A Multi-model Super-Resolution Training and Reconstruction Framework

Ninghui Yuan, Dunbo Zhang, Qiong Wang, Li Shen

A Multi-model Super-resolution Training and Reconstruction Framework

Ninghui Yuan[1], Dunbo Zhang[1], Qiong Wang[1], and Li Shen[1,2]

[1] School of Computer, National University of Defense Technology Changsha, Hunan, China, 410073

[2] Corresponding Author: lishen@nudt.edu.cn

Abstract. As a popular research field of computer vision, super-resolution is currently widely studied. In the past, the size of the training set required for super-resolution work was too large. A large training set would cause more resource requirements, and at the same time, the time overheads of data transmission would also increase. Moreover, in super-resolution work, the relationship between the complexity of the image and the model structure is usually not considered, and images are recovered in same depth. This method often cannot meet the SR-reconstruction needs of all images. This paper proposes a new training and reconstruction framework based on multiple models. The framework prunes the training set according to the complexity of the images in the training set, which significantly reduces the size of the training set. At the same time, the framework can select the specific depth according to the image features of the images to recover the images, which helps to improve the SR-reconstruction effect. After testing different models, our framework can reduce the amount of training data by 41.9% and reduce the average training time from 2935 minutes to 2836 minutes. At the same time, our framework can improve the average SR-reconstruction effect of 65.7% images, optimize the average perceptual index from 3.1607 to 3.0867, and optimize the average SR-reconstruction time from 101.7 seconds to 66.7 seconds.

Keywords: Super-resolution, classification, fusion, multi-model.

1 Introduction

Super-resolution(SR) is an important branch of machine vision. The main function of super-resolution is to improve the clarity of the enlarged image and reduce the image quality degradation caused by image upscaling. From the simple mathematical methods to the methods based on deep learning, the effect of SR-reconstruction is constantly improving. After the deep learning method is widely used, the optimization method of deep learning is applied to the field of SR. The optimization direction of deep learning methods is not clear, which also makes the optimization cycle of the SR models to be longer. At present, super-resolution still faces some challenges:

Firstly, the SR model becomes more and more complex, which will cause overfitting problems. In the process of SR-reconstruction, too deep or too shallow models can reduce SR-reconstruction effect. Insufficient model depth will lead to insufficient extraction of image feature information, which can reduce the SR-reconstruction effect. However, too deep models will cause overfitting and increase resource requirements. Therefore, we hope the model depth matches the complexity of SR-reconstruction. There has been some works using multiple models for SR-reconstruction, such as MMSR(a Multi-Model Super Resolution framework) [21]. However, due to the limitation of the classification effect and the impact on the pruning of the training set, the SR-reconstruction effect of MMSR is not satisfied. In MMSR, images with simple textures are classified into lower depth models for SR-reconstruction, and a better SR-reconstruction effect is obtained in lower depth models than in deeper models. The SR-reconstruction result score is better because there are fewer artifacts and distortions in the SR-reconstruction result, but these SR-reconstruction results may lose detailed texture information. Therefore, if the images with simple textures can be recovered in models of different depths, and finally the results of the SR-reconstruction are fused, the SR-reconstruction results will get better score and lose less texture.

Secondly, in order to ensure the SR-reconstruction effect, a larger training set is generally required, which greatly increases the training time and the resource requirements. The SR training set is obtained by processing an image set composed of many images. During the processing, the images in the original image set are first cut into fragments. For training more easily, the fragments are often processed into specific data file formats. We observed the image fragments after cutting and found that there are many image fragments having almost no texture features, such as the sky, the water surface, the wall, and so on. These fragments have little impact on training, so the fragments can be pruned. The size of the training set after pruning will be greatly reduced, which can reduce time overhead and storage overhead.

Based on above analysis, we propose and implement a new SR framework based on multiple-depth models. The framework is divided into two parts: training module and reconstruction module. In the training module, the training set is reduced by pruning. In the reconstruction module, the test set is classified into models of different depths for training. The images recovered in the shallowest model need to be recovered in a medium-depth model, and finally fused as the final reconstruction result. The contributions of our framework are as follows:

– A SR framework based on multiple-depth models and an image classification strategy based on random forest are proposed. Unlike the previous single-depth model construction method, our framework can accurately assign different types of images to the appropriate models for SR-reconstruction, thereby improving SR-reconstruction effect. At the same time, multi-depth SR-reconstruction can also improve the parallelism of the SR-reconstruction process and accelerate the SR-reconstruction process. our framework can increase the average SR-reconstruction effect of 65.7% images, optimize the average perceptual index from 3.1607 to 3.0867, and reduce the average SR-reconstruction time by 34.4%.

- An image fusion algorithm is proposed. In the experiment, the fusion algorithm can significantly reduce the loss of texture information of images and improve the visual effect of the SR-reconstruction results.
- A pruning algorithm based on image texture features and edge information is proposed to prune the training set of the SR model. Through the pruning algorithm, the size of the training set is decreased by 41.9% and the average training time is reduced by 3.4%.

The rest of this paper is organized as follows. Section 2 lists some related works. Section 3 introduces our framework in detail. In section 4, experimental results are given and the performance of our framework is evaluated. And in section 5, some conclusions are given.

2 Related Work

2.1 Super-resolution

Super-resolution: SR-reconstruction is an important work in the field of machine vision [3, 5–7, 12, 13, 18, 22, 23]. Before the deep learning methods were widely used, SR-reconstruction mainly relied on mathematical methods to calculate and predict the upscaled images information through the information in the original small images. SRCNN (Super-Resolution Convolutional Neural Network) [1] [2] introduced the method of deep learning to SR-reconstruction for the first time. SRCNN achieved SR-reconstruction effect superior to the past mathematical methods through a three-layer convolution neural network. After SRCNN, deep learning based methods have gradually become the mainstream in the field of super-resolution. The optimization of deep learning based methods mainly comes from the continuous increase of the depth of the model and the optimization of the network structure. For example, VDSR (Very Deep network for Super-Resolution) [8] introduced ResNet into SR and many models are constructed with ResNet such as MSRResNet [11]. SRDenseNet [17] introduced DenseNet, and SRGAN (Super-Resolution using a generative adversarial network) [11] introduced GAN (generative adversarial network) [4]. ESRGAN [19] is based on both DenseNet and GAN. These networks have made breakthrough progress. At the same time, RED (Residual convolutional Encoder-Decoder networks) [15], DRCN (Deeply-recursive convolutional network for image super-resolution) [9], LapSRN (Deep laplacian pyramid networks for fast and accurate super-resolution) [10], SftGAN (Recovering Realistic Texture in Image Super-resolution by Deep Spatial Feature Transform) [20] and other network structures have also achieved good results. Among them, most effective models are based on ResNet, GAN, and DenseNet. In this paper, to fully verify the effect of our framework, we have also tested the models of these three structures. We put MSRResNet, SRGAN, and ESRGAN into our framework.

In addition to the model structure, how to quantitatively evaluate the SR-reconstruction results is also a problem to be considered in SR work. In the early models, traditional evaluation metrics are mainly based on mathematical calculation, such as PSNR (Peak-Signal to Noise Ratio), SSIM (Structural SIMilarity) and other methods. These evaluation metrics are traditional imagery evaluation methods, which quantify the image structure, signal-to-noise ratio, and other indicators, but these evaluation standards cannot match well with the feel of human eye. SRGAN adopts the perceptual index (PI) [14] [16] to evaluate the SR-reconstruction results. The lower a perceptual index score is, the better the image quality is. Therefore, PI is also used as the evaluation metric of image quality in this paper.

2.2 Image Classification

The effect of the multi-depth framework depends greatly on the effect of the classification results. The current image classification methods are mainly divided into two types, one is the mathematical method and the other is the deep learning method.

Traditional methods mainly use mathematical methods to describe image features and classify images through some classification structures. For example, Decision Tree and random forests can be used for image classification. Due to its simple training, clear structure and strong interpretability, decision tree is used to classify images commonly. The effect of classification depends on the features provided by the user to the decision tree. Random forest is a set of decision trees. Multiple decision trees vote on the classification results, which can improve the accuracy of classification. The extraction of image features mainly depends on some mathematical methods. The characteristics of images mainly include edge characteristics, image channel value variance and so on. The edge detection operator can be used to extract the texture of the image well. Common edge detection operators include sobel operator, canny operator and so on. In MMSR, the TVAT [21] algorithm is proposed, which can well calculate the variance of the image channel value.

Non-traditional methods mainly use deep learning models for classification. At present, such methods have achieved good classification results. The deep learning based approaches extract and generate image features through convolutional neural networks and other network structures and classify images based on their features.

The advantages of traditional image classification methods is that they have strong interpretability and does not require a complicated training process. The advantage of the non-traditional classification methods is that they have a good effect on some complex cases that cannot be solved by traditional methods. In the multi-depth framework, not only the effect but also the efficiency needs to be considered, therefore our framework uses traditional methods for classification.

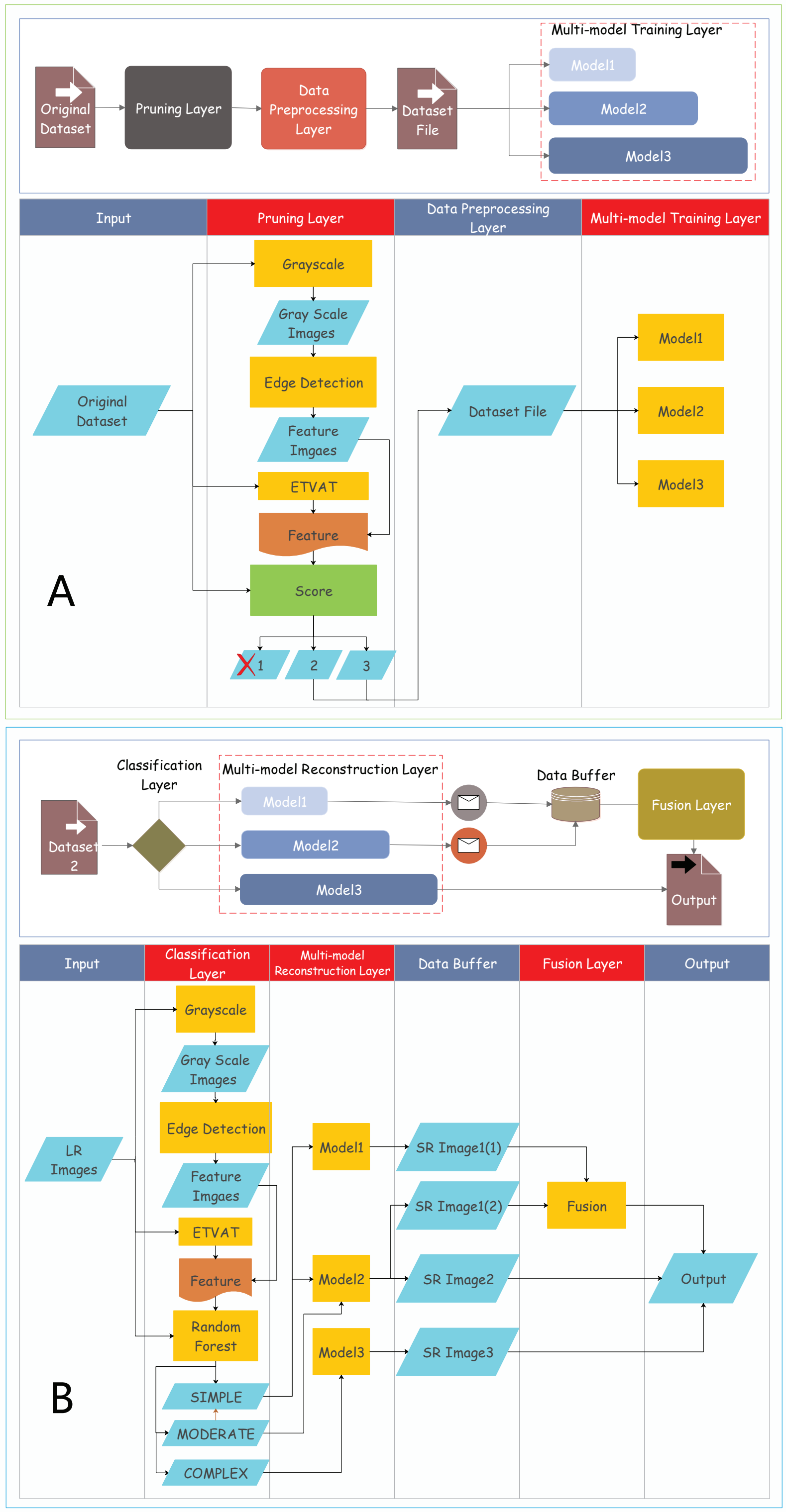

Fig. 1. The structure and data flow of the training module.

3 Multi-Model SR training and reconstruction framework

In order to address the issues mentioned above, we propose a multi-model training and SR-reconstruction framework. The framework is mainly divided into training module and SR-reconstruction module. In the experiment, we adjust the model depth by adjusting the number of basic blocks in the model. The basic block is the unit that constitutes the model. A basic block contains several convolutional layers. The depth of the model can be easily adjusted by setting the number of basic blocks in a model. The structure and data flow of the training module are shown in the A part of Fig. 1. The task of the training module is mainly to train the multiple models and reduce the size of the training set. The main design idea of the training module is to prune the images with less texture from the original training set. The size of the training set after pruning is significantly reduced compared to that of the original data set. The structure and data flow of the reconstruction module are shown in the part B of Fig. 1. The task of the reconstruction module is to recover low-resolution images. We input images into different models for SR-reconstruction through the classification layer, thereby improving the effect and efficiency of reconstruction. At the same time we propose fusion layer to prevent the image reconstructed in the low-depth model from losing too much texture information.

3.1 Pruning layer

In the original training set, the data is saved in the form of images. Some images in the original training set do not have complex texture features, and even some of the images are filled by solid colors. These images have little effect on the training results, and can be removed from the training set. The pruning layer uses ETVAT(Enhanced TVAT) and the edge detection operator to determine which images should be pruned. In the pruning layer, the training image is grayscaled firstly. The purpose of grayscale is to reduce the interference of noise and illumination when performing edge detection. After grayscaling, the module uses edge detection operators to extract the features of the images and get the feature maps. The final feature maps will be used to score the images together with ETVAT.

As shown in Fig. 2, the pruning algorithm we propose is based on TVAT and edge detection operators. In part A, we use the edge detection operator to calculate score1. We first use the edge detection operator to calculate the edge feature map of the image. After that, we add the gray values of the pixels in the feature map as the first score, and this score is called Edge_score. Although this calculation method is very simple and reduces the amount of information in the edge map, it is enough for judging the complexity of the image. And we propose a voting mechanism of three channels(R,G,B) instead of calculating the mean value of the three channels to calculate score2, because we found that there are large differences in the change of different channels. Therefore, in part B we use the values of the three channels to perform TVAT operations and vote. If the change rate of more than two values exceeds the threshold, we think that this point should be calculated into the final TVAT value. We call this method ETVAT, ETVAT will help us get one score in the voting mechanism. Then in part C we use ETVAT and Edge_score to calculate the final score. The images with the final score of 1 will be pruned. In the pruning algorithm, there are several adjustable thresholds: threshold0, threshold1, threshold2, threshold3, threshold4, which are used to adjust the strength of pruning. In the experiment of this paper, we adjust the strength of pruning to about 40%.

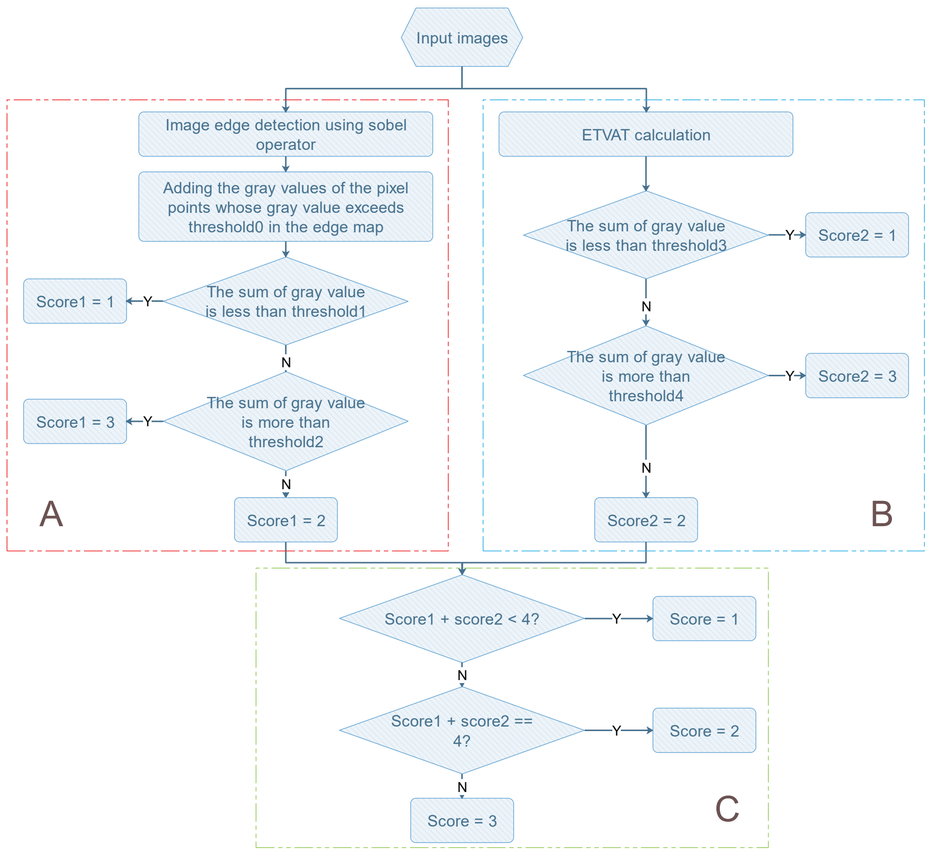

Fig. 2. The process of pruning algorithm.

3.2 Classification layer

After the low-resolution images enters the reconstruction module, they will be classified first. We find that the training model required for images with simple textures is often shallower than that for images with complex textures. Therefore, we can also classify images based on the texture complexity of the images. The feature extraction process in the classification layer is like that in the pruning layer. After the feature extraction is completed, we use random forest to classify the images. The images are divided into three categories labeled as “SIMPLE”, “MODERATE” and “COMPLEX”. The three labels represent images with the simplest texture, the image with moderate texture complexity and images with the most complicated texture. In the multi-model reconstruction layer, images labeled as “SIMPLE” enters Model1 and Model2 for SR-reconstruction at the same time, images labeled as “MODERATE” enters Model2 for SR-reconstruction, and images labeled as “COMPLEX” enters Model3 for SR-reconstruction. After SR-reconstruction, the reconstruction results of the images labeled “MODERATE” and “COMPLEX” directly enter the output layer as the results. Images labeled as “SIMPLE” have two output results, one from Model1 and the other from Model2. These two SR-reconstruction results will enter the fusion layer. After the fusion layer fuses the two results, the fusion result is output to the output layer.

The classification layer is one of the core components of the reconstruction module. Through observation, we found that complex images often require more complex models for SR-reconstruction, while simple images often do not require particularly complex model reconstruction. Based on this observation, we simplify the classification problem to how to classify images based on their complexity. We have designed a classification layer based on random forest.

We use ETVAT and Edge_score calculation methods to extract image features, and the extracted features are used as a new training set. This training set contains only feature values and label, which is suitable for the construction of Random Forests. And because we have a limited number of extracted features (you can set the number of feature extraction yourself, in this paper we extracted 5 features, which are the three feature values of ETVAT, Edge_score, and Edge_score with threshold limit). It is worth noting that too many features may cause overfitting.

The next step is the construction of random forests. The main tunable parameters are the number of trees in the random forest and the maximum tree depth.

3.3 Fusion layer

The fusion layer is another important functional layer in the reconstruction module. Its purpose is to avoid the loss of detailed information of the reconstructed images from the simple model. The fusion algorithm is shown in algorithm 1. There is a buffer before the fusion layer because the speeds of SR-reconstruction in different models are different. We must wait for both results to be reconstructed before they can be input into the fusion layer. It has three inputs, SR image1 and SR image2 to be fused and feature map generated using the sobel operator, which identifies that pixels in SR image1 are in the texture areas or in non-texture area. For each pixel, if it locates in the texture ares, the texture information generated by model2 will be mainly added to the result SR

Algorithm 1 Image Fusion algorithm

Input: SR image1, SR image2, feature map
Output: the result SR image

for i in row of feature image **do**
  for j in col of feature image **do**
    if $thefeature[i][j] = 1$ **then**
      $fusionresult[i][j] = 0.3 * SRimage1[i][j] + 0.7 * SRimage2[i][j]$.
    else
      $fusionresult[i][j] = 0.7 * SRimage1[i][j] + 0.3 * SRimage2[i][j]$.
    end if
  end for
end for

image. Otherwise, the non-texture information generated by the model1 will be mainly added to the result.

4 Experiments

4.1 Environment Setup

To evaluate the performance of our mechanism, we construct a small-scale cluster which consists of three heterogeneous "CPU+GPU" nodes. The main system parameters of each node are listed in Table. 1.

Table 1. System parameters of each node.

HW/SW Module	Description
CPU	Intel ® Xeon® E5-2660 v3 @2.6GHz x2
GPU	NVIDIA Tesla K80 x2
Memory	64 GB
OS	Linux CentOS 7.4
Development Environment	Anaconda 3, Pytorch 1.0

4.2 Experiment details

We use DIV_2K dataset train models, and use PRIM dataset to test. DIV_2K is a commonly used super-resolution training set with a total of 800 images, and PRIM dataset is a standard super-resolution competition test set with a total of 100 images. To verify the effectiveness of the model, we put different types of models into the framework for testing and compare the effects with the original model. Most of the current SR models are based on ResNet, DenseNet and GAN structures. Therefore, in the experiment, we use MSRResNet to represent ResNet structures, SRGAN to represent ResNet and GAN structures, and ESRGAN to represent DenseNet and GAN structures. These three models can get good SR-reconstruction effect and they can be used as basic models to verify the effect of our framework.

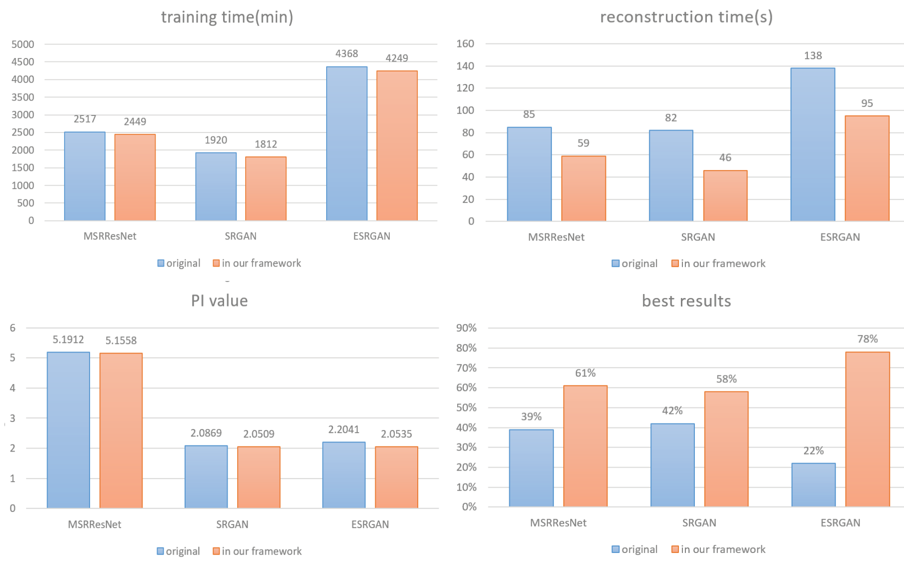

Fig. 3. Comparisons of the effects of our framework and the baseline models

In the experiment, to effectively reduce the size of the training set, we adjusted the pruning strength to about 40%. The size of the training set running in the original model was 21401.8M, and the size of the training set after the operation was reduced to 12436.6M. The experimental results are shown in Fig. 3. For MSRResNet, we can reduce the training time from 2517 minutes to 2449 minutes. The SR-reconstruction quality of 61% test images has been improved and the average perceptual index has dropped from 5.1912 to 5.155833, at the same time SR-reconstruction time has been optimized from 85 seconds to 59 seconds. For SRGAN, the training time can be reduced from 1920 minutes to 1812 minutes. SR-reconstruction quality of 58% test images have been improved and the average perceptual index has dropped from 2.0869 to 2.0509, while SR-reconstruction time has been optimized from 82 seconds to 46 seconds. For ESRGAN, the training time can be reduced from 4368 minutes to 4249 minutes. The SR-reconstruction quality of 78% test images has been improved and the average perceptual index has dropped from 2.2041 to 2.0535. The SR-reconstruction time has been optimized from 138 seconds to 95 seconds.

5 Conclusion

In the study of super-resolution, we find that a single model cannot satisfy the requirements of all input images to achieve the optimal SR-reconstruction effect. At the same time, we find that the size of the training set is usually too large, which also brings great difficulties to the research. In order to solve these problems, this paper proposes a super-resolution training and reconstruction framework based on multiple models. The framework is divided into training module and reconstruction module. In the training module, the pruning layer prunes the training set based on ETVAT and Edge_score, which effectively reduces the size of the training set and reduces the difficulty of training. The multi-model training layer trains multiple models at the same time. In the reconstruction module, the classification layer uses random forest for classification. The classification layer is based on random forests. The classification layer classifies different kinds of images into a suitable model for reconstruction and uses the fusion layer to increase the texture information of the SR-reconstruction result. In the experiment, we put MSRResNet, SRGAN and ESRGAN into the framework for optimization. After testing different models, our framework can reduce the amount of training data by 41.9% and reduce the average training time by 3.4%. For reconstruction, our framework can increase the average SR-reconstruction effect of 65.7% images, optimize the average perceptual index from 3.1607 to 3.0867, and reduce the average SR-reconstruction time by 34.4%. Our framework can improve the effect of super-resolution models while reducing the resource requirements.

6 Acknowledgement

This work was supported by the National Science Foundation under Grant No. 61972407 and Guangdong Province Key Laboratory of Popular High Performance Computers 2017B030314073.

References

1. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. CoRR **abs/1501.00092** (2015), http://arxiv.org/abs/1501.00092
2. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. CoRR **abs/1608.00367** (2016), http://arxiv.org/abs/1608.00367
3. Fu, Y., Zhang, T., Zheng, Y., Zhang, D., Huang, H.: Hyperspectral image super-resolution with optimized rgb guidance. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
4. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial networks. ArXiv **abs/1406.2661** (2014)
5. Gu, J., Lu, H., Zuo, W., Dong, C.: Blind super-resolution with iterative kernel correction. CoRR **abs/1904.03377** (2019), http://arxiv.org/abs/1904.03377
6. Haris, M., Shakhnarovich, G., Ukita, N.: Recurrent back-projection network for video super-resolution. CoRR **abs/1903.10128** (2019), http://arxiv.org/abs/1903.10128

7. Hu, X., Mu, H., Zhang, X., Wang, Z., Tan, T., Sun, J.: Meta-sr: A magnification-arbitrary network for super-resolution. CoRR **abs/1903.00875** (2019), http://arxiv.org/abs/1903.00875
8. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. CoRR **abs/1511.04587** (2015), http://arxiv.org/abs/1511.04587
9. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. CoRR **abs/1511.04491** (2015), http://arxiv.org/abs/1511.04491
10. Lai, W., Huang, J., Ahuja, N., Yang, M.: Deep laplacian pyramid networks for fast and accurate super-resolution. CoRR **abs/1704.03915** (2017), http://arxiv.org/abs/1704.03915
11. Ledig, C., Theis, L., Huszar, F., Caballero, J., Aitken, A.P., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. CoRR **abs/1609.04802** (2016), http://arxiv.org/abs/1609.04802
12. Li, S., He, F., Du, B., Zhang, L., Xu, Y., Tao, D.: Fast spatio-temporal residual network for video super-resolution. CoRR **abs/1904.02870** (2019), http://arxiv.org/abs/1904.02870
13. Li, Z., Yang, J., Liu, Z., Yang, X., Jeon, G., Wu, W.: Feedback network for image super-resolution. CoRR **abs/1903.09814** (2019), http://arxiv.org/abs/1903.09814
14. Ma, C., Yang, C., Yang, X., Yang, M.: Learning a no-reference quality metric for single-image super-resolution. CoRR **abs/1612.05890** (2016), http://arxiv.org/abs/1612.05890
15. Mao, X., Shen, C., Yang, Y.: Image restoration using convolutional auto-encoders with symmetric skip connections. CoRR **abs/1606.08921** (2016), http://arxiv.org/abs/1606.08921
16. Mittal, A., Soundararajan, R., Bovik, A.: Making a "completely blind" image quality analyzer. Signal Processing Letters, IEEE **20**, 209–212 (03 2013). https://doi.org/10.1109/LSP.2012.2227726
17. Tong, T., Li, G., Liu, X., Gao, Q.: Image super-resolution using dense skip connections. pp. 4809–4817 (10 2017). https://doi.org/10.1109/ICCV.2017.514
18. Wang, L., Wang, Y., Liang, Z., Lin, Z., Yang, J., An, W., Guo, Y.: Learning parallax attention for stereo image super-resolution. CoRR **abs/1903.05784** (2019), http://arxiv.org/abs/1903.05784
19. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Loy, C.C., Qiao, Y., Tang, X.: ESRGAN: enhanced super-resolution generative adversarial networks. CoRR **abs/1809.00219** (2018), http://arxiv.org/abs/1809.00219
20. Xintao Wang, Ke Yu, C.D., Loy, C.C.: Recovering realistic texture in image super-resolution by deep spatial feature transform. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
21. Yuan, N., Zhu, Z., Wu, X., Shen, L.: Mmsr: A multi-model super resolution framework. In: Tang, X., Chen, Q., Bose, P., Zheng, W., Gaudiot, J.L. (eds.) Network and Parallel Computing. pp. 197–208. Springer International Publishing, Cham (2019)
22. Zhang, K., Zuo, W., Zhang, L.: Deep plug-and-play super-resolution for arbitrary blur kernels. CoRR **abs/1903.12529** (2019), http://arxiv.org/abs/1903.12529
23. Zhang, S., Lin, Y., Sheng, H.: Residual networks for light field image super-resolution. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)