FINE-GRAINED DYNAMIC LOSS FOR ACCURATE SINGLE-IMAGE SUPER-RESOLUTION

Haoquan Wang, Gang Zhang

School of Microelectronics, Tianjin University, Tianjin, China

Zhichun Lei

University of Applied Science Ruhr West, Germany

ABSTRACT

With a wide range of applications, single-image super-resolution (SISR) is a very important computer vision task. There are many SISR strategies which are based on the CNNs improvement, such as residual connections, deeper networks, and attention mechanisms. Although such kind of strategies often improve the SISR performance, they increase CNNs' computational complexity and complex image texture areas still remain hard to be reconstructed. In particular, all the existing loss functions are minimized in the whole dynamic range, e.g. 0-255 in case of 8 bit image, which causes difficulty of CNNs' learning in texture image regions above all. Developing new loss function provides a promising SISR solution, i.e. one should be beyond the existing regression loss functions, which encounter problem in reconstructing the image texture details. For such goal, this paper proposes a dynamic fine-grained loss function. It consists of both an existing regression loss function and a new pixel classification loss together with a dynamic regression range loss. Extensive experiments conducted on the benchmark SISR show that the proposed method can achieve better results and without extra computational cost.

Index Terms— image super-resolution, fine-grained dynamic loss, classification loss

1. INTRODUCTION

Image super-resolution is a technique that utilizes low-resolution images or low-resolution image sequences to recover high-resolution images. The higher the resolution of an image, the sharper the image and the more information it can carry. However, due to the limitation of imaging equipment and the influence of external factors, it is not always possible to obtain high-resolution images, so the super-resolution reconstruction algorithm was born. It uses software to improve image resolution and is widely used in remote sensing and medical image processing. At present, the mainstream methods include interpolation-based [1], reconstruction based [2] and learning-based methods [3].

In recent years, the convolutional neural networks (CNNs) based super-resolution methods have been extensively studied and have achieved tremendous improvement. As a pioneering work, Dong et al. proposed the super-resolution convolutional neural network (SRCNN) [4] which only has three convolutional layers. Zhang et al. [5] further uses residual connections [6] and attention machanisms [5] to increase the quality of SR images.

Although existing SISR methods can provide good image super-resolution results than traditional algorithms, they still suffer from some shortcomings. For instance, such kind of methods apply the same network structure to process all the image spectral components, although flat image areas are less demanding to the SISR task than the textures image areas. Besides, these often improve the SISR reconstruction performance at the expense of considerably increasing the computational cost. On the contrary, Kong et al. proposed ClassSR [7] which deals with the different image spectral components with different networks. The textures image areas (e.g., hair, feathers) will be processed by more sophisticated operations than the flat (e.g., sky, land) image areas to relieve the computation burden. However, ClassSR approach still encounters problems, e.g. how many spectral ranges are necessary to correctly divide the image regions into textures areas and flat areas.

Many studies on SISR have been reported, they still face two challenges. First, though other algorithms such as attention mechanism can improve the performance of complex areas in image, there's still a gap to the ground truth (GT) one. Secondly, the number of parameters and computational cost of the network becomes higher to acquire better results. The presented SISR networks use loss functions that cover the full image grey value range, e.g. 0-255 for 8 bit images. It may increase the regression difficulty, in particular in textures image regions, because the image signals of different image regions do not obey the characteristics of fully dynamic range. This paper proposes a dynamic fine-grained loss for image super-resolution, and brings classification loss [8] in SISR, the main contributions include:

(1) This paper innovatively proposes a dynamic fine-grained loss based on the classification loss, which improves the performance without extra computational cost.

(2) Ablation experiments show that the method can greatly improve the reconstruction effect, especially in images with rich texture details.

(3) Extensive experiments conducted on benchmark SISR

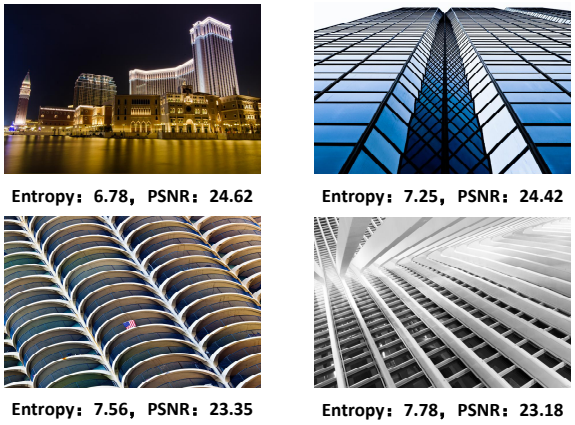

Fig. 1. Example of image entropy and PSNR.

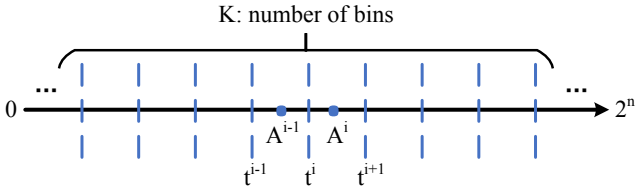

Fig. 2. Dividing bins evenly for pixel, the anchor A^i is the center between two thresholds t^i and $t^i + 1$.

show that the proposed method is also applicable in other networks and can increase the quality of SR images.

2. PROPOSED METHOD

In this section, the authors verify the performance of regression network in textures images and flat images. The entropy is used to define the texture complexity of images. Generally, it is difficult to reconstruct the images with larger entropy for more texture details. To prove this, we use LapSRN [9] to get SR images. The entropy and PSNR are computed from the HR and SR images. As shown in Figure 1, images with low entropy get better performance. Next, we introduced the dynamic fine-grained loss, which can improve the performance of SISR network.

2.1. Anchor Classification Loss

Different from the regression loss, as one can see in Figure 2, we choose a discretization method to generate several increasing scalar values of each pixel with threshold of 0 to 2^n. Each scalar value represents an independent bin [10]. The formula can be written as:

$$t^i = \frac{2^n * i}{K} \tag{1}$$

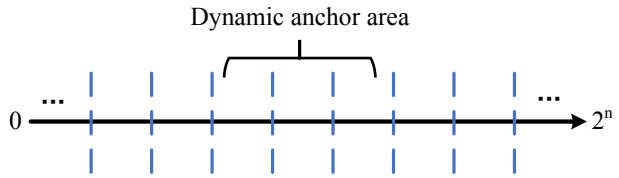

Fig. 3. Dynamic anchor area, which is composed of several bins.

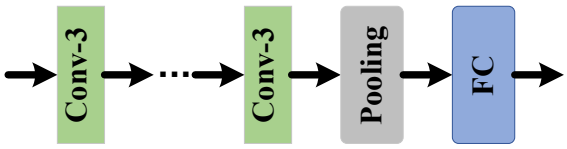

Fig. 4. Class-Module: aims to generate the dynamic anchor area.

where $t^i \in \{t^0, t^1, ..., t^K\}$ are discretization thresholds for pixel. There are K bins in total, we take each bin's center as our pixel anchor, then we get K pixel anchors. Each anchor's value is defined as follows:

$$A^i = \frac{t^i + t^{i+1}}{2} \tag{2}$$

where $A^i \in \{A^0, A^1, ..., A^{K-1}\}$ indicates the i_{th} pixel anchor. For pixel z whose value meets $t^i <= z < t^{i+1}$, it will be assigned to anchor A^i called *gt anchor*. The formula of anchor classification loss of each pixel is expressed as follows:

$$L_{c_1} = -\sum_{i=1}^{K} p_i.log(q_i) \tag{3}$$

where p_i represents the real distribution, which is got from the GT pixel of an image in question, and q_i represents the predicted distribution, which results from the output of the trained network, K is the number of bins and equals 32 in this paper.

2.2. Dynamic Regression Loss

With the classification step above, the network can easily find the area near the gt anchor, the next step is how to generate the final precise pixel. However, particularly those images with rich texture detail, the classification result of bin is often several bins away from the gt anchor. For example, if the index of classified bin is 2 or more bins away from gt anchor, it is meaningless to regression in anchor area. Thus dynamic regression loss is necessary. As shown in Figure 3, in dynamic regression loss [11], anchor area is formed from numbers of bins on both sides of the predicted bin including the predicted bin itself. The number of bins was computed dynamically with a Class-Module. The target of Class-Module

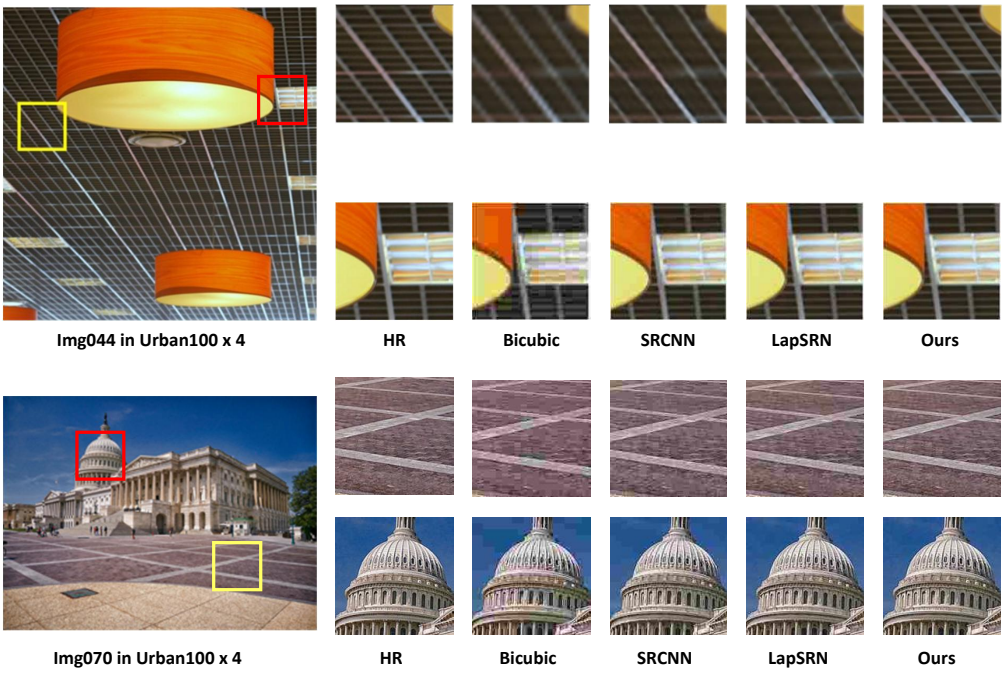

Fig. 5. Qualitative Comparison with SR methods on Urban100 x 4

is to generate the dynamic anchor area. As shown in Figure 4, we design the Class-Module as simple classification network, which consists of five convolutions, pooling and fully-connected layers. This is a simple lightweight network with low computational cost, which can already achieve satisfactory classification results. Referring to ClassSR, we propose loss for Class-Module with:

$$L_{c2} = -\sum_{i=1}^{M-1}\sum_{j=i+1}^{M}|P_i(x) - P_j(x)| \tag{4}$$

where M is the number of classes. In the paper, we set M to 3 and get three anchor area classification results P0, P1 and P2, which amount to 3, 5 and 7 respectively. They take odd number becasue one forms the anchor area from both sides of the predicted bin, as mentioned earlier. The L_{c2} is the negative number of distance sum between each class probability for a same sub-image.

The L1 regression loss is given:

$$L_r = \frac{1}{hw}\sum_{i=1}^{h}\sum_{j=1}^{w}|S_{i_j} - T_{i_j}| \tag{5}$$

where h and w is the height and width of an image in question. S_{i_j} stands for the pixel value predicted by the trained network, T_{i_j} is gt anchor. The final loss function is given as:

$$L = L_r + \frac{1}{hw}\sum_{i=1}^{h}\sum_{j=1}^{w}(\alpha L_{c1} + \beta L_{c2}) \tag{6}$$

where α and β are the weights to balance different loss functions.

Table 1. Average PSNR using the same model (LapSRN) with Classification Loss of Static anchor area (CS Loss, static anchor area set to 5), Fine-Grained Dynamic Loss(FD Loss, dynamic anchor area).

Name	Set5	Set14	BSD100	Urban100
Baseline	31.65	28.27	27.36	25.34
+CS Loss	32.15	28.83	27.89	25.84
+FD Loss	**32.23**	**28.98**	**27.98**	**25.96**

3. EXPERIMENTS

Following [12], we use 800 high-quantity training images from DIV2K dataset as the training set. Several benchmark datasets are used for testing, namely Set5 [13], Set14 [14], BSD100 [15], and Urban100 [15], each with diverse characteristics. We perform the experiments with bicubic degradation models. All the results are evaluated with two commonly used metrics: PSNR (peak-to-noise-ratio) and SSIM (structural similarity index), on the Y channel of the YCbCr space. During training, we perform random vertical flipping and horizontal flipping on the training set. At each training mini-batch, low resolution RGB patches with size 64x64 are provideds as input. The model is trained using the ADAM optimizer [16] with learning rate set to the maximum

Table 2. Average PSNR/SSIM for 2x, 3x, 4x SR. The best results are heghlighted.

Scale	Method	Params	Mult-Adds	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM
2x	Bicubic	-	-	33.66 / 0.9299	30.24 / 0.8688	29.56 / 0.8431	26.88 / 0.8403
	SRCNN	57 K	52.7 G	36.66 / 0.9542	32.45 / 0.9067	31.36 / 0.8879	29.50 / 0.8946
	VDSR	665 K	612.6 G	37.53 / 0.9590	33.05 / 0.9130	31.90 / 0.8960	30.77 / 0.9140
	LapSRN	862 K	186.2 G	37.76 / 0.9590	33.42 / 0.9166	32.09 / 0.8978	30.92 / 0.9256
	Ours	934 K	208.4 G	**38.36 / 0.9684**	**34.02 /0.9210**	**32.54 / 0.9026**	**31.67 / 0.9314**
3x	Bicubic	-	-	30.39 / 0.8682	27.55 / 0.7742	27.21 / 0.7385	24.46 / 0.7349
	SRCNN	57 K	52.7 G	32.75 / 0.9090	29.30 / 0.8215	28.41 / 0.7863	26.24 / 0.7989
	VDSR	665 K	612.6 G	33.67 / 0.9210	29.78 / 0.8320	28.83 / 0.7990	27.14 / 0.8290
	LapSRN	862 K	186.2 G	33.92 / 0.9234	30.01 / 0.8345	29.11 / 0.8023	27.51 / 0.8312
	Ours	934 K	208.4 G	**33.48 / 0.9286**	**30.56 /0.8396**	**29.62 / 0.8065**	**28.14 / 0.8367**
4x	Bicubic	-	-	28.42 / 0.8104	26.00 / 0.7027	25.96 / 0.6675	23.14 / 0.6577
	SRCNN	57 K	52.7 G	30.48 / 0.8628	27.50 / 0.7513	26.90 / 0.7101	24.52 / 0.7221
	VDSR	665 K	612.6 G	31.35 / 0.8830	28.02 / 0.7680	27.29 / 0.7260	25.18 / 0.7540
	LapSRN	862 K	186.2 G	31.65 / 0.8863	28.27 / 0.7746	27.36 / 0.7282	25.34 / 0.7572
	Ours	934 K	208.4 G	**32.23 / 0.8896**	**28.98 /0.7792**	**27.98 / 0.7324**	**25.96 / 0.7618**

convergent value (10^{-3}), applying weight normalization in all convolutional layers. The learning rate is decreased by half every $2x10^5$ back-propagation iterations. The proposed architecture has been implemented using PyTorch [17] and trained on NVIDIA 2080 Ti.

3.1. Ablation Study

In order to prove the effectiveness of the proposed method, LapSRN is selected as the baseline. In the training part, the number of bin is 32, and the other parts are exactly the same except the loss function. As shown in Table 1, the PSNR of the original LapSRN for Set5 is 31.65 dB; replace with the anchor classification loss (with anchor area as 5), the PSNR value is higher than baseline by 0.5 dB. Additinally with the dynamic anchor area the PSNR increases by 0.58 dB. In baseline, the larger the value of image entropy, the lower the PSNR of the reconstructed image. However, with our method, especially for images with rich texture details, the PSNR improvement after reconstruction is more obvious.

3.2. Evaluation results

Our method is evaluated on several classic benchmarks such as Set5, Set14, BSD100 and Urban100, and compared with the original baseline. The experimental results are given in Table 2. As shown in the table, our method improves PSNR on LapSRN compared with baseline. This shows the advantages of the proposed method, especially in X4 SR, the proposed method is far more than baseline, and it also shows that our method can be transplanted to other networks. In addition to the quantitative results, we visualize the super resolution results in Figure 5, which including some common scenes. It can be observed that our method can reconstruct the texture consistent with HR while other methods fail to restore the texture and produce some irrelevant artifacts.

4. CONCLUSION

For image reconstruction, one has been using the regression loss. However, for images with complex texture details, image reconstruction becomes difficult, and the network can not achieve good performance. In order to solve this problem, this paper proposes a dynamic classification regression loss, which first transforms the regression problem into a classification problem, then determines the anchor area, and then carries out small-scale regression within the range of dynamic anchor area, so as to reduce the learning difficulty of the network and improve the reconstruction performance. Compared with regression loss, our method has made a significant improvement in PSNR, especially for images with rich texture details. Moreover, a large number of experiments show that our method can be applied to more networks and improve the reconstruction performance.

5. REFERENCES

[1] Lei Zhang and Xiaolin Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE transactions on Image Processing*, vol. 15, no. 8, pp. 2226–2238, 2006.

[2] Kaibing Zhang, Xinbo Gao, Dacheng Tao, and Xuelong Li, "Single image super-resolution with non-local means and steering kernel regression," *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4544–4556, 2012.

[3] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen, "Single image super-resolution via a holistic attention network," in *European Conference on Computer Vision*. Springer, 2020, pp. 191–207.

[4] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*. Springer, 2014, pp. 184–199.

[5] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 286–301.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] Xiangtao Kong, Hengyuan Zhao, Yu Qiao, and Chao Dong, "Classsr: A general framework to accelerate super-resolution networks by data characteristic," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12016–12025.

[8] Katarzyna Janocha and Wojciech Marian Czarnecki, "On loss functions for deep neural networks in classification," *arXiv preprint arXiv:1702.05659*, 2017.

[9] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 624–632.

[10] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao, "Deep ordinal regression network for monocular depth estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2002–2011.

[11] Raphaël Turcotte, Yajie Liang, Masashi Tanimoto, Qinrong Zhang, Ziwei Li, Minoru Koyama, Eric Betzig, and Na Ji, "Dynamic super-resolution structured illumination imaging in the living brain," *Proceedings of the National Academy of Sciences*, vol. 116, no. 19, pp. 9586–9591, 2019.

[12] Jie Liu, Wenjie Zhang, Yuting Tang, Jie Tang, and Gangshan Wu, "Residual feature aggregation network for image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2359–2368.

[13] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," 2012.

[14] Roman Zeyde, Michael Elad, and Matan Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*. Springer, 2010, pp. 711–730.

[15] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2010.

[16] Zijun Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 2018, pp. 1–2.

[17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," 2017.