

Perceptual image quality using Dual generative adversarial network

Masoumeh Zareapoor ^a, Huiyu Zhou ^b, Jie Yang ^a

^aSchool of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China

^bDepartment of Informatics, University of Leicester, United Kingdom

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 on 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 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 distribution 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 but still there is no unique solution exists, in particular for high magnification ratios. Each pixel loss which used by the existence approaches do not properly capture perceptual variances between output and input images [5][35]. 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. [15]. 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 [11][20][22]. 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 from GMAN [8] and [33], we propose to employ two generators with two discriminators, which are based on an image-to-image model. The proposed architecture termed as *DualGAN*, as shown in Figure 1. Similar to the regular GAN, the objective of the generators, are to increase the mistake of the discriminator. Moreover, unlike the other GAN variations that using 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, act 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 towards 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. Section 3 and 4, presents the proposed architecture with the mixture generator-discriminator extension to the GAN framework. Section 5 contains the experimental results and related discussions. Finally, Section 6 concludes the paper. It's 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 [1][3][4][6][16]. The main concept of Generator adversarial network (GAN) [15] 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, determines whether is it fake (generated by generator) or from the real image.

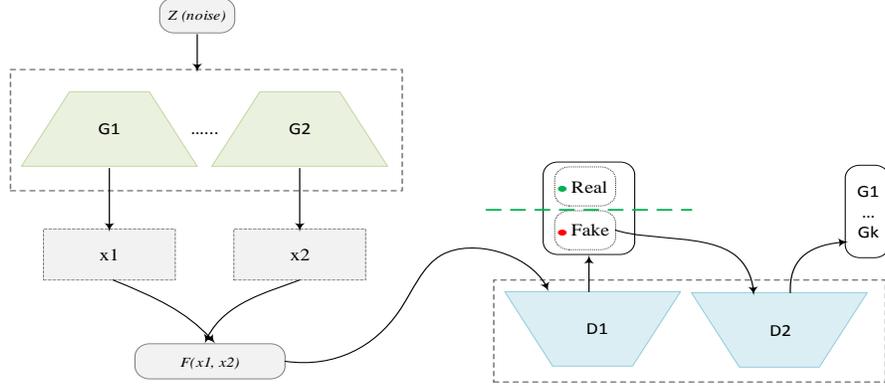


Figure 1: DualGAN consists of a pairs of generators and discriminators; G_1, G_2 and D_1, D_2 . The generative model with a couple generators trains for generating realistic artificial images and the discriminative model with a couple discriminators, along with determining whether an image is real or fake, it also identify the realted generators that generated 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 reduce the model parameters at the optimal level.

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 [6][11][29]. In the standard GAN, there is no way to control what to be generated, since there is no information for the learning generators. However, [2] proposed a new method in order to define more condition for the generator so that the generated image can be 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 [7], DCGAN [16], WGAN [13], ImprovedGAN [29] and DGAN [33]. These methods in fact are straight-forward to design and implementation. Recent attempts to improve the GANs results and solve the training issues by training additional generators and discriminators. D2GAN [17] is a new approach which using 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. [8], which uses several discriminators to improve the generator learning. Recently, Arora et al. [14] 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. [19] 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 proposed by Ghosh et al. [25]. It trains multiple generators with a multi-class discriminator. Their model designed to improve the objective function of discriminator to push multiple generators towards generating diverse modes. Reed et al. [20] also proposed a GAN-based method which is able to generate 64^2 images and can barely generate intense object details. Accordingly, StackGAN [22] is proposed to stack two GANs in order to improve the [20] by generating 256^2 images. SS-GAN [26] 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 then produce an output image. In [34], the author present 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 refining 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 [23] is another method that uses multiple generators to generate images from coarse to fine using Laplacian pyramid [27]. 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 [33][46]. In term 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 produces multiple samples and it would increase the chance of sharing more details with model distributions. Multiple generators focus to completing the missing details for producing the higher resolution images. Multiple discriminators allows 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 $\{D_j, j \in \hat{S}\}$, where D_j is a function from binary space $[0, 1]$. Training the discriminator enforce the output to get high value 1 when x is from distribution D_{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 [15]:

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

Practically the above equation will be solved by the following material.

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

where θ_d, θ_g are the discriminator and generator parameters, α indicates the learning rate and t is number of iteration. The Proposed DualGAN is illustrated in Figure 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 $x_1 \sim p_{x_1}$, $x_2 \sim p_{x_2}$ distribution, respectively. We denote the distribution of the generators as, p_{g_1}, p_{g_2} and both the generators using multilayer perceptrons [36]:

$$g_1(z) = g_1^m(g_1^{m-1}(\dots g_1^2(g_1^1(z)))) \text{ and for } g_2(z) = g_2^n(g_2^{n-1}(\dots g_2^2(g_2^1(z))))); \quad (3)$$

where m and n indicates the number of layers in two generators g_1, g_2 with the condition $m = \text{or } \neq n$. 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 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 sharing the weights. In fact, the generators use the shared high-level representation for fooling the discriminator. Salimans et al. [29] 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 [29], we used Feature matching loss in order to train the mixture data distribution in the generators.

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

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 [47]. 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 determine the real or fake samples, and another one acts as classifier to classify that, the samples are from which generators. Two discriminators will be defined as:

$$d_1(x_1) = d_1^t(d_1^{t-1}(\dots d_1^2(d_1^1(x_1)))) \text{ and } d_2(x_2) = d_2^q(d_2^{q-1}(\dots d_2^2(d_2^1(x_2)))); \quad (5)$$

where t, q is the number of layers in the d_1 and d_2 discriminators. d_1 Maps the input to a probability scores and then estimate 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 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 (.) will be as:

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

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, 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 [15] with alternating gradient update steps [29].

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 otherside, 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 $\widehat{D}_1, \widehat{D}_2$ yields the following equations:

$$\widehat{D}_1 = \alpha \times \frac{p_{true\ data}(x)}{p_G(Mix)} ; \quad \widehat{D}_2 = \beta \times \frac{p_G(Mix)}{\sum_{j=1}^2 \pi p_{G_j}(Mix)} \text{ where } p_G(Mix) = \sum_{j=1}^2 \pi(p_{G_j}(x)) \quad (7)$$

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

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

As the original GAN, the objective of generators are to minimize JS divergence between the data distributions while Maximize it between the generators [29]. We verify the maximal loss function by setting $D_1 = D_2 = 0$; $\frac{\alpha p_{true\ data}(x)}{D_1} - p_G(x)$, and $\frac{\beta p_G(x)}{D_2} - p_{true\ data}(x) = 0$. Note that, the

discriminator d_1 takes input from G and determine whether the samples are fake or real data. Next, the fake samples take 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 multiclass-classifier (depends on the number of generators, i.e. in this paper, both the discriminator 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 [36] 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 [38], 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 standard convolutional layer (CONV) except the last layers which are based on fully connection layers (FC). We observed that Leaky rectified linear unit (LReLU) [31] works better rather than ReLU, especially for the diverse samples which is 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 consists of “six” fractional convolutional layers while the discriminators have six convolutional layer plus two fully connection layers. The generators and discriminators are parameterized by ϑ_G, ϑ_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 [39] is used to implement our model, Adam optimizer [12] and Momentum set to 0.0002 [12] and 0.5 respectively, also weights initialized from an isotropic Gaussian, μ (0, 0.01) and zero biases. The details of the networks are given in Table 1 and Table 2. In addition, it’s worth to mention that, we implemented the proposed model in a system with following features; Intel i7-6850K CPU with a 64 GB Ram and an NVIDIA GTX Geforce 1080 Ti GPU and the operating system is Ubuntu 16.04.

Table 1. Designed generative model in DualGAN. FL-CONV, indicates the fractional length convolutional, BN is the batch normalization and PReLU represent 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

5. Experimental evaluation

We conduct a series of experiments to evaluate the proposed model and compare it with other realted 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, DIV2K 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 [4], VDSR [24], LapSRN [30], and also several known variation of GAN including, DCGAN [16], ProGAN [40], BEGAN [32], GOGAN [10], Unrolled GAN [21], GMAN [8], MAGAN [28], ACGAN [2], COGAN [36], D2GAN [17] and InfoGAN [7].

Table 2. Designed discriminative model in DualGAN. Conv, indicates 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

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 of preserving sharp edges, but 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) [46]. The results are given in Table 3, Table 4, Table 5, Figure 2, Figure 3, Figure 4, Figure 5 and Figure 6.

5.1. Results and Comparisons

In order to demonstrate the effectiveness of our model extensive qualitative and quantitative performance is 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 term of PSNR on DIV2K 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, they suffers from blur region. Even the state of the art D2GAN [17], GoGAN [10] and DCGAN [16] doesn't 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 exhibit an acceptable results at the high scaling factors. The second best results if for ACGAN [2] and BEGAN [32], while, the worse visualization results is for DCGAN and [37]. Note that, the results of Figure 2 is evaluated at $8 \times$ scaling factor. Similarly, we show visual comparison of GAN variations for $4 \times$ in Figure 3. It clearly observes that, our method accurately reconstructs the fine lines and grid patterns.



Figure 2. Visual comparison of SR results at scaling factor 8. The top images are the Ground truth images. We used several baselines as D2GAN [17], ProGAN [40], UR-DGAN [21], DCGAN [16], CoGAN [36], InfoGAN [7], BEGAN [32], GMAN [8], ACGAN [2], GoGAN [10], Johnson et al. [37], MAGAN [28] and our DualGAN. However, D2GAN and SFT-GAN are capable of generating richer and visual texture in comparing to other methods. Our model yields the better results comparing others. (Zoom in for best review).

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 [18], VDSR [24], LapSR [30], DRNN [43], DCGAN [16], GP-GAN [9], D2GAN [17], SRGAN [11]. We evaluated the results on three datasets; BSD-100, CIFAR-10 and DIV2K. The evaluation metrics are, PSNR and SSIM. Our model performs favorably against the current approaches, and having comparable results with GMAN [8] and StackGAN [22]. 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 corresponds to GP-GAN and DCGAN, while at the 8 scaling factor, D2GAN performs better than other baselines. Similarly, the results based on DIV2K 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 have 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.

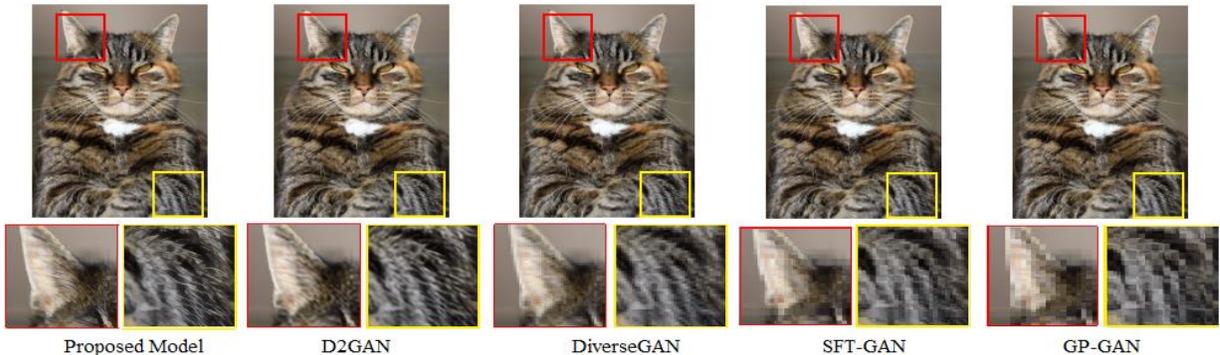


Figure 3. Visual comparison for $4 \times$ SR, based on different GAN structure.

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

Figure 4. The results convey that, our model requires less iteration to achieve a good result, also have a robust performance comparing to the DCGAN and D2GAN. However, state of the art D2GAN [17] does not provide a stable performance and needs more iteration to achieve comparable performance with DCGAN [16].

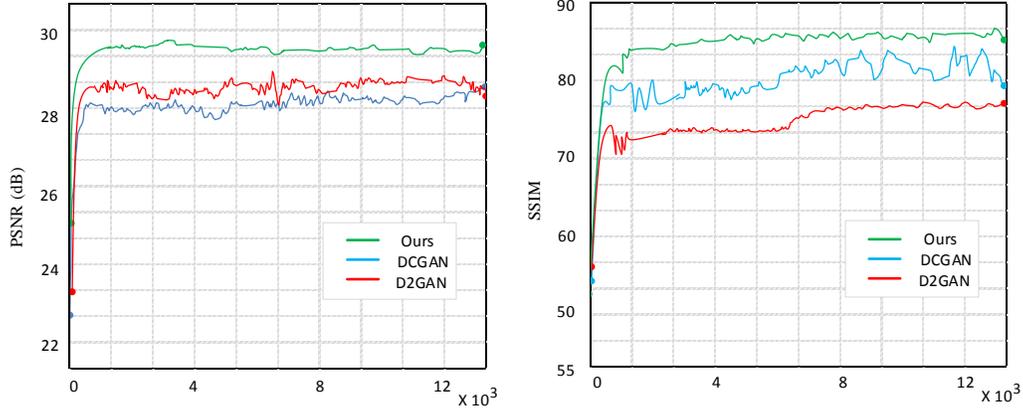


Figure 4. Comparison of PSNR and SSIM values on the CIFAR-10 dataset using three different network structures. DCGAN [16], D2GAN [17], and our proposed DualGAN. The result is evaluated at 4 \times .

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 Figure 5. We evaluated the results with the same machine which we tested our model. Figure 5.a, 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 5. b, 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 have a competitive performance with SFT-GAN [41], GMAN [8] and InfoGAN [7].

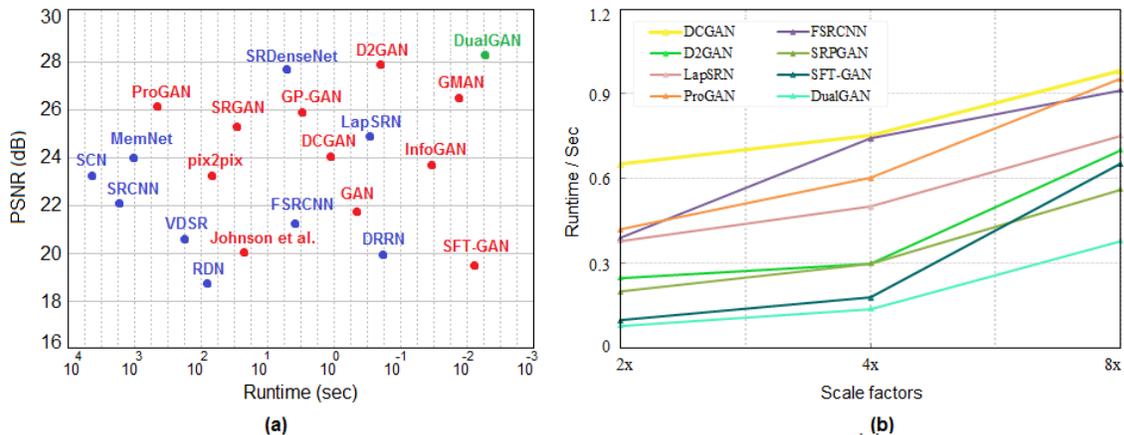


Figure 5. (Left), Convergence of different methods for 4 \times super resolutions. We set the size of input images to 128×128 for all methods and the results evaluated on CIFAR-10 dataset. The baselines are; SRCNN [4], FSRCNN [42], VDSR [24], DRNN [43], SFT-GAN [41], RDN [44], SRDenseNet [18], SCN [3] MemNet [45], LapSRN [30], GAN [15], SRGAN [11], ProGAN [40], DCGAN [16], GP-GAN [9], InfoGAN [7], Johnson et al. [37], D2GAN [17], and proposed DualGAN. The GAN based methods indicates by **red**, while, CNN-based methods indicated by **blue**. **(Right),** the trade-off between runtime and upscaling scales.

Quality of the generated images; another experiments is designed to show the quality of the generated images by our model against state of the art methods in Figure 6. We selected two practically well-suited images from DIV2K dataset for a visual comparison since they contain sharp and smooth edges. The results convey that, the proposed model clearly provide 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.

Table 3: Average PSNR/SSIM for BSD-100 dataset. The baselines are selected from GAN and CNN based methods such as; SRDenseNet [18], VDSR [24], LapSR [30], DRNN [43], DCGAN [16], GP-GAN [9], D2GAN [17], SRGAN [11] and our DualGAN model. The highest measures are (PSNR [dB], SSIM) in bold and blue, the second highest in green. [6× and 8× scale factor].

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	29.98	0.8917	27.06	0.8595
D2GAN	29.06	0.9096	27.17	0.8876
SRGAN	27.81	0.8852	21.07	0.8067
DualGAN	29.62	0.9107	27.85	0.8911

Table 4: Average PSNR/SSIM for CIFAR-10 dataset. The highest measures are (PSNR [dB], SSIM) in bold and blue, the second highest in green. [6× and 8× scale factor].

Methods	PSNR (6 x)	SSIM (6 x)	PSNR (8 x)	SSIM (8 x)
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	29.52	0.8699	24.65	0.8384
GP-GAN	28.83	0.8709	20.06	0.8295
D2GAN	26.96	0.8546	25.03	0.8576
SRGAN	26.39	0.7961	23.87	0.7249
DualGAN	30.17	0.9044	25.87	0.8611

Table 5: Average PSNR/SSIM for DIV2K dataset. The highest measures are (PSNR [dB], SSIM) in bold and blue, the second highest in green. [6× and 8× scale factor].

Methods	PSNR (6 x)	SSIM (6 x)	PSNR (8 x)	SSIM (8 x)
SRDenseNet	28.93	0.8913	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	0.8835	26.09	0.8596
DualGAN	29.34	0.8901	26.85	0.8711

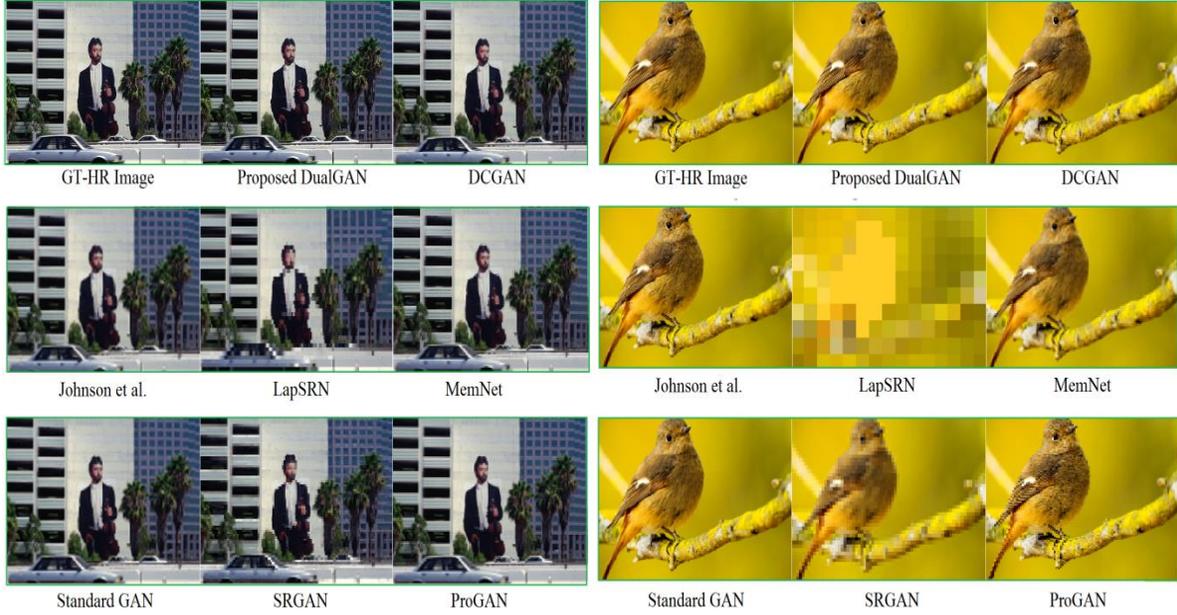


Figure 6. Image quality improvement and comparing with other techniques at $4\times$ SR. The first image in both the samples is the ground truth data; next is the proposed DualGAN. The baselines are DCGAN [16], Johnson et al. [37], LapSRN [30], MemNet [45], Original GAN [15], SRGAN [11] and ProGAN [40]. The results convey that, our model is able to generate the sharper images with respect to the other baselines. However, LapSRN have a worse performance and couldn't discover the high frequencies details properly. ProGAN and DCGAN are the second best results in order to generating the sharper and clean images.

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 have 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 is 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, 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.

Acknowledgment: This research is partly supported by NSFC, China (U1803261, 61876107, 61572315); and 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.

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

References

- [1] Ding, L., Zhang, H., Xiao, J. et al. *Neural Comput & Applic* (2018). <https://doi.org/10.1007/s00521-018-3777-6>.
- [2] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier GANs, *International Conference on Machine Learning*, (PMLR),(2017), pp. 2642-2651.
- [3] Wang, Z., Liu, D., Yang, J., Han, W., Huang, T.: Deep networks for image super-resolution with sparse prior. In: *ICCV*. (2015).
- [4] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2016.
- [5] Zareapoor, M., Zhang, J. & Yang, J. *Multimed Tools Appl* (2019). <https://doi.org/10.1007/s11042-019-7361-6>.
- [6] M. Zareapoor, D. K. Jain, J. Yang. Local spatial information for image super-resolution. *Cognitive Systems Research*, 52 (2018), pp. 49-57.
- [7] X. Chen, Y. Duan, R. Houthoof, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2016.
- [8] Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative multi-adversarial networks. *arXiv preprint arXiv:1611.01673*, 2016.
- [9] H. Wu, S. Zheng, J. Zhang, K. Huang, GP-GAN: Towards Realistic High-Resolution Image Blending, (2017). *arXiv:1703.07195v2*.
- [10] F. Juefei-Xu, V. N. Boddeti, and M. Savvides. Gang of gans: Generative adversarial networks with maximum margin ranking. *arXiv preprint arXiv:1704.04865*, 2017.
- [11] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, vol. abs/1609.04802, 2016. [Online]. Available: <http://arxiv.org/abs/1609.04802>.
- [12] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014.
- [13] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: *Proceedings of the 34th International Conference on Machine Learning*, (2017), pp.214–223.
- [14] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). *arXiv preprint arXiv:1703.00573*, 2017
- [15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: *Proceeding of Advances in Neural Information Processing Systems*, (2014), pp. 2672–2680.
- [16] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, in: *Proceeding of International Conference on Learning Representations*, (2015), *arXiv:1511.06434*.
- [17] Tu Dinh Nguyen, Trung Le, Hung Vu, and Dinh Phung. Dual discriminator generative adversarial nets. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pp. accepted, 2017.
- [18] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, July 2017
- [19] Ilya Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, Bernhard Schölkopf. Adagan: Boosting generative models. *arXiv preprint arXiv:1701.02386*, 2017
- [20] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, H. Lee, Generative adversarial text-to-image synthesis, *proceedings of ICML*, (2016), pp. 1060-1069.
- [21] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016..
- [22] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, D. N. Metaxas, StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. *Proceeding of the ICCV*, (2017), pp.5907-5915.
- [23] E. Denton, S. Chintala, A. Szlam, R. Fergus, Deep generative image models using a laplacian pyramid of adversarial networks, in: *Proceeding the NIPS*, (2015), pp.1486–1494.
- [24] J. Kim, J. K. Lee, K. M. Lee. Accurate image super-resolution using very deep convolutional networks, in: *Proceedings of the CVPR*, (2016), pp.1646–1654.
- [25] A. Ghosh, V. Kulharia, V. P. Nambodiri, P. H. S. Torr, P. K. Dokania. Multi-agent diverse generative adversarial networks, *proceeding of the CVPR*, (2017), pp. 8513-8521.
- [26] X. Wang, A. Gupta, “Generative image modeling using style and structure adversarial networks,” *arXiv preprint arXiv:1603.05631*, 2016

- [27]P. J. Burt and E. H. Adelson, “The laplacian pyramid as a compact image code,” in Readings in Computer Vision. Elsevier, 1987, pp. 671–679.
- [28]Ruohan Wang, Antoine Cully, Hyung Jin Chang, and Yiannis Demiris. Magan: Margin adaptation for generative adversarial networks. arXiv preprint arXiv:1704.03817, 2017
- [29]T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen. Improved techniques for training gans, proceeding of the NIPS, (2016), pp. 2234-2242.
- [30]W.S. Lai, J.-B. Huang, N. Ahuja, M.-H. Yang. Deep laplacian pyramid networks for fast and accurate superresolution. In CVPR, pages 624–632, 2017
- [31]Maas A., Hannun A.Y & Ng A.Y. 2013. “Rectifier Nonlinearities Improve Neural Network Acoustic Models.
- [32]David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717, 2017.
- [33]Mark Kliger, Shachar Fleishman. Novelty Detection with GAN. arXiv:1802.10560v1 [cs.CV], 2018.
- [34]J. Yang, A. Kannan, D. Batra, and D. Parikh, “Lr-gan: Layered recursive generative adversarial networks for image generation,” arXiv preprint arXiv:1703.01560, 2017
- [35]Zareapoor, M., Shamsolmoali, P., Yang, J., Learning depth super-resolution by using multi-scale convolutional neural network. Journal of Intelligent & Fuzzy Systems, (2019), vol. 36, no. 2, pp. 1773-1783.
- [36]Ming-Yu. Liu, Oncel Tuzel, “Coupled generative adversarial networks,” in Advances in neural information processing systems, 2016, pp. 469–477
- [37]J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016
- [38]Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.
- [39]Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. TensorFlow: A system for large-scale machine learning. 2016.
- [40]Y. Wang, F. Perazzi, B.M. Williams, A.S. Hornung, O.S. Hornung, C. Schroers. A Fully Progressive Approach to Single-Image Super-Resolution, (2017), arXiv:1804.02900v2
- [41]X. Wang, K. Yu, C. Dong, C. C. Loy. Recovering Realistic Texture in Image Super-resolution by Deep Spatial Feature Transform. CVPR, 2018. arXiv:1804.02815v1.
- [42]C. Dong, C. C. Loy, and X. Tang. Accelerating the Super-Resolution Convolutional Neural Network, European Conference on Computer Vision (ECCV). (2016), pp. 391-407.
- [43]Y. Tai, J. Yang, X. Liu. Image super-resolution via deep recursive residual network, proceedings of the CVPR, (2017), pp. 2790-2798.
- [44]Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: CVPR. (2018).
- [45]Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration. In: ICCV. (2017).
- [46]Chen, R., Qu, Y., Li, C. et al. Neural Comput & Applic (2018). <https://doi.org/10.1007/s00521-018-3886-2>.