Signal Processing: Image Communication 74 (2019) 191–200

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

Diverse adversarial network for image super-resolution ☆

Masoumeh Zareapoor [a,b], M. Emre Celebi [c], Jie Yang [a,*]

[a] Department of Automation, Shanghai Jiao Tong University, Shanghai, China
[b] Department of Computer Science, Tokyo University of Technology, Tokyo, Japan
[c] Department of Computer Science, University of Central Arkansas (UCA), Conway, AR, USA

ARTICLE INFO

Keywords:
Super-resolution
Adversarial network
Diverse GAN
Deep learning

ABSTRACT

Recently, there is a fast growth in Generative adversarial network and many works have appeared focusing not only images but also videos. Despite of remarkable success of GAN in image super resolution, it suffers from the major problem of poor perceptual quality. While employing a GAN for super resolution, it tends to generate over-smoothed images that lacks high frequency textures and do not look natural. We propose an intuitive generalization to Generative Adversarial Network and its conditional variations to address the problem of image super-resolution and improves the test quality of images. DGAN is a diverse GAN architecture incorporating multiple generators and a single discriminator. The main intuition is to employ multiple generators, instead of using a single one as in the original GAN. To enforce that multiple generators produce diverse samples, the discriminator trains a loss function to distinguish between real and fake samples by designed margins, and multiple generators alternately produce realistic samples by minimizing their losses. In fact, this paper addresses 2 main challenges; recovering realistic texture low resolution images and speed up the training process. We perform extensive experiments and compare the proposed model with other variants of GAN to demonstrate the efficiency and stability of the proposed model in both quantitative and qualitative benchmarks.

1. Introduction

Super-resolution (SR) is a technique that refers to obtain a high resolution image form its low-resolution version. Previously, this technology was not as attractive as it is today. However, over time with the growth of technologies, super-resolution has been evolved in many crucial applications such as remote sensing [1], object recognition [2], security surveillance [3], and medical imaging [4]. High resolution images can easily produce their corresponding low resolution (LR) images by using resolution degradation. However, inverse mapping, restoration from LR to HR images is a challenging task due to the lack of image details and sharpness edges. Recently, large numbers of super-resolution methods have been proposed and those which use Deep learning are superior. In the last four years, since the advent of the pioneering work [5], many progresses has been done on super-resolution models and several methods have been proposed not only for images but also for videos and range images, which mostly are based on convolution neural network (CNN). Even though the results of the current CNN based methods are largely blurry and over-smoothed, because they have not fully exploited all features from the original input image (low-resolution), and the fine details cannot be recovered [6–9]. Therefore, it is still highly challenging to obtain a high quality image from corresponding low-resolution image. Inspired by CNN, recently generative adversarial network (GAN) [10] has demonstrated impressive performance and gained immense popularity in a variety of computer vision tasks. Briefly, GANs are known to generate sharp and plausible images. It is comprised of two networks: a generator G and a discriminator D, where both are involved in minimax game. In fact, the discriminator learns to distinguish between the generated samples (from generator distribution p_g) and the real data points (from ground truth distribution p_d), while, the generator learns to generate new samples and maximize the mistake of the discriminator. In the GAN model each network wishes to minimize its own cost function, i.e. $f^D\left(\theta^D, \theta^G\right)$ for the discriminator and $f^G(\theta^D, \theta^G)$ for the generator. Despite of significant success of GANs, it suffers from a major problem of perceptual quality and training instability [7,11,12]. While employing a GAN for super resolution, it tends to generate blurry and over-smoothed images that lacks high frequency textures and thus do not look natural. According to results, theoretically, convergence guarantees the generator learning the right data, but, practically, it is difficult to reach this claim. In addition, due to these complex nets, the GAN architecture is unstable and it is crucial to set up a network in the best way possible. To effectively settle the current issues in

☆ This research is partly supported by NSFC, China (U1803261, 61876107, 61572315); and 973 Plan, China (2015CB856004).
* Corresponding author.
E-mail address: jieyang@sjtu.edu.cn (J. Yang).

https://doi.org/10.1016/j.image.2019.02.008
Received 23 October 2018; Received in revised form 16 February 2019; Accepted 18 February 2019
Available online 6 March 2019

GAN based super-resolution models, we propose a new GAN model, which is based on an image-to-image model. Based on recent works in GAN [13,14], we propose to use multiple generators instead of using one such that each generator aims to maximize the mistakes of the common discriminator. We call this architecture the diverse GAN, as shown in Fig. 1. We believe that in this way, the model can produce many diverse samples and also every generator may share different information which will be useful for the discriminator. However, using multiple generators may lead to trivial solution, wherein, all the generators train to produce a similar samples and the discriminator receives an overloaded details. To solve this problem we design the discriminator by adapting least square function as loss function that along with finding the real and fake samples, also inform the generator that the generated images are fake. In addition, in order to improve the learning process and increase the model stability we propose to organize a gradual learning strategy by breaking the difficult generative tasks into sub-problem. In this paper, we prove that, combining the diverse GAN with the gradual learning strategy allows us to generate plausible samples and improve the image's quality at the high scale factor (up to $\times 8$). Our contributions are four-fold. We proposed a new generative adversarial network by using multiple generators and a single discriminator. To control the diversity samples which generated by different generators, we used least square function as loss function for the discriminator. We also provide sufficient analysis and show that the proposed modification in objective function of discriminator push the generators to learn together as a mixture model. The training process in GAN is usually unstable and sensitive to the data distribution; here we address this problem by using gradual learning strategy from small to large. We analyze proposed DGAN through extensive experiments and compare it with other variation of GAN, and empirically show that our model along with outpacing all other GAN models, also it is able to generate high quality images. The rest of this paper is organized as follows. In Section 2, we discussed the related works. Section 3 is a brief description about GAN architecture, and the proposed model is presented in Section 4. Section 5 shows the experimental results and evaluation results. Finally, Section 6 concludes the paper.

2. Related works

In this section, we present a brief description of the existing methods and the background concepts, which are helpful for understanding our model. The Generator adversarial network (GAN) was first introduced by Goodfellow et al. [10] and the main idea behind it was to define a mutual game between two networks: discriminator D and generator G. The generator input is noise that generates samples as output. While the discriminator receives the real and the generated samples, it is optimized to distinguish the noise (e.g., fake images) from the real images. Recent overview papers on SR techniques include [15]; we refer the interested readers for more details. WGAN is a recent techniques which proposed by Arjovsky et al. [16]. The authors claim that the difficulties in GAN training are due to JS divergence and thus they proposed a new model termed Wasserstein, by using Kantorovich–Rubinstein duality [17]. Another related approach is BEGAN [18] which build upon WGAN by using an autoencoder based equilibrium enforcing technique along with the Wasserstein distance. To successfully train GAN many tricks is employed such as; thoroughly select an appropriate architecture [19], minibatch discrimination [20], and noise injection [21]. In addition, several hierarchical GANs have been proposed [22,23], which define a generator and a discriminator for each level of the image pyramid. Hoang et al. [24] proposed a GAN variation wherein it is able to train many generators and discover different modes of the data. This strategy allows the network to grow hierarchically and generate real-like samples. The authors reported the superiority of their proposed model as compared to others, and it can avoid the mode collapsing problem. Liu et al. [25] presented Coupled GAN for improving the image resolution. The proposed methods contains two generators with shared parameters to learn the joint

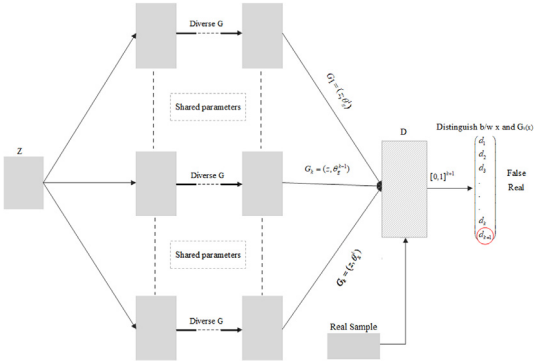

Fig. 1. Diverse GAN architecture used k generators and a binary discriminator trained end to end. The output of discriminator is k + 1, which signify the k number of generators and real data distribution.

distribution of the data. Wang et al. [26] used multiple discriminators in their architecture. Their work is inspired by Durugkar et al. [27], who used one generator and multiple discriminators. The authors claim that their results outpace the [22]. Ghosh et al. [28] also proposed a model for improving the above techniques by using multiple generators and a single discriminator. Isola et al. [29] also proposed a model, which relies on the conditional GAN and aims to transfer images from one representation to another. Another approach which is proposed by [13] is to unroll the optimization of discriminator in order to create a surrogate objective for the updating generator during training process. In [20], the authors presented a new GAN-based framework for semi-supervised learning, wherein the discriminator network not only classifies the fake images from the real ones, but also finds the probabilities of belonging to each class. Another work that exploits deep layers of convolution for GAN architecture is called DCGAN, introduced by Radford et al. [19]. We pointed out that another method, which is called LAPGAN (Laplacian pyramid of generative adversarial networks), is proposed by Denton et al. [11]. Their model constructs a Laplacian pyramid to generate multi-resolution images from low-resolution images. Nowozin et al. [30] mentioned that the regular GAN [10] is a special case of Jensen–Shannon divergence, which can be generalized as arbitrary f-divergences [31]. The most recent published work is proposed by Qi [32] which is called Loss-Sensitive GAN. This work conveys the assumption that the real samples should have smaller losses than fake samples and they proved the proposed loss function has a non-vanishing gradient. Although GANs have made successful progress, there are still many unsolved problems such as training instability and high-resolution generation. In this paper, we show a new way to unite multiple generators and a discriminator, in which the discriminator will provide the correct information for updating the generator, and then the generator will generate samples that are very similar to real ones. The motivation of the proposed model is that, jointly producing multiple samples by uniting multiple generators and a discriminator. More details, the multiple generators at the subsequent branches will focus to completing the missing details for producing the higher resolution images. In terms of employing multiple generators, our work is close to [24,25,27]. However, while using multiple generators, our model explicitly enforces them to capture diverse modes, then we design a sophisticated discriminator based on least square loss function.

3. Preliminaries

The basic GAN consists of two networks, generative and discriminative which are simultaneously trained. The generator trains to generate fake samples which are very similar to real samples, and the discriminator trains to distinguish the real samples from the fake samples. Given

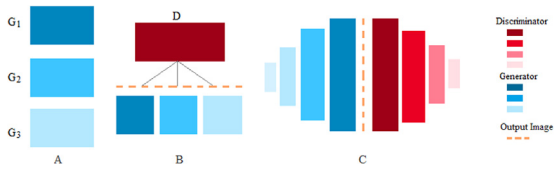

Fig. 2. Overviews of our GAN frameworks. **A**, is diverse generators (multiple generators). **B** is a single discriminator. **C** progressively trains symmetric discriminator and generators. A and C can be viewed as decomposing high-resolution tasks in our proposed network.

a set of sample z from the real data distribution $D_{real} = (x_i)_i^n$. Let G_u denotes the generators, where is often neural network in practice. Here u denotes the parameters of the generators. Similarly, D_v denotes the discriminators, and v is the parameters of the discriminator. GAN trains to obtain $\theta^{(G)}$ that can generate samples from the data distribution D_g, and the discriminator learns to recognize whether the image is a generated image D_g or a real data D_{real}. The basic GAN [10] is the training parameters *u, v* so as to optimize the following objective function:

$$min_{G_u} max_{D_v} \mathbb{E}_{x\sim D_{real}} log D\left(x;\theta_d\right) + \mathbb{E}_{z\sim D_z} log\left(1 - D\left(G\left(z;\theta_g\right);\theta_d\right)\right). \quad (1)$$

where, $D_g \& D_z$ are the empirical distributions of training samples. For a random sample x, which can either belong to D_{real} or D_g and also the parameter θ_d, we have a binary class as $D\left(x,\theta_d\right) \in [0,1]$ which is a score based on the probability of x (belong to real data or generated data). In Eq. (1), the discriminator should give a high score for real samples, while minimize it for generated samples from D_g; also the generator works exactly opposite of the discriminator such that it learns to maximize the score for the fake samples; *i.e.*, it aims to minimize the $\mathbb{E}_{z\sim D_z} log(1 - D(G\left(z;\theta_g\right);\theta_d))$ while maximizing the $\mathbb{E}_{z\sim p_z} \log D(G\left(z;\theta_g\right);\theta_d)$. In fact, the generator and discriminator are involved in a minimax game and the generator learns to generate real looking samples as $D_g = D_{real}$. for the original objective function in Eq. (1), the optimal value from $D\left(x,\theta_d\right) \in [0,1]$ is as: $D(x) = \frac{P_{real}(x)}{P_{real}(x)+P_g(x)}$, where, $P_{real}(x)$ is the density of sample x in the real distribution, and $P_g(x)$ is the density of the sample x in the distribution generated by generator G. Using this discriminator correspond to minimizing the JS divergence between the real distribution D_{real} and generator distribution D_g. Therefore, the Jensen–Shannon (JS) divergence for these two distributions is calculated as: $d_{JS} = \frac{1}{2}(KL(\mu \left\| \frac{\mu+\upsilon}{2}\right.) + KL(\upsilon \left\| \frac{\mu+\upsilon}{2}\right.))$.

4. Diverse generator network

Recently GANs have demonstrated great performance in various tasks. However, they still face challenges in generating high resolution images, especially natural images. In this paper we propose a new framework for image super-resolution by increasing the generator's capacity and modifying the objective function of the discriminator. The proposed architecture DGAN employs multiple generators that share their information and parameters. That means, there is same input for all generators and the image is generated from the different branches of the network. We first review the proposed model baseline then describe how we increase the image realism and resolution.

4.1. Hierarchical-nested generators

We introduce multiple generators which can bring a number of design possibilities and can explore two extreme that; more generators G better estimating $min_G V$ *(D, G)* and also multiple G can be a better match to the discriminator. We have two networks $G_k(z;\theta_k^g)$ and $D(x;\theta^d)$ as, *k* generator and a discriminator. We displayed our proposed architecture in Fig. 1. We used multiple generators $G_{1:k}$ that are designed with the objective that the mixture of their induced distribution would estimate the final data distribution, while they preserve their distinct. Our idea is to use mixture of several distributions instead of single distribution. Multiple generators act as *mixture model* and are used to capture more details. The proposed approach is a novel adversarial architecture with three components; set of generators, a discriminator, and objective function. Each generator maps z to $x = G_k(z)$ and induces a distinct distribution p_{gk}, which means, k generators together would induce a mixture of k distribution that termed as p_{total}. The discriminator aims to classify the real and fake samples between this generated samples and training data over $(k+1)$ distribution; i.e., k generators plus true data distribution. Note that, the multiple generators are jointly trained to produce images of different scales. Mathematically, we reformulate the k number of generators as: $\min_G \max_D(V(D,G1),\ldots,V(D,Gk))$. Based on this observation, *i*th generator involves generating a sample x_i. In more details, for the set of *k* generators, the discriminator receives $k+1$ input (real sample plus generated samples), i.e., if the score being at $k+1$ - index as $(D_{k+1}(.))$ then it represent the probability that the sample is real data distribution and if the score being at $n \in \{1,2,\ldots,k\}$th then it represents the probability of that the sample is generated images by *n*th generators. As the final output of the discriminator is binary $\delta\epsilon\{0,1\}^{k+1}$, if the sample belongs to the *n*th generator then $\delta(n) = 1$, otherwise, $\delta(k+1) = 1$. Consequently, based on this theory we can formulate the discriminator as: $\max_{\theta_d} \mathbb{E}_{x\sim p} F(\delta, D(x;\theta_d))$; where $F(.,.)$ is the loss function. Intuitively, in order to handle these diverse samples and correctly classify them, we use least square function (LS) as loss function. LS function allows discriminator that along with finding the real and fake samples, also; correctly update the generator that it produced the fake samples. However, the objective of each generator in training process is the same as standard GAN. The gradient for each generator is computed as $\nabla_{\theta g} \log(1 - D_{k+1}\left(G_i\left(z;\theta_g^i\right);\theta d\right))$, also the discriminator can update all the generators in parallel. In the case of discriminator, the gradient is calculated as: for given $x \sim p$-either fake or real- and δ, we have: $\nabla_{\theta d} log D_n\left(x;\theta_d\right)$. Here, since we use multiple generators, we need to use joint objective in order to create a mixture model, where, each generator represents a value "$-(k+1)\log(k+1) + klogk$", (for *k*-generator). Therefore, the generators are optimized to jointly estimate multiscale image distribution by minimizing the loss function, where τ_g represent the loss function at the *i*th scale: $\tau_g = \sum_{i=1}^k g_i, \tau_{g_i} = \frac{1}{2}\mathbb{E}_{xi} \sim p_{g_i}\left[\log\left(1 - D_{si}\right)\right]$. More formally, given *D and* $G_{1:k}$, the objective for training the generator is to minimize:

$$\mathbb{E}_{x\sim p_{data}}[log D(x)] + \mathbb{E}_{x\sim p_{total}}\left[\log(1 - D(x))\right] - (k+1)\log(k+1) + klogk \quad (2)$$

Assuming both D and $G_{1:k}$ the JSD between the $G_{1:k}$(mixture distribution) and the real data distribution are minimal if $p_{data} = p_{model}$, and JSD would be maximal, if each generator produce a distinct samples. On the other side, the output of the discriminator goes through an objective function in order to classify the given input, and then attributing higher scores to the generated fake samples and low scores to the real samples. If we define the margin boundary as "σ", and generated samples G (z), real data samples as x, the discriminator and generator losses $l_D.l_G$ can be calculated as:

$$l_D(x,z) = D(x) + [\sigma - D(G(z))]^+, \text{ where } [.]^+ = max(0,.)$$
$$l_G(z) = D(G(z)) \quad (3)$$

Minimizing l_G is similar to maximizing the $[.]^+$, and the proposed $l_D(x,z)$ allows us to generate higher quality results. As all k generators share the same objective function, we use same backpropagation passes to update their weights.

4.2. Model architecture

For the generator in the proposed model, we employ the recent developed EUSR model [38]. The structure is shown in Fig. 2. It is

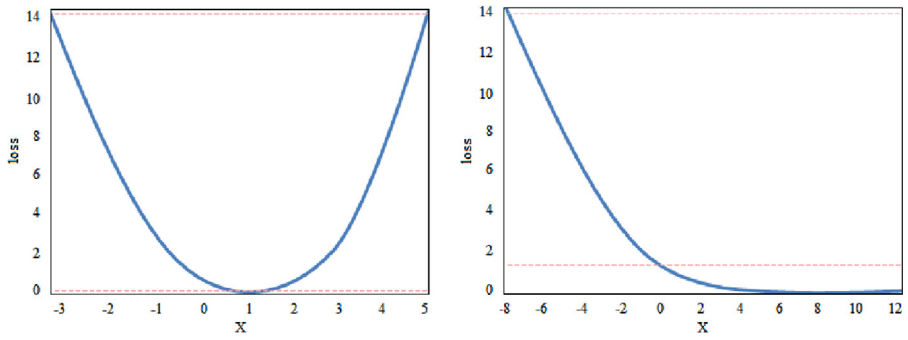

Fig. 3. Left: The least square loss function. Right: The sigmoid loss functions.

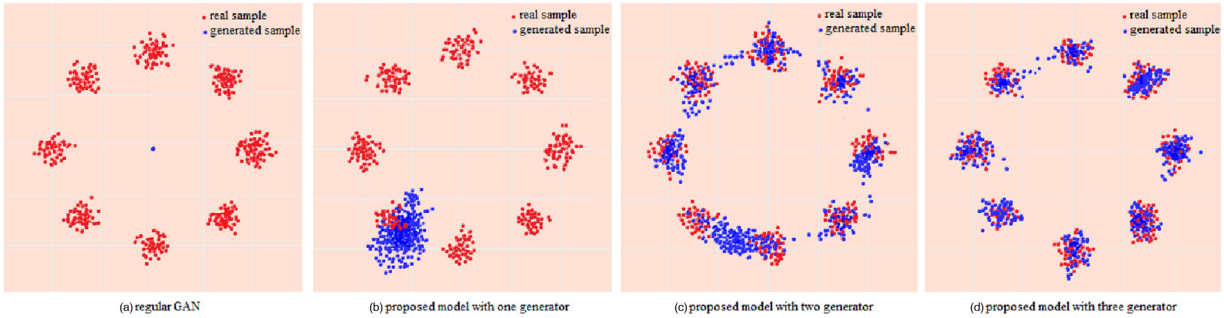

Fig. 4. The proposed model trained on synthetic data with: (a) regular GAN [10]; (b) one generator. (c) 2 generators. (d) 3 generators. Generated data are in blue and real data are in red.

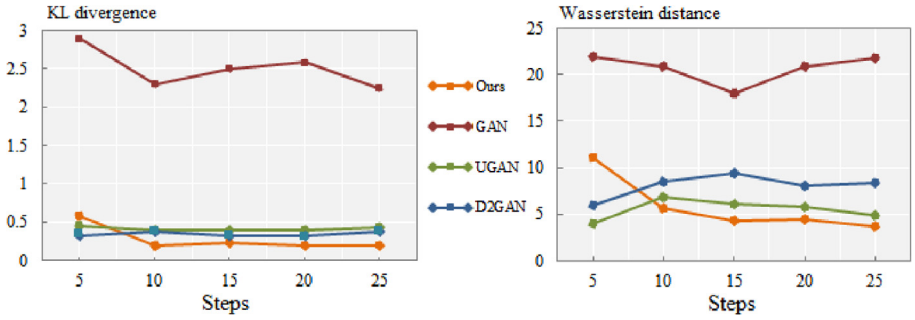

Fig. 5. Comparison of our model with GAN variation based on KL and Wasserstein distance. The lower results are better.

Table 1
Network hyperparameters for generated synthetic data.

Parameters	Features	Activation functions
$Z \sim \mu$ (0, **I**)	256	–
FC (fully connected)	128	ReLU
FC	128	ReLU
FC	2	Tanh
D (x)	2	–
FC	128	Leaky ReLU
FC	–	Least square functions
No. of generators	3	
Number of iterations	25,000	
Leaky ReLU slope	0.02	
Learning rate		$\beta = 0.125$
Optimizer	Adam ($\beta_1 = 0.5, \beta_2 = 0.999$)	

multiscale approach in three different scales (×4, ×6 and ×8) in which performing simultaneously. For the discriminator, we used the same network of SRGAN [39]. However, as we are dealing with diverse samples which generated from multiple generators, we found that the adversarial loss in the discriminator, it is not appropriate here, thus we replaces it with least square function. More concretely, we have some generators G that takes a random variable z as input and outputs a sample x. Core to our model is modifying the GAN architecture based on following criterion. First is the generator network which we used multiple generator G instead of single G, and a single discriminator. This way allows the network to produce more diverse samples. As we followed the architecture of [37], low level features are extracted by two residual blocks, while the higher features are extracted by residual module. Second is the trend towards distinguishing the fake and real samples which it is done by discriminator. We designed the discriminator to correctly classify the real and fake samples and also update the generator that generated the fake sample. Moreover, the loss function used in discriminator of our model helps to stabilize the learning process. Third is batch normalization, it helps to deal with training problems that arises because of the poor initialization and also helps gradient flow in deep layers. The ReLU activation is used in all the generators except the output layer which uses Tanh function. We also avoided applying batchnorm to the input layer of discriminator as we have not applied it to the generator output layer. As we discussed in Section 3, the discriminator of regular GAN acts as a classifier that classify the real and fake samples, where the adopted loss function is sigmoid cross entropy. The sigmoid cross entropy function is formulated as: $Loss = -ylog\rho(x) - (1-y)\log(1-\rho(x))$, where $\rho(.)$ is the

Table 2
Comparison of our model with state of the art approaches in term of PSNR, SSIM and RMSE evaluated on three benchmarks; Set5, Set14, BSD100, and Urban100. The first best results are stressed by **blue** color and the second best results with **green** color. [Scale factor 4×].

	Set14			Set5		
	PSNR	SSIM	RMSE	PSNR	SSIM	RMSE
SRDenseNet	23.09	0.7351	11.1885	26.63	0.7843	12.120
DCGAN	25.17	0.7758	10.0117	24.51	0.8097	11.543
RDN	24.42	0.6957	13.1528	25.10	0.6701	14.214
ResGAN	22.83	0.7109	14.9981	25.06	0.7495	12.033
GP-GAN	26.67	0.8439	12.5128	24.81	0.8406	12.193
DGAN (ours)	**31.62**	**0.9166**	**6.1045**	**32.85**	**0.8911**	**6.1708**
SRGAN	**28.54**	**0.8917**	**7.0983**	**30.11**	**0.9028**	**6.9312**
StackGAN	**29.78**	**0.9005**	**6.8316**	**29.04**	**0.8996**	**7.3451**
TAC-GAN	**31.09**	**0.8914**	**7.8412**	**29.86**	**0.8964**	**8.4307**
HDGAN	**29.06**	**0.8546**	**7.4312**	**28.17**	**0.7876**	**16.125**

	BSD100			Urban100		
	PSNR	SSIM	RMSE	PSNR	SSIM	RMSE
SRDenseNet	25.76	0.7391	12.4133	23.08	0.6513	14.332
DCGAN	25.49	0.6209	14.5610	27.84	0.7520	16.578
RDN	26.53	0.7145	13.2752	24.62	0.7399	16.992
ResGAN	24.28	0.7518	10.9579	25.95	0.7058	14.671
GP-GAN	26.17	0.8571	11.2385	23.63	0.8329	12.654
DGAN (ours)	**31.53**	**0.9105**	**7.5128**	**29.41**	**0.8991**	**8.5477**
SRGAN	**29.05**	**0.8657**	**10.0193**	**27.16**	**0.8763**	**9.1409**
StackGAN	**28.90**	**0.8894**	**9.0658**	**27.70**	**0.8597**	**11.133**
TAC-GAN	**28.93**	**0.8905**	**9.5734**	**29.15**	**0.8695**	**10.140**
HDGAN	**30.18**	**0.8914**	**9.9740**	**28.64**	**0.8851**	**9.2395**

sigmoid loss function. In Fig. 3, we plot the sigmoid cross entropy loss function against least square loss function. From the figure it observes that when x is relatively large, the sigmoid loss will be saturated, while at the same condition the lease square loss will be increased. The reason is that, the least square loss function strongly penalizes the samples to correctly classify them, and then relieves the saturation problem of GAN models. Therefore, if the sigmoid floss is substituted by the least square loss, the model will be converging to a soothed state. Moreover, another main property of least square function is to stabilize the learning process and thus it allows us to explore more powerful network architectures. Based on above consideration the network will be trained to address:

$$= \min_{G_k} \max_{D} \frac{1}{N} \left\{ l_a + \mu(l_p + l_g) \right\} \tag{4}$$

l_a is least square loss function. For a fixed generators G, the objective function of the discriminator is to maximize $\mathbb{E}_{x\sim pdata} log\, D_{k+1}(x) + \sum_1^k \mathbb{E}_{xi\sim pgi} log\, D_i(x_i)$. However, from the minimax problem in Eq. (2), the optimal generator $G^* = [G_1, \ldots, G_k]$ induces the generated distribution $P^*_{total}(X) = \sum_{k=1}^k \pi_k P G^*_k$ which is very close to the real data. From the Eq. (2), we can reformulate the G* and D* as follow:

$$G^* = \frac{\pi_k P G^*_k}{\sum_{i=1}^k \pi_j P G^*_k(x)} \quad D^* = \frac{p_{data}(x)}{p_{data}(x) + p_{total}(x)} \tag{5}$$

However, the objective function for the k generator calculated as:

$$\begin{aligned} \tau(G_k) = {} & \mathbb{E}_{x\sim P_{data}} \left[log \frac{p_{data}(x)}{p_{data}(x) + p_{total}(x)} \right] \\ & + \mathbb{E}_{x\sim P_{total}} \left[log \frac{p_{total}(x)}{p_{data}(x) + p_{total}(x)} \right] \\ & - \delta \left\{ \sum_{k=1}^{k} \pi_k \mathbb{E}_{x\sim PG_k} [log \frac{\pi_k P G_k(x)}{\sum_{j=1}^k \pi_j P G_j(x)}] \right\} \end{aligned} \tag{6}$$

The generators at the intermediate branches gradually generate the images from small to large in order to accomplish the final goal which is the generating high-resolution image, and then the discriminator, estimates the probability between the fake and real images.

4.3. Implementation details

The optimization of GAN formulates a minimax problem in which with an optimal discriminator D, learning objective turns to finding generators that minimizes the JSD. Thus, for the discriminator network in the proposed method we employ the network which used in SRGAN [39]. The network consists of "ten" convolutional layers followed by batch normalization units, and leaky ReLU activations with $\alpha = 0.02$. However as we uses multiple generators in order to generate diverse samples, to handle these diversity the least square function is substituted with final sigmoid activation in last layer of discriminator.

Consequently, the output of feature maps are processed by two dense layers and the final loss function which we used least square function here, will determine the probability that the input image is real or fake. The generators $G_1, G_2, \ldots, G_k$ are convolutional neural networks parameterized by ϑ_G. They share parameters in all layers except for the input layers. 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. We set the number of residual block in each generator to 80 in order to improve the learning capacity of the networks. The algorithm of sampling from the k generators is described in Algorithm 1.

We use TensorFlow [40] to implement our model. In all experiments, we use shared parameters among all generators except the input layer and also for the discriminator except the output layer. Moreover, we set Adam optimizer to 0.0002 learning rate and momentum 0.5, also weights initialized from an isotropic Gaussian, μ (0, 0.01) and zero biases. We use ReLU activation for all generators while we observe that for the discriminator Leaky ReLU with slope 0.2 is more suitable. For the synthetic experiment, we followed the proposed experiment in [14] to explore the effectiveness of multiple generators. To this end, as the structure of [14] we sample training data from 2D mixture of 8 Gaussian distribution with covariance $0.002I$. This small variance allows creating low density regions and then separates the modes. Three models is employed, each having a simple architecture, stating with input layer of 256 noises which drawn from "isotropic multivariate Gaussian distribution μ (0, $\boldsymbol{I}$)", and two fully connected hidden layer

Table 3
Comparison of VDSR [33], DCGAN [19], ProGan [23], DRRN [34], IDN [35], SFT-GAN [36], MemNet [37] and our proposed model on four benchmark datasets: (Set5, Set14 and BSD100). The highest measures are (PSNR [dB], SSIM) in bold and **blue**, the second highest in **green**. [6× and 8× scale factor].

Set 5	6 ×		8 ×	
	PSNR	SSIM	PSNR	SSIM
VDSR	21.17	0.7263	19.03	0.7043
DCGAN	28.49	0.8358	28.11	0.8097
DRRN	23.42	0.6957	22.10	0.6701
ProGAN	26.71	0.8504	24.98	0.8433
LSGAN	27.86	0.8924	26.67	0.8881
IDN	26.63	0.8309	21.06	0.7495
SFT-GAN	29.96	0.8846	26.17	0.7876
MemNet	30.08	0.9196	27.83	0.7653
DGAN (ours)	**30.62**	**0.9107**	**87.35**	**0.8911**
Set 14	6 ×		8 ×	
	PSNR	SSIM	PSNR	SSIM
VDSR	23.09	0.7151	21.63	0.7443
DCGAN	29.17	0.8858	26.51	0.8697
DRRN	24.42	0.6957	23.10	0.6701
ProGAN	26.38	0.8579	22.41	0.7396
LSGAN	28.61	0.8836	26.19	0.8603
IDN	25.83	0.7109	23.06	0.7495
SFT-GAN	27.96	0.8546	24.17	0.7876
MemNet	26.34	0.8956	23.66	0.8209
DGAN (ours)	**28.62**	**0.9003**	**26.85**	**0.8911**
BSD100	6 ×		8 ×	
	PSNR	SSIM	PSNR	SSIM
VDSR	24.64	0.7108	24.19	0.7357
DCGAN	25.17	0.8058	23.51	0.8192
DRRN	23.42	0.7757	25.10	0.7601
ProGAN	24.77	0.8105	23.67	0.8009
LSGAN	27.09	0.8594	25.35	0.8394
IDN	22.83	0.7109	19.06	0.7495
SFT-GAN	28.16	0.8546	24.17	0.7876
MemNet	26.44	0.8617	25.88	0.8392
DGAN (ours)	**29.62**	**0.8937**	**27.85**	**0.8811**

that followed by 128 ReLU units. However for a single discriminator, only one hidden layer of 128 ReLU units is used. Also, the diversity parameters set to 0.125. More details are given in Table 1. We also used RMSE, PSNR and SSIM as evaluation metrics.

Algorithm 1- Mixture of generators for proposed model

Step 1- samples noise z from the P_z
Step 2- sample a generator index from $(\pi_1, \pi_2, \ldots \pi_k)$.
Step 3- $h = f\vartheta_G(z)$
Step 4- $x = g\vartheta_G(h)$
Step 5- return generated data x.

5. Experimental evaluation

In this section, we evaluate the performance of the proposed model and conduct a series of experiments to compare it with other prominent methods especially WGAN [16], MemNet [37], BEGAN [18], Stack-GAN [41]. We have two sets of experiments; one is based on synthetic data another one is based on real-world datasets. The aim of using synthetic data is to show the effect of the number of generators in the model. In essence, we want to visualize and evaluate the learning behavior of our model with using multiple generators and demonstrate its stability and efficacy in a larger and wider data space. The result of synthetic data is given in Fig. 4. We employ one, two and three generators for 25,000 epochs. The results show that, the model with one generator having similar behavior of regular GAN. The model with two and three generators would successfully cover all 8 modes. The network details and specification is given in Table 1. Next for more quantitative evaluation we use several real word datasets and show the results in the same experimental setting. We used three widely adopted datasets; Set5, Set 14, BSD 100 and CIFAR-10. CIFAR-10 contains 50,000 training images of 10 classes, including: bird, deer, dog, airplane, automobile, cat, frog, horse, ship, and truck. We choose Set5, Set14 and BSD 100, since they consist of natural scenes images. Results on these benchmarks show that, our model generates more faithful and more diverse samples than the baselines. We select the baselines from CNN-based methods such as, SRCNN [5], VDSR [33], MemNet [37], LapSRN [42], and also several known variation of GAN including, SRGAN [39], SFT-GAN [36], BEGAN [18], D2GAN [14] and Unrolled-GAN [13]. For re-implementing the baselines we followed their structures and released codes with the same setting as ours. From results it is observed that, the non GAN based methods despite of preserving sharp edges, they produces blurry textures. However, the perceptual quality of GAN based methods is better than others and even they could improve the high frequency details. However, our proposed model comparing to the baselines leading to more natural and realistic textures.

Fig. 6. Image quality improvement across different techniques. We used two sample images. In both the images, the first top is the ground truth image (Ground-T). The results show that, our proposed model is capable of generating richer and more realistic textures among other methods even have competition with D2GAN. MemNet in the both images tend to produce the unpleasant images and could not properly capture the fine details of the image. Regular GAN based on LS loss also could not generate a realistic texture; the generated images are blurry and cloudy. (Zoom in for best view).

5.1. Comparisons and results

We present an extensive qualitative and quantitative performance on our proposed model on various synthetic and real world datasets. We also employ two measures KL-divergence [43] and Wasserstein distance [16] as criterion for comparison and show the effectiveness of our models compared to GAN's variations. Fig. 5 clearly shows the superiority of our model over; Unrolled-GAN [13], D2GAN [14] and regular GAN [10]. In both curves, our model almost reduced to zero. Also our model stability can be observed in these figures, since it is much less fluctuating compared to others. We also train our model with the highest upsampling scales; $\{4\times\}, \{6\times\}$ *and* $\{8\times\}$ between low and high resolution images. During training, we equally split the samples for every upsampling scale. We show the qualitative evaluation on the set of datasets such as; Set5, Set14, CIFAR-10 and BSD100.

Fig. 4 show the effect of using number of generators on generated samples. The results achieved on synthetic data for 25,000 epochs. We trained three generators and the model with single generator behaves similarly to the regular GAN. The models with 2 and 3 generators successfully cover 8 modes, but the ones with two generators have fewer points scattered between adjacent modes. Finally, the model with three generators performs well and could cover all eight modes. We evaluate the results with three quality metrics; PSNR, SSIM and RMSE. We compare two variation of the proposed method: one is the proposed structure with least square loss function; another is the proposed structure with sigmoid loss function. We plot the results in Fig. 3. Empirically, least square loss function has better result comparing to sigmoid loss. However, we find that the sigmoid loss function is caused the saturation problem. In fact, the curve in Fig. 5-Left indicates that when x is relatively large then the sigmoid loss will be saturated. As shown in Fig. 5-right, least square loss function will saturate when x = 1, which is an example of successful GAN learning. Therefore, if we replace sigmoid loss by least square loss, the model will be able to converge to a good state. Next, Tables 2 and 3 show the quantitative at three scales; 4 ×, 6 × and 8 ×. We compare the performance of our model with eight states of the art GAN methods; SRDenseNet [44], SRGAN [39], DCGAN [19], ResGAN [12], GP-GAN [45], StackGAN [41], TAC-GAN [46] and HDGAN [47]. Quantitative results in terms of PSNR, SSIM and RMSE are summarized in Table 2. The results convey that our model not only has a compatible performance compared to the state of the art methods, but also has a simple implementation, stable results and less training time. The best result is stressed with blue color, and the green colors are the second best results. Our proposed model outpaces the other methods which are shown in green colors by 1–2.8% ratio. In particular our model achieves the lower RMSE values, which has been shown the higher perceptual quality. Note that, Table 2 is achieved at 4 upsampling factors.

Fig. 6 shows the results of generated image by different methods including ours. The CNN-based methods; RDN and MemNet fail to generate better results. In contrast, GAN-based method is able to generate better results, but it still contains significant blur and cloudy points. We note that, our proposed model improves the regular GAN result by adopting multiple generators instead of using one generator. Multiple generators can produce diverse samples which would direct us to better results. As it observes, our model is able to generate much clearer images comparing to regular GAN. Furthermore, we then

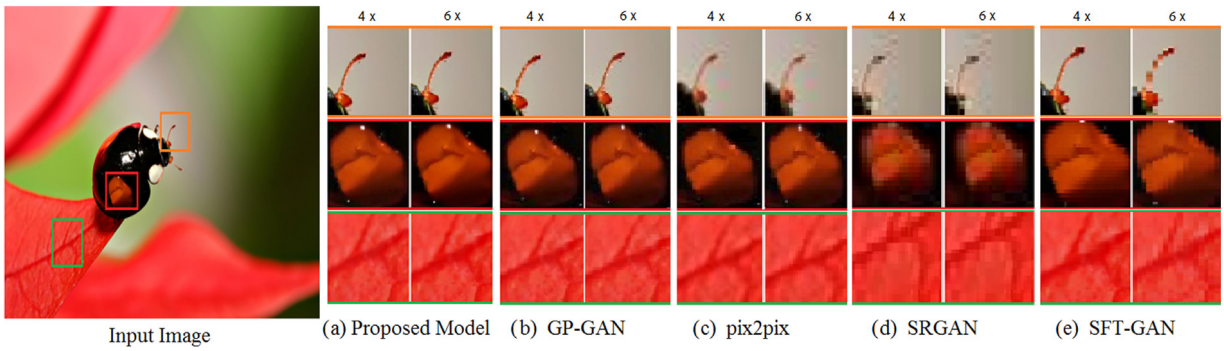

Fig. 7. Results at 4× and 6× super-resolution on CIFAR-10 dataset. More results are shown in the supplementary material including FSIM and VIF metrics. (zoom in for best review).

Fig. 8. Visual comparison of SR results at 4× scaling factor. Comparison of the proposed model with other method; MemNet [37], SRDenseNet [44], LapSRN [42], StackGAN [41], SRGAN [39] and SFT-GAN [36]. We used two sample images. GAN-based methods (SRGAN, SFT-GAN, StackGAN and ours) clearly outperform the other approaches in term of perceptual quality. Both the images show that our method is capable to captures the characteristics of fine lines and building brick. The result is done at 4× scale factors. (Zoom in for best view).

evaluate the performance of our proposed model and the most recent SR-based deep learning methods at higher scaling factors (6 × and 8 ×). The results are reported in Table 3. The baselines used here are; VDSR [33], DCGAN [19], DRRN [34], ProGAN [23], IDN [35], SFT-GAN [36], MemNet [37]. The proposed method performs favorably against other techniques in term of PSNR, SSIM. The GAN based methods are indicated with green color in order to make more recognizable the GAN methods and CNN methods. As the results show, GAN based methods significantly outpaces other deep learning methods. MemNet and ProGAN methods despite having smooth training process did not show a stunning performance. Their performance is lower than other GAN based methods. The proposed model performs favorably against existing methods in all three scales; 4 ×, 6 × and 8 ×.

We show visual comparison on BSD 100 and DIV2K datasets in Figs. 7 and 8. From the results it observes that, our model is able to correctly reconstruct the fine structures, grid patterns, and the dark spots in the image backgrounds. However, the CNN based methods could not resolve the fine structures well even at 4 × scaling factors. The results from GAN based methods are more realistic comparing to CNN based methods. Among other GAN based methods, stackGAN and TAC-GAN show a better performance. At the same time our model achieves perceptual quality similar to SRGAN and StackGAN in term of RMSE. The lower the perceptual index is, the better the perceptual quality. Fig. 8 presents an overview of different approaches including the current state of the art in terms of PSNR. We selected two practically well-suited images for a visual comparison since they contain sharp and smooth edges. The previous methods have significant improvements on the sharp edges, however, even SFT-GAN which is considered as the most recent state of the art in GAN methods, still suffers from a blur region where the image does not have sufficient details to provide for

the system. From the results, it can be seen that our proposed model can provide more clear results in comparison with the others. We believed the least square loss function allowed the generator to generate samples which are quite similar to the real one. As the generators play a vital role in GAN, we need to provide the most complete information for updating it.

We compared our model to several GAN based and non-GAN models. In the experiments, we train the networks with the different scaling factors; 4×, 6× and 8×. The table implies that the results of methods based on GANs outpace other non-GANs. Therefore we can conclude that GAN based methods are well-suited methods in image super-resolution. From the results it is clearly observed that the proposed model achieved superior performance in all measures; it even has a compatible performance with SFT-GAN, ProGAN, and MemNet. However, for the high scaling factor 8×, the second best method is DCGAN and GP-GAN that shows better performance compared to other prominent methods. For the 4× scaling factor, the second best results are for MemNet and ProGan methods.

6. Conclusion

In this paper, we present an effective framework, diverse generative adversarial network to generate meaningful samples for image super-resolution. The proposed model consists of multiple generators – which gradually grow from small to large – along with one discriminator network. This learning strategy helps to balance both networks in order to obtain stable results, and we also provide theoretical analysis of DGAN that the proposed objective of discriminator allows multiple generators to learn together as a mixture model. We believe that our proposed model not only has a simple implementation in comparison with the other GAN variation but also presents superior results. In essence, this work concludes two main aspects. The first is that the objective function which is designed for the discriminator significantly improves the GAN performance by guiding the processing of updating the generators. And the second lies in the learning structure for multiple generators which we believe are more stable and efficient for generative networks. In addition, for the future direction, we would like to estimate the number of generators and discriminators needed for a particular dataset.

Conflict of interest

The authors declare that there is no conflict of interest in this paper.

References

[1] W. Wu, X. Yang, K. Liu, Y. Liu, B. Yan, H. Hua, A new framework for remote sensing image super resolution: Sparse representation-based method by processing dictionaries with multi-type features, J. Syst. Archit. 64 (2016) 63–75.

[2] H. Chen, X. He, L. Qing, Q. Teng, C. Ren, SGCRSR: Sequential gradient constrained regression for single image super-resolution, Signal Process., Image Commun. 66 (2018) 1–18.

[3] P. Shamsolmoali, M. Zareapoor, D.K. Jain, V.K. Jain, J. Yang, Deep convolution network for surveillance records super-resolution, Multimedia Tools Appl. (2018) 1–15.

[4] D.H. Trinh, M. Luong, F. Dibos, J.M. Rocchisani, C.D. Pham, T.Q. Nguyen, Novel example-based method for super-resolution and denoising of medical images, IEEE Trans. Image Process. 23 (4) (2014) 1882–1895.

[5] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2016) 295–307.

[6] M. Zareapoor, D.K. Jain, J. Yang, Local spatial information for image super-resolution, Cogn. Syst. Res. 52 (2018) 49–57.

[7] G. Lin, Q. Wu, L. Chen, L. Qiu, X. Wang, T. Liu, X. Chen, Deep unsupervised learning for image super-resolution with generative adversarial network, Signal Process., Image Commun. 68 (2018) 88–100.

[8] M. Zareapoor, P. Shamsolmoali, D.K. Jain, H. Wang, J. Yang, Kernelized support vector machine with deep learning: An efficient approach for extreme multiclass dataset, Pattern Recognit. Lett. 115 (1) (2018) 4–13.

[9] M. Zareapoor, P. Shamsolmoali, J. Yang, Learning depth super-resolution by using multi-scale convolutional neural network, J. Intell. Fuzzy Systems (2018) http://dx.doi.org/10.3233/JIFS-18136.

[10] 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.

[11] 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.

[12] M. Wang, H. Li, F. Li, Generative Adversarial Network based on Resnet for Conditional Image Restoration, (2017). arXiv:1707.04881v1.

[13] Luke Metz, Ben Poole, David Pfau, Jascha Sohl-Dickstein, Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163, 2016.

[14] Tu Dinh Nguyen, Trung Le, Hung Vu, Dinh Phung, Dual discriminator generative adversarial nets, in: Advances in Neural Information Processing Systems, Vol. 29 (NIPS), 2017, (in press).

[15] K. Nasrollahi, T.B. Moeslund, Super-resolution: A comprehensive survey, Mach. Vis. Appl. 25 (2014) 1423–1468.

[16] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 214–223.

[17] C. Villani, Optimal transport: Old and new, Amer. Math. Soc. 47 (4) (2009) 723–727.

[18] D. Berthelot, T. Schumm, L. Metz, Began: Boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717, 2017.

[19] 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:151106434.

[20] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, in: Proceeding of the NIPS, 2016, pp. 2234–2242.

[21] Casper Kaae Sonderby, Jose Caballero, Lucas Theis, Wenzhe Shi, Ferenc Huszar, Amortised map inference for image super-resolution, 2016. URL https://arxiv.org/abs/1610.04490v1.

[22] T. karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of gans for improved quality, stability, and variation, in: Proceeding of the ICLR, 2018, arXiv:171010196v3.

[23] 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.

[24] Quan Hoang, Tu Dinh Nguyen, Trung Le, Dinh Phung, Multi-Generator Generative Adversarial Nets. arXiv preprint arXiv:1708.02556, 2017. 1.

[25] M.-Y. Liu, O. Tuzel, Coupled generative adversarial networks, in: Advances in Neural Information Processing Systems, 2016.

[26] T.C. Wang, M.Y. Liu, J.Y. Zhu, A. Tao, J. Kautz, B. Catanzaro, High-resolution image synthesis and semantic manipulation with conditional GANs, in: Proceeding of the CVPR, 2017.

[27] I. Durugkar, I. Gemp, S. Mahadevan, Generative multi-adversarial networks, in: Proceeding of the ICLR, 2017, arXiv:161101673.

[28] A. Ghosh, V. Kulharia, V.P. Namboodiri, P.H.S. Torr, P.K. Dokania, Multi-agent diverse generative adversarial networks, in: Proceeding of the CVPR, 2017, pp. 8513–8521.

[29] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: CVPR, 2017, pp. 1125–1134.

[30] S. Nowozin, B. Cseke, R. Tomioka, F-gan: Training generative neural samplers using variational divergence minimization, in: h Conference on Neural Information Processing Systems (NIPS), 2016, pp. 271–279.

[31] X. Nguyen, M.J. Wainwright, M.I. Jordan, Estimating divergence functional and the likelihood ratio by convex risk minimization, IEEE Trans. Inform. Theory 56 (11) (2010) 5847–5861.

[32] G.J. Qi, Loss-sensitive generative adversarial networks on lipschitz densities, (2018), arXiv:1701.06264v6.

[33] 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.

[34] Y. Tai, J. Yang, X. Liu, Image super-resolution via deep recursive residual network, in: Proceedings of the CVPR, 2017, pp. 2790–2798.

[35] Z. Hui, X. Wang, X. Gao, Fast and Accurate Single Image Super-Resolution via Information Distillation Network.

[36] X. Wang, K. Yu, C. Dong, C.C. Loy, Recovering realistic texture in image super-resolution by deep spatial feature transform, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018.

[37] Y. Tai, J. Yang, X. Liu, C. Xu, Memnet: A persistent memory network for image restoration, in: ICCV, 2017.

[38] J.H. Kim, J.S. Lee, Deep residual network with enhanced upscaling module for super-resolution, in: Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2018.

[39] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, Photo-realistic single image super-resolution using a generative adversarial network, in: Proceeding of the CVPR, 2017, pp. 105–114.

[40] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Lev-enberg, D. Man_e, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Vi_egas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, \TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorow.org.

[41] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, D.N. Metaxas, Stack-GAN: text to photo-realistic image synthesis with stacked generative adversarial networks, in: Proceeding of the ICCV, 2017, pp. 5907–5915.

[42] W.-S. Lai, J.-B. Huang, N. Ahuja, M.-H. Yang, Deep laplacian pyramid networks for fast and accurate superresolution, in: CVPR, 2017, pp. 624–632.

[43] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. (1951).

[44] T. Tong, G. Li, X. Liu, Q. Gao, Image super-resolution using dense skip connections, in: Proceeding of the ICCV, 2017, pp. 4799–4807.

[45] H. Wu, S. Zheng, J. Zhang, K. Huang, GP-GAN: Towards Realistic High-Resolution Image Blending, (2017). arXiv:1703.07195v2.

[46] A. Dash, J. Gamboa, S. Ahmed, M. Liwicki, M.Z. Afzal, TAC-GAN – Text Conditioned Auxiliary Classifier Generative Adversarial Network. CVPR, 2017.

[47] Z. Zhang, Y. Xie, L. Yang, Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network.