



# Perceptual image quality using dual generative adversarial network

Masoumeh Zareapoor<sup>1</sup> · Huiyu Zhou<sup>2</sup> · Jie Yang<sup>1</sup>

Received: 31 December 2018 / Accepted: 9 May 2019 / Published online: 17 May 2019  
© Springer-Verlag London Ltd., part of Springer Nature 2019

## Abstract

Generative adversarial networks have received a remarkable success in many computer vision applications for their ability to learn from complex data distribution. In particular, they are capable to generate realistic images from latent space with a simple and intuitive structure. The main focus of existing models has been improving the performance; however, there is a little attention to make a robust model. In this paper, we investigate solutions to the super-resolution problems—in particular perceptual quality—by proposing a robust GAN. The proposed model unlike the standard GAN employs two generators and two discriminators in which, a discriminator determines that the samples are from real data or generated one, while another discriminator acts as classifier to return the wrong samples to its corresponding generators. Generators learn a mixture of many distributions from prior to the complex distribution. This new methodology is trained with the feature matching loss and allows us to return the wrong samples to the corresponding generators, in order to regenerate the real-look samples. Experimental results in various datasets show the superiority of the proposed model compared to the state of the art methods.

**Keywords** Image processing · Perceptual quality · Data distribution · Generative adversarial network · Classification

## 1 Introduction

Image super-resolution is a technique that attracts much attention and progress in recent years. Despite the great progress and achievements, still, there is no unique solution exists, in particular for high magnification ratios. Each pixel loss which used by the existence approaches does not properly capture perceptual variances between output and input images [1, 2]. Thus, for the high upscaling factor (i.e., scale factor 4 or more), it is difficult to recover the high frequencies details in the images. Generative adversarial

network (GANs) is a conglomerate of deep learning and generative model that is proposed by Goodfellow et al. [3]. GANs models are known to produce realistic samples from latent space in a simple manner. In their original setting, they employ two neural networks based on adversarial training in a minimax game. Generator  $G$  is trained to produce fake samples from a noise space, whereas the discriminator learns how to make difference between fake (generated samples) and real (true data) samples. Since the advent of GANs, many works have been appeared which using GANs in different computer vision applications in particular, simulating complex data distributions such as images, videos and texts [4–6]. However, they suffer from a major problem of perceptual quality and also are extremely difficult to train. Given this, limits the GANs applicability, and recently some attempts have been appeared based on joint supports of the data distribution and using hierarchical models in contrast to the original GAN which is a direct model. In this paper, we propose a novel model that generalizing the GAN framework to multiple generators and discriminators, in terms of stabilizing the training process as well as improving sample diversity. Borrowing

---

✉ Jie Yang  
jieyang@sjtu.edu.cn  
Masoumeh Zareapoor  
mzarea222@gmail.com  
Huiyu Zhou  
hz143@leicester.ac.uk

<sup>1</sup> School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China

<sup>2</sup> Department of Informatics, University of Leicester, Leicester, UK

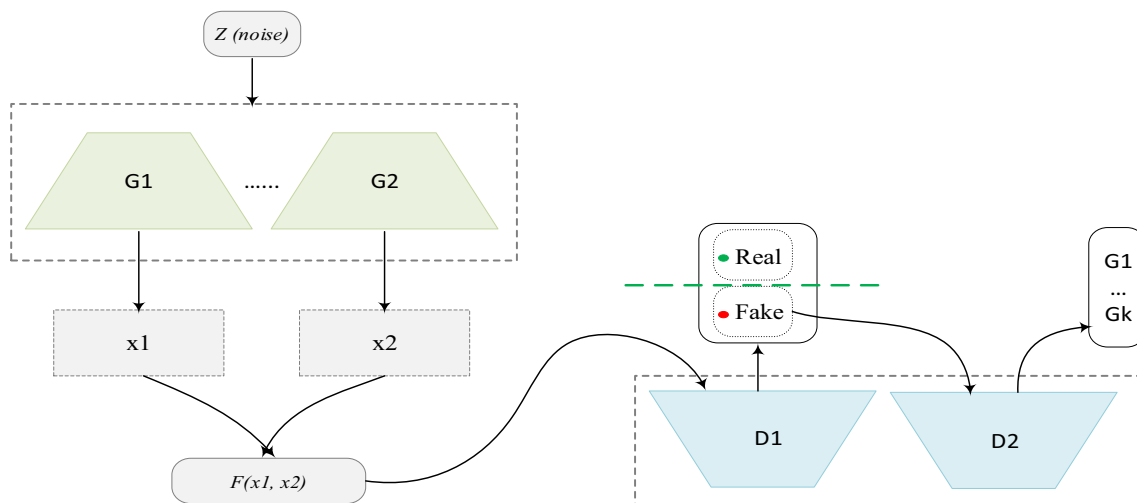
from GMAN [7, 8], we propose to employ two generators with two discriminators, which are based on an image-to-image model. The proposed architecture termed as *DualGAN*, is shown in Fig. 1. Similar to the regular GAN, the objective of the generators is to increase the mistake of the discriminator. Moreover, unlike the other GAN variations that use multi-generators in their structure, our proposed model simultaneously trains both the generators and the data distribution will be obtained from the mixture of their induced distributions. In terms of the fact, multiple generators may affect the trivial solution, in which all the generators attempt to generate similar sample images. Based on this observation, we address this problem by designing two discriminators in our architecture; such that, one of them determines the real or fake samples while another one acts as classifier in order to identify the related generators that generated the wrong samples. We prove that, our model is able to effectively learn complex data distribution, in order to generate real-look samples and could significantly improve the image's quality even at the highest scale factor  $\times 8$ . The main objective of this paper lies on: (i) propose a new variation generative adversarial model to train a couple of generators and discriminators with enforcing a better Jensen–Shannon (JS) divergence among the generators; (ii) optimizing objective function toward minimizing the JS divergence between the mixture of data distributions and the real data distribution by using feature matching loss; (iii) a comprehensive evaluation on real-world datasets in order to prove the effectiveness of our model with respect to other variation of GANs. The paper including introduction consists of six main sections.

In next section, we review and summarize the GAN-based models. Sections 3 and 4, present the proposed architecture with the mixture generator-discriminator extension to the GAN framework. Section 5 contains the experimental results and related discussions. Finally, Sect. 6 concludes the paper. It is worth to mention that, throughout the paper, we use the following notations for sake of brevity;  $I_{LR}$  (low-resolution image) and  $I_{SR}$  (super-resolution image).

## 2 Related works

There was a drastic growth for generative models in the last few years. Substantial methods have been proposed to address the image super-resolution problems [9–13]. The main concept of generator adversarial network (GAN) [3] states an adversarial game between two networks:  $D$ -discriminator network and  $G$ -generator network. The generator draws the synthetic images from the noise input, and the discriminator receives input from the real and the synthetic samples and determines whether is it fake (generated by generator) or from the real image.

Moreover, GAN alternatively optimizes the generator and discriminator using stochastic gradient-based learning. However, training of GAN suffers from the main problems, as mode collapse and difficulties in the implementation and unstable results [4, 12, 14]. In the standard GAN, there is no way to control what to be generated, since there is no information for the learning generators. However, [15] proposed a new method in order to define more condition for the generator so that the generated image can be



**Fig. 1** DualGAN consists of a pairs of generators and discriminators:  $G_1, G_2$  and  $D_1, D_2$ . The generative model with a couple generators trains for generating realistic artificial images and the discriminative model with a couple discriminators, along with determining whether an image is real or fake, it also identifies the related generators that generate the wrong samples. We use the weight-sharing constraint for

all layers of the generative models,  $g_1$  and  $g_2$ . We also use the weights-sharing constraint for the last layer of the discriminative model,  $d_1$  and  $d_2$ . The “weight-sharing constraint” permits the proposed model how to learn a joint data distribution of images and also reduces the model parameters at the optimal level

designed with desired target. Most GAN-based methods follow same structure and using a generator and a discriminator in their model with a minor variation. Some of the most realist GAN variations in this categories are: InfoGAN [16], DCGAN [13], WGAN [17], ImprovedGAN [14] and DGAN [8]. These methods in fact are straightforward to design and implementation. Recent attempts to improve the GANs results and solve the training issues are by training additional generators and discriminators. D2GAN [18] is a new approach which uses two discriminator in its architecture to find a rational distribution across the data by minimizing the KL (Kullback–Leibler) and the reverse KL divergence. Another framework is proposed by Durugkar et al. [7], which uses several discriminators to improve the generator learning. Recently, Arora et al. [19] proposed MIX + GAN approach which is another direction of GAN. The method is based on training several generators and discriminators with different parameters. However, this method is computationally expensive to train, due to lack of parameter sharing and there is no mechanism to enforce the divergence between generators. Tolstikhin et al. [20] proposed a new variation of GAN, termed as AdaGAN, to introduce a robust reweighting scheme for preparing a training data for GAN. Another model in what we follow is MAD-GAN that is proposed by Ghosh et al. [21]. It trains multiple generators with a multi-class discriminator. Their model is designed to improve the objective function of discriminator to push multiple generators toward generating diverse modes. Reed et al. [5] also proposed a GAN-based method which is able to generate  $64^2$  images and can barely generate intense object details. Accordingly, StackGAN [6] is proposed to stack two GANs in order to improve the [5] by generating  $256^2$  images. SS-GAN [22] is another method which comprises two GANs that, one is used for generating a surface normal map, and another GAN takes input from the generated samples and noise  $z$  and then produce an output image. In [23], the author presents LR-GAN method which learns to generate image foreground and background by using different generators and a single discriminator. The authors experimentally proved that by separating the generation of foreground and background image content, they can produce sharper images. Some other researchers believe that, instead of using different generators that perform separately to produce different task, the models can use multiple generators with similar structures wherein each generators refine the details of the results from the previous generator. Then, the last generator will be generating the final result. When using this strategy, the model can share the weights and parameters among generators, and it helps to smooth the training process. LAPGAN [24] is another method that uses multiple generators to generate images from coarse to fine using Laplacian pyramid [25]. Both

generators have the same structure; each generator takes a noise vector as input, and then, output will be a generated image. The only difference in the structures of these generators is the size of input and output dimension. With respect to the existing GAN-based image super-resolution techniques, that have achieved a successful progress, however, there are still some unsolved problems such as training instability and high-resolution generation [8, 26]. In terms of the fact, in this paper, we want to introduce a new mode of GANs to significantly use the potential advantages of generators and discriminators. The motivation of the proposed model is to jointly produce multiple samples, and it would increase the chance of sharing more details with model distributions. Multiple generators focus to complete the missing details for producing the higher resolution images. Multiple discriminators allow the model to accurately classify the generated samples and stabilize the model training in the best possible way. In addition, it is easy to see that the training difficulty will be decreased in the proposed methodology.

### 3 Dual generative adversarial network

The standard GAN involves two networks in its structure:  $G$ -generative and  $D$ -discriminative; which are simultaneously trained. Let  $X$  and  $Z$  be the true variable and latent variables. The generative model uses data distribution;  $p_G(x, z) = p_G(z)p_G(x|z)$  to generate samples. If  $\{G_i, i \in S\}$ , where  $G_i$  is a function, the generator defines a distribution  $D_{G_i}$  from Gaussian distribution and generate  $h$ , then apply the  $G_i$  on the generated  $h$  and achieve  $x = G_i(h)$ . Similarly, for the discriminator model  $\{D_j, j \in S'\}$ , where  $D_j$  is a function from binary space  $[0, 1]$ . Training the discriminator enforces the output to get high value 1 when  $x$  is from distribution  $D_{\text{true}}$  and a low value 0 when  $x$  is from the distribution  $D_{G_i}$ . The GAN framework with a  $G$  and a  $D$  can be jointly trained as [3]:

$$\min_g \max_d V(d, g) = \mathbb{E}_{x \sim D_{\text{true}}} [-\log d(x)] + \mathbb{E}_{z \sim p_z} [-\log(1 - d(g(z)))] \quad (1)$$

Practically, the above equation will be solved by the following material:

$$\begin{aligned} \theta_d^{t+1} &= \theta_d^t - \alpha^t \nabla_{\theta_d} V(d^t, g^t), \text{ and} \\ \theta_g^{t+1} &= \theta_g^t + \alpha^t \nabla_{\theta_g} V(d^{t+1}, g^t) \end{aligned} \quad (2)$$

where  $\theta_d, \theta_g$  are the discriminator and generator parameters,  $\alpha$  indicates the learning rate and  $t$  is number of iteration. The proposed DualGAN is illustrated in Fig. 1; our contribution in generative model is to use mixture of many distributions which is available in the training space,

instead of one. The proposed model consists of two generators  $G_1, G_2$  and two discriminators  $D_1, D_c$  such as, one of the discriminator acts as multi-class classifier. A high-resolution image  $I_{HR} \in [0, 1]^{\omega \times h \times c}$  is downsampled to a low-resolution image by:  $I_{LR} = \hat{d}(I_{HR}) \in [0, 1]^{\omega \times h \times c}$ , based on, the width, height and color channel,  $(\omega \times h \times c)$ .

### 3.1 Generative models

Let  $X_1, X_2$  be the generated images from  $G_1$  and  $G_2$  by using  $x_1 \sim p_{x_1}, x_2 \sim p_{x_2}$  distributions, respectively. We denote the distribution of the generators as,  $p_{g_1}, p_{g_2}$  and both the generators using multilayer perceptrons [27]:

$$\begin{aligned} g_1(z) &= g_1^m(g_1^{m-1}(\dots g_1^2(g_1^1(z)))) \text{ and for} \\ g_2(z) &= g_2^n(g_2^{n-1}(\dots g_2^2(g_2^1(z)))) \end{aligned} \tag{3}$$

where  $m$  and  $n$  indicate the number of layers in two generators  $g_1, g_2$  with the condition  $m =$  or  $\neq n$ . Each generator has a single distribution, and two generators together induce a mixture distribution from both; we term it as  $P_{MG}$ , and its corresponding coefficient can be as  $\pi = [\pi_1, \pi_2]$ . As the objective of generator is to minimize the JS divergence between the mixtures of generated data distribution and the true data distribution and maximize the JS divergence among two generators. The generative models gradually decode information from more abstract to more complex details. Note that, this learning process is opposed to the discriminator. In this process  $\theta_{g_1} = \theta_{g_2}$ , that means, we force the generators to have identical structures and share the weights. However, in the discriminative network, only the last layers of discriminators share the weights. In fact, the generators use the shared high-level representation for fooling the discriminator. Salimans et al. [14] proposed an approach for semi-supervised classification by using GAN model, termed as SSL-GAN. In their work, the discriminator is considered as multi-class classifier and improved the GAN convergence by optimizing the generator using feature matching loss. Here, inspired from the same work [14], we used feature matching loss in order to train the mixture data distribution in the generators.

$$L_F(x) = \min_{\theta_{g_1}, \theta_{g_2}} \mathbb{E}_{x \sim p_{\text{data}}(x)}[f(x)] - \mathbb{E}_{x \sim p(z)}[f(G(z; \theta_G))] \tag{4}$$

The feature matching loss function is used to allow the generators to control the mixture data distribution which on one hand has support which does not overlap with high-density areas of the real data, but still close to the data distribution [28]. Experimentally, we observed that when the generative model is trained with a feature matching loss, Eq. 4 is able to generate samples from mixture data distribution that fall onto the data manifold and has an impressive ability to generate high-quality samples.

### 3.2 Discriminative model

Let  $d_1, d_2$  be the discriminators of our proposed model, in which  $d_1$  determines the real or fake samples, and another one acts as classifier to classify that, the samples are form which generators. Two discriminators will be defined as:

$$\begin{aligned} d_1(x_1) &= d_1^t(d_1^{t-1}(\dots d_1^2(d_1^1(x_1)))) \text{ and } d_2(x_2) \\ &= d_2^q(d_2^{q-1}(\dots d_2^2(d_2^1(x_2)))) \end{aligned} \tag{5}$$

where  $t, q$  are the number of layers in the  $d_1$  and  $d_2$  discriminators.  $d_1$  Maps the input to a probability scores and then estimates the output as fake or real samples. In the next step, the output will be transferred to the  $d_2$  in order to find the related generators and return the wrong samples to its corresponding. We force both the discriminators to have the same layers in their architecture to prevent the mode collapse problem, and this is achieved by sharing the weights at the last layers as:  $\theta_{d_1} = \theta_{d_2}$ . Moreover, this weight-sharing helps to reduce the number of parameters in the discriminative models. Therefore, the proposed framework will be formulated as:  $\max_{G_1, G_2} \min_{D_1, D_2} V(d_1, d_2, g_1, g_2)$  for both the generators  $\theta_{g_1} = \theta_{g_2}$  and similarly, for the discriminators  $\theta_{d_1}^l = \theta_{d_2}^l$  which is having shared weights in the last layers, and then the function  $V(\cdot)$  will be as:

$$\begin{aligned} &\mathbb{E}_{x_1 \sim p_{x_1}}[-\log d_1(x_1)] + \mathbb{E}_{z \sim p_z}[-\log(1 - d_1(g_1(z)))] \\ &+ \mathbb{E}_{x_2 \sim p_{x_2}}[-\log d_2(x_2)] + \mathbb{E}_{z \sim p_z}[-\log(1 - d_2(g_2(z)))] \end{aligned} \tag{6}$$

The generative models  $G$  with two generators work for synthesizing images with a mixture distribution for confusing the discriminative models. Accordingly, the discriminative model  $D$ , receive the input from  $G$  and real data distribution, tries to classify them as the training data distribution or generated data distribution and also, identify the generators that generated the wrong images. The collaboration between the generators in the generative model and discriminators in the discriminative models is based on the weight-sharing constraint. Our proposed model will be trained by backpropagation [3] with alternating gradient update steps [14].

## 4 Model training

Learning proposed model relies on samples which are trained from the joint data distributions. Weight-sharing constraints are an important factor in our contribution, which can enable the networks to control their common information and improve the training performance. Moreover, the sharing weight constraint allows the model to

minimize the number of parameters and degrade the complexity to the original GAN. In our proposed model, all the generators are part of deep convolutional neural networks which can share the weights in all layers excluding the input layer. The input layer maps the noise  $z$  to the first hidden layer activation  $h$ . In the other side, the discriminators also employ a convolutional neural network and shares parameters in all layers except for the last layer. The generators used a sequence of upsampling layers which let us to add more details to generate a high-resolution image. However, only downsampling block is used for the discriminators. For generators  $G_1, G_2$  with their mixture weights  $Multi(\pi)$ , the optimal discriminators  $\hat{D}_1, \hat{D}_2$  yield the following equations:

$$\begin{aligned} \hat{D}_1 &= \alpha \times \frac{p_{\text{true data}}(x)}{p_G(\text{Mix})}; \\ \hat{D}_2 &= \beta \times \frac{p_G(\text{Mix})}{\sum_{j=1}^2 \pi p_{G_j}(\text{Mix})}; \text{ where} \\ p_G(\text{Mix}) &= \sum_{j=1}^2 \pi(p_{G_j}(x)) \end{aligned} \tag{7}$$

In fact, it can be seen that  $\hat{D}_2$  is a general case of  $\hat{D}_1$  which classifies the wrong samples into their corresponding generators. Based on these observations, we reformulate the objective function for the generative model as:

$$\begin{aligned} LS_G &= \mathbb{E}_{x \sim p_{\text{true data}}} \left[ \log \frac{p_{\text{true data}}(x)}{p_{G_1}(x) + p_{G_2}(x)} \right] \\ &+ \mathbb{E}_{x \sim p_G(x)} \left[ \log \frac{p_G(x)}{p_{G_1}(x) + p_{G_2}(x)} \right] \\ &- \beta \left[ \sum_{k=1}^2 \pi \mathbb{E}_{x \sim p_{G_k}(x)} \left[ \log \frac{\pi p_{G_k}(x)}{\sum_{j=1}^2 \pi p_{G_j}(x)} \right] \right] \end{aligned} \tag{8}$$

As the original GAN, the objective of generators is to minimize JS divergence between the data distributions while maximize it between the generators [14]. We verify the maximal loss function by setting  $D_1 = D_2 = 0$ ;  $\frac{\alpha p_{\text{true data}}(x)}{D_1} - p_G(x)$ , and  $\frac{\beta p_G(x)}{D_2} - p_{\text{true data}}(x) = 0$ . Note that, the discriminator  $d_1$  takes input from  $G$  and determines whether the samples are fake or real data. Next, the fake samples are taken as input by discriminator  $d_2$  in order to indicate the corresponding generator that generated the fake samples. The first discriminator is binary valued; however, the second discriminator acts as multi-class-classifier (depends on the number of generators, i.e., in this paper, both the discriminators have binary values, since we have only two generators in our model).

### 4.1 Implementation details

In the proposed model, to design the generators, we followed [27] and use the fractional length convolutional (FL-CONV) instead of standard CONV layer. Each FCONV layer followed by batch normalization and the parameterized rectified linear unit (PReLU) process [29], except the output layer, which uses feature matching loss (Eq. 4) in order to generate a desired pixel range values. However, the discriminators of our model are based on the standard convolutional layer (CONV) except the last layers which are based on fully connection layers (FC). We observed that leaky rectified linear unit (LReLU) [30] works better rather than ReLU, especially for the diverse samples which are produced by multiple generators. We also applied batch normalization in every layer, except the output layer of the discriminators which uses sigmoid units. The generators consist of “six” fractional convolutional layers while the discriminators have six convolutional layer plus two fully connection layers. The generators and discriminators are parameterized by  $\vartheta_G, \vartheta_D$ , respectively. The input layer for generator  $G_k$  is parameterized by the mapping  $f\vartheta_G(z)$  that maps the sampled noise  $z$  to the first hidden layer activation  $h$ . TensorFlow [31] is used to implement our model, Adam optimizer [32] and momentum set to 0.0002 [32] and 0.5, respectively, also weights initialized from an isotropic Gaussian,  $\mu$  (0, 0.01) and zero biases. The details of the networks are given in Tables 1 and 2. In addition, it is worth to mention that, we implemented the proposed model in a system with following features, Intel i7-6850 K CPU with a 64 GB RAM and an NVIDIA GTX Geforce 1080 Ti GPU and the operating system is Ubuntu 16.04.

### 5 Experimental evaluation

We conduct a series of experiments to evaluate the proposed model and compare it with other related approaches. In fact, we want to visualize and evaluate the learning behavior of our model using two generators and demonstrate its stability and efficacy based on different datasets. The experiments are conducted on three widely used datasets: BSD-100, DIV2 K and CIFAR. Results and evaluations on these dataset show that our model is able to generate more faithful and more diverse samples than the baselines. We compared our proposed DualGAN with some alternative approaches. We select the baselines from CNN-based methods such as, SRCNN [11], VDSR [33], LapSRN [34] and also several known variation of GAN including, DCGAN [13], ProGAN [35], BEGAN [36], GOGAN [37], Unrolled GAN [38], GMAN [7], MAGAN

**Table 1** Designed generative model in DualGAN. FL-CONV indicates the fractional length convolutional; BN is the batch normalization, and PReLU represents the parametric rectified linear units

| Layer | $G_1$                              | $G_2$                              |
|-------|------------------------------------|------------------------------------|
| 1     | FL-CONV, (1024,K4x4,S1), BN, PReLU | FL-CONV, (1024,K4x4,S1), BN, PReLU |
| 2     | FL-CONV,(512,K3x3,S2), BN, PReLU   | FL-CONV, (512,K3x3,S2), BN, PReLU  |
| 3     | FL-CONV,(256,K3x3,S2), BN, PReLU   | FL-CONV, (256,K3x3,S2), BN, PReLU  |
| 4     | FL-CONV,(128,K3x3,S2), BN, PReLU   | FL-CONV, (128,K3x3,S2), BN, PReLU  |
| 5     | FL-CONV,(64,K3x3,S2), BN, PReLU    | FL-CONV, (64,K3x3,S2), BN, PReLU   |
| 6     | FL-CONV,(32,K3x3,S2), BN, PReLU    | FL-CONV, (32,K3x3,S2), BN, PReLU   |
| 7     | FL-CONV,(3,K3x3,S1), LF            | FL-CONV, (3,K3x3,S1), LF           |

**Table 2** Designed discriminative model in DualGAN. Conv indicate the convolutional layer; BN is the batch normalization, and LReLU represents the leaky rectified linear units

| Layer | $D_1$                          | $D_2$                          |
|-------|--------------------------------|--------------------------------|
| 1     | CONV,(32,K5x5,S2), BN, LReLU   | CONV,(32,K5x5,S2), BN, LReLU   |
| 2     | CONV,(64,K5x5,S2), BN, LReLU   | CONV,(64,K5x5,S2), BN, LReLU   |
| 3     | CONV,(128,K5x5,S2), BN, LReLU  | CONV,(128,K5x5,S2), BN, LReLU  |
| 4     | CONV,(256,K3x3,S2), BN, LReLU  | CONV,(256,K3x3,S2), BN, LReLU  |
| 5     | CONV,(512,K3x3,S2), BN, LReLU  | CONV,(512,K3x3,S2), BN, LReLU  |
| 6     | CONV,(1024,K3x3,S2), BN, LReLU | CONV,(1024,K3x3,S2), BN, LReLU |
| 7     | FC, Sigmoid                    | FC, Sigmoid                    |

[39], ACGAN [15], COGAN [27], D2GAN [18] and InfoGAN [16].

For re-implementing the baselines, we followed their source codes with the same setting as ours. From results, it is observed that, the CNN-based methods despite preserving sharp edges, they produces blurry textures, and the perceptual quality of GAN-based methods is better, even they could improve the high frequency details. In addition, we used two well-known image quality metrics: peak signal-to-noise ratio (PSNR) and structural similarity values (SSIM) [26]. The results are given in Tables 3, 4, 5 and Figs. 2, 3, 4, 5 and 6.

## 5.1 Results and comparisons

In order to demonstrate the effectiveness of our model, extensive qualitative and quantitative performance are prepared. We also train our model with different scaling factors:  $\{4\times\}$ ,  $\{6\times\}$  and  $\{8\times\}$  between low- and high-resolution images. We used the source codes of various algorithms to evaluate the runtime on the same machine which is used to implement our model. Figure 2 shows an overview of twelve methods including the current prominent works in GAN and CNN in terms of PSNR on DIV2 K datasets which is well-suited for visual comparison, and it contains the images with sharps edges and textured regions. From the results, it observes that the GAN-based methods have a good performance on edge reconstruction; however,

**Table 3** Average PSNR/SSIM for BSD-100 dataset

| Methods    | PSNR (6×)    | SSIM (6×)     | PSNR (8×)    | SSIM (8×)     |
|------------|--------------|---------------|--------------|---------------|
| SRDenseNet | 27.17        | 0.7851        | 26.03        | 0.7843        |
| VDSR       | 28.49        | 0.8358        | 25.11        | 0.8097        |
| LapSRN     | 23.42        | 0.6957        | 22.10        | 0.6701        |
| DRNN       | 19.98        | 0.7791        | 18.52        | 0.6128        |
| DCGAN      | 25.71        | 0.8529        | 24.09        | 0.8619        |
| GP-GAN     | <b>29.98</b> | <i>0.8917</i> | <i>27.06</i> | 0.8595        |
| D2GAN      | <i>29.06</i> | <b>0.9096</b> | <i>27.17</i> | <i>0.8876</i> |
| SRGAN      | 27.81        | <i>0.8852</i> | 21.07        | 0.8067        |
| DualGAN    | <i>29.62</i> | <b>0.9107</b> | <b>27.85</b> | <b>0.8911</b> |

The baselines are selected from GAN- and CNN-based methods such as: SRDenseNet [41], VDSR [33], LapSR [34], DRNN [42], DCGAN [13], GP-GAN [43], D2GAN [18], SRGAN [4] and our DualGAN model. The highest measures are (PSNR [dB], SSIM) in bold; the second highest is in italics. [ $6\times$  and  $8\times$  scale factor]

**Table 4** Average PSNR/SSIM for CIFAR-10 dataset

| Methods        | PSNR (6×)    | SSIM (6×)     | PSNR (8×)    | SSIM (8×)     |
|----------------|--------------|---------------|--------------|---------------|
| SRDenseNet     | 24.97        | 0.7160        | 20.11        | 0.7002        |
| VDSR           | 27.34        | 0.8123        | 25.53        | 0.7891        |
| LapSRN         | 23.99        | 0.7604        | 20.49        | 0.6326        |
| DRNN           | 20.81        | 0.6129        | 19.63        | 0.6892        |
| DCGAN          | <b>29.52</b> | <i>0.8699</i> | 24.65        | 0.8384        |
| GP-GAN         | 28.83        | <i>0.8709</i> | 20.06        | 0.8295        |
| D2GAN          | 26.96        | 0.8546        | <i>25.03</i> | <i>0.8576</i> |
| SRGAN          | 26.39        | 0.7961        | 23.87        | 0.7249        |
| <b>DualGAN</b> | <b>30.17</b> | <b>0.9044</b> | <b>25.87</b> | <b>0.8611</b> |

The highest measures are (PSNR [dB], SSIM) in bold the second highest is in italics. [6 × and 8 × scale factor]

**Table 5** Average PSNR/SSIM for DIV2 K dataset

| Methods    | PSNR (6×)    | SSIM (6×)     | PSNR (8×)    | SSIM (8×)     |
|------------|--------------|---------------|--------------|---------------|
| SRDenseNet | 28.93        | <b>0.8913</b> | 25.43        | 0.7976        |
| VDSR       | 26.23        | 0.8502        | 24.59        | 0.8004        |
| LapSRN     | 23.42        | 0.6957        | 22.10        | 0.6701        |
| DRNN       | 21.56        | 0.6735        | 21.09        | 0.6537        |
| DCGAN      | 27.09        | 0.8627        | 25.64        | 0.7789        |
| GP-GAN     | 25.83        | 0.7981        | 25.06        | 0.8495        |
| D2GAN      | 27.96        | 0.8707        | 24.17        | 0.7876        |
| SRGAN      | 29.18        | <i>0.8835</i> | <i>26.09</i> | <i>0.8596</i> |
| DualGAN    | <b>29.34</b> | <i>0.8901</i> | <b>26.85</b> | <b>0.8711</b> |

The highest measures are (PSNR [dB], SSIM) in bold; the second highest is in italics [6 × and 8 × scale factor]



**Fig. 2** Visual comparison of SR results at scaling factor 8. The top images are the ground truth images. We used several baselines as D2GAN [18], ProGAN [35], UR-DGAN [38], DCGAN [13], CoGAN [27], InfoGAN [16], BEGAN [36], GMAN [7], ACGAN [15],

GoGAN [37], Johnson et al. [40], MAGAN [39] and our DualGAN. However, D2GAN and SFT-GAN are capable of generating richer and visual texture in comparing with other methods. Our model yields the better results comparing others. (Zoom in for best review)

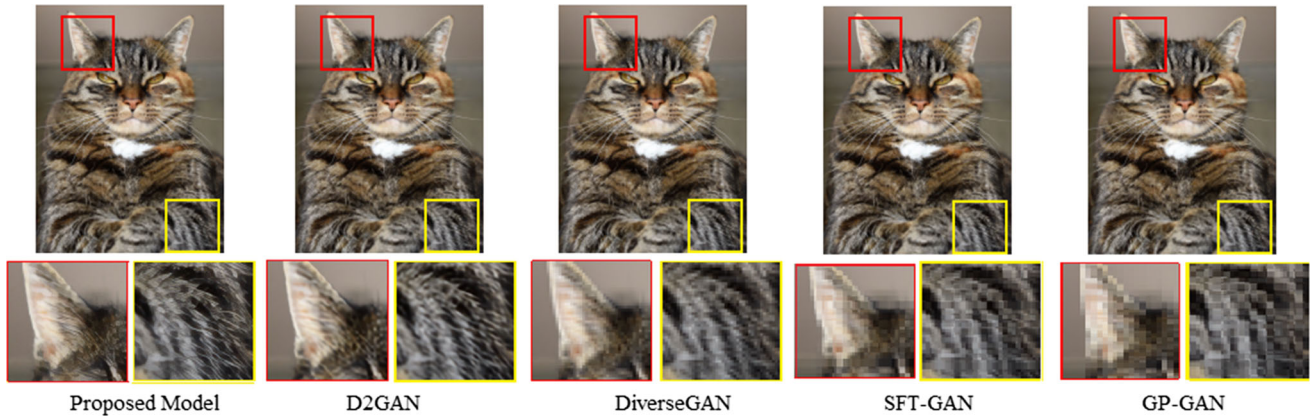


Fig. 3 Visual comparison for  $4 \times$  SR, based on different GAN structures

Fig. 4 Comparison of PSNR and SSIM values on the CIFAR-10 dataset using three different network structures: DCGAN [13], D2GAN [18] and our proposed DualGAN. The result is evaluated at  $4 \times$

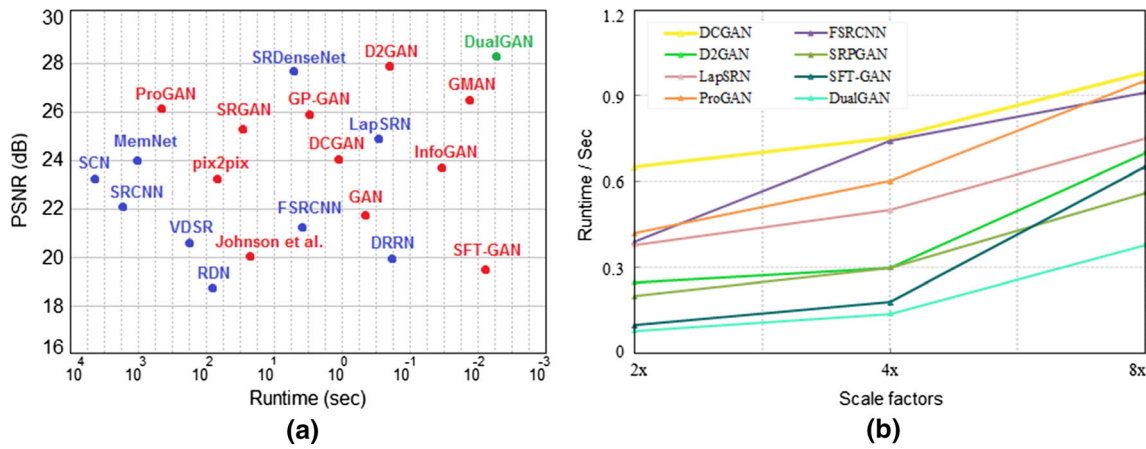
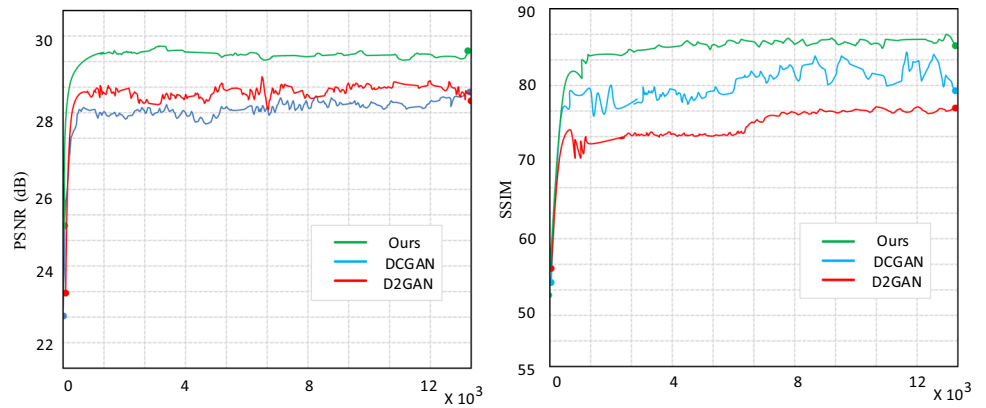


Fig. 5 (Left) Convergence of different methods for  $4 \times$  super resolutions. We set the size of input images to  $128 \times 128$  for all methods and the results evaluated on CIFAR-10 dataset. The baselines are: SRCNN [11], FSRCNN [45], VDSR [33], DRNN [42], SFT-GAN [44], RDN [46], SRDenseNet [41], SCN [10] MemNet [47], LapSRN

[34], GAN [3], SRGAN [4], ProGAN [35], DCGAN [13], GP-GAN [43], InfoGAN [16], Johnson et al. [40], D2GAN [18] and proposed DualGAN. The GAN-based methods are indicated by red, while, CNN-based methods are indicated by blue. (Right), the trade-off between runtime and upscaling scales

they suffer from blur region. Even the state of the art D2GAN [18], GoGAN [37] and DCGAN [13] does not provide clean and sharp details at the high scaling factors. While the proposed model with respect to the baselines is

able to produce the sharper edges and exhibits an acceptable results at the high scaling factors. The second best results are for ACGAN [15] and BEGAN [36], while the worse visualization results are for DCGAN [40]. Note that,





**Fig. 6** Image quality improvement and comparing with other techniques at  $4 \times$  SR. The first image in both the samples is the ground truth data; next is the proposed DualGAN. The baselines are DCGAN [13], Johnson et al. [40], LapSRN [34], MemNet [47], Original GAN [3], SRGAN [4] and ProGAN [35]. The results convey that our model

is able to generate the sharper images with respect to the other baselines. However, LapSRN has a worse performance and could not discover the high frequencies details properly. ProGAN and DCGAN are the second best results in order to generate the sharper and clean images

the results of Fig. 2 are evaluated at  $8 \times$  scaling factor. Similarly, we show visual comparison of GAN variations for  $4 \times$  in Fig. 3. It clearly observes that our method accurately reconstructs the fine lines and grid patterns.

Next, we show the quantitative results in Table 3–5, on  $4 \times$  and  $8 \times$  factors. We compared our model to several GAN- and CNN-based models, such as: SRDenseNet [41], VDSR [33], LapSR [34], DRNN [42], DCGAN [13], GP-GAN [43], D2GAN [18], SRGAN [4]. We evaluated the results on three datasets: BSD-100, CIFAR-10 and DIV2 K. The evaluation metrics are PSNR and SSIM. Our model performs favorably against the current approaches and having comparable results with GMAN [7] and StackGAN [6]. Based on BSD-100 dataset, the best results belong to GP-GAN, D2GAN and ours, at 8 scaling factor. For the CIFAR-10 dataset, at 4 scaling factor, the best results correspond to GP-GAN and DCGAN, while at the 8 scaling factor, D2GAN performs better than other baselines. Similarly, the results based on DIV2 K dataset imply that, at 4 scaling factor, SRGAN and SRDenseNet, performs better than other baselines in terms of PSNR and SSIM, while, at the 8 scaling factor, only SRGAN has a pleasant result. In sum, the methods based on GAN outpace CNN-based methods. Therefore, we can conclude that GAN-based methods are well-suited methods in image super-resolution.

In addition, to validate the effectiveness of our model comparing to other approaches, we plot the convergence curve in terms of PSNR and SSIM on the CIFAR-10 dataset.

The results are given in Fig. 4. The results convey that our model requires less iteration to achieve a good result and also have a robust performance comparing to the DCGAN and D2GAN. However, state of the art D2GAN [18] does not provide a stable performance and needs more iteration to achieve comparable performance with DCGAN [13].

*Execution time:* we evaluated the trade-offs between the runtime and performance of PSNR on the CIFAR-10 dataset for different scaling factors. Results are plotted in Fig. 5. We evaluated the results with the same machine which we tested our model. Figure 5a, shows the performance of PSNR versus runtime for 4 scaling factors. The CNN-based methods are drawn with the blue color, and the GNN-based methods are presented with the red color (*in order to make it clearer*). Figure 5b, shows the performance of PSNR with different scaling factors. From the results, it observes that the speed of our model is faster than all the existing methods and has a competitive performance with SFT-GAN [44], GMAN [7] and InfoGAN [16].

*Quality of the generated images:* another experiments are designed to show the quality of the generated images by our model against state of the art methods in Fig. 6. We selected two practically well-suited images from DIV2 K dataset for a visual comparison since they contain sharp and smooth edges. The results convey that the proposed model clearly provides better results in comparison with the others and is able to correctly reconstruct the fine structures, grid patterns and the dark spots in the image

backgrounds. The experiments proved our claim regarding the performance of the proposed DualGAN model.

## 6 Conclusion

In this paper, we propose a simple and effective framework, *DualGAN*, for fast and accurate image super resolution. The proposed model consists of two generators and discriminators which additionally extended to the GAN framework. The generators used mixture data distribution in order to generate a realistic image and the discriminators designed to accurately classify the inputs and also identify the generators that generated the wrong samples. We showed the effectiveness of our proposed model in comparison with the other variation of GAN-based methods. Our model not only has a simple implementation but also presents superior results. Using multi-generators with a mixture data distribution optimizes the networks and helps to smooth the training process. The main aspects of this work are to balance the network with a couple of generators-discriminators; proposing mixture data distribution and also train the generators with feature matching loss, which can reduce the network parameters and speed up the training. With this proposed methodology, we believe that the results are more stable and efficient rather than other popular generative models. In addition, for the future direction, we would like to estimate the number of generators and discriminators needed for a particular dataset.

**Acknowledgements** This research is partly supported by NSFC, China (U1803261, 61876107, 61572315); 973 Plan, China (2015CB856004). H. Zhou was supported by UK EPSRC under Grant EP/N011074/1, Royal Society-Newton Advanced Fellowship under Grant NA160342 and European Union's Horizon 2020 research and innovation program under the Marie-Sklodowska-Curie grant agreement No 720325.

## Compliance with ethical standards

**Conflict of interest** We have no conflict of interest to declare.

## References

- Zareapoor M, Zhang J, Yang J (2019) Towards realistic image via function learning. *Multimed Tools Appl*. <https://doi.org/10.1007/s11042-019-7361-6>
- Zareapoor M, Shamsolmoali P, Yang J (2019) Learning depth super-resolution by using multi-scale convolutional neural network. *J Intell Fuzzy Syst* 36(2):1773–1783
- Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: *Proceeding of advances in neural information processing systems*, pp 2672–2680
- Ledig C, Theis L, Huszar F, Caballero J, Aitken AP, Tejani A, Totz J, Wang Z, Shi W (2016) Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, vol. abs/1609.04802, 2016. [Online]. <http://arxiv.org/abs/1609.04802>
- Reed S, Akata Z, Yan X, Logeswaran L, Schiele B, Lee H (2016) Generative adversarial text-to-image synthesis. In: *Proceedings of ICML*, pp 1060–1069
- Zhang H, Xu T, Li H, Zhang S, Huang X, Wang X, Metaxas DN (2017) StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. In: *Proceeding of the ICCV*, pp 5907–5915
- Durugkar IP, Gemp I, Mahadevan S (2016) Generative multi-adversarial networks. *ICLR. CoRR*, abs/1611.01673
- Zareapoor M, Celebi ME, Yang J (2019) Diverse adversarial network for image super-resolution. *Signal Process Image Commun* 74:191–200. <https://doi.org/10.1016/j.image.2019.02.008>
- Ding L, Zhang H, Xiao J et al (2018) An improved image mixed noise removal algorithm based on super-resolution algorithm and CNN. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-018-3777-6>
- Wang Z, Liu D, Yang J, Han W, Huang T (2015) Deep networks for image super-resolution with sparse prior. In: *ICCV*
- Dong C, Loy CC, He K, Tang X (2016) Image super-resolution using deep convolutional networks. *IEEE Trans Pattern Anal Mach Intell* 38(2):295–307
- Zareapoor M, Jain DK, Yang J (2018) Local spatial information for image super-resolution. *Cogn Syst Res* 52:49–57
- Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. In: *Proceeding of international conference on learning representations* [arXiv:1511.06434](https://arxiv.org/abs/1511.06434)
- Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X (2016) Improved techniques for training GANs. In: *Proceeding of the NIPS*, pp 2234–2242
- Odena A, Olah C, Shlens J (2017) Conditional image synthesis with auxiliary classifier GANs. In: *International conference on machine learning (PMLR)*, pp 2642–2651
- Chen X, Duan Y, Houthoofd R, Schulman J, Sutskever I, Abbeel P (2016) Infogan: interpretable representation learning by information maximizing generative adversarial nets. In: *Advances in neural information processing systems*
- Arjovsky M, Chintala S, Bottou L (2017) Wasserstein generative adversarial networks. In: *Proceedings of the 34th international conference on machine learning*, pp 214–223
- Nguyen TD, Le T, Vu H, Phung D (2017) Dual discriminator generative adversarial nets. In: *Advances in neural information processing systems* 29 (NIPS) (accepted)
- Arora S, Ge R, Liang Y, Ma T, Zhang Y (2017) Generalization and equilibrium in generative adversarial nets (gans). *arXiv preprint* [arXiv:1703.00573](https://arxiv.org/abs/1703.00573)
- Tolstikhin I, Gelly S, Bousquet O, Simon-Gabriel C-J, Schölkopf B (2017) Adagan: boosting generative models. *arXiv preprint* [arXiv:1701.02386](https://arxiv.org/abs/1701.02386)
- Ghosh A, Kulharia V, Namboodiri VP, Torr PHS, Dokania PK (2017) Multi-agent diverse generative adversarial networks. In: *Proceeding of the CVPR*, pp 8513–8521
- Wang X, Gupta A (2016) Generative image modeling using style and structure adversarial networks. *arXiv preprint* [arXiv:1603.05631](https://arxiv.org/abs/1603.05631)
- Yang J, Kannan A, Batra D, Parikh D (2017) Lr-gan: layered recursive generative adversarial networks for image generation. *arXiv preprint* [arXiv:1703.01560](https://arxiv.org/abs/1703.01560)
- Denton E, Chintala S, Szlam A, Fergus R (2015) Deep generative image models using a Laplacian pyramid of adversarial networks. In: *Proceeding the NIPS*, pp 1486–1494

25. Burt PJ, Adelson EH (1987) The Laplacian pyramid as a compact image code. In: Readings in computer vision. Elsevier, pp 671–679
26. Chen R, Qu Y, Li C et al (2018) Single-image super-resolution via joint statistic models-guided deep auto-encoder network. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-018-3886-2>
27. Liu M-Y, Tuzel O (2016) Coupled generative adversarial networks. In: Proceedings of the advances in neural information processing systems (NIPS 2016), Barcelona, Spain, pp 469–477
28. Kliger M, Fleishman S (2018) Novelty detection with GAN. [arXiv:1802.10560v1](https://arxiv.org/abs/1802.10560v1) [cs.CV]
29. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: ICCV
30. Maas A, Hannun AY, Ng AY (2013) Rectifier nonlinearities improve neural network acoustic models
31. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M et al (2016) TensorFlow: a system for large-scale machine learning. In: OSDI, vol 16, pp 265–283
32. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. *CoRR*, vol. abs/1412.6980
33. Kim J, Lee JK, Lee KM (2016) Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the CVPR, pp 1646–1654
34. Lai WS, Huang J-B, Ahuja N, Yang M-H (2017) Deep Laplacian pyramid networks for fast and accurate superresolution. In: CVPR, pp 624–632
35. Wang Y, Perazzi F, Williams BM, Hornung AS, Hornung OS, Schroers C (2017) A fully progressive approach to single-image super-resolution. [arXiv:1804.02900v2](https://arxiv.org/abs/1804.02900v2)
36. Berthelot D, Schumm T, Metz L (2017) Began: boundary equilibrium generative adversarial networks. *CoRR*, abs/1703.10717
37. Juefei-Xu F, Boddeti VN, Savvides M (2017) Gang of gans: generative adversarial networks with maximum margin ranking. *arXiv preprint* [arXiv:1704.04865](https://arxiv.org/abs/1704.04865)
38. Metz L, Poole B, Pfau D, Sohl-Dickstein J (2016) Unrolled generative adversarial networks. *arXiv preprint* [arXiv:1611.02163](https://arxiv.org/abs/1611.02163)
39. Wang R, Cully A, Chang HJ, Demiris Y (2017) Magan: Margin adaptation for generative adversarial networks. *arXiv preprint* [arXiv:1704.03817](https://arxiv.org/abs/1704.03817)
40. Johnson J, Alahi A, Fei-Fei L (2016) Perceptual losses for real-time style transfer and super-resolution. In: ECCV
41. Huang G, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: CVPR
42. Tai Y, Yang J, Liu X (2017) Image super-resolution via deep recursive residual network. In: Proceedings of the CVPR, pp 2790–2798
43. Wu H, Zheng S, Zhang J, Huang K (2017) GP-GAN: towards realistic high-resolution image blending. [arXiv:1703.07195v2](https://arxiv.org/abs/1703.07195v2)
44. Wang X, Yu K, Dong C, Loy CC (2018) Recovering realistic texture in image super-resolution by deep spatial feature transform. In: CVPR. [arXiv:1804.02815v1](https://arxiv.org/abs/1804.02815v1)
45. Dong C, Loy CC, Tang X (2016) Accelerating the super-resolution convolutional neural network. In: European conference on computer vision (ECCV), pp 391–407
46. Zhang Y, Tian Y, Kong Y, Zhong B, Fu Y (2018) Residual dense network for image super-resolution. In: CVPR
47. Tai Y, Yang J, Liu X, Xu C (2017) Memnet: a persistent memory network for image restoration. In: ICCV

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.