Lightweight Unsupervised Super-Resolution with Emphasis on Structural Similarity

1st Xinghao Chen
School of Electrical and Computer Engineering
Georgia Institute of Technology
Shenzhen, China
xchen785@gatech.edu

2nd Meng Zhang
School of Electrical and Computer Engineering
Georgia Institute of Technology
Shenzhen, China
mzhang495@gatech.edu

***Abstract*—Contemporary state-of-the-art methods for image super-resolution (SR) are principally supervised deep models, which cost heavy computatioanl efforts and do not fully exploit the information in the input image itself. Our work, based on [3], aims at improving the SR performance of residual dense network. We build an unsupervised, relatively shallow model for image super-resolution, which majorly focuses on the struture similarity (SSIM) instead of peak signal-to-noise ratio (PSNR) of output. Along with our model achieving competitive results especially in SSIM, we further propose its data augmentation techniques and a creative loss function combining SSIM and PSNR.**

***Index Terms*—super resolution, unsupervised learning**

I. Introduction

Single image super-resolution (SISR) is to generate an output image of high resolution (HR) based on a low-resolution (LR) input, which is potentially of lower subjective and objective quality. In recent years, deep-learning-based implementations [1] [2] [3] [4] have won a significant edge over other methods. In these state-of-the-art deep models, efficient architectures like dense blocks and attention mechanisms are applied to combine low- and high-level information. With context of attention and features extracted from the LR image, the deep models can estimate more delicate kernels for better interpolation.

However, the models mentioned above require heavy computational efforts and large datasets. Often tens of millions of parameters and trillions of operations are included in the model. In order to reduce computational efforts (because we do not have powerful GPUs), our work, inspired by [5], attempts to train a lightweight, image-specific model. Our training set contains only the image that we are to perform SR on. Meanwhile, with respect to the output quality, we focus majorly on SSIM because it preserves high-frequency information better than PSNR, and is based on the assumption that the human visual system extracts structure. Yet we still would like to preserve decent PSNR.

II. Network Design

Referring to [3], we also designed a residual dense network. However, limited by computational power, we introduced only 8 layers of 64×64 convolution + ReLU operations.

A. Use of Dense Connection

The DenseNet model proposed by [6] has lead a trend of using dense connections. With the features generated from each layer serving as the input of all subsequent layers, dense connection encourages the reuse of features, which potentially saves computational efforts, enhances data propagation and extracts multi-hierarchy information efficiently.

B. Use of Residual Learning

Residual learning, which learns the difference between the output and the ground-truth, had long been a widely employed strategy in SR even before ResNet [7] was put forward. Global residual learning in SR, implemented simply via an element-wise addition of input to the output, avoids a huge difference of SR output between one image block and another. Therefore, residual strategy can also reduce the complexity of models.

III. Training Strategies

We are to denote the input image as the high-resolution (HR), and the desired output as the super-resolution (SR). Enlightened by [5], we furthermore degrade the input HR image with bicubic interpolation. The futher-degraded image, denoted as the low-resolution (LR), is then bicubically upscaled to HR size again.

The downscale-upscale process is denoted as "back-projection". Afterwards, blocks of 128×128 pixels are picked from the HR image, and we train the model with paired blocks from the HR and the bicubically upscaled LR image. Some data augmentation techniques, which will be introduced in subsection III-B, are also used before downscaling the HR image. In order to generate the final SR output, the HR image is bicubically upscaled to SR size, and amended by our model.

A. Estimation of Kernel: Supervised versus Unsupervised

If we view SR as an interpolation problem, we may find that a supervised model is designated to estimate the upscaling interpolation kernel. In many cases, including that of popular datasets, we do not know the downscaling kernel, and there is little reason for us to assume that downscaling kernels of different LR images are the same, or of very few categories. Consequently, supervised models learn different kernels from different pairs of LR and HR images, and may not exploit the best performance for each single image. In other words, the generalization ability of supervised kernels is therefore doubtful. Our unsupervised model, however, is trained only with the image blocks from the LR image. In this way the problem of generalization is partially averted. Though we used primarily the bicubic kernel in our experiments, we assume that the output can be potentially further improved if the downscaling kernel is actually known.

B. Data Augmentation strategies

In each epoch of training, we randomly execute some augmentation operations on the HR image before downscaling it. The HR image may take affine transforms including rotation and rescaling. Shifting is implemented so that the boudaries of the image does not remain the same. We also randomly, but at a small probability, apply various kernels in back-projection. The downscaling and upscaling kernels are not necessarily the same in a single back-projection operation. Besides, we have considered adding random noise in back-projection, but we suspect that noise without delicate control may improve generalization only for noisy HR images (which is not a foremost demand in our unsupervised method) at a heavy price of overall performance. Also we had little time and limited computational power to implement noise.

C. The Gradual-Upscaling Strategy

This technique is imported from [5]. We tried to train the model with gradually upscaled image. That is, in back-projection, we did not directly upscale the LR image by a factor of 2. Instead, our upscaling factors in (x, y) directions are chosen to be $(1, 1.5), (1.5, 1), (1.5, 2), (2, 1.5), (2, 2)$ in sequence. Each re-scaled LR image is used to train the model continuously so that the factor-2 features are learned gradually. This technique, also used by [5], derives a slight improvement in both PSNR and SSIM.

IV. Loss Metrics: the Improvement of SSIM

One of our principal goals is to improve the SSIM of output, compared to the output of RDN. Initially our idea was to use specialized image enhancement techniques on the output image, but we did have no confidence in any method to amplify SSIM. The cost of manual efforts was also a concern that stopped us to develop an SSIM-focused enhancement from scratch. But we did improve SSIM simply by using it as part of the loss function.

In SR problems, classical and dominating choices of loss functions are L1 and L2 losses (i.e. the mean absolute error and mean square error, respectively). These losses undoubtedly do good to PSNR, but not necessarily generates the best SSIM. Therfore, inspired by [8] which proposes a new loss combining L1 and multi-scale SSIM, we choose our loss simply as

$$-40\text{SSIM} - \text{PSNR} \tag{1}$$

where PSNR is in decibel (typically ranging from 20 to 40), and SSIM $\in [0, 1]$. Our design, compared to traditional L1 and L2 losses, stresses less emphasis on PSNR when the output PSNR is large enough, since our PSNR loss gives logarithmic, instead of linear or quadratic element-wise error. Compared with L1 and L2 losses, our PSNR loss leads the importance of pixel-wise error to decay exponentially. With enough (regulated by the coefficent of SSIM) PSNR ensured, the model would pay primary efforts for SSIM. The potential advantages of our loss will be futher discussed in section V.

V. Experiments and Results

We only applied our model to bicubic-degraded Set14 for $2\times$ scaling. (We did not pick an experiment with the best result among many experiments!) The average output PSNR and SSIM is 31.894 and 0.9425, respectively. If the gradual-upscaling strategy mentioned in III-C is not applied, the result is 31.891 and 0.9417. This result outperformed [3], [5] and many other state-of-the-art models in SSIM, but we did not achieve the best PSNR. Our method preserves the primary features of the image, but some high-frequency information can still be lost

(See fig. 1). Without exploiting the information from HR images, it might be more difficult for our model to infer very detailed structures thoroughly. Additional weak supervision using HR images, which contain more abundant details, might be a solution to the problem.

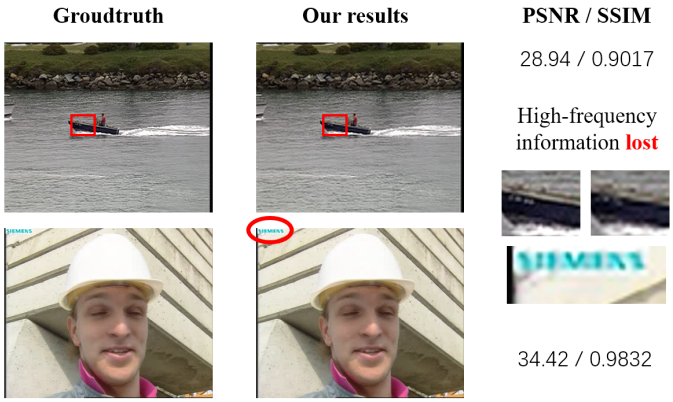

Fig. 1. Examples of lost features

Besides, given our results, we are to raise a hypothesis that our SSIM-based loss outputs relatively better SSIM, but worse PSNR when the image is of high contrast, and vice versa (See fig. 2). It might be the intrinsic nature of the SSIM loss function to emphsize the global contrast, which is not possessed by element-wise losses. More results generated by merely element-wise losses are needed to support our hypothesis.

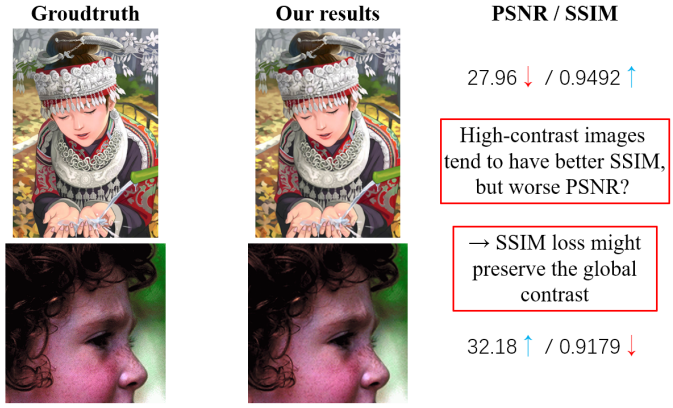

Fig. 2. Hypothesis on the nature of SSIM loss

VI. Conclusion

Applying a wide range of techniques, our unsupervised model generates decent output from Set14, fulfiling our goal raised in the project proposal. Using residual dense blocks and the unsupervised strategy, we save large amounts of computation but extract the inherent features from the input image efficiently. The generalization problem is partially averted. Data augmentation tactics improves the performance of our model. We then discuss our loss function combining logarithmic element-wise error and SSIM and raised the hypothesis about its advantage in preserving contrast. We may possibly further amend the PSNR of our output through weak supervision which extracts more detailed features from the HR image. Finally we surmise that the SSIM loss might preserve better contrast.

Although most of the approaches used by us have been common practice in digital image processing, we have managed to build a well-founded creative combination in our project. We failed to come up with any fire-new method, which was expected in our initial proposal, to directly improve the output SSIM, but we are lucky in that neural networks has made it for us automatically.

Acknowledgement

We would really like to thank our professor and teaching assistant who are industriously commited to all the lectures, homework, exams and project which help us efficiently investigate the most important topics and most requisite knowledge! Without your affluent course, we could have gone through much more uncertainty and diffculty if we had to help ourselves groping about in the whole territory of digital images.

References

[1] Zhang, Yulun , et al. "Image Super-Resolution Using Very Deep Residual Channel Attention Networks." (2018).
[2] Dai, Tao , et al. "Second-order Attention Network for Single Image Super-Resolution." 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) IEEE, 2019.
[3] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) IEEE, 2018.
[4] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deepresidual networks for single image super-resolution," CVPRW, 2017.
[5] A. Shocher, N. Cohen, and M. Irani, "zero-shot super-resolution using deep internal learning," CVPR, 2018.
[6] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," CVPR, 2017.
[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," CVPR, 2016.
[8] Zhao, Hang , et al. "Loss Functions for Neural Networks for Image Processing." CVPR, 2015.