Theory and Practice of Science and Technology
2025, VOL. 6, NO. 9, 212-216
DOI: 10.47297/taposatWSP2633-456940.20250609

# Add Sharpening to the Super-Resolution Image Processing

Zhixuan Wang

School of Automation, Wuhan University of Technology, Wuhan, Hubei, 430070, China

**ABSTRACT**
With the extensive application of images in fields such as security surveillance, medical imaging, and satellite remote sensing, the issue of low resolution has become increasingly prominent. Super-resolution technology can reconstruct low-resolution images into high-resolution ones, thereby enhancing clarity and quality. However, existing methods still have shortcomings in detail restoration and edge enhancement. To address this, this paper introduces a sharpening module into the deep learning-based NinaSR super-resolution model to enhance edges and texture details. Experimental results show that this method effectively improves image sharpness and clarity while maintaining the overall structure, with a PSNR increase of approximately 0.038 dB compared to the original model. Additionally, it demonstrates a faster convergence speed in the early training stage, providing a new approach for high-resolution image reconstruction.

**KEYWORDS**
Image super-resolution; Deep learning; Sharpening processing; NinaSR model; Edge enhancement

## 1 Introduction

In today's digital age, images, as important carriers of information, are widely used in numerous fields such as security monitoring, medical imaging, satellite remote sensing, and high-definition display. However, due to the limitations of imaging equipment, losses during transmission, and the influence of environmental factors, the obtained images often have the problem of low resolution, resulting in blurred image details and information loss, which cannot meet the requirements of practical applications. Super-resolution image processing technology can convert low-resolution images into high-resolution ones through algorithms, thereby enhancing the clarity and quality of the images, and improving their visual effect and recognisability. Dong, C. et al. applied deep learning to image super-resolution processing, jointly optimizing all layers on the traditional SR method based on sparse coding, achieving significant improvements in both speed and quality. After conducting image processing using deep learning and SRCNN (Super-resolution Convolutional Neural Network) architecture, Yu Men et al. attempted the original implementation of image super-resolution using deep convolutional networks and provided detailed structures and codes. Based on the ideas of classic architectures such as SRCNN, this study selected the more lightweight and efficient NinaSR model as the basic framework.

Based on the image super-resolution technology, adding sharpening can not only enhance the resolution of the image, but also strengthen the edge and detail information of the image through the sharpening operation, making the image clearer and sharper, and further improving the quality and visual effect of the image. This has significant practical application value for fields that require high-resolution and high-definition images, such as medical diagnosis, security monitoring, remote sensing mapping, etc. Dian, R. et al. developed a deep HSI sharpening method (referred to as DHSIS) for fusing LR-HSI with HR-MSI. This method directly learns the image prior through residual learning based on deep convolutional neural networks and has made progress compared with predecessors in terms of reconstruction accuracy and running time.

Based on previous research, this paper selects the relatively mature super-resolution model NinaSR and adds a sharpening module on the basis of this model to enhance the edge and texture details of the image. Experimental verification shows that this method not only maintains the overall structure of the image but also enhances its clarity and sharpness, providing new ideas and solutions for high-resolution image processing.

## 2 Introduction to the Super-resolution Model NinaSR

### 2.1 General introduction of the model

The task of super-resolution is to reconstruct the corresponding high-resolution image based on the low-resolution image, and this process can be roughly formalized as a nonlinear mapping function $F_\theta$:

$$I_{SR} = F_\theta(I_{LR}),\ I_{SR} \approx I_{HR}$$

Among them, low-resolution images $I_{SR} \epsilon R^{H\times W\times C}$, reconstruct the corresponding high-resolution image $I_{HR} \epsilon R^{rH\times rW\times C}$, H and Wrepresent the height and width of the image respectively, C indicate the number of channels, r represent method

* **Corresponding Author:** Zhixuan Wang, Wang0818_zx@whut.edu.cn

multiples, θ represent the parameters of the model to be learned.

The NinaSR model is a lightweight super-resolution model proposed based on deep convolutional neural networks. Due to its lightweight and efficient characteristics, this study selects it as the basic super-resolution model. The core idea is to achieve the mapping from low-resolution images to high-resolution images by using residual learning and convolutional feature extraction while ensuring a relatively low computational cost.

The basic framework of NinaSR is divided into four parts: Head, Body, Upsample, and Tail.

## 2.2 Module introduction

### 2.2.1 Head module

The Head module is mainly a convolutional layer used to extract initial features from the input image:

$$F_0 = \sigma(W_{head} * I_{LR} + b_{head})$$

Among them, $W_{head}$ and $b_{head}$ is the weight and bias of the convolution kernel, $\sigma(\cdot)$ represents the activation function. In the code used in this article, the Head convolutional layer adopts a $3 \times 3$ convolutional kernel, with the number of channels set to 64, to ensure that effective features can be extracted without causing excessive computational complexity.

### 2.2.2 Body module

The Body module is composed of multiple residual blocks. The structure of a single residual block is:

$F_{res} = F_{in} + R(F_{in})$ Among them, the residual mapping function $R(\cdot)$ consists of two layers of convolution and a nonlinear activation function:

$$R(F_{in}) = W_2 * \sigma(W_1 * F_{in} + b_1) + b_2$$

Among them, $W_1$ and $W_2$ are usually $3 \times 3$ convolution kernels, and the number of channels remains at 64. When the Body contains N residual blocks, the overall expression is:

$$F_{body} = B^{(N)}(F_0)$$

Among them, $B^{(N)}(\cdot)$ represents a composite mapping of stacking N residual blocks.

### 2.2.3 Upsample module

The upsampling part adopts PixelShuffle. Let the magnification be s, then:

$$F_{up} = \text{PixelShuffle}_s(W_{up} * F_{body} + b_{up})$$

Among them, $W_{up}$ is usually a $3 \times 3$ convolution kernel, and the number of output channels is $C \cdot s^2$.

PixelShuffle rearranges the channel dimensions into spatial dimensions:

$$\text{PixelShuffle}_s: R^{H \times W \times (C \cdot s^2)} \rightarrow R^{(sH) \times (sW) \times C}$$

### 2.2.4 Tail module

The Tail module is a convolutional layer used to generate the final super-resolution image:

$$I_{SR} = W_{tail} * F_{up} + b_{tail}$$

Among them, $W_{tail}$ is a $3 \times 3$ convolutional kernel, and the number of output channels is the color channel C of the image (usually 3, that is, RGB).

# 3 Introduction to sharpening algorithms

In the task of image super-resolution, the model not only needs to restore the global structure of low-resolution images but also reconstruct high-frequency details. The original NinaSR model uses convolutional layers for initial feature extraction in the Head module. In this experiment, a Learnable Sharpen Layer is introduced before the Head module to enhance the edge information of the input image, thereby improving the model's ability to recover details.

## 3.1 Principle of sharpening algorithm

Sharpening operation essentially involves high-pass filtering of an image, emphasizing the rapidly changing parts (regions with larger gradients) in the image, that is, the edges.

Let the input image be:

$$I(x,y) \in R^{H \times W \times C}$$

Among them, H,W,C respectively represent the height, width, and number of channels of the image. Sharpening can be expressed in the following form:

$$I_{sharp} = I(x,y) + \lambda \cdot (I(x,y) - G_\sigma * I(x,y))$$

Among them , $G_\sigma * I(x,y)$ represents Gaussian blur applied to the image $I(x,y)$; $I(x,y) - G_\sigma * I(x,y)$ represents the extraction of high-frequency components; λ is the sharpening intensity coefficient, which is used to control the amplitude of enhancement.

In traditional sharpening algorithms, λ is a fixed parameter. In this experiment, λ is set as a learnable parameter. Compared with the traditional fixed-parameter sharpening, the learnable parameter λ enables the network to adaptively adjust the sharpening intensity for different images or even different regions of the same image, thereby having greater flexibility and performance potential.

### 3.2 Sharpening implementation

In this experiment, the code of the sharpening algorithm can be simplified as:

```
class LearnableSharpen(nn.Module):
def __init__(self, channels):
super().__init__()
#Initialize the sharpening intensity parameter α for each channel
self.weight = nn.Parameter(torch.ones(1, channels, 1, 1))
# Define the convolution kernel K (mean convolution, used for smoothing)
kernel = torch.ones((channels, 1, 3, 3)) / 9.0
self.register_buffer("kernel", kernel)
def forward(self, x):
# Low-pass filter K*I
smooth = F.conv2d(x, self.kernel, padding=1, groups=x.shape)
return x + self.weight * (x - smooth)
```

### 3.3 Overall implementation

The sharpening layer is placed before the Head module of NinaSR, which can enhance the high-frequency information in the input image and suppress the blurring caused by downsampling or compression.

The complete flowchart is:


Figure 1

## 4 Result presentation

### 4.1 Trend of PSNR changes

In the experiment, the training performance of the original NinaSR_b0 model and the improved model with a learnable sharpening layer was compared in the DIV2K dataset with a 2x super-resolution task. Peak signal-to-noise ratio (PSNR) is a commonly used indicator for evaluating the quality of super-resolution images. As shown in Table 1, with the increase of training rounds, the PSNR values of both models continue to rise and tend to converge. In the early stage of training, the improvement speed of PSNR was relatively fast. After about 30 rounds, the growth trend of PSNR slows down and stabilizes around 50 rounds. The PSNR values of the final two models were both around 31dB, which was at a relatively good level of image quality. Overall, the PSNR curve of the improved model throughout the training process is slightly higher than that of the original model, demonstrating a faster convergence speed and a higher stability value.

### 4.2 PSNR comparison in key rounds

The PSNR values of the two models under several key training rounds were compared and analyzed. The specific comparison is as follows:

(1)Round 10: The PSNR of the original model was 30.3528dB, and that of the improved model was 30.4445dB, an increase of approximately 0.092dB.

(2) Round 20: The PSNR of the original model was 31.0551 dB, and that of the improved model was 31.1274 dB, an increase of approximately 0.072 dB.

(3) Round 30: The PSNR of the original model was 31.4062 dB, and that of the improved model was 31.4123 dB, with an improvement of only approximately 0.006 dB.

(4) Round 40: The PSNR of the original model was 31.5795 dB, and that of the improved model was 31.5778 dB, with the two almost on par.

(5) Round 50: The PSNR of the original model was 31.6801 dB, and that of the improved model was 31.7177 dB, an increase of approximately 0.038 dB.

From the data of the above key rounds, it can be seen that the improved model is slightly higher than the original model in each key round, with the greatest difference in the 10th round and the subsequent differences gradually decreasing.

Overall, the introduction of learnable sharpening layers enables the model to gain more significant performance

advantages in the early stage of convergence and still maintain a slight improvement in the final results.

### 4.3 The influence of sharpening layers on edge recovery

The purpose of adding a learnable sharpening layer is to enhance the model's ability to restore image details and edges. The experimental results show that the improved model exhibits better reconstruction effects in areas rich in details, especially at the edges and textures. Although PSNR is a global metric, its improvement reflects the enhanced model's ability to capture high-frequency information. The learnable sharpening layer enhances high-frequency features, enabling the model to restore the edge contours of the original image more accurately and thereby improving the overall image quality. It was observed in the visualization reconstruction results that the image edges after using the sharpening layer were clearer and the contrast was higher. This indicates that this layer has a positive effect on edge recovery, verifying the rationality of introducing this mechanism.

### 4.4 PSNR comparison table

Table 1 lists the comparison of PSNR values of the two models in each training round (1 to 50 rounds).

Table 1 Comparison of PSNR values for each training round between the original model and the improved model

| round | Original model PSNR (dB | Improved Model PSNR (dB |
|---|---|---|
| 1 | 26.7351 | 26.7760 |
| 2 | 28.3732 | 28.4454 |
| 3 | 29.0863 | 29.1780 |
| 4 | 29.4554 | 29.5615 |
| 5 | 29.7047 | 29.7958 |
| 6 | 29.8899 | 29.9404 |
| 7 | 30.0389 | 30.1003 |
| 8 | 30.1461 | 30.2296 |
| 9 | 30.2586 | 30.3169 |
| 10 | 30.3528 | 30.4445 |
| 11 | 30.4538 | 30.4990 |
| 12 | 30.5160 | 30.6132 |
| 13 | 30.6030 | 30.6853 |
| 14 | 30.7028 | 30.7098 |
| 15 | 30.7484 | 30.8157 |
| 16 | 30.8316 | 30.8950 |
| 17 | 30.8816 | 30.9524 |
| 18 | 30.8807 | 31.0178 |
| 19 | 30.9789 | 31.0717 |
| 20 | 31.0551 | 31.1274 |
| 21 | 31.0834 | 31.1557 |
| round | Original model PSNR (dB | Improved Model PSNR (dB |
| 22 | 31.1440 | 31.2043 |
| 23 | 31.1832 | 31.2178 |
| 24 | 31.2234 | 31.2597 |
| 25 | 31.2534 | 31.3134 |
| 26 | 31.2908 | 31.3306 |
| 27 | 31.3101 | 31.3609 |
| 28 | 31.3204 | 31.3609 |
| 29 | 31.3651 | 31.3740 |
| 30 | 31.4062 | 31.4123 |
| 31 | 31.4275 | 31.4404 |
| 32 | 31.4422 | 31.4826 |
| 33 | 31.4653 | 31.4963 |
| 34 | 31.4668 | 31.5197 |
| 35 | 31.4979 | 31.5209 |
| 36 | 31.5185 | 31.5571 |
| 37 | 31.5313 | 31.5637 |
| 38 | 31.5527 | 31.5768 |
| 39 | 31.5637 | 31.5786 |
| 40 | 31.5795 | 31.5778 |
| 41 | 31.5818 | 31.6180 |
| 42 | 31.5866 | 31.6197 |
| 43 | 31.6103 | 31.6377 |
| 44 | 31.6291 | 31.6341 |
| 45 | 31.6434 | 31.6513 |
| 46 | 31.6480 | 31.6572 |
| 47 | 31.6586 | 31.6857 |
| 48 | 31.6707 | 31.6979 |
| 49 | 31.6720 | 31.7010 |
| 50 | 31.6801 | 31.7177 |

## 5 Summary

In this study, based on the lightweight super-resolution model NinaSR, we propose an improved method of introducing a learnable image sharpening Layer at the model input end. The main motivation of this method is to sharpen and enhance low-resolution images before feature extraction, thereby improving the edge and detail expression ability in the input signal and providing richer texture information for subsequent deep feature learning.

We conducted experiments on the improved NinaSR_b0 model on the DIV2K dataset. During the training process, 50 epochs were set, with a loss function of L1loss, and the model was learned on the bicubic downsampled data. We separately recorded the PSNR metrics during the training process of adding the sharpening module (ruihua) and the original model (original)

The results show that in the early stage of training, the PSNR convergence speed of the sharpened model is slightly faster than that of the original model, indicating that input enhancement helps the model learn effective features more quickly. During the middle to late training period (20 to 50 rounds), the PSNR of the sharpened model remained consistently higher than that of the original model, indicating that it performed better in edge detail recovery. Overall, the introduction of the sharpening module has improved the visual quality and reconstruction accuracy of the NinaSR model.

However, this study still has some deficiencies:

(1) Model complexity and speed impact: Although the computational load of the sharpening layer is relatively low, when deployed on mobile devices, additional convolution operations may still affect the inference speed and require further optimization.

(2) Fixed sharpening parameters: In the current experiment, the sharpening intensity α is set to a fixed value and is not adaptively adjusted according to the image content, which may lead to over-sharpening in scenes with high noise or weak textures.

(3) Single evaluation index: This experiment mainly uses PSNR for comparison and does not conduct a systematic assessment of perception indicators, thus failing to comprehensively reflect subjective visual quality.

The future research directions can be developed from the following aspects:

(1) Adaptive sharpening: Introduce learnable sharpening parameters or use an attention mechanism to enable the network to automatically adjust the sharpening intensity according to the characteristics of the image;

(2) Lightweight optimization: In response to the deployment requirements on mobile devices, methods such as convolution approximation, model quantization, or knowledge distillation can be explored to reduce computational overhead while maintaining performance improvement.

(3) More comprehensive assessment: In future experiments, indicators such as SSIM and LPIPS should be incorporated, and combined with subjective visual evaluation, a more comprehensive assessment system should be formed.

In conclusion, this study demonstrates that introducing a sharpening layer into the NinaSR model can effectively enhance the performance of super-resolution reconstruction, providing a new approach for the improvement of lightweight super-resolution algorithms.

## References

[1] Dong, C., Loy, C. C., He, K., & Tang, X. (2015). Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence, 38(2), 295-307.

[2] Men, Y., Luo, H., & Zheng, Q. (2021). Image super-resolution with convolutional neural networks. Preprint. ResearchGate.

[3] Dian, R., Li, S., Guo, A., & Fang, L. (2018). Deep hyperspectral image sharpening. IEEE transactions on neural networks and learning systems, 29 (11), 5345-5355.