sensors

MDPI

Article

SR-FEINR: Continuous Remote Sensing Image Super-Resolution Using Feature-Enhanced Implicit Neural Representation

Jinming Luo [1], Lei Han [1], Xianjie Gao [2], Xiuping Liu [1] and Weiming Wang [1,*]

[1] School of Mathematics and Science, Dalian University of Technology, Dalian 116024, China
[2] Department of Basic Sciences, Shanxi Agricultural University, Jinzhong 030801, China
* Correspondence: wwmdlut@dlut.edu.cn

Abstract: Remote sensing images often have limited resolution, which can hinder their effectiveness in various applications. Super-resolution techniques can enhance the resolution of remote sensing images, and arbitrary resolution super-resolution techniques provide additional flexibility in choosing appropriate image resolutions for different tasks. However, for subsequent processing, such as detection and classification, the resolution of the input image may vary greatly for different methods. In this paper, we propose a method for continuous remote sensing image super-resolution using feature-enhanced implicit neural representation (SR-FEINR). Continuous remote sensing image super-resolution means users can scale a low-resolution image into an image with arbitrary resolution. Our algorithm is composed of three main components: a low-resolution image feature extraction module, a positional encoding module, and a feature-enhanced multi-layer perceptron module. We are the first to apply implicit neural representation in a continuous remote sensing image super-resolution task. Through extensive experiments on two popular remote sensing image datasets, we have shown that our SR-FEINR outperforms the state-of-the-art algorithms in terms of accuracy. Our algorithm showed an average improvement of 0.05 dB over the existing method on $\times 30$ across three datasets.

Keywords: remote sensing image super-resolution; implicit neural representation; position encoding

Citation: Luo, J.; Han, L.; Gao, X.; Liu, X.; Wang, W. SR-FEINR: Continuous Remote Sensing Image Super-Resolution Using Feature-Enhanced Implicit Neural Representation. *Sensors* **2023**, *23*, 3573. https://doi.org/10.3390/s23073573

Academic Editor: Gemine Vivone

Received: 22 February 2023
Revised: 18 March 2023
Accepted: 25 March 2023
Published: 29 March 2023

1. Introduction

With the development of satellite image processing technology, the application of remote sensing has increased [1–5]. However, low spatial, spectral, radiometric, and temporal resolutions of current image sensors and complicated atmospheric conditions make it hard to use remote sensing. Consequently, extensive super-resolution (SR) methods have been proposed to improve the low quality and low resolution of remote sensing images.

SR reconstruction is a method used for generating high-resolution remote sensing images, which combines a large number of images with similar content. Generally, remote sensing image SR reconstruction algorithms can be classified into three categories: single remote sensing image SR reconstruction [6–11], multiple remote sensing image SR reconstruction [12,13], and multi/hyperspectral remote sensing image SR reconstruction [14]. Since the latter two approaches have poor SR effects, registration fusion, multi-source information fusion, and other issues, more research studies have been focusing on single remote sensing image SR reconstruction.

Single remote sensing image SR (SISR) methods can be divided into two categories based on the generative adversarial network and the convolution neural network. Although both GAN-based networks and CNN-based networks can achieve good results in SISR, they can only scale the low-resolution (LR) image with an integer factor, which makes the obtained high-resolution (HR) image inconvenient for downstream tasks. One way to solve this problem is to represent a discrete image continuously with implicit neural

representation. Continuous image representation allows recovering arbitrary resolution imaging by modeling the image as a function defined in a continuous domain. For a continuous domain, the best way to describe an image is to fit this image as a function of continuous coordinates. Our method is motivated by recent advances in implicit neural representation for 3D shape reconstruction [15]. The concept behind implicit functions is to represent a signal as a function that maps coordinates to the corresponding signal (e.g., signed distance to a 3D object surface). In remote sensing image super-resolution, the signals can be the RGB values of an image. Multi-layer perceptron (MLP) is a common way to implement implicit neural representation. Instead of fitting unique implicit functions for each object, encoder-based approaches are suggested to predict a latent code for each item in order to share information across instances. The implicit function is then shared by all objects, and it accepts the latent code as an extra input. Although the encoder-based implicit function method is effective in a 3D challenge, it can only successfully represent simple images and is unable to accurately represent remote sensing images.

To solve the problem of the expression ability of encoder-based implicit neural representations, this paper explores different positional encoding methods in image representation for the image SR task, and proposes a novel feature-enhanced MLP network to enhance the approximation ability of the original MLP. Our main contributions are as follows:

1. We are the first to adopt the implicit neural representation into remote sensing image SR tasks. With our method, one can obtain significant improvements in AID and UC Merced datasets.
2. We propose a novel feature-enhanced MLP architecture to make use of the feature information of the low-resolution image.
3. The performances of different positional encoding methods are investigated in implicit neural representations for continuous remote sensing image SR tasks.

2. Related Works

In this section, we will briefly review the implicit neural representation and the related methods, including positional encoding and continuous image SR.

2.1. Implicit Neural Representation

The implicit neural representation is essentially a continuously differentiable function that maps the coordinates into the signals. It has been widely used in many fields, such as shape parts [16,17], objects [18–21], or scenes [22–25]. The implicit neural representation is a data-driven method. It is trained from some form of data as a signal distance function. Many 3D-aware image generation methods use convolutional architectures. Park et al. [18] proposed using neural networks to fit scalar functions for the representation of 3D scenes. Mildenhall et al. [26] proposed a neural radiance field (Nerf) to implicitly represent a scene. It takes images of the same scene taken from different viewpoints as inputs and uses a neural network to learn a static 3D scene implicitly. Based on these images, the trained neural network can render images from any perspective. However, the present work based on implicit neural representation does not perform very well in the spatial and temporal derivatives. In terms of image generation, Chen et al. [27] proposed a local implicit image function (LIIF). It feeds the coordinates and the features corresponding to the MLP and outputs a RGB signal for the coordinates. Since the coordinates of images with arbitrary resolution are continuous, LIIF can represent images with arbitrary resolutions.

2.2. Positional Encoding

In order to capture the positional relationships, a method called positional encoding is introduced in [28,29]. Positional encoding is essentially a map from a position space to a high-dimensional vector space. For the continuous image SR task, 2D image coordinates are mapped into high-dimensional vectors. The common method used in [29] employs sinusoidal positional encoding by manually designing. The performance of the hand-designed approach depends on the weights of the sinusoidal positional encoding,

which lacks flexibility. In order to improve the flexibility of the positional encoding, Parmar et al. [30] introduced a learnable embedding vector for each position for 1D cases. Although the trainable embedding method has the potential to capture more complex positional relationships, the learnable parameters are largely increased with the increasing dimensionality of the positional input coordinates. For the purpose of capturing more complex position relationships, for instance, the similarity of positions in an image, a novel learnable positional encoding was proposed in [31]. In their proposed method, a function is learned to map multi-dimensional positions into a vector space based on the Fourier transform. The obtained vectors are fed into the MLP. In our work, we will also focus on the learnable positional encoding method.

2.3. Continuous Image SR

Image SR is a reconstruction task that restores a realistic and more detailed high-resolution image from a LR image. It is an important class of computer vision image processing techniques. However, it is an ill-posed problem because a specific LR image corresponds to a set of possible high-resolution images. Due to the powerful characterization and extraction capabilities of deep learning in both low-resolution and high-resolution spaces, deep learning-based image SR tasks have significantly improved in both qualitative and quantitative terms. Dong et al. [32] were the first to research single natural image SR based on deep learning, called SRCNN. It uses a bicubic interpolation to scale a LR image to a target size. Then, these images are fed into a three-layer convolutional network to fit a nonlinear map. The output is a HR image. In [33], a novel network, FSRCNN, was proposed to improve the inference speed of SRCNN. However, the SRCNN model not only learns how to generate high-frequency information, but it also needs to reconstruct low-frequency information, which greatly reduces its efficiency. Kim et al. [34] proposed VDSR to increase the depth of the network by employing the residual connect. Remote sensing images are different from natural images, as they often have coupled objects and environments, and the images span a wide range of scales. In order to make full use of the environmental information, Lei et al. [35] proposed a VDSR-based network called a local–global combined network (LGCNet).

It is evident that all the methods mentioned above upsample the input LR images before feeding them into the model for learning, which slows down the convergence speed of the model and also greatly increases the memory overhead. The ESPCN model [36] proposed a sub-pixel convolution operation as an efficient, fast, and non-parametric pixel rearrangement upsampling method, which significantly improved the training efficiency of the network. To further improve the expressive power of the model, the SRResNet model was proposed in [37], which utilized the residual module widely used in image classification tasks. At the same time, the confrontational generation loss function was first adopted to the image SR problem, which achieved satisfactory results. In [38], the EDSR model was proposed to further optimize the above network structure. Additionally, the performance of the EDSR model was further improved by removing the batch normalization layer and the second activation layer from the residual module. Later, several models were proposed to enhance the network's performance, including the RDN model [39] and the RCAN model [40]. To adaptively fuse the extracted multi-scale information, Wang et al. [41] proposed an adaptive multiscale feature fusion network for SR of remote sensing images.

However, the above methods can only upsample an image to a specific scale. To generate the HR image of arbitrary resolution, MetaSR, ref. [42] introduced a meta-upscale module, which employs a single model to upsample the input image to arbitrary resolution by dynamically predicting weights. However, it cannot achieve satisfactory results for the resolutions outside of the training distribution. Therefore, Chen et al. [27] proposed a local implicit image function (LIIF) by taking advantage of the neural implicit representation. In their method, the coordinates and the features corresponding are fed to the MLP to obtain a RGB signal. Since the coordinates are continuous, the HR image can be presented in arbitrary resolution. However, LIIF ignores the influence of positional encoding on

image generation. Therefore, in this work, the coordinate was encoded to obtain more high-dimensional information about the coordinates, which can produce more realistic HR images. Figure 1 shows the results of our method, which can scale the input image into an arbitrary resolution.

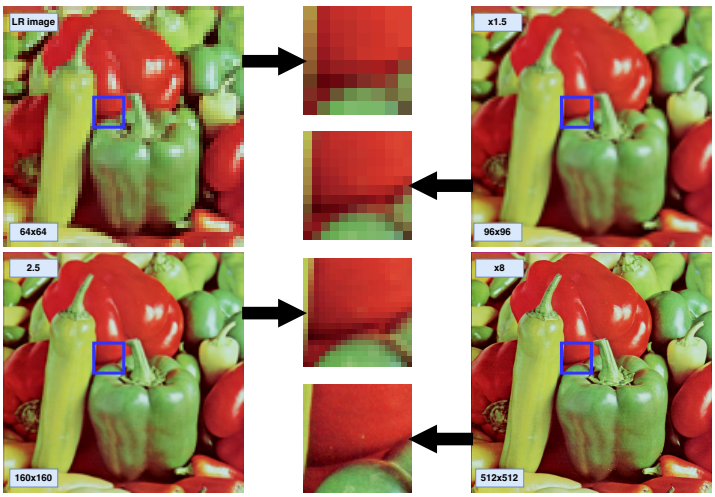

Figure 1. An image in the continuous domain can be presented in arbitrary high resolution.

3. Method

Image SR is a common task in computer vision that outputs a high-resolution image I_{H} based on the input LR image I_{L}. In other words, for each continuous coordinate $\mathbf{p}$ in the high-resolution image I_{H}, we need to calculate a signal at this coordinate, denoted as $c_{\mathbf{p}}$. In the image SR task, the signal for a coordinate is the RGB value. In the following section, we will introduce the details of our method.

3.1. Network Overview

The main part of the proposed network is illustrated in Figure 2. It is composed of three major components: the feature extraction module (E_ψ), the positional encoding module (E_ϕ), and the feature-enhanced MLP module (M_θ).

For a given discrete image $I \in \mathbb{R}^{H\times W\times 3}$, we define the coordinate bank B_I as a subset of $[-1,1]^2$:

$$\begin{aligned} B_I = \quad & \{(x,y)|x \in \{-1+\frac{1}{H}, -1+\frac{3}{H}, \cdots, 1-\frac{1}{H}\}, \\ & y \in \{-1+\frac{1}{W}, -1+\frac{3}{W}, \cdots, 1-\frac{1}{W}\}\} \end{aligned} \tag{1}$$

For a LR image I_{L}, the feature extraction module E_ψ is used to extract the features $F \in \mathbb{R}^{(\#B_{I_{\mathrm{L}}})\times l}$ of the LR image. For a coordinate $\mathbf{p} \in B_{I_{\mathrm{H}}}$ in a HR image I_{H}, the feature at $\mathbf{p}$ can be set as the nearest point feature in $B_{I_{\mathrm{L}}}$, which can be formulated as:

$$f_{\mathbf{p}} = F_{\mathbf{q}^*},\ \mathbf{q}^* = \underset{\mathbf{q}\in B_{I_{\mathrm{L}}}}{\arg\min}\, d(\mathbf{p},\mathbf{q}). \tag{2}$$

The positional encoding module E_ϕ is used to encode the coordinate $\mathbf{p}$ into a high-dimensional space. The output encoding vector at this position is formulated as:

$$g_{\mathbf{p}} = \mathtt{concat}(E_\phi(\mathbf{p}), \mathbf{p}). \tag{3}$$

We will discuss the performances of three commonly used positional encoding methods in Section 5.2.

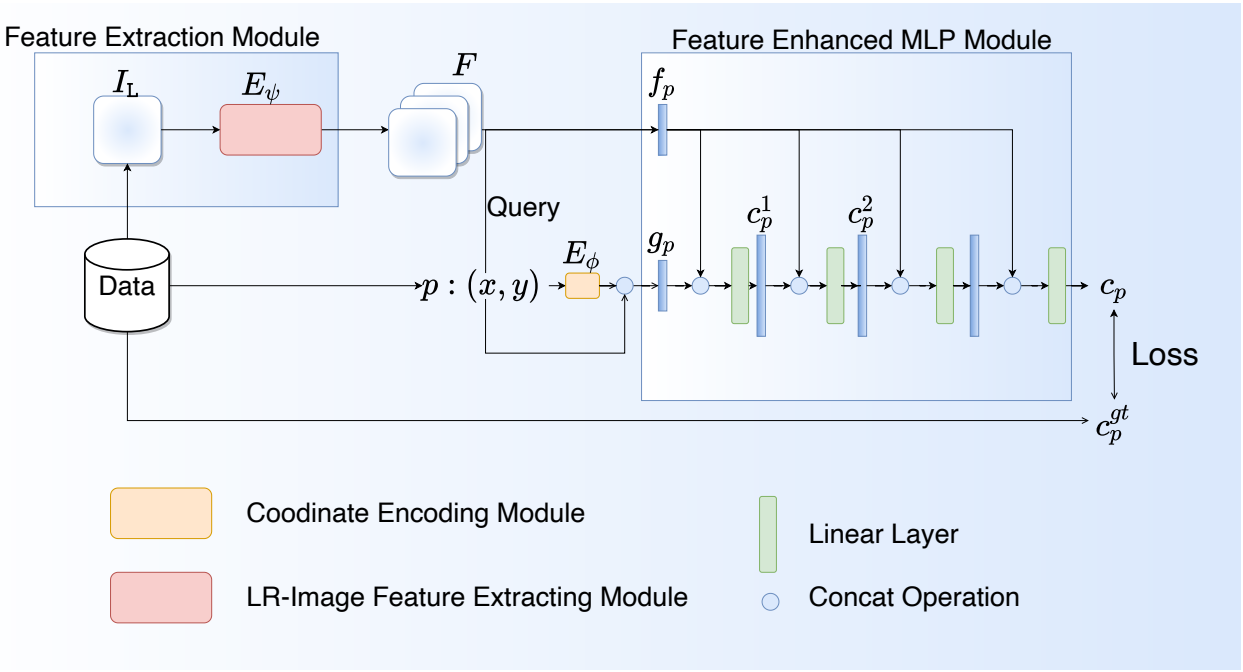

Figure 2. The architecture of the proposed model. The blue rectangles indicate the feature vectors corresponding to the coordinates.

With the feature $f_{\mathbf{p}}$ and the encoding vector $g_{\mathbf{p}}$, the feature-enhanced MLP module M_θ is used to reconstruct the signal $c_{\mathbf{p}}$, which can be formulated as:

$$c_{\mathbf{p}} = M_\theta(f_{\mathbf{p}}, g_{\mathbf{p}}). \tag{4}$$

Consequently, for any coordinate $\mathbf{p} \in \mathcal{P}$, $\mathcal{P}$ is the set of coordinates $\mathbf{p}$ in the high-resolution image I_{H}, and the L1 loss is used as the reconstruction loss:

$$L = \sum_{\mathbf{p}\in\mathcal{P}} ||c_{\mathbf{p}} - c_{\mathbf{p}}^{gt}||_1^2, \tag{5}$$

The complete training and inference processes are presented in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1: Training process of continuous super-resolution using SR-FEINR.

Input: A low-resolution image I_{L}, a high-resolution image I_{H}
Output: A trained model M_θ

Initialize the parameters of the model M_θ
Extract features F from I_{L} using the feature extractor E_ψ
Encode the coordinates of I_{H} using the position encoder E_ϕ
for $\mathbf{p} \in B_{I_H}$ **do**
    Find the nearest point $\mathbf{q}^*$ in $B_{I_{\mathrm{L}}}$ to $\mathbf{p}$ using a distance metric d
    Set the feature at $\mathbf{p}$ to $f_{\mathbf{p}} = F_{\mathbf{q}^*}$
    Set the encoding vector at $\mathbf{p}$ to $g_{\mathbf{p}} = E_\phi(\mathbf{p})$
Update the parameters of the model M_θ using stochastic gradient descent with the following loss function:

$$L = \frac{1}{|B_{I_{\mathrm{H}}}|} \sum_{\mathbf{p}\in B_{I_{\mathrm{H}}}} ||M_\theta(f_{\mathbf{p}}, g_{\mathbf{p}}) - c_{\mathbf{p}}^{gt}||_1^2,$$

where $c_{\mathbf{p}}^{gt}$ is the ground-truth signal value at coordinate $\mathbf{p}$

Algorithm 2: Inference process of continuous super-resolution using SR-FEINR

Input: A low-resolution image I_{L}
Output: A reconstructed high-resolution image $\hat{I}_{\mathrm{H}}$
Define coordinate bank B_I for images I_{L} and $\hat{I}_{\mathrm{H}}$
Extract features F from I_{L} using feature extractor E_ψ
Encode the coordinates of $\hat{I}_{\mathrm{H}}$, using position encoder E_ϕ
for $\mathbf{p} \in B_{\hat{I}_H}$ **do**
    Find the nearest point $\mathbf{q}^*$ in $B_{I_{\mathrm{L}}}$ to $\mathbf{p}$ using a distance metric d
    Set the feature at $\mathbf{p}$ to $f_{\mathbf{p}} = F_{\mathbf{q}^*}$
    Set the encoding vector at $\mathbf{p}$ to $g_{\mathbf{p}} = E_\phi(\mathbf{p})$
    Reconstruct the signal at $\mathbf{p}$ using M_θ: $c_{\mathbf{p}} = M_\theta(f_{\mathbf{p}}, g_{\mathbf{p}})$
Construct a high-resolution image $\hat{I}_{\mathrm{H}}$ from signals $c_{\mathbf{p}}$
return $\hat{I}_{\mathrm{H}}$

3.2. Feature Extraction Module and Positional Encoding Module

3.2.1. Feature Extraction

As mentioned in [27], we used EDSR and RDN to extract the features of the low-resolution image. The feature extraction process in EDSR includes inputting a low-resolution image, extracting high-level features through convolutional layers, enhancing features through residual blocks, fusing features through feature fusion modules, and outputting a feature map. The feature extraction process in RDN includes inputting a low-resolution image, extracting feature maps through convolutional layers and residual dense networks, expanding features through feature expansion modules, fusing features through feature fusion modules, and finally upsampling and reconstructing the image.

For a low-resolution image $I_{\mathrm{L}} \in \mathbb{R}^{H \times W \times 3}$, to enrich the information of each latent code in the feature space, we update the features using the feature-unfolding method, which can be formulated as:

$$F_i' = \mathtt{concat}(\{F_j\}_{d(i,j)<\epsilon}). \tag{6}$$

Afterward, we obtain the features of the low-resolution image F; the features of the continuous coordinate $f_{\mathbf{p}}$ can be calculated using Equation (2) and fed into the feature-enhanced MLP module M_θ.

3.2.2. Positional Encoding

To encode the coordinate $\mathbf{p}$, we use the following equation:

$$\begin{aligned} E(\mathbf{p}) = \quad & (\sin(\omega_0 \pi \mathbf{p}), \cos(\omega_0 \pi \mathbf{p}), \sin(\omega_1 \pi \mathbf{p}), \\ & \cos(\omega_1 \pi \mathbf{p}), \cdots \sin(\omega_n \pi \mathbf{p}), \cos(\omega_n \pi \mathbf{p})), \end{aligned} \tag{7}$$

where $\omega_0, \omega_1, \ldots,$ and ω_n are coefficients and n is related to the dimension of the encoding space.

As illustrated in Figure 3, the details of three common positional encoding methods are described, which are the hand-craft approach, the random approach, and the learnable approach. In the hand-craft approach, ω_i is fixed as $\omega_0 = b^0, \cdots, \omega_n = b^L$, where b and L are hyperparameters. The difference between the random approach and the normal positional encoding is that the weights ω_i are randomly selected and not specified. The weights ω_i are sampled from a normal distribution $\mathcal{N}(\mu, \Sigma)$, where μ and Σ are hyperparameters.

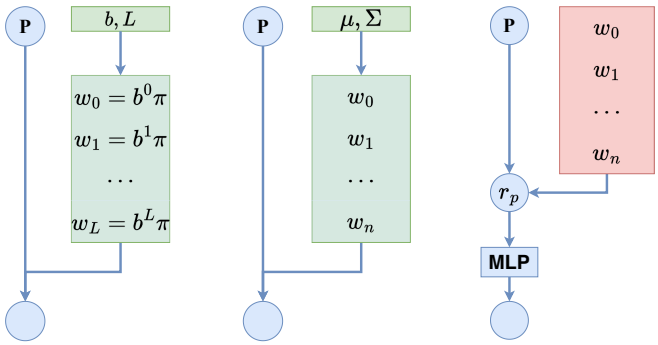

Figure 3. The structures of three positional encoding methods. The blue circle P represents the coordinate. The green rectangles indicate the hyperparameters of the Fourier features. The red rectangle indicates the learnable parameters.

For the learnable approach, the encoding vector of each position is represented as a trainable code by a learnable mapping of the coordinate. A major advantage of this method for multidimensional coordinates is that it is naturally inductive and can handle test samples with arbitrary lengths. Another major advantage is that the number of parameters does not increase with the sequence length. This method is composed of two components: learnable Fourier and a MLP layer. To extract useful features, learnable Fourier features map an M-dimensional position $\mathbf{p}$ into an F-dimensional Fourier feature vector called $r_{\mathbf{p}}$. The definition of learnable Fourier features is roughly the same as Equation (7),

$$r_{\mathbf{p}} = \frac{1}{\sqrt{F}}(\sin(\omega_0\pi\mathbf{p}), \cos(\omega_0\pi\mathbf{p}), \sin(\omega_1\pi\mathbf{p}), \cos(\omega_1\pi\mathbf{p}), \cdots \sin(\omega_n\pi\mathbf{p}), \cos(\omega_n\pi\mathbf{p})), \tag{8}$$

where $\omega_0, \cdots, \omega_n$ are trainable parameters, $n = \frac{F}{2} - 1$ defines both the orientation and wavelength of the Fourier features. The linear projection coefficients $\omega_0, \cdots, \omega_n$ are initialized with a normal distribution $\mathcal{N}(0, \gamma^{-2})$. The MLP layer is a simple neural network architecture for implicit neural representation with a GELU activation function:

$$E_\phi(\mathbf{p}) = \tau(r_{\mathbf{p}}, \eta), \tag{9}$$

where $\tau(.)$ is the perceptron parameterized by η.

Since the weights are learnable, the expression power of the encoding vector is more flexible. Therefore, in our work, we focus on learnable positional encoding.

3.3. Feature-Enhanced MLP for Reconstruction

In order to make use of the information in the LR image, we propose a feature-enhanced MLP module M_θ to reuse the feature of the LR image. The latent code $f_{\mathbf{p}}$ at the coordinate $\mathbf{p}$ of the LR image and the encoded coordinate feature vector $g_{\mathbf{p}}$ are fed into the first hidden layer of the MLP. This process is defined as

$$c^1_{\mathbf{p}} = h_1(f_{\mathbf{p}}, g_{\mathbf{p}}), \tag{10}$$

where h_1 is the first hidden layer of the MLP, $c^1_{\mathbf{p}}$ is the output vector of the first hidden layer.

Then we concatenate the image feature vector $f_{\mathbf{p}}$ with the output feature of the previously hidden layer. At this point, Equation (10) is transformed into

$$c^2_{\mathbf{p}} = h_2(f_{\mathbf{p}}, c^1_{\mathbf{p}}), \tag{11}$$

where h_2 is the second hidden layer of the MLP, $c^2_{\mathbf{p}}$ is the output vector of the second hidden layer.

In our method, the MLP is constructed with five perceptron layers to obtain better results compared to LIIF [27]. The MLP model can be written as:

$$c_{\mathbf{p}} = h_{N-1}(f_{\mathbf{p}}, h_{N-2}(f_{\mathbf{p}}, h_{N-3}(f_{\mathbf{p}}, \cdots, h_1(f_{\mathbf{p}}, g_{\mathbf{p}})))), \tag{12}$$

where $h_i(.)$ is the ith hidden layer and $c_{\mathbf{p}}$ is the predicted RGB value for coordinate $\mathbf{p}$.

3.4. Implementation Details

Two feature extraction modules are considered in this work, which are EDSR and RDN. In the three positional encoding approaches, we chose the learnable positional encoding because it was more conducive to the learning of the network and it performed better in our experiment. As for the MLP setting of the feature-enhanced MLP network M_θ, we chose a five-layer 256-d multilayer perceptron (MLP) with the GELU activation function.

4. Experiments

4.1. Experimental Dataset and Settings

In our experiment, we used a common dataset DIV2K [43] for the ablation study and two common remote sensing datasets: UC Merced [44] and AID [45]. In the field of remote sensing SISR, these datasets have been heavily utilized [35,46,47].

- AID dataset [45]: This dataset contains 30 classes of remote sensing scenes, such as an airport, railway station, square, and so on. Each class contains hundreds of images with a resolution of 600 × 600. In our experiment, we chose two types of scenes, an airport and a railway station, to evaluate different methods. The images in each scene were split into the train set and test set with a ratio of 8:2, and then we randomly picked five images from the train set as the valid set for each scene.
- UC Merced Dataset [44]: This dataset contains 21 classes of remote sensing scenes, such as an airport, baseball diamond, beach, and so on. Each class contains 100 images with a resolution of 256 × 256. We split the dataset into the train set, test set, and valid set with a ratio of 4:5:1.
- DIV2K dataset[43] : This dataset contains 1000 high-resolution natural images and corresponding LR images with scales ×2, ×3, and ×4. We used 800 images as the training set and 100 images in the DIV2k validation set as the test set, which followed prior work [27].

In our training process, the low-resolution image I_{L} and the coordinate-RGB pairs $O = \{(\mathbf{p}, c_{\mathbf{p}})\}_{\mathbf{p}\in A}$ of the high-resolution image can be obtained by the following steps: (1) the high-resolution image in the training dataset is cropped into a $48r_i \times 48r_i$ patch I_{P}, where r_i is sampled from a uniform distribution $U(1, 4)$; (2) I_{P} is downsampled with the bicubic interpolation method to generate its LR image I_{L} with a resolution of 48 × 48; (3) for an original $48r_i \times 48r_i$ image patch I_{P}, the coordinate bank is constructed $B_{I_{\mathrm{P}}}$. For each coordinate $\mathbf{p} \in B_{I_{\mathrm{P}}}$, its RGB value is denoted as $c_{\mathbf{p}}$. Then, the coordinate–RGB pair set I_{P} is constructed as $O^{\mathtt{full}} = \{(\mathbf{p}, c_{\mathbf{p}})\}_{\mathbf{p}\in B_{I_{\mathrm{P}}}}$; 4) the 48 × 48 coordinate–RGB pairs $O = \{(\mathbf{p}, c_{\mathbf{p}})\}_{\mathbf{p}\in A}$ are randomly chosen from $O^{\mathtt{full}}$ to evaluate the network.

We implemented SRCNN, VDSR, and LGCNet based on the settings given in [48]. For other experiments, we adapted the same training settings given in [27]. Specifically, we used the Adam optimizer [49] with an initial learning rate 1×10^{-4}. All of the experiments were trained for 1000 epochs with a batch size of 16, and the learning rate decayed by a factor of 0.5 every 200 epochs.

4.2. Evaluation Metrics

To evaluate the effectiveness of the proposed method, two commonly used evaluation indicators were used in [50–53]. The most popular method for evaluating the quality of outcomes is PSNR (the peak signal-to-noise ratio). For a RGB image, the PSNR can be calculated as follows:

$$PSNR = 10\log_{10}\left(\frac{255^2 \times N_p}{MSE}\right). \tag{13}$$

where N_p is the total number of pixels in the image and MSE is the mean squared error, which can be calculated as:

$$MSE = \frac{1}{3N_p}\sum_{i=1}^{N_p}\sum_{c=1}^{3}[I(i)_c - K(i)_c]^2$$

where $I(i)_c$ and $K(i)_c$ represent the intensity values of the ith pixel in the original and reconstructed images in the cth color channel, respectively.

The structural similarity index (SSIM) can be used to measure the similarity between two RGB images. The SSIM index can be calculated as follows:

$$\text{SSIM}(I, K) = \frac{(2\mu_I\mu_K + c_1)(2\sigma_{IK} + c_2)}{(\mu_I^2 + \mu_K^2 + c_1)(\sigma_I^2 + \sigma_K^2 + c_2)} \tag{14}$$

where μ_I, μ_K, σ_I, σ_K, and σ_{IK} are the mean, standard deviation, and cross-covariance of the intensity values of the original and reconstructed images in the three color channels, respectively. The constants c_1 and c_2 are small positive constants to avoid instability when the denominator is close to zero. Note that the above equations assume that the original and reconstructed RGB images have the same resolution. If the images have different resolutions, they need to be resampled before calculating PSNR and SSIM.

5. Results and Analysis

In this section, we compare our method with several state-of-the-art image super-resolution methods, including the bicubic interpolation, SRCNN [32], VDSR [34], LGCNet [35], EDSR[38], and two continuous image super-resolution methods, i.e., MetaSR [42] and LIIF [27]. The bicubic interpolation, SRCNN [32], VDSR [34], LGCNet [35], EDSR [38], and RDN [39] depend on the magnified scale. These methods require different models for different upsampling scales during training, i.e., they cannot use the same model for arbitrary SR scales. EDSR-MetaSR, EDSR-LIIF, and EDSR-ours use EDSR as the feature extraction module. RDN-LIIF and RDN-ours use RDN as the feature extraction module.

5.1. Results on the Three Datasets

5.1.1. Comparison Results on the AID Dataset

Since the AID dataset has 30 scene categories, we only randomly selected 2 categories to show the comparison results, which are the airport and the railway station. The results are listed in Table 1 for upscale factors ×2, ×3, ×4, ×6, ×12, and ×18, where the bold text represents the best results. It can be observed that our method obtains competitive results for in-distribution scales compared to the previous methods. For out-of-distribution, our method significantly outperforms the other methods in both the PSNR and SSIM. In addition to the quantitative analysis, we also conducted qualitative comparisons, which are shown in Figures 4 and 5. In Figure 4, the ×3 SR results of a railway station for different methods are shown, where two regions are zoomed in to show the details (see the red and green rectangles). The PSNR values are listed in the left-bottom corner of each image. In Figure 5, we show the ×4 SR results of an airport for different methods. From these figures, we can see that our method has the clearest details and the highest PSNR value.

Table 1. Quantitative comparisons between the AID test set (PSNR (dB) and SSIM). (RS∗: railway station, the bold in table is the highest value).

Dataset	Method	In-Distribution (SSIM↑/PSNR↑)			Out-of-Distribution (SSIM↑/PSNR↑)		
		×2	×3	×4	×6	×12	×18
Airport	Bicubic	0.8887/31.37	0.7949/28.39	0.7187/26.73	-	-	-
	SRCNN [32]	0.8917/31.99	0.8049/28.95	0.7336/27.22	-	-	-
	LGCNet [35]	0.8978/32.43	0.8127/29.19	0.7389/27.34	-	-	-
	VDSR [34]	0.9025/32.72	0.8211/29.56	0.7515/27.70	-	-	-
	EDSR [38]	0.9376/34.67	0.8246/29.08	0.7488/27.44	-	-	-
	EDSR-MetaSR [42]	0.9375/34.71	0.8611/30.95	0.7885/28.83	0.6822/26.47	0.5452/23.57	0.5010/22.28
	EDSR-LIIF [27]	0.9374/34.71	0.8617/30.97	0.7892/28.87	0.6849/26.54	0.5529/23.66	0.5082/22.35
	EDSR-ours	**0.9377/34.72**	**0.8617/31.00**	**0.7899/28.90**	**0.6860/26.58**	**0.5537/23.69**	**0.5091/22.39**
RS∗	Bicubic	0.8863/31.70	0.7753/28.39	0.6801/26.53	-	-	-
	SRCNN [32]	0.8967/32.21	0.7992/29.03	0.7088/27.06	-	-	-
	LGCNet [35]	0.9033/32.58	0.8045/29.18	0.7111/27.11	-	-	-
	VDSR [34]	0.9088/32.88	0.8147/29.52	0.7270/27.50	-	-	-
	EDSR [38]	0.9417/35.19	0.8127/29.04	0.7217/27.30	-	-	-
	EDSR-MetaSR [42]	0.9412/35.18	0.8570/31.11	0.7690/28.76	0.6311/26.09	0.4562/22.93	0.4049/21.63
	EDSR-LIIF [27]	0.9413/35.19	0.8575/31.13	0.7696/28.78	0.6330/26.13	0.4594/22.94	0.4063/21.62
	EDSR-ours	**0.9414/35.20**	**0.8577/31.16**	**0.7711/28.83**	**0.6347/26.18**	**0.4610/23.01**	**0.4076/21.67**

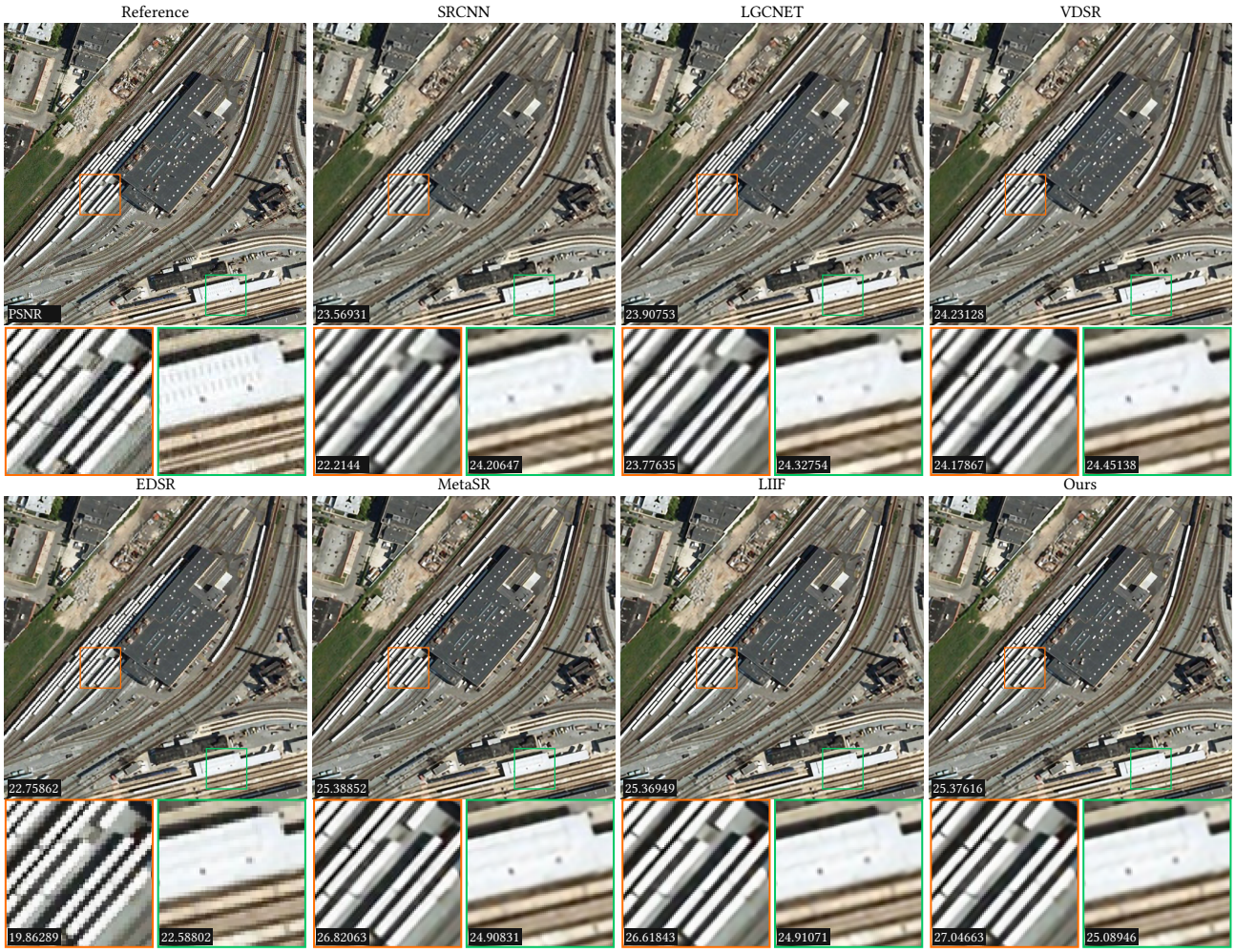

Figure 4. Comparison results of the ×3 scale on the railwaystation_190 scene of the AID dataset. Two local regions are zoomed in to show the detailed results. The PSNR values are listed in the bottom-left corners.

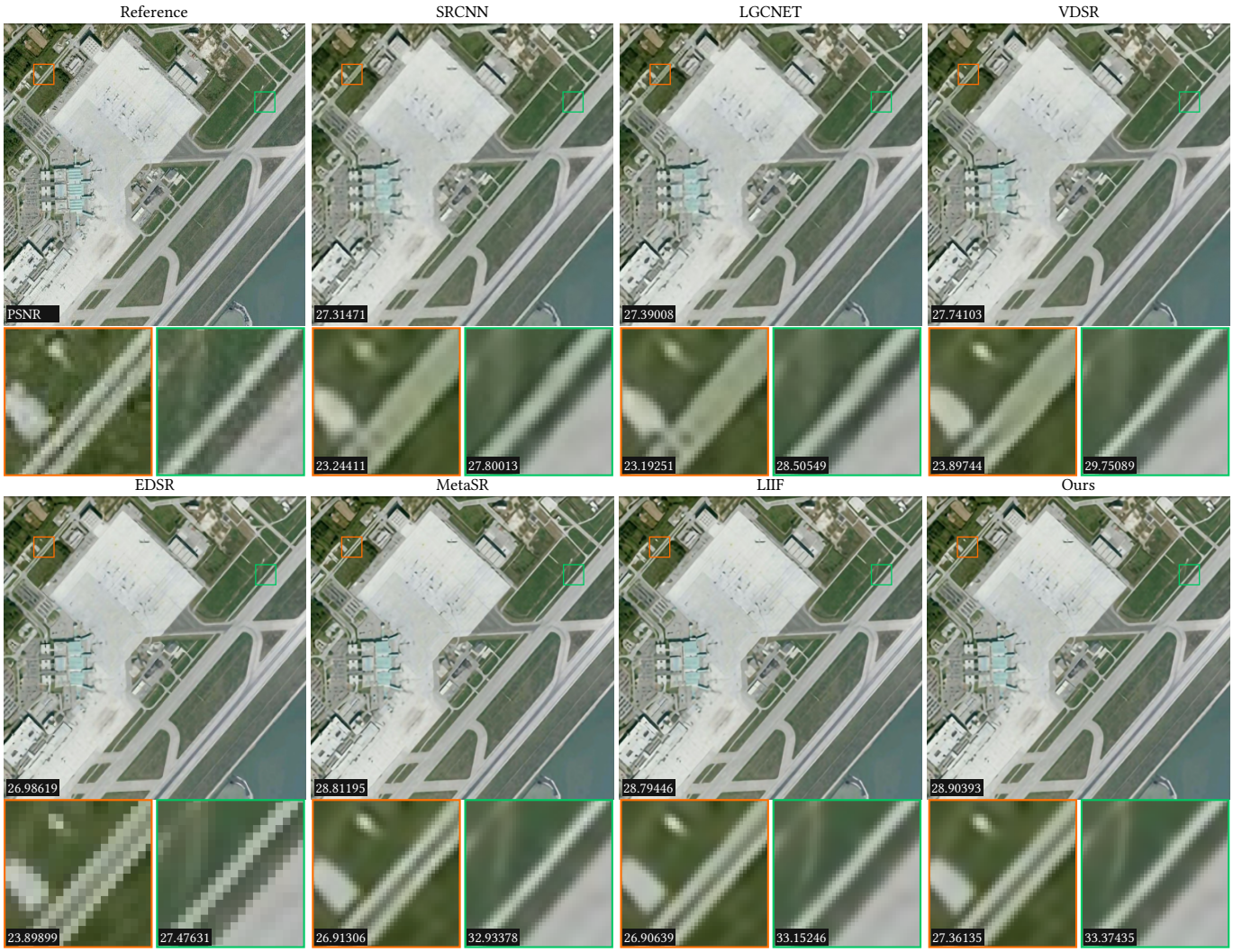

Figure 5. Comparison results of ×4 scale on the Airport_240 scene of the AID dataset. Two local regions are zoomed in to show the detailed results. The PSNR values are listed in the bottom-left corners.

5.1.2. Comparison Results on UCMerced Dataset

Different from the AID dataset,UCMerced dataset has smaller number of images and categories. Therefore, our model is trained and tested on the whole dataset. The quantitative comparison results of these methods on the UCMerced dataset are listed in Table 2. From this table we can see, our results are higher than LIIF at all magnification scales. In addition, we also visualize the SR results for different methods in Figure 6. From a visual point of view, both LIIF and our method outperform the other methods. Although the visualization results of LIIF and our method are similar, the PSNR values of the whole image and the local regions of our method are larger than LIIF, which means our method is slightly better than LIIF.

Table 2. Mean SSIM and PSNR (dB) of the UC Merced dataset(the bold in table is the highest value).

Method	In-Distribution (SSIM↑/PSNR↑)			Out-of-Distribution (SSIM↑/PSNR↑)		
	×2	×3	×4	×6	×12	×18
Bicubic	0.8796/30.79	0.7636/27.47	0.6729/25.66	-	-	-
SRCNN [32]	0.9151/32.76	0.8095/28.83	0.7217/26.74	-	-	-
LGCNet [35]	0.9208/33.20	0.8180/29.09	0.7300/26.93	-	-	-
VDSR [34]	0.9262/33.66	0.8351/29.65	0.7486/27.43	-	-	-
EDSR [38]	0.9246/34.16	0.8158/29.86	0.6932/26.12	-	-	-
EDSR-MetaSR [42]	0.9262/34.43	0.8285/30.22	0.7454/27.91	0.6173/25.23	0.4477/22.13	0.3973/20.89
EDSR-LIIF [27]	0.9260/34.45	0.8285/30.20	0.7445/27.89	0.6185/25.23	0.4510/22.10	0.4005/20.85
EDSR-ours	**0.9259/34.46**	**0.8287/30.26**	**0.7465/27.96**	**0.6202/25.31**	**0.4521/22.20**	**0.4013/20.94**

Figure 6. Comparison results of the ×4 scale on the dense residential_88 scene of the UC Merced dataset. Two local regions are zoomed in to show the detailed results. The PSNR values are listed in the bottom-left corners.

5.1.3. Comparison Results on the DIV2K Dataset

Unlike the above two datasets, the images in the DIV2K dataset are mainly natural. Since our method is proposed for remote sensing image SR, we only conducted the quantitative comparisons on this dataset. In this dataset, we compare two versions of our method with Bicubic, EDSR, EDSR-MetaSR, EDSR-LIIF, and RDN-LIIF. The EDSR-ours and RDN-ours use EDSR and RDN to extract features, respectively. The comparison results are listed in Table 3. From this table, we can see that for EDSR, our method has the best

performance from the ×3 scale. For the ×2 scale, LIIF and EDSR-MeatSR are better than our method as they are trained for this scale. Regarding the RDN, we only compare it with LIIF. The comparison results demonstrate that our method can achieve the best results at high scales.

Table 3. Quantitative comparison on the DIV2K validation set (PSNR (dB)), the bold in table is the highest value.

Method	In-Distribution (PSNR↑)			Out-of-Distribution (PSNR↑)				
	×2	×3	×4	×6	×12	×18	×24	×30
Bicubic	31.01	28.22	26.66	24.82	22.27	21.00	20.19	19.59
EDSR [38]	34.55	30.90	28.94	-	-	-	-	-
EDSR-MetaSR [42]	34.64	30.93	28.92	26.61	23.55	22.03	21.06	20.37
EDSR-LIIF [27]	**34.67**	30.96	29.00	26.75	23.71	22.17	21.18	20.48
EDSR-ours	34.60	**30.97**	**29.02**	**26.78**	**23.75**	**22.22**	**21.23**	**20.53**
RDN-LIIF [27]	**34.99**	**31.26**	29.27	26.99	23.89	22.34	21.31	20.59
RDN-ours	34.88	31.24	**29.28**	**27.01**	**23.93**	**22.38**	**21.35**	**20.63**

5.2. Ablation Study

In this section, we perform ablation studies to assess the effectiveness of each module, where the EDSR is used as the feature encoder. Based on the baseline LIIF model, we progressively add the positional encoding module and feature-enhanced MLP module to evaluate their effectiveness. In order to further evaluate the effectiveness of the proposed feature-enhanced MLP module, we replace the features with coordinates and embed them into the MLP. The results of the ablation study are shown in Table 4. In this table, LIIF is our baseline. LIIF + PE is the combination of LIIF and the positional encoding module. LIIF + PE + FE is the combination of the positional encoding module and the feature-enhanced MLP module, which is our method. Based on LIIF + PE + FE, the features in the feature-enhanced MLP module are replaced with coordinates, and the resulting network is LIIF + PE + PF*. From this table, we can see that LIIF + PE + FE (our method) outperforms the LIIF at all scales except for the ×2 scale. This result proves that the learning ability of the network can be effectively improved by embedding the image features into the hidden layer of the MLP.

Table 4. Quantitative comparison of the ablation study (PSNR(dB)), the bold in table is the highest value.

	In-Distribution (PSNR↑)			Out-of-Distribution (PSNR↑)				
	×2	×3	×4	×6	×12	×18	×24	×30
LIIF [27]	**34.67**	30.96	29.00	26.75	23.71	22.17	21.18	20.48
LIIF + PE	34.53	30.91	28.97	26.73	23.72	22.20	21.21	20.51
LIIF + PE + FE	34.60	**30.97**	**29.02**	**26.78**	**23.75**	**22.22**	**21.23**	**20.53**
LIIF + PE + PF*	34.52	30.91	28.96	26.72	23.71	22.18	21.19	20.50

The positional encoding module is an important module in the proposed method. As described in Section 3.2, there are three commonly used positional encoding methods, which are the hand-craft approach, random approach, and learnable approach. Therefore, in this section, we will discuss the effectiveness of these methods on the remote sensing image SR task. The comparison results are listed in Table 5. In this table, LIIF + PE-hand represents the network with the hand-craft positional encoding method, where $b = 2$ and $L = 10$. i.e., $\omega_i = 2^i, i = 0, 1, \cdots, 9$. LIIF + PE-random shows that the weights are chosen randomly from a normal distribution. In this network, the hyperparameters are set as $\mu = 100$ and $\Sigma = 0$. The LIIF + PE-learning is the network with the learnable positional encoding method. Weights are learned through a MLP. The function $\tau(.)$ is a 2-layer MLP with the GELU activation and hidden dimensions of 256. The dimensions of the Fourier feature vector F are set to 768. γ is set to 10 in the normal distribution $\mathcal{N}(0, \gamma^{-2})$. From Table 5, we can see that LIIF outperforms the other methods for in-distribution scales, which are ×2, ×3, and ×4. However, after the ×6 scale, LIIF + PE + learnable achieves the

best performance among all methods. Therefore, the learnable positional encoding method is used in our network.

Table 5. Quantitative comparison of three different positional encoding approaches in Figure 3 (PSNR(dB)), the bold in table is the highest value.

	In-Distribution (PSNR↑)			Out-of-Distribution (PSNR↑)				
	×2	×3	×4	×6	×12	×18	×24	×30
LIIF [27]	**34.67**	**30.96**	**29.00**	**26.75**	23.71	22.17	21.18	20.48
LIIF + PE-hand	31.65	28.47	26.98	25.07	22.46	21.18	20.35	19.75
LIIF + PE-random	34.56	30.86	28.94	26.70	23.70	22.17	21.19	20.49
LIIF + PE-learnable	34.53	30.91	28.97	26.73	**23.72**	**22.20**	**21.21**	**20.51**

6. Conclusions

In this paper, we propose a novel network structure for continuous remote sensing image SR. By using the LIIF as our baseline, two important modules are introduced to improve its performance, which are the positional encoding module and the feature-enhanced MLP module. The positional encoding module can capture complex positional relationships by using more coordinate information. The feature-enhanced MLP module is constructed by adding prior information from the LR image to the hidden layer of MLP, which can improve the expression and learning ability of the network. Extensive experimental results demonstrate the effectiveness of the proposed method. It is worth noting that our method outperforms the state-of-the-art methods for magnifications outside of the training distribution, which is important in practical applications.

As far as we know, the inference speed of the MLP is a bit slow, which limits the application of our method. In the literature, there are some acceleration algorithms for the MLP architecture, which can be used to decrease the inference time. Therefore, we will attempt to integrate these methods into our algorithm to improve its efficiency.

Author Contributions: Conceptualization, J.L.; Methodology, J.L.; Validation, L.H. and X.G.; Investigation, L.H.; Resources, X.G. and W.W.; Writing—original draft, J.L. and L.H.; Writing—review & editing, W.W. and X.L.; Visualization, X.G.; Supervision, W.W. and X.L.; Project administration, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the National Natural Science Foundation of China (nos. 62172073, 61976040, and 12101378) and the Natural Science Foundation of Liaoning Province (no. 2021-MS-110).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zou, Z.; Chen, C.; Liu, Z.; Zhang, Z.; Liang, J.; Chen, H.; Wang, L. Extraction of Aquaculture Ponds along Coastal Region Using U2-Net Deep Learning Model from Remote Sensing Images. *Remote Sens.* **2022**, *14*, 4001. [CrossRef]
2. Lv, Z.; Huang, H.; Li, X.; Zhao, M.; Benediktsson, J.A.; Sun, W.; Falco, N. Land cover change detection with heterogeneous remote sensing images: Review, progress, and perspective. *Proc. IEEE* **2022**, *110*, 1976–1991. [CrossRef]
3. Meng, X.; Liu, Q.; Shao, F.; Li, S. Spatio–Temporal–Spectral Collaborative Learning for Spatio–Temporal Fusion with Land Cover Changes. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5704116. [CrossRef]
4. Chen, C.; Liang, J.; Xie, F.; Hu, Z.; Sun, W.; Yang, G.; Yu, J.; Chen, L.; Wang, L.; Wang, L.; et al. Temporal and spatial variation of coastline using remote sensing images for Zhoushan archipelago, China. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *107*, 102711. [CrossRef]
5. Meng, X.; Shen, H.; Yuan, Q.; Li, H.; Zhang, L.; Sun, W. Pansharpening for cloud-contaminated very high-resolution remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 2840–2854. [CrossRef]

6. Zhihui, Z.; Bo, W.; Kang, S. Single remote sensing image super-resolution and denoising via sparse representation. In Proceedings of the 2011 International Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping, Xiamen, China, 10–12 January 2011; pp. 1–5.
7. Haut, J.M.; Fernandez-Beltran, R.; Paoletti, M.E.; Plaza, J.; Plaza, A.; Pla, F. A new deep generative network for unsupervised remote sensing single-image super-resolution. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6792–6810. [CrossRef]
8. Zhang, N.; Wang, Y.; Zhang, X.; Xu, D.; Wang, X. An unsupervised remote sensing single-image super-resolution method based on generative adversarial network. *IEEE Access* **2020**, *8*, 29027–29039. [CrossRef]
9. Lei, S.; Shi, Z.; Zou, Z. Coupled adversarial training for remote sensing image super-resolution. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 3633–3643. [CrossRef]
10. Dong, X.; Sun, X.; Jia, X.; Xi, Z.; Gao, L.; Zhang, B. Remote sensing image super-resolution using novel dense-sampling networks. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 1618–1633. [CrossRef]
11. Lei, S.; Shi, Z. Hybrid-scale self-similarity exploitation for remote sensing image super-resolution. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5401410. [CrossRef]
12. Salvetti, F.; Mazzia, V.; Khaliq, A.; Chiaberge, M. Multi-image super-resolution of remotely sensed images using residual attention deep neural networks. *Remote Sens.* **2020**, *12*, 2207. [CrossRef]
13. Arefin, M.R.; Michalski, V.; St-Charles, P.L.; Kalaitzis, A.; Kim, S.; Kahou, S.E.; Bengio, Y. Multi-image super-resolution for remote sensing using deep recurrent networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 206–207.
14. Chen, H.; Zhang, H.; Du, J.; Luo, B. Unified framework for the joint super-resolution and registration of multiangle multi/hyperspectral remote sensing images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 2369–2384. [CrossRef]
15. Chibane, J.; Alldieck, T.; Pons-Moll, G. Implicit functions in feature space for 3d shape reconstruction and completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6970–6981.
16. 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, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7154–7164.
17. Genova, K.; Cole, F.; Sud, A.; Sarna, A.; Funkhouser, T.A. Deep Structured Implicit Functions. *arXiv* **2019**, arXiv:1912.06126.
18. Park, J.J.; Florence, P.; Straub, J.; Newcombe, R.; Lovegrove, S. Deepsdf: Learning continuous signed distance functions for shape representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 165–174.
19. 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, Seattle, WA, USA, 14–19 June 2020; pp. 2565–2574.
20. Michalkiewicz, M.; Pontes, J.K.; Jack, D.; Baktashmotlagh, M.; Eriksson, A. Implicit surface representations as layers in neural networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4743–4752.
21. Gropp, A.; Yariv, L.; Haim, N.; Atzmon, M.; Lipman, Y. Implicit geometric regularization for learning shapes. *arXiv* **2020**, arXiv:2002.10099.
22. Sitzmann, V.; Zollhöfer, M.; Wetzstein, G. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1121–1132.
23. 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, Seattle, WA, USA, 14–19 June 2020; pp. 6001–6010.
24. Peng, S.; Niemeyer, M.; Mescheder, L.; Pollefeys, M.; Geiger, A. Convolutional occupancy networks. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 523–540.
25. Chabra, R.; Lenssen, J.E.; Ilg, E.; Schmidt, T.; Straub, J.; Lovegrove, S.; Newcombe, R. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 608–625.
26. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 405–421.
27. 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, Nashville, TN, USA, 19–25 June 2021; pp. 8628–8638.
28. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. Convolutional sequence to sequence learning. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1243–1252.
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.
30. Parmar, N.; Vaswani, A.; Uszkoreit, J.; Kaiser, L.; Shazeer, N.; Ku, A.; Tran, D. Image transformer. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 4055–4064.

31. Li, Y.; Si, S.; Li, G.; Hsieh, C.J.; Bengio, S. Learnable fourier features for multi-dimensional spatial positional encoding. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 15816–15829.
32. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a deep convolutional network for image super-resolution. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 184–199.
33. Dong, C.; Loy, C.C.; Tang, X. Accelerating the super-resolution convolutional neural network. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 391–407.
34. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 1646–1654.
35. Lei, S.; Shi, Z.; Zou, Z. Super-resolution for remote sensing images via local–global combined network. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1243–1247. [CrossRef]
36. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.
37. 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, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
38. 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, Honolulu, HI, USA, 21–26 July 2017; pp. 136–144.
39. 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, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2472–2481.
40. 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), Munich, Germany, 8–14 September 2018; pp. 286–301.
41. Wang, X.; Wu, Y.; Ming, Y.; Lv, H. Remote sensing imagery super-resolution based on adaptive multi-scale feature fusion network. *Sensors* **2020**, *20*, 1142. [CrossRef]
42. 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, Long Beach, CA, USA, 15–20 June 2019; pp. 1575–1584.
43. 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, Honolulu, HI, USA, 21–26 July 2017; pp. 126–135.
44. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
45. Xia, G.S.; Hu, J.; Hu, F.; Shi, B.; Bai, X.; Zhong, Y.; Zhang, L.; Lu, X. AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3965–3981. [CrossRef]
46. Haut, J.M.; Paoletti, M.E.; Fernandez-Beltran, R.; Plaza, J.; Plaza, A.; Li, J. Remote sensing single-image superresolution based on a deep compendium model. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1432–1436. [CrossRef]
47. Qin, M.; Mavromatis, S.; Hu, L.; Zhang, F.; Liu, R.; Sequeira, J.; Du, Z. Remote sensing single-image resolution improvement using a deep gradient-aware network with image-specific enhancement. *Remote Sens.* **2020**, *12*, 758. [CrossRef]
48. Lei, S.; Shi, Z.; Mo, W. Transformer-Based Multistage Enhancement for Remote Sensing Image Super-Resolution. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–11. [CrossRef]
49. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
50. Huynh-Thu, Q.; Ghanbari, M. Scope of validity of PSNR in image/video quality assessment. *Electron. Lett.* **2008**, *44*, 800–801. [CrossRef]
51. Hore, A.; Ziou, D. Image quality metrics: PSNR vs. SSIM. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 2366–2369.
52. Liu, Y.; Wang, L.; Cheng, J.; Li, C.; Chen, X. Multi-focus image fusion: A survey of the state of the art. *Inf. Fusion* **2020**, *64*, 71–91. [CrossRef]
53. Zhu, Z.; He, X.; Qi, G.; Li, Y.; Cong, B.; Liu, Y. Brain tumor segmentation based on the fusion of deep semantics and edge information in multimodal MRI. *Inf. Fusion* **2023**, *91*, 376–387. [CrossRef]