# Postprint: Research on Image Super-Resolution Algorithms Based on Improved Convolutional Neural Networks

**Authors:** Hu Xiaohui, Zhang Jianguo

**Date:** 2019-01-28T00:00:00+00:00

## Abstract

To address issues such as overfitting in mapping functions and insufficient convergence of loss functions in existing convolutional neural network-based image super-resolution reconstruction algorithms, improvements are proposed through integration with existing visual recognition algorithms and deep learning theory. First, the number of layers in the original SRCNN network is increased from 3 to 13, and a self-gated activation function, Swish, is introduced to replace activation functions commonly used in previous network models such as Sigmoid and ReLU. By fully leveraging the advantages of the Swish function, the overfitting problem is effectively mitigated, enabling better learning and utilization of the mapping relationship from low-resolution to high-resolution images to guide image reconstruction. Subsequently, the theory of the Newton-Raphson iteration method is incorporated into the traditional network loss function, further accelerating convergence speed. Finally, experiments demonstrate that the improved convolutional neural network model can effectively enhance image clarity, achieving further improvements in both subjective visual quality and objective parameter evaluation metrics.

## Full Text

## Preamble

**Vol. 37 No. 4**
**Application Research of Computers**
**ChinaXiv Cooperative Journal**

# Research on Image Super-Resolution Algorithm Based on Improved Convolutional Neural Network

**Hu Xiaohui, Zhang Jianguo**
(School of Electronics & Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China)

**Abstract:** Existing convolutional neural network (CNN) based image super-resolution restoration algorithms suffer from overfitting of mapping functions and insufficient convergence of loss functions. To address these issues, this paper proposes improvements by integrating contemporary visual recognition algorithms and deep learning theory. First, the original SRCNN network depth is increased from 3 to 13 layers, and a self-gated activation function called Swish is introduced to replace commonly used activation functions such as Sigmoid and ReLU. By leveraging the advantages of Swish, the proposed method effectively avoids overfitting and better learns the mapping relationship between low-resolution and high-resolution images to guide image reconstruction. Second, the Newton-Raphson iteration method is incorporated into the traditional network loss function to further accelerate convergence speed. Experimental results demonstrate that the improved CNN model can effectively enhance image clarity, achieving superior performance in both subjective visual quality and objective evaluation metrics.

**Keywords:** low resolution; super-resolution; convolutional neural network; image processing; restoration

## 0 Introduction

Image super-resolution (SR) restoration technology was first proposed in the 1960s, initially referred to as spectral extrapolation, though it did not gain widespread acceptance at the time. In 1984, Tsai et al. [1] introduced a method to reconstruct single-frame high-resolution images from low-resolution inputs using signal processing techniques to recover high-frequency information beyond the imaging system' s cutoff frequency. This approach enabled high-frequency information recovery during the imaging process and achieved high-resolution image reconstruction, sparking significant research interest that continues to this day [18–21].

Current research in image super-resolution reconstruction primarily employs three approaches: reconstruction-based methods, interpolation-based methods, and learning-based methods [2]. Reconstruction-based techniques use single or multiple low-resolution (LR) images to reconstruct high-resolution (HR) images by establishing an observation model and solving its inverse problem. However, this approach is ill-posed and yields non-unique solutions, as any given LR image may correspond to multiple possible HR reconstructions [2]. Interpolation methods, such as the classic bilinear and bicubic interpolation, directly utilize

image prior information through mathematical modeling. While effective for certain images, these methods produce unsatisfactory results for most images, particularly at higher magnification factors.

Due to the limitations of reconstruction and interpolation methods, researchers have increasingly focused on learning-based approaches. In recent years, with the rapid development of deep learning, various intelligent algorithms and neural network models have been widely applied to image super-resolution reconstruction. These methods leverage image databases or the images themselves to build models, design learning strategies, and train model capabilities to actively learn the correlation between HR and LR images, ultimately generating HR images through specialized functional layers. For instance, Chan et al. [3] introduced locally linear embedding from manifold learning to compute weighted averages for reconstruction, assuming geometric similarity between HR and LR information. Yang et al. [4] applied sparse coding theory [5, 17] to establish complete dictionaries mapping LR to HR images, using prior constraints to achieve sparsity before final SR reconstruction. Chang et al. [6] proposed a sparse coding network approach that integrates sparse representation, mapping, and reconstruction modules into a unified framework for collaborative optimization. Wang et al. [7] developed an SR method using a learned iterative shrinkage and thresholding algorithm to create a feedforward neural network (SCN) for sparse coding and decoding, achieving improved PSNR at higher magnification factors with enhanced computational efficiency.

Gu et al. [8] applied convolutional sparse coding to SR reconstruction, demonstrating that LR and HR filter learning provides valuable guidance for designing deep network filter banks while preserving spatial information and improving reconstruction quality. Dong et al. [9] first introduced convolutional neural networks (CNN) to super-resolution reconstruction with SRCNN, which consists of three convolutional layers for patch extraction, non-linear mapping, and final reconstruction, enabling end-to-end learning from LR to HR images based on sparse coding and feature learning theory. However, SRCNN proved difficult to train, highly sensitive to hyperparameter changes, and limited by its shallow three-layer architecture. Yang et al. [10] proposed the Deep Edge Guided Recurrent Residual Network (DEGREE), which decomposes image signals into different frequency bands for reconstruction before combining them to preserve important detail information, partially addressing SRCNN' s underutilization of image priors and detail loss issues.

While CNN-based image super-resolution algorithms have achieved significant improvements [12], further enhancements remain possible. Building upon Dong et al.' s foundational work [8], this paper proposes improvements to the original SRCNN architecture.

---

# 1 Methodology

## 1.1 Model Architecture

This work employs a 13-layer convolutional neural network [11] with Swish as the activation function. The first layer preprocesses input images using bicubic interpolation to obtain the desired LR representation. Leveraging Swish' s compatibility with local response normalization, layers 2 through 11 extract features from LR images to produce LR feature patches. Layers 12 and 13 learn the feature mapping relationship between LR and HR patches to generate HR patches, which are finally combined through overlapping and averaging to produce the reconstructed HR image. The network structure is illustrated in [Figure 1: see original paper].

## 1.2 Feature Extraction and Convolution Operations

Classic neural networks traditionally used the Sigmoid activation function, derived from neuroscience theory to model neuronal excitation upon reaching certain thresholds. Despite its strong theoretical interpretability, Sigmoid is rarely used in practice because it causes network saturation and vanishing gradients. When neuronal activations saturate at 0 or 1, gradients vanish, preventing signal propagation through the neuron. Additionally, Sigmoid outputs are not zero-centered, causing asymmetric information to propagate to higher layers and introducing oscillations during gradient descent.

In recent deep learning research, ReLU [13] has replaced Sigmoid as the preferred activation function due to its computational efficiency and stable gradient behavior. The ReLU function, expressed as $f(x) = \max(0, x)$, produces gradients of either 0 or 1, maintaining stable gradient magnitudes across varying network depths. As shown in [Figure 2: see original paper], ReLU enables significantly faster SGD convergence compared to Sigmoid, requiring only thresholding rather than complex computations. However, when extremely large gradients flow through a ReLU neuron, parameter updates may permanently deactivate the neuron, resulting in "dead" neurons with zero gradients that never reactivate during training.

To address ReLU' s limitations, this paper proposes the self-gated activation function Swish, illustrated in [Figure 3: see original paper]. The mathematical formulation is given by:

$$Swish(x) = x \cdot \sigma(\beta x) = \frac{x}{1 + e^{-\beta x}}$$

where $W_1$ represents convolutional kernels, $B_1$ denotes bias vectors, "*"indicates convolution operations, and $W_1$ has dimensions $c \times f_1 \times f_1$ with $c$ being the number of input channels and $\beta$ a trainable parameter.

Unlike ReLU, which has zero gradient for $x < 0$, Swish represents both positive and negative values, offering sparse, non-monotonic, and smooth characteristics

that reduce neuron deactivation from large gradient flows. For $x < 0$, Swish gradients approach zero asymptotically but never equal zero, preventing neuronal death. As a self-gating mechanism, Swish requires only a single scalar input compared to conventional gating functions needing multiple scalars, allowing seamless replacement of ReLU without adjusting hidden capacity or parameter counts. Experimental results confirm that Swish achieves expected performance improvements at scale.

The mean squared error (MSE) loss function is employed for training, defined as:

$$Loss = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2$$

where $y_i$ represents the ground truth value and $y_i'$ the CNN prediction for the $i$-th data point. MSE effectively measures average error magnitude, with smaller values indicating better model accuracy. As shown in [Figure 4: see original paper], MSE loss with Swish activation demonstrates favorable prediction performance.

### 1.3 Nonlinear Mapping

In the overall model [14], the first layer extracts features from each LR image patch. Layers 2 through 11 perform nonlinear mapping, expressed mathematically as:

$$F_2(x) = \max(0, W_2 * F_1(x) + B_2)$$

where $W_2$ denotes convolutional kernels and $B_2$ represents bias vectors. If $n_1$-dimensional vectors undergo nonlinear mapping, they produce $n_2$-dimensional vectors, transforming LR feature patches into HR patches through end-to-end learning. With $N$ convolutional kernels, the process generates $N$ high-resolution image patches for reconstruction.

### 1.4 Image Reconstruction

In the reconstruction stage, $W_3$ functions as an averaging filter and $B_3$ is a bias vector with dimension $c$. Following nonlinear feature mapping, HR image patches are predicted and overlapped, with the final HR image obtained through averaging. This process can be implemented as a convolutional layer generating the final high-resolution image, where the averaging method effectively acts as a predefined filter on feature maps.

Minimizing squared error is typically solved via gradient descent, which finds the steepest descent direction to optimize network parameters. Two widely used variants are batch gradient descent [15] and stochastic gradient descent (SGD)

[16]. Batch gradient descent updates parameters using the entire dataset, as expressed in equations (7) and (8), where $\alpha$ denotes the step size and $X = \{x_1, x_2, ..., x_n\}$ represents the training set with $m$ samples per class. This approach offers higher computational accuracy but suffers from slow training speed for large datasets.

### 1.5 Training and Optimization

Traditional fully connected neural networks generate enormous parameter counts, making training extremely time-consuming or even intractable. CNNs address this challenge by combining local receptive fields, weight sharing, and spatial/temporal subsampling to achieve translation, scale, and deformation invariance. A typical CNN architecture follows: Input → Convolution → Activation → Convolution → Activation → Pooling → Activation → Convolution → Activation → Pooling → Fully Connected.

CNN learning algorithms reconstruct errors while continuously adjusting connection weights and biases through forward and backward propagation to optimize network parameters. Forward propagation transfers feature information without fundamental differences from standard neural networks. For layer $l$ with output $x_l$, weights $w_l$, and biases $b_l$, forward propagation is expressed as:

$$x_l = f(u_l) = f(w_l \cdot x_{l-1} + b_l)$$

where $f$ represents the activation function, for which this work adopts Swish.

Backward propagation corrects model parameters through error information. During backpropagation, errors from reduced regions must be upsampled to reconstruct larger region errors from previous layers. Common CNN loss functions include MSE and cross-entropy. The objective is to compute residuals for each unit, with sub-functions deriving residuals for corresponding output units. This work employs the MSE loss function shown in equation (6).

The above formulation describes the training error for sample $n$. While SGD updates parameters using a single sample per iteration, reducing training time, it may converge to suboptimal local minima rather than global optima due to high gradient variance. Batch gradient descent, conversely, uses all samples for stable convergence but suffers from slow training speed. Random gradient descent' s single-sample iterations cause erratic direction changes, preventing rapid convergence to local optima. Selecting an appropriate gradient computation method is crucial for improving network learning rates [16].

To address these limitations, this paper introduces the Newton-Raphson iteration method. As a second-order method, Newton-Raphson converges faster than first-order gradient descent. Geometrically, Newton-Raphson fits a quadratic surface to the local neighborhood, whereas gradient descent uses a planar approximation. The quadratic surface typically provides superior fitting, yielding

a descent path closer to the true optimal trajectory, as illustrated in [Figure 5: see original paper]. Path A represents Newton-Raphson' s iteration trajectory, while path B shows gradient descent' s trajectory.

Newton-Raphson exhibits stronger global judgment capabilities compared to gradient descent. While gradient descent proceeds locally from the initial point in a zigzag pattern toward extrema, Newton-Raphson directly searches for optimal paths considering function convexity. The method evaluates not only current slope magnitude but also whether the slope will increase after a step. In terms of convergence speed, gradient descent exhibits linear convergence, whereas Newton-Raphson demonstrates superlinear convergence of at least second order.

Newton-Raphson addresses batch gradient descent' s slow training and SGD' s insufficient accuracy, though it retains some limitations for different CNN architectures requiring hyperparameter tuning through experimentation. Nevertheless, it provides substantial improvements over traditional gradient methods (the original SRCNN used SGD, as shown in [Figure 6: see original paper]), accelerating convergence, reducing oscillations, improving learning rates, and shortening training time.

---

## 2 Experimental Results and Analysis

Experiments were conducted on an Intel CoreTM i7-8500Y CPU @ 4.20 GHz with NVIDIA GTX1070Ti GPU [15] and 12 GB RAM, running 64-bit Ubuntu 18.04.1 LTS, MATLAB R2016b, CUDA 10.0, and OpenCV 3.2.0. The proposed method was compared against classical bicubic interpolation (BI), Deep Edge Guided Recurrent Residual Network (DEGREE) [10], SRCNN, and the improved SRCNN algorithm. Training utilized 90 images from the ImageNet database, widely adopted in single-image super-resolution evaluations.

Performance was assessed using Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). PSNR quantitatively measures error between processed and original images, with higher values indicating less distortion. SSIM values approaching 1 signify high structural similarity to the original image, representing superior reconstruction quality.

The PSNR is computed as:

$$PSNR = 10 \log_{10} \left( \frac{(2^n - 1)^2}{MSE} \right)$$

where $H$ and $W$ represent image height and width, $n$ denotes bits per pixel (typically 8), and MSE is the mean squared error.

The SSIM expression is:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

where $\mu_x$ and $\mu_y$ are mean values, $\sigma_x^2$ and $\sigma_y^2$ are variances, $\sigma_{xy}$ is covariance, $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ are stability constants with $L$ as the pixel dynamic range and $k_1 = 0.01$, $k_2 = 0.03$ as default values.

Experiments comparing Swish activation with Newton-Raphson-based error functions against batch gradient descent on the Set14 test set demonstrate significant improvements, with average PSNR curves shown in [Figure 7: see original paper]. To isolate the contributions of each component, separate experiments evaluated Swish activation alone and Newton-Raphson iteration alone against comparative algorithms on Set14. Results in [Figure 8: see original paper] and [Figure 9: see original paper] confirm that both Swish and Newton-Raphson individually improve performance, with combined usage yielding optimal results as evidenced in [Figure 7: see original paper].

Beyond PSNR improvements, additional validation on Set5 and Set14 test sets further demonstrates the enhanced network' s effectiveness over SRCNN and DEGREE, with quantitative results presented in .

Visual comparisons in [Figure 10: see original paper]-[Figure 13: see original paper] show processing results for four test images using four algorithms. DEGREE, SRCNN, and the proposed method all outperform traditional interpolation, confirming the advantages of learning-based reconstruction. The proposed algorithm exhibits superior edge preservation and detail clarity, particularly in magnified regions of baboon, pepper, Lenna, and bird images. While BI produces blurry results with unclear edges, and SRCNN/DEGREE show modest improvements, the proposed method captures finer high-frequency information, yielding sharper edges, more complete preservation, and significantly enhanced image acuity.

---

## 3 Conclusion

This paper theoretically analyzed CNN learning and training processes, identifying limitations of conventional mapping functions (ReLU) and gradient descent methods in image reconstruction. Through empirical research, novel mapping functions (Swish) and loss functions were proposed and validated on standard benchmark datasets Set5 and Set14. Experimental results demonstrate significant improvements over existing algorithms, achieving excellent performance in both subjective visual evaluation and objective quantitative metrics.

---

## References

[1] Sun Xu, Li Xiaoguang, Li Jiafeng, et al. Advances in image super-resolution restoration based on in-depth learning [J]. Journal of Automation, 2017, 43(5): 697-709.

[2] Su Heng, Zhou Jie, Zhang Zhihao. Overview of super-resolution image reconstruction methods [J]. Journal of Automation, 2013, 39(8): 1201-1213.

[3] Chan Takming, Zhang Junping, Pu Jian, et al. Neighbor embedding based super-resolution algorithm through edge detection and feature selection [J]. Pattern Recognition Letters, 2009, 30(5): 494—502.

[4] Yang Jianchao, Wright J, Huang T S, et al. Image super-resolution via sparse representation [C]. IEEE Trans on Image Processing, 2010, 19(11): 2861—2873.

[5] Pan Zongxu, Yu Jing, Hu Shaoxing et al. Single image super-resolution algorithm based on self-similarity of multi-scale structure [J]. Journal of Automation, 2014, 40(4): 594—603.

[6] Chang Hong, Yeung D Y, Xiong Yimin. Super-resolution through neighbor embedding [C]//Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2004: 275-282.

[7] Wang Zhaowen, Liu Ding, Yang Jianchao, et al. Deep networks for image super-resolution with sparse prior [C]//Proc of IEEE International Conference on Computer Vision. Washington DC: IEEE Computer Society, 2015: 370-378.

[8] Gu Shuhang, Zuo Wangmeng, Xie Qi, et al. Convolutional sparse coding for image super-resolution [C]//Proc of IEEE International Conference on Computer Vision. Washington DC: IEEE Computer Society, 2015: 1823-1831.

[9] Dong Chao, Chen C L, He Kaiming, et al. Image super-resolution using deep convolutional networks [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2016, 38(2): 295-307.

[10] Yang Wenhan, Feng Jiashi, Yang Jianchao, et al. Deep edge guided recurrent residual learning for image super-resolution [EB/OL]. (2016-07-18). https://arxiv.org/pdf/1604.08671.pdf.

[11] Karen Simonyan, Andrew Zisserman. Very deep convolutional networks for large-scale image recognition [EB/OL]. (2015-04-10). https://arxiv.org/abs/1409.1556.

[12] Wang Xiaoming, Huang Feng, Liu Shaopeng, et al. Improved self-learning super-resolution reconstruction method for single image [J/OL]. Application Research of Computers, 2019, 36(9). http://www.arocmag.com/article/02-2019-09-052.html.

[13] Xu Bing, Wang Naiyan, Chen Tianqi, et al. Empirical evaluation of rectified activations in convolutional network [EB/OL]. (2015-11-27). https://arxiv.org/abs/1505.00853.

[14] Nair V, Hinton G E. Rectified linear units improve restricted Boltzmann machines [C]//Proc of the 27th International Conference on International Conference on Machine Learning. 2010: 807-814.

[15] Liu Cun, Li Yuanxiang, Zhou Yongjun, et al. Video image super-resolution reconstruction method based on convolution neural network [J/OL]. Application Research of Computers, 2019, 36(4). http://www.arocmag.com/article/02-2019-04-057.html.

[16] Xiao Jinsheng, Liu Enyu, Zhu Li, et al. Improved image super-resolution algorithm based on convolution neural network [J]. Journal of Optoelectronics, 2017, 37(3): 0318011.

[17] Li Min, Cheng Jian, Le Xiang, et al. Super-resolution reconstruction of sparse dictionary coding [J]. Journal of Software, 2012, 23(5): 1315-1324.

[18] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification [C]//Proc of IEEE International Conference on Computer Vision. Washington DC: IEEE Computer Society, 2015: 1026-1034.

[19] Timofte R, Agustsson E, Van Gool L, et al. Ntire 2017 challenge on single image super-resolution: Methods and results [C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2017: 1110-1121.

[20] Liao Renjie, Tao Xin, Li Ruiyu, et al. Video super-resolution via deep draft-ensemble learning [C]//Proc of IEEE International Conference on Computer Vision. Washington DC: IEEE Computer Society, 2015.

[21] Du Bo, Xiong Wei, Wu Jia, et al. Stacked convolutional denoising auto-encoders for feature representation [J]. IEEE Trans on Cybernetics, 2017, 47(4): 1017-1027.

*Note: Figure translations are in progress. See original paper for figures.*