Single Image Super Resolution Based on Generative Adversarial Networks

Kai Li[a], Ye Liang[b], Shenghao Yang[a], Jinfang Jia[a], Jianqiang Huang[a,*], Xiaoying Wang[a]
[a]State Key Laboratory of Plateau Ecology and Agriculture,Department of Computer Technology and Application,Qinghai University, Xining, Qinghai; 810016, China; [b]Rocket Force University of Engineering, Basic Course Dep, Foreign language Office,Xi'an,Shanxi; 710025, China

ABSTRACT

Deep neural networks based on SRGAN single image super-resolution reconstruction can generate more realistic images than CNN-based super-resolution deep neural networks. However, when the network is deeper and more complex, unpleasant artifacts can result. Through a lot of experiments, we can use the ESRGAN model to avoid such problems. When using the ESRGAN model for super-resolution reconstruction, the perceived index of the resulting results does not reach a lower value. There are two reasons for this: (1)ESRGAN does not expand the feature maping. ESRGAN uses 128*128 to obtain the feature information of the image by default, and can't get more image information better. (2) ESRGAN did not re-optimize the generated image. Therefore, we propose ESRGAN-Pro to optimize ESRGAN for the above two aspects, combined with a large amount of training data, and get a better perception index and texture.

Keywords: Generative Adversarial Networks, super resolution, deep learning framework, image processing

1. INTRODUCTION

Super-resolution reconstruction(SR) refers to the technique of recovering high-resolution (HR) images from a low-resolution (LR) image or image sequence. Single Image Super Resolution Reconstruction (SISR) refers to the technique of generating high resolution images from a single low resolution image. Due to its flexibility, simplicity and practicality, it has received wide attention. It is widely used in hyperspectral imaging, medical imaging and so on.

Traditional single-image super-resolution reconstruction methods include interpolation-based methods, reconstruction-based methods, and learning-based methods that either utilize internal similarities of the same image or learn mapping of external low-resolution and high-resolution sample pairs function. Due to the breakthrough progress in deep learning in other computer vision fields, people try to introduce deep neural networks to solve the problem of image super-resolution reconstruction by constructing deep-level network for end-to-end training.

At present, the commonly used deep learning model can be divided into the model based on interpolation preprocessing, the model based on original image processing, the model based on hierarchical feature and the model based on high frequency detail according to the input information of the image, the training process and feature extraction, and the application of high frequency details, etc.

2. METHOD

2.1 ESRGAN[1]

First, GAN is used as the basic framework for ESRGAN. The use of GAN for super-resolution is to address the shortcomings of the lack of high-frequency information and detail in the results of traditional methods, including deep learning methods. Although the psnr and ssim images of GAN are small, the added details may differ from the actual details. The texture is relatively simple, but it always has details, the visual feel is better, and the details are rich. This is the advantage of using GAN to solve super-resolution reconstruction.

Second, ESRGAN converts the basic remaining units of SRGAN into residual dense blocks (RRDBs). The residuals are introduced into the network architecture to learn the residuals between the high definition image and the LR image. Using RRDB at the same time can have a deeper neural network, further improving the texture of the restoration, because the depth model can capture more image features. In addition, we have found that deeper models can reduce

Eleventh International Conference on Digital Image Processing (ICDIP 2019) edited by Jenq-Neng Hwang, Xudong Jiang, Proc. of SPIE Vol. 11179, 111790T © 2019 SPIE · CCC code: 0277-786X/19/$21 · doi: 10.1117/12.2539692

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 04 Jun 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

unpleasant noise.Figure 1 is a schematic diagram of the RRDB model, and Figure 2 is a schematic diagram of a dense block.

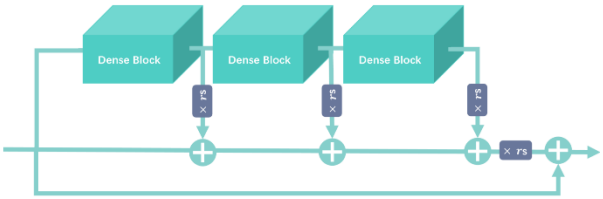

Figure 1. Diagram of the RRDB model.

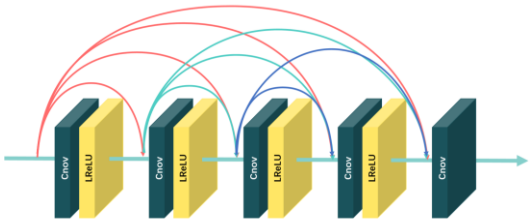

Figure 2. Schematic diagram of the Dense Block model.

In addition, compared to SRGAN[2], ESRGAN has also made some innovations in the GAN model. Change the original GAN framework to RaGAN (Relativistic Average GAN). RaGAN is used because the discriminator should also reduce the probability that "actual data is true" and increase the probability of "pseudo-data is true" to determine whether the image is more realistic than the other, rather than determining whether the image is true or false. At the same time, using RaGAN helps to learn sharper edges and finer textures.

ESRGAN removes the Batch Normalization[3] layer from SRGAN to achieve consistent performance without artifacts. It does not degrade performance, but it saves computing resources and memory usage. Furthermore, models with BN layers are more likely to introduce unpleasant artifacts when the network is deeper and more complex. Figure 3 shows that SRGAN produces artifacts that affect image quality in deeper neural networks. After removing the Batch Normalization layer, ESRGAN uses the residuals multiplied by the number in the range [0,1] to reduce the residuals and prevent unstable networks from appearing. At the same time, smaller initialization is used, and it is easier to train the residual structure when the variance of the initialization parameters becomes smaller.

Figure 3. Artifacts generated by the SRGAN deep neural network.

In the ESRGAN generation network part, SRResNet[2] is used as the basic network architecture. In order to transfer most of the calculations to the feature space of the LR image and reduce the amount of calculation. As shown in Figure 4. Each residual block contains two 3x3 convolutional layers. After the convolutional layer, PReLU acts as an activation function, and two 2x sub-pixel convolution layers are used to increase the feature size. Parametric leaky ReLU (PReLU) works better on larger data sets but has overfitting risk on smaller data sets. It also prevents the problem of gradient disappearing.

Using the mean square error to optimize SRResNet (the generated network part of SRGAN), it is possible to obtain a result with a high peak signal to noise ratio. Calculate the perceptual loss on the high-level features of the trained VGG model to optimize ESRGAN, and combine the ESRGAN discriminant network to obtain a higher PSNR. However, in order to get a better perceptual result. ESRGAN uses the perceived loss as part of generating network loss, so that the processed picture result can meet the requirements of the perceptual index and achieve the optimal perceptual result.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 04 Jun 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

Equation 1 is the loss function of the generated network, L_{PI} is the perceptual loss function, and L_1 is the pixel-by-pixel loss function.

$$L_G = L_{PI} + \lambda L_G^{Ra} + \eta L_1 \quad (1)$$

In addition, sub-pixel convolution layers are also used for upsampling in the generation network. In the discriminant network part, there are 8 convolutional layers. As the number of network layers deepens, the number of features increases continuously, and the feature size decreases. The activation function is selected as LeakyReLU, and finally obtained by two fully connected layers and the final sigmoid activation function. The probability of being predicted as a natural image.

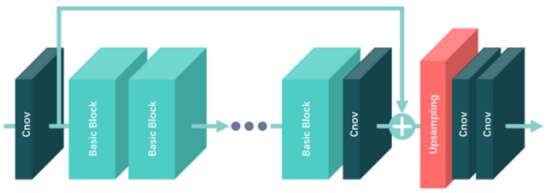

Figure 4. SRResNet Basic Structure.

3. OPTIMIZING DIRECTION

Computer vision related research is basically related to images, so the optimization direction should be optimized and adjusted around the relevant parameters of the image. In addition, optimization of the structure of the neural network is also necessary.Next, we will introduce the optimization direction for ESRGAN.

3.1 Data set

In the dataset, we will mainly explain the problem of dataset partitioning. This is because when we design the neural network, we need to train and test the neural network, and this is based on the existing dataset. Therefore, the quality of the data set will be related to the degree of learning in the neural network. In general we only have a data set N containing n samples, as shown in Equation 2:

$$\mathrm{N} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\} \quad (2)$$

To use the dataset for training and testing, you need to partition the dataset to produce the required training set, validation set, and test set. The following describes the ESRGAN-Pro data set partitioning method Hold-out Method. Figure 5 is a schematic diagram of the Hold-out method. Generally, when the number of dataset samples is large, Hold-out can be used. However, when the number of dataset samples is small, the model evaluation results formed by this partitioning method may not be stable and accurate.

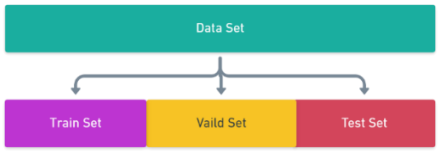

Figure 5. Hold-out Method.

Figure 6. Cross Validation Method.

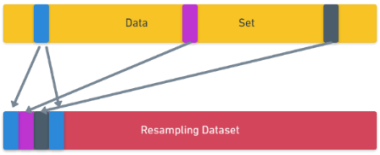

Figure 7. Bootstrap Resamplin Method.

Before deciding to use the Hold-out Method, both Cross Validation and Bootstrap Resampling have been considered, but the reason for using the Hold-out method is composed of two aspects. Figure 6 is a schematic diagram of the Cross Validation partitioning method, and Figure 7 is a schematic diagram of the Bootstrap Resampling partitioning method.

First, Cross Validation divides the data set N into mutually exclusive subsets of similar size. Therefore the result is relatively more accurate. However, there is a problem with Cross Validation, that is, when the data set is large, it requires a lot of computational power and is inefficient. If Bootstrap Resampling is used for data partitioning, the probability that a sample data set m is not in N' is as shown in Equation 3.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 04 Jun 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

$$\lim_{m\to\infty}\left(1+\frac{1}{m}\right)^{m}=\frac{1}{e}\approx 0.368 \tag{3}$$

The Bootstrap Resampling method is useful when the data set is small and it is difficult to effectively partition the data set. However, the data set generated by the Bootstrap Resampling method changes the distribution of the initial data set, resulting in incorrect training results.

Therefore, ESRGAN-Pro uses the Hold-out method for dataset partitioning. The total number of images in our training dataset is about 138,843 images. The larger dataset size does not apply to Cross Validation and Bootstrap Resampling.

3.2 HR patch-size

We have found through extensive experiments that using larger patch-sizes can achieve better image quality in training deep neural networks, because the expanded image sensing range can help deep neural networks capture more useful image information. On the basis of 10 and 20 RRDB models, we changed the HR patch-size to 128*128 and 192*192 respectively, and used the same training set (urban100) for training. The result is shown in Figure 8,9. Two different depths of neural networks were observed to achieve better image quality at larger patch-sizes.

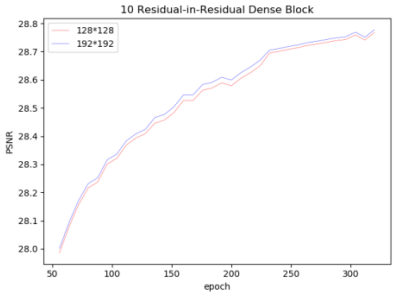

Figure 8. The number of RRDBs is 10 different patch-size renderings.

Figure 9. The number of RRDBs is 23 different patch-size renderings.

3.3 Loss Function

The loss function is the basis for judging the prediction of the neural network. The following mainly explains the $L_1$and L_2 functions, and the Mix loss function used in ESRGAN-Pro.

The L_1 and L_2 loss functions and peak signal-to-noise ratio (PSNR) are independent of human perception of image quality. Using the L_1 and L_2 loss functions alone does not capture the perception of images by the human visual system. Therefore, in order to get better image quality than human perception, we combine L_1 and MS-SSIM to get the Mix loss function.

In order to combine the image details under different resolutions and observation conditions into the quality evaluation algorithm in the multi-scale method, the MS-SSIM method (Multi-Scale Structural SIMilarity Index, MS-SSIM)[4] is proposed. Equation 4 is the calculation. MS-SSIM loss function.

$$\mathcal{L}^{MS-SSIM}(P)=1-MS-SSIM(\tilde{p}) \tag{4}$$

MS-SSIM and SSIM are not particularly sensitive to uniform deviations. This can result in brightness changes or color shifts, which usually become more dim. MS-SSIM retains the contrast of the high frequency region better than other loss functions of the experiment. On the other hand, L_1 preserves the weighting error of color and brightness regardless of the local structure, but does not produce exactly the same contrast as MS-SSIM. To capture the best features of the two error functions, combine them. The loss function Mix[5] obtained by mixing MS-SSIM and L_1 proposed by Hang Zhao et al. Equation 5 is the Mix loss function.

$$\mathcal{L}^{Mix}=\alpha\cdot\mathcal{L}^{MS-SSIM}+(1-\alpha)\cdot G_{\sigma_G^M}\cdot\mathcal{L}^{l_1} \tag{5}$$

The definition of this mixed loss function is very simple, basically the weighted sum of MS-SSIM and L_1 but because the G-Gaussian distribution parameter is needed for the MS-SSIM backpropagation error, it is also multiplied in the L_1 part. The distribution parameters only.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 04 Jun 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

Through experiments, α was set at 0.84, and the contribution of the two-part loss was approximately equal. We use EDSR as the basic model, using the Nvidia recurring Mix method and Pytorch's own loss function and test set Urban100[6] for testing. PSNR, SSIM and MS-SSIM are used as evaluation indicators. The results are shown in Table 1 below. From the experimental results, it is better to use Mix as a loss function to obtain better image quality.

Table 1. Results of different loss functions on EDSR.

Result	L1	L2	Mix
PSNR	28.6281	27.9721	28.8834
SSIM	0.8821	0.8645	0.8992
MS-SSIM	0.9521	0.9511	0.9530

3.4 Distributed computing

We used Pytorch's[7] distributed computing in this competition to divide small batch samples into more small, and run each smaller small batch of samples in parallel. We ran ESRGAN with adjusted parameters and structure on a single card and did not join the Pytorch parallelization module. Then we do the data parallel operation by using torch.nn.DataParallel in Pytorch, which can be parallelized on multiple GPUs in the batch dimension. But the current recommendation from Pytorch is a multi-process single GPU, implemented using torch.nn.DistributedDataParallel. We tried single-process single GPU, multi-process single GPU, multi-process multi-GPU during this competition (currently, the cluster node only has two graphics cards). Table 2 below shows the time we used ESRGAN to calculate under the same data size.

Table 2. Training time.

Result	Single-process single-GPU	Multi-process single GPU	Multi-process multi-GPU
Time(h）	300.96	195.6	125.04

We can see from the training time that using distributed computing, there will be better performance under multi-process multi-GPU, because in the case of multi-process, you can make full use of the computing performance of each GPU to achieve the fastest Training time. At the same time, we also consider whether the training time will be further shortened in the case of multi-process multi-GPU between multiple nodes. However, since our cluster does not have too many GPUs, it does not verify the proposed multi-level multi-GPU.

4. EXPERIMENT

In this section, we will illustrate the data set and hyperparameters of our model training. Then Compare optimized ESRGAN (ESRGAN-Pro) with several other models using multiple benchmark data sets.

4.1 Data Set

First, we used the DIV2K[8] and Flickr[9] data sets for a total of 3,450 2K images in the training dataset selection. The DIV2K dataset is primarily a high-resolution image dataset for image super-resolution, and the Flickr dataset is 2,650 high-resolution images collected through the Flickr official website. Enrich our training set by merging two data sets. In addition, we can use previous experimental experience to find that a data set with a large number of rich textures helps generative network to generate images that are more in line with the human visual system. At the same time, in order to get more data sets, we used the method of randomly flipping, rotating 90°, 180°, 270° and splitting the images.Since our original training set is 2K resolution pictures, we can learn more repeated textures and detailed part of the image information by dividing the pictures. After segmenting the image, our dataset scale was expanded from 3,450 images to 138,843 images. In addition, we use Set5[10], Set14[11], Urban100[6], and the data set provided in PIRM 2018[12] for verification.

4.2 Training details

The SR image is calculated under the scale factor of X4. Therefore, we use the matlab to downsample the HR image to obtain the LR image. The RRDB is set to 23. We have found through experiments that more RRDB obtain clearer texture, but more RRDB will take more time to complete the training. Therefore, 23 RRDB modules were selected through multiple experiments, which achieve the fastest training time without affecting the results. At the same time, when we pass the training data to the ESRGAN-Pro model, the batch-size is 16, and this parameter is the better hyperparameter adjusted through our training process. A larger HR patch-size can make the model learn more rich

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 04 Jun 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

image feature information, but it will also cause the training speed to be too slow, so we choose a relatively large HR patch-size of 192*192, and we Use multi-process multi-GPU to speed up training time.Then, pre-training models with pixel-by-pixel loss help GAN-based methods to obtain image results that are more consistent with the vision system. The reason is as follows:

(1) It can avoid the undesired local optimum generated by the generative network.

(2) After using the pre-training model, the discriminator receives a relatively good super-resolution image at the outset instead of a super-resolution image with artifacts, which helps it pay more attention to texture discrimination.

We used the Pytorch framework to implement our ESRGAN-Pro on this super-resolution competition, while using two NVIDIA Tesla V100 Tensor Cores for training.

4.3 Experimental results

We use the models EDSR, WDSR optimized for PSNR and the models SRGAN, ESRGAN, ESRGAN-Pro optimized for the vision system to perform super-resolution reconstruction of the baseline data set. In addition, since there is currently no standard for evaluating image quality, we choose the calculation indicators provided by the ASC official in this competition: Perceptual Index and RMSE. Figure 10 shows the results of different models on the same picture.

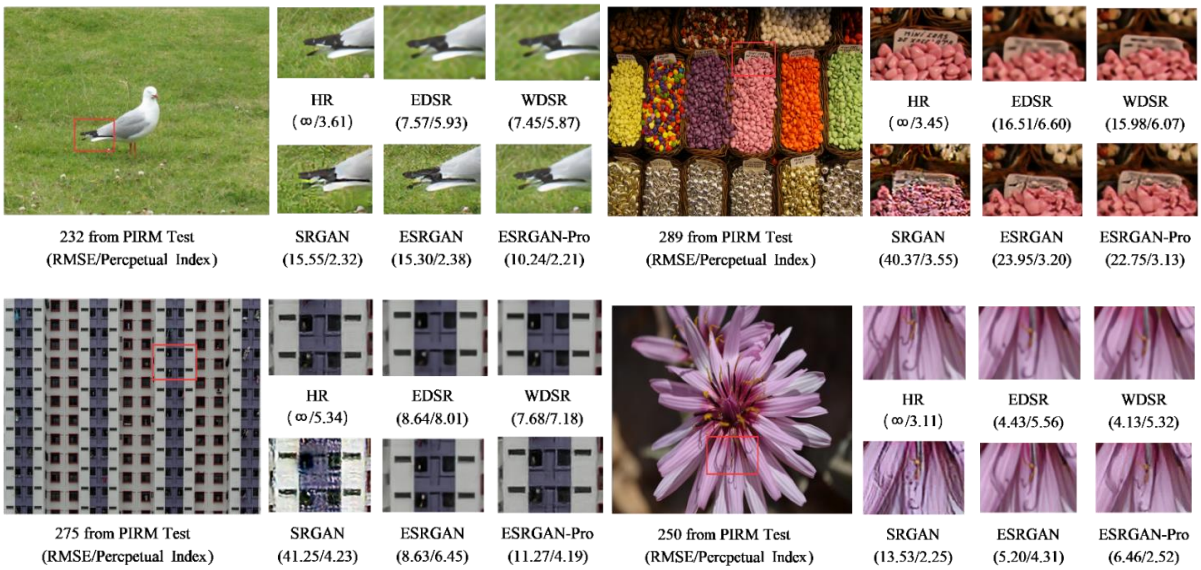

Figure 10. Results of ESRGAN-Pro. ESRGAN-Pro can produce more realistic textures and details, while SRGAN can get a very low perceptual index, but it will produce a lot of unreal artifacts.

We can see from Figure 10 that the ESRGAN-Pro optimized for ESRGAN will be better in texture and detail than ESRGAN and other models. From the different results obtained by image 232, we found that the results processed in ESRGAN and ESRGAN-Pro produced clearer textures and details on the results of the other models. The GAN-based super-resolution reconstruction model generates x4 super-resolution images often produces unnatural textures and noise that does not conform to the vision system. At the same time, the CNN-based models(EDSR, WDSR), because they are too focused on PSNR optimization, the resulting image details are often too smooth, lacking some realism of detail. In order to verify the effect of ESRGAN-Pro again, we verified it with different test sets, and the results are shown in Table 3.

Table 3. Results of different SR models.

Model	Training Set	Valid Set	Test Set	Perceptual Index	RMSE
ESRGAN	DIV2K	DIV2K	PIRM	2.1383	14.2212
ESRGAN-PRO	DF2K	DIV2K	PIRM	1.867	16.884
WDSR-A	DIV2K	Urban100+Set5	PIRM	3.8012	14.4879
RDN	DIV2K	Set5	Urban100	4.3211	14.974
EDSR	DIV2K	NONE	NONE	4.9035	10.7299

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 04 Jun 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

4.4 Post processing

Super-resolution images processed by the deep learning model can be post-processed to improve the perceptual index and RMSE. In this competition we used SRN-Deblur, Back-projection method for post-processing.

SRN-Deblur[13] mainly uses the encoder-decoder ResBlock network to amplify the advantages of CNN networks, and at the same time can obtain a larger sensing range, which is essential for the blurring of blurred images. So we use SRN-Deblur to post-process our ESRGAN-Pro, and the resulting image can indeed restore clearer and more natural details, as shown in Figure 11.

Figure 11. Results of post processing.

The Back-projection method is a method of recording the degree of fit of the pixels of a given image to the distribution of pixels in the histogram model. And we have improved the back projection combined with image super resolution. First, the result of the deep learning model is subjected to downsampling of X4 to obtain an image of the same resolution as the LR image. Then, pixel-by-pixel comparisons with LR images are used to recover details and textures that are not learned through the model. The original result is then reconstructed twice by upsampling the missing details and textures, resulting in a super-resolution image through back projection. Figure 12 shows the results obtained by back-projection.

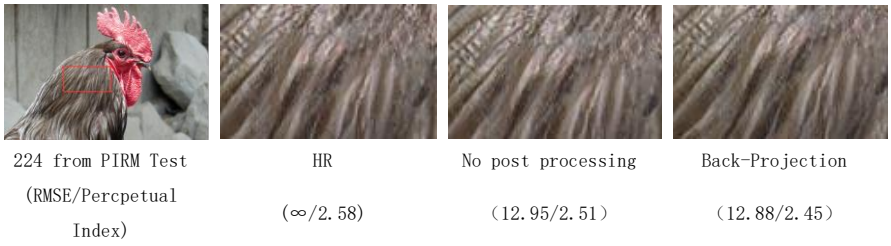

Figure 12. Results of Back-Projection Processing.

We have obtained a lot of experiments to conclude that the above results are consistent. Post-processing of super-resolution images with SRN-Deblur gives good detail and texture, but it also introduces more noise so that the results obtained by RMSE will be larger. Using the improved Back-Projection to post-process the results, we found that we can get more natural details without introducing more noise. Therefore, in the end we used Back-Projection as image post-processing operation in this competition.

5. SUMMARY

First of all, we have studied several excellent single-image super-resolution reconstruction models in recent years and compared their respective characteristics. And then we reproduced the training process of these models and tested them to calculate the corresponding scores. By comparison, we found that the ESRGAN model achieved the optimal perceptual index, and the RMSE value is legal. Through an in-depth study of the ESRGAN model, a series of optimizations have been carried out focusing on several aspects:

(1) In terms of data sets, the data set is divided by Hold-out method, which increases the size of the data set. The total number of data set used in the end is about 138,843 images.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 04 Jun 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

(2) Adjust patch-size. Experiments show that larger patch-size can achieve better image quality in training deep neural networks, because the extended image sensing range can help deep neural networks capture more useful images Information, we get a better image quality by increasing the patch-size to 192*192.

(3) Loss function, through research, compared with the traditional L1, L2 loss function, the Mix loss function obtained by mixing MS-SSIM and L1 has a better effect on practical applications.

(4) Distributed computing, through experiments, using distributed computing, there will be a better performance under multi-process multi-GPU, because in the case of multi-process, you can make full use of the computing performance of each GPU to achieve the most Fast training time.

Through the above optimizations, we proposed the ESRGAN-Pro model and conducted experiments. The results show that the ESRGAN-Pro model is better in texture and detail than ESRGAN and other models.

In addition, we use Back-Projection method to post-process the results to obtain better perceptual index and RMSE.

ACKNOWLEDGEMENT

The authors are grateful to the reviewers for valuable comments that have greatly improved the paper. This paper is partially supported by the national Natural Science Foundation of China (No.61762074), National Natural Science Foundation of Qinghai Province (No.2019-ZJ-7034) "Qinghai Province High-end Innovative Thousand Talents Program - Leading Talents" Project Support.

REFERENCES

[1] Wang X, Yu K, Wu S, et al. Esrgan: Enhanced super-resolution generative adversarial networks[C]//European Conference on Computer Vision. Springer, Cham, 2018: 63-79.
[2] Ledig C, Theis L, Huszár F, et al. Photo-realistic single image super-resolution using a generative adversarial network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4681-4690.
[3] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015.
[4] Rouse D M, Hemami S S. Analyzing the role of visual structure in the recognition of natural image content with multi-scale SSIM[C]//Human Vision and Electronic Imaging XIII. International Society for Optics and Photonics, 2008, 6806: 680615.
[5] Zhao H, Gallo O, Frosio I, et al. Loss functions for neural networks for image processing[J]. arXiv preprint arXiv:1511.08861, 2015.
[6] Huang J B, Singh A, Ahuja N. Single image super-resolution from transformed self-exemplars[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 5197-5206.
[7] Ketkar N. Introduction to pytorch[M]//Deep learning with python. Apress, Berkeley, CA, 2017: 195-208.
[8] Agustsson E, Timofte R. Ntire 2017 challenge on single image super-resolution: Dataset and study[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2017: 126-135.
[9] Timofte R, Agustsson E, Van Gool L, et al. Ntire 2017 challenge on single image super-resolution: Methods and results[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2017: 114-125.
[10] Bevilacqua M, Roumy A, Guillemot C, et al. Low-complexity single-image super-resolution based on nonnegative neighbor embedding[J]. 2012.
[11] Zeyde R, Elad M, Protter M. On single image scale-up using sparse-representations[C]//International conference on curves and surfaces. Springer, Berlin, Heidelberg, 2010: 711-730.
[12] Blau Y, Mechrez R, Timofte R, et al. The 2018 PIRM Challenge on Perceptual Image Super-Resolution[C]//European Conference on Computer Vision. Springer, Cham, 2018: 334-355.
[13] Tao X, Gao H, Shen X, et al. Scale-recurrent network for deep image deblurring[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 8174-8182.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 04 Jun 2022
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use