ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20250204002330
Original scientific paper

HFDDLap: High-Low Frequency Differentiation Dynamic Laplacian Pyramid Network for Image Super-Resolution

Peituan LIU, Tie LI*, Rui LI, Bo SONG

Abstract: Existing single image super-resolution (SR) algorithms based on convolution neural networks (CNNs) have achieved usable visual results. However, they often encounter artifacts and blurring, particularly at large scaling factors (e.g., 4×, 8×), due to significant loss of high-frequency information. To address these challenges, we propose a novel high-low frequency differentiation dynamic Laplacian pyramid network (HFDDLap). Our approach introduces a learnable high-low frequency differentiation convolution (HLC) within high-low frequency differentiation residual channel attention blocks (HL-RCAB) to effectively capture and differentiate high- and low-frequency components, enhancing detail preservation. Additionally, we employ a dynamic deconvolution (DDC) that adaptively generates upsampling kernels based on input features, improving reconstruction accuracy by reducing feature distortion. Extensive experiments on 4× and 8× SR demonstrate that our proposed method effectively reconstructs edge details, produces satisfactory SR results and outperforms some state-of-the-art (SOTA) methods in terms of evaluation metrics.

Keywords: dynamic deconvolution; Laplacian pyramid network; nonlinear mapping; single image super-resolution

1 INTRODUCTION

Single image super-resolution (SISR) is a crucial task in image restoration, aiming to reconstruct high-resolution (HR) images from low-resolution (LR) inputs. There are many practical applications that need SISR, such as public security [1] and remote sensing [2]. However, solving the super-resolution (SR) problem remains challenging because it is an ill-posed problem: an infinite number of different HR results can be reconstructed from a single LR image. Early research efforts on SR algorithms using interpolation [3-5] and learning-based methods [6-8] provided a solid theoretical foundation, but these earlier methods are less effective in processing for images with complex structures. The advent of convolutional neural networks (CNNs) brought new possibilities to SR algorithms, with Dong et al. [9] pioneering proposing a Super-Resolution Convolutional Neural Network (SRCNN) with three convolutional layers learning a nonlinear mapping from LR to HR images. Since then, a series of CNN-based approaches [10-24] have been developed, surpassing traditional mathematical and theoretical methods in terms of performance. To overcome challenges such as gradient disappearance or explosion while enhancing the reconstruction performance of SR networks, techniques such as residual learning [14, 19], dense learning [15, 25], recursive learning [10, 22], multi-path learning [26, 27], and attention mechanisms [15, 33] have been incorporated to improve them. Additional network architectures like generative adversarial networks (GANs) [28-32], graph neural networks (GNNs) [33], binarized networks [34] , AdderNet [35], and transformer [67, 68] have been developed to cater to specific requirements.

However, the increasing demand for higher image resolution has led to a growing trend of tackling the challenge of comparatively large scaling-factor (e.g., 4×, 8×) SISR. While several impressive methods [21, 53, 60-63] have achieved remarkable results with large-factor SR, these methods still have two limitations.

First, in large-factor SR, the detailed information in the LR images is insufficient to reconstruct an HR result with rich texture information. To enhance the learning of the nonlinear mapping from LR to HR images, an increasing number of studies use the gradient as a priori information to improve reconstruction outcomes. For instance, Ma et al. [32] extracted the gradient from original LR images using a fixed edge detection operator (e.g., Sobel [36], Canny [37], or Prewitt [38]) and directly supplied it to the network as prior information. Cai et al. [26] extracted the gradient from a HR image by subtracting the HR image blurred with a Gaussian kernel and treating the gradient as an extra input.

Although these methods demonstrate that introducing gradient information into the network yields more explicit texture details, they also introduce visual artifacts due to pixel shifts when the gradient branches are fused with the main network branch. Moreover, as the scaling-factor increases, the fixed gradient extraction operator cannot adaptively handle finer textures, leading to a loss of information.

Second, the key to image SR is establishing the correlation between pixels in the LR and HR spaces. However, as the scale factor increases, the SR results may lead to greater reconstruction errors. As shown in Fig. 1, there are infinite up-sampling solutions from LR to HR. Existing methods tend to learn an average solution for all images, which yields acceptable results for small scaling-factors. However, with large-factor SR tasks, the size of the HR space increases exponentially while the size of the LR space is minimal. Consequently, an average solution amplifies the reconstruction error. To alleviate this problem, it becomes crucial to design a suitable up-sampling operator for large-factor SR.

In this work, we propose a high-low frequency differentiation dynamic Laplacian pyramid network (HFDDLap) to overcome the limitations of existing approaches. To address the first problem of losing information, we introduce a high frequency differentiation convolution (HC). This module functions as a learnable Prewitt [38] operator, dynamically adjusting its filtering weights based on the input image's texture features. As the scaling factor increases, the HC module automatically focuses on finer high-frequency structures, effectively preventing detail loss. Specifically, we design two methods to extract high frequency information by differentiating the

pixels in the sliding window horizontally and vertically and then convolving the differenced image with kernel weights. We then incorporate low frequency differentiation convolution (LC) to complement the low frequency information missing from HC. The combination of LC and HC forms the proposed high-low frequency differentiation convolution (HLC).

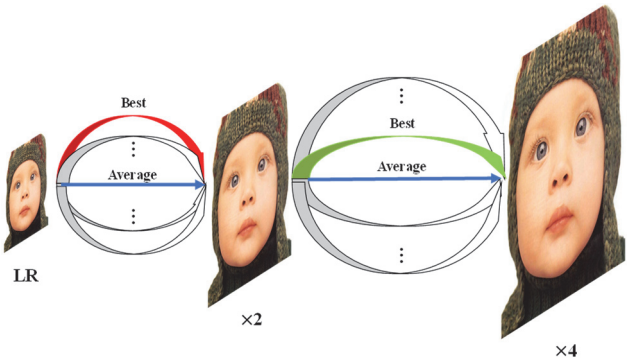

Figure 1 The mapping process from low-resolution (LR) space to high-resolution (HR) space involves infinite pathways. These pathways grow exponentially as the scaling-factor increases

Addressing the second limitation of error amplification, we introduce a dynamic convolution (DC) [39] to our approach, enhancing it into a dynamic deconvolution (DDC) for the up-sampling operation. The DDC module adaptively generates upsampling kernels conditioned on the input features, allowing the model to tailor the upsampling process to the specific content of each image region. Specifically, multiple additional experts are added to the deconvolutional layer, multiplied by routing weights computed from the input features, and fused into a new kernel for deconvolution. Inspired by Laplacian pyramid framework [40], we also employ the Laplacian pyramid as the network structure, taking advantage of the pyramid's subtle feature differences at different scaling-factors and enabling the network to achieve optimal performance with multiple scaling factors.

We evaluate our HFDDLap network using five datasets: Set5 [41], Set14 [42], BSD100 [43], Urban100 [44], and Manga109 [45]. The results demonstrate that HFDDLap offers good visual quality and outperforms state-of-the-art SR methods on both objective and subjective measures.

The main contributions of this work can be summarized as follows.

1) We design a novel HFDDLap, based on a Laplacian Pyramid structure to cope with the lack of texture and detail retention in large-factor SR operations.

2) We develop a creative HLC for integrating gradient information generated through pixel differentiation into the network to enhance textures and details of SR images.

3) We introduce an innovative DDC for dynamically extracting features, decreasing reconstruction error by mining the potential features within the image and activating the appropriate expert for up-sampling.

4) We conduct extensive experiments on large-factor SR images and demonstrate that HFDDLap outperforms existing general purpose SR methods and even surpasses some methods specifically designed for handling large-scale SR.

The remainder of this paper is organized as follows. In Section 2, we provide a review of related work on edge detectors and dynamic convolution theory. Section 3 presents a detailed description of the HFDDLap network. Section 4 presents our extensive experiments and corresponding analyses. Section 5 presents our conclusions.

2 RELATED WORKS

2.1 CNN-Based Single Image Super-Resolution (SISR) Models

Since SRCNN [9] was proposed by Dong et al., a plethora number of CNN-based SISR methods [10-24] have been proposed in recent years. The compelling performance of CNNs has revitalized the SR field, and their exceptional visual results increasingly set these methods apart from traditional ones. For instance, Kim et al. augmented the number of CNN layers, implemented the residual structure in the SR network, and proposed very deep convolutional networks (VDSR) [19]. Recursive structures are integrated into the deep recursive residual network (DRRN) [10] and deeply-recursive convolutional network (DRCN) [22] to enhance features. In contrast to methods that up-sample LR images as the input of the model, the fast super-resolution convolutional neural network (FSRCNN) [52] places deconvolution at the end of the network for up-sampling, significantly boosting the operation speed of the algorithm. This post-up-sampling approach has become the norm, inspiring numerous variant networks. For example, Lim et al. further deepened the network depth by stacking 32 residual blocks and proposed the enhanced deep residual networks (EDSR) [53]. However, these post-up-sampling approaches continue to grapple with significant detail loss when tackling large-factor SR. To address this, Lai et al. introduced deep Laplacian pyramid networks (LapSRN) [21] and later advanced the use of this structure to MS-LapSRN [54] for better performance. Although the Laplace pyramid structure has proven beneficial for large-factor SR, it remains extremely challenging.

In the realm of SR methods, a prevalent challenge is the loss of high-frequency details and the introduction of artifacts, especially when dealing with large scaling factors. No matter how these existing approaches alter the up-sampling operation's position, they typically seek an average solution instead of finding an optimal solution for each image by fully considering its characteristics. This average solution causes serious errors when dealing with large-factor SR problems. Therefore, we attempt to integrate edge detection and dynamic convolution theory in the design of HFDDLap to address these challenges.

2.2 Edge Detector

Fixed operators were the primary methods for edge detection for decades. Thanks to the robust data representation capability of CNNs, learnable operators that provide more accurate results have recently emerged. Both central difference convolution (CDC) [46, 47] and pixel difference convolution (PDC) [48] employ the relationship between neighboring pixels to compute pixel differences and subsequently convolve with the learnable kernel to extract gradient information. Specifically, CDC can be considered a particular case of PDC, wherein edges are

detected by differences between the center pixel and its neighboring pixels. These methods are all derivatives of local binary patterns (LBP) [49]. Similar concepts are also applied in extracting high frequency information from feature maps by our proposed HLC. However, HLC requires further refinement for SR task, because the SR result is a complete image rather than just the image edge.

2.3 Dynamic Convolution & Conditionally Parameterized Convolution

DC draws inspiration from the recent works of dynamic neural networks (DY-NNs) that select a part of the existing model and omit the rest based on the input image. In other words, during the inference stage, a multitude of sub-networks are dynamically generated according to the input image to optimize performance. Unlike DY-NNs, DC enables the model to dynamically select the most suitable kernel based on the input image or feature by directly adding kernels to the same convolutional layer. This motivation aligns closely with the idea of conditionally parameterized convolution (CondConv) [50], which learns specialized convolutional kernels for each example. The distinction between the two methods lies in the normalization function used when calculating the routing weight, with DC regarded as a unique form of conditional convolution.

3 PROPOSED METHOD

In this section, we introduce our proposed HFDDLap in detail, focusing on the design motivations of the network architecture as shown in Fig. 2. In addition, we also introduce the theory behind two main components of HFDDLap, HLC and DDC, explaining the difference between HLC, vanilla convolution, and pixel difference convolution; and the advantages of DCC compared to vanilla deconvolution.

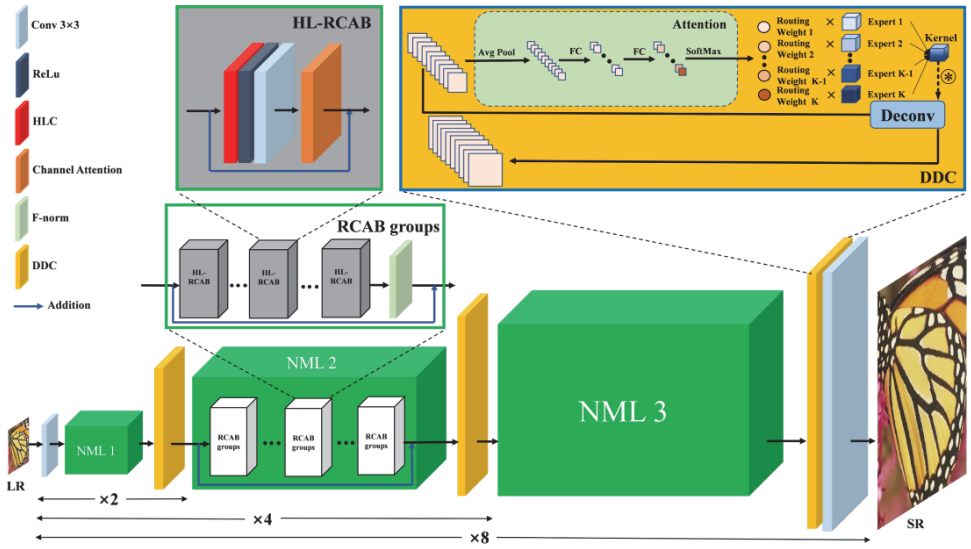

Figure 2 The network structure of the proposed HFDDLap. The network follows the Laplace pyramid structure and consists of three levels dedicated to ×2 SR, ×4 SR and×8 SR, respectively. Each level shares the same structure, which includes a nonlinear mapping layer (NML) comprised of stacked residual channel attention blocks (RCAB) groups and a dynamic deconvolution (DDC) component. The HL-RCAB is formed by replacing the first convolution in RCAB with high-low frequency differentiation convolution (HLC). In the attention block of DDC, AvgPool, FC, and Soft Max refer to average pooling, fully connection layers, and Soft Max function, respectively.

3.1 High-Low Frequency Differentiation Dynamic Laplacian Pyramid Network (HFDDLap)

HFDDLap is designed around a Laplacian pyramid framework, which has proven to be an effective strategy for tackling the large-factor SR [61]. Additionally, we employ the progressive network structure within HFDDLap, dividing the intricate up-sampling into multiple sub-processes. For instance, three levels of up-sampling are designed to manage 2×, 4× and 8× SR, respectively, with each sub-process exclusively addressing 2× SR.

In the first level, the LR image enters the network. After shallow feature extraction through a convolution layer, it proceeds through a nonlinear mapping layer (NML) comprised of several stacked residual channel attention blocks (RCAB) groups. Multiple high-low frequency differentiation residual channel attention blocks (HL-RCABs) are stacked with an F-norm [17] layer to normalize the output features in each RCAB group. Then mapped features are fed into our proposed DDC to execute the up-sampling operation. Similarly, the input for the subsequent levels consists of the output features from the previous level rather than the LR image, with a convolutional layer placed after the DDC in the final level for reconstructing the SR image from features.

The model can be mathematically formulated as,

$$X_l = \begin{cases} \left(f_M\left(f_E\left(I_{\mathrm{LR}}\right)+f_E\left(I_{\mathrm{LR}}\right)\right)\right)\uparrow, & l=1, \\ \left(f_M\left(X_{l-1}\right)+X_{l-1}\right)\uparrow, & 1<l\le L, \end{cases} \tag{1}$$

$$I_{\mathrm{SR}} = f_R\left(X_L\right) \tag{2}$$

where X_l denotes the output features at the l-th level, and L signifies the number of levels. I_{LR} denotes the LR input; $f_E(\cdot)$ and $f_M(\cdot)$ represent the feature extractor and nonlinear mapping function, respectively; and $\uparrow$ denotes the DDC for the up-sampling operation. X_L is fed into reconstruction function $f_R(\cdot)$ to produce the SR result I_{SR}.

Rather than relying on a complex loss function, HFDDLap minimizes the Manhattan distance (the L_1-norm) between the HR and LR image pairs. The training objective function can be written as,

$$\mathcal{L}_D\left(I^{\mathrm{LR}}, I^{\mathrm{HR}}; \theta\right)=\frac{1}{N}\sum_{i=1}^{N}F\left(I_{\mathrm{LR}}^{i}\right)-I_{\mathrm{HR}_1}^{i}\mathrm{x} \tag{3}$$

where θ signifies the HFDDLap parameters; N represents the number of image pairs; I_{LR}^{i} and I_{HR}^{i} are the image pairs from the given training set $\left\{I_{\mathrm{LR}}^{i}, I_{\mathrm{HR}}^{i}\right\}_{i=1}^{N}$; and $F(\cdot)$ denotes the HFDDLap. More training details will be presented in section 4.

3.2 High-Low Frequency Differentiation Convolution (HLC)

Texture information within an image is the most noticeable characteristic when evaluating the visual results of the image. In essence, the content within an image is most easily discerned via sharp edge details, and a key criterion for image restoration is minimizing the loss of texture information. Moreover, the edge structure of the reconstructed image is a significant measure of the performance of the SR model. Hence, metrics like peak signal-to-noise ratio (PSNR) and structural similarity (SSIM), used to assess image quality similarity and overall structural resemblance, are extensively employed in SR. Thus, extracting ample texture information from LR images is critical.

One plausible answer lies in using image gradient information. Prior studies [26, 32] integrated a fixed gradient extractor in the SR networks, demonstrating that supplementary gradient information could improve the network performance. However, merging gradient information into the main network inevitably causes pixel offset and artifacts, and a fixed gradient extractor may not offer the optimal adaptability for LR images in large-factor SR. Hence, there is a need to develop a learnable operator that can dynamically distinguish edge and smooth regions while improving gradient extractor adaptability. In this context, residual networks provide a potential solution, which can be mathematically formulated as,

$$X_n^R = X_{n-1}^R + R\left(X_{n-1}^R\right) \tag{4}$$

where X_{n-1}^R and X_n^R represent the input and output features of the n-th residual block, respectively, and $R(\cdot)$ represents the residual block. X_{n-1}^R is connected to the residual block as an identity, and residual block learns the residual between X_{n-1}^R and X_n^R. Given that most low frequency information is transmitted behind the residual blocks through the skip connection, i.e., the residual blocks only retain the high frequency information, the gradient extractors play a significant role in extracting this high frequency information. Inspired by this structure, we proposed the HC to replace vanilla convolution with a learning residual. However, using HC alone to learn the residuals is not flawless. Thus, we proposed our HLC to map the low frequency information in the residual features by combining the HC with the LC. In other words, HLC is conceptually divided into HC and LC, with each responsible for mapping of high frequency and low frequency information, respectively.

Fig. 3 shows the convolution processes of Prewitt, HC, and HLC. The HC process, designed based on Prewitt, calculates the gradients of the image by determining the first derivative. A given 3 × 3 field $\left\{f\left(z_i\right)\right\}_{i=1}^{9}$ allows Prewitt to compute area gradients, which can be mathematically formulated as,

$$\nabla f\left(z\right)=\sqrt{{g_x}^2+{g_y}^2}\approx\left|g_x\right|+\left|g_y\right|\approx g_x+g_y \tag{5}$$

$$g_x=\frac{\partial f\left(z\right)}{\partial x}\approx\left(z_1+z_4+z_7\right)-\left(z_3+z_6+z_9\right) \tag{6}$$

$$g_y=\frac{\partial f\left(z\right)}{\partial y}\approx\left(z_1+z_2+z_3\right)-\left(z_7+z_8+z_9\right) \tag{7}$$

where g_x and g_y denote horizontal and vertical gradients of a 3 × 3 field, respectively, and z_i represents the pixels in the field. This process is akin to subtracting pixels on one side of the field from those on the other (see Fig. 3a).

In contrast, HC develops kernels as a learnable operator, which acquires gradient information by convolving the learnable weights with pixel residuals (see Fig. 3b). The convolution process of HC can be mathematically written as,

$$\nabla f\left(z\right)=\lambda_1\boldsymbol{f}_x^{hc}\left(g_x,\,\omega\right)+\lambda_2\boldsymbol{f}_y^{hc}\left(g_y,\,v\right) \tag{8}$$

$$\begin{aligned}&\boldsymbol{f}_x^{hc}\left(g_x,\,\omega\right)=\left(z_1-z_2\right)\cdot\omega_1+\ldots+\left(z_9-z_8\right)\cdot\omega_9=\\&\sum\nolimits_{i\in\{1,\,4,\,7\}}\omega_i\cdot z_i+\sum\nolimits_{i\in\{3,\,6,\,9\}}\omega_i\cdot z_i+\ldots\end{aligned} \tag{9}$$

$$\begin{aligned}&\boldsymbol{f}_y^{hc}\left(g_y,\,v\right)=\left(z_1-z_4\right)\cdot v_1+\ldots+\left(z_9-z_6\right)\cdot\leq v_9=\\&\sum\nolimits_{i=1}^{N=3}v_i\cdot z_i+\sum\nolimits_{i=7}^{N=9}v_i\cdot z_i+\ldots\end{aligned} \tag{10}$$

where $\boldsymbol{f}_x^{hc}(\cdot)$ and $\boldsymbol{f}_y^{hc}(\cdot)$ denote horizontal and vertical HC operations, respectively; ω and v denote the HC parameters; and λ_1 and λ_2 denote the weights of horizontal and vertical gradients. It is easy to see that HC is Prewitt with the addition of learnable weights. Compared with Prewitt, HC obtains more accurate gradient information through the difference between the surrounding area and the middle area.

However, an image with only high frequency information is incomplete. Therefore, LC is required to process low frequency information. It is not a complicated operation only replacing each weight with a value of zero with a learned weight in each HC kernel. HC and LC together form the proposed HLC, a convolution incorporating gradient information and outperforming the vanilla convolution. The mathematical formulation of transformation can be written as,

$$\begin{aligned}\text{opt} &= \lambda_1 \boldsymbol{f}_x^{hlc}\left(g_x, \omega\right) + \lambda_2 \boldsymbol{f}_y^{hlc}\left(g_y, v\right) \\ &= \lambda_1\left(\boldsymbol{f}\left(g_x, \omega\right) + \sum_{i\in\{2,\,58\}} \omega_i \cdot z_i\right) + \\ &+\lambda_2\left(\boldsymbol{f}\left(g_y, v\right) + \sum_{i=4}^{N=6} v_i \cdot z_i\right)\end{aligned} \tag{11}$$

where $\boldsymbol{f}_x^{hlc}(\cdot)$ and $\boldsymbol{f}_y^{hlc}(\cdot)$ denote the horizontal and vertical HLC operations, respectively (see Fig. 3c).

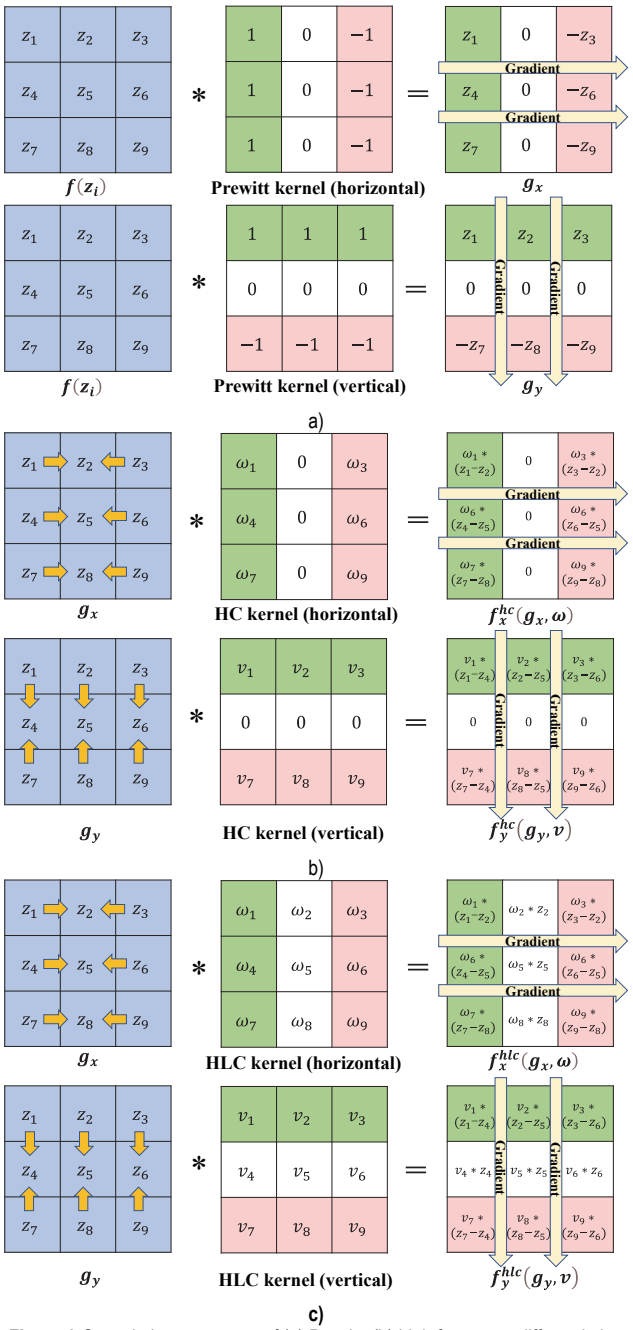

Figure 3 Convolution processes of (a) Prewitt, (b) high frequency differentiation convolution (HC), and (c) HLC. Here, $f(z_i)$ represents a 3 × 3 area of the input image. The yellow arrows indicate the differentiation process, while ω_i and v_i represent the parameters in the horizontal and vertical kernels, respectively. In (a), $f(z_i)$ is convolved with kernels to obtain the gradient components g_x and g_y. HLC and HC utilize the pixel subtraction of g_x and g_y to extract more precise gradient

Eq. (11) makes clear that one HLC operator performs two convolutional processes, significantly increasing the computational cost. To alleviate this problem, we switched the object of the difference from pixels to weights. Then, by merging two kernels and extracting features in one convolutional process, HLC becomes a vanilla convolution operation, which can be mathematically formulated as,

$$\begin{aligned}&\text{opt} = \lambda_1 \boldsymbol{f}_x^{hlc}\left(g_x, \omega\right) + \lambda_2 \boldsymbol{f}_y^{hlc}\left(g_y, v\right) = \\ &\left(\lambda_1\omega_1 + \lambda_2 v_1\right)\cdot z_1 + \ldots + \left(\lambda_1\omega_9 + \lambda_2 v_9\right)\cdot z_9 = \\ &\sum_{i=0}^{N=9} \hat{\omega}\cdot z_i = \boldsymbol{f}\left(z, \hat{\omega}\right)\end{aligned} \tag{12}$$

where $\hat{\omega}$ signifies the HLC parameters, and $\boldsymbol{f}(\cdot)$ denotes the convolutional operation.

3.3 Dynamic Deconvolution (DDC)

Deconvolution is a core up-sampling operation used to reconstruct SR images within various networks. However, deconvolution tends find an average solution of different images to accommodate diverse image features. Given the infinite paths from the LR images to the HR images, the average solution may incur significant reconstruction errors, especially as the scaling-factor increases. While the average solution may be acceptable for 2× SR [52], errors may increase exponentially for 4× or 8× SR (see Fig. 1), resulting in unsatisfactory visual results for images with unique features. Therefore, the average solution is not ideal for large-factor SR.

To mitigate this, we introduce DDC to reduce the reconstruction error caused by deconvolution by introducing additional experts and calculating routing weights for these experts. Essentially, DDC aims to learn specific "semantics" in the image that serve as conditions for activating different experts. The attention block within DDC calculates routing weights for each expert based on specific "semantics" in the image, determining which expert to use for processing the image. Unlike vanilla deconvolution that learns an average solution for the same kernel, DDC dynamically seeks the best solution following the routing weights of the corresponding experts, significantly reducing the reconstruction error.

DDC, depicted in Fig. 2 (highlighted in the yellow background), consists mainly of routing weights and corresponding experts. Routing weights are calculated for each expert through an attention block before the experts are combined. Specifically, the input feature maps ($\boldsymbol{X} \in \boldsymbol{R}^{b \times c \times w \times h}$, where b, c, w and h represent batch size, channel size, width, and height of features, respectively) are squeezed by pooling and processed through two fully connection layers for nonlinear mapping. The output feature maps ($\boldsymbol{X} \in \boldsymbol{R}^{b \times k \times 1 \times 1}$, where k denotes the numbers of experts) are fed into SoftMax to calculate routing weights $\boldsymbol{r} \in \boldsymbol{R}^{b \times k}$, which are assigned to experts as important factors. Each expert ($E_k \in \boldsymbol{R}^{C_{\text{out}} \times C_{\text{in}} \times s \times s}$, where C_{out}, C_{in} and s denote output number of channels, input number of channels and kernel size, respectively) is essentially a convolutional kernel. DDC uses the merged experts as the kernel for deconvolution. After the routing weight calculation is completed, each expert is compressed into one dimension and concatenated with other experts to

form a merged kernel $E_{nk} \in \boldsymbol{R}^{k \times C_{\text{out}} \times C_{\text{in}} \times s \times s}$. The final kernel $E \in \boldsymbol{R}^{b \times C_{\text{out}} \times C_{\text{in}} \times s \times s}$ is obtained by reshaping the result of $\times E_{nk}$, and used for deconvolution with groups = b.

Following another training strategy [39], we set a high temperature for SoftMax and a gradual decay during the first 10 epochs to flatten attention to enable all experts to be adequately trained. This can be mathematically formulated as,

$$r_k = \frac{\exp\left(f_c\left(\text{com}\left(X_k^D\right)/\tau\right)\right)}{\sum_i \exp\left(f_c\left(\text{com}\left(X_i^D\right)/\tau\right)\right)} \tag{13}$$

where X_k^D denotes the input feature, com(·) is the compressing operation, $f_c(\cdot)$ represents two full connection layers, and τ is the temperature. The original SoftMax can be mathematically expressed in the same form ($\tau = 1$). As the temperature increases, the sparsity of the routing weights decreases. Thus, setting a high temperature early in the training session helps the experts to be fully trained. Further details of the temperature settings are given in section 4.

4 EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we present the implementation specific and extensive experimental results to evaluate the effectiveness of our proposed method. We also compare the subjective and objective results of other state-of-the-art methods on several mainstream test datasets.

4.1 Datasets & Evaluation Metrics

We used extensive datasets for our performance evaluation. All of the training images were high-quality 2K resolution images from DIV2K [51], which contains 800 training and 100 validation images. HR-LR image pairs were obtained by down-sampling training images with the Bicubic algorithm. HR images were down-sampled to 1, 0.8, 0.7, 0.6, and 0.5 times the original size for data augmentation. The input LR images were randomly cropped to 64 × 64 and 40 × 40 RGB patches for 4× and 8× SR, respectively. The corresponding HR images' sizes matched the LR image' sizes times the scaling-factor. These image pairs, subjected to random horizontal and vertical flips and 90-degree rotations, acted as the input and ground truth, respectively. We employed the Laplace pyramid structure to reconstruct high-quality large-factor SR images, with the network reconstructing the small-factor SR images only accounting for a small portion of the overall network framework. All of the experiments focused on large scaling-factor SR, i.e., 4× and 8× SR. During the testing phase, we evaluated model performance using the Set5 [41], Set14 [42], B100 [43], Urban100 [44], and Manga109 [45] datasets. We utilize two evaluation metrics, PSNR and SSIM, to measure the performance of our proposed method for a comprehensive comparison with existing methods.

4.2 Implementation Details

During the training phase, we supplied our model with 16 LR patches as a minibatch and used the Adam method for optimization with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate was set to 10^{-4} and halved every 200 epochs. The entire training process spanned 1000 epochs using an NVIDIA 3060 GPU. Our proposed method, HFDDLap, incorporated a chain of $N = 3$ RCAB groups in each nonlinear mapping layer, with each RCAB group comprised of a chain of $M = 3$ HL-RCAB blocks. The number of experts in the DDC was set at $k = 3$.

4.3 Comparison with State-of-the-art Methods

We compared HFDDLap with several other state-of-the-art SR methods across several objective indicators, visual performance, and parameter quantities. The comparison methods included SRCNN [9], FSRCNN [52], VDSR [19], DRCN [22], DRRN [10], LapSRN [21], MS-LapSRN [54], MSRN [12], MemNet [55], SRResNet [28], SRGAN [28], SRDenseNet [25] and MFSN [66], all of which have a similar network size and can perform 4× SR and 8× SR. Then computational complexity of these algorithms varies, particularly when the scaling factor is large. For instance, the first five methods take pre-sampled images as input, significantly increasing computational complexity. The number of computations of LapSRN [21] and MS-LapSRN [54] increase exponentially as the scaling factor rises, while the computational cost of the remaining methods remains largely unchanged. We scrutinize the relationship between computational complexity and performance to support this comparison. Given the strong performance of SRFBN-S [57], EMASRN [58], CFSRCNN [16], SMSR [59] and SCFDN [64] on 4× SR, these are included in the 4× SR comparison. Despite its large model size, EDSR [53] was included in the 8× SR comparison to demonstrate our method's robust performance on large-factor SR.

4.3.1 Quantitative Comparison

Tab. 1 shows the 4× and 8× SR results over the benchmark datasets in terms of PSNR and SSIM for the methods compared. The best results are highlighted in red, and the second-place results are highlighted in blue. To enhance our model's performance, we incorporated a self-ensemble [65] approach, denoting that variant as HFDDLap+. Tab. 1 shows that our method outperformed the others on the benchmark datasets. Even when compared to EDSR, with 10 times as many parameters as HFDDLap, our method was considerably more efficient and effective. Specifically, HFDDLap+ and HFDDLap secured the top and the third position, and MFSN takes the second position on every benchmark dataset for 4× SR. HFDDLap+ distinctly outpaced others in terms of PSNR and SSIM. PSNR increased by over 0.1 dB on Manga109 compared to the second-place finisher. Both HFDDLap and HFDDLap+ also exceled in terms of PSNR and SSIM for 8× SR. HFDDLap+ achieved the best performance on nearly all of the benchmarks, with HFDDLap placing second in terms of SSIM. HFDDLap+ is only 0.01 dB lower than MFSN in terms of PSNR on Set14 dataset, but

SSIM is much higher than MFSN. EDSR, on the other hand, came in second place on the Set14, Urban100, and Manga109 datasets in terms of PSNR. Notably, EDSR employs a larger number of features, set at 256, which is significantly higher than HFDDLap's 64. As a result, EDSR is higher parameter count and much greater computational demand than other methods in support of its performance. However, HFDDLap's performance is not only comparable to EDSR but even better in some respects, which is sufficient to prove its effectiveness for large-factor SR.

Table 1 Quantitative evaluation on benchmark datasets. Average PSNR and SSIM values for 4× and 8× on datasets Set5, Set14, B100, Urban100 and Manga109. Red indicates the best performance and blue indicates the second-best performance among all methods

Method	Scale	Set5 [41]	Set14 [42]	B100 [43]	Urban100 [44]	Manga109 [45]
Bicubic	4×	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	24.89/0.7866
SRCNN [9]	4×	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221	27.58/0.8555
FSRCNN [52]	4×	30.72/0.8660	27.61/0.7550	26.98/0.7150	24.62/0.7280	27.90/0.8610
VDSR [19]	4×	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524	28.83/0.8870
DRCN [22]	4×	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510	28.93/0.8854
LapSRN [21]	4×	31.54/0.8850	28.19/0.7720	27.32/0.7270	25.21/0.7560	29.09/0.8900
MS-LapSRN [54]	4×	31.74/0.8890	28.26/0.7740	27.43/0.7310	25.51/0.7680	29.54/0.8970
DRRN [10]	4×	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638	29.46/0.8960
MSRN [12]	4×	32.07/0.8903	28.60/0.7751	27.52/0.7273	26.04/0.7896	30.17/0.9034
MemNet [55]	4×	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630	29.42/0.8942
SRResNet [28]	4×	32.02/0.8930	28.50/0.7780	27.53/0.7330	26.05/0.7810	29.49/0.8990
SRGAN [28]	4×	32.05/0.8910	28.49/0.7820	27.61/0.7360	26.09/0.7830	30.70/0.9080
SRDenseNet [25]	4×	29.46/0.8380	26.60/0.7180	25.74/0.6660	24.50/0.7360	27.79/0.8560
SRFBN-S [57]	4×	31.98/0.8923	28.45/0.7779	27.44/0.7313	25.71/0.7719	29.91/0.9008
EMASRN [58]	4×	32.17/0.8948	28.57/0.7809	27.55/0.7351	26.01/0.7838	30.41/0.9076
CFSRCNN [16]	4×	32.06/0.8920	28.57/0.7800	27.53/0.7333	26.03/0.7824	30.30/0.9048
SMSR [59]	4×	32.12/0.8932	28.55/0.7808	27.55/0.7351	26.11/0.7868	30.54/0.9085
SCFDN [64]	4×	32.25/0.8955	28.64/0.7827	27.58/0.7365	26.16/0.7891	30.60/0.9091
MFSN [66]	4×	32.36/0.8969	28.67/0.7835	27.60/0.7374	26.26/0.7911	30.72/0.9110
HFDDLap (Ours)	4×	32.25/0.8957	28.64/0.7829	27.59/0.7369	26.21/0.7894	30.59/0.9096
HFDDLap+ (Ours)	4×	32.37/0.8970	28.71/0.7846	27.64/0.7382	26.35/0.7924	30.85/0.9122
Bicubic	8×	24.40/0.6580	23.10/0.5660	23.67/0.5480	20.74/0.5160	21.47/0.6500
SRCNN [9]	8×	25.33/0.6900	23.76/0.5910	24.13/0.5660	21.29/0.5440	22.46/0.6950
FSRCNN [52]	8×	20.13/0.5520	19.75/0.4820	24.21/0.5680	21.32/0.5380	22.39/0.6730
VDSR [19]	8×	25.93/0.7240	24.26/0.6140	24.49/0.5830	21.70/0.5710	23.16/0.7250
DRCN [22]	8×	25.93/0.6743	24.25/0.5510	24.49/0.5168	21.71/0.5289	23.20/0.6686
LapSRN [21]	8×	26.14/0.7370	24.35/0.6200	24.54/0.5850	21.81/0.5800	23.39/0.7340
MS-LapSRN [54]	8×	26.34/0.7558	24.57/0.6273	24.65/0.5895	22.06/0.5963	23.90/0.7564
DRRN [10]	8×	26.18/0.7380	24.42/0.6220	24.59/0.5870	21.88/0.5830	23.60/0.7420
MSRN [12]	8×	26.59/0.7254	24.88/0.5961	24.70/0.5410	22.37/0.5977	24.28/0.7517
MemNet [55]	8×	26.16/0.7414	24.38/0.6199	24.58/0.5842	21.89/0.5825	23.56/0.7387
SRResNet [28]	8×	26.62/0.7560	24.55/0.6240	24.65/0.5870	22.05/0.5890	23.88/0.7480
SRGAN [28]	8×	23.04/0.6260	21.57/0.4950	21.78/0.4420	19.64/0.4680	20.42/0.6250
SRDenseNet [25]	8×	25.99/0.7040	24.23/0.5810	24.45 0.5300	21.67/0.5620	23.09/0.7120
EDSR [53]	8×	27.00/0.7757	24.98/0.6410	24.77/0.5963	22.42/0.6136	24.51/0.7748
MFSN [66]	8×	26.97/0.7738	25.00/0.6413	24.80/0.5967	22.45/0.6161	24.51/0.7765
HFDDLap (Ours)	8×	27.02/0.7811	24.90/0.6410	24.81/0.5966	22.41/0.6162	24.45/0.7812
HFDDLap+ (Ours)	8×	27.13/0.7837	24.99/0.6431	24.85/0.5979	22.51/0.6196	24.59/0.7849

Table 2 Comparisons of SR algorithms in terms of numbers of parameters and execution time for 4× and 8× on dataset Set5

Methods	4×		8×	
	#Params. / M	Time / s	#Params. / M	Time / s
DRCN [22]	1.8	0.4798	1.8	0.5729
LapSRN [21]	0.8	0.0244	0.8	0.0304
DRRN [10]	0.3	0.1228	0.3	0.1688
MSRN [12]	6.0	0.0420	6.1	0.0312
MemNet [55]	0.7	0.0446	0.9	0.0474
SRDenseNet [25]	2.0	0.0604	2.3	0.0628
EMASRN [58]	0.5	0.1338	-	-
CFSRCNN [16]	1.4	0.0434	-	-
SCFDN [64]	2.4	0.3218	-	-
EDSR [53]	-	-	45.5	0.0940
HFDDLap (Ours)	2.2	0.0868	3.3	0.1232

The comparison parameter counts and computational demand of the methods is depicted in Tab. 2, with execution time tests averaged over the Set5 dataset. Although HFDDLap does not lead the pack in terms of parameter count, it had excellent execution time, even outperforming other methods with fewer parameters. For instance, DRCN uses a recursive strategy, allowing multiple iterative feature extractions in the network, while EMASRN integrates an EM self-attention mechanism to identify inherent correlations within images over several iterations. Despite not introducing a substantial number of parameters, these iterations amplify the computational demands. SRDenseNet and HFDDLap employ a progressive up-sampling approach, increasing the complexity of the network. Despite this, the strong performance delivered by the progressive up-sampling approach for large-factor SR means the complexity of HFDDLap is not excessively high.

4.3.2 Qualitative Comparison

Fig. 4 presents the SR results for the tested methods on the Urban100 and Manga109 datasets for 4× SR. Most of the methods reconstruct images with blurred or even incorrect textures and details, accompanied by undesirable blurring artifacts. In contrast, our HFDDLap consistently reconstructed images with heightened precision, enhancing textures and details, and sharpening edges. Consider "img_099" from Urban100 as an example: early methods such as bicubic and LapSRN failed to reconstruct results with precise edges. Recent approaches like SRFBN-S, EMASRN, and CFSRCNN delivered superior visual results, but with significantly blurred texture details. CFSRCNN generated notably unpleasant artifacts. The gradient information within HFDDLap enabled it to generate images with sharper structures, better texture detail retention, and closer resemblance to HR images.

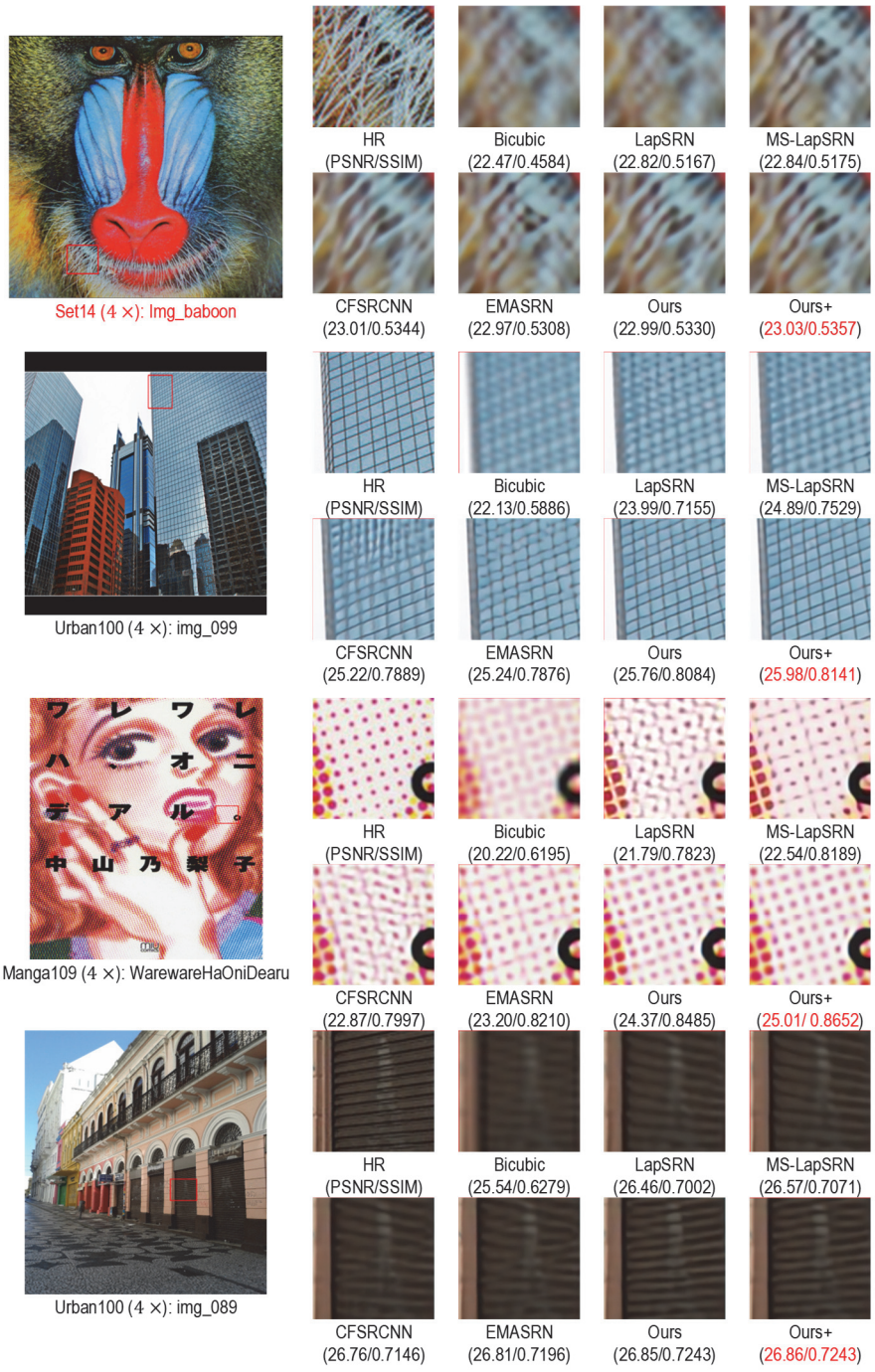

Figure 4 Visual comparisons with state-of-the-art SISR methods for 4× SR on the Set14, Urban100, and the Manga 109 datasets. Best viewed on screen

Fig. 5 shows the zoomed-in results of several algorithms on Urban100 and the Manga109 datasets for 8× SR. For large-factor SR, considerable information is lost in low resolution images, typically resulting in blurry reconstructed SR images. Despite these extreme conditions, HFDDLap still retained more texture than other methods. For example, the "img-046" image from Urban100, when reconstructed by the Bicubic algorithm, is entirely unrecognizable. LapSRN and MS-LapSRN provided slight improvement over bicubic, preserving

some coarser textures, though certain areas remain very blurry. MSRN and EDSR offered more visually appealing results and clearer texture details. Nonetheless, HFDDLap reconstructed the most abundant texture and detail among all methods, preserving stripes in multiple directions.

Figure 5 Visual comparisons with state-of-the-art SISR methods for 8× SR on the Urban100 and Mang109 datasets; Best viewed on screen

4.4 Ablation Studies

4.4.1 High-Low Frequency Differentiation Convolution (HLC)

The HLC is a critical component of our method, replacing the first convolutional layer in the RCAB block. As illustrated in Tab. 3, we conducted an ablation study using three models, each tested on the Set5, Set14 and Manga109 datasets for 4× SR.

Table 3 Ablation study on the high-low differentiation convolution (HLC) in terms of PSNR on Set5, Set14 and Manga109 dataset for 4× SR

Model Index		$Model_1$	$Model_2$	$Model_3$
HLC	HC	×	√	√
	LC	×	×	√
Datasets	Set5	32.09	31.66	32.25
	Set14	28.50	27.85	28.64
	Manga109	30.21	29.06	30.59

a) HLC & Vanilla Convolution

$Model_1$ and $Model_3$ juxtapose our method with vanilla convolution and with HLC respectively. Both models use DDC with 3 experts (k = 3) as the up-sampling operator. Without HLC, $Model_1$ experiences a drastic drop in PSNR. By replacing vanilla convolution with HLC in $Model_3$, given the results on Set5 dataset, PSNR increased by 0.15 dB. This result underscores the effectiveness of HLC, validating that integrating gradient information to the network enhances HFDDLap's performance.

b) HLC with One or Two Convolutional Operators

Since SR tasks differ from edge detection tasks, the former generates a complete image, while the latter only edge information, LC is incorporated into HLC to extract low frequency information. In $Model_2$, HC is solely responsible for feature extraction without LC. As anticipated, removing LC led to a substantial drop in PSNR, performing even more poorly than vanilla convolution ($Model_1$). Fig. 6 shows the reconstruction results of $Model_2$ (HC only) and $Model_3$ (HLC) for "butterfly" in the Set5 dataset.

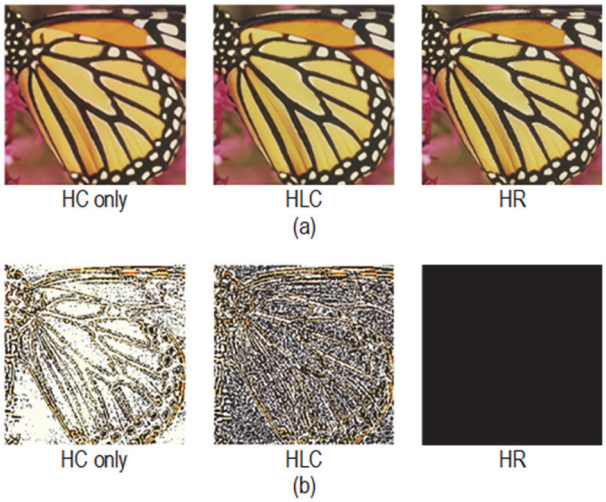

Figure 6 Visual comparisons with $Model_3$ (HLC) and $Model_2$ (HC only). (a) Original HR image and corresponding SR results. (b) SR residual results

Both models' results demonstrate satisfactory texture detail retention, but those of $Model_2$ exhibit clear color distortion compared to $Model_3$, especially in the butterfly wing area. In contrast, the result of $Model_3$ is more similar to the HR image in texture and color. The differences between $Model_2$ and $Model_3$ are even more visually in the residual images, with $Model_2$ showing significant error in the low frequency region, which accounts for the color

deviation. This can be attributed to $Model_2$ discarding the LC responsible for low frequency mapping, leading to subpar data representation performance in this region.

c) HLC with One or Two Convolutional Operators

As expressed in Eq. (11), two convolutional operations (extraction of horizontal and vertical gradient information) are required to complete one HLC. Given the linearity of the convolution operation, the process is optimized to one convolutional operation as depicted in Eq. (12). The comparative results of the two operation methods on the Set5 dataset for 4× SR are displayed in Tab. 4. As expected, the PSNR results generated by the two operations do not differ significantly, with HLC employing a single convolutional operation performing slightly better. This may be because two convolution operations cause varying degrees of pixel shift in the generated feature maps, and the addition operator exacerbated this error. The optimized HLC also reduces the computational burden by almost half compared to the previous approach, while achieving superior performance. However， the extra convolutional operation explains the dramatic increase in execution time on the Set5 dataset. Given these results, it may be advisable to use only one convolutional operation in each HLC.

Table 4 Ablation study on high-low differentiation convolution (HLC) with one conv. and two conv. in terms of PSNR and execution time on Set5 dataset for 4× SR

	HLC with one Conv.	HLC with two Conv.
PSNR	32.25	32.22
Time /s	0.0868	0.4709

4.4.2 Dynamic Deconvolution (DDC)

We propose DDC to enhance the existing up-sampling operations and improve model performance. Two primary factors influencing the performance of the DDC merit an ablation study: the temperature setting and the number of experts. Tab. 5 and Fig. 7 show the effect of the different numbers of experts and temperature settings on the results of the Set5, B100 and Manga109 datasets for 4× SR.

Table 5 Ablation study on the dynamic deconvolution (DC) in terms of PSNR under different temperature settings on B100, Set14 and Manga109 dataset for 4× SR

Datasets	Set5	B100	Manga109
$\tau = 1$ train	32.22	27.58	30.55
$\tau = 15$ train	32.22	27.56	30.57
$\tau = 30$ train	32.25	27.59	30.59
$\tau = 45$ train	32.21	27.58	30.58

a) The number of experts in DDC

The results for different numbers of experts ($k = 2$, $k = 3$, $k = 4$) and the results for vanilla deconvolution on the Set5 dataset for 4× SR are shown in Fig. 7. The DDC with $k = 3$, as shown by the red line in Fig. 7, outperformed all other results with a consistent temperature throughout training ($\tau = 1$). Compared to vanilla deconvolution, DDC with $k = 3$ improved PSNR by 0.26 dB, and the DDC with $k = 4$ improved PSNR by 0.13 dB. It is evident that the number of experts did not correlate positively with performance, therefore, leading us to use DDC with $k = 3$ for all models. However, DDC with $k = 2$ failed to surpass vanilla deconvolution. This can be ascribed to two factors. First, the low number of experts led to the inability of DDC to reconstruct satisfactory results. Second, a few experts were not sufficiently trained during the training process, failing to fulfill their intended role. Hence, it is crucial to establish a high temperature during training to circumvent this issue.

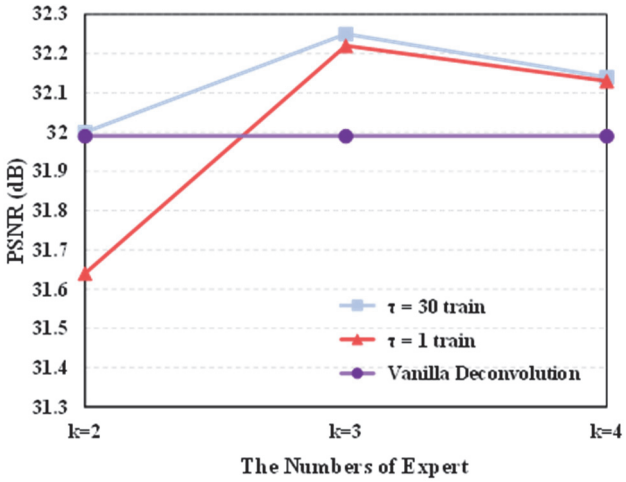

Figure 7 Ablation study of the dynamic deconvolution in terms of PSNR under different temperature settings and various numbers of experts on the Set5 dataset for 4× SR

b) The temperature setting in the DDC

Four temperature settings were adopted in separate experiments. The first experience kept the temperature $\tau = 1$ throughout the entire training process, while the second set the temperature $\tau = 30$ in the first epoch, reducing it gradually to $\tau = 1$ over 10 epochs. As indicated by the blue line in Fig. 7, starting training with a higher temperature significantly improved DDC performance, with all results better than vanilla deconvolution. This improvement likely stems from employing high temperatures early in training to fully activate all experts, enhancing the attention mechanism's exploratory capacity. Subsequently, lowering the temperature increases sparsity, thereby improving model convergence. In addition, a high temperature setting yielded better results with fewer experts, while with four experts, it only slightly enhanced performance. This could be because when the number of experts is too large, the network lacks sufficient performance to ensure that each expert can be adequately trained. Adjusting the temperature can only mitigate this issue. Based on these findings, we set the DDC in all of the models to the expert number $k = 3$ and trained all of the models with a decreasing temperature. To corroborate the impact of temperature setting on the performance of DDCs, we conducted experiments utilizing a range of temperature settings: $\tau = 1$, $\tau = 15$, $\tau = 30$ and $\tau = 45$. The outcomes of these tests are shown in Tab. 5, performed on the Set5, B100 and Manga109 datasets for 4× SR. An important observation is that the model trained with $\tau = 30$ consistently outperformed its counterparts across all test datasets. Interestingly, it appears that there was no direct positive correlation between the temperature settings and the DDC performance. Considering all of these results, we chose to employ a temperature setting of $\tau = 30$ for all of the training runs of the HFDDLap model.

c) Expert activation in DDC

To validate and verify that each expert in the DDC is functioning, we examined the activation of each expert in the last DDC in the 4× SR model. Tab. 6 shows the average activation results under the two temperature settings for each expert on the Set5 dataset. It is evident that experts trained in high-temperature scenarios were activated to varying degrees. Conversely, models trained at constant temperature were heavily reliant on the third expert to function, with the first expert activated only sporadically and the second expert never started. Despite only two experts being activated, DDC's performance still surpassed that of vanilla deconvolution.

Table 6 Average activation results for each expert on Set5 dataset for 4× SR

Expert Index	$Expert_1$	$Expert_2$	$Expert_3$
$\tau = 30$ train	27.63%	35.04%	37.33%
$\tau = 1$ train	1%	0	99%

4.5 Model Structure Analysis

To investigate the impact of single versus multiple scaling-factors on the SR results. we altered HFDDLap's progressive up-sampling structure to a post-up-sampling structure for performance comparison. For example, in the 4× HFDDLap model, the first DDC for up-sampling twice was moved behind all nonlinear mapping layers. Tab. 7 presents PSRN results on the Set5 dataset. The progressive up-sampling structure indeed outperformed the post up-sampling structure. The Laplacian pyramid structure extracted features at different scales, akin to incorporating multi-scale prior information into the network. This not only facilitated the network to discern differences and connections between different scale features but also enabled more information from feature maps of larger sizes, thereby enhancing the texture richness in the SR results.

Table 7 Results of HFDDLAP with progressive up-sampling structure and post up-sampling structure in terms of PSNRon set5 dataset for 4× SR

	progressive up-sampling	post up-sampling
PSNR	32.25	32.21

4.6 Applicability Analysis for DDC

To test the applicability of DDC in other SR networks, we selected EDSR-baseline [53], a classical SR network, for an up-sampling operations comparison. Sub-pixel convolution and deconvolution are widely adopted in SR networks, with EDSR-baseline employing sub-pixel convolution for up-sampling. We substituted the up-sampling block in EDSR-baseline with deconvolution and DDC, respectively, and retrained all three of the models under identical conditions for 4× SR. Given that the stride size of deconvolution should not exceed the kernel size to prevent the information loss [17], we positioned two deconvolutions (DDCs) in the up-sampling block, with each performing 2× SR. Tab. 8 shows the PSRN results of the three models on the Set5 dataset. The terms EDSR-baseline-DDC and EDSR-baseline-DC refer to EDSR-baseline models with the up-sampling block replaced by DDC and vanilla deconvolution, respectively. EDSR-baseline-DC surpassed the original model with the two-stage up-sampling design, albeit with a marginal improvement. However, compared to the original model, EDSR-baseline-DDC showed significant performance enhancement, attesting to DDC's adaptability to other SR networks.

Table 8 Results of EDSR-baseline with different up-sampling blocks in terms of PSNR on Set5 dataset for 4× SR

	EDSR-baseline	EDSR-baseline-DC	EDSR-baseline-DDC
PSNR	32.04	32.09	32.14

5 CONCLUSION

This paper presents the HFDDLap model for SR tasks with relatively large scaling-factors. Experimental findings verify that the proposed approach effectively addresses the texture detail loss in LR images, and artifacts and pixel displacement due to exponential growth of reconstruction error in SR tasks with larger scaling-factors. Concretely, HLC is engineered to extract the gradient information by pixel differentiation, providing additional a priori information to enhance the texture and detail of SR images. DDC is introduced to uncover the potential features of the image and adaptively select the suitable expert for the up-sampling operation based on these features. A sequence of ablation studies for HLC and DDC bolster these assertions. Integrating HLC and DDC into a network with a Laplacian pyramid structure enables it to reconstruct visually superior results in large-factor SR operations. Experimental results confirm that HFDDLap outperforms other state-of-the-art methods. Despite these promising outcomes, HFDDLap exhibits certain limitations. Its performance may degrade when applied to real-world scenarios involving unknown or complex degradation patterns, such as motion blur, noise, or compression artifacts, which are not adequately represented in the training data. Additionally, SR tasks with larger factors (e.g., 20× SR, 30× SR) remain a substantial challenge. The extraction of effective information in these more drastically degraded environments is a direction we will explore in future work.

6 REFERENCES

[1] Liu, X. (2020). Real-time Defogging of Single Image of IoTs-based Surveillance Video Based on MAP. *Tehnički vjesnik*, *27*(4), 1262-1269. https://doi.org/10.17559/TV-20200527085338

[2] Yang, Y., Wang, Y., Wang, H., Zhang, L., Zhao, E., Song, M., & Yu, C. (2024). Spectral-enhanced sparse transformer network for hyperspectral super-resolution reconstruction. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *17*, 17278-17291. https://doi.org/10.1109/JSTARS.2024.3457814

[3] Keys, R. (1981). Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing*, *29*(6), 1153-1160. https://doi.org/10.1109/TASSP.1981.1163711

[4] Sanchez-Beato, A. & Pajares, G. (2008). Noniterative interpolation-based super-resolution minimizing aliasing in the reconstructed image. *IEEE transactions on image processing*, 17(10), 1817-1826. https://doi.org/10.1109/TIP.2008.2002833

[5] Zhou, F., Yang, W., & Liao, Q. (2012). Interpolation-based image super-resolution using multisurface fitting. *IEEE Transactions on Image Processing*, *21*(7), 3312-3318.

https://doi.org/10.1109/TIP.2012.2189576

[6] Yang, J., Wright, J., Huang, T. S., & Ma, Y. (2010). Image super-resolution via sparse representation. *IEEE transactions on image processing*, *19*(11), 2861-2873. https://doi.org/10.1109/TIP.2010.2050625

[7] Huang, Y., Li, J., Gao, X., He, L., & Lu, W. (2018). Single image super-resolution via multiple mixture prior models. *IEEE Transactions on Image Processing*, *27*(12), 5904-5917. https://doi.org/10.1109/TIP.2018.2860685

[8] Hu, Y., Wang, N., Tao, D., Gao, X., & Li, X. (2016). SERF: A simple, effective, robust, and fast image super-resolver from cascaded linear regression. *IEEE Transactions on Image Processing*, *25*(9), 4091-4102. https://doi.org/10.1109/TIP.2016.2580942

[9] Dong, C., Loy, C. C., He, K., & Tang, X. (2014). Learning a deep convolutional network for image super-resolution. *Proceedings of the European conference on computer vision*, 184-199. https://doi.org/10.1007/978-3-319-10593-2_13

[10] Tai, Y., Yang, J., & Liu, X. (2017). Image super-resolution via deep recursive residual network. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3147-3155. https://doi.org/10.1109/CVPR.2017.298

[11] Dai, T., Cai, J., Zhang, Y., Xia, S. T., & Zhang, L. (2019). Second-order attention network for single image super-resolution. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 11065-11074. https://doi.org/10.1109/CVPR.2019.01132

[12] Li, J., Fang, F., Mei, K., & Zhang, G. (2018). Multi-scale residual network for image super-resolution. *Proceedings of the European conference on computer vision*, 517-532. https://doi.org/10.1007/978-3-030-01237-3_32

[13] López-Tapia, S. & de la Blanca, N. P. (2021). Fast and robust cascade model for multiple degradation single image super-resolution. *IEEE Transactions on Image Processing*, 30, 4747-4759. https://doi.org/10.1109/TIP.2021.3074821

[14] Huang, Y., Li, J., Gao, X., Hu, Y., & Lu, W. (2021). Interpretable detail-fidelity attention network for single image super-resolution. *IEEE Transactions on Image Processing*, *30*, 2325-2339. https://doi.org/10.1109/TIP.2021.3050856

[15] Zhang, X., Gao, P., Liu, S., Zhao, K., Li, G., Yin, L., & Chen, C. W. (2020). Accurate and efficient image super-resolution via global-local adjusting dense network. *IEEE Transactions on multimedia*, *23*, 1924-1937. https://doi.org/10.1109/TMM.2020.3005025

[16] Tian, C., Xu, Y., Zuo, W., Zhang, B., Fei, L., & Lin, C. W. (2020). Coarse-to-fine CNN for image super-resolution. *IEEE Transactions on Multimedia*, *23*, 1489-1502. https://doi.org/10.1109/TMM.2020.2999182

[17] Liu, Y., Wang, S., Zhang, J., Wang, S., Ma, S., & Gao, W. (2021). Iterative network for image super-resolution. *IEEE Transactions on Multimedia*, *24*, 2259-2272. https://doi.org/10.1109/TMM.2021.3078615

[18] Hu, Y., Li, J., Huang, Y., & Gao, X. (2021). Image super-resolution with self-similarity prior guided network and sample-discriminating learning. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(4), 1966-1985. https://doi.org/10.1109/TCSVT.2021.3093483

[19] Kim, J., Lee, J. K., & Lee, K. M. (2016). Accurate image super-resolution using very deep convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1646-1654. https://doi.org/10.1109/CVPR.2016.182

[20] Haris, M., Shakhnarovich, G., & Ukita, N. (2018). Deep back-projection networks for super-resolution. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1664-1673. https://doi.org/10.1109/CVPR.2018.00179

[21] Lai, W. S., Huang, J. B., Ahuja, N., & Yang, M. H. (2017). Deep laplacian pyramid networks for fast and accurate super-resolution. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 624-632. https://doi.org/10.1109/CVPR.2017.618

[22] Kim, J., Lee, J. K., & Lee, K. M. (2016). Deeply-recursive convolutional network for image super-resolution. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1637-1645. https://doi.org/10.1109/CVPR.2016.181

[23] Ahn, N., Kang, B., & Sohn, K. A. (2018). Fast, accurate, and lightweight super-resolution with cascading residual network. *Proceedings of the European conference on computer vision*, 252-268. https://doi.org/10.1007/978-3-030-01249-6_16

[24] Song T, Wen R, & Zhang L. (2024). RoughSet-DDPM: An Image Super-Resolution Method Based on Rough set Denoising Diffusion Probability Model. *Tehnički vjesnik*, *31*(1), 162-170. https://doi.org/10.17559/TV-20230717000808

[25] Tong, T., Li, G., Liu, X., & Gao, Q. (2017). Image super-resolution using dense skip connections. *Proceedings of the IEEE international conference on computer vision*, 4799-4807. https://doi.org/10.1109/ICCV.2017.514

[26] Cai, Q., Li, J., Li, H., Yang, Y. H., Wu, F., & Zhang, D. (2022). TDPN: Texture and detail-preserving network for single image super-resolution. *IEEE Transactions on Image Processing*, *31*, 2375-2389. https://doi.org/10.1109/TIP.2022.3154614

[27] Wang, X. (2024). GAN-DNADE: Image Encryption Algorithm Based on Generative Adversarial Network and DNA Dynamic Encoding. *Computer Science and Information Systems*, *21*(4,) 1673-1697. https://doi.org/10.2298/CSIS240314053W

[28] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A. et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4681-4690. https://doi.org/10.1109/CVPR.2017.19

[29] Sun, W., Gong, D., Shi, Q., van den Hengel, A., & Zhang, Y. (2021). Learning to zoom-in via learning to zoom-out: Real-world super-resolution by generating and adapting degradation. *IEEE Transactions on Image Processing*, *30*, 2947-2962. https://doi.org/10.1109/TIP.2021.3049951

[30] Mahapatra, D., Bozorgtabar, B., & Garnavi, R. (2019). Image super-resolution using progressive generative adversarial networks for medical image analysis. *Computerized Medical Imaging and Graphics*, *71*, 30-39. https://doi.org/10.1016/j.compmedimag.2018.10.005

[31] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C. et al. (2018). Esrgan: Enhanced super-resolution generative adversarial networks. *Proceedings of the European conference on computer vision workshops*, 63-79. https://doi.org/10.1007/978-3-030-11021-5_5

[32] Ma, C., Rao, Y., Cheng, Y., Chen, C., Lu, J., & Zhou, J. (2020). Structure-preserving super resolution with gradient guidance. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7769-7778. https://doi.org/10.1109/CVPR42600.2020.00779

[33] Yan, Y., Ren, W., Hu, X., Li, K., Shen, H., & Cao, X. (2021). SRGAT: Single image super-resolution with graph attention network. *IEEE Transactions on Image Processing*, *30*, 4905-4918. https://doi.org/10.1109/TIP.2021.3077135

[34] Zhang, L., Lang, Z., Wei, W., & Zhang, Y. (2021). Embarrassingly simple binarization for deep single imagery super-resolution networks. *IEEE Transactions on Image Processing*, *30*, 3934-3945. https://doi.org/10.1109/TIP.2021.3066906

[35] Song, D., Wang, Y., Chen, H., Xu, C., Xu, C., & Tao, D. (2021). Addersr: Towards energy efficient image

super-resolution. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 15648-15657. https://doi.org/10.1109/CVPR46437.2021.01539

[36] Sobel, I. & Feldman, G. (1968). A 3×3 isotropic gradient operator for image processing, Pattern Classif Scene Anal, 271-272.

[37] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, *6*, 679-698. https://doi.org/10.1109/TPAMI.1986.4767851

[38] Prewitt, J. M. (1970). Object enhancement and extraction. *Picture processing and Psychopictorics*, *10*(1), 15-19.

[39] Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., & Liu, Z. (2020). Dynamic convolution: Attention over convolution kernels. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 11030-11039. https://doi.org/10.1109/CVPR42600.2020.01104

[40] Wang, L., Wang, Y., Lin, Z., Yang, J., An, W., & Guo, Y. (2021). Learning a single network for scale-arbitrary super-resolution. *Proceedings of the IEEE international conference on computer vision*, 4801-4810. https://doi.org/10.1109/ICCV48922.2021.00476

[41] Bevilacqua, M., Roumy, A., Guillemot, C., & Alberi-Morel, M. L. (2012). Low-complexity single-image super-resolution based on nonnegative neighbor embedding, *British Machine Vision Conference*. 135.1-135.10. https://doi.org/10.5244/C.26.135

[42] Zeyde, R., Elad, M., & Protter, M. (2012). On single image scale-up using sparse-representations. *Proceedings of Curves and Surfaces: 7th International Conference*, 711-730. https://doi.org/10.1007/978-3-642-27413-8_47

[43] Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *Proceedings of eighth IEEE international conference on computer vision*, *2*, 416-423. https://doi.org/10.1109/ICCV.2001.937655

[44] Huang, J. B., Singh, A., & Ahuja, N. (2015). Single image super-resolution from transformed self-exemplars. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5197-5206. https://doi.org/10.1109/CVPR.2015.7299156

[45] Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., & Aizawa, K. (2017). Sketch-based manga retrieval using manga109 dataset. *Multimedia tools and applications*, *76*, 21811-21838. https://doi.org/10.1007/s11042-016-4020-z

[46] Yu, Z., Zhao, C., Wang, Z., Qin, Y., Su, Z., Li, X. et al. (2020). Searching central difference convolutional networks for face anti-spoofing. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5295-5305. https://doi.org/10.1109/CVPR42600.2020.00534

[47] Yu, Z., Wan, J., Qin, Y., Li, X., Li, S. Z., & Zhao, G. (2020). NAS-FAS: Static-dynamic central difference network search for face anti-spoofing. *IEEE transactions on pattern analysis and machine intelligence*, *43*(9), 3005-3023. https://doi.org/10.1109/TPAMI.2020.3036338

[48] Su, Z., Liu, W., Yu, Z., Hu, D., Liao, Q., Tian, Q. et al. (2021). Pixel difference networks for efficient edge detection. *Proceedings of the IEEE/CVF international conference on computer vision*, 5117-5127. https://doi.org/10.1109/ICCV48922.2021.00507

[49] Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, *24*(7), 971-987. https://doi.org/10.1109/TPAMI.2002.1017623

[50] Yang, B., Bender, G., Le, Q. V., & Ngiam, J. (2019). Condconv: Conditionally parameterized convolutions for efficient inference. *Advances in neural information processing systems*, *32*.

[51] Agustsson, E. & Timofte, R. (2017). Ntire 2017 challenge on single image super-resolution: Dataset and study. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 126-135. https://doi.org/10.1109/CVPRW.2017.150

[52] Dong, C., Loy, C. C., & Tang, X. (2016). Accelerating the super-resolution convolutional neural network. *Proceedings of the European conference on computer vision*, *14*, 391-407. https://doi.org/10.1007/978-3-319-46475-6_25

[53] Lim, B., Son, S., Kim, H., Nah, S., & Mu Lee, K. (2017). Enhanced deep residual networks for single image super-resolution. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 136-144. https://doi.org/10.1109/CVPRW.2017.151

[54] Lai, W. S., Huang, J. B., Ahuja, N., & Yang, M. H. (2018). Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE transactions on pattern analysis and machine intelligence*, *41*(11), 2599-2613. https://doi.org/10.1109/TPAMI.2018.2865304

[55] Tai, Y., Yang, J., Liu, X., & Xu, C. (2017). Memnet: A persistent memory network for image restoration. *Proceedings of the IEEE international conference on computer vision*, 4539-4547. https://doi.org/10.1109/ICCV.2017.486

[56] Wang, Z., Liu, D., Yang, J., Han, W., & Huang, T. (2015). Deep networks for image super-resolution with sparse prior. *Proceedings of the IEEE international conference on computer vision*, 370-378. https://doi.org/10.1109/ICCV.2015.50

[57] Li, Z., Yang, J., Liu, Z., Yang, X., Jeon, G., & Wu, W. (2019). Feedback network for image super-resolution. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3867-3876. https://doi.org/10.1109/CVPR.2019.00399

[58] Zhu, X., Guo, K., Ren, S., Hu, B., Hu, M., & Fang, H. (2021). Lightweight image super-resolution with expectation-maximization attention mechanism. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(3), 1273-1284. https://doi.org/10.1109/TCSVT.2021.3078436

[59] Wang, L., Dong, X., Wang, Y., Ying, X., Lin, Z., An, W., & Guo, Y. (2021). Exploring sparsity in image super-resolution for efficient inference. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4917-4926. https://doi.org/10.1109/CVPR46437.2021.00488

[60] Anwar, S. & Barnes, N. (2020). Densely residual laplacian super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(3), 1192-1204. https://doi.org/10.1109/TPAMI.2020.3021088

[61] Guo, Y., Chen, J., Wang, J., Chen, Q., Cao, J., Deng, Z. et al. (2020). Closed-loop matters: Dual regression networks for single image super-resolution. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5407-5416. https://doi.org/10.1109/CVPR42600.2020.00545

[62] Haris, M., Shakhnarovich, G., & Ukita, N. (2018). Deep back-projection networks for super-resolution. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1664-1673. https://doi.org/10.1109/CVPR.2018.00179

[63] Liu, Y., Wang, S., Zhang, J., Wang, S., Ma, S., & Gao, W. (2021). Iterative network for image super-resolution. *IEEE Transactions on Multimedia*, *24*, 2259-2272. https://doi.org/10.1109/TMM.2021.3078615

[64] Zhou, Y., Liang, T., Jiang, Z., & Men, A. (2022, June). Efficient network removing feature redundancy for single image super-resolution. *Proceedings of IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, 1-6. https://doi.org/10.1109/BMSB55706.2022.9828763

[65] Timofte, R., Rothe, R., & Van Gool, L. (2016). Seven ways to improve example-based single image super resolution. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1865-1873. https://doi.org/10.1109/CVPR.2016.206

[66] Li, M., Zhao, Y., Zhang, F., Luo, B., Yang, C., Gui, W., & Chang, K. (2024). Multi-scale feature selection network for lightweight image super-resolution. *Neural Networks*, *169*, 352-364. https://doi.org/10.1016/j.neunet.2023.10.043

[67] Canchila, S., Meneses-Eraso, C., Casanoves-Boix, J., Cortés-Pellicer, P., & Castelló-Sirvent, F. (2024). Natural language processing: An overview of models, transformers and applied practices. *Computer Science and Information Systems*, *21*(3), 1097-1145. https://doi.org/10.2298/CSIS230217031C

[68] Xiao, Y., Yuan, Q., Jiang, K., He, J., Lin, C. W., & Zhang, L. (2024). TTST: A top-k token selective transformer for remote sensing image super-resolution. *IEEE Transactions on Image Processing*, *33*, 738-752. https://doi.org/10.1109/TIP.2023.3349004

Contact information:

Peituan LIU
Tiangong University,
School of Computer Science and Technology,
Tianjin 300387, China
E-mail: liupeituan@163.com

Tie LI
(Corresponding author)
Tiangong University,
School of Computer Science and Technology,
Tianjin 300387, China
E-mail: tielitg@163.com

Rui LI
Tiangong University,
School of Computer Science and Technology,
Tianjin 300387, China
E-mail: lr_1696@126.com

Bo SONG
Tiangong University,
School of Computer Science and Technology,
Tianjin 300387, China
E-mail: 1452254876@qq.com