Optimizing Image Super-Resolution with Deep Gradient Guidance and Enhanced Multiscale Blocks

Wilson Martinez, Brown Williams

Lehigh University,Lehigh University

wilson00@gmail.com,bwbw1239@gmail.com

Abstract: To address the issue of poor visual quality and structural distortion in existing image super-resolution reconstruction models, a new approach utilizing a deep gradient guidance generative adversarial network is introduced. This model incorporates a gradient branch within the generator, which transfers gradient image features and integrates this gradient information with the image branch to maintain edge integrity. Inspired by MSRB, ResNext, and Inception, an enhanced multi-scale residual block is developed and integrated into the core modules of both the image and gradient branches, facilitating the capture of multi-scale information. The discriminator is enhanced with WGAN-GP to bolster the stability of network training. When compared to perception-driven algorithms like SRGAN, ESRGAN, and NatSR, this new approach more effectively prevents structural distortions and elevates the quality of the generated images. The computational complexity of this model is 23.7GFLOPs, significantly lower than that of ESRGAN and SPSR by approximately 1/4 and 1/10, respectively.

Keywords: Image super-resolution; Structural distortion; WGAN-GP; Gradient guidance; Adversarial training; Multi-scale residual block.

1. Introduction

Single Image Super-Resolution (SISR) is a hot research topic in the field of computer vision. Its purpose is to generate high resolution (HR) images from single low resolution (LR) images, improve the visual perception quality of images and provide richer image information. It is a classic low-level vision problem [1]. Image super-resolution reconstruction is widely used in video surveillance, remote sensing imaging, medical image analysis and other fields [2].

With the development of deep learning, image super-resolution reconstruction based on deep learning has become the mainstream method. Dong et al [3]. proposed image super resolution convolutional neural network (SRCNN), which learned the mapping relationship between high- and low-resolution images through convolutional neural network and achieved good reconstruction results. Kim et al[4].used residual learning to build a very deep convolutional neural network for image super-resolution(VDSR). Ledig et al[5]. proposed the image super resolution generative adversarial network (SRGAN) and optimized the model via perceptual loss [6]. Lim et al [7]. fully considered the influence of BN layer in image super-resolution algorithms and proposed an enhanced deep residual network for image super-resolution (EDSR), which removed BN layer in residual block.

Although the existing image super-resolution reconstruction algorithms based on convolutional neural network and generative adversarial network (GAN) have improved in image perception quality and objective index, some algorithms, such as ESRGAN [8], SRGAN, cannot recover high-frequency details well, and geometric distortion will occur in the reconstruction process. Ma et al [9]. proposed structure-preserving super-resolution (SPSR), which used gradient image to protect edge detail. SPSR used RRDB as basic module in image branch and gradient branch, which made model have large parameters and floating-point operations per second (FLOPs). In this paper, we proposed a deep gradient guidence network based on generative adversarial network named DGGGAN and proposed an enhanced multi-scale residual block as a basic block. The results of experiments show that our DGGGAN can obtain better performance and lower parameters and FLOPs.

2. Methord

In this part, we first introduce the overall architecture of our DGGGAN model, then we show the details of multi-scale residual block, structure of the discriminator and the objective function.

2.1. Structure of the generator

As show in figure1, our DGGGAN consists of imagebranch and gradient branch. The structure of the image branchis same as the existing image super-resolution reconstructionmodel based on deep learning. The shallow feature is extracted by one or more convolutional layers and the deep feature is drawn by multiple basic modules. The deep featureis sent to the up-sample module to amplify, the final output result is obtained through a convolution layer with 3 output channels. The composition of gradient branch is exactly same as the image branch. In addition, in order to enhance the information exchange between image branch and gradient branch, after several basic modules in the image branch, the gradient image is calculated and transferred into the gradientbranch to fuse with the features transferred in the gradient branch, different from SPSR, we only use basic modules to extract and transmit feature map of gradient images instead ofbasic module and convolution.

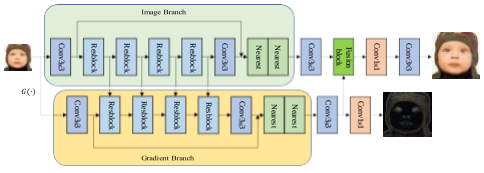


Fig.1 Overall framework of the DGGGAN generator

In figure1, the gradient images is obtained via $A=[1,0,-1]$,which is a convolutional layer with 1×3 convolutional kernal, A^T is a transpose of the A. Input a LR image S, we can calculate the gradient image as follows:

$$\nabla S_x = A * S$$

$$\nabla S_y = A^T * S \tag{1}$$

Where ∇S_x and ∇S_y are gradient of x and y of the input image S ,$*$ is the convolutional calculate. Direction of gradient is not calculated because the large of the gradient is already represent the sharpness of image edges.

2.2. Multi-scale residual block

As show in figure2, Combine the advantages of ResNext and Inception, we propose an enhanced multi-scale residual block. Firstly, the double branches of the MSRB is extended to four branches, these branches are mainly composed of 3×3 and 5×5 convolution. Then we use 1×1 convolution to improve non-linear expression ability. Plus we use LeakyReLU to solve the problem of neuron death except 1×1 convolution. Last, we use SEblock to improve the dependency between channels.

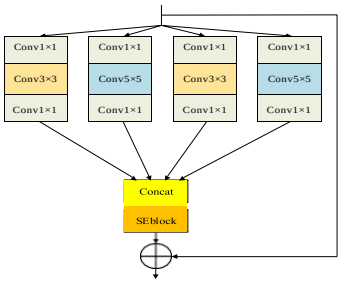


Fig.2 Multi-scale residual block

The input and the output of multi-scale residual block is denote x_{i-1} and x_i, C is feature after fusion of four branches, multi-scale residual feature can be obtained as follows:

$$x_i = x_{i-1} + SEblock(C) \tag{2}$$

2.3. Squeeze-and-excitation block

The core idea of SEblock[10] is to improve the interdependence between channels and adaptively correct the characteristic response strength between channels by using global loss. SEblock mainly includes squeeze and excitation.

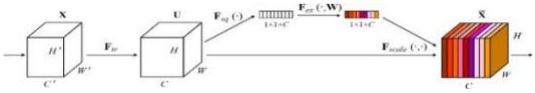


Fig.3 Squeeze-and-excitation block

In figure3, input X is a $H'\times W'\times C'$ feature map, use squeeze for feature map X, firstly, get a $H\times W\times C$ feature map U by using convolution F_{tr}, then, squeeze the feature map U by Global average pooling layer:

$$z_c = F_{sq}(U_c) = \frac{1}{W\ H}\sum_{i}\sum_{j} u_c(i,j) \tag{3}$$

Where σ is Sigmoid, δ is ReLU, W_1 and W_2 are weight matrix of two fully connection layers. Once the gate mechanism is obtained, the output $\tilde{X}$is:

$$\tilde{x}_c = F_{scale}(u_c, s_c) = s_c \cdot u_c \tag{5}$$

2.4. Structure of the discriminator

As show in figure4, our discriminator is roughly similar to VGG, but the convolutional kernel is 4 × 4 with stride 2, which is used for down sampling. Same as WGAN [11], the last layer of the discriminator does not use Sigmoid. Moreover, the discriminator uses gradient penalty [12] to independently punish the gradient of the discriminator for each input, the BN layer will modify the gradient with the batch processing information, so the BN layer is abandoned in the discriminator in our model.

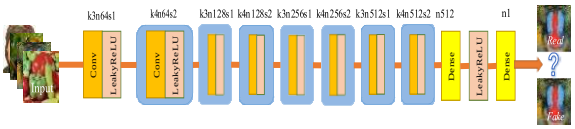


Fig.4 Architecture of discriminator

2.5. Objective function

Existing image super-resolution algorithms based on CNN often use L1 or L2 loss optimization. These methods can obtain better PSNR and SSIM, but the generated images are too smooth. The image super-resolution reconstruction algorithms based on generative adversarial network often uses perceptual loss to optimize model, the generated image contains richer high-frequency details, but the generated image is prone to structural distortion, resulting in poor visual quality.

For the above questions, we introduce gradient loss in our loss function to ensure that the generated image does not have structural distortion. The gradient loss between HR gradient and SR gradient can be expressed as:

$$L_{grad} = \frac{1}{n}\sum_{i=1}^{n}(|G(I_{HR}) - G(I_{SR})|_1^1) \tag{6}$$

Where $G(.)$ is calculate gradient image, we calculate L1loss between HR gradient and SR gradient.

The total loss function can be expressed as:

$$L = \theta L^{percept} + \alpha L^{IB}_{pixel} + \mu L^{IB}_{grad} + \beta L^{IB}_{adv} + \varepsilon L^{IB}_{adv_{grad}} + \eta L^{GB}_{grad} \tag{7}$$

Where $L_{percept}$ is perceptual loss, we use VGG19 as feature extractor, then calculate feature loss between HR image and SR image, $\varphi(.)$ is denote VGG network, the feature loss can be obtained as :

$$L^{percept} = \frac{1}{n}\sum_{i=1}^{n}(|\varphi(I^{HR}) - \varphi(I^{SR})|_1^1) \tag{8}$$

L^{pixel}_{IB} is pixel loss of image branch, we use L1loss to express pixel loss:

$$L^{IB}_{pixel} = \frac{1}{n}\sum_{i=1}^{n}(|I^{HR} - I^{SR}|_1^1) \tag{9}$$

$$W \times H \quad i=1 \quad j=1$$

The excitation operation is obtained through the squeeze feature z and learn the feature weight of each channel. The learned c features should be able to stimulate important features and suppress useless features. Therefore, a gate mechanism is constituted by using two fully connection layers:

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)) \quad (4)$$

L^{grad}_{IB} and L^{grad}_{GB} are gradient loss of image branch and gradient branch. L^{adv}_{IB} and $L^{adv_{grad}}_{IB}$ are adversarial loss of image branch, we use WGAN-GP to express the adversarial loss:

$$L = E_{x \sim p_r}[D_\omega(x)] - E_{x \sim p_g}[D_\omega(x)] + \lambda E_{\hat{x} \sim p_{\hat{x}}}[(\|\nabla_x D(\hat{x})\| - 1)^2] \quad (10)$$

Where p_r and p_g are real sample distribution and fake

sample distribution, $p_{\hat{x}}$ is distribution after sampling between the real sample and the generated sample.

3. Experiments

3.1. Datasets

We choose DIV2K as train dataset, which has 800 high resolution images. Firstly, we crop 800 HR images to 480×480 sub-images by using stride 24, after croping, there are 32208 sub-images in the DIV2K train dataset. Using random rotation and random crop to enhance train dataset. The input LR image is set to 48×48, therefore, the output SR image is 192×192.Set5, Set14, BSD100 and Urban100 test dataset are selected to test performance of the generator.

3.2. Experiments details

We set θ=1.0, α=0.01, μ=0.01, β=0.005, ε=0.005, η=0.5 in eq.7 and λ=10 in eq.10.The optimizer of Generator and discriminator select Adam with β_1 =0.9, β_2 =0.999, initial learning rate is set to 1 × 10-4 and descend a half at [50K,100K,200K,300K],set batchsize=16, train GAN model total 500K iters.There are 10 multi-scale residual blocks in image branch and calculate gradient image every 3 multi-scale blocks, the upsample factory is set to 4 × .We use PyTorch to design all experiments on NVIDIA RTX 3080 GPUs.

For model performance, we choose PSNR and SSIM as objective indicators. Before calculate, every RGB images should be converted to YCbCr space and calculate PSNR and SSIM only on Y channel. Plus, PI and LPIPS are used to evaluate perceptual quality. Lower PI and LPIPS value mean the model has better perceptual quality.

3.3. Experiment results

Validation of gradient branch effectiveness: In order to prove the effect of gradient branch, we remove the gradient branch and only retain the image branch, meanwhile, we set μ=ε=η=0 in eq.10, we calculate PI in both cases. All results of this ablation experiment are shown in table1:

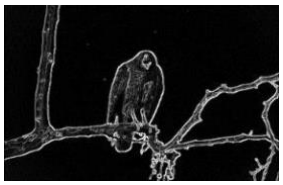
Table 1. Gradient branch validity verification (PI)

Gradient branch	Set5	Set14	BSD100	Urban100
×	3.5716	3.0795	2.7136	3.9815
√	3.2559	2.7565	2.3508	3.5486

The results of table 1 show that the performance of the model without gradient branch is obviously lower than the model using image branch and gradient branch. The PI value on four test datasets are decline 0.3157,0.3230,0.3628, 0.4329. This experiment proves the model can obtain better performance when the model contains gradient branch.

In figure5, we choose 42049.png in BSD100 dataset and show the visualize results of the output of gradient branch. The greater change of gray value, the greater change of gradient value, sharpened edge details can be observed in gradient images. This prior information is more conducive to restore high-frequency details and guide the reconstruction of edge details.

(a) HR image



(b) HR gradient

(c) SR gradient

Fig.5 Visualize of the output of gradient branch

Validation of SEblock: In order to verify the effectiveness of SEblock in the multi-scale residual module, remove the SEblock in the multi-scale residual unit, and compare the performance of the generator with and without SEblock in the multi-scale residual module on the Set5 dataset. The results are shown in table2:

Table 2. Results of validation of the SEblock

SEblock	PI	FLOPs
×	3.3427	23.6G
√	3.2559	23.7G

In table2, when the multi-scale residual cell in the image branch and gradient branch does not contain SEblock, the PI value is 3.3427. The PI value is 3.2559 when the image branch and gradient branch contains the SEblock. However, the FLOPs of the generator only increase 0.1GFLOPs after use SEblock in the multi-scale residual module, it means that the SEblock can improve the performance of the generated network without significantly increasing the computational complexity of the model.

Compare with other algorithm:We choose some PSNR-dirven algorithms EDSR, RCAN, RDN and DBPN, some perceptual-deiven algorithms SRGAN, ESRGAN, NatSR and SPSR. All results are shown in table3 and table4:

Table 3. Objective index of different algorithms on test detasets (PSNR/SSIM)

	Set5	Set14	BSD100	Urban100
EDSR	32.46/0.8966	28.79/0.7874	27.70/0.7417	26.64/0.8031
RCAN	32.63/0.9002	28.87/0.7889	27.77/0.7435	26.81/0.8088
RDN	32.46/0.8990	28.81/0.7871	27.72/0.7419	26.61/0.8028
DBPN	32.47/0.8980	28.82/0.7859	27.73/0.7401	26.37/0.7944
ESRGAN	30.45/0.8677	26.28/0.7783	25.31/0.6506	24.36/0.7017
SRGAN	29.16/0.8613	26.17/0.7841	25.46/0.6485	24.40/0.6988
NatSR	30.99/0.8800	27.51/0.8140	26.45/0.6831	25.46/0.7432
SPSR	30.40/0.8627	26.64/0.7930	25.51/0.6576	24.80/0.7237
DGGGAN	30.48/0.8601	26.69/0.7884	25.49/0.6573	24.72/0.7093

In table3, we can see that the PSNR-driven algorithms obtain the highest PSNR and SSIM value on four test datasets, the PSNR value of perceptual-driven algorithms ESRGAN,SRGAN,NatSR,SPSR and DGGGAN is lower

than PSNR-driven algorithms EDSR, RCAN, RDN and DBPN. SRGAN obtain the lowest PSNR value on Set5 and Set14 datasets and slightly higher than ESRGAN on BSD100 and Urban100 datasets, because the loss function of SRGAN is only determined by perceptual loss and adversarial loss, optimize this function will not reduce MSE value. The PSNR and SSIM value of DGGGAN on the four test sets are close to SPSR.

The perceptual index of different algorithms are shown in table4. The PI and LPIPS of all perceptual-driven algorithms are lower than PSNR-driven algorithms, it means that these algorithms can generate images with better visual quality. Our DGGGAN obtain the best PI value on four test datasets. For LPIPS, our DGGGAN is generally close of SPSR.

Table 4. Perceptual index of different algorithms on test datasets (PI/LPIPS)

	Set5	Set14	BSD100	Urban100
EDSR	5.9819/0.2088	5.2594/0.2963	5.2625/0.3249	4.9844/0.2923
RCAN	6.3749/0.2158	5.7127/0.3106	5.7588/0.3317	5.4181/0.2944
RDN	6.0092/0.2134	5.4633/0.3039	5.5412/0.3299	5.2502/0.3162
DBPN	6.1324/0.2108	5.4596/0.2985	5.4915/0.3250	5.1360/0.2748
ESRGAN	3.7522/0.0748	2.9261/0.1329	2.4793/0.1614	3.7704/0.1229
SRGAN	3.9820/0.0882	3.0851/0.1663	2.5459/0.1980	3.6980/0.1551
NatSR	4.1648/0.0939	3.1094/0.1758	2.7801/0.2114	3.6523/0.1500
SPSR	3.2743/0.0644	2.9036/0.1318	2.3510/0.1611	3.5511/0.1184
DGGGAN	3.2559/0.0641	2.7565/0.1355	2.3508/0.1645	3.5486/0.1107

Table 5. Params and FLOPs of different perceptual-driven algorithms

	SFTGAN	NatSR	ESRGAN	SRGAN	SPSR	DGGGAN
Params	1.8M	5.0M	16.7M	0.7M	24.8M	14.6M
FLOPs	0.8G	12.5G	89.7G	7.0G	265.1G	23.7G

Table5 shows the number of parameters and the FLOPs of some perceptual-driven algorithms. SPSR has 24.8M parameters and 265.1GFLOPs, our DGGGAN only has 14.6M parameters and 23. 7GFLOPs.Compare with the ESRGAN and SPSR, the FLOPs of DGGGAN is decline 1/4 and 1/10. It shows that although the number of parameters and the FLOPs is lower than SPSR, the performance of our DGGGAN is close to SPSR.

Figure6 choose some visualize results of perceptual-driven algorithms. Perceptual-driven algorithms always optimize by using generative adversarial network and perceptual loss, which can generate high perceptual quality images. Our DGGGAN use multi-scale residual blocks to catch the multi-scale feature, make the model produce richer details. For edge structure protection, SPSR and DGGGAN use gradient images to guide reconstruction, so, images generated by SPSR and DGGGAN do not produce structural distortion. Although SFTGAN use semantic segmentation to preserve edge details, the performance of SFTGAN is greatly affected by the performance of semantic segmentation model, which cannot guide reconstruction well.

Fig.6 visualize results of perceptual-driven algorithms, the first row is baby.png from Set5, the second row is img005.png from Urban100 and the last row is img054.png from Urban100

4. Conclusion

In this paper, we proposed a deep gradient guidance network for image super-resolution based on generative adversarial network DGGGAN. Our DGGGAN algorithm can sufficiently use gradient information to prevented structural distortion and reconstructed high vision quality images.

In the future work, we will consider to use the algorithm into the video super-resolution and decline the parameters of the model, then use algorithm on the embedded platform.

Acknowledgements

This paper was financially supported by National Natural Science Foundation of China (No.61961037).

References

[1] Tang Yanqiu, Pan Hong, Zhu Yaping, et al. A survey of image super-resolution reconstruction[J].Joural of electronic, 2020, 48(7):1407-1419.

[2] Nan Fangzhe, Qian Yurong, Xing Yanni, et al.Survey of single image super-resolution based on deep learning[J]. Application Research of Computers, 2020, 37(2):321-326.

[3] Dong Chao, Loy C C, He Kaiming, et al. Learning a deep convolutional network for image super-resolution[C] // Proc of the European Conference on Computer Vision (ECCV). Germany: Springer, 2014:184-199.

[4] Kim J, Lee J K, Lee K M, et al. Accurate image super-resolution using very deep convolutional networks[C] // Proc of the Conference on Computer Vision and Pattern Recognition(CVPR). USA: IEEE，2016: 1646-1654.

[5] Ledig C, Theis L, Huszár F, et al. Photo -realistic single image super-resolution using a generative adversarial network[C] // Proc of the Conference on Computer Vision and Pattern Recognition (CVPR). USA: IEEE, 2017:105-114.

[6] Johnson J , Alahi A ,Fei-Fei L. Perceptual losses for real-time style transfer and super-resolution[J]. arXiv preprint arXiv:1603.08155, 2016.

[7] Lim B, Son S, Kim H, et al. Enhanced deep residual networks for single image super-resolution[C]//Proc of the Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). USA :IEEE, 2017:1132-1140.

[8] Wang Xintao, Yu Ke, Wu Shixiang, et al. ESRGAN:Enhanced super-resolution generative adversarial network[C]//Proc ofthe European Conference on Computer Vision(ECCV). Germany: Springer, 2018:63-79.

[9] Ma Cheng, Rao Yongming, Cheng Yean, et al. Structure-preserving super-resolution with gradient guidance[C]//Proc of the Conference on Computer Vision and Pattern Recognition (CVPR). USA :IEEE, 2020:7769-7778.

[10] Hu J, Shen L, Albanie S, et al. Squeeze-and-excitation networks[J]. arXiv preprint arXiv:1709.01507, 2017.

[11] Martin A, Soumith C, Léon B. Wasserstein GAN[J]. arXiv preprint arXiv:1701.07875, 2017.

[12] Ishaan G, Faruk A, Martin A, et al. Improved training of Wasserstein GANs[J]. arXiv preprint arXiv:1704.00028v3, 2017.