Deformable CNN With Position Encoding For Arbitrary-Scale Super-Resolution

Yuanbin Ding[1], Kehan Zhu[1], Ping Wei[1], Yu Lin[2] (✉), and Ruxin Wang[3] (✉)

[1] National Pilot School of Software, Engineering Research Center of Cyberspace, Yunnan University, Kunming, China
{dyb2000,zhukehan03}@foxmail.com, weip@ynu.edu.cn
[2] Kunming Institute of Physics, Kunming, China
lwlinyu@163.com
[3] Alibaba Group, Beijing, China
rosinwang@gmail.com

Abstract. Implicit neural representation (INR) has been widely used to learn continuous representation of images, as it enables arbitrary-scale super-resolution (SR). However, most existing INR-based arbitrary-scale SR methods simply concatenate neighboring features and directly stack the position information with the image features, without fully exploiting the correlations among the input information. This processing method may produce artifacts and erroneous texture in the SR image. To address this problem, we propose a deformable CNN with position encoding (DCPE). Our method consists of three main components: (1) Deformable Feature Unfolding (DFU) module, which selectively concatenates the image features to ensure accurate recovery of texture; (2) Fusion With Learned Position Encoding (FPE) module, which generates position encoding that can be better fused with image features, thereby enhancing the correlation between them; and (3) Deep ResMLP module, which enhances the representation capability of the local implicit image function to focus more on learning the high-frequency information of the image, thus reducing the generation of artifacts in SR image. We conduct extensive experiments and demonstrate that our method outperforms previous methods in both qualitative and quantitative evaluations.

Keywords: arbitrary-scale super-resolution · implicit neural representation · deformable CNN · position encoding.

1 Introduction

Single-image super-resolution is a fundamental computer vision task that aims to recover a low-resolution (LR) image into a corresponding high-resolution (HR) image. Most SR methods use convolutional neural networks to extract features and append an upsampling module at the end, which can reconstruct the LR image and generate a high-quality HR image. However, these traditional SR methods often have limitations: they can only perform SR on a fixed scale. In practical scenarios, the limitations of single-scale SR methods become apparent as they inadequately cater to the multi-faceted demands of real-life applications, and training a dedicated model for each scale

is impractical. Hence, the proposal of arbitrary-scale SR methods is necessary, as they can achieve SR at arbitrary scales with only one model.

Most of the existing arbitrary-scale SR methods achieve their goals by preserving the backbone network of traditional SR models while substituting the original standard up-sampling module with one capable of arbitrary-scale up-sampling. This is a simple and effective way to transform the SISR method into an arbitrary-scale SR method and improve the SR performance of the original network. In Meta-SR[13], the Meta-Upscale Module is proposed to replace the traditional up-sampling module and achieve SR at arbitrary scales. However, Meta-SR exhibits limited generalization ability when confronted with large-scale SR scenarios beyond its training scope. To overcome the limitation of Meta-SR, implicit neural representation is introduced by LIIF for arbitrary-scale super-resolution, which parameterizes the signal as a continuous function and maps the coordinates to the corresponding signals. Based on this idea, LIIF proposes a local implicit image function that replaces the traditional up-sampling module. The local implicit image function employs a multi-layer perceptron (MLP) to map the 2D coordinates and the local features to the RGB values. Since the coordinates are continuous values, it can naturally achieve arbitrary-scale SR, even for scales not seen during training.

It is noteworthy that the input to the local implicit image function consists exclusively of the position information of the target pixel and its corresponding feature vector. Consequently, the processing method of position information and feature vectors is crucial for restoring high-quality SR images. However, LIIF [6] simply stacks them together, resulting in a limited correlation between the stacked components, which may lead to the distortion of image texture. UltraSR [38] and IPE [24] enrich the position information by combining it with periodic encoding, but the periodic encoding is fixed and may not be optimal for different scales. As for the processing method of feature vectors, some previous methods [6, 20] concatenate all the features within a 3×3 neighborhood. However, this approach may aggregate some irrelevant or redundant features that could negatively impact texture recovery.

In order to address these problems, we propose a deformable CNN with position encoding, named DCPE. Unlike previous methods [6, 20], we exploit the available information more effectively, concatenate the extracted LR feature information correctly, and fuse the processed feature information with the learned position encoding deeply to enhance the correlation between different types of information. We also use a residual-structured MLP to enhance the representation capability of the local implicit image function, thereby improving the quality of the SR image.

Our principal contributions can be summarized as follows:

- We propose a deformable CNN with position encoding (DCPE) for arbitrary-scale SR, which can concatenate feature vectors that are useful for recovering texture, while deeply fusing the learned position encoding with the image features to obtain SR images with correct texture.
- We propose Deep ResMLP, which optimizes the MLP structure by combining local and global residual connections. This approach enables the network to learn the high-frequency information of the image more effectively, thus reducing the artifacts in the output.

– We conduct extensive experiments on DIV2K and four other benchmark datasets, demonstrating that DCPE outperforms previous methods in most cases.

2 Related Work

2.1 Implicit neural representation

Many natural signals (e.g. images, shapes of objects, etc.) are continuous, but computers can only use discrete storage and representation methods. To overcome the physical limitations of computers and to connect with the continuous representation of the real world, implicit neural representation has attracted increasing attention and research [8, 26, 27] due to its excellent ability to represent continuous signals. Implicit neural representation is a method that approximates a continuous function with a neural network, typically using an MLP to map 2D/3D coordinates to the signals at that location. When an object is modeled as an implicit neural function, the memory required to parameterize the signal depends only on the complexity of the underlying signal, not on the spatial resolution, which greatly enhances the usability of implicit neural representation. Implicit neural representation was initially applied to 3D scenes, such as 3D shape modeling [2, 7, 12], 3D scene modeling [15, 32], and 3D structure rendering [27, 28, 23, 3]. Recently, implicit neural representation has also emerged in 2D applications, such as image SR [6, 38, 20, 24], which can naturally achieve infinite resolution with implicit neural representation, and is of great significance for arbitrary-scale SR.

2.2 Single image super-resolution (SISR)

SISR is the task of transforming LR image into HR image. SR methods can be classified into three categories: interpolation-based, reconstruction-based, and learning-based. Presently, the most effective and influential methods are learning-based methods. Convolutional neural network (CNN) has been widely used in SR reconstruction studies due to their excellent detail characterization ability. CNN can implicitly learn the prior knowledge of the image and use it to generate superior SR outputs. SRCNN [10] was the first CNN-based SR method, which consisted of three convolutional layers. It used bicubic interpolation to upscale the LR image to the target resolution size as input, and then obtained the SR image by applying the SRCNN. Later, ESPCN [31] introduced an efficient sub-pixel convolution layer at the end of the network, which learned a set of upsampling filters to map the LR features to the HR output. This approach avoided using bicubic interpolation to upscale the LR image before feeding the image into the network, which reduced the computational complexity and improved the model performance.

After that, most CNN-based SR methods adopted a similar structure: a backbone network to extract LR image features, followed by an upsampling module to generate SR images. Various new network designs were also proposed for the backbone network, such as VDSR [16], EDSR [22], IRCNN [41] using residual learning; RDN [43] using a combination of dense and residual connections; DRCN [17], DRRN [33] using recurrent networks; SRGAN [19], ESRGAN [36] using generative adversarial networks

to obtain perceptually pleasing texture; RCAN [42], SAN [9] using different attention mechanisms; and IPT [5], SwinIR [21], SwinFIR [40] using transformer structures. While these advancements continue to improve SR performance, most of these methods remain confined to single-scale SR applications, limiting their practical utility in diverse real-world scenarios.

2.3 Arbitrary-scale super-resolution

Arbitrary-scale SR is the task of transforming LR image into HR image at arbitrary scales. Due to the limitations of single-scale SR, arbitrary-scale SR has attracted more attention recently, and the first method to propose it was Meta-SR [13], which used a Meta-SR upsampling module instead of the traditional single-scale upsampling module. This approach enabled the existing SISR methods to adapt to arbitrary-scale SR easily. The Meta-SR upsampling module could dynamically predict the weights of the upsampling filters for any scale factor, and then use them to generate HR images. Inspired by Meta-SR, RSAN [11] and Arb-SR [35] were proposed, which could perform asymmetric SR. Later, Chen et al. [6] proposed the LIIF, which replaced the traditional single-scale upsampling module with a local implicit image function. LIIF took the position information of the target resolution image and the nearest LR feature vector as inputs, and predicted the RGB values at that position. LIIF had better generalization ability for large-scale factors and bridged the gap between 2D discrete and continuous representations.

After LIIF was proposed, many researchers improved it. For example, Xu et al. [38] proposed UltraSR, which deeply integrated spatial coordinates and periodic encoding with the implicit neural representation; Lee et al. [20] proposed a Local Texture Estimator (LTE), which characterized the image texture in 2D Fourier space and enabled the implicit function to reconstruct the image continuously while capturing details; Liu et al. [24] proposed Integrated Position Encoding (IPE), which extended traditional position encoding by aggregating frequency information over pixel regions to enhance the expressiveness of implicit neural networks.

3 Methods

In this section, we present a comprehensive introduction to the novel method called DCPE, which is designed for arbitrary-scale SR. Our method begins by estimating the sampling offsets for each feature reference point using a dedicated offset estimation network. These offsets determine the precise location of sampling points. Bilinear interpolation is subsequently employed to sample features at these specified locations. The sampled features are then concatenated to enrich the feature information of the corresponding reference point. Next, we combine the position information with scale, and obtain the position encoding with the same dimension as the concatenated image features through an MLP. We deeply fuse position encoding with the concatenated feature information. Finally, to improve the expressiveness of the local implicit image function, we increase the depth of the implicit neural network by using global and local residual connections. The overall structure is shown in Figure 1 (a).

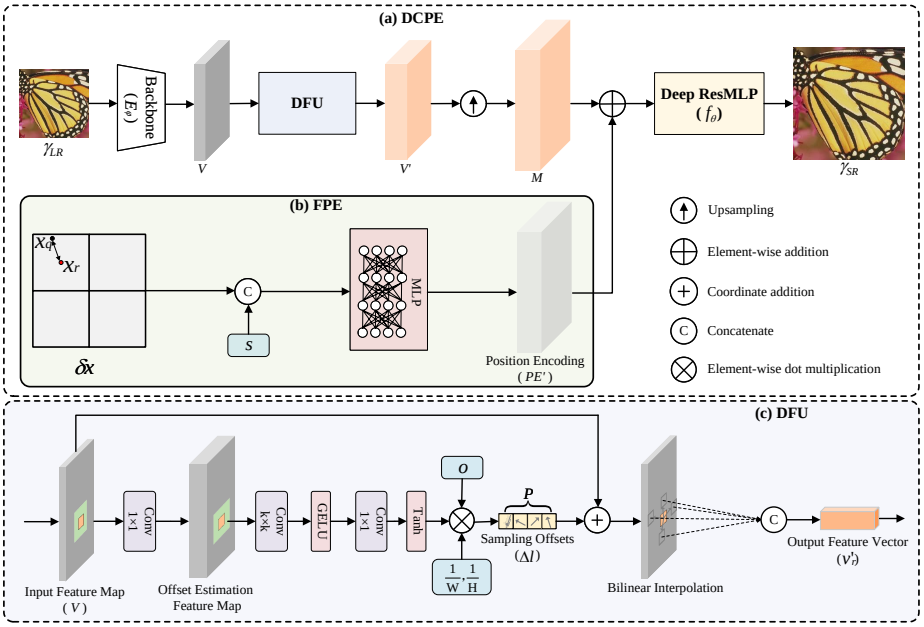

Fig. 1: The overall structure of our proposed deformable CNN with position encoding (DCPE) is illustrated in sub-figure (a). First, the backbone network extracts the LR image features and obtains V. Then, the DFU module (sub-figure (c) shows the detailed structure) selectively samples the features from V to get information-rich features V', and upsample V' to obtain M using nearest-neighborhood interpolation. Next, the FPE module (sub-figure (b)) generates the position encoding and fuses it deeply with M, and finally, the SR image is obtained by the Deep ResMLP module.

The feature extraction process of LR images can be defined as follows:

$$V = E_\varphi(\gamma_{LR}), \tag{1}$$

where $\gamma_{LR} \in \mathbb{R}^{H\times W\times 3}$ represents the LR image, and E_φ is the backbone network used by the model. Extracting the features of γ_{LR} through the backbone network, we obtain $V \in \mathbb{R}^{H\times W\times C}$.

We can define our method as follows:

$$\begin{gathered}\gamma_{SR} = f_\theta\left(\emptyset_{DFU}\left(V\right)_\uparrow, \emptyset_{FPE}\left(\delta x, s\right)\right) \\ \delta x \propto x_q - x_r,\end{gathered} \tag{2}$$

where x_q is the coordinate of the query point in the HR image domain, and x_r is the coordinate of the nearest reference point to x_q in the LR image domain. Both x_r and x_q have the value range of $[-1, 1]$, δx signifies the relative distance between x_q and x_r, while s represents the scale factor. $\uparrow$ stands for the nearest-neighborhood interpolation,

and $\gamma_{SR} \in \mathbb{R}^{sH \times sW \times 3}$ denotes the final SR image. $\emptyset_{DFU}(\bullet)$ and $\emptyset_{FPE}(\bullet)$ are both trainable functions. The former serves to enrich the information contained in $v_r \in V$, while the latter is responsible for generating the position encoding. f_θ denotes a local implicit image function parameterized by θ, which maps coordinates to corresponding RGB values. This function is shared by all images.

3.1 Deformable Feature Unfolding (DFU)

We propose the Deformable Feature Unfolding module to enrich the information of V. Unlike LIIF [6], which uses Feature Unfolding to concatenate all neighboring features within a 3×3 range around the reference point, DFU selectively concatenates the feature vectors that can enhance the texture of the super-resolution (SR) image. DFU avoids treating each feature in the range equally and performing simple feature concatenation without discrimination, which solves the problem of incorrect texture in SR images effectively. The detailed structure of DFU is shown in Figure 1 (c).

Specifically, we introduce an offset estimation network that predicts multiple sampling offsets for each reference point, inspired by the Deformable Attention Transformer (DAT)[37]. We first apply a 1×1 convolution on the input feature map V to change its dimension and obtain the offset estimation feature map. For each reference point, we use a $k \times k$ convolution layer to extract feature information within the $k \times k$ range around the reference point (the light green part of the offset estimation feature map, padded with zero vectors outside the boundary), which contributes to generating the final sampling offsets. Then, we use a GELU activation layer and a 1×1 convolution to get the sampling offsets $\Delta L \in \mathbb{R}^{P\times2}$, P stands for the number of sampling points, which can be defined as follows:

$$\Delta L = \{(\Delta x_p, \Delta y_p)\}_{p\in\{1,2,3,...,P\}}, \tag{3}$$

where Δx_p and Δy_p represent the offsets of the p-th sampling point along the x-axis and y-axis, respectively. To maintain training stability and avoid excessively large offsets, we employ the Tanh activation function to constrain sampling offsets within the range of $[-1, 1]$. Subsequently, the Δx and Δy components of all sampling offsets are normalized by the width (W) and height (H) of the input feature maps, respectively. We also multiply them by an offset range factor o to control the range of the offsets, and finally obtain a reasonable range of the sampling offsets $\Delta l \in \mathbb{R}^{P\times2}$. It is defined as follows:

$$\Delta l = o \cdot \tanh(\frac{\Delta L}{(W, H)}). \tag{4}$$

Subsequently, we determine each location of sampling points according to the sampling offsets and the location of the reference point, and use bilinear interpolation to obtain the feature vectors at locations of sampling points. We concatenate P sampled feature vectors to get new feature vectors. This process is defined as follows:

$$v'_r = \text{Concat}(\{V_{lr+\Delta l_{rp}}\}_{p\in\{1,2,3,...,P\}}), \tag{5}$$

where l_r denotes the coordinate of a feature reference point in the LR image domain, and Δl_{rp} denotes one of its sampling offsets. Then $l_r + \Delta l_{rp}$ is one of its sampling

coordinates, and $V_{lr+\Delta l_{rp}}$ is a feature vector obtained by bilinear interpolation at that position. A new feature vector ${v'}_r$ is obtained after applying Equation 5, and we perform this operation for each feature reference point in the LR image domain to obtain $V' \in \mathbb{R}^{H \times W \times PC}$. V' enlarges the receptive field, selectively concatenates the information around V_{l_r}, discards useless or even harmful information, enhances the content within each feature vector, and facilitates the restoration of texture. Finally, we upsample the latent representation V' using nearest-neighbor interpolation to obtain $M \in \mathbb{R}^{sH \times sW \times PC}$.

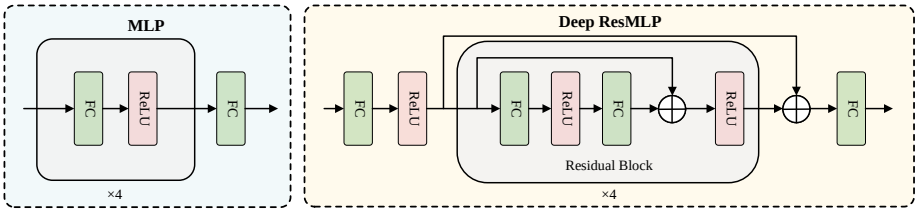

Fig. 2: The figure shows the original MLP in LIIF [6] (left) and our proposed Deep ResMLP (right) structure.

3.2 Fusion With Learned Position Encoding (FPE)

The core idea of implicit neural representation is to map position information to RGB values, so the representation of position information is crucial. Previous work [6, 30] simply stacked the position information with the feature vectors, which resulted in low correlation and unequal information amount between them. This uneven information distribution makes it challenging for the local implicit image function to exploit the relationship between them effectively. Meanwhile, to solve the problem that the fixed encoding method is difficult to optimize, we introduce an approach that seamlessly integrates learned position encoding with image features inspired by the Transformer [34].

Figure 1 (b) illustrates our method. The position information we use is δx, which represents the relative distance between x_q and x_r. This is similar to previous work [6, 38, 30], but we employ distinct encoding and combination methods for the δx. We input δx into a three-layer MLP to obtain a position encoding $PE \in \mathbb{R}^{sH \times sW \times PC}$ with the same dimension as M. Then we add PE to M. This method balances the information amount of the position and the image features, and enhances their association.

However, the position information can only inform the network about the orientation and distance of the target pixel with respect to the LR feature reference points, failing to convey how much space the pixel should occupy within the entire SR image. This deficiency may affect the overall structure of the SR image. Therefore, we introduce a scale factor, and stack it with δx before feeding them into MLP to obtain $PE' \in \mathbb{R}^{sH \times sW \times PC}$. Finally, we add PE' to M and feed it into $f_\theta(\cdot)$.

Table 1: Quantitative results on the DIV2K validation set (PSNR (dB)). The table compares the performance of several arbitrary-scale SR methods. EDSR-baseline [22] uses models trained at specific scales, and other methods use the same model at all scales ($\times 2$ - $\times 30$). † indicates that the method is implemented by [20]. Bold indicates the best performance.

Method	In-scale			Out-of-scale				
	$\times 2$	$\times 3$	$\times 4$	$\times 6$	$\times 12$	18	$\times 24$	$\times 30$
Bicubic [22]	31.01	28.22	26.66	24.82	22.27	21.00	20.19	19.59
EDSR-baseline [22]	34.55	30.90	28.94	-	-	-	-	-
EDSR-baseline-Meta-SR [13]	34.64	30.93	28.92	26.61	23.55	22.03	21.06	20.37
EDSR-baseline-LIIF [6]	34.66	30.96	29.00	26.75	23.71	22.17	21.18	20.48
EDSR-baseline-LTE [20]	34.72	31.02	29.04	26.81	23.78	22.23	21.24	20.53
EDSR-baseline-DCPE (ours)	**34.78**	**31.07**	**29.11**	**26.87**	**23.84**	**22.33**	**21.34**	**20.66**
RDN-Meta-SR [13]	35.00	31.27	29.25	26.88	23.73	22.18	21.17	20.47
RDN-LIIF [6]	34.99	31.26	29.27	26.99	23.89	22.34	21.31	20.59
RDN-LTE [20]	35.04	31.32	29.33	27.04	23.95	22.40	21.36	20.64
RDN-DCPE (ours)	**35.06**	**31.33**	**29.34**	**27.07**	**24.00**	**22.47**	**21.45**	**20.76**
SwinIR-Meta-SR† [13]	35.15	31.40	29.33	26.94	23.80	22.26	21.26	20.54
SwinIR-LIIF† [6]	35.17	31.46	29.46	27.15	24.02	22.43	21.40	20.67
SwinIR-LTE [20]	**35.24**	**31.50**	**29.51**	**27.20**	24.09	22.50	21.47	20.73
SwinIR-DCPE (ours)	35.23	31.49	29.50	**27.20**	**24.11**	**22.55**	**21.54**	**20.82**

Table 2: Quantitative results on benchmark datasets (PSNR (dB)). The table compares the performance of several arbitrary-scale SR methods. each method uses the same model at all scales ($\times 2 - \times 10$). All arbitrary-scale SR methods use EDSR-baseline as backbone. Bold indicates the best performance.

Dataset	Method	In-scale			Out-of-scale		
		$\times 2$	$\times 3$	$\times 4$	$\times 6$	$\times 8$	$\times 10$
Set5	EDSR-baseline-Meta-SR [13]	37.99	34.38	32.05	28.69	26.72	25.42
	EDSR-baseline-LIIF [6]	37.99	34.40	32.18	28.95	26.98	25.61
	EDSR-baseline-LTE [20]	**38.04**	34.43	32.24	28.97	27.04	25.69
	EDSR-baseline-DCPE (ours)	38.03	**34.48**	**32.27**	**29.03**	**27.05**	**25.72**
Set14	EDSR-baseline-Meta-SR [13]	33.61	30.27	28.51	26.31	24.79	23.69
	EDSR-baseline-LIIF [6]	33.57	30.33	28.63	26.45	24.92	23.83
	EDSR-baseline-LTE [20]	**33.72**	**30.37**	28.65	26.50	**24.99**	23.88
	EDSR-baseline-DCPE (ours)	33.71	**30.37**	**28.68**	**26.53**	24.98	**23.90**
B100	EDSR-baseline-Meta-SR [13]	32.17	29.09	27.54	25.74	24.69	23.95
	EDSR-baseline-LIIF [6]	32.16	29.11	27.59	25.84	24.80	24.06
	EDSR-baseline-LTE [20]	32.21	29.14	27.62	25.87	24.82	**24.08**
	EDSR-baseline-DCPE (ours)	**32.22**	**29.15**	**27.64**	**25.88**	**24.83**	**24.08**
Urban100	EDSR-baseline-Meta-SR [13]	32.05	28.10	25.94	23.58	22.28	21.40
	EDSR-baseline-LIIF [6]	32.09	28.17	26.12	23.75	22.44	21.54
	EDSR-baseline-LTE [20]	32.29	28.32	26.24	23.85	22.53	21.64
	EDSR-baseline-DCPE (ours)	**32.33**	**28.36**	**26.28**	**23.88**	**22.56**	**21.65**

3.3 Deep ResMLP

We found that an MLP formed by simple concatenation of fully connected layers and activation functions has limited expressive power and struggles to effectively map coordinates to RGB values, as illustrated in the left side of Figure 2. Therefore, we introduce the Deep ResMLP network, which adds residual connections to increase the network depth and allows the network to focus more on learning high-frequency information. Deep ResMLP contains multiple residual blocks, each with two fully connected layers, two activation layers and a short residual connection, as shown by the gray boxes on the right side of Figure 2. In addition, within the entire Deep ResMLP network, a long residual connection spans all residual blocks, with fully connected layers positioned before and after it. The overall structure is shown on the right side of Figure 2. The experimental results show that our Deep ResMLP can effectively enhance the expressive power of the local implicit image function, achieve superior SR performance, and reduce the generation of artifacts in SR images.

4 Experiments

4.1 Datasets and Metrics

The DIV2K dataset [1] contains 1000 images with 2K resolution, divided into 800 for training, 100 for testing, and 100 for validation. We trained all models on the training set of the DIV2K dataset. To evaluate the model performance, we used the validation set of DIV2K, as well as four benchmark datasets: Set5 [4], Set14 [39], B100 [25], and Urban100 [14]. We used peak signal-to-noise ratio (PSNR) as our evaluation metric. Following previous methods [13, 6, 35, 38, 20, 24], we calculated PSNR values for the three RGB channels of the DIV2K validation set, and for the Y channel of the YCbCr format of the benchmark datasets.

4.2 Implementation detail

Most of our implementation settings are the same as LTE [20]. The training scale factors s are uniformly distributed in $\mathcal{U}(1, 4)$, which we call In-scale, and the scale factors larger than $\times 4$ are called Out-of-scale. To obtain the training image pairs, we randomly crop the HR image in DIV2K to a size of $48s \times 48s$ and then use the bilinear interpolation of PyTorch [29] to downscale the cropped image to 48×48. We use the obtained 48×48 LR image as the input to the model. In the training stage, to ensure that the shape of ground-truths in a batch is the same as that of LR images, and to reduce the memory consumption and accelerate the training speed, we randomly sample 48×48=2304 pixels in the cropped HR image, and record coordinates of each sampled pixel in HR image domain. During training, we only upsample the feature information on the sampled coordinates, and then perform backpropagation. We choose Adam [18] as the optimizer with betas of 0.9 and 0.999, respectively, and L1 loss [22] to train our models. All models are trained on an NVIDIA RTX 4090 24GB GPU for 1000 epochs, and the batch size of the models is set to 8. We use the CNN-based models with the upsampling module removed as the backbone. For the model with SwinIR [21] as the backbone, the initial

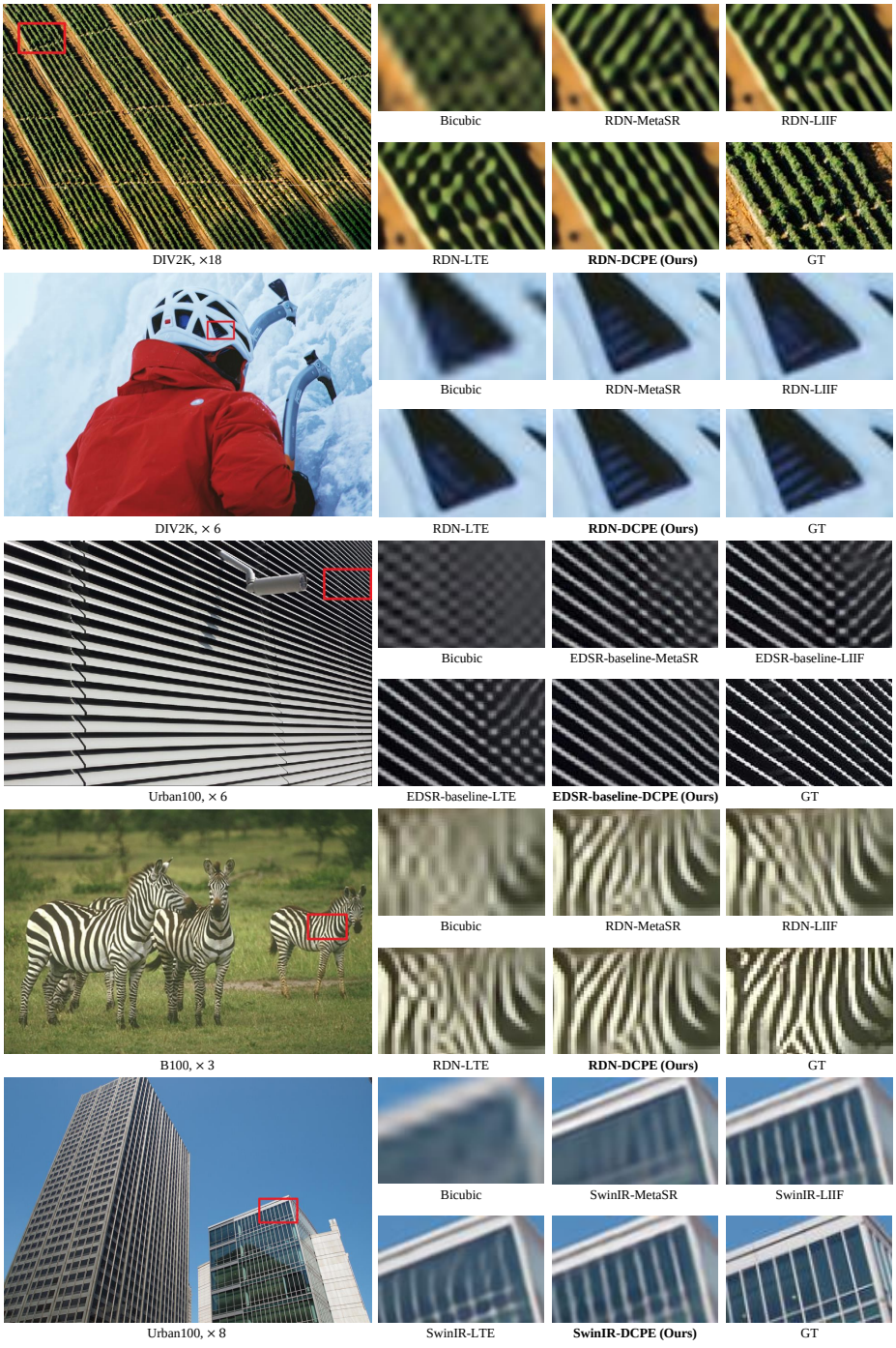

Fig. 3: Qualitative comparison of different arbitrary-scale SR methods. In the large image on the left side, red boxes indicate the selected area for comparison, and the source dataset of the images and the corresponding scale factor used for the comparison are labeled below the large image. The smaller images on the right side display detailed SR images generated by each method, with the respective backbone and method name indicated below each small image.

learning rate is set to 2e-4 and decayed by a factor of 0.5 at epochs of [500, 800, 900, 950], respectively. For the models with EDSR-baseline [22] or RDN [43] as the backbone, the initial learning rate is set to 1e-4 and decays to half of the previous learning rate for every 200 epochs of training.

4.3 Evaluation

Quantitative results. Table 1 compares the performance of several arbitrary-scale SR methods (Meta-SR [13], LIIF [6], LTE [20], and our DCPE) on the DIV2K validation set, using EDSR-baseline [22], RDN [43], and SwinIR [21] as the backbone respectively. The table shows that our model outperforms the other models at all scales with EDSR-baseline [22] and RDN [43] as backbones. With SwinIR [21] as the backbone, our model is slightly lower than LTE [20] by 0.01 dB in the in-scale distribution, but achieves the best performance in the out-of-scale distribution with a maximum difference of 0.09 dB ($\times$30). DCPE also has a more significant improvement in SR performance at large scales regardless of the backbone used, especially at $\times$30 with EDSR-baseline as the backbone, where DCPE achieves a maximum improvement of 0.13 dB over LTE. Table 2 compares the performance of each method on benchmark datasets. Our method achieves superior performance in most cases compared to other methods.

Qualitative results. Figure 3 shows the qualitative analysis of the benchmark dataset and the DIV2K dataset. The figure demonstrates that our method generates SR images with more accurate texture and fewer artifacts than other arbitrary-scale SR methods, both in the in-scale and out-of-scale distributions. The figure also compares various methods using EDSR-baseline, RDN or SwinIR as the backbone, and reveals that our method significantly outperforms other methods on various datasets, regardless of the backbone. This excellent performance can be attributed to our proposed DCPE, which effectively mitigates distortion of texture and maximizes the recovery of texture from the ground-truths.

Table 3: Quantitative ablation study on module validity validation of DCPE. We evaluated the results on the DIV2K validation set (PSNR (dB)), and used EDSR-baseline [22] as the backbone for all DCPE models. DCPE(-D) denotes the model without the DFU module, DCPE(-F) denotes the model without the FPE module, and DCPE(-R) denotes the model without the Deep ResMLP module.

Method	In-scale			Out-of-scale		
	$\times$2	$\times$3	$\times$4	$\times$6	$\times$12	$\times$18
DCPE	**34.78**	**31.07**	**29.11**	**26.87**	**23.84**	**22.33**
DCPE(-D)	34.74	31.03	29.06	26.82	23.78	22.25
DCPE(-F)	34.75	31.05	29.09	26.85	23.83	22.32
DCPE(-R)	34.72	31.02	29.06	26.83	23.81	22.30

4.4 Ablation Study

In this section, we conduct a series of experiments to demonstrate the effectiveness of the various modules of DCPE and to select the optimal parameters for the model, using EDSR-baseline [22] as the backbone for all models.

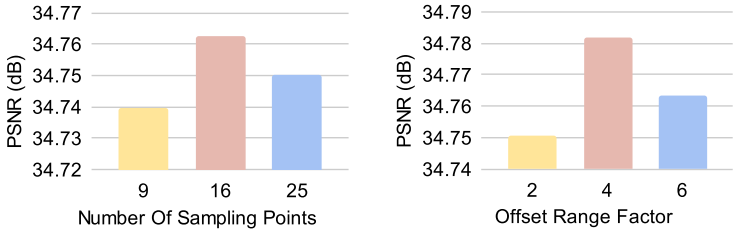

Fig. 4: This figure shows the effect of the number of sampling points (left) and the offset range factor (right) in the DFU module on DIV2K ($\times 2$).

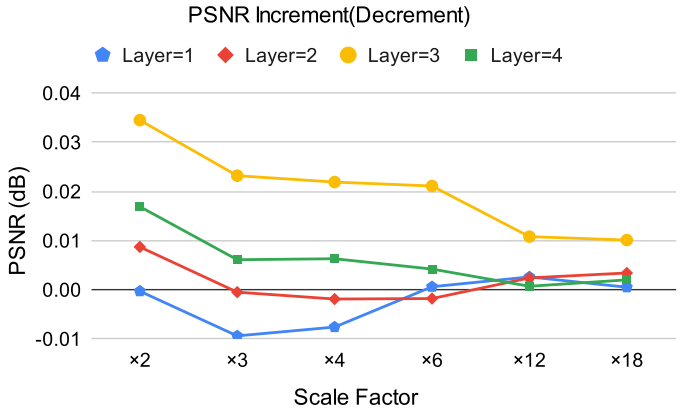

Fig. 5: This figure shows the change in model performance when the number of layers of MLP in the FPE module is 1, 2, 3, and 4, respectively. The increment/decrement refers to the difference between the results from these ablation studies and the DCPE(-F) in Table 3.

Module validity validation. We conducted a series of experiments to assess the effectiveness of different modules within DCPE. We removed various modules individually and retrained new models with the remaining components kept unchanged. Table 3 shows the results. Regarding implementation details, for DCPE(-D), we applied nearest-neighbor interpolation to upsample V to $M' \in \mathbb{R}^{sH \times sW \times PC}$, then fused

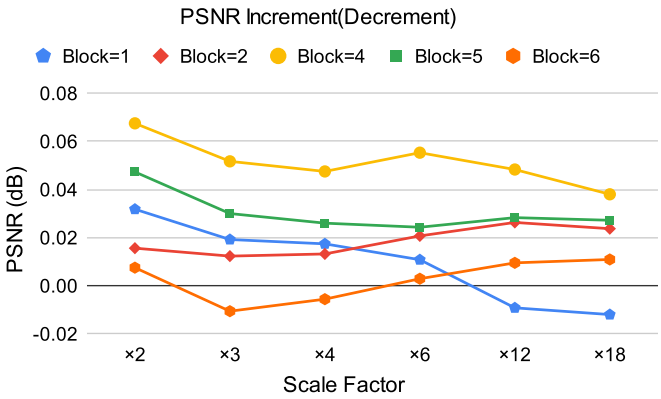

Fig. 6: This figure shows the change in model performance when the number of residual blocks in the Deep ResMLP module is 1, 2, 4, 5, and 6, respectively. The increment/decrement refers to the difference between the results from these ablation studies and the EDSR-baseline-LIIF in Table 1.

M' with PE' and fed it to f_θ; for DCPE(-F), we stacked the position information (δx) and scale factor together on M and fed it to f_θ; for DCPE(-R), we removed the Deep ResMLP module and used the same decoder settings as LIIF [6], that is, the network in Figure 2 (left) as f_θ. The experimental findings reveal that each module has a positive impact on SR performance. Notably, the DFU module exhibits the most significant enhancement effect in the out-of-scale distribution, and the Deep ResMLP module notably improves SR performance in the in-scale distribution.

Parameter Selection. We conducted a series of individual ablation studies to determine the optimal parameters for the DFU module, the FPE module, and the Deep ResMLP module. For the DFU module, we investigated the impact of varying the number of sampling points P and the offset range factor o on model performance, as illustrated in Figure 4. The results indicate that the model performs significantly better when employing sixteen sampling points, and the optimal offset range factor is four. To produce the final sampling offsets that cover the $k \times k$ region around the reference point, we set $k = o + 1$. Hence, we use a convolutional layer with a kernel size of 5×5 to extract the offset estimation feature map, i.e., $k = 5$. For the FPE module, we varied the number of layers in MLP and measured the model performance. Figure 5 shows the results. The figure indicates that the model with three layers of MLP performs better than the others in all scales. For the Deep ResMLP module, we concentrated on assessing the impact of the number of residual blocks on model performance. We only kept the Deep ResMLP module and removed the other two modules for this experiment, so we compared our results with EDSR-baseline-LIIF [6] in Table 1. Figure 6 shows the results. The results revealed that employing four residual blocks consistently yielded superior performance across all scales. We used the optimal parameters from these experiments to get the final results in previous experiments.

5 Conclusion

In this paper, we have proposed a deformable CNN with position encoding for arbitrary-scale super-resolution. Our network correctly concatenates image feature information using the DFU module, and obtains the position encoding with the same dimension as the image features through the FPE module, which promotes the fusion of position information with image features. We also use deep residual connections to improve the expressive power of the local implicit image function. We conducted extensive experiments on the DIV2K and benchmark datasets. The experimental results demonstrate that our method achieves superior SR performance compared to other arbitrary-scale SR methods in both quantitative and qualitative assessments.

Acknowledgments. This work is supported in part by the National Natural Science Foundation of China under Grant 62101480 and 62162067, the Yunnan Foundational Research Project under Grant No. 202201AU070034 and No. 202201AT070173, Research and Application of Object detection based on Artificial Intelligence, in part by the Yunnan Province expert workstations under Grant202305AF150078.

References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 126–135 (2017)
2. Atzmon, M., Lipman, Y.: Sal: Sign agnostic learning of shapes from raw data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2565–2574 (2020)
3. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5855–5864 (2021)
4. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding (2012)
5. Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., Gao, W.: Pre-trained image processing transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12299–12310 (2021)
6. Chen, Y., Liu, S., Wang, X.: Learning continuous image representation with local implicit image function. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8628–8638 (2021)
7. Chen, Z., Tagliasacchi, A., Zhang, H.: Bsp-net: Generating compact meshes via binary space partitioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 45–54 (2020)
8. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5939–5948 (2019)
9. Dai, T., Cai, J., Zhang, Y., Xia, S.T., Zhang, L.: Second-order attention network for single image super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11065–11074 (2019)

10. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence **38**(2), 295–307 (2015)
11. Fu, Y., Chen, J., Zhang, T., Lin, Y.: Residual scale attention network for arbitrary scale image super-resolution. Neurocomputing **427**, 201–211 (2021)
12. Genova, K., Cole, F., Vlasic, D., Sarna, A., Freeman, W.T., Funkhouser, T.: Learning shape templates with structured implicit functions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7154–7164 (2019)
13. Hu, X., Mu, H., Zhang, X., Wang, Z., Tan, T., Sun, J.: Meta-sr: A magnification-arbitrary network for super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1575–1584 (2019)
14. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5197–5206 (2015)
15. Jiang, C., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T., et al.: Local implicit grid representations for 3d scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6001–6010 (2020)
16. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1646–1654 (2016)
17. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1637–1645 (2016)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4681–4690 (2017)
20. Lee, J., Jin, K.H.: Local texture estimator for implicit representation function. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1929–1938 (2022)
21. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1833–1844 (2021)
22. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 136–144 (2017)
23. Liu, L., Gu, J., Zaw Lin, K., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. Advances in Neural Information Processing Systems **33**, 15651–15663 (2020)
24. Liu, Y.T., Guo, Y.C., Zhang, S.H.: Enhancing multi-scale implicit learning in image super-resolution with integrated positional encoding. arXiv e-prints pp. arXiv–2112 (2021)
25. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. vol. 2, pp. 416–423. IEEE (2001)
26. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4460–4470 (2019)

27. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
28. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3504–3515 (2020)
29. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
30. Sarmad, M., Ruspini, L., Lindseth, F.: Photo-realistic continuous image super-resolution with implicit neural networks and generative adversarial networks. In: Proceedings of the Northern Lights Deep Learning Workshop. vol. 3 (2022)
31. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1874–1883 (2016)
32. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. Advances in Neural Information Processing Systems **32** (2019)
33. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3147–3155 (2017)
34. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
35. Wang, L., Wang, Y., Lin, Z., Yang, J., An, W., Guo, Y.: Learning a single network for scale-arbitrary super-resolution. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4801–4810 (2021)
36. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: Proceedings of the European conference on computer vision (ECCV) workshops. pp. 0–0 (2018)
37. Xia, Z., Pan, X., Song, S., Li, L.E., Huang, G.: Vision transformer with deformable attention. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4794–4803 (2022)
38. Xu, X., Wang, Z., Shi, H.: Ultrasr: Spatial encoding is a missing key for implicit image function-based arbitrary-scale super-resolution. arXiv preprint arXiv:2103.12716 (2021)
39. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7. pp. 711–730. Springer (2012)
40. Zhang, D., Huang, F., Liu, S., Wang, X., Jin, Z.: Swinfir: Revisiting the swinir with fast fourier convolution and improved training for image super-resolution. arXiv preprint arXiv:2208.11247 (2022)
41. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep cnn denoiser prior for image restoration. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3929–3938 (2017)
42. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European conference on computer vision (ECCV). pp. 286–301 (2018)
43. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2472–2481 (2018)