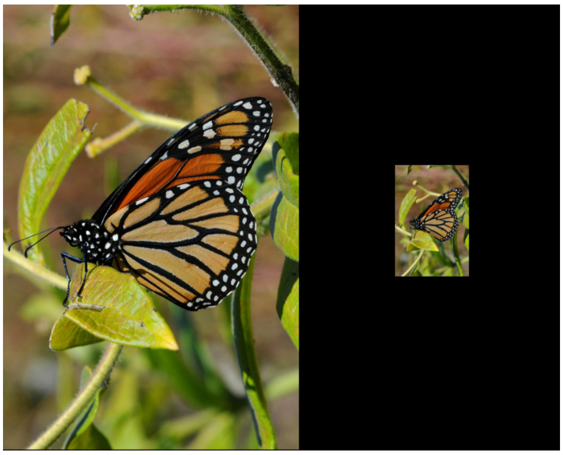

Figure 2. Example LR (right) and HR (left) training pairs from ImageNet dataset, scaling factor = 4×.

Secondly, for SRGAN training, we use a subset of ImageNet (Deng et al., 2009), which contains 1000 HR images. The image from this dataset will first be downscaled to the bicubic LR image in SRGAN, then it will be passed to the well-trained CycleGAN network as input. CycleGAN will output an approximately LR image based on the input, and pass it back to SRGAN. The output real-LR-like image and the original HR image will form a pair as the training data for SRGAN. The task for SRGAN here is to generate HR images based on the pair data, the output HR image should be as similar as possible to the ground truth image in the pair. Notice that although we use the term “pair" here to describe the training data, the CSR-GAN network only takes a single HR image as input. It will generate the corresponding real-LR-like image inside the network itself.

Apart from the testing set in DIV2K, for a fair comparison with the baseline model SRGAN, we use the same testing dataset in SRGAN: Set5 (Bevilacqua et al., 2012) and Set14 (Zeyde et al., 2010) as the “known" degradation LR images. We obtained the LR images by downsampling its corresponding HR images using a bicubic kernel with scale factor $r = 4$, which is the same as described in SRGAN's work (Ledig et al., 2017).

All images from the dataset were randomly cropped to a fixed size 96×96 by default before training and testing (which could be automatically done by using PyTorch package *torchvision.transforms.RandomCrop*) to speed up the training and testing process and reduce the GPU memory usage. An example training pair is shown in Figure 2, the LR image is downscaled with factor 4×.

2.2. Evaluation

We plan to use Peak Signal-to-Noise Ratio (PSNR) (in dB) (Wang et al., 2004), structure similarity (SSIM) (Wang et al., 2004) and Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018) to evaluate the performance. PSNR and SSIM are the most commonly used evaluation metrics for image similarity comparison. PSNR represent a measure of the peak signal error between the referenced image I^{HR} and the reconstructed image I^{SR} based on mean square error (MSE) metric. The equation of PSNR is given by:

$$\mathrm{PSNR}\left(I^{HR}, I^{SR}\right) = 10\log_{10}\left[\frac{R^2}{\mathrm{MSE}\left(I^{HR}, I^{SR}\right)}\right] \tag{2}$$

where R is the maximum possible pixel value of the image. The higher PSNR score means the higher similarity of two given images at pixel-level.

SSIM measure brightness, contrast and structure of the referenced image and the processed image. The equation of SSIM is defined as follow:

$$l(I^{HR}, I^{SR}) = \frac{2\mu_{I^{HR}}\mu_{I^{SR}} + c_1}{\mu^2_{I^{HR}} + \mu^2_{I^{SR}} + c_1} \tag{3}$$

$$c(I^{HR}, I^{SR}) = \frac{2\sigma_{I^{HR}}\sigma_{I^{SR}} + c_2}{\sigma^2_{I^{HR}} + \sigma^2_{I^{SR}} + c_2} \tag{4}$$

$$s(I^{HR}, I^{SR}) = \frac{\sigma_{I^{HR}I^{SR}} + c_3}{\sigma_{I^{HR}}\sigma_{I^{SR}} + c_3} \tag{5}$$

$$SSIM(I^{HR}, I^{SR}) = l(I^{HR}, I^{SR})\cdot c(I^{HR}, I^{SR})\cdot s(I^{HR}, I^{SR}) \tag{6}$$

where μ calculates the average value of I^{HR} and I^{SR}, σ^2 is the variance of I^{HR} and I^{SR}, $\sigma_{I^{HR}I^{SR}}$ is the covariance of x and y, and c1 and c2 are two constants that balanced the whole equation. The higher SSIM is, the two images are more alike in the independence of brightness and contrast, and the properties of the object structure in the scene.

LPIPS is a recent proposed learned metric, which can measure the perceptual quality of reconstruction (Zhang et al., 2018). Lower LPIPS score represents higher perceptual similarity. The equation of LPIPS is defined as follow:

$$d\left(x, x_0\right) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| w_l \odot \left(\hat{y}^l_{hw} - \hat{y}^l_0 hw\right)\right\|^2_2 \tag{7}$$

where d is the distance between x and x_0. The feature stack is extracted from the lth layer and unit-normalised in the channel dimension. The vector W_l is used to deflate the number of activation channels, and then the L2 distance is calculated. Finally, the equation averaging over each feature space and summing over channels.

3. Methodology

3.1. SRGAN

To solve single image super-resolution (SISR), Super-Resolution Generative Adversarial Network (SRGAN) was introduced in (Ledig et al., 2017), and SRGAN was selected as our baseline model.

Same as the GAN network which was introduced by Goodfellow *et al.* (2014), SRGAN consists of two main parts: generator network G and discriminator D. So in the training process of SRGAN, we first input I^{HR} which is ground truth of the HR images and then the model will use Gaussian filter to degrade the HR images into counterpart LR images I^{LR}. In our practice, the degradation method based on the bicubic downscale approach is used to do the degradation process to generate low-resolution images I^{LR}. The main goal of generator G is to estimate super-resolved image I^{SR} from low-resolution image I^{LR}, and the discriminator is to distinguish super-resolved images I^{SR} from real images I^{HR}. Using GAN systems, the generator G and discriminator will be jointly optimized, to allow the picture generated by generator G to "fool" discriminator D.

The first part is generator network, which is a feed-forward convolutional neural network G_{θ_G} with parameters $\theta_G = \{W_{1:L}; b_{1:L}\}$, where $W_{1:L}$ and $b_{1:L}$ are the weights and bias in L-layer network of generator network. By training the network, we attempt to optimize the parameters using specifically designed loss function l^G which will be further discussed in Equation 9. The main optimization function for generator network is shown below:

$$\hat{\theta}_G = \underset{\theta_G}{argmin}\ \frac{1}{N}\sum_{n=0}^{N} l^G(G_{\theta_G}(I^{LR}_n), I^{HR}_n) \tag{8}$$

Based on the GAN model, the generator final loss function l^G which is also the perceptual loss function consists of two parts: content loss l^G_{Con} and adversarial loss l^G_{Adv}

$$l^G = l^G_{Con} + 10^{-3} l^G_{Adv} \tag{9}$$

For content loss l^G_{Con}, the most common used optimization content loss is the pixel-wise MSE loss. However, the MSE optimization function will cause blurry results which lacking high-frequency content. Hence, VGG loss, is introduced:

$$l^{SR}_{VGG/i,j} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(I^{LR})_{x,y})^2 \tag{10}$$

where $\phi_{i,j}$ is the feature map gained by the j-th convolution (after activation) before i-th max-pooling layer in VGG-19 network, $W_{i,j}$,$H_{i,j}$ is the dimensions of respective feature maps in the VGG network. The new proposed loss allows visually superior image generation, helping to overcome

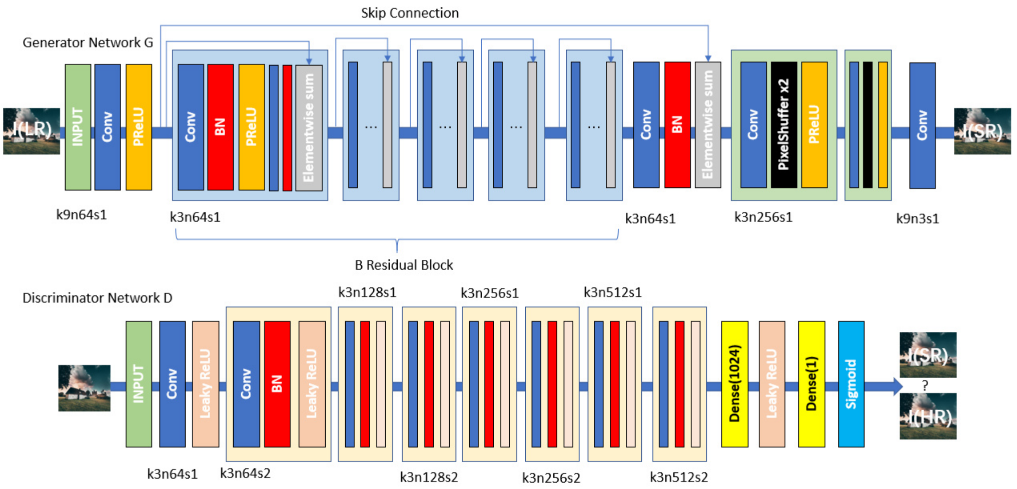

Figure 3. Architecture of Generator G and Discriminator D, where k is kernel size, n is the number of feature maps and s is stride for each layer.

ill-posed inverse problems of decoding nonlinear feature representations. So in our baseline model of SRGAN, we use VGG loss as content loss based on the VGG19_36th pre-trained model to find the maximum square error between feature maps.

For adversarial loss l^G_{Adv}, which is the generative loss, it is based on probabilities of the discriminator:

$$l^G_{Adv} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR})) \tag{11}$$

The second part is the discriminator network D_{θ_D} with parameter θ_D.The main function of the discriminator is to distinguish super-resolved images I^{SR} from real images I^{HR}. The objective function is given by:

$$\min_G \max_D \left[E_{x,y}[\log D(y)] + E_{x,z}[\log(1 - D(G(x)))]\right] \tag{12}$$

where the discriminator tries to maximize this objective function, and the generator tries to minimize it. Once the network is convergent, the outputs from the generator should well approximate the desired distribution, in our case, the generator should be able to generate images that as realistic as possible. The optimization problem of discriminator D will use the gradient descent method to optimize, where binary cross-entropy loss will be used in real practice.

Figure 3 illustrates the architecture of the generator network and discriminator network. The generator network consists of several convolutional layers with 9x9 kernels and 64 feature maps, each is followed by ParametricReLU (He et al., 2015a). Then, the blue block in B residual blocks consist of two convolutional layers with 3x3 kernels and 64 feature maps, followed by batch-normalization layers (Ioffe & Szegedy, 2015) and ParametricReLU. Besides, a deep residual network ResNet (He et al., 2015b) with skip-connection within B residual blocks is employed. Finally, two sub-pixel convolution layers (Shi et al., 2016) are used to increase resolution of image.

For discriminator network, the LeakyReLU(Mehralian & Karasfi, 2018) with activation($\alpha = 0.2$) is used. The structure consists of eight convolutional layers with 3x3 filter kernels, which increase feature maps from 64 to 512. Finally, two Dense layers and the Sigmoid activation layer are employed to gain the probability of classification.

3.2. CycleGAN

Using SRGAN, we can train the generative network to generate HR images. However, because the LR images used in SRGAN training are all obtained by the same degradation method, SRGAN has unsatisfactory generalisation ability to generate HR images from unseen and realistic LR images, which are suffering from other degradation processes with different formation of blur and noise.

To improve the generalisation ability to fit various circumstances of LR images in the real world, we use another GAN structure, CycleGAN, which is introduced by Zhu et al. (2017), as the prepossessing approach of real-world LR images. CycleGAN is an unsupervised image-to-image translation method that can learn the mapping from the real-world LR domain to the synthetic LR images domain.

The basic ideas of CycleGAN illustrate in Figure 4. As demonstrating in Figure4(a), the CycleGAN contains two different mapping function: $G : X \to Y$ and $F : Y \to X$ with their corresponding discriminator network D_X and D_Y. D_X help generator F to translate inputs Y into the outputs X that are hard to classify by D_Y, vice versa for D_Y and G. In Figure 4 (b) and (c), the cycle consistency loss is presented to enhance the regularization of mappings. When we translate inputs in one domain to outputs in another domain and back again, we will transfer the cycle-consistency loss to

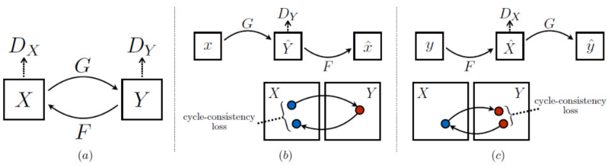

Figure 4. Basic Methodology of CycleGAN (**?**)

optimize it to where it started. Figure 3 (b) shows forward cycle-consistency loss:$x \to G(x) \to F(G(x)) \approx x$, (c) is back cycle-consistency loss:$y \to G(y) \to G(F(y)) \approx y$.

We apply the adversarial loss of both mapping function for G and F and cycle consistency loss subjected to both G and F. So the for the mapping function $G : X \to Y$ and its discriminator D_Y, the adversarial loss is expressed as following:

$$\begin{aligned} \mathbb{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_d ata(y)}[\log D_Y(y)] \\ + \mathbb{E}_{x \sim p_d ata(x)}[\log(1 - D_Y(G(x)))] \end{aligned} \tag{13}$$

where the function G is attempting to generate images G(x) which should be similar to real image y, while discriminator D_Y is attempting to distinguish generated images G(x) and real image y. So the G aims to minimize the object against an adversary D that tries to maximize it, namely $\underset{G}{min}\,\underset{D_Y}{max}\,\mathbb{L}_{GAN}(G, D_Y, X, Y)$. Similarly we can define the adversarial loss for the mapping function $F : Y \to X$ and its discriminator D_X: $\mathbb{L}_{GAN}(F, D_X, Y, X)$, and the objective should be $\underset{F}{min}\,\underset{D_X}{max}\,\mathbb{L}_{GAN}(F, D_X, Y, X)$:

$$\begin{aligned} \mathbb{L}_{GAN}(F, D_X, Y, X) = \mathbb{E}_{x \sim p_d ata(x)}[\log D_X(x)] \\ + \mathbb{E}_{y \sim p_d ata(y)}[\log(1 - D_X(F(Y)))] \end{aligned} \tag{14}$$

Using adversarial training theoretically can learn function G and F to map two domains respectively, but the network can map some images in one domain to any random set of images in another domain, so, only adversarial training can not guarantee the specific input x_i can map the targeted output y_i. Therefore, the function should be cycle-consistent which is shown in Figure 4 (b) and (c). The CycleGAN should use image translation cycle to bring x back to original domain:i.e.$x \to G(x) \to F(G(x)) \approx x$ and $y \to F(y) \to G(F(y)) \approx y$. So we introduce the cycle consistency loss:

$$\begin{aligned} \mathbb{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_d ata(x)}[||F(G(x)) - x||_1] \\ + \mathbb{E}_{y \sim p_d ata(y)}[||G(F(y)) - y||_1] \end{aligned} \tag{15}$$

The total loss function are the weighted sum of these three items above:

$$\begin{aligned} \mathbb{L}(G, F, D_x, D_Y) = L_{GAN}(G, D_Y, X, Y) \\ + L_{GAN}(F, D_X, Y, X) \\ + \lambda L_{cyc}(G, F) \end{aligned} \tag{16}$$

where λ controls the relative importance of the two objectives. The main goal of CycleGAN is:

$$G^*, F^* = arg \underset{G,F}{min}\, \underset{D_x,D_Y}{max}\, L(G, F, D_x, D_Y) \tag{17}$$

Using CycleGAN, we try to build a model that can map real-world unknown LR images and LR images with a specific degradation method. In our practice, the specific degradation method in SRGAN is a method based on a bicubic downscale.

3.3. CSR-GAN

After introducing the methodology of SRGAN (Ledig et al., 2017)and CycleGAN(Zhu et al., 2017), we propose a new SISR network CSR-GAN (Cycle Super-Resolution Generative Adversarial Network), which is the improvement version of SRGAN for dealing with real-world LR images.

To better generalize the real-world LR images, we first train CycleGAN with real-world LR images and bicubic LR images, and store the CycleGAN as a pre-trained model.

Then when we training CSR-GAN, at the initial step, we input HR images, and then use the bicubic degradation method to generate counterpart LR images. Next, we use a pre-trained CycleGAN model to convert the bicubic LR images into images that are similar to real-world LR images. Finally, we use the newly generated images to put into the previous SRGAN based GAN model.

In CSR-GAN, the CycleGAN and SRGAN are not jointly optimized, but regard the CycleGAN as a pre-trained model to process the LR images. The CSR-GAN is based on the SRGAN but is different from SRGAN for input images during training.

4. Experiment 1: Baseline Model Testing

4.1. Description

The problem of “models that trained on the dataset from specific degradation method does not perform well on real cases" proposed by Lugmayr et al.(2019) was mentioned in section 1. We first test the performance of the pre-trained SRGAN[1] model on the mainstream datasets: Set5 & Set14, DIV2K (Bicubic LR) and DIV2K (Unknown LR) to verify if the same problem recurred on our Baseline SRGAN model. Both Set5 and Set14 are downsampled manually using bicubic kernel (these two widely-used datasets is utilized to check if the performance of this pre-trained SRGAN reached the expectation comparing to the original SRGAN and other state-of-the-art models), while DIV2K contains both bicubic LR and unknown downgrading LR data. The expected testing results on DIV2K_Unknown should be lower than the ones on DIV2K_Bicubic to show the failure of SRGAN in handling the real-world LR images, since this network was only trained on DIV2K bicubic data.

[1] https://github.com/Lornatang/SRGAN-PyTorch.

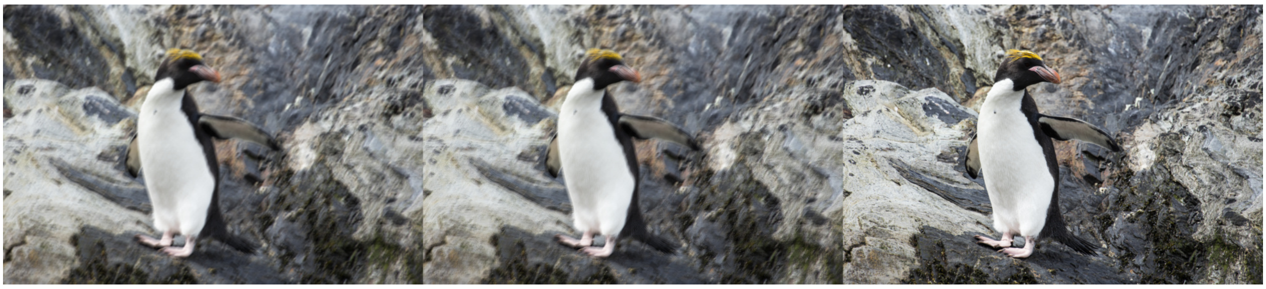

Figure 5. DIV2K unknown downgrading operators x4 first test image 0801 (Agustsson & Timofte, 2017) result: bicubic upscaling (left), SRGAN generated SR (mid) and ground truth HR (right), where the pre-trained SRGAN is trained on DIV2K bicubic LR images.

Notice that this pre-trained model is not the exact model from the original paper (which should be trained on a random sample of 350 thousand images from the ImageNet database (Deng et al., 2009) as the author of SRGAN did not provide open-source code. Thus our test result on Set5 and Set14 might be different from the result indicated in the paper by Ledig et al. (2017).

In this experiment, we did not yet change any hyperparameter. All the hyperparameter settings remained unchanged and the results were gained by testing the benchmark on the SRGAN pre-trained model implemented in the PyTorch environment.

4.2. Results

We use three metrics: PNSR, SSIM and LPIPS to evaluate the outputs from pre-trained SRGAN. The results are shown in Table 1. The pre-trained model's performance on Set5 and Set14 is similar to the results listed in the paper. However, it can be seen from this table that given the degradation mismatch with that of training, the performance of SRGAN decreases drastically from { ***PSNR 27.49; SSIM 0.8161; LPIPS 0.2287*** } on **DIV2K_Bicubic** to { ***PSNR 18.50; SSIM 0.4252; LPIPS 0.4897*** } on **DIV2K_Unknown** . Hence, it verified the hypothesis we made in the first place, and it also motivates us to further improve SRGAN in real-world cases.

Dataset	PSNR↑	SSIM↑	LPIPS↓
Set5 (x4)	30.62	0.8789	0.1516
Set14 (x4)	25.96	0.7608	0.2305
DIV2K_Bicubic (x4)	27.49	0.8161	0.2287
DIV2K_Unknown (x4)	18.50	0.4252	0.4897

Table 1. Evaluation results of pre-trained SRGAN on Set5, Set14, DIV2K_Bicubic and DIV2K_Unknown (Upscale-factor: 4x)

The specific example can be found in Figure 5, which compares the super-resolution image generated by SRGAN with the expected target (ground truth). We can see that the Blur was not be correctly handled by the pre-trained SR. The test result for that single image (*0801x4.png*) is { ***PSNR 21.34; SSIM 0.4892; LPIPS 0.5317*** }. This certain example clearly shows the failure of the SISR function of the pre-trained SRGAN regarding unknown degrading images, where the perceptual quality of SR is no better than simply upscaled image by bicubic. This example also proves that the LPIPS (Zhang et al., 2018) as a perceptual metric can reflect the perceptual judgement.

5. Experiment 2: CycleGAN Training

5.1. Description

To better map the real-world resolution images with bicubic LR images, we train CycleGAN to learn the mapping. In this experiment, we use DIV2K unknown downgrading ×4 training dataset (800 pictures) as the real-world resolution image dataset, and we use DIV2K bicubic downgrading ×4 training dataset (800 pictures) as the bicubic resolution image dataset.

In this experiment, we did not change any hyperparameter of the CycleGAN model[2], so all the hyperparameter settings remained unchanged as defined by Zhu et al. (2017). The learning rate is 0.0002, with 200 epochs. In the first 100 epochs, the learning rate remains unchanged, and the next 100 epochs learning rate will linearly decay to zero. The test set using the DIV2K bicubic downgrading ×4 validation dataset (100 pictures). Because the DIV2K dataset has the corresponding unknown downgrading x4 validation dataset, so we can use PSRN and SSIM to evaluate the result. In the test process, we input bicubic downgrading ×4 pictures into the pre-trained CycleGAN model, and then generate synthetic real-world images by CycleGAN. we evaluate PSRN and SSIM between synthetic LR real-world images and ground truth of LR real-world images.

5.2. Results

As shown in Table 2, the PSNR is 29.3581 and SSIM is 0.8292, which clearly shows that the CycleGAN successfully map the bicubic LR images into unknown LR images. The specific example can be found in Figure 6.

Dataset	PSNR	SSIM
DIV2K (x4)	29.3581	0.8292

Table 2. Test results of our trained CycleGAN. The results represent the similarity between the CycleGAN generated real-world-like LR and the real world LR images

[2] https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

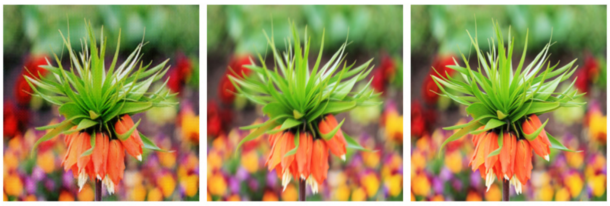

Figure 6. CycleGAN result for DIV2K dataset (Agustsson & Timofte, 2017) example 0803: synthetic unknown (left), DIV2K unknown (mid) and DIV2K bicubic (right), where the PSRN and SSIM for this example is 28.2543 and 0.8834 respectively

6. Experiment 3: CSRGAN Training

6.1. Description

After the CycleGAN was trained and finetuned for the style transformation task mapping from bicubic LR images to real-world LR images, it was then embedded in the SRGAN dataset preprocessing section *(srgan-pytorch/dataset.py)*. In this part, unsupervised style transformation mapping should be brought into effect in restoring the blur and CMOS noise that occurred in real-world LR images (commonly seen in photos taken by smartphones). Though the whole network still only receives HR images as the input target, it no longer uses self-generated bicubic Lr images as the counterparts. Instead, the CSR-GAN would use the self-generated real-world-like Lr images.

Then this CSR-GAN model was trained on the same subset of the DIV2K to preview the super-resolution results (comparing with the baseline) and tune the hyper-parameters. At last, we fed the entire training set to our new model and compare the performance on the DIV2K unknown test set with the ones of state-of-the-art models. The settings we used for CSR-GAN were as follows : {20000 PSNR epochs and 2000 gan epochs; batch size 8; image-size(for randomCrop) 96; upscale-factor 4x; PSNR & gan learning rate 1e-4}. The test results are listed below in Table 3.

6.2. Results

		PSNR	SSIM	LPIPS
SRGAN	Set 5	30.62	.8779	.1516
	Set 14	25.96	.7608	.2305
	DIV2K_Bicubic	27.49	.8161	.2287
	DIV2K_Unknown	**18.50**	**.4252**	**.4897**
CSRGAN	Set 5	28.17	.8196	.1829
	Set 14	25.05	.7135	.2351
	DIV2K_Bicubic	25.64	.7939	.2437
	DIV2K_Unknown	**21.36**	**.5912**	**.4203**

Table 3. Test results of our CSRGAN on datasets Set 5, Set 14, DIV2K_Bicubic and DIV2K_Unknown in comparison with the corresponding result of pretrained SRGAN (Upscale-factor: 4x)

It can be seen from Table 3 that with training data improved by the embedded cycleGAN, the testing result on the DIV2k unknown downgrading LR images has made a great progress {***PSNR: 21.36; SSIM: 0.5912; LPIPS:0.4203***} comparing to the {***PSNR 18.50; SSIM 0.4252; LPIPS 0.4897***}. This conclusion can be revealed by the SR output comparison shown in Figure 7. On the other hand, the results concerning bicubic data (Set 5, set 14 and DIV2K_Unknown) has been slightly affected. For solving this kind of score reduction that occurred on traditional datasets, we proposed another possible training technique in the subsection 6.4 *Possible Improvement*.

6.3. Other Test

Given this impressive result, the NITRE 2020 real-world LR image challenge (Track 2) smartphone images validation dataset is also passed to a pre-trained model as it was designed for this. Since there is no Ground truth for those LR images, it is not possible to judge the performance by metric. The result is moderately satisfying from our point of view. And if permitted by the Covid-19 lockdown policy, we would like to invite some volunteers to conduct **MOS** (*Mean Opinion Score*) experiment to subjectively review these SR images in the future.

6.4. Possible Improvement

One possible way to further improve the performance of the SISR network on real-world LR images is to modify the network structure. replace the current SR network with the state-of-the-art SR models, say ESRGAN, and evolve the current style-transform network into a more powerful one, say StyleGAN. The architecture of joint training (e.g. CycleSR) is also worth trying. The detail of these related networks will be discussed in section7.

Another idea is to bring the “mask" mechanism of the Transformer model from the NLP field into the CSRGAN, i.e. when handling the input HR images, it has a probability of 0.5 to generate the real-world like LR and a probability of 0.5 to generate the bicubic downsampling LR. This might help improve the network generalization ability, which means the CSRGAN can increase the generation quality on real-world LR image but with no obvious degradation of the test metric on the mainstream bicubic LR data.

7. Related Work

Before the widely used deep learning (DL) framework, most SISR algorithms are traditional hand-crafted methods, including statistical method, edge-based method, prediction-based method, and patch-based method (Yang et al., 2014).

With the development of powerful GPU (Krizhevsky et al., 2012), and the availability of public access dataset (Deng et al., 2009), the DL method dominates the research methodology of computer vision, including SISR. CNN based SISR network is one of the earliest traditional DL method of SISR algorithm, such as Super-Resolution Convolutional Neural Network (SRCNN) (Dong et al., 2015). Though these CNN based frameworks greatly improve the perfor-

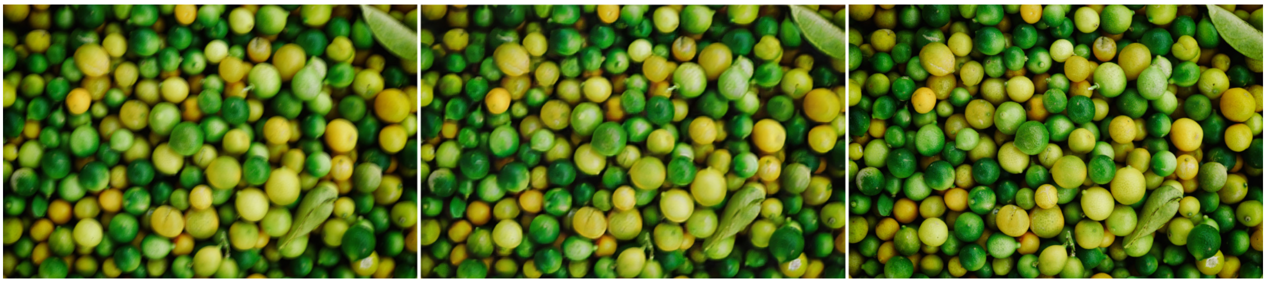

Figure 7. DIV2K unknown downgrading operators x4 valid set image 0802 (Agustsson & Timofte, 2017) output comparison: Pretrained SRGAN generated SR (left), **Our CSRGAN generated SR (mid)** and the ground truth HR (right).

mance of traditional non-DL based method, it has some drawbacks. One is that it can result in blurry and overly smooth effect, which have been heavily criticized by some SISR works (Ledig et al., 2017).

To address this issue, some researchers attempt to use the GAN-based SISR framework for solving SISR task, such as SRGAN (Ledig et al., 2017). GAN is first proposed by Goodfellow *et al.* (2014). The adversarial ability of GAN encourages the generator to produce a perceptually similar image rather than focusing on pixel-level similarity. With the combination of perceptual loss (Johnson et al., 2016), GAN is able to generate perceptually superior solution image than the CNN framework in the SISR task.

Most DL-based algorithms are trained by specific blurrings, such as bicubic degradation or Gaussian kernel degradation. While a model with a specific downsampling setting in training cannot produce good results for real-world LR images (Yang et al., 2019). Bulat *et al.* (2018) first came up with the idea to use GAN structure to simulate the image degradation process through unpaired image data. Compare to SRGAN, Bulat *et al.*'s result has shown to perform better than on real-world LR case in their work.

Similar to Bulat *et al.*, Chen *et al.* attempt to use CycleGAN network to model the degradation process of real-LR image (Chen et al., 2020). CycleGAN is first proposed by Zhu *et al.* in 2017, which is an image-to-image translation network that can learn the mapping between an input image and an output image. Compare to the GAN framework, CycleGAN is trained in an unsupervised way when paired training data are not available. Chen *et al.* use this property of CycleGAN to learn the mapping between real-LR and bicubic LR when the real-world real-LR and HR pair is insufficient. The approximate LR image from CycleGAN after training has shown great similarity in Chen *et al.*'s work.

8. Conclusions

In this project, we proposed an architecture for SISR: CSR-GAN, which can approximate the real LR scenario. Instead of training the model based on paired LR-HR data where LR is generated from a specific degradation method, we came up with a novel approach, which consists of an unsupervised image translation network CycleGAN to generate the real-LR-like image, and a supervised learning model SRGAN that train on the real-LR-like image from CycleGAN. Both quantitative and qualitative experiment demonstrate that our proposed framework outperforms the baseline model SRGAN on unknown degradation dataset DIV2K.

The CSR-GAN can be applied in many specific areas by retraining the CycleGAN part, not limited to smartphone photography. For example, it can imitate the monitor recordings of city surveillance cameras, so that the finetuned CSR-GAN can be more capable of the SR task under the monitor recording scenario. It would be useful for clearly recognizing the features of criminal in crime scene OR the number of vehicle License Plate in bumper to bumper traffic. This network design has great potential to be explored. for computer vision-related tasks.

References

Agustsson, Eirikur and Timofte, Radu. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

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

Bulat, Adrian, Yang, Jing, and Tzimiropoulos, Georgios. To learn image super-resolution, use a gan to learn how to do image degradation first. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 185–200, 2018.

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

Chen, Shuaijun, Han, Zhen, Dai, Enyan, Jia, Xu, Liu, Ziluan, Xing, Liu, Zou, Xueyi, Xu, Chunjing, Liu, Jianzhuang, and Tian, Qi. Unsupervised image super-resolution with an indirect supervised path. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 468–469, 2020.

Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dong, Chao, Loy, Chen Change, He, Kaiming, and Tang, Xiaoou. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pp. 184–199. Springer, 2014.

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

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. *Advances in neural information processing systems*, 27: 2672–2680, 2014.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015a. URL http://arxiv.org/abs/1502.01852.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015b. URL http://arxiv.org/abs/1512.03385.

Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

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

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Ledig, Christian, Theis, Lucas, Huszár, Ferenc, Caballero, Jose, Cunningham, Andrew, Acosta, Alejandro, Aitken, Andrew, Tejani, Alykhan, Totz, Johannes, Wang, Zehan, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690, 2017.

Lugmayr, A., Danelljan, M., and Timofte, R. Unsupervised learning for real-world super-resolution. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 3408–3416, 2019. doi: 10.1109/ICCVW.2019.00423.

Mehralian, M. and Karasfi, B. Rdcgan: Unsupervised representation learning with regularized deep convolutional generative adversarial networks. In *2018 9th Conference on Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium*, pp. 31–38, 2018. doi: 10.1109/AIAR.2018.8769811.

Shi, Wenzhe, Caballero, Jose, Huszár, Ferenc, Totz, Johannes, Aitken, Andrew P., Bishop, Rob, Rueckert, Daniel, and Wang, Zehan. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *CoRR*, abs/1609.05158, 2016. URL http://arxiv.org/abs/1609.05158.

Tai, Ying, Yang, Jian, and Liu, Xiaoming. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3147–3155, 2017.

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

Yang, Chih-Yuan, Ma, Chao, and Yang, Ming-Hsuan. Single-image super-resolution: A benchmark. In *European conference on computer vision*, pp. 372–386. Springer, 2014.

Yang, Wenming, Zhang, Xuechen, Tian, Yapeng, Wang, Wei, Xue, Jing-Hao, and Liao, Qingmin. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21(12):3106–3121, 2019.

Zeyde, Roman, Elad, Michael, and Protter, Matan. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pp. 711–730. Springer, 2010.

Zhang, Richard, Isola, Phillip, Efros, Alexei A, Shechtman, Eli, and Wang, Oliver. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Zhang, Xuaner Cecilia, Chen, Qifeng, Ng, Ren, and Koltun, Vladlen. Zoom to learn, learn to zoom, 2019.

Zhu, Jun-Yan, Park, Taesung, Isola, Phillip, and Efros, Alexei A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.