Neurocomputing 366 (2019) 140–153

Contents lists available at ScienceDirect

Neurocomputing

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

G-GANISR: Gradual generative adversarial network for image super resolution

Pourya Shamsolmoali[a], Masoumeh Zareapoor[a,b], Ruili Wang[c,*], Deepak Kumar Jain[d], Jie Yang[a]

[a] Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China
[b] Department of Computer Science, Tokyo University of Technology, Tokyo, Japan
[c] Institute of Natural and Mathematical Sciences (INMS), Massey University, Auckland, New Zealand
[d] Institute of Automation, Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Article history:
Received 21 October 2018
Revised 2 May 2019
Accepted 31 July 2019
Available online 6 August 2019

Communicated by Xiaofeng Zhu

Keywords:
Image super-resolution
Loss functions
GAN
CNN
Gradual learning

ABSTRACT

Adversarial methods have demonsterated to be signifiant at generating realistic images. However, these approaches have a challenging training process which partially attributed to the performance of discriminator. In this paper, we proposed an efficient super-resolution model based on generative adversarial network (GAN), to effectively generate reprehensive information and improve the test quality of the real-world images. To overcome the current issues, we designed the discriminator of our model based on the Least Square Loss function. The proposed network is organized by a gradual learning process from simple to advanced, which means from the small upsampling factors to the large upsampling factor that helps to improve the overall stability of the training. In particular, to control the model parameters and mitigate the training difficulties, dense residual learning strategy is adopted. Indeed, the key idea of proposed methodology is (i) fully exploit all the image details without losing information by gradually increases the task of discriminator, where the output of each layer is gradually improved in the next layer. In this way the model efficiently generates a super-resolution image even up to high scaling factors (e.g. $\times 8$). (ii) The model is stable during the learning process, as we use least square loss instead of cross-entropy. In addition, the effects of different objective function on training stability are compared. To evaluate the model we conducted two sets of experiments, by using the proposed gradual GAN and the regular GAN to demonstrate the efficiency and stability of the proposed model for both quantitative and qualitative benchmarks.

1. Introduction

Image super-resolution is a classic problem in computer vision. It aims to inscribe the details of an image, more details provide better resolution. Previously, this technology was not as attractive as it is today. However, over time with the growth of technologies, the need for resolution enhancement in some crucial applications cannot be overlooked in areas such as remote sensing [3], object recognition [5], security surveillance [1], and medical imaging [2]. High resolution (HR) 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 difficult task due to the lack of image texture details and sharpness edges. Recently, large numbers of super-resolution methods have been proposed and those which use Deep learning are superior. Due to the nature of deep learning which is based on non-linearity and ability to imitate any transformation and mapping, it is considered as a good fit for super-resolution problems. Since then, progress has been done on image super-resolution, 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 current CNN based methods cannot get fully satisfactory perceptual quality, because they have not fully exploited all features from the original input image (low-resolution), and some of the details may be lost during the training process. Thus, the corresponding results will be apparently undesirable. Another common issue in CNN models is an objective function. The CNN based super-resolution models used pixel-wise loss functions such as *l2* (least square errors) in their structures, which aim to reduce the *MSE* (mean square error) while increasing the similarity metric *PSNR* (peak signal to noise ratio) between model estimation and the ground-truth

* Corresponding author.
E-mail address: ruili.wang@massey.ac.nz (R. Wang).

https://doi.org/10.1016/j.neucom.2019.07.094
0925-2312/

image. However, as discussed in [19,26,35,37], those metrics do not consider the visual quality of the image. Therefore, their results lead to overall blurring and low perceptual quality. Inspired by CNN, recently generative adversarial network (GAN) [15] has demonstrated impressive performance and gained immense popularity in a variety of computer vision tasks. GAN is a class of neural network that learns to generate samples from a particular image input. It is comprised of two networks: a generator G and a discriminator D, which are in competition with each other. In fact, the generator learns to generate new samples and the discriminator learns to-distinguish between the generated samples and the real data points. In the GAN model each network wishes to minimize its own cost function, i.e. $f^D(\theta^D, \theta^G)$ for the discriminator and $f^G(\theta^D, \theta^G)$ for the generator. Generating super resolution images is a difficult task. Firstly, due to the lack of capacity to obtain small details (which are simply visible in a super-resolution images), and secondly, since the training process is unstable and lengthy. Recently, it has been pointed out that the main reason for these issues is the high dimensional spaces, which could be handled by a proper objective function [37]. By using an indecent loss, the discriminator recognizes the forgery samples (the generated samples) as the real samples with the least errors, because the samples are in the correct side of the margin boundary. This wrong decision has a negative impact on the updating process of the generator. 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 GAN based super-resolution models, we propose a new GAN model, which is based on an image-to-image model by organizing a gradual learning process from the small upsampling factors to the large upsampling factors. The loss function has the operative driver in the learning network. However, this key issue has not been properly considered before. Most existing methods try to improve the results by optimizing the network structure or designing new layers, and generally, they used the defaults loss [1,3]. These local losses are poorly correlated with the image's quality as it is perceived by a human observer. If the discriminator is considered as an energy-based function, then we can improve GAN stability. Based on these observations, this paper centered largely on the loss function and we designed a new discriminator that used the least square loss function and gradually training following generator; the parameters of the proposed least square model is simple to implement and has a fast computation rate. We proved that our GAN model has ability to deal with multiscale factors (up to $\times 8$). In the end, we proved that the proposed model adopting the least square is more stable than using Wasserstein GAN. This proposed learning process (simple to advanced) allows us to significantly improve the training result and could retain all the image information. To improve the image resolution and obtain realistic results, we designed our discriminator based on a least square function. The features obtained from the discriminator are exploited in order to create a more robust objective function, in contrast with current GAN which uses a classification network to generate the loss function. Least square [42] has the ability to appropriately separate the fake samples from the real samples by marginalizing the fake samples. In fact, the least square function controls the samples based on their distance to the margin, and so it helps to find more real samples for updating the generator. In this paper, we proved the power of the least square function to alleviate the current problems, by generating more gradient for updating the generator.

Our contributions are four-fold: (i) we proposed a new variation of generative adversarial network with adopting least square loss function for the discriminator which enables a stepwise quality enhancement by using the output of the previous layer. (ii) Opposed to the existing methods, we replaced the batch normalization with instance normalization [43] to obtain all the vital information. (iii) We evaluated the proposed model over several datasets and conducted two sets of experiments, direct learning strategy via the gradual learning strategy. (iv) in addition, we observed that the residual learning is beneficial in our model, as it speeds up the convergence. Thus, we adopted dense residual learning (contains both dense and skip connections) in our proposed architecture to simplify the training process. In fact, our contribution mainly focuses on this ongoing discussion (*apply densely connected residual network in the adversarial networks, and also adapting gradual learning strategy instead of direct learning*). In order to show the effect of least square in adversarial networks, we evaluated the result of our network with different loss functions, including Wasserstein [13]. We believe, the discrminator of our model can be prevented from becoming over-confident by adopting least square loss and it enables the generator to generate higher quality images in comparison with other approaches. The rest of this paper is organized as follows. In Section 2, we discussed the related works. Section 3 presents the proposed model architecture. Section 4 shows the experimental results and evaluation results. Finally, Section 5 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. [15] 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 (e.g., the images). The game between G and D is the minimax objective as shown in Eq. (2). There are substantial methods investigating the usage of GANs for different applications, such as image synthesis, pixel to-pixel translation, medical applications, etc. However, training of GAN suffers from many problems such as mode collapse, vanishing gradients, difficulties in the implementation and unstable results [6,7,11,29]. Arjovsky et al. [13] noted that the difficulties in GAN training are due to Jensen–Shannon (JS) divergence and thus they proposed a new model termed Wasserstein-1, $W(q, p)$. Their model uses Kantorovich–Rubinstein duality [17] as a value function: $\min_G \max_D E_{x\sim Pr}[D(x)] - E_{x\sim pg}[D(\tilde{x})]$, where D is the Lipschitz functions and Pg is the distribution. The aim of this method is to estimate the value of $K(P_r, P_\theta)$ where $W(P_r, P_\theta)$ indicates as Wasserstein distance and K is the Lipschitz constant. Based on these observations, discriminator called critic that try to estimate and minimize the difference gap between the samples. However, GAN by adopting Wasserstein may lead to optimization of the implementation difficulties and vanishing gradients. Gulrajani et al. [18] followed their previous work and improved the training problem by adding an extra gradient penalty to the value function as an alternative way to impose the Lipschitz constant. This strategy helps to implement a stable training with less hyperparameter tuning. Due to the massive variations and complexity in image content, it is still challenging for GAN to generate diverse images with sufficient details. Xiao et al. [25] proposed a learning strategy that allows the network to grow hierarchically when new training data are added. The authors reported the superiority of their proposed model as compared to others. Reed et al. [20] proposed a potential method for synthesizing images which contain text descriptions based on the conditional GAN [44]. In addition, several hierarchical GANs have been proposed [22–24], which define a generator and a discriminator for each level of the image pyramid. Wang et al. [31] used multiple discriminators in their architecture. Their work is inspired by Durugkar et al. [28], who used one generator and

multiple discriminators. The authors claim that their results outpace the study in [32]. Ghosh et al. [30] also proposed a model for improving the above techniques by using multiple generators and a single discriminator. Isola et al. [21] also proposed a model, which relies on the conditional GAN and aims to transfer images from one representation to another. In [29], 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. [16]. We pointed out that another method, which is called LAPGAN (Laplacian pyramid of generative adversarial networks), is proposed by Denton et al. [23]. Their model constructs a Laplacian pyramid to generate multi-resolution images from low-resolution images. Nowozin et al. [33] mentioned that the regular GAN [15] is a special case of Jensen–Shannon divergence, which can be generalized as arbitrary f-divergences [27]. The most recent published work is proposed by Qi [14] 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. Karras et al. [32] proposed a new variaotion of GAN which uses progressive learning strategy instead of directly learns high resolution image. 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 manner to unite generators and discriminators, in which the discriminator will provide the correct information for updating the generator, and accordingly the generator will generate real-look samples. Our model differes from the prior relative works [23,32,39] in several architecture choices for both the G and D networks. Our designed generator is a sequence of deconvolutional layer, however, the discriminator network is structured in reverse order of generator and uses only convolutional layers. We designed our discriminator by using least squre loss fucntion which is able to penalize the wrong samples, while for generator linear activation with Tanh is used. In addition, it is easy to see the training difficulty will be decreased as the dense residual network is adopted into the propsoed architecture. We show that, the usage of dense connections provides deep feature extraction for the network. However, residual connections offer feature reuse to the networks that both are essential for the network. The combination of these models can massively improve the SR performance of the proposed mode. It is worth to mention that, we used same activation fucntion for both the networks as Paramertic rectified linear units (PReLU) [7] after every Conv/D-Conv layers except the last layer.

3. Proposed method

Recently GAN [15] have demonstrated great performance in various tasks. However, in image super-resolution, the quality of images which are generated by GANs still does not meet the real images' resolution. One of the main concerns in this regard can be the loss function; usually, the loss function which is used in some of GAN models only works properly at the initial steps. Consequently, the discriminator cannot provide the right information for updating the generator. In regular GAN, while updating the generator, the embedded loss function creates vanishing gradient problems for the samples which are distinguished as the real samples but in fact, they are far from the real data. In other words, the discriminator fails to properly differentiate between the real and fake samples, and then the information provided for the updating generator will not be accurate, hence, the appropriate generator updating will be impossible. Based on these observations, we adopt the least square loss function for the discriminator in order to settle the current concerns and provide correct information to ease the generator updating. Consequently, the generator can generate samples which are very close to real samples. Updating the generator has a very important role in improving GAN, even though it has been noted that the generator is of most interest in GAN. Hence, this paper attempts to design a new model which can improve the overall result and the stability of the learning process, since the GAN learning is practically unstable. In order to reduce the computational time and ease the training process, we proposed a progressive solution in both generator and discriminator. In the following subsections, we introduce the proposed architecture for the super-resolution problem.

3.1. Network architecture

Each super-resolution phase contains generator and discriminator networks which are shown in Fig. 1. For training, low-resolution image (Image^{LR}) is the form of the high-resolution counterpart (Image^{HR}), obtained by using a Gaussian filter to Image^{HR} and applying downsampling with d factor. The generator is a feed-forward densely residual network, $G_{\theta G}$, parameterized by $\theta_G = W$; that is the weights and b is the biases of the Lth layer. Following equation is used to obtain the parameters,

$$\hat{\theta} = argmin_{\theta G}\frac{1}{N}\sum_{n=1}^{N} L^{SR}\left(G_{\theta G}\left(Image_n^{LR}\right), Image_n^{HR}\right), \tag{1}$$

while L^{SR} is the loss function and $Image_n^{HR}$, $Image_n^{lR}$ are the combination of high-resolution and low-resolution images. As Fig. 1(a) shows in the generator the Image $^{\text{LR}}$ (input image) is passed over several convolution blocks followed by batch normalization and PReLU activation. The output is also passed through a densely connected residual block with skip connections. All blocks have convolutional layers with 3×3 filters size and 64 feature maps. Their output is passed through a sequence of upsampling steps, where each step doubles the size of input image. Finally, the output is passed via a convolution layer to get the super-resolution image (Image^{SR}). Based on the desired scaling, the number of upsampling steps can be modified. The adversarial min–max problem is defined by,

$$min_{\theta G} max_{\theta D} Image^{HR} E_{Ptrain}\left(Image^{HR}\right)\left[logD_{\theta D}\left(Image^{HR}\right)\right] + Image^{LR} E_{PG\left(Image^{HR}\right)}\left[\log 1 - D_{\theta D}\left(G_{\theta G}\left(Image^{HR}\right)\right)\right] \tag{2}$$

A generative network G is trains with the goal of misleading a differentiable discriminator D that is trained to discriminate super-resolution (SR) images from the actual images. G produces solutions that are quite similar to the real images and therefore not simple to classify by D. This inspires perceptually superior solutions and is superior to solutions achieved by minimizing pixel-wise MSE. D handles the expansion problem by using Eq. (2). The discriminator Fig. 1(b) has multiple convolutional layers that are densely connected and used a long skip connection to aggregate the features. The kernels are increasing by a factor of 2 from 32 to 512. Strided convolutions decrease the dimension of the image when the number of features is doubled. The resulting 512 feature maps are followed by fully connected layer and a final softmax activation to get a probability map, which is used to classify the image as real or fake. Algorithm 1 shows the main steps of our proposed model.

3.2. Gradual generator network

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

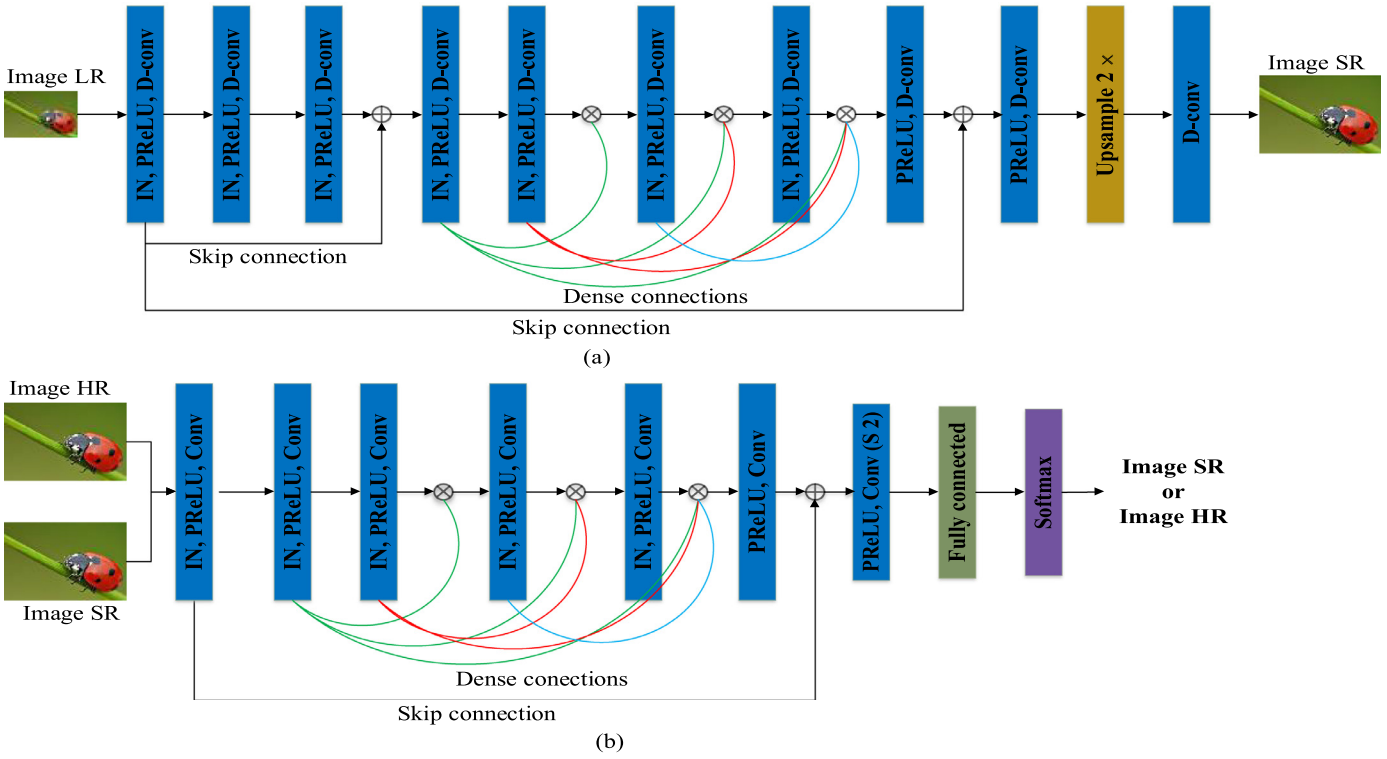


Fig. 1. Proposed network architecture; (a) G- network and (b) D-network. Conv/deconv, denotes as convolutional/deconvolution. The layers with IN indicate that the layer is followed by an instance normalization. S2 is the stride=2 for downsampling operation. FC denotes a fully-connected layer. In the proposed model, D loss is the least square function and G loss is the Tanh.

Algorithm 1 Proposed G-GANISR.

Input: low resolution image X, Generator G, Discriminator D;
Initialize the network and other hyperparameters; θ_g, *and* θ_d
Output: Image super resolution Y
loop
for number of progressive loops **do**
for number of training iteration **do**
sample noise z from p_z : $\{z_1, z_2, \ldots, z_i\}$
sample training data x from true data distribution p_{data} : $\{x_1, x_2, \ldots, x_i\}$
optimize the discriminator parameters θ_d using least square loss function via Eq. (4)
end for
for the number of epochs **do**
sample noise z from p_z : $\{z_1, z_2, \ldots, z_i\}$
sample training data x from true data distribution p_{data} : $\{x_1, x_2, \ldots, x_i\}$
update G's parameters θ_g via Eq. (6)
end for
end for
Return θ_g, θ_d

the fake samples which are generated by the generator. Here we have given a brief description of the proposed GAN framework. We have two networks $G(z;\ \theta^{(G)})$ and $D(x;\ \theta^{(D)})$ as a generator and discriminator. GAN aims to train $G(z;\ \theta^{(G)})$ that can generate samples from the data distribution p_x, by transforming a random input vector z to $x{=}G(z;\ \theta^{(G)})$, while the discriminator $D(x;\ \theta^{(D)})$ learns to recognize whether an image is a generated image or a real one. The GAN can be introduced as:

$$min_G max_D V(D,G) = \mathbb{E}_{x\sim p_x}[log D_{\theta^{(D)}}(x)] + \mathbb{E}_{z\sim p_z}[log(1 - D_{\theta^{(D)}}(G_{\theta^{(G)}}(z)))] \quad (3)$$

where p_x *and* p_z are the empirical distributions of training samples. The proposed architecture for generator is given in Fig. 1 (top image). To simplify the training process for the generator and avoid unnecessary details, we add dense and skip connections between each layer i and n. Each skip connection simply concatenates all channels at layer i with those at layer $n - i$. The discriminator receives two input categories: real images and synthetic images with arbitrary noise. Hence, it must implicitly distinguish two sources of errors: either unrealistic images or realistic images of the wrong class that mismatch the information. On the other hand, G and D must be balanced during training to effectively evolve together; otherwise, if either of them becomes a little stronger, then the training fails. It was observed that the training failure is due to optimization towards divergence between real data and generated samples. It means when we measure the distance between the training distribution and the generated distribution, if there is no overlap between distributions, the gradient will vanish from the discriminator and then stall the training [32]. The regular GAN which was proposed by Goodfellow [15], used Jensen–Shannon divergence as a distance metric [27], and the formulation is still being improved. To make the network efficient in training and have better convergence performance, we adopt dense and skip connections into the generator, similar to [6,36]. We consider the

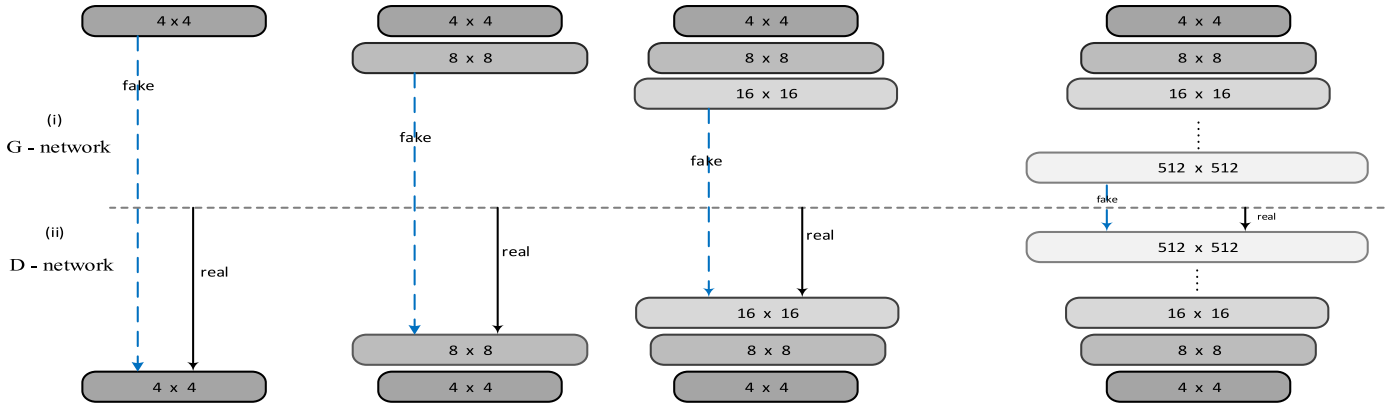


Fig. 2. Gradual learning strategy. The proposed model contains a generator G and a discriminator D, which both trains gradually. We start from a low resolution image of 4×4 pixels. We gradually add layers to G and D; this allows the generated image gradually increases layer by layer; and all the existing layers remain trainable during the process. This learning model produces a high resolution with no lost image details and also significantly speeds up the training process. Note that this figure shows the example of generating images by using gradual growth at 512×512.

discriminator as a binary classification, for the set of input vectors $\{x_i\}_{i=1}^{N}$ with the corresponding target $y \in \{0, 1\}$, where 0 is the predication of real samples, and 1 is the predication of fake samples. In this paper, we improve the GAN formulation by adopting the least square function for super-resolution problem. GAN lacks a proper objective function, and implementing an appropriate loss function is the ongoing topic of several researchers [10,19]. The proposed generator is as follows:

$$DCIPR(K) - DCIPR(2K) - DCIPR(4K) - - - DCIR(128K) - DCIPR(1) - Tanh$$

here K is the low spatial resolution, and the proposed network progressively increases the resolution of the generated images performing $2\times$ upsampling of the previous input (see Fig. 1a). *DCIPR* is a deconvolution layer followed by instance normalization and Parametric rectified linear unit (PReLU) activation [7]. And the last layer of generator contains the Loss function in which we used Tanh. The generator outputs an upscale image of shape 512×512.

3.3. Discriminator-least square loss

Even though there has been a huge progress in GAN, there is still some limitation in the existing models based super-resolution task. The discriminator network cannot accurately distinguish the real samples from the generated samples which cause difficulty in updating the generator. Our objective in this paper is to design a model in which both the generator and the discriminator grow gradually from the small upsampling scale to a large upsampling scale (see Fig. 2). Our proposed model-observation is similar to [28,32,39], however, we used a single generator and a single discriminator with different network architecture. This learning strategy speeds up the training and improves the stability of GAN. In our model, we consider the discriminator as a binary classifier to classify the incoming data as real or fake. The traditional GAN used sigmoid cross entropy loss function, which recently has been replaced by a number of alternatives such as Wasserstein [13] or the absolute deviation [11]. As we discussed before, when updating the generator, we will encounter the vanishing gradient problem for those samples that are on the right side of the margin boundary but in fact, they are far from real samples. This means, if we use fake samples to update the generator and try to convince the discriminator that the samples are real, generally no error is incurred because these samples are placed in the correct side of the margin boundary (where the real samples are placed). In essence, these samples are far from real data, and we just want to pull them towards the real data. Another problem with the default loss functions is their intractable optimization, which is due to non-convexity or discontinuity of the default loss functions. In order to settle this problem, we adopt the least square as loss function for discriminator, since least square function is powerful enough to give a sudden move to the fake samples and pull them towards the margin. The basic idea for least square function states that it is an estimation technique allowing for the straightforward use of regularization. Suppose we want to classify data points as *1* or *0*. For the given data point *"x"*, we calculate a function $f(x)$, if the value of the function is proper and large enough, then that particular point is encoded as *1* or otherwise *0*. In the general loss function for the large value of $f(x)$, if the true label is *1* then the loss goes to zero. These losses are not able to correctly penalize the label points, and just keep them away from intervening with the model updates. However, the least square function is able to strongly penalize the data points for being on either the right or wrong side of the margin boundary with the penalty that is proportional to the distance from the margin boundary (i.e. apply large penalties to samples far over the correct side). In the case that the data are separable, no penalty is incurred. Therefore, based on these observations, we can expect the least square loss function to be able to generate samples which are close to real data. X is data with Y target vector, in which we encode the classes as *0* and *1*. Mathematically, *LS* loss learns parameters by minimizing the following training set:

$$min \sum_{i=1}^{N} (t_i - y_i)^2 = \sum_{i=1}^{N} (1 - z_i)^2 \tag{4}$$

This formula possess several advantages; easy to implement, fast to train, robust and stable. When the target label t is discrete, e.g., binary classification in which t_i contains either *0* or *1*, the residues $\varepsilon_i = t_i + y_i$ typically have much larger magnitudes than other losses. Since, $l_{ls} = (1 - z_i)^2 = \varepsilon_i^2$ is very sensitive to $|\varepsilon_i|$, we used this advantage for GAN to possibly generate data samples which are close to real. On the other hand, *LS* not only penalizes samples which are misclassified ($z_i < 0$) or classified with the improper margin ($z_i < 1$), but also penalizes the samples that are placed on the far distance to the margin boundary on the correct side ($z_i > 1$). These observations may not be ideal for some methods or tasks, but in super-resolution task based GAN models can be considered an advantage. For discriminator D we apply a DCGAN [16] network of stride convolutions to gradually decrease the spatial dimensions. The proposed discriminator D is shown in the bottom part of Fig. 1. The structure of the discriminator network is

as follows:

$$CIPR(128K) - CIPR(64K) - CIPR(32K) - - - CIPR(K) - CIPR(1) - LS$$

here K is the minimum spatial resolution of (4×4), and we progressively decrease the spatial resolution at every layer until it reaches 4×4 spatial resolution. *CIPR* indicates the convolutional layers followed by instance normalization and parametric ReLU (PReLU). And the last layer contains a discriminator loss function, which we proposed to adopt the least square function. The full architecture of gradual learning with its relative details are provided in Fig. 2.

3.4. Model formulation

The high resolution images provide a facility for the discriminator to simply distinguish the fake images from the real images, and then the gradient problem will appear [32]. Let us briefly discuss our approach before focusing on technical details. The proposed network consists of three important parts, generator, discriminator and object function. The generator G: $R^z \rightarrow R^D$ utilizes the pyramid structure $\{l_1, l_2, l_3\}$ to extract features in different scales. As the gradual upsampling strategy has been used to improve the learning process then the generator merges the input details according to the features, as shown in the top part of Fig. 1. Accordingly, discriminator D: $R^D \rightarrow \{0, 1\}$ acts as a guide to update the G parameters through distinguishing the synthesized images from the corresponding ground truth. We adopted the least squares loss function to settle the gradient vanishing problem, and it could help the generator to generate samples which are close to real data. We consider the discriminator as a binary classification problem. The least square loss function is the simplest and also an inexpensive function, which can significantly save the computational efficiency. The traditional objective function of GAN can be expressed in the optimization problems as:

$$\min_G \max_D \mathbb{E}_{x\sim Pdata(x),\ \ z\sim p(z)}[\log(1 - D(x, G(x, z)))] + \mathbb{E}_{x\sim pdata(x,y)}[\log D(x)] \tag{5}$$

where x is the input data samples and z is the random noise vector. Several works have been done to optimize the above function to find the best parameters in G to fully complicate D while updating the weights in D to accurately distinguish the fake samples generated by G and the ground truth. In this paper, we improved the GAN performance by adopting the least square loss function to improve discriminator's prediction and then it could pass correct data for updating the generator.

$$\begin{aligned}\min_D V_{LS-GAN}(D) &= \frac{1}{2}\mathbb{E}_{x_1, x_{2\sim Pdata(x_1).Pdata(x_2)}}\left[(D(x_1, x_2|y) - b)^2\right] \\ &\quad + \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}\left[(D(G(z|y)) - a)^2\right] \\ \min_G V_{LS-GAN}(G) &= \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}\left[(D(G(z|y)) - C)^2\right]\end{aligned} \tag{6}$$

According to Eq. (5), the objective function is transformed into two Min problems; $\min_D V_{LS-GAN}(D)$ and $\min_G V_{LS-GAN}(G)$. We have shown the input low resolution images as X, and y is the ground truth. The corresponding targets are indicated by a & b, in which: a is the fake samples and b is the real samples. The value which is used to convince the discriminator that the generated samples are real is indicated by C. The main advantage of using least square as objective function is its strict nature for penalizing the samples. In contrast to the existing GANs based methods, which generally causes no loss for the samples that are in the far distance on the correct side of margin boundary, the Least Square would apply a large penalty to the samples which are far from the correct side or even the wrong side of the margin boundary. In fact, it pulls up all the samples (either correct or wrong side) towards the margin in order to participate in classification. Further, when the generator starts the updating, the model keeps the discriminator's parameters fixed, and hence, this penalization helps the generators to generate the samples that are quite similar to real. The GAN model attempts to minimize the objective function by training the discriminator in order to learn the objective function to have a stable generator. To minimize the divergence between generator and data distributions, the following equation is used:

$$\begin{aligned}C(G) &= 2D_{JS} - \log(4) \\ C(G) &= D_{KL}\left(p_{data}\|\tfrac{1}{2}(p_{data}+p_q)\right) + D_{KL}\left(Q\|\tfrac{1}{2}(p_{data}+p_q)\right) - \log(4)\end{aligned} \tag{7}$$

if $p_{data} = p_q$, then we will achieve the $C(G) = -\log(4)$, as above Eq. (6). Since the regular GAN is more general, inspired by the work which proposed by Nguyen et al. [27], we extend the vibrational divergence to recover the GAN and generalize it to random f-divergence. For two data distributions P and Q that possess density functions p and q, respectively, based on dx measure, we have the following equation.

$$C(G) = (\mathbb{E}_{x\sim P}[D(x)]) - \mathbb{E}_{x\sim Q}[f^*(D(x))],\ \ D : x \rightarrow R \tag{8}$$

Based on [25], we can derive the D for G. Note that a and b are the targets in this work, i.e. fake generated sample a, and real samples b:

$$D_G^*(x) = f'\left(\frac{p(x)}{p(x)+q(x)}\right) = \frac{bp_{data}(x) + a\ p_q(x)}{p_{data}(x) + p_q(x)} \tag{9}$$

$$= \mathbb{E}_{x\sim Q}\log\frac{Q(x)}{P(x)} = \mathbb{E}_{x\sim Q}\log\frac{1 - D^*(x)}{D^*(x)} \approx \mathbb{E}_{x\sim Q}\log\frac{1 - D(x)}{D(x)} \tag{10}$$

In practice, when the performance of G is poor, then D rejects the samples with high confidence because the generated samples are strongly different from the training data. Hence, instead of training G in order to minimize $g\ (1 - D(G(z)))$, we can train G to maximize $\log(G(z))$. Moreover, D is allowed to reach its optimum and estimate the probability $P(Y = y|x)$, where y is a target which indicates that whether the generated sample x is part of p_{data} or p_q, and then p_q is updated to improve Eq. (4):

$$\begin{aligned}C(G) &= \mathbb{E}_{x\sim pdata}[\log D_G^*(x)] + \mathbb{E}_{z\sim pz}[\log(1 - D_G^*(G(z)))] \\ &= \mathbb{E}_{x\sim pdata}[\log D_G^*(x)] + \mathbb{E}_{x\sim pq}[\log(1 - D_G^*(x))]\end{aligned} \tag{11}$$

then p_q convergence to p_{data}. Based on these observations, as we have a binary output "a (fake samples) and b (real samples)", we optimized Eq. (11) as:

$$\begin{aligned}C(G) &= \mathbb{E}_{x\sim pdata}\left[\left(\frac{bp_{data}(x) + ap_q(x)}{p_{data}(x) + p_q(x)} - C\right)^2\right] \\ &\quad + \mathbb{E}_{x\sim pq}\left[\left(\frac{bp_{data}(x) + ap_q(x)}{p_{data}(x) + p_q(x)} - C\right)^2\right]\end{aligned} \tag{12}$$

We explore the relation of using least square and f-divergence loss functions as:

$$\begin{aligned}\min_D V_{LS-GAN}(D) &= \frac{1}{2}\mathbb{E}_{x_{\sim Pdata(x)}}\left[(D(x|y) - b)^2\right] \\ &\quad + \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}\left[(D(G(z|y)) - a)^2\right] \\ \min_G V_{LS-GAN}(G) &= \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}\left[(D(G(z|y)) - C)^2\right] \\ &\quad + \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}\left[(D(G(z|y)) - C)^2\right]\end{aligned} \tag{13}$$

Table 1
Our architecture trained with different losses in term of PSNR on four benchmark datasets: Set5, Set14, BSD100, and Urban100. Least square loss yields the highest PSNR values since it strongly minimized the distance to ground-truth. The models trained with other losses yield lower PSNRs. Best results are shown in bold.

Dataset	Perceptual loss (l_p)	Baseline with MSE (l_e)	Wasserstein (W)	Least square (LS)
Set5	27.20	26.15	**29.34**	**31.84**
Set14	26.52	25.93	27.11	**29.67**
BSD100	24.18	23.49	**28.58**	**28.12**
Urban100	26.07	25.81	25.36	**27.96**

Further, to gain stability, we replaced batch normalization with instance normalization. The instance normalization computes the mean and the variance of the samples in each mini-batch and performs normalization, which fits the samples in the mini-batch to balance distribution. We have listed the main steps of our proposed model in Algorithm 1.

4. 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, ResGAN, GP-GAN, and DCGAN. This paper used four benchmark datasets for the experiments including; Set5, Set14, BSD100, and Urban-100. All experiments are achieved with the highest scale factors, 4×, 6× and 8× between low and high-resolution images. We have used the following measures to fairly evaluate the performance of different methods: Structural Similarity Index (SSIM), Visual Information Fidelity (VIF) and Peak Signal to Noise Ratio (PSNR); in these metrics the lower scores indicate higher diversity of generated images. To validate our model, we compared it with state of the arts GAN and other deep learning methods. In particular, we used the most recent GAN based super resolution approaches as baselines including; Res-GAN [34], E-Net [35], DiverseGAN [19], DCGAN [16], WGAN [13], CGAN [44], UR-GAN [4], SRGAN [36], SRPGAN [37], ProGAN [39], SFT-GAN [40] and StackGAN [22], and the most fast CNN based methods including DRRN [38], VDSR [41], SRDenseNet [10]. We re-implemented the baselines based on their online material supplementary and provided source-code. The entire network is trained on Intel i7-6850, 64GB RAM and GeForce GTX 1080 Ti and used torch and tenserflow framework. In the following subsections, we explain the training details and parameters used for our super-resolution model.

4.1. Implementation details

Based on above intuitions, we enhance GAN performance by proposing a new model, in which G is a deep deconvolution contains dense and skip connections with a progressive structure (gradual upsampling in each layer) and D is a deep convolution adopt least square function as an objective function in order to encourage more convergence between the generated samples x_q and real data x_{data}. For the gradual structure, we followed the recent work in [32]. We use all the training data with the n classes x^n to train the generator G (z), and z is the random noise input. Our basic contribution is to train GAN with low resolution images (as input), and then progressively increase the resolution by adding layers to the network as depicted in Fig. 2. This gradual strategy allows training to first learn from large scale structure and then from the finer scale details (i.e. termed as learning from easy to difficult). We have given an example to show how we can progressively increase the proportion of the image. Let i_{max} be the maximum iteration, i_t is the number of current iterations, then the proportion of the image at the current iteration will calculated by; $P_{sk} = 0.1 + \min(0.8, (\frac{i_t}{i_{max}})^*)$, where * is the adjustable hyperparameter and indicates the * as a default value which is *1* in this experiment. Moreover, GAN is often prone to increases in the signal magnitudes and it causes an unethical competition between two networks (generator and discriminator).

Therefore, the earlier methods suggested addressing this issue by setting batch normalization in both generator and discriminator. These normalization methods usually adapted to eliminate the covariate shifts. However, we believe the actual need in GAN based image super-resolution is to constrain noises and competitions since we believe that more details would provide better resolution. Thus, we used a different approach and substituted the instance normalization [43] with the batch normalization. Moreover, we thoroughly set the weight initialization by using trivial $N(0, 1)$ and explicitly scaled the weights as: $W_i = \frac{w_i}{c}$ where c is the normalization in each layer and w is the weight; this initializer scenario is proved by He et al. [7]. In fact, this strategy helps to stay independent from the scale parameter during the training. To control the magnitude in the generator and discriminator, we normalize the feature vector in the generator after each convolution layer. To configure this step, we followed the work that was proposed by Krizhevsky et al. [8] as: $b_{x,y} = a_{x,y}/\sqrt{\frac{\sum_{j=0}^{N-1}(a_{x,y}^j)^2 + \epsilon}{N}}$ where N is the number of features and *a, b* are the original and normalized feature vectors. As suggested by He et al. [7], we used PReLU activation in the generator and discriminator for all layers, except for the output which uses a Tanh function, since this strategy allowed the model to learn quickly [16].

Replacing batch normalization by instance normalization helps deal with the training problem which is due to poor initialization and allows the gradient flow in deeply. This could prevent the generator from failing most of the samples by a few points which is a common problem in GAN. Additionally, we have not applied any pre-processing approaches; all weights were initialized from a zero-centered normal distribution with 0.02 deviation. We observed the 0.001 learning rate is too high, thus used 1×10^{-4} and then decayed by a factor of 2 every 100 iterations. Also, momentum term β_1 with the standard value of *0.9* does not match with our parameters and we set the Adam optimizer [12] with value *0.5* instead. Based on Fig. 2, our training starts by both *G* and *D* having low resolution images, and then we gradually add layers to *G* and *D*, and hence increase the resolution of the generated image. Our sample images will progressively grow at 512×512. The discriminator network is structured in reverse order of generator, and its input layer is stated by 512×512. All convolutional layers in D are composed of kernels of size 5×5 with a stride 2 followed by instance normalization. All deconvolution layers in G are composed of the kernel of size 3×3 with stride 2 followed by instance normalization. The first layer in G is the latent samples, which is followed by a fully connected layer. The last layer in D is the loss

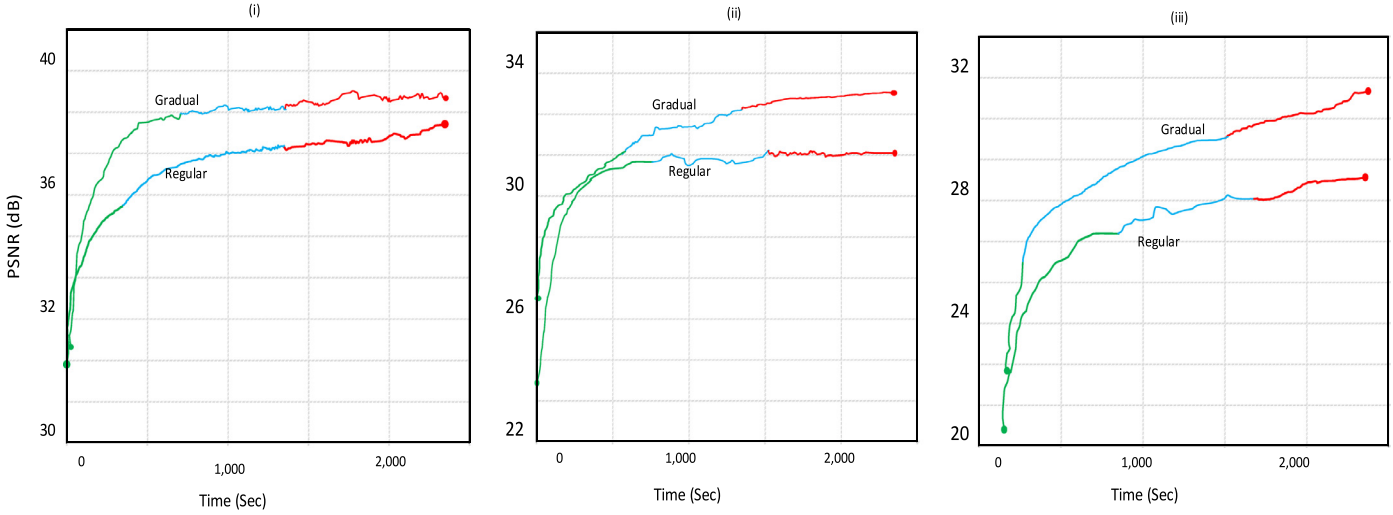


Fig. 3. Effect of gradual learning on training speed and convergence. We trained three scale factors (4 ×, 6 ×, 8 ×) to plot the PSNR. (i), scale factor 4; (ii) scale factor 6; and (iii) scale factor 8. We show the expenses-time by using line colors. The red lines need more time to train and the green lines indicate it requires much less time to reach the highest results. As observed, the 6 × requires less computation and times for each update compared to 8 ×.

Table 2
Comparison of our model with the state of the art GANs in term of PSNR on four benchmarks; Set5, Set14, BSD100, and Urban100. The first best results are stressed by blue color and the second best results with green color.

Models	Set5	Set14	BSD100	Urban100
WGAN [13]	27.09	29.63	26.76	25.08
DCGAN [16]	29.17	28.51	23.49	21.84
cGAN [44]	23.42	25.10	26.53	20.62
Res-GAN [34]	25.83	27.06	21.28	19.95
GP-GAN [9]	29.96	24.17	28.93	28.15
G-GANISR	31.62	29.85	30.17	29.60

function; we used least square function to pass the correct samples for possibly updating G. The experimental results are summarized in Tables 1–4, and Figs. 3–9.

4.2. Comparisons and results

We have two sets of experiments; one is the proposed progressive strategy and another one is the regular GAN. The result of progressive GAN is presented in Fig. 3. We trained different upscaling factors; 4×, 6× and 8×. The curve shows the relationship between the PSNR results and the expenses time (min). The results show that gradual GAN considerably reduces the training time and the resolution quality is improved much faster compared to regular GAN. As the results shown, the 6× scale factor requires less training time for each update. In both plots, progressive learning requires less time to reach the highest PSNR. Despite great progress in GAN models, it is still a fancy model for super-resolution problems. Typically, GAN suffers from a training process which is an unstable and lengthy process. We evaluated the effect of our adaptive least square loss function with the different objective function. Quantitative results are summarized in Table 1. From the results, it is observed that our networks with least square loss function empirically yield the highest PSNR values compared to other losses. However, the Perceptual loss could achieve the lowest values as compared to other loss functions. Furthermore, we compare the performance of our model with five states of the art GAN methods; WGAN [13], DCGAN [16], CGAN [44], Res-GAN [34], GP-GAN [9]. We selected the baselines based on their structures which are similar in some way to our proposed model but the difference is on loss function and the learning strategy. Quantitative results in terms of PSNR 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 second best results belong to DCGAN and GP-GAN. Our proposed model outpaces these two methods by 1–1.8% ratio. Res-GAN despite having smooth training process did not show a stunning performance. Moreover, in order to prove the stability of the proposed model, we compared it with WGAN which is the state-of-the-art GAN in stability. We followed the structure of [15] and set the hyperparameters c *and* γ to 30 and 150. With progressive growth, in both the generator and discriminator, firstly, the starting layers converged at the initial steps of the network, and then refined the representation by progressive small scale increases in the new layers. We found that the GAN based least square loss function outpaces the WGAN [13]. Note that WGAN, despite showing low performance (Table 1) has a stable and smooth training process. For the evaluation, Set5, Set14, BSD100 and U-100 datasets have been used.

Another experiment is to train the GAN with different architecture, i.e. we want to show the similarity of the generated image to the ground-truth under various conditions. In fact, we want to show the transformation of the same image when using instance normalization and batch normalization (BN) in the GAN network. The first experiment is used by including BN, the second

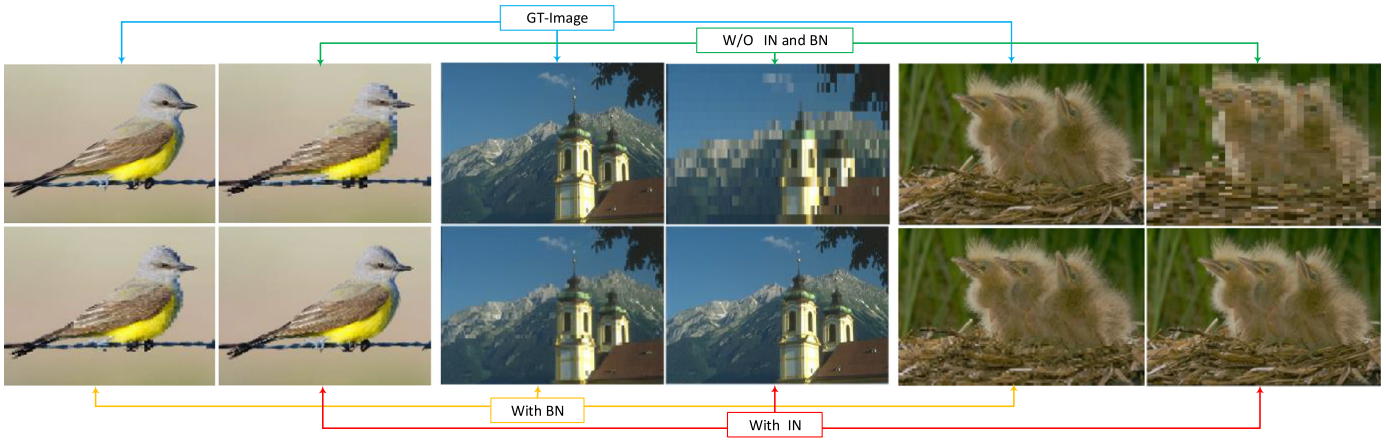


Fig. 4. We implemented our architecture under various conditions; we used three sample images dark and colorful background. From left to right: in each sample, the first top images are ground truth that indicated as "GT image". Also the second top image in each sample is generated without using IN/BN in G that indicated as "W/O IN and BN". At the bottom the first images are generated by using BN in G and D that indicated as "W/BN"; also, the second image at bottom is generated by IN in G and D that indicated as "W/IN". As it is obvious, the image which is generated by instance normalization in both G and D is more similar to Ground-truth. The worst is skipping BN and IN from architecture and we train the model without it. In all training setting, we used Adam and set α *and* β to 0.002 and 0.5, respectively.

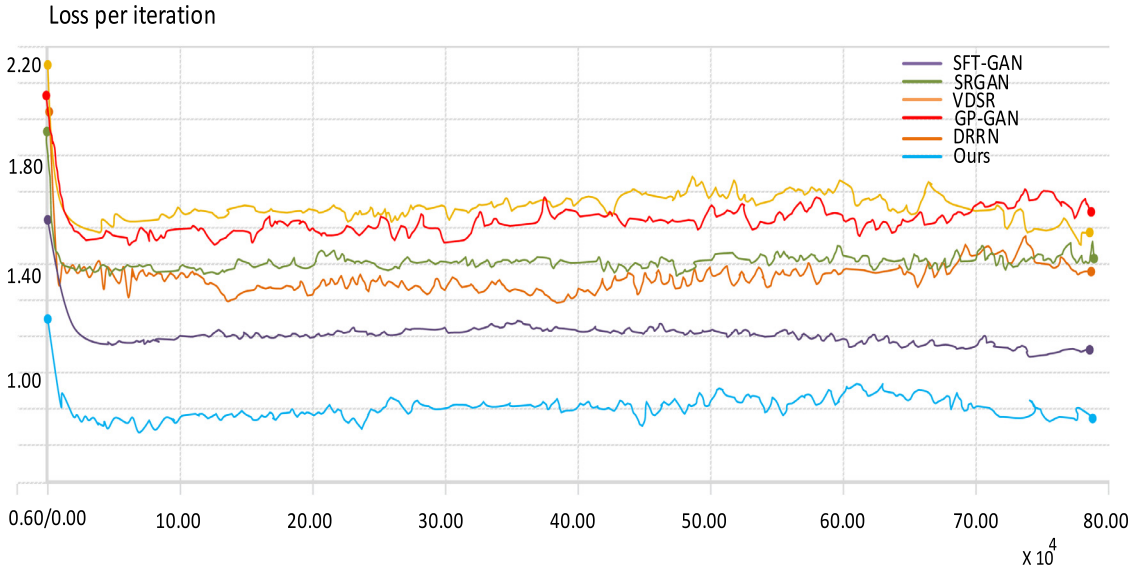


Fig. 5. Performance based on loss evolution: Proposed model vs. other GAN approaches at scale factor 6 on BSD100 dataset. The lowest curves (blue) denote our model, in fewer iterations, it reached lower loss compared to others. The highest loss is with SFT-GAN (orange curve).

experiment is used by replacing instance normalization (IN) with BN, and the last one is used by excluding BN and IN in the generator. Note that, in all experiments, we used Adam optimizer [12]. The result is presented in Fig. 4. As it is observed; the image which is generated using IN is more similar to the real image than by using other techniques. However, the worst case belongs to the result of excluding BN/IN, which has a severe degree of mode collapse. Note that ground truth is available for this result (first image from the right).

We also have shown the evaluation of our model with other baselines in term of Loss in Fig. 5. If we pay attention to the results, we can see how the blue curve (our network) shows the lower loss at the beginning of training with less iteration compared to other methods which means that the proposed network can converge to the better results, thus we can conclude that our model needs significantly fewer iterations to reach a lower loss.

We also investigate the influence of long skip connection and dense connection on the number of epochs based on PSNR and spending times. The result is plotted in Fig. 6, and is evaluated on BSD100 dataset for $8\times$ scaling factors. Times is assessed on the same machine that we implemented all the experiments. We conducted four sets of experiments such as; our proposed architecture which contains skip and dense connection (blue curve), our model with using only dense connection (red curve), our model with using only skip connection (purple curve) and our gradual learning architecture without using skip and dense connection (green curve). The results clearly convey that, dense connection significantly improve the PSNR results and it reach to the high value in PSNR after initial epochs compares to the skip connection (left plot). However, dense connection requires more computation cost comparing to the skip connection (right plot). Our proposed model by using dense and skip connection could significantly reduce the times while preserving a high values for PSNR in all the epochs. Usage of dense connections provides deep feature extraction for the network. However, residual connections offer feature reuse to the networks that both are essential for the network. The combination of these models can massively improve the SR performance of the proposed mode.

The performance of our proposed model in term of PSNR, SSIM, and VIF is given in Tables 3 and 4. We compared our model to

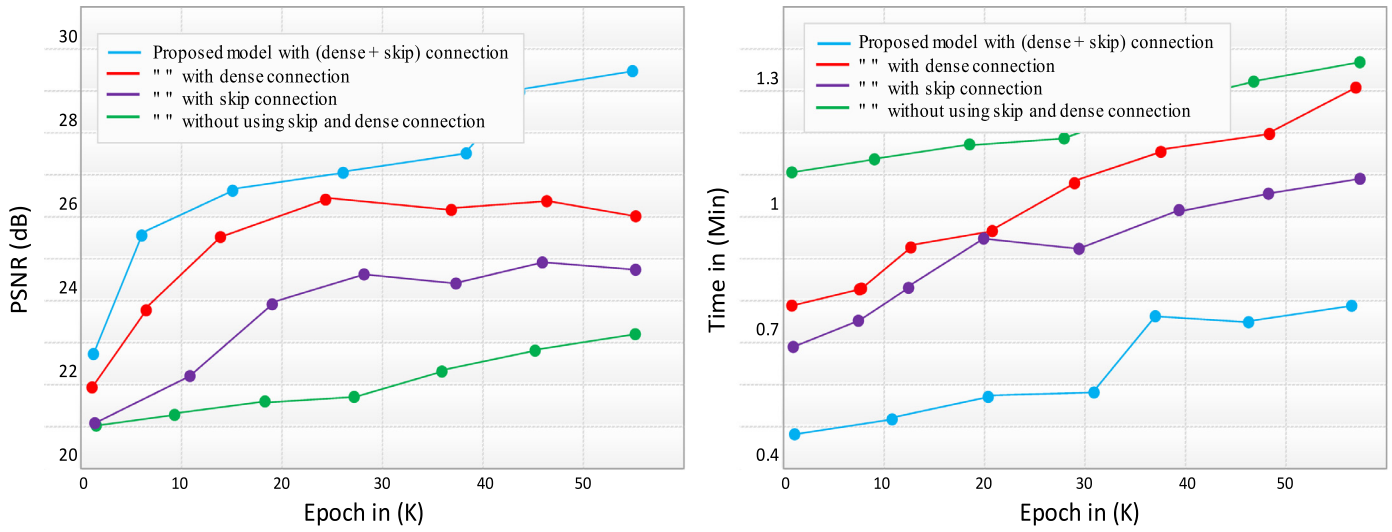


Fig. 6. Performance of runtime and PSNR on the various network structures. We used four sets experiments, on BSD100 dataset at the highest upscaling factor 8 ×. PSNR is evaluated on the left figure and time is assessed in the right figure. As it observes, the blue curve (our proposed model) needs less times for performance. Dense connection clearly improve the network (high value in PSNR), however needs more times for the convergence. Skip connection need less time for the convergence but it can't preserve the good value for PSNR.

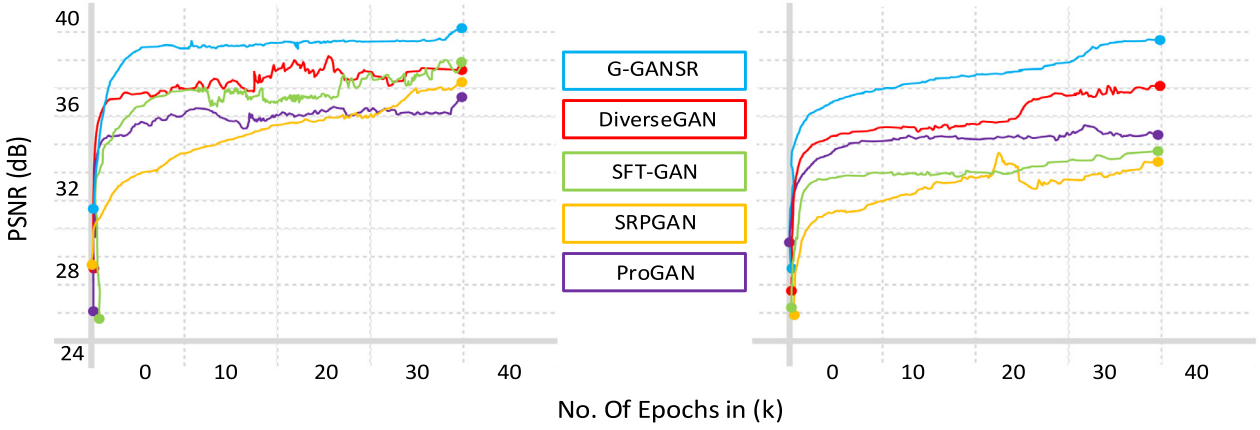


Fig. 7. Convergence curve based for different GAN based approaches. Left image: is evaluated on BSD100 dataset; Right image: is evaluated on Urbant100 dataset. The baselines are DiveseGAN [19], SFT-GAN [40], SRGAN [36] and ProGAN [39]. In both datasets, our proposed model outperform the baselines in term of PSNR.

Table 3
Comparison of Isola et al. [21], E-Net [35], ProGan [39], DRRN [38], SRPGAN [37], DCGAN [16], SRDenseNet [10], UR-DGN [4] and our proposed model on four benchmark datasets: (Set5 and Set14). The highest measures are (PSNR [dB], SSIM, VIF) in bold and blue, the second highest in green. [6× and 8× scale factor].

Scale factors		Measures	G-GANISR	Isola et al.	E-Net	ProGan	DRRN	SRPGAN	DCGAN	SRDenseNet	UR-DGN
SEt5	× 6	PSNR	**33.82**	28.87	27.64	29.07	27.19	**31.04**	**32.17**	24.81	26.57
		SSIM	**0.9124**	**0.9008**	0.8253	0.8817	0.8362	0.8729	**0.9097**	0.7884	0.8692
		VIF	**0.4108**	0.4016	0.3597	0.2816	0.3184	0.3875	**0.4155**	0.2512	0.3864
	× 8	PSNR	**31.11**	28.36	25.79	28.34	24.87	27.54	**31.40**	21.53	21.97
		SSIM	**0.9082**	0.8851	0.8104	0.8801	0.8290	0.8511	**0.9103**	0.7639	0.8303
		VIF	**0.4096**	0.3817	0.3451	0.2796	0.3027	0.3604	**0.4108**	0.2405	0.3271
Set14	× 6	PSNR	**30.56**	23.97	29.64	29.54	25.97	26.19	**30.02**	26.37	27.15
		SSIM	**0.8881**	0.7805	**0.8703**	0.8301	0.7924	0.8187	**0.8934**	0.8011	0.8415
		VIF	**0.3789**	**0.3695**	0.3414	0.3248	0.3052	0.3441	0.3615	0.2652	0.3202
	× 8	PSNR	**28.07**	20.98	23.51	**28.14**	24.81	22.43	**29.17**	21.84	20.99
		SSIM	**0.8803**	0.7765	0.8094	0.8097	0.7734	0.8017	**0.8733**	0.7918	0.8015
		VIF	**0.3652**	0.3148	0.2940	0.3011	0.3016	0.3102	**0.3542**	0.2591	0.3048

Table 4
Comparison of Isola et al. [21], E-Net [35], ProGan [39], DRRN [38], SRPGAN [37], DCGAN [16], SRDenseNet [10], UR-GAN [4] and our proposed model on four benchmark datasets: (BSD100 and U100). The highest measures are (PSNR [dB], SSIM, VIF) in bold and blue, the second highest in green. [6× and 8× scale factor].

Scale factors		Meastures	**G-GANISR**	Isola et al.	E-Net	ProGan	DRRN	SRPGAN	DCGAN	SRDenseNet	UR-GAN
BSD100	× 6	PSNR	**31.23**	29.97	30.59	**31.43**	26.38	27.48	**31.07**	22.53	27.64
		SSIM	**0.9273**	**0.9211**	0.9218	**0.9237**	0.8312	0.8652	**0.8942**	0.7958	0.8619
		VIF	**0.4052**	**0.4103**	0.3988	0.4003	0.3274	0.3497	0.3996	0.2374	0.3791
	× 8	PSNR	**29.18**	27.53	25.62	**29.07**	20.79	23.18	**29.68**	20.97	21.02
		SSIM	**0.9065**	**0.9088**	0.8716	0.8938	0.7968	0.8504	**0.8824**	0.7851	0.7968
		VIF	**0.3852**	**0.3914**	0.3648	0.3714	0.2899	0.3259	0.3691	0.2205	0.3154
Urban100	× 6	PSNR	**29.18**	28.15	27.12	28.72	25.45	26.92	28.58	23.81	24.36
		SSIM	**0.8803**	**0.9018**	0.8739	0.8526	0.8192	0.8451	**0.8805**	0.7716	0.8523
		VIF	**0.4021**	**0.3801**	**0.3824**	0.3582	0.3174	0.3267	0.3618	0.2281	0.3218
	× 8	PSNR	**27.23**	22.85	**24.67**	20.66	21.84	23.49	21.92	18.64	20.08
		SSIM	**0.8750**	0.8309	**0.8519**	0.8063	0.7896	0.8016	0.8138	0.7565	0.8039
		VIF	**0.3941**	0.3364	**0.3653**	0.2938	0.3009	0.3103	0.2997	0.1930	0.2797

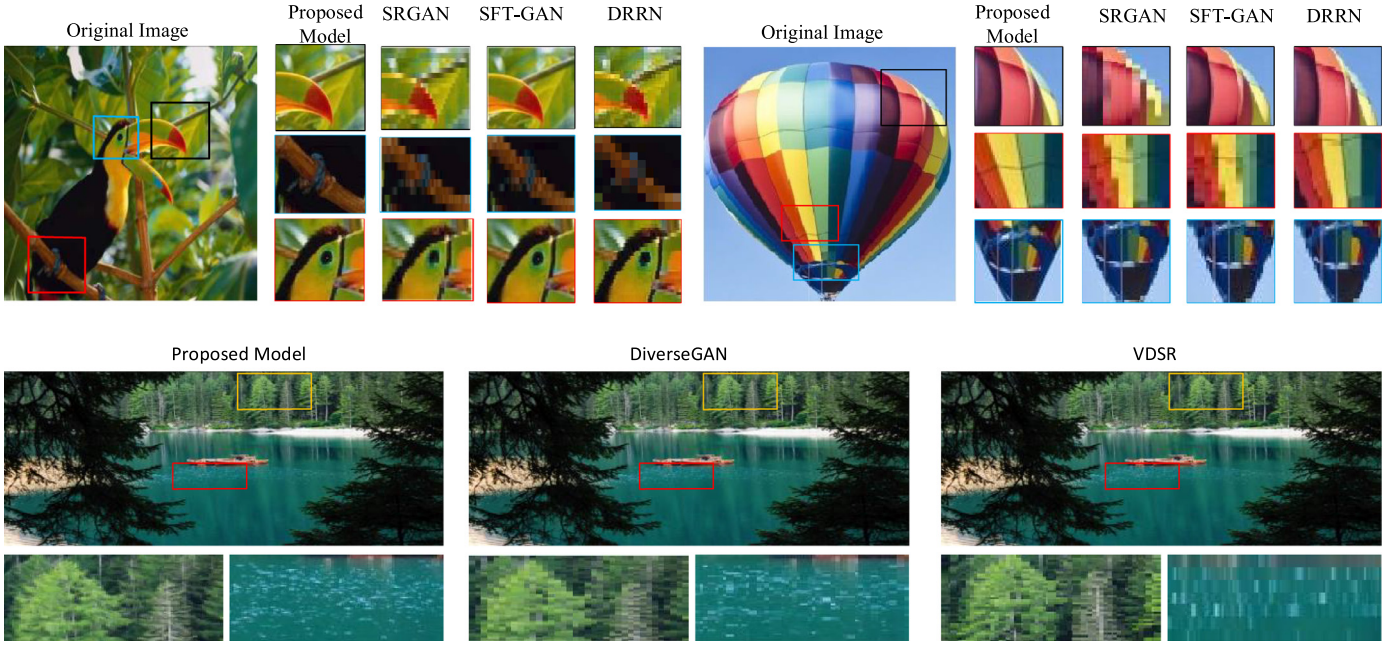


Fig. 8. Comparison of the proposed model with other methods at 4× and 8× scales super-resolution. We used three sample images. The GAN based methods clearly outperform PSNR-oriented approaches in term of perceptual quality. The tops images are evaluated at 4× upscaling; from left to right, the first images in each sample indicate as Ground-truth, the results of our proposed model "G-GANISR" based least square loss function. The baselines that used are SRGAN [36]; SFT-GAN [40]; DRRN [38]. The bottom images are evaluated at 8× upscaling; we used DiverseGAN [19] and VDSR [41] as baselines. For the best review, zoom in and then it is seen that our proposed model is capable of generating a richer and more realistic image among different categories. The second best result was evaluated with SFT-GAN which tends to produce unpleasant texture.

several GAN based and non-GAN models. In the experiments, we train the networks with two highest scaling factors; 4× and 8×. The table implies that the results of methods based on GAN outpace other non-GAN. 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 the techniques proposed by [21] and DCGAN [16]. Based on BSD100 dataset, the second best results is for ProGAN [39] and [21]. The best results in Urban100 datasets is for proposed G-GANISR, [21] and [35]. For Set5 and Set14 datasets, our proposed model and DCGAN [16] perform better than other methods. However, for the high scaling factor 8×, the second best method is DCGAN that shows better performance compared to other prominent methods. For the 4× scaling factor, the second best results are for [21] and [39] methods.

Next, we focus on convergence comparisons between five GAN-based methods, including ours: DiverseGAN [19], SFT-GAN [40], SRPGAN [37], ProGAN [39], and our G-GANISR, and the result are reported in Fig. 7. In particular, we use a low resolution image with a spatial resolution of 128 × 128, and perform a scale factors; 4×. As it observes, the proposed model is achieved the convergence

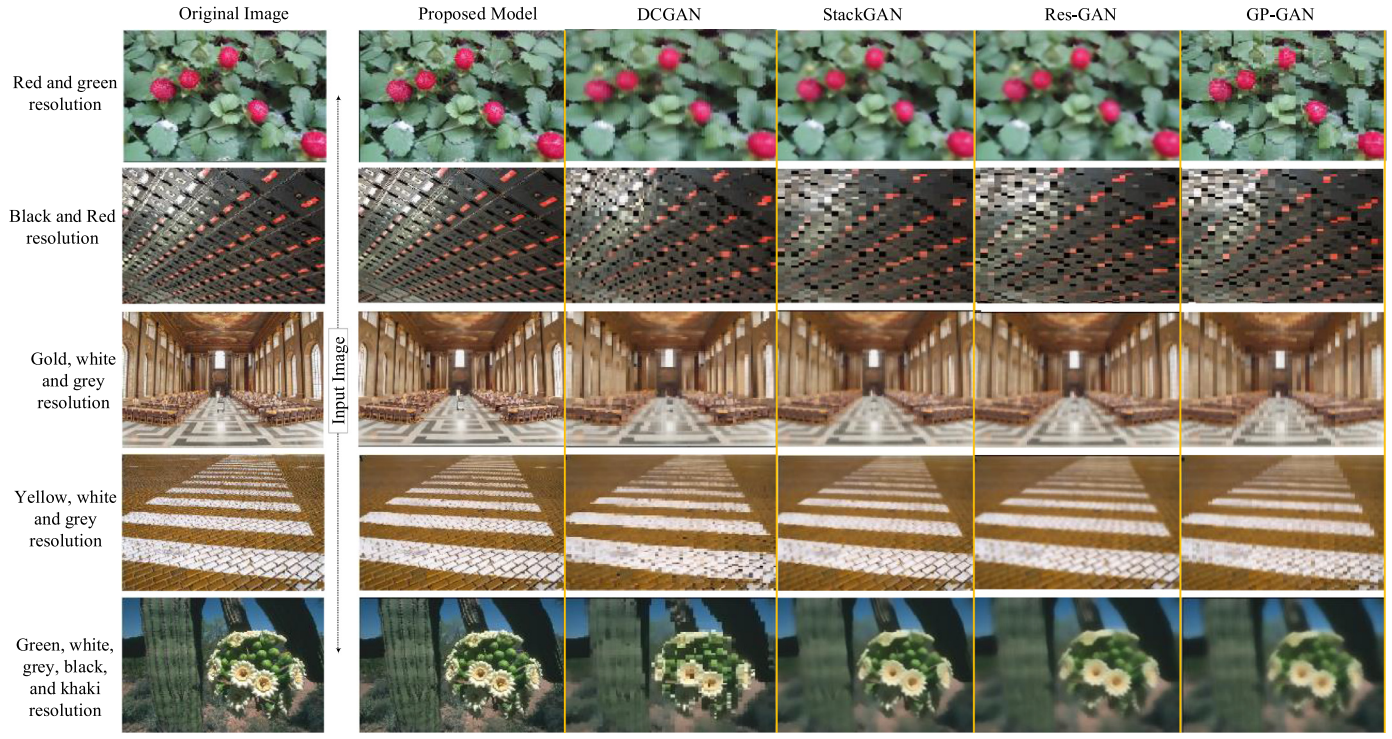


Fig. 9. The generated images with different state of the art methods. We improved the visual quality of the generated images using proposed network. From left to right, first column is the original image, second column is the generated images from proposed network "G-GANISR" and next five columns are the generated images from other techniques which are; DCGAN [16], StackGAN [22], Res-GAN [34] and GP-GAN [9], respectively. The SR images generated from the same LR images (i.e. in each row the input which is the first left image is the same). As it observes, the images generated by our proposed network have better visual quality, comparing to others.

after a few epochs with a high value in PSNR. Note that, the results is based on BSD100 and Urban100 datasets at 4 upscaling factor.

Fig. 8 presents an overview of different approaches including the current state of the art in terms of PSNR at 4× and 8× scaling factor. We selected three 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 [40] 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. The experiments proved our claim regarding the performance of the proposed gradual GAN.

We show a visual quality of baselines and our proposed model in Fig. 9. As it observes, our proposed network which is incorporating gradual learning in GAN and adopting least square loss function in discriminator is able to generate more plausible details and the results significantly improved the visual quality of the images. However, other methods fail to recover the fine structure, and thus generate blurred images. One of the reasons can be due to information which is provided by discriminator in order to update the generator. By providing the incorrect details for updating generator, the network will fail to produce the high quality image. Therefore as we noted before, if we adapt a suitable loss function in discriminator which could provide us the most plausible information, then the generated image will be closed to the original image.

5. Conclusion

In this paper, we address three well-known issues in image super resolution approaches; improving the image resolution in particular perceptual quality, because adversarial training generally produces artifacts in the outputs which can degrade the image textures. Second component lies on improving the training stability. And the third component is to improve the model in term of runtime. Thus, we proposed an efficient GAN model which is able to produce state of the art results based on quantitative and qualitative measures. The proposed model consists of a generator and a discriminator with different loss functions. In fact, we gradually improve the image texture step by step for the generator and discriminator to optimize the networks. This learning strategy helps to balance both networks in order to obtain stable results, and hence the presented technique can efficiently learn to reconstruct the high-resolution results step by step. Moreover, we introduce a new loss function for the discriminator which is able to accurately distinguish the fake samples from the real samples and then pass the true information for updating the generator. We believe that our proposed model not only has a simple implementation in comparison with the other GAN based methods but also presents superior results. In essence, this work concludes two main aspects. The first is that the loss function which is used for the discriminator significantly improves the GAN performance by guiding the processing of updating the generator. And the second lies in the learning structure (gradual growing) which we believe is more stable and efficient for generative networks.

Declaration of Competing Interest

None.

References

[1] P. Shamsolmoali, M. Zareapoor, D.K. Jain, V.K. Jain, J. Yang, Deep convolution network for surveillance records super-resolution, Multimed. Tools Appl. (2018) 1–15, doi:10.1007/s11042-018-5915-7.

[2] 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.

[3] 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.

[4] X. Yu, F. Porikli, Ultra-resolving face images by discriminative generative networks, in: Proceeding of European Conference on Computer Vision, ECCV, 2016, pp. 318–333.

[5] J. Park, B.G. Nam, H.J. Yoo, A high-throughput 16× super resolution processor for real-time object recognition soc, in: Proceedings of European Solid-State Circuits Conference, ESSCIRC, 2013, pp. 259–262.

[6] Z. Feng, J. Lai, X. Xie, J. Zhu, Image super-resolution via a densely connected recursive network, Neurocomputing 316 (2018) 270–276.

[7] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, ICVPR, 2015, pp. 1026–1034.

[8] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, J. Commun. ACM 60 (6) (2017) 84–90.

[9] H. Wu, S. Zheng, J. Zhang, K. Huang, GP-GAN: Towards realistic high-resolution image blending. arXiv preprint arXiv:1703.07195, 2017.

[10] T. Tong, G. Li, X. Liu, Q. Gao, Image super-resolution using dense skip connections, in: Proceedings of International Conference on Computer Vision, ICCV, 2017, pp. 4799–4807.

[11] J. Zhao, M. Mathieu, Y. LeCun, Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126, 2016.

[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] Guo-Jun Qi, Loss-sensitive generative adversarial networks on lipschitz densities. arXiv preprint arXiv:1701.06264, 2017.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings 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. arXiv preprint arXiv:1511.06434, 2015.

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

[18] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of wasserstein gans, in: Proceedings of NIPS, 2017, pp. 5769–5779.

[19] M. Zareapor, M.E. Celebi, J. Yang, Diverse adversarial network for image super-resolution, Signal Process. Image Commun. 74 (2019) 191–200.

[20] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, H. Lee, Generative adversarial text-to-image synthesis, in: Proceedings of International Conference on Machine Learning, ICML, 2016, pp. 1060–1069.

[21] P. Isola, J.-.Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceeding of Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 1125–1134.

[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, in: Proceeding of International Conference on Computer Vision, 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] L. Guimin, W. Qingxiang, Q. Lida, H. Xixian, Image super-resolution using a dilated convolutional neural network, Neurocomputing 275 (2016) 1219–1230.

[25] T. Xiao, J. Zhang, K. Yang, Y. Peng, Z. Zhang, Error-driven incremental learning in deep convolutional neural network for large-scale image classification, in: Proceedings of the ACM International Conference on Multimedia, 2014, pp. 177–186.

[26] Y. Liang, J. Wang, S. Zhou, Y. Gong, N. Zheng, Incorporating image priors with deep convolutional neural networks for image super-resolution, Neurocomputing 194 (2016) 340–347.

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

[28] I. Durugkar, I. Gemp, S. Mahadevan, Generative multi-adversarial networks. arXiv preprint arXiv:1611.01673, 2016.

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

[30] A. Ghosh, V. Kulharia, V.P. Namboodiri, P.H.S. Torr, P.K. Dokania, Multi-agent diverse generative adversarial networks, in: Proceeding of Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 8513–8521.

[31] 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, Proceeding of Conference on Computer Vision and Pattern Recognition, CVPR, 2017.

[32] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196, 2017.

[33] S. Nowozin, B. Cseke, R. Tomioka, f-gan: training generative neural samplers using variational divergence minimization, in: Proceeding of Conference on Neural Information Processing Systems, NIPS, 2016, pp. 271–279.

[34] M. Wang, H. Li, F. Li, Generative Adversarial Network based on Resnet for Conditional Image Restoration. arXiv preprint arXiv:1707.04881, 2017.

[35] M.S.M. Sajjadi, B. Schölkopf, M. Hirsch, Enhancenet: single image super-resolution through automated texture synthesis, in: Proceedings of IEEE International Conference on Computer Vision, ICCV, 2017, pp. 4501–4510.

[36] 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 Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 105–114.

[37] B. Wu, H. Duan, Z. Liu, G. Sun, Srpgan: Perceptual generative adversarial network for single image super resolution. arXiv preprint arXiv:1712.05927, 2017.

[38] Y. Tai, J. Yang, X. Liu, Image super-resolution via deep recursive residual network, in: Proceeding of Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 2790–2798.

[39] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, C. Schroers, A fully progressive approach to single-image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 864–873.

[40] X. Wang, K. Yu, C. Dong, C. Change Loy, Recovering realistic texture in image super-resolution by deep spatial feature transform, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 606–615.

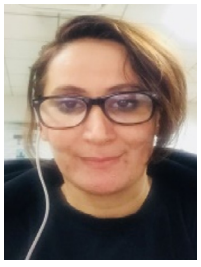
[41] J. Kim, J.K. Lee, K.M. Lee, Accurate image super-resolution using very deep convolutional networks, in: Proceeding of Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 1646–1654.

[42] J.H. Krijthe, M. Loog, Implicitly constrained semi-supervised least squares, in: Proceedings of the International Symposium on Intelligent Data Analysis, 2015, pp. 158–169.

[43] D. Ulyanov, A. Vedaldi, V. Lempitsky, Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022, 2016.

[44] S.O. Mehdi Mirza, Conditional generative adversarial nets, in: Proceedings of Deep Learning Workshop NIPS, 2014.

Pourya Shamsolmoali, Received PhD degree in computer science and graduated from Jamia Hamdard University, India and Shanghai Jiao Tong University, China, from 2016 to 2017 he was Associate researcher at the Advanced Scientific Computing Division in Euro-Mediterranean Center on Climate Change Foundation, Italy. Currently he is a researcher at Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University. In 2018 he selected as a young talented scientist by China ministry of education. His research activities focus on Machine learning, Image Processing, Computer Vision and Deep Learning.



Masoumeh Zareapoor, received Ph.D in computer science from Jamia Hamdard University, New Delhi, India in 2015. Currently, she is working as associate researcher in Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University. Prior to that, she was associate researcher in Tokyo University of technology. Her research activities focus on Computer Vision, Image Processing, and Machine Learning.

Ruili Wang received the Ph.D. degree in computer science from Dublin City University, Dublin, Ireland. He is currently a Professor of Artificial Intelligence with the School of Natural and Computational Sciences, Massey University, Auckland, New Zealand, and the Director of the Centre of Language and Speech Processing. His research interests include speed processing, language processing, image processing, data mining, and intelligent systems. Dr. Wang is an Associate Editor and an Editorial Board member for international journals, such as Knowledge and Information Systems, Applied Soft Computing, etc. He was the recipient of the Marsden Fund, one of the most prestigious research grants in New Zealand.

Deepak Kumar Jain, received PhD. from National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China. His research interests include computer vision, artificial Intelligence and face recognition.

Jie Yang, received his Ph.D. from the Department of Computer Science, Hamburg University, Germany, in 1994. Currently, he is a professor at the Institute of Image Processing and Pattern recognition, Shanghai Jiao Tong University, China. He has led many research projects (e.g., National Science Foundation, 863 National High Tech. Plan), had one book published in Germany, and authored more than 200 journal papers. His major research interests are object detection and recognition, data fusion and data mining, and medical image processing.