remote sensing

MDPI

Article

Enhancing Remote Sensing Image Super-Resolution with Efficient Hybrid Conditional Diffusion Model

Lintao Han [1,2], Yuchen Zhao [1,*], Hengyi Lv [1], Yisa Zhang [1], Hailong Liu [1], Guoling Bi [1] and Qing Han [3]

[1] Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China; hanlintao19@mails.ucas.ac.cn (L.H.); lvhengyi@ciomp.ac.cn (H.L.); zhangyisa18@mails.ucas.edu.cn (Y.Z.); liuhailong@ciomp.ac.cn (H.L.); biguoling@ciomp.ac.cn (G.B.)
[2] College of Materials Science and Opto-Electronic Technology, University of Chinese Academy of Sciences, Beijing 100049, China
[3] College of Physics and Telecommunication Engineering, Zhoukou Normal University, Zhoukou 466001, China; hanq@zknu.edu.cn
* Correspondence: zhaoyuchen@ciomp.ac.cn

Abstract: Recently, optical remote-sensing images have been widely applied in fields such as environmental monitoring and land cover classification. However, due to limitations in imaging equipment and other factors, low-resolution images that are unfavorable for image analysis are often obtained. Although existing image super-resolution algorithms can enhance image resolution, these algorithms are not specifically designed for the characteristics of remote-sensing images and cannot effectively recover high-resolution images. Therefore, this paper proposes a novel remote-sensing image super-resolution algorithm based on an efficient hybrid conditional diffusion model (EHC-DMSR). The algorithm applies the theory of diffusion models to remote-sensing image super-resolution. Firstly, the comprehensive features of low-resolution images are extracted through a transformer network and CNN to serve as conditions for guiding image generation. Furthermore, to constrain the diffusion model and generate more high-frequency information, a Fourier high-frequency spatial constraint is proposed to emphasize high-frequency spatial loss and optimize the reverse diffusion direction. To address the time-consuming issue of the diffusion model during the reverse diffusion process, a feature-distillation-based method is proposed to reduce the computational load of U-Net, thereby shortening the inference time without affecting the super-resolution performance. Extensive experiments on multiple test datasets demonstrated that our proposed algorithm not only achieves excellent results in quantitative evaluation metrics but also generates sharper super-resolved images with rich detailed information.

Keywords: remote sensing; image super-resolution; neural network; diffusion model; transformer; feature extraction

Citation: Han, L.; Zhao, Y.; Lv, H.; Zhang, Y.; Liu, H.; Bi, G.; Han, Q. Enhancing Remote Sensing Image Super-Resolution with Efficient Hybrid Conditional Diffusion Model. *Remote Sens.* **2023**, *15*, 3452. https://doi.org/10.3390/rs15133452

Academic Editors: Igor Yanovsky and Jing Qin

Received: 10 June 2023
Revised: 2 July 2023
Accepted: 6 July 2023
Published: 7 July 2023

1. Introduction

Remote-sensing images are captured using optical remote-sensing imaging technologies, such as aircraft and remote-sensing satellites. These images record radiation information on the Earth's surface and find applications in various fields, including environmental monitoring, military target recognition, and land resource exploration [1]. Accurate prediction and analysis in remote-sensing applications require high-resolution images with rich detailed information. However, the resolution of remote-sensing images is often limited by imaging equipment, and factors such as blur, downsampling, noise, and compression further reduce image quality. This results in a reduction in the image resolution and the loss of high-frequency information, which is crucial for effective analysis of the images [2]. Improving hardware equipment in remote-sensing imaging systems is an effective way to solve the problem of low resolution, but it also requires significant additional costs. Therefore, it is necessary to develop practical and cost-effective super-resolution algorithms

to enhance the resolution of remote-sensing images. Super-resolution (SR) algorithms aim to improve image resolution while providing finer spatial details, thus compensating for the weaknesses of satellite images. By enhancing resolution and preserving high-frequency information in images, SR algorithms reduce the dependence on hardware upgrades, thereby improving efficiency and reducing costs [3,4].

Single-image super-resolution (SISR) is a current research hotspot in the field of computer vision [5], aiming to recover high-resolution (HR) images and rich high-frequency information from low-resolution (LR) images. The study of SISR is of great significance to both industry and academia. However, SISR is an ill-posed problem, and due to the loss of high-frequency information, the image super-resolution process involves multi-mapping from the LR to HR space, resulting in multiple solution spaces for any LR input. Existing algorithms aim to determine the correct solution from the solution space. Currently, numerous methods have been proposed for SISR, which can be categorized into three main categories: interpolation-based methods, reconstruction-based methods [6–8], and learning-based methods [5,9–14].

Interpolation-based methods are simple and effective algorithms for SISR. These methods increase the resolution of low-resolution images through interpolation, including nearest-neighbor interpolation, bilinear interpolation, and bicubic interpolation [1]. However, it should be noted that in these interpolation methods, high-frequency information is severely lost during the upsampling process due to the lack of external prior information. Reconstruction-based methods in super-resolution use self-information and prior knowledge of images as constraints to optimize the quality of super-resolved images [6–8]. Although these methods can overcome the limitations of interpolation-based methods, they require manual parameter tuning, have slow convergence speeds, and have high computational costs. Therefore, they may not be suitable for handling complex and diverse scenarios in remote-sensing image applications.

With the improvement in computer performance, the theory of deep learning has flourished in multiple application domains [15,16], and significant progress has been made in deep-neural-network-based super-resolution algorithms [1]. In contrast with the aforementioned methods, learning-based methods represent the mapping relationship between LR and HR remote-sensing images by establishing a neural network learning model. Compared with traditional methods, learning-based methods make use of a large number of LR and HR image pairs as external prior information. Deep convolutional neural networks (CNNs) have strong feature representation capabilities and faster inference speeds and can achieve end-to-end training. Researchers have proposed a series of deep-learning-based SISR algorithms based on CNNs [5,9–14], which show significant improvements in super-resolution performance compared with traditional algorithms. However, CNN-based super-resolution models still face some challenges in remote-sensing image super-resolution tasks. Most CNN models do not consider the complex textures and structures present in remote-sensing images, limiting their ability to recover high-frequency details in super-resolved images. Since the proposal of denoising diffusion probabilistic models (DDPMs) [17], DDPMs have been widely used in many natural scene reconstruction tasks, including super-resolution tasks. Subsequently, researchers have proposed methods to improve DDPMs to address existing problems based on the characteristics of image super-resolution tasks. To address the over-smoothing and mode collapse problems in previous learning-based super-resolution algorithms, Li et al. proposed a diffusion-based method for face super-resolution (SRDiff) [18], which was the first attempt to apply diffusion models to single-image super-resolution. A low-resolution image is used as a conditional input, and the Gaussian noise latent variable is gradually transformed into a super-resolution image through a Markov chain. Additionally, residual prediction was introduced to accelerate the convergence speed of the neural network during practical operations. Liu et al. proposed a detail-complementary generative diffusion model (DMDC) [2] for remote-sensing image super-resolution, which includes detailed supplementary tasks to improve the restoration ability of DMDC. The proposed model solves the problems

of insufficient attention to small targets, lack of model understanding, and detail supplementation in traditional optimization models. However, the above algorithms overlook the importance of input feature conditioning and the ability to maintain details during the training process, resulting in lower-quality super-resolution remote-sensing images and longer inference times when these algorithms are applied to remote-sensing images. To address these challenges, we propose a diffusion-model-based method that leverages the powerful generative capabilities of the diffusion model to reconstruct high-resolution remote-sensing images.

In summary, the main contributions of this paper are as follows:

1. This paper proposes a remote-sensing image super-resolution network based on the diffusion model. By using the comprehensive features of low-resolution images extracted with a transformer network and CNN as conditions to guide image generation, the diffusion model can fully utilize the conditional features to predict the noise data distribution and effectively recover high-resolution images from noise. The powerful generative capability of the diffusion model enables it to fully understand image information, addressing the shortcomings of previous neural-network-based remote-sensing super-resolution methods that typically fail to obtain high-fidelity detailed images at high magnifications.
2. A Fourier high-frequency spatial constraint is proposed to emphasize high-frequency spatial loss and optimize the reverse diffusion direction. By emphasizing high-frequency spatial loss through the Fourier high-frequency spatial constraint, missing high-frequency information in low-resolution remote-sensing images can be restored, significantly improving the quality of remote-sensing image super-resolution. The method can generate more textured and detailed information, while reducing the diversity of the diffusion model, and produce super-resolved images that are closer to the original images, achieving precise detailed information reconstruction.
3. To address the time-consuming issue in the reverse diffusion process of the diffusion model, a feature-distillation-based method is proposed that shortens the inference time without affecting the super-resolution performance.
4. This paper not only tested the proposed algorithm on the commonly used RSOD [19] and UC Merced Land Use [20] remote-sensing image datasets but also verified its effectiveness on the real dataset Gaofen-2 [21]. The experimental results show that our proposed method outperforms other comparable super-resolution algorithms in both quantitative metrics and visual quality.

The rest of this paper is organized as follows. Section 2 briefly introduces the application of CNNs in remote-sensing image super-resolution and the related concepts and research progress of the diffusion model. Section 3 elaborates on our proposed remote-sensing image super-resolution method based on the efficient hybrid conditional diffusion model and the implementation details of each part. Section 4 presents a large number of experimental details and discusses the effectiveness of our proposed method. Finally, Section 5 summarizes the entire paper.

2. Related Work

2.1. Remote-Sensing Image Super-Resolution Based on CNNs

Dong et al. [5] proposed the first three-layer CNN architecture for image super-resolution, known as SRCNN. Subsequently, the emergence of residual networks [22] allowed an increase in the number of network layers, enabling deep neural networks to learn high-level features and reducing training difficulty. Based on residual networks, Kim et al. [9] proposed a 20-layer CNN for image super-resolution, called VDSR. RDN [13] developed a deep network using dense blocks that fully utilized the hierarchical features of all previous layers. Zhang et al. [11] incorporated a channel attention (CA) module into the residual structure using the SE block for inspiration, forming a very deep network called RCAN. Haris et al. [23] proposed DDBPN based on the idea of iterative upsampling and

downsampling, which provides an error feedback mechanism. SRFBN [24] utilizes the hidden state in an RNN to achieve feedback for super-resolution.

Inspired by the successful application of CNNs to traditional images, more and more remote-sensing image super-resolution methods are adopting deep learning techniques and achieving good results. Lei et al. [25] proposed a local–global combined network (LGCNet) based on a CNN for remote-sensing image super-resolution. Inspired by back-projection networks, Pan et al. [26] proposed residual dense projection blocks to enhance the resolution of remote-sensing images. Gu et al. [4] drew inspiration from some emerging concepts in deep learning, such as channel attention, and proposed residual squeeze-and-excitation blocks as building blocks for super-resolution networks. To avoid overfitting and excessive parameters, Chang and Luo et al. [27] introduced bidirectional convolutional long short-term memory layers to learn feature correlations from each recursion.

Due to the ability of generative adversarial networks (GANs) to generate more visually pleasing remote-sensing super-resolution images and achieve better quantitative metrics, GANs have gradually become the backbones of super-resolution networks. Ma et al. [3] proposed a GAN-based method to enhance the resolution of remote-sensing images, called dense residual GAN (DRGAN). Specifically, DRGAN modified the loss function of the reference Wasserstein GAN to improve reconstruction accuracy and avoid gradient vanishing. Jiang et al. [28] also proposed an edge-enhancement network (EEGAN) that utilizes adversarial learning strategies for robust satellite image SR reconstruction, which is particularly good at restoring sharp edges. The diffusion model and GAN model used in this paper differ in terms of image super-resolution. The diffusion model can capture the complex statistical information of the visual world, inferring structures at higher scales than low-resolution inputs. However, GAN models often suffer from mode collapse, resulting in the generated samples lacking diversity. In addition, recent studies have shown that diffusion models based on image conditioning are superior to regression-based models in terms of image super-resolution. Therefore, diffusion models have certain advantages in image super-resolution.

2.2. Diffusion Model

As shown in Figure 1, commonly used generative models include GANs [29], variational autoencoders (VAEs) [30], and normalizing flows (NFs) [31]. Each of these generative models can generate high-quality samples, but each method has its own limitations. GAN models can be unstable during training without careful parameter tuning, and can easily suffer from mode collapse [32] and produce low-quality samples. Samples generated with a VAE with autoencoding structures can be blurry and lack detailed information. Flow-based models require a specialized architecture to construct reversible transformations.

The diffusion model [17,33,34] is also a generative model and is inspired by non-equilibrium thermodynamics. It defines a Markov chain with a diffusion step, gradually adding random noise to the data, and then learns the reverse diffusion process (reverse Markov diffusion chain) to construct the desired data samples from the noise. The learning process of the diffusion model is fixed, and the data dimension of the latent variables is the same as that of the original data.

In recent years, many generative models based on diffusion models have been proposed, including diffusion probability models [33], conditional score models [35], and denoising diffusion probability models (DDPM) [17]. Among them, DDPMs have been widely used in various scenarios, such as image coloring, super-resolution, inpainting, and semantic editing. In 2015, Sohl-Dickstein et al. [33] introduced the diffusion probability model, which gradually destroys the structure of the data distribution during the forward diffusion process and then restores the structure of the data by learning the reverse diffusion process to generate highly flexible and easy-to-handle data generative models. In 2020, Ho et al. [17] proposed the denoising diffusion probability model and demonstrated that the diffusion model could actually generate high-quality samples. The diffusion probability model is a parameterized Markov chain that can be trained using variational

inference. The fractional generative model proposed by Song et al. [36] generates images by solving stochastic differential equations using a neural-network-estimated score function and (Refs. [17,33]) can be regarded as the discrete form of the fractional generative model. Rombach et al. [37] proposed a latent diffusion model that can significantly improve the training and sampling efficiency of denoising diffusion models without reducing the quality of the diffusion model, achieving state-of-the-art results in image patching and class-conditional image synthesis. DiffusionCLIP [38] uses the contrastive language-image pretraining (CLIP) loss and pre-trained diffusion model for text-guided image processing. ILVR [39] proposed a method to guide the DDPM generation process, which can generate high-quality images based on given reference images. CCDF [40] proposed starting from a single forward diffusion with better initialization, which can significantly reduce the number of sampling steps for reverse conditional diffusion.

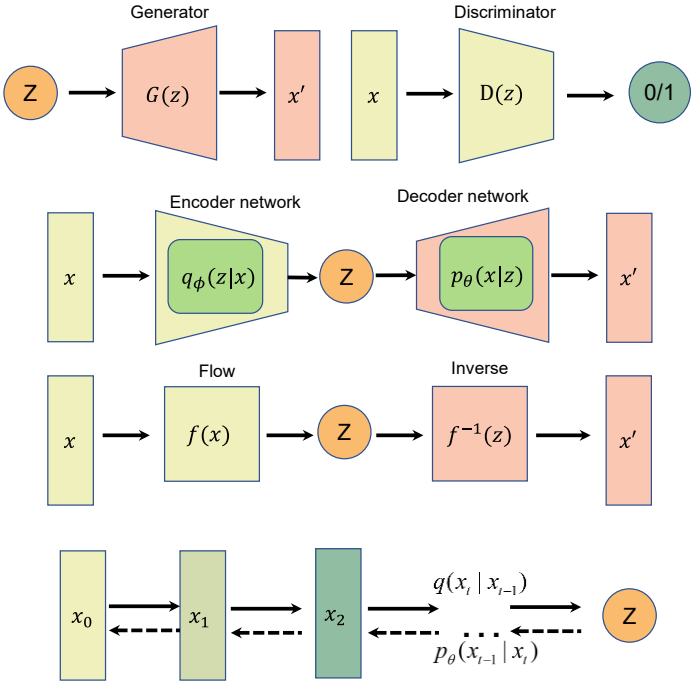


Figure 1. Comparison between the schematic diagrams of four generative models, from top to bottom: generative adversarial network (GAN), variational autoencoder (VAE), normalizing flow (NF), and diffusion model.

The diffusion model has made impressive progress in the field of image generation, surpassing the performance of GANs and emerging as a new type of generative model. In addition, the diffusion model has achieved state-of-the-art results in fields such as speech synthesis tasks [34] and image translation [41]. The diffusion model obtains results from posterior probability sampling instead of using traditional end-to-end inference methods, making it able to handle various distribution changes. The trained model can be generalized to out-of-distribution (OOD) test data and has achieved impressive results, especially in solving one-to-many problems such as image super-resolution. In this study, we first used simulated noisy signals for diffusion to generate high-quality images. As shown in Figure 2, the process of using the diffusion model for image super-resolution typically includes two processes: a forward diffusion process and a reverse diffusion process. The diffusion process gradually adds Gaussian noise to an image, and the reverse diffusion process is implemented through a parameterized Markov chain. Each Markov step is modeled with a deep neural network, which can learn how to invert the forward diffusion process to approximate the true data distribution to the greatest extent possible through the variational inference optimization of the network parameters.

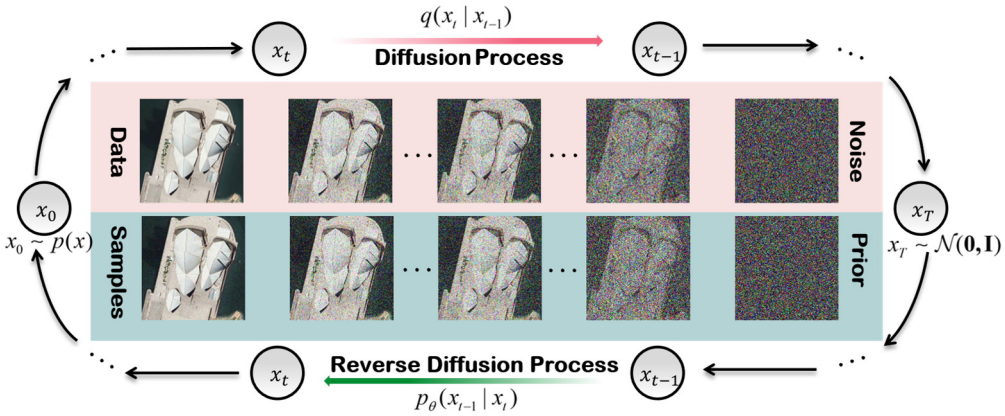


Figure 2. The diffusion process and reverse diffusion process of the diffusion model used for image super-resolution.

3. Proposed Method

3.1. Principles of Super-Resolution Using Diffusion Model

3.1.1. Diffusion Model

The diffusion model is an important generative model in machine learning, consisting of two main processes: a forward diffusion process and a reverse diffusion process. During the diffusion stage, the image data gradually become corrupted by noise until they completely become random noise. Intuitively, the forward process continuously adds noise to the data x_0, while the generation process continually removes noise to obtain the original data x_0. First, the true data distribution $x_0 \sim q(x)$ is defined, and small Gaussian noise is gradually added during the diffusion process. Assuming that T steps are taken in total, a series of noisy samples x are generated, which are latent variables with the same dimensions as the original data $x_0 \sim q(x)$. The noise parameters during the diffusion process are determined by an increasing sequence of $\beta_{1:T} \in (0,1]^T$, and for convenience of calculation and formula representation, let $\alpha_t := 1 - \beta_t$, $\overline{\alpha}_t := \prod_{n=1}^{t} \alpha_n$, where $\beta_1 < \beta_2 < \cdots < \beta_T$. The forward process transforms the distribution of the original data $q(x_0)$ step by step into the distribution of the latent variables $q(x_T)$, which can be described using the following formula:

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{1}$$

where

$$\begin{aligned} q(x_t|x_{t-1}) &= \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}) \\ &= \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)\mathbf{I}) \end{aligned} \tag{2}$$

The data distribution at any given time can be calculated without the need for any iteration through the derivation of Formula (3):

$$\begin{aligned} q(x_t|x_0) &= \int q(x_{1:t}|x_0)dx_{1:(t-1)} \\ &= \sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\varepsilon \\ &= \mathcal{N}(x_t; \sqrt{\overline{\alpha}_t}x_0, (1-\overline{\alpha}_t)\mathbf{I}) \end{aligned} \tag{3}$$

where

$$\varepsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I}) \tag{4}$$

As t increases, the proportion of noise becomes larger, and the proportion of original data becomes smaller. Gaussian noise occupies a larger proportion, and the distribution of $q(x_t|x_0)$ tends to $\mathcal{N}(\mathbf{0},\mathbf{I})$. At this point, it can be considered that the diffusion process of the model has been completed.

The reverse diffusion process in the diffusion model uses a Markov chain to transform a simple Gaussian probability distribution into a complex distribution in the real data. This process transforms the distribution of the latent variables $p_\theta(x_T)$ into the data distribution $p_\theta(x_0)$. Since the noise added in the forward process is very small each time, we assume that $p_\theta(x_{t-1}|x_t)$ is also a Gaussian distribution. As $p_\theta(x_{t-1}|x_t)$ is an unknown probability distribution, it can be fitted using a neural network. Herein, θ represents the parameters of the neural network.

When β_T is set close enough to 1, $q(x_t|x_0)$ approaches the standard normal distribution for all x_0. Therefore, $p_\theta(x_T)$ can be set to the standard normal distribution, i.e., $p_\theta(x_T) := \mathcal{N}(\mathbf{0},\mathbf{I})$. The joint probability distribution of the reverse diffusion process can be expressed using the following formula:

$$p_\theta(x_{0:T}) := p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{5}$$

where

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t,t), \sigma_\theta(x_t,t)^2\mathbf{I}) \tag{6}$$

By decomposing μ_θ into x_t and noise, an approximate value for the mean can be obtained as

$$\mu_\theta(x_t,t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\varepsilon_\theta(x_t,t)\right) \tag{7}$$

By setting the variance $\sigma_\theta(x_t,t)^2$ as a constant $\widetilde{\beta}_t$ related to β_t, the trainable parameters only exist in the mean, and the generation process can be expressed as

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\varepsilon_\theta(x_t,t)\right) + \widetilde{\beta}_t\mathbf{I} \tag{8}$$

where ε_θ denotes a neural network with the same input and output, wherein the noise predicted by the neural network ε_θ at each step is used for the reverse diffusion process.

Our goal is to find the parameters θ that maximize the double target data distribution $p_\theta(x_0)$, as shown in Equation (9). This is achieved by adding a non-negative KL divergence term to the negative log-likelihood function $-\log p_\theta(x_0)$ of the target data distribution $p_\theta(x_0)$, which constitutes an upper bound on the negative log-likelihood.

$$\begin{aligned} -\log p_\theta(x_0) &\leq -\log p_\theta(x_0) + D_{KL}[q(x_{1:T} \mid x_0) \| p_\theta(x_{1:T} \mid x_0)] \\ &= -\log p_\theta(x_0) + \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[\log \tfrac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})/p_\theta(x_0)}\right] \\ &= -\log p_\theta(x_0) + \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[\log \tfrac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} + \log p_\theta(x_0)\right] \\ &= \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[\log \tfrac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right] \end{aligned} \tag{9}$$

Continuing to expand the result in the above equation yields the following:

$$\begin{aligned} L_{VLB} &= L_T + L_{t-1} + \ldots + L_0 \\ L_T &= D_{KL}(q(x_T \mid x_0)||p_\theta(x_T)) \\ L_{t-1} &= D_{KL}(q(x_t \mid x_{t-1}, x_0)||p_\theta(x_t \mid x_{t+1})); \ 1 \leq t \leq T-1 \\ L_0 &= -\log p_\theta(x_0 \mid x_1) \end{aligned} \tag{10}$$

$p_\theta(x_{t-1}|x_t)$ is expressed as $N(x_{t-1}; \mu_\theta(x_t,t), \widetilde{\beta}_t\mathbf{I})$, and its corresponding diffusion process posterior $q(x_{t-1}|x_t,x_0)$ is expressed as $N(x_{t-1}; \widetilde{\mu}_t(x_t,x_0), \widetilde{\beta}_t\mathbf{I})$, where

$$\widetilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\varepsilon) \tag{11}$$

$$\widetilde{\beta}_t = \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t \tag{12}$$

The final loss function can be written as the root-mean-squared error between the means of the two distributions:

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \| \widetilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t) \|^2 \right] + C \tag{13}$$

To simplify the expression, the following loss function is minimized during the training process:

$$L_{t-1} = \mathbb{E}_{x_0, \epsilon, t} \left[\| \epsilon - \epsilon_\theta \left(\sqrt{\overline{\alpha}_t} x_0 + \sqrt{1 - \overline{\alpha}_t} \epsilon, t \right) \|^2 \right] \tag{14}$$

During the inference process, the latent variable $x_T \sim \mathcal{N}(x_T; \mathbf{0}, \mathbf{I})$ is first sampled from the standard normal distribution, and then it is sampled from it again using the formula detailed above to obtain x_{t-1}.

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t, - \frac{\beta_t}{\sqrt{1 - \overline{\alpha}_t}} \epsilon_\theta(x_t, t) \right) \tag{15}$$

$$\sigma_\theta(x_t, t) = \left(\frac{1 - \overline{\alpha}_{t-1}}{1 - \overline{\alpha}_t} \beta_t \right)^{\frac{1}{2}} \tag{16}$$

where $t \in \{T, T-1, \ldots, 1\}$, and the iteration continues until $p_\theta(x_0)$ is computed.

3.1.2. Super-Resolution-Based Diffusion Model

In the previous section, we introduced the principle of the diffusion model. Our proposed super-resolution method for remote-sensing images is also based on the T-step diffusion model, as shown in Figure 2. It mainly includes the diffusion process from left to right and the inverse diffusion process from right to left. Assuming that the distribution of high-resolution images in the given training set is $x_0 \sim p(x_0)$, as shown in Equation (2), Gaussian noise is continuously added to a clean image during the diffusion process to produce a series of noisy images, $x_1, \ldots, x_{t-1}, x_T$. As the number of steps increases, the high-resolution image x_0 gradually loses its original characteristics, x_T equivalent to an isotropic Gaussian distribution. The inverse diffusion process is the opposite of the diffusion process, as shown in Equations (5)–(7). The latent variable $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is gradually denoised and transformed into a high-resolution image. We use a neural network ϵ_θ to simulate this denoising process and predict the noise added at each step in the diffusion process through the neural network, with the LR image encoding as the input condition. In practical operation, a high-resolution image is not directly used as x_0; rather, the residual between the high-resolution image and the image $up(x_{LR})$ obtained by upsampling the low-resolution image is used. In the following chapters, we will introduce in detail the hybrid conditional features for low-resolution image encoding, the conditional noise predictor, as well as the training and inference processes.

3.2. Overview of Neural Network Model

3.2.1. Hybrid Conditional Features

As illustrated in Figure 3, we present the overall flowchart of our proposed hybrid conditional diffusion model for remote-sensing image super-resolution. This algorithm utilizes the diffusion model to represent the data points' diffusion patterns in the latent space, thereby learning the underlying structure within the dataset. The neural network (U-Net) is employed to learn the reverse diffusion process, which can generate high-resolution remote-sensing images from random noise images through the reverse diffusion procedure. The three inputs to the U-Net neural network are the latent variables x_t at time t, the low-resolution image features x_c, and the time t, respectively. For detailed information regarding these three inputs, please refer to the structure of the conditional noise predictor in Figure 4. The previous diffusion models do not pay much attention to the importance of the conditional features of the low-resolution input in the diffusion model. However, these

features can better guide the generation of high-resolution images in practice. Therefore, as shown in Figure 3, we designed hybrid conditional features in this paper, which include global high-level features and local visual features. The global high-level features are captured through the transformer network, while the local visual features are captured with our proposed convolutional neural network. The following sections detail the specific implementation steps of these two feature extraction methods:

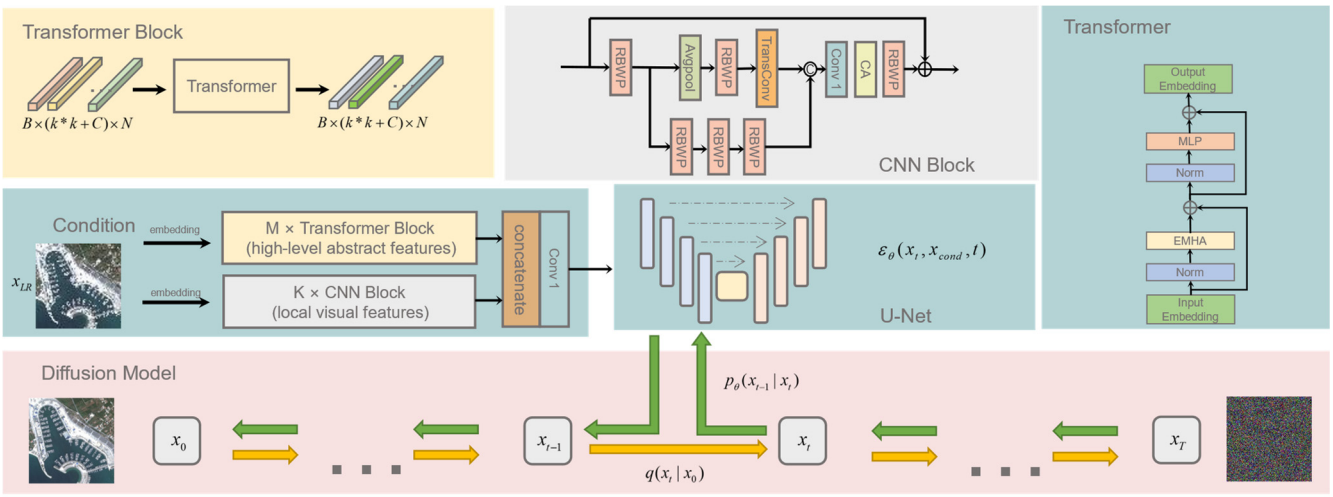


Figure 3. The primary components of the hybrid conditional diffusion model proposed for super-resolution of remote-sensing images.

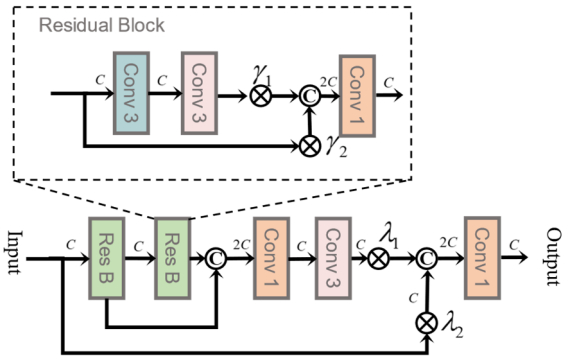


Figure 4. Framework of the proposed residual block with parameter (RBWP).

To obtain global high-level features from a low-resolution image, we selected a transformer structure similar to that in [42] as the backbone of the feature extraction network. The transformer captures long-distance dependencies between image blocks through self-attention, enabling it to acquire high-level global visual features. As shown in Figure 3, we first embed the input low-resolution image $I_{LR} \in \mathbb{R}^{H\times W\times 3}$ to obtain the feature $F \in \mathbb{R}^{H\times W\times C}$, where C is the number of feature channels. We then unfold the input feature into a sequence, which can be viewed as a series of flattened feature blocks $F_{p_i} \in \mathbb{R}^{k^2\times C}, i = \{1,\ldots,N\}$ obtained by dividing the feature into small blocks. The sequence contains $N = HW/k^2$ feature blocks, each with a dimension of $k^2 \times C$, where k^2 represents the size of the feature block, C is the number of channels, and N is considered the length of the sequence. The serialized features are then inputted to the transformer architecture. Assuming that the input sequence to the transformer block is E_i and the output sequence is E_o, we then obtain

$$E_{inter} = EMHA(Norm(E_i)) + E_i \tag{17}$$

$$E_0 = MLP(Norm(E_{inter})) + E_{inter} \tag{18}$$

where E_{inter} represents the intermediate representation of features, $Norm$ denotes layer normalization, MLP represents the multi-layer perceptron, and $EMHA$ represents efficient multi-head self-attention [42].

The overall structure of the CNN we used is shown in Figure 3, which mainly consists of the residual block with parameter (RBWP) illustrated in Figure 4. The learnable parameters in RBWP can be regarded as reallocating available resources to the part with the maximum amount of information, thereby encouraging the feature extraction network to focus on useful information.

Assuming that the input of RBWP is $x_i \in \mathbb{R}^{H\times W\times C}$, and $\mathcal{F}(\cdot)$ represents a nonlinear mapping, RBWP can be represented with the following formula:

$$x_{i+1} = \mathcal{C}_{1\times 1}([\lambda_1 \otimes x_i, \lambda_2 \otimes \mathcal{F}(x_i)]) \tag{19}$$

where $\otimes$ denotes element-wise multiplication, and the nonlinear mapping $\mathcal{F}(\cdot)$ consists of two residual blocks (Res Bs), a 1×1 convolutional layer for channel reduction, and a 3×3 convolutional layer for information fusion. Inside a Res B, there are two 3×3 convolutional kernels and learnable parameters λ_1 and λ_2. Assuming the input of the Res B is $y_i \in \mathbb{R}^{H\times W\times C}$, it can be represented with the following formula:

$$y_{i+1} = \mathcal{C}_{1\times 1}([\gamma_1 \otimes \mathcal{C}_{3\times 3}(\mathcal{C}_{3\times 3}(y_i)), \gamma_2 \otimes y_i]) \tag{20}$$

where y_{i+1} represents the output of the Res B, $\mathcal{C}_{1\times 1}$ and $\mathcal{C}_{3\times 3}$ denote the convolutional layers with 1×1 and 3×3 kernels, respectively, γ_1 and γ_2 represent the learnable parameters, $\otimes$ denotes multiplication, and $[\cdot, \cdot]$ denotes the aggregation of two feature maps.

Subsequently, we concatenate the high-level global visual features and local visual features obtained from the transformer network and the CNN. After concatenation, we employ a 1×1 convolution operation to fuse these two sets of features. Ultimately, this serves as one of the inputs, denoted as x_{cond}, for the U-Net architecture.

3.2.2. Conditional Noise Predictor (U-Net)

The network architecture of our conditional noise predictor $\epsilon_\theta(x_t, x_c, t)$ is shown in Figure 5. The network adopts the encoder–decoder structure of U-Net, which can effectively capture the details and global information in an image, is easy to train, and has a stable training process. The skip connections can help the network better learn the spatial information of the image and alleviate the problems of gradient vanishing and overfitting. The inputs of the network are the latent variable x_t at time t, the low-resolution image feature x_c, and the time t. According to Equations (15) and (16), the noise at time t in the reverse diffusion process can be predicted via the well-trained conditional noise predictor ϵ_θ, and then $\mu_\theta(x_t, t)$ and $\sigma_\theta(x_t, t)$ can be obtained, and the next latent variable x_{t-1} can be sampled. By repeatedly iterating, the super-resolution remote-sensing image can be obtained. The U-Net network serves as the main network of the conditional noise predictor. Firstly, the network transforms the input into feature maps using two-dimensional convolution and a Mish activation function. Then, the feature map of the LR image is fused with the feature map of x_t and input into the U-Net main network. According to the design by Ho et al., time t is encoded into t_e using transformer sinusoidal positional encoding and embedded into the Res block through a multi-layer perceptron (MLP). The main structure of the U-Net network consists of the encoder step, middle step, and decoder step. The detailed structures of each part will be introduced below.

As shown in Figure 5, the input of the Res block is $x_i \in \mathbb{R}^{H\times W\times C}$, and $\mathcal{F}(\cdot)$ represents the nonlinear mapping branch that includes a 3×3 convolution and Mish activation function. The Res block can be expressed with the following equation:

$$x_{i+1} = \mathcal{F}(\mathcal{F}(x_i) \oplus x_e) + x_i \tag{21}$$

where x_{i+1} represents the output of the Res block.

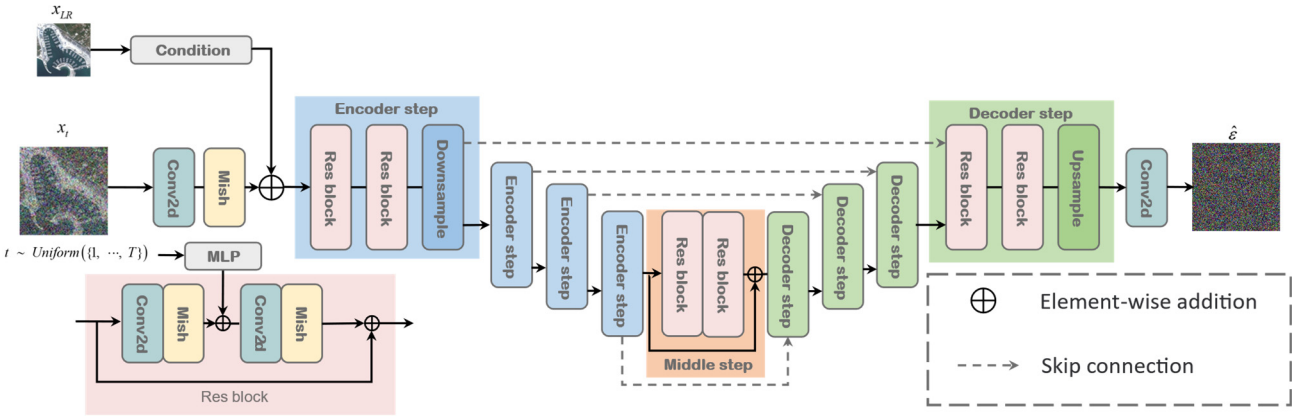


Figure 5. The architecture of conditional noise predictor (U-Net).

Each encoder step contains two Res blocks and one downsampling block, where the downsampling block uses a 2D convolution with a stride of 2 to reduce the feature map size by a stride of 2. Let E_n be the output of the n-th layer of the encoder, which can be expressed with the following equation:

$$E_n = f_n(E_{n-1}) \tag{22}$$

$$f_n(\cdot) = conv(\mathrm{Res}(\mathrm{Res}(\cdot), s = 2) \tag{23}$$

The middle step consists of two Res blocks and residual structures, which can be formulated as

$$M_o = k_n(M_i) \tag{24}$$

$$k_n(\cdot) = \mathrm{Res}(\mathrm{Res}(M_i)) \oplus M_i \tag{25}$$

where M_o and M_i are the input and output of the middle step, respectively.

Each decoder step contains two Res blocks and one upsampling block, where the upsampling block uses transpose convolution to double the feature map size. Let x be the output of the n-th layer of the encoder, which can be expressed with the following equation:

$$D_n = g_n(D_{n+1}, E_n) \tag{26}$$

$$g_n(\cdot) = transpose(\mathrm{Res}(\mathrm{Res}(\cdot), s = 2) \tag{27}$$

where transpose denotes transpose convolution with a stride of 2 to achieve upsampling. D_{n+1} represents the output of the $(n + 1)$-th layer of the decoder, and E_n represents the output of the n-th layer of the encoder. Finally, we reconstruct the predicted noise value by applying a 2D convolution to the output D_0 of the decoder. This predicted value $\hat{\varepsilon}$ is then used to recover the latent variable x_{t-1} at the next time step.

3.3. Fourier High-Frequency Spatial Constraints

The purpose of remote-sensing image super-resolution is to increase the high-frequency information in low-resolution images. How to obtain the lost high-frequency information has become the key to solving the super-resolution problem. For super-resolution methods based on diffusion models, it has been proven that adding pixel-level constraints in the reverse diffusion process of the model can guide the diffusion process [2], leading to more precise remote-sensing image super-resolution reconstruction. In order to further improve the efficiency of the model to reconstruct more detailed information and narrow the gap

with high-resolution images, we propose a Fourier high-frequency spatial loss function in this paper to better enhance the lost high-frequency information restoration capability in LR images. By directly emphasizing the high-frequency content through the frequency components calculated with the fast Fourier transform (FFT) [43], the proposed loss function can generate remote-sensing super-resolution images with more detailed information and fine objects. Moreover, it provides global constraints during training rather than local pixel loss in the spatial domain. This high-frequency information greatly contributes to small target recognition and the clarity of remote-sensing images.

The Fourier transform is widely used to analyze the frequency components of signals, and it can also be applied in the field of image processing, such as for image enhancement, image compression, and image analysis [44]. The Fourier transform represents the changes in pixel brightness in an image as a series of frequencies, including their amplitudes and phases. This representation can help us better understand the content and features of the image, such as edges, textures, and shapes. As shown in Figure 6, the 2D discrete Fourier transform (DFT) is a special form of the continuous Fourier transform (CFT) that can transform digital images $x \in \mathbb{R}^{H\times W\times C}$ from the spatial domain into the frequency domain. The Fourier space consists of standard orthogonal basis functions, where complex frequency components $X \in \mathbb{C}^{U\times V\times C}$ describe the characteristics of the spectrum. It should be noted that for multi-channel images, the Fourier transform can be applied to each channel separately and then combined. This process can be represented with the following formula:

$$F(u,v) = \sum_{x=0}^{H-1}\sum_{y=0}^{W-1} f(x,y)\cdot e^{-i2\pi(\frac{ux}{H}+\frac{vy}{W})} \tag{28}$$

where $H \times W$ represents the size of the image, (x, y) denotes the pixel coordinates in the spatial domain, $f(x, y)$ represents the pixel value, (u, v) represents the coordinates of the spatial frequency in the spectrum, $F(u, v)$ represents the complex frequency value, and c and i represent the Euler's number and imaginary unit, respectively. Using Euler's formula, we can obtain

$$e^{-i2\pi(\frac{ux}{M}+\frac{vy}{N})} = \cos 2\pi\Big(\frac{ux}{M}+\frac{vy}{N}\Big) - i\sin 2\pi\Big(\frac{ux}{M}+\frac{vy}{N}\Big) \tag{29}$$

The amplitude spectrum and phase spectrum of the Fourier transform are obtained via

$$|F(u,v)| = \sqrt{R^2(u,v)+I^2(u,v)} \tag{30}$$

$$\varphi(u,v) = \arctan(I(u,v)/R(u,v)) \tag{31}$$

where $I(u, v)$ and $R(u, v)$ are the real and imaginary parts of the Fourier transform, respectively.

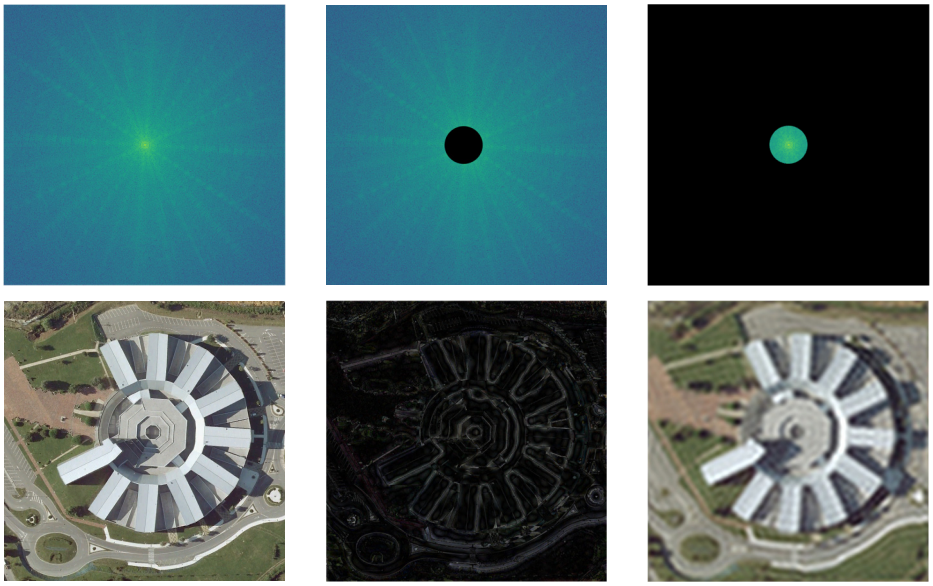
$$f(x,y) = \sum_{x=0}^{H-1}\sum_{y=0}^{W-1} f(u,v)\cdot e^{i2\pi(\frac{ux}{H}+\frac{vy}{W})} \tag{32}$$

Then, we can obtain the high-frequency and low-frequency features of the corresponding image.

By using the FFT to transform these two images into the frequency domain, we can obtain the high-frequency feature region in the Fourier space by applying a mask. Our idea is to calculate the loss in the high-frequency region of the Fourier space. The difference between the two vectors can be expressed as follows:

$$d\Big(\vec{r_r},\vec{r_f}\Big) = \|\vec{r_r}-\vec{r_f}\|_2^2 = \Big|F_r(u,v)-F_f(u,v)\Big|^2 \tag{33}$$

In order to more accurately represent the error, the loss function includes two parts: the amplitude loss $\left||\overrightarrow{r_{HR}}| - |\overrightarrow{r_{SR}}|\right|$ and phase loss $\theta_{HR} - \theta_{SR}$ at the location u, v [45], as shown in the Figure 7.



(**a**) (**b**) (**c**)

Figure 6. (**a**) The original image and its corresponding frequency spectrum, (**b**) the effect after applying a high-pass filter to the frequency spectrum, (**c**) the effect after applying a low-pass filter to the frequency spectrum.

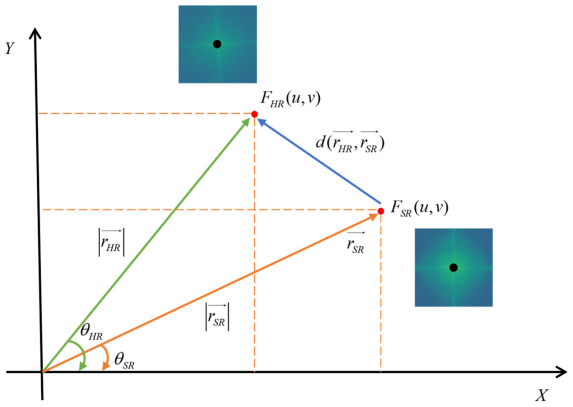


Figure 7. Schematic diagram of the high-frequency feature loss function.

The transformation into the entire high-frequency spectrum can be represented with the following formula:

$$\mathcal{L}_{\mathcal{F},|\cdot|} = d(F_{HR}, F_{SR}) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}||F_{HR}(u,v)| - |F_{SR}(u,v)|| \tag{34}$$

$$\mathcal{L}_{\mathcal{F},\angle} = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}||\varphi_{HR}(u,v)| - |\varphi_{SR}(u,v)|| \tag{35}$$

Finally, the total Fourier high-frequency spatial loss is obtained as follows:

$$\mathcal{L}_{\mathcal{F}} = \frac{1}{2}\mathcal{L}_{\mathcal{F},|\cdot|} + \frac{1}{2}\mathcal{L}_{\mathcal{F},\angle} \tag{36}$$

This loss function consists of two parts: amplitude loss $\mathcal{L}_{\mathcal{F},|\cdot|}$ and the corresponding phase loss $\mathcal{L}_{\mathcal{F},\angle}$. Then, the pixel loss function is added as a constraint term to the diffusion model to generate higher-quality images. Finally, the total loss function is obtained as follows:

$$\mathcal{L}_{pixel} = \frac{1}{HW}\sum_{h=0}^{H-1}\sum_{w=0}^{W-1}\left|y_{HRh,w} - y_{SRh,w}\right| \tag{37}$$

$$\mathcal{L}_{sum} = \alpha\mathcal{L}_{pixel} + \beta\mathcal{L}_{\mathcal{F}} \tag{38}$$

where α and β are hyperparameters used to control the weighting of the two loss functions.

3.4. Training and Inference Process

As shown in Algorithm 1, during the training phase, we first prepare the model $\epsilon_\theta(x_t, x_c, t)$ to be trained and randomly initialize its parameters (line 1). The LR-HR image pairs $\mathcal{D} = (x^i_{LR}, y^i_{HR})^N_{i=1}$ are used as the training dataset (line 2), and the input low-resolution images x_{LR} are passed through the pre-trained hybrid feature network to obtain the low-resolution image features (line 4). Then, during training, we randomly sample an image pair (x_{LR}, y_{HR}) from the dataset (line 6), randomly sample a time t from a uniform distribution $\{1, \ldots, T\}$ (line 8), and calculate the latent variable at time t according to the formula (line 3). Next, we feed x_t, x_c, t into the noise predictor $\epsilon_\theta(x_t, x_c, t)$ and optimize it through gradient descent (line 10).

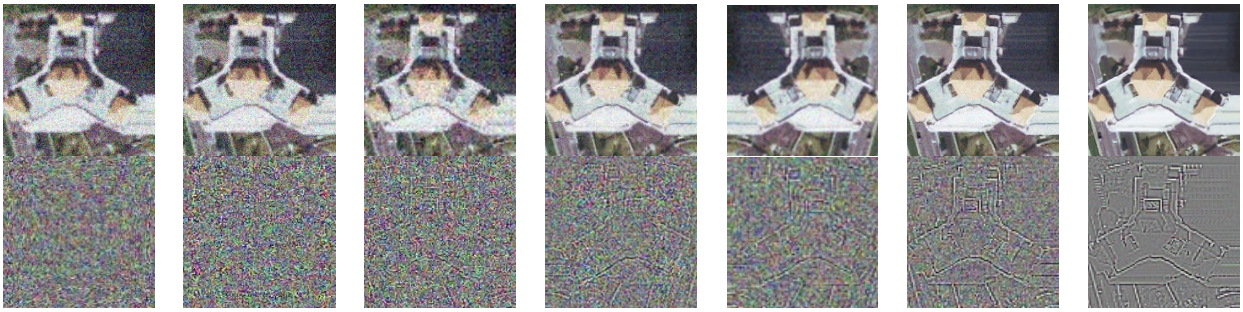
Algorithm 1 Training process.

1: **The model to be trained:** $\epsilon_\theta(x_t, x_c, t)$
2: **Dataset:** $\mathcal{D} = (x^i_{LR}, y^i_{HR})^N_{i=1}$
3: **The latent variable at time t:** $x_t = \sqrt{\overline{\alpha}_t}(x_{HR} - up(x_{LR})) + \sqrt{1-\overline{\alpha}_t}\epsilon$
4: **Input low-resolution image features:** $x_c = \mathcal{C}_{local}(x_{LR}) + \mathcal{C}_{global}(x_{LR})$
5: **Loss function:** $\mathcal{L}_\theta = \|\epsilon - \epsilon_\theta(x_t, x_c, t)\|^2_2 + \mathcal{L}_{sum}$
5: **While not converged do**
6: $(x_{LR}, y_{HR}) \sim \mathcal{D}$ ▷ Sample data
7: $\epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ ▷ Sample noise
8: $t \sim U(\{1, \ldots, T\})$ ▷ Sample time
9: Take gradient step on Loss $\mathcal{L}_\theta$
10: $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_\theta$ ▷ Optimization
11: **End while**

As shown in Algorithm 2, the inference process requires T steps, starting with $t = T$ (line 5). At this point, x_T is sampled from a standard normal distribution $\mathcal{N}(\mathbf{0},\mathbf{I})$ (line 4), and a residual image x_{t-1} with different levels of noise is output at each iteration (line 7). When $t > 1$, we sample z from a standard normal distribution $\mathcal{N}(\mathbf{0},\mathbf{I})$, and when $t = 1$, it is set to 0 (line 6). Then, using the noise predictor $\epsilon_\theta(x_t, x_c, t)$ with x_t, x_c, t as input, we calculate x_{t-1} (line 7), and x_0 serves as the final output. The super-resolution image is obtained by adding the residual image x_0 to the up-sampled low-resolution image $up(x_{LR})$. The intermediate images generated at each stage of the inference process in the diffusion model are presented as shown in Figure 8.

Algorithm 2 Inference process.

1: **The trained model:** $\epsilon_\eta(x_t, x_c, t), \mathcal{C}_{local}(x_{LR}), \mathcal{C}_{global}(x_{LR})$
2: **The low-resolution image to be SR:** x_{LR}
3: **Features of the low-resolution image:** $x_c = \mathcal{C}_{local}(x_{LR}) + \mathcal{C}_{global}(x_{LR})$
4: $x_T \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ ▷ Sample x_T
5: **for** $t = T, \ldots, 1$ **do**
6: $z \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ **if** $t > 1$, **else** $z = 0$ ▷ Sample noise
7: $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(x_t, x_c, t)\right) + \widetilde{\beta}_t^{\frac{1}{2}}z$ ▷ Sample x_t
8: **end for**
9: **Return** $x_0 + up(x_{LR})$



(**a**) Step 99 (**b**) Step 89 (**c**) Step 69 (**d**) Step 49 (**e**) Step 29 (**f**) Step 19 (**g**) Output

Figure 8. (**a**–**f**) depict the process of image reconstruction using the diffusion model, where the image on top represents $x_{\{1,\ldots,T\}}+up(x_{LR})$, and the image at the bottom represents $x_{\{1,\ldots,T\}}$. (**g**) represents the result of the image reconstruction, where the image on top represents $x_0 + up(x_{LR})$, and the image at the bottom represents x_0.

3.5. Inference Acceleration of Diffusion Model

Generating a high-resolution remote-sensing image x_0 from random noise x_T involves a reverse diffusion process that includes nearly a hundred steps. Therefore, a key challenge in using diffusion models for remote-sensing image super resolution is how to address the time cost resulting from multiple iterations. One effective method to address this issue is to use a smaller noise prediction model such as U-Net. Currently, there are many model compression methods available, including manually designing lightweight networks, pruning, quantization, neural architecture search (NAS), and knowledge distillation. Among these, knowledge distillation is a widely used and high-performing model compression method. It can transfer knowledge learned from a large teacher network to a smaller student network with minimal performance loss. The teacher network is typically a single complex network or a collection of networks with good performance and generalization ability. During the training process, the teacher network can learn mapping relationships, and the student network can improve its performance by learning the target task knowledge from the teacher network. To avoid the significant impact of distillation on the super resolution results, this paper introduced a feature distillation method [46] into the diffusion model super resolution to reduce the time cost in the reverse diffusion process.

First, we replaced the original U-Net network with a smaller U-Net network. The input and output channel numbers of each convolutional layer in the smaller U-Net network were reduced by half, while the input channel number of the input layer was kept unchanged. To address the issue of mismatched feature sizes between the smaller U-Net network and the original U-Net network, a 1×1 convolutional layer was applied between them. As shown in Figure 9, we defined the loss $L_{feature}$ of the student model learning the intermediate hidden layer features of the teacher model as follows:

$$L_{feature}(W_\eta, W_r) = \frac{1}{2}\sum_i \|u_i(x; W_\theta) - G(u_i(x; W_\eta); W_r)\|^2 \tag{39}$$

where W_θ represents the weights of the teacher model, W_η represents the weights of the student model, u_i represents the i-th output feature map that needs to be matched between the teacher and student networks, and G is a convolutional layer function designed to address inconsistencies between the hidden layers of the teacher and student models. After passing through this convolutional layer, the output features of the student network can match the feature dimension of the teacher's features.

L_{soft} represents the difference between the output results of the student and teacher models, while L_{hard} represents the difference between the output and the high-resolution image target. From this, the joint loss function L_{total} can be obtained.

$$L_{total} = \alpha L_{feature} + \beta L_{soft} + \gamma L_{hard} \tag{40}$$

The variables α, β, and γ represent the weight hyperparameters of the respective loss functions. These three parameters are empirically set to $\alpha = \frac{1}{10}$, $\beta = \frac{2}{5}$, and $\gamma = \frac{1}{2}$. The process of training the student model mirrors the steps involved in training the teacher model.

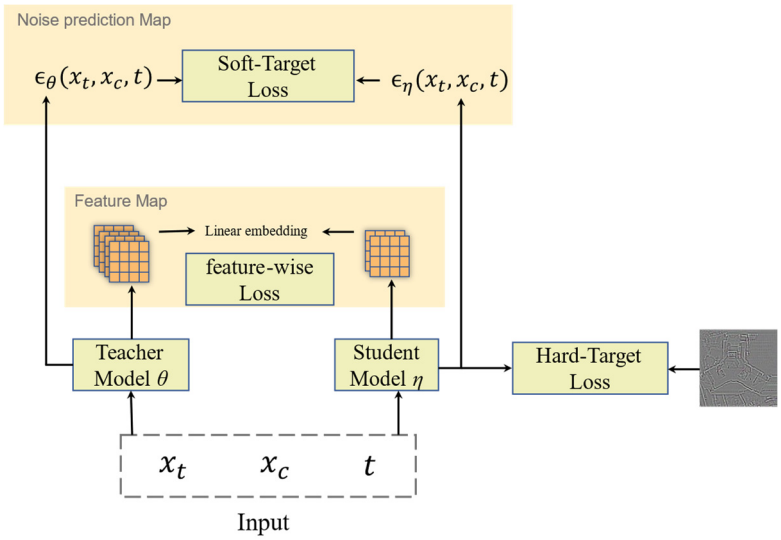


Figure 9. The feature distillation method applied to diffusion model super resolution.

4. Experiment

This section is divided with subheadings. It provides a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

4.1. Settings

4.1.1. Dataset

We used two publicly available datasets, AID [47] and RSSCN7 [48], for our training data. The RSSCN7 dataset contains 2800 remote-sensing images from seven typical scenes: grassland, forests, farmland, parking lots, residential areas, industrial areas, and rivers/lakes. Each category includes 400 images, which are sampled at four different scales. The AID dataset is a large-scale aerial image dataset that collects sample images from Google Earth. Although Google Earth images are post-processed from the raw optical aerial images to render them in RGB, there is no significant difference between them and actual optical aerial images. Therefore, the AID dataset can also be used as a training dataset for remote-sensing image super-resolution algorithms.

As shown in Figures 10–12, to demonstrate the generalization capability of our proposed algorithm, we conducted experiments on two datasets, RSOD [19] and UC Merced Land Use [20], and validated our results with the real-world Gaofen-2 dataset [21]. The UC Merced Land Use dataset is a scene recognition dataset released by the Computer Vision Lab at the University of California, Merced. The images in the dataset are sourced from the United States Geological Survey (USGS) National Map Urban Area Imagery collection and include 21 categories, such as agricultural areas, airplanes, and baseball fields. The RSOD dataset is a dataset for object detection in remote-sensing images. It contains four categories of objects, including airplanes, playgrounds, overpasses, and oil drums. The dataset was released by Wuhan University in 2015 and contains a total of 976 images. The Gaofen-2 dataset [21] comes from the Gaofen-2 (GF-2) satellite, which is the first civil optical remote-sensing satellite in China with a spatial resolution of less than 1 m, carrying two cameras with a spatial resolution of 1 m (panchromatic) and 4 m (multispectral). The dataset was acquired from the satellite and has a spatial resolution of up to 0.8 m at the ground level.

Figure 10. Display of images in different scene categories in the UC Merced Land Use test set.

Figure 11. Display of images in different scene categories in the RSOD test set.

Figure 12. Display of images in different scene categories in the Gaofen-2 test set.

4.1.2. Implementation Details

We propose a model consisting of a conditional noise predictor U-Net, with U-Net channels set to $C = 64$, as well as a transformer network and a CNN for extracting low-

resolution image features, with $N = 4$, $K = 6$, and $C = 64$ channels. The conditional noise predictor uses the Adam [49] optimization method to update model parameters, with β_1 and β_2 set to 0.9 and 0.999, respectively, and a batch size of 8. To improve the model's stability, a series of data augmentation operations, such as rotation and flipping, were applied to the training dataset. The number of steps for the forward and reverse diffusion processes in the diffusion model were set empirically to 100, while the noise schedule $\beta_1, \ldots, \beta_T$ and $\alpha_1, \ldots, \alpha_T$ were set according to [50]. The learning rate was initially set to 1×10^{-4} and decreased by a factor of 10 every 20 epochs. We performed 5 identical training and validation runs for each experiment to obtain an average result and increase the persuasiveness of the experiments. All experiments were conducted using PyTorch 1.12.1 [51] and Python 3.9, with CUDA 11.7 and CuDNN 8.2.1, on a server with an Intel Core i9-12900K CPU, 64 GB RAM, and an NVIDIA GeForce RTX 3090 GPU.

4.1.3. Evaluation Metrics

To effectively evaluate the effectiveness of the algorithm proposed in this paper, we employed several widely used objective image quality assessment methods in the super-resolution field. The details of these image quality assessment methods are provided below.

The mean square error (MSE) is used to represent the intensity of image distortion by calculating the average difference between the pixel values of the reference image and the distorted image. The MSE can be calculated using the following formula:

$$MSE = \frac{1}{WH} \sum_{j=1}^{H} \sum_{i=1}^{W} \left(\mathbf{I}_{ref}(i,j) - \mathbf{I}_{dist}(i,j) \right)^2 \tag{41}$$

where $\mathbf{I}$ represents the input image, and H and W denote its height and width, respectively. $\mathbf{I}(i,j)$ represents the pixel value of the image at location (i,j).

The peak signal-to-noise ratio (PSNR) of an image is a physical measure that represents the ratio of the maximum value of a signal to the maximum value of the distorted signal. PSNR is often used as a quantitative indicator for image quality enhancement. When evaluating distorted images, the PSNR can be calculated using the maximum grayscale value and the mean square error (MSE) between the distorted and reference images.

$$PSNR = 10 \log_{10} \left(\frac{D^2}{MSE} \right) \tag{42}$$

where D represents the dynamic range of the pixel values, which is typically 256 for 8-bit images.

Natural images have strong inter-pixel dependencies, which form the structural information of the images. Compared with PSNR and MSE, which evaluate image quality based on pixel-level differences, SSIM can effectively measure changes in the structural information of the image. Therefore, SSIM is better suited to the human visual system (HVS). The SSIM algorithm compares images from three perspectives—luminance, contrast, and structure—and combines the results to obtain the structural similarity index (SSIM). The calculation process is as follows:

$$SSIM(x,y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{43}$$

where μ_x and μ_y are the mean values of x and y, σ_x and σ_y are the variances of x and y, σ_{xy} is the covariance of x and y, and c_1, c_2 are two constants to avoid division by zero.

4.2. Comparisons with State-of-the-Art Algorithms

In this section, we compare the leading super-resolution algorithms for general images, including SRCNN [5], VDSR [9], SAN [12], DDBPN [23], and RDN [13], with those designed

specifically for remote-sensing images, such as MHAN [10] and EEGAN [28]. The source code for these benchmark methods can be downloaded from the authors' websites, and the relevant parameters were strictly configured according to the authors' recommendations in their publications. We trained and tested these methods under the same conditions on the RSOD and UCMerced_Land datasets, as shown in Tables 1 and 2. Unlike general images, remote-sensing images contain complex scenes and small targets, which may render models that perform well on general image datasets unsuitable for remote-sensing images. Our model achieved competitive results in both PSNR and SSIM metrics for different scale factors (×2, ×4, and ×8), outperforming the other methods by 1–3 points in PSNR and SSIM for ×4 and ×8 scale factors. These results suggest that our model is superior to other methods.

Table 1. Comparison between different remote-sensing image super-resolution methods on the UCMerced_Land test dataset, with evaluation metrics including PSNR and SSIM values, at scale factors of ×2, ×4, and ×8.

Method	×2 PSNR/SSIM	×4 PSNR/SSIM	×8 PSNR/SSIM
Bicubic	30.55/0.890	25.37/0.698	22.15/0.481
SRCNN [5]	32.20/0.917	26.35/0.730	22.52/0.515
VDSR [9]	33.22/0.925	27.02/0.764	23.01/0.534
SAN [12]	33.61/0.934	27.42/0.775	23.21/0.540
DDBPN [23]	33.67/0.931	27.49/0.771	23.54/0.571
RDN [13]	33.69/0.933	27.54/0.781	23.52/0.567
MHAN [10]	33.61/0.927	27.40/0.764	23.56/0.559
EEGAN [28]	33.54/0.926	27.30/0.770	23.44/0.553
Ours	33.76/0.930	27.60/0.788	23.68/0.581

Table 2. Comparison between different remote-sensing image super-resolution methods on the RSOD test dataset, with evaluation metrics including PSNR and SSIM values, at scale factors of ×2, ×4, and ×8.

Method	×2 PSNR/SSIM	×4 PSNR/SSIM	×8 PSNR/SSIM
Bicubic	29.91/0.942	26.71/0.807	24.21/0.638
SRCNN [5]	30.42/0.951	27.22/0.834	24.55/0.656
VDSR [9]	30.87/0.960	27.53/0.859	24.89/0.673
SAN [12]	31.08/0.961	27.74/0.865	25.08/0.694
DDBPN [23]	31.13/0.964	27.76/0.872	25.10/0.703
RDN [13]	31.16/0.963	27.80/0.871	25.13/0.704
MHAN [10]	31.18/0.967	27.71/0.862	25.19/0.696
EEGAN [28]	31.19/0.973	27.69/0.863	25.20/0.702
Ours	31.16/0.968	27.86/0.876	25.33/0.710

Due to significant differences in remote-sensing images across various scenes, we further tested our proposed method on remote-sensing images of different scenes to demonstrate its universality and robustness. Specifically, we conducted experiments on remote-sensing images of different types of scenes and the results, as shown in Tables 3–8, indicate that our proposed method achieves competitive results on remote-sensing images of various scenes. Notably, our method performs particularly well on images of complex scenes with rich textures, such as buildings or forests, as indicated by the higher SSIM values.

Table 3. Performance comparison between different remote-sensing image super-resolution methods on the UCMerced_Land test dataset for various scenes at scale factor of ×2, with evaluation metrics including PSNR and SSIM values.

Scene	SRCNN [5]	VDSR [9]	SAN [12]	DDBPN [23]	RDN [13]	MHAN [13]	EEGAN [28]	Ours
	PSNR/SSIM							
Agricultural	32.14/0.831	32.18/0.831	32.31/0.829	32.22/0.829	32.33/0.826	32.25/0.832	32.14/0.827	32.32/0.829
Airplane	32.96/0.924	34.46/0.939	34.89/0.943	35.01/0.944	35.10/0.944	34.10/0.933	34.86/0.943	35.20/0.949
Baseball diamond	35.33/0.892	35.84/0.899	36.19/0.902	36.22/0.902	36.25/0.903	36.28/0.901	36.14/0.902	36.24/0.901
Beach	38.58/0.958	39.08/0.963	39.33/0.965	39.36/0.965	39.38/0.965	39.34/0.963	39.33/0.965	39.38/0.963
Buildings	31.98/0.916	33.39/0.931	33.94/0.935	33.99/0.935	34.01/0.936	33.93/0.928	33.88/0.934	34.10/0.938
Chaparral	30.43/0.929	30.86/0.936	30.96/0.937	31.01/0.937	31.05/0.938	30.94/0.934	30.97/0.937	31.01/0.934
Dense residential	32.72/0.943	34.10/0.956	34.58/0.959	34.64/0.959	34.79/0.961	34.54/0.951	34.42/0.958	34.75/0.966
Forest	33.46/0.907	33.88/0.914	34.05/0.915	34.04/0.915	34.09/0.916	33.97/0.912	34.01/0.915	34.11/0.916
Freeway	33.68/0.942	35.82/0.959	36.34/0.961	36.45/0.962	36.51/0.963	36.47/0.965	36.17/0.961	36.50/0.963
Golf course	35.86/0.902	36.31/0.909	36.53/0.913	36.55/0.913	36.57/0.913	36.56/0.914	36.47/0.912	36.58/0.913
Harbor	29.58/0.955	31.42/0.97	32.24/0.974	32.36/0.974	32.54/0.975	32.48/0.976	32.21/0.973	32.50/0.971
Intersection	33.59/0.934	34.58/0.944	35.01/0.948	35.11/0.949	35.17/0.950	35.19/0.952	34.92/0.948	35.22/0.953
Medium residential	29.10/0.893	30.06/0.909	30.35/0.913	30.41/0.914	30.51/0.914	30.58/0.916	30.30/0.913	30.49/0.911
Mobile home park	28.82/0.911	30.05/0.928	30.45/0.932	30.53/0.933	30.59/0.934	30.59/0.936	30.39/0.932	30.55/0.933
Overpass	31.03/0.914	33.01/0.935	33.59/0.940	33.77/0.941	33.72/0.941	33.74/0.945	33.65/0.940	33.78/0.941
Parking lot	27.46/0.918	28.56/0.935	29.15/0.940	29.28/0.941	29.41/0.942	29.36/0.946	29.09/0.940	29.40/0.944
River	29.87/0.873	30.21/0.883	30.32/0.886	30.34/0.886	30.35/0.887	30.31/0.889	30.32/0.886	30.33/0.884
Runway	33.08/0.916	34.54/0.931	35.22/0.936	35.28/0.937	35.44/0.938	35.45/0.941	35.21/0.935	35.46/0.938
Sparse residential	31.12/0.881	31.6/0.889	31.75/0.892	31.77/0.892	31.81/0.893	31.85/0.894	31.73/0.892	31.83/0.891
Storage tanks	32.05/0.913	33.24/0.929	33.68/0.933	33.74/0.934	33.77/0.934	34.77/0.936	33.63/0.933	33.80/0.930
Tennis court	33.70/0.929	35.06/0.944	35.51/0.948	35.55/0.948	35.63/0.949	35.53/0.944	35.43/0.947	35.66/0.952

Table 4. Performance comparison between different remote-sensing image super-resolution methods on the RSOD test dataset for various scenes at scale factor of ×2, with evaluation metrics including PSNR and SSIM values.

Scenes	SRCNN [5]	VDSR [9]	SAN [12]	DDBPN [23]	RDN [13]	MHAN [13]	EEGAN [28]	Ours
	PSNR/SSIM							
Aircraft	34.67/0.963	35.23/0.968	35.34/0.969	35.41/0.971	35.40/0.970	35.45/0.972	35.48/0.970	35.52/0.973
Oil tank	29.69/0.974	30.05/0.977	30.22/0.977	30.27/0.979	30.27/0.979	30.30/0.979	30.34/0.980	30.38/0.982
Overpass	28.64/0.932	29.07/0.939	29.14/0.940	29.27/0.942	29.25/0.942	29.33/0.943	29.35/0.943	29.35/0.945
Playground	28.67/0.953	29.14/0.959	29.30/0.960	29.37/0.962	29.34/0.962	29.43/0.963	29.45/0.963	29.44/0.963

Table 5. Performance comparison between different remote-sensing image super-resolution methods on the UCMerced_Land test dataset for various scenes at scale factor of ×4, with evaluation metrics including PSNR and SSIM values.

Scene	SRCNN [5]	VDSR [9]	SAN [12]	DDBPN [23]	RDN [13]	MHAN [13]	EEGAN [28]	Ours
	PSNR/SSIM							
Agricultural	25.95/0.489	25.95/0.496	26.25/0.506	26.26/0.506	26.38/0.508	26.27/0.505	26.16/0.503	26.45/0.506
Airplane	26.76/0.778	27.98/0.808	28.52/0.818	28.68/0.821	28.69/0.822	27.96/0.805	28.45/0.816	28.72/0.818
Baseball diamond	30.71/0.758	31.17/0.770	31.47/0.777	31.51/0.778	31.55/0.779	31.17/0.770	31.34/0.775	31.57/0.777
Beach	32.64/0.850	33.05/0.863	33.20/0.867	33.21/0.867	33.23/0.868	33.02/0.863	33.17/0.866	33.20/0.867
Buildings	25.28/0.757	26.41/0.794	27.05/0.808	27.05/0.810	27.19/0.812	26.54/0.795	26.80/0.803	27.45/0.808
Chaparral	24.64/0.736	24.99/0.756	25.23/0.767	25.28/0.769	25.34/0.772	25.04/0.759	25.20/0.765	25.43/0.767
Dense residential	25.38/0.783	26.32/0.821	26.85/0.835	26.95/0.839	27.04/0.841	26.37/0.820	26.70/0.832	26.95/0.837
Forest	27.59/0.692	27.77/0.706	27.90/0.713	27.90/0.713	27.92/0.715	27.78/0.707	27.85/0.711	28.07/0.716
Freeway	27.40/0.802	28.63/0.837	29.38/0.851	29.53/0.855	29.58/0.856	28.66/0.836	29.22/0.85	29.58/0.851
Golf course	31.65/0.782	31.99/0.790	32.23/0.796	32.26/0.797	32.30/0.798	31.97/0.790	32.18/0.795	32.23/0.796
Harbor	21.52/0.784	22.35/0.821	22.92/0.837	22.89/0.839	23.01/0.842	22.51/0.821	22.67/0.829	23.12/0.847
Intersection	26.66/0.770	27.32/0.791	27.72/0.803	27.82/0.806	27.92/0.808	27.45/0.792	27.63/0.801	28.05/0.814
Medium residential	23.66/0.677	24.29/0.709	24.65/0.723	24.73/0.726	24.76/0.727	24.30/0.707	24.53/0.718	24.85/0.723
Mobile home park	23.07/0.725	23.73/0.759	24.11/0.773	24.20/0.777	24.22/0.778	23.77/0.759	24.02/0.769	24.33/0.780
Overpass	25.40/0.724	26.39/0.762	27.14/0.789	27.26/0.791	27.31/0.794	26.47/0.766	26.88/0.779	27.34/0.789
Parking lot	20.76/0.707	21.16/0.739	21.50/0.752	21.60/0.753	21.63/0.754	21.23/0.737	21.39/0.747	21.68/0.758
River	25.61/0.656	25.88/0.676	26.04/0.686	26.05/0.687	26.06/0.688	25.9/0.677	26.01/0.684	26.04/0.693

Table 5. *Cont.*

Scene	SRCNN [5]	VDSR [9]	SAN [12]	DDBPN [23]	RDN [13]	MHAN [13]	EEGAN [28]	Ours
	PSNR/SSIM							
Runway	27.53/0.777	29.41/0.819	30.19/0.830	30.45/0.833	30.38/0.834	29.54/0.816	30.04/0.828	30.39/0.841
Sparse residential	26.47/0.680	26.83/0.699	27.04/0.706	27.07/0.708	27.08/0.709	26.85/0.699	26.98/0.705	27.04/0.706
Storage tanks	26.43/0.741	27.14/0.773	27.64/0.790	27.72/0.793	27.78/0.795	27.23/0.775	27.52/0.785	27.74/0.790
Tennis court	27.83/0.766	28.49/0.790	29.01/0.807	29.14/0.811	29.18/0.813	28.55/0.791	28.85/0.802	29.21/0.807

Table 6. Performance comparison between different remote-sensing image super-resolution methods on the RSOD test dataset for various scenes at scale factor of ×4, with evaluation metrics including PSNR and SSIM values.

Scene	SRCNN [5]	VDSR [9]	SAN [12]	DDBPN [23]	RDN [13]	MHAN [13]	EEGAN [28]	Ours
	PSNR/SSIM							
Aircraft	30.23/0.869	30.84/0.884	30.92/0.887	31.16/0.892	31.06/0.890	31.20/0.893	31.25/0.894	31.30/0.896
Oil tank	27.52/0.905	27.65/0.914	27.74/0.918	27.82/0.923	27.77/0.920	27.82/0.922	27.86/0.924	27.89/0.928
Overpass	25.25/0.746	25.5/0.768	25.55/0.771	25.66/0.78	25.63/0.778	25.68/0.782	25.71/0.783	25.80/0.788
Playground	25.88/0.835	26.15/0.853	26.29/0.856	26.32/0.863	26.28/0.861	26.34/0.864	26.37/0.866	26.46/0.872

Table 7. Performance comparison between different remote-sensing image super-resolution methods on the UCMerced_Land test dataset for various scenes at scale factor of ×8, with evaluation metrics including PSNR and SSIM values.

Scene	SRCNN [5]	VDSR [9]	SAN [12]	DDBPN [23]	RDN [13]	MHAN [13]	EEGAN [28]	Ours
	PSNR/SSIM							
Agricultural	23.34/0.266	23.38/0.276	23.36/0.277	23.53/0.296	23.46/0.291	23.46/0.298	23.32/0.294	23.60/0.316
Airplane	22.22/0.594	23.13/0.637	23.44/0.638	24.09/0.669	24.02/0.664	24.01/0.668	23.95/0.663	24.22/0.672
Baseball diamond	27.26/0.619	27.81/0.636	28.00/0.641	28.30/0.651	28.28/0.653	28.22/0.641	28.14/0.639	28.39/0.662
Beach	29.29/0.725	29.72/0.737	29.84/0.740	29.98/0.746	29.97/0.745	29.96/0.742	29.88/0.740	30.06/0.763
Buildings	20.53/0.516	21.51/0.570	21.86/0.58	22.47/0.617	22.44/0.612	22.44/0.588	22.36/0.572	22.52/0.628
Chaparral	20.47/0.350	20.54/0.370	20.59/0.377	20.67/0.388	20.69/0.390	20.63/0.382	20.51/0.377	20.75/0.394
Dense residential	20.50/0.512	21.21/0.564	21.42/0.567	21.87/0.604	21.86/0.601	21.84/0.592	21.77/0.596	21.92/0.613
Forest	24.62/0.435	24.72/0.450	24.78/0.454	24.83/0.462	24.84/0.463	24.87/0.466	24.74/0.461	24.91/0.471
Freeway	23.07/0.527	23.57/0.555	24.02/0.601	24.57/0.641	24.48/0.641	24.46/0.636	24.38/0.631	24.64/0.653
Golf course	27.98/0.662	28.78/0.681	28.96/0.683	29.33/0.693	29.27/0.691	29.23/0.694	29.11/0.696	29.46/0.697
Harbor	17.07/0.527	17.37/0.563	17.61/0.575	17.85/0.607	17.89/0.606	17.84/0.594	17.70/0.595	17.96/0.617
Intersection	22.43/0.530	22.88/0.558	23.12/0.566	23.43/0.589	23.44/0.587	23.42/0.588	23.35/0.587	23.58/0.595
Medium residential	20.29/0.424	20.73/0.457	20.89/0.463	21.19/0.491	21.17/0.488	21.12/0.488	21.08/0.483	21.25/0.495
Mobile home park	18.91/0.457	19.31/0.491	19.56/0.501	19.89/0.534	19.88/0.529	19.82/0.529	19.78/0.522	19.94/0.538
Overpass	22.01/0.482	22.62/0.516	22.84/0.526	23.37/0.558	23.26/0.556	23.27/0.555	23.16/0.554	23.48/0.562
Parking lot	17.01/0.403	17.21/0.434	17.27/0.436	17.36/0.463	17.36/0.459	17.31/0.455	17.27/0.458	17.44/0.461
River	23.36/0.475	23.60/0.492	23.70/0.497	23.85/0.509	23.84/0.508	23.86/0.501	23.72/0.496	23.96/0.519
Runway	22.70/0.585	23.67/0.621	24.33/0.634	25.07/0.659	25.12/0.658	25.16/0.650	25.05/0.653	25.14/0.670
Sparse residential	23.15/0.456	23.51/0.477	23.67/0.482	23.80/0.495	23.82/0.494	23.88/0.491	23.77/0.495	23.94/0.496
Storage tanks	23.12/0.551	23.53/0.576	23.66/0.581	23.98/0.602	23.93/0.598	23.91/0.596	23.80/0.589	24.06/0.612
Tennis court	23.91/0.570	24.42/0.597	24.63/0.602	25.04/0.628	25.01/0.623	25.05/0.627	25.02/0.625	25.13/0.637

Table 8. Performance comparison between different remote-sensing image super-resolution methods on the RSOD test dataset for various scenes at scale factor of ×8, with evaluation metrics including PSNR and SSIM values.

Scenes	SRCNN [5]	VDSR [9]	SAN [12]	DDBPN [23]	RDN [13]	MHAN [13]	EEGAN [28]	Ours
	PSNR/SSIM							
Aircraft	26.67/0.738	27.25/0.756	27.51/0.764	27.46/0.761	27.61/0.768	27.71/0.772	27.70/0.772	27.81/0.779
Oil tank	25.18/0.739	25.43/0.753	25.73/0.770	25.69/0.768	25.77/0.771	25.80/0.776	25.76/0.777	25.89/0.786
Overpass	22.72/0.508	22.97/0.531	23.12/0.545	23.11/0.542	23.10/0.545	23.25/0.558	23.26/0.559	23.30/0.561
Playground	23.62/0.651	23.89/0.673	24.07/0.685	24.05/0.682	24.06/0.688	24.18/0.695	24.20/0.696	24.27/0.704

Since the visual differences between various SR algorithms for a ×2 scale factor are not significant, this paper compared the visual effects of various algorithms for ×4

and ×8 scale factors, as shown in Figures 13 and 14. The proposed model effectively distinguishes between roof textures and road signs, separates closely spaced individual targets, and accurately reconstructs the color information and texture details of high-resolution images, restoring most of the details (including roof details and dense trees). The generated images exhibit more intricate details and higher contrast, demonstrating that our algorithm can recover high-resolution images with rich semantic information from low-resolution images that contain minimal detailed information, without producing excessive additional information.

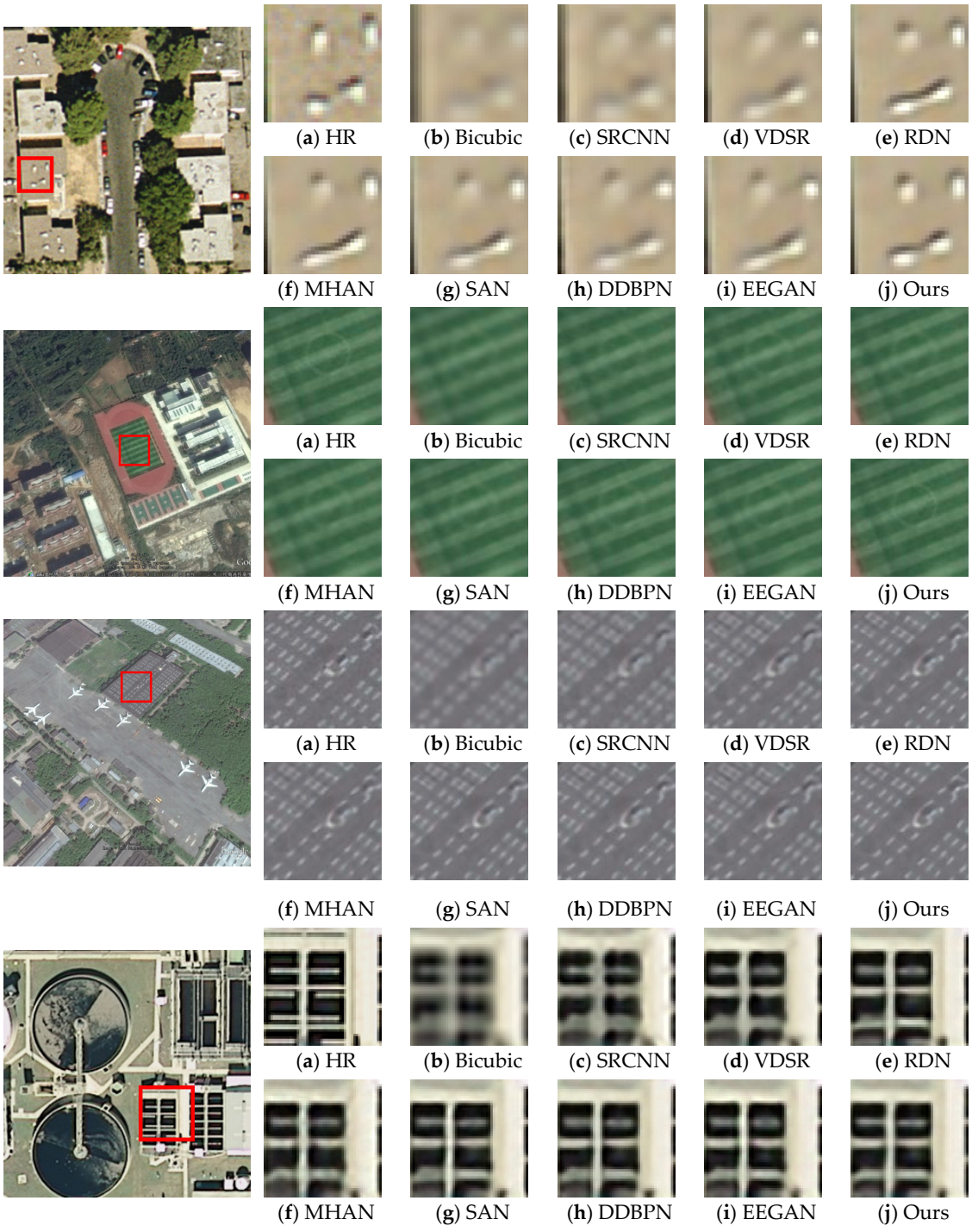


Figure 13. SR results at scale factor of ×4 on the test dataset [19,20] using different approaches (**b**–**j**), and (**a**) represents the original high-resolution image for each approach.

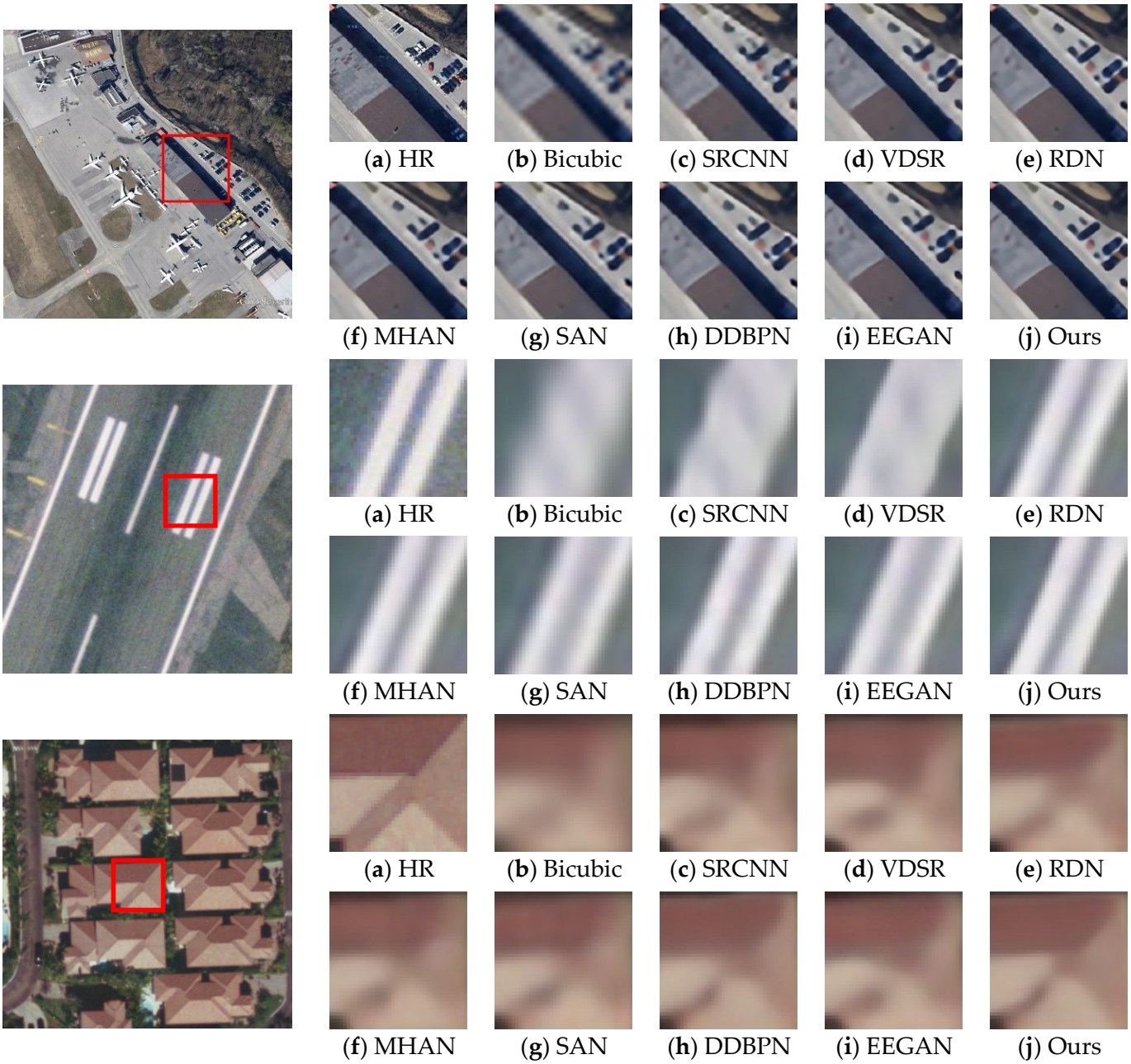


Figure 14. *Cont.*

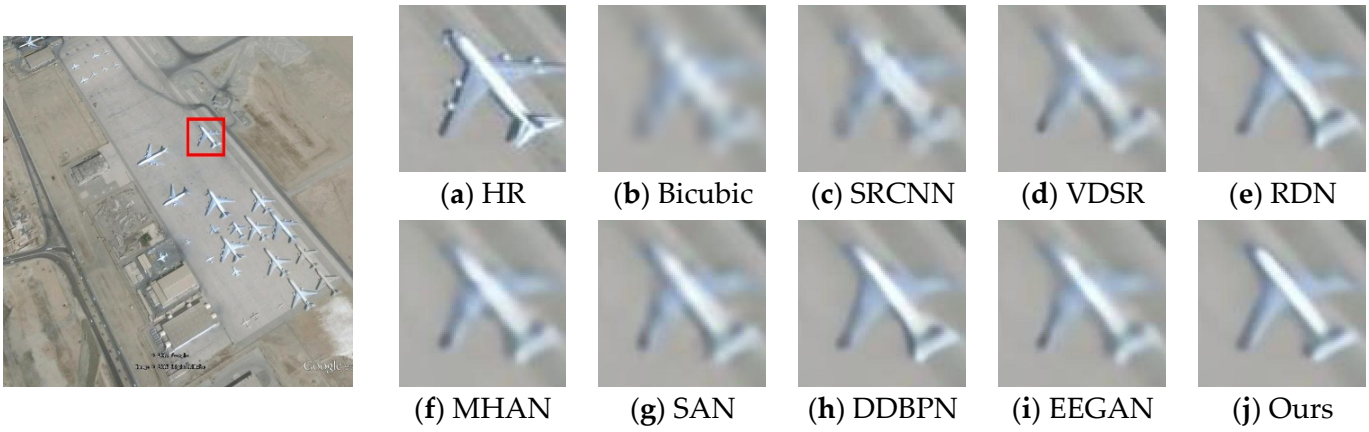


Figure 14. SR results at scale factor of ×8 on the test dataset [19,20] using different approaches (**b**–**j**), and (**a**) represents the original high-resolution image for each approach.

Furthermore, to verify the generalization performance of our proposed algorithm and its performance on real remote-sensing datasets, we validated our algorithm on the Gaofen-2 dataset [21]. Since there is no reference image available for real datasets, we compared the different methods from the perspective of human visual perception, as shown in Figures 15 and 16. Our proposed algorithm recovers images with higher contrast and sharper edges, while the results generated by other methods are blurry and lack detailed information.

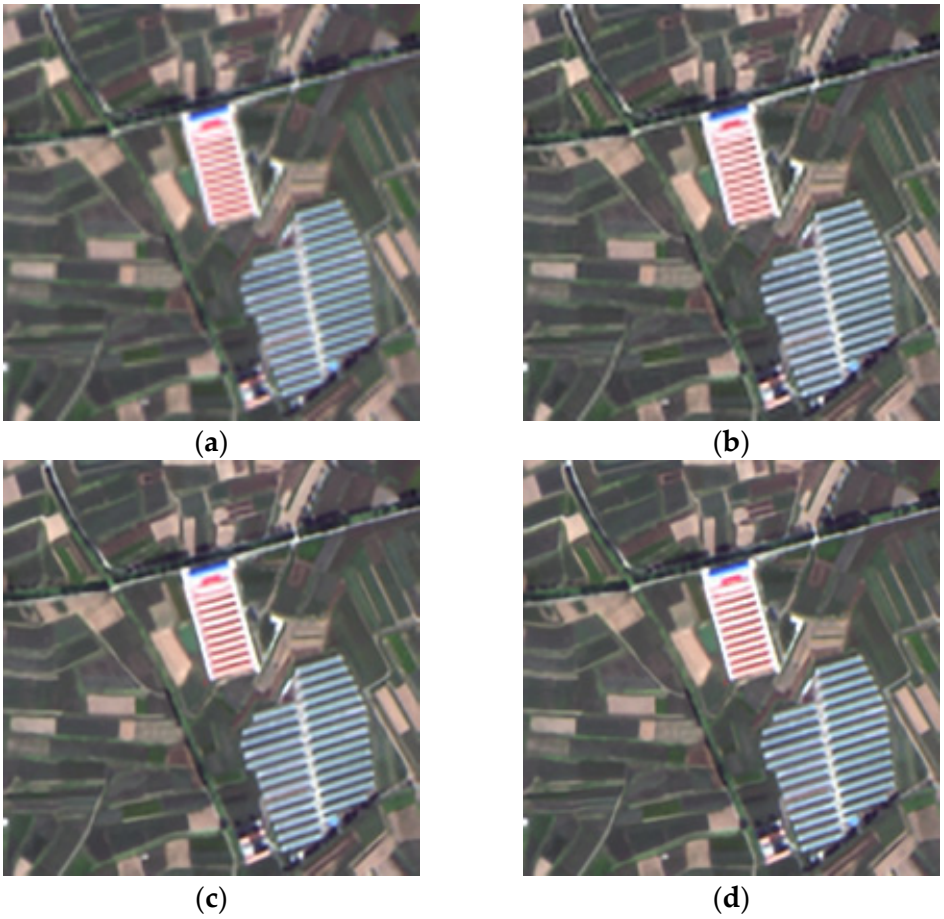


Figure 15. SR results at scale factor of ×2 on the real-world Gaofen-2 dataset [21] using different approaches (**a**–**d**). (**a**) Bicubic; (**b**) SRCNN [5]; (**c**) MHAN [13]; (**d**) Ours.

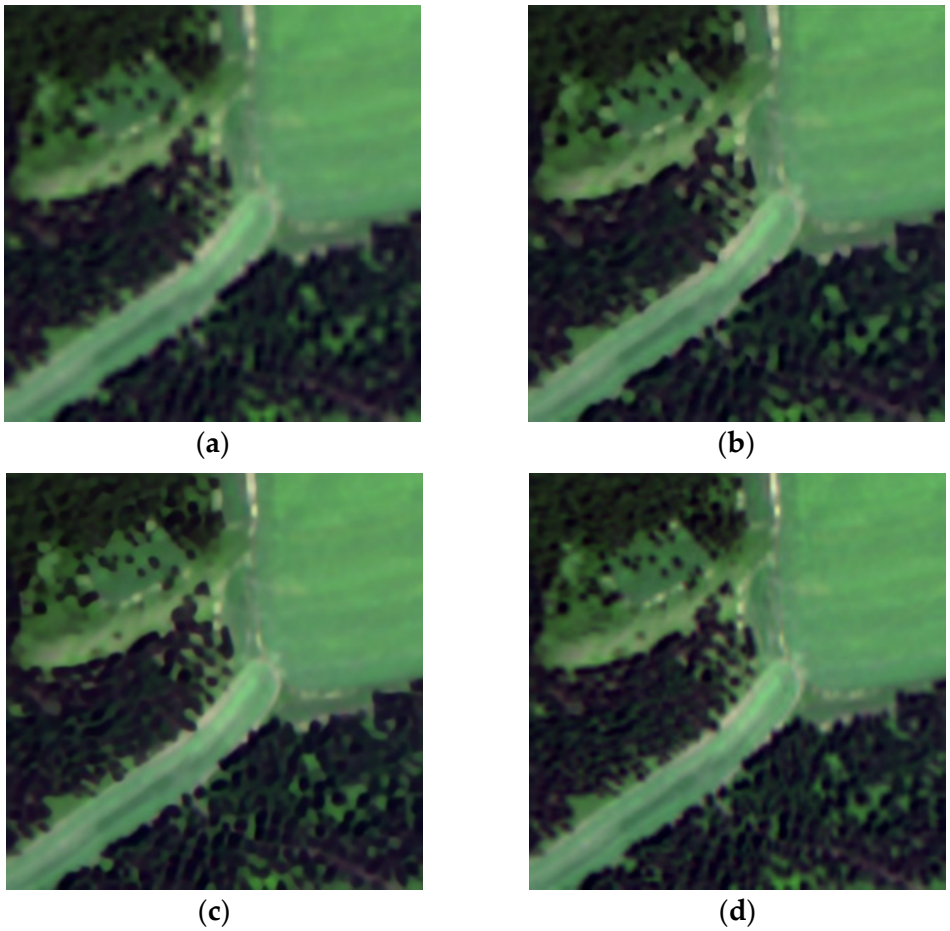


Figure 16. SR results at scale factor of ×4 on the real-world Gaofen-2 dataset [21] using different approaches (**a**–**d**). (**a**) Bicubic; (**b**) SRCNN; (**c**) MHAN; (**d**) Ours.

4.3. Comparison between Time Consumption and Performance before and after Distillation

In this section, we analyze the effectiveness of the proposed feature distillation method through extensive experiments on the RSOD test dataset. The experimental results in Tables 9 and 10 demonstrate that, with the same experimental configuration, the compressed U-Net model reduces the size by nearly 2 times, and the reverse diffusion time for a single image is reduced by approximately 56%. In a quantitative metrics comparison, the performance of the compressed model is only slightly inferior to that of the original model. As shown in Figure 17, the visual comparison between the compressed model and the original model reveals only minor differences that are imperceptible to the human eye. Furthermore, as shown in Table 11, we compared the inference time of our proposed algorithm with that of traditional deep-learning-based end-to-end super-resolution algorithms. It can be seen that our algorithm takes an order of magnitude more time than other algorithms. In our future work, we will address these issues by using a more efficient U-Net network and model compression methods.

Table 9. Comparison between original model and compressed model in terms of parameters, computation, and time consumption. The size of the input image was 256 × 256 pixels, with a scale factor of ×4.

Model	Params (10^6)	GFLOPs	Time (ms) *
Original	9.07	45.2	856
Distillation	4.52	22.6	463

* The reverse diffusion process consumes time.

Table 10. The comparison between quantitative results of the original model and the compressed model on the RSOD test dataset.

RSOD	×2 PSNR/SSIM	×4 PSNR/SSIM	×8 PSNR/SSIM
Original	31.16/0.968	27.86/0.876	25.33/0.710
Distillation	31.10/0.961	27.24/0.865	25.27/0.704

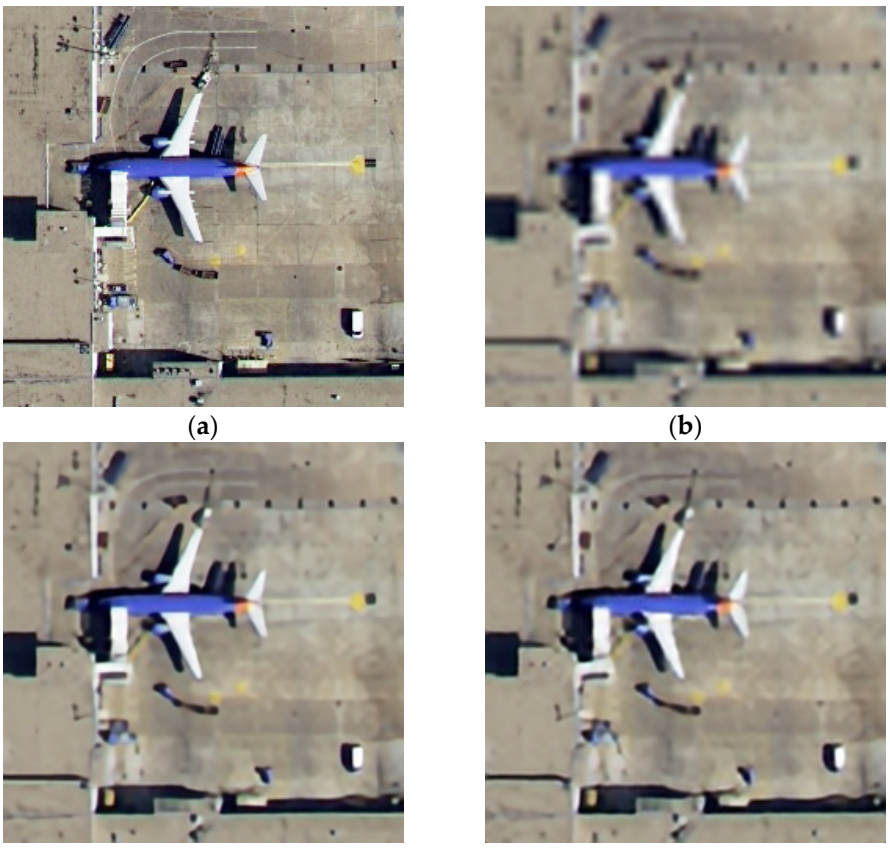


Figure 17. The visual quality comparison between the original model and the compressed model with a scale factor of ×4. (**a**) Ground truth; (**b**) Bicubic; (**c**) Original; (**d**) Distillation.

Table 11. Comparison between the time consumptions of different super-resolution algorithms during the model inference process. The size of the input image was 256 × 256 pixels, with a scale factor of ×4.

Model	SRCNN	VDSR	RDN	MHAN	SAN	DDBPN	EEGAN	Ours
Time (ms) *	1.7	3.0	16	14.8	17.3	36.9	27.5	463

* The time consumed during the inference process of the model.

4.4. Ablation Study

In this section, we demonstrate the importance of the transformer network and CNN in the hybrid conditional feature extraction as well as the high-frequency spatial constraint in our proposed diffusion model through six ablation experiments. All experiments were conducted on the UCMerced_Land test dataset, and the quantitative metrics of PSNR and SSIM were used to evaluate the super-resolution performance. As shown in Table 12, the absence of any of the three components has a negative impact on the objective performance metrics of the generated images. Among them, the high-frequency spatial constraint plays an important role. Even when considering the other two components, the lack of high-

frequency spatial constraints resulted in a decrease of approximately 0.22 dB compared with the best PSNR result.

Table 12. This paper investigated the impact of different module combinations in the proposed hybrid conditional diffusion model on the super-resolution performance of remote-sensing images. All experiments were conducted on the UCMerced_Land test dataset.

Description		**Different Types of Combinations**					
Module		**1**	**2**	**3**	**4**	**5**	**6**
Hybrid conditional feature	Transformer network	✔	✘	✔	✔	✘	✔
	CNN	✘	✔	✔	✘	✔	✔
Fourier high-frequency spatial constraint		✘	✘	✘	✔	✔	✔
×2	PSNR	33.16	33.18	33. 25	33.47	33.59	33.76
	SSIM	0.916	0.918	0.922	0.913	0.921	0.930
×4	PSNR	27.28	27.37	27.48	27.44	27.52	27.60
	SSIM	0.764	0.765	0.771	0.771	0.782	0.788
×8	PSNR	23.34	23.37	23.36	23.50	23.57	23.68
	SSIM	0.548	0.550	0.551	0.572	0.571	0.581

5. Conclusions

In this paper, we proposed a diffusion-model-based framework for remote-sensing image super-resolution, named EHC-DMSR, which utilizes a hybrid conditional diffusion model architecture. The transformer network and CNN are used to extract comprehensive features from low-resolution images, which are then used as guidance in image generation. Furthermore, to constrain the diffusion fusion model and generate more high-frequency information, we proposed a Fourier high-frequency spatial constraint to emphasize high-frequency spatial loss and optimize the reverse diffusion direction. To address the time-consuming issue of the diffusion model in the reverse diffusion process, we proposed a feature-distillation-based model compression method for the diffusion model to reduce the computational load of U-Net, thereby shortening the inference time without affecting the super-resolution performance. Extensive experiments on the synthetic dataset RSOD, real dataset Gaofen-2, and large-scale experiments demonstrated that our proposed algorithm achieves excellent results in both quantitative evaluation metrics and generates clearer, more detailed super-resolution images at high scale factors compared with other advanced algorithms. Although our proposed model achieved excellent visual quality and objective evaluation scores, compared with other learning-based super-resolution algorithms, the inference time of the model is longer due to the use of a more complex transformer architecture to extract global features, which may result in wasted computational resources. In addition, the noise prediction network in our study heavily borrows the U-Net network structure from DDPM, and the influence of the noise prediction model on the diffusion model has not been explored. We hope that researchers can make improvements in the above aspects in the future to promote the practical application of diffusion models in remote-sensing image super-resolution and extend our work to more low-level vision tasks such as image restoration.

Author Contributions: Conceptualization, L.H. and Q.H.; methodology, L.H.; software, L.H.; validation, Y.Z. (Yuchen Zhao), H.L. (Hengyi Lv) and G.B.; formal analysis, Y.Z. (Yisa Zhang); writing—original draft preparation, L.H.; writing—review and editing, Y.Z. (Yuchen Zhao); visualization, Q.H.; supervision, Y.Z. (Yuchen Zhao); project administration, G.B.; funding acquisition, H.L. (Hailong Liu). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (62005269).

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the editors and reviewers for their hard work and valuable advice.

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

References

1. Wang, X.; Yi, J.; Guo, J.; Song, Y.; Lyu, J.; Xu, J.; Yan, W.; Zhao, J.; Cai, Q.; Min, H. A Review of Image Super-Resolution Approaches Based on Deep Learning and Applications in Remote Sensing. *Remote Sens.* **2022**, *14*, 5423. [CrossRef]
2. Liu, J.; Yuan, Z.; Pan, Z.; Fu, Y.; Liu, L.; Lu, B. Diffusion Model with Detail Complement for Super-Resolution of Remote Sensing. *Remote Sens.* **2022**, *14*, 4834. [CrossRef]
3. Ma, W.; Pan, Z.; Yuan, F.; Lei, B. Super-Resolution of Remote Sensing Images via a Dense Residual Generative Adversarial Network. *Remote Sens.* **2019**, *11*, 2578. [CrossRef]
4. Gu, J.; Sun, X.; Zhang, Y.; Fu, K.; Wang, L. Deep Residual Squeeze and Excitation Network for Remote Sensing Image Super-Resolution. *Remote Sens.* **2019**, *11*, 1817. [CrossRef]
5. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 295–307. [CrossRef] [PubMed]
6. Xu, Y.; Li, J.; Song, H.; Du, L.; Muhammad, T. Single-Image Super-Resolution Using Panchromatic Gradient Prior and Variational Model. *Math. Probl. Eng.* **2021**, *2021*, 9944385. [CrossRef]
7. Huang, Y.; Li, J.; Gao, X.; He, L.; Lu, W. Single Image Super-Resolution via Multiple Mixture Prior Models. *IEEE Trans. Image Process.* **2018**, *27*, 5904–5917. [CrossRef] [PubMed]
8. Yang, Q.; Zhang, Y.; Zhao, T.; Chen, Y. Single image super-resolution using self-optimizing mask via fractional-order gradient interpolation and reconstruction. *ISA Trans.* **2018**, *82*, 163–171. [CrossRef]
9. 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, Seattle, WA, USA, 27–30 June 2016; pp. 1646–1654.
10. Zhang, D.; Shao, J.; Li, X.; Shen, H.T. Remote Sensing Image Super-Resolution via Mixed High-Order Attention Network. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 5183–5196. [CrossRef]
11. 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.
12. 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, Long Beach, CA, USA, 16–20 June 2019; pp. 11065–11074.
13. 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, 18–23 June 2018; pp. 2472–2481.
14. 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.
15. ElHaj, K.; Alshamsi, D.; Aldahan, A. GeoZ: A Region-Based Visualization of Clustering Algorithms. *J. Geovisualization Spat. Anal.* **2023**, *7*, 15. [CrossRef]
16. Harrie, L.; Oucheikh, R.; Nilsson, Å.; Oxenstierna, A.; Cederholm, P.; Wei, L.; Richter, K.-F.; Olsson, P. Label Placement Challenges in City Wayfinding Map Production—Identification and Possible Solutions. *J. Geovisualization Spat. Anal.* **2022**, *6*, 16. [CrossRef]
17. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *arXiv* **2020**, arXiv:2006.11239.
18. Li, H.Y.; Yang, Y.F.; Chang, M.; Chen, S.Q.; Feng, H.J.; Xu, Z.H.; Li, Q.; Chen, Y.T. SRDiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing* **2022**, *479*, 47–59. [CrossRef]
19. Long, Y.; Gong, Y.; Xiao, Z.; Liu, Q. Accurate Object Localization in Remote Sensing Images Based on Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2486–2498. [CrossRef]
20. 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.
21. Tong, X.-Y.; Xia, G.-S.; Lu, Q.; Shen, H.; Li, S.; You, S.; Zhang, L. Land-cover classification with high-resolution remote sensing images using transferable deep models. *Remote Sens. Environ.* **2020**, *237*, 111322. [CrossRef]
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 27–30 June 2016; pp. 770–778.
23. Haris, M.; Shakhnarovich, G.; Ukita, N. Deep back-projection networks for super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1664–1673.
24. Li, Z.; Yang, J.; Liu, Z.; Yang, X.; Jeon, G.; Wu, W. Feedback network for image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3867–3876.
25. 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]
26. Pan, Z.; Ma, W.; Guo, J.; Lei, B. Super-resolution of single remote sensing image based on residual dense backprojection networks. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 7918–7933. [CrossRef]

27. Chang, Y.; Luo, B. Bidirectional Convolutional LSTM Neural Network for Remote Sensing Image Super-Resolution. *Remote Sens.* **2019**, *11*, 2333. [CrossRef]
28. Jiang, K.; Wang, Z.; Yi, P.; Wang, G.; Lu, T.; Jiang, J. Edge-Enhanced GAN for Remote Sensing Image Superresolution. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5799–5812. [CrossRef]
29. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative Adversarial Networks: An Overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [CrossRef]
30. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
31. Rezende, D.; Mohamed, S. Variational inference with normalizing flows. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 1530–1538.
32. Thanh-Tung, H.; Tran, T. Catastrophic forgetting and mode collapse in GANs. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Neural Network, Glasgow, UK, 19–24 July 2020; pp. 1–10.
33. Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 2256–2265.
34. Kong, Z.; Ping, W.; Huang, J.; Zhao, K.; Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. *arXiv* **2020**, arXiv:2009.09761.
35. Batzolis, G.; Stanczuk, J.; Schönlieb, C.-B.; Etmann, C. Conditional image generation with score-based diffusion models. *arXiv* **2021**, arXiv:2111.13606.
36. Song, Y.; Sohl-Dickstein, J.; Kingma, D.P.; Kumar, A.; Ermon, S.; Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv* **2020**, arXiv:2011.13456.
37. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10684–10695.
38. Kim, G.; Kwon, T.; Ye, J.C. Diffusionclip: Text-guided diffusion models for robust image manipulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 2426–2435.
39. Choi, J.; Kim, S.; Jeong, Y.; Gwon, Y.; Yoon, S. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv* **2021**, arXiv:2108.02938.
40. Chung, H.; Sim, B.; Ye, J.C. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12413–12422.
41. Saharia, C.; Chan, W.; Chang, H.; Lee, C.; Ho, J.; Salimans, T.; Fleet, D.; Norouzi, M. Palette: Image-to-image diffusion models. In Proceedings of the ACM SIGGRAPH 2022 Conference Proceedings, Vancouver, BC, Canada, 7–11 August 2022; pp. 1–10.
42. Lu, Z.; Li, J.; Liu, H.; Huang, C.; Zhang, L.; Zeng, T. Transformer for single image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 457–466.
43. Brigham, E.O.; Morrow, R.E. The fast Fourier transform. *IEEE Spectr.* **1967**, *4*, 63–70. [CrossRef]
44. Pandey, S.; Singh, M.P.; Pandey, V. Image transformation and compression using Fourier transformation. *Int. J. Curr. Eng. Technol.* **2015**, *5*, 1178–1182.
45. Fuoli, D.; Van Gool, L.; Timofte, R. Fourier space losses for efficient perceptual image super-resolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2360–2369.
46. Chen, W.; Peng, L.; Huang, Y.; Jing, M.; Zeng, X. Knowledge Distillation for U-Net Based Image Denoising. In Proceedings of the 2021 IEEE 14th International Conference on ASIC (ASICON), Kunming, China, 26–29 October 2021; pp. 1–4.
47. 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]
48. Zou, Q.; Ni, L.; Zhang, T.; Wang, Q. Deep Learning Based Feature Selection for Remote Sensing Scene Classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2321–2325. [CrossRef]
49. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
50. Nichol, A.Q.; Dhariwal, P. Improved denoising diffusion probabilistic models. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 8162–8171.
51. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.M.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019.