Article

Image Degradation Modeling for Real-World Super Resolution via Conditional Normalizing Flow

Wang Xu *, Renwen Chen, Qinbang Zhou and Fei Liu

State Key Laboratory of Mechanics and Control of Mechanical Structures, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; rwchen@nuaa.edu.cn (R.C.); zhouqinbang@nuaa.edu.cn (Q.Z.); liufei@nuaa.edu.cn (F.L.)

* Correspondence: xuwang@nuaa.edu.cn

Abstract: In recent years, deep-learning-based super-resolution (SR) methods have obtained impressive performance gains on synthetic clean datasets, but they fail to perform well in real-world scenarios due to insufficient real-world training data. To tackle this issue, we propose a conditional-normalizing-flow-based method named IDFlow for image degradation modeling that aims to generate various degraded low-resolution (LR) images for real-world SR model training. IDFlow takes image degradation modeling as a problem of learning a conditional probability distribution of LR images given the high-resolution (HR) ones, and learns the distribution from existing real-world SR datasets. It first decomposes the image degradation modeling into blur degradation modeling and real-world noise modeling. It then utilizes two multi-scale invertible networks to model these two steps, respectively. Before applied into real-world SR, IDFlow is first trained supervisedly on two real-world datasets RealSR and DPED with negative log-likelihood (NLL) loss. It is then used to generate a large number of HR-LR image pairs from an arbitrary HR image dataset for SR model training. Extensive experiments on IDFlow with RealSR and DPED are conducted, including evaluations on image degradation stochasticity, degradation modeling, and real-world super resolution. Two known SR models are trained with IDFlow and named as IDFlow-SR and IDFlow-GAN. Testing results on the RealSR and DPED testing dataset show that not only can IDFlow generate realistic degraded images close to real-world images, but it is also beneficial to real-world SR performance improvement. IDFlow-SR achieves $4\times$ SR performance gains of 0.91 dB and 0.161 in terms of image quality assessment metrics PSNR and LPIPS. Moreover, IDFlow-GAN can super-resolve real-world images in the DPED testing dataset with richer textures and maintain clearer patterns without visible noises when compared with state-of-the-art SR methods. Quantitative and qualitative experimental results well demonstrate the effectiveness of the proposed IDFlow.

Citation: Xu, W.; Chen, R.; Zhou, Q.; Liu, F. Image Degradation Modeling for Real-World Super Resolution via Conditional Normalizing Flow. *Appl. Sci.* **2021**, *11*, 4735. <https://doi.org/10.3390/app11114735>

Academic Editor: Federico Divina

Received: 9 March 2021

Accepted: 17 May 2021

Published: 21 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: image reconstruction; super resolution; image degradation modeling; deep learning; normalizing flow

1. Introduction

Image super resolution (SR) targets the recovery of a visually pleasing high-resolution (HR) image from its low-resolution (LR) version. With the rapid development of deep learning and convolution neural networks (CNNs), CNN-based methods [1–5] have made impressive SR performance gains in recent years. However, most of the existing methods are trained supervisedly on synthetic paired datasets where LR images are commonly obtained by bicubic interpolation from HR images. Even though this kind of data acquisition could provide fine results in clean settings, it causes a data distribution mismatch between real-world images and synthetic images. Bicubic down-sampling would alter image characteristics, as visible corruptions such as sensor noise in natural images are reduced in the meantime. On the other hand, the real-world image degradation is much more complicated than a single down-sampling process. Thus, the state-of-the-art SR methods trained on the synthetic datasets cannot perform as well as expected in real-world scenarios.

In order to improve the generalization ability of SR models, many works have been proposed [6–10]. Generally, image degradation could be formulated as a combination of blurring and noise addition. Following the definition, Zhang et al. [6] introduced anisotropic Gaussian blur kernels as well as additive white Gaussian noise into HR-LR pair generation to simulate real image characteristics. They then proposed and trained a CNN model, SRMD [6], for blind super resolution. Since SRMD takes degradation maps as input as well, SRMD has to find suitable degradation parameters on real images via a grid search strategy, which is time-consuming. Recently, some works have made efforts to collect paired image datasets from real-world scenarios. Chen et al. [7] captured the images of 100 city scenes printed on postcards based on two imaging systems (DSLR and smartphone) at different focal lengths and obtained a real-world paired dataset known as City100 [7]. Cai et al. [8] collected paired HR-LR images on the same scene by adjusting the focal length of a digital camera. They also developed a registration algorithm to align the image pairs. A real-world SR dataset named RealSR [8] was built with three different scale factors. However, obtaining such paired training datasets requires expensive digital camera devices and needs complex image post-processing to align LR and HR images. As a result, the number of image pairs and the diversity of scenes are insufficient. City100 [7] only contains 100 aligned HR-LR image pairs, while RealSR [8] contains 559 scenes in total, which limits the further applications of SR models into a wider range of scenarios.

Besides collecting real-world datasets, there are some methods [9,11,12] proposed to construct HR-LR images using unsupervised or semi-supervised learning. These methods utilize a Generative Adversarial Network (GAN) to learn the distribution of real-world LR images. However, training a GAN based network to model the degradation process with unpaired LR and HR images suffers from training instability and may lead to undesired results. Furthermore, given an HR image, only one generated LR result can be predicted once the network is trained, which does not take account for the fact that multiple LR images from one scene may have the same HR version.

As discussed above, to improve the performance of existing SR models in real-world scenarios lies in how to effectively build training SR datasets that is close to reality. To address this issue, we propose a method based on conditional normalizing flow named IDFlow to investigate image degradation modeling (IDM). Unlike deterministic mapping learning in [9,12], IDFlow takes IDM as the problem of learning a conditional probability distribution of degraded LR images given the HR versions. It learns the distribution from the existing real-world datasets and is trained with a single negative log likelihood (NLL) loss in a supervised manner. Once trained, IDFlow can easily generate multiple degraded LR images given the HR version by exploring the learnt distribution, maintaining the stochastic nature of image degradation. Thus, we introduce IDFlow to generate large-scale realistic training data from an arbitrary HR image dataset for SR model training. We conduct extensive experiments on existing real-world datasets. The results demonstrate that not only can IDFlow generate realistic LR images close to the real-world images, but it can also further improve the real-world performance of the existing SR models. Compared with the state-of-the-art SR methods, the SR models trained with IDFlow can produce cleaner SR results with richer textures from real-world noisy images.

2. Related Work

2.1. Real-World Super Resolution

In recent years, CNN-based SR methods have been the mainstream in the field. Since Dong et al. [1] proposed SRCNN, numerous CNN architectures have been proposed for SR [2,3,6,13]. Even though these methods achieve impressive performance in terms of fidelity, they would generate visually blurry images, especially for a large-scale factor. To tackle this drawback, various Generative Adversarial Network (GAN)-based SR methods [4,14,15] (e.g., ESRGAN [4]) are proposed. They introduce adversarial loss as well as perceptual loss to further enhance the texture details of SR images. However, these

methods are trained on synthetic clean data that do not contain any noisy or blurry data, leading to poor performance in real-world scenarios.

To alleviate this issue, methods [6,16] have been proposed by introducing synthetic noisy and blurry data into model training and testing. Though these methods enhance the real-world performance and robustness of the SR model, they need to take the prior about blur/noise as part of the input, therefore limiting the scope of application. Shocher et al. [17] proposed ZSSR, which learns the internal statistics of test images during testing. Meanwhile, some recent works have focused on collecting paired HR-LR real-world images. Chen et al. [7] captured the images of 100 city scenes based on two imaging systems and obtained a real-world paired dataset known as City100 [7]. Cai et al. [8] collected paired HR-LR images on the same scene by adjusting the focal length of a digital camera and proposed a RealSR [8] dataset (559 scenes). However, it requires a high degree of manual costs and strict conditions to collect such paired datasets. The number and diversity of these datasets are limited.

2.2. Degradation Learning

In order to solve the problem of SR methods in real applications, some works have been proposed to model the image degradation. Bulat et al. [12] first utilized a GAN-based network to learn degradation from HR to LR in an unsupervised manner. Later, Fritsche et al. [9] proposed DSGAN to learn image degradation via frequency separation. Bell-Kligler et al. [11] proposed KernelGAN to estimate the blur kernels of corrupted images and combined ZSSR as K-ZSSR [11] for SR application. However, these methods have problems of convergence and mode collapse, and the losses need to be carefully fine-tuned. More importantly, both of them try to learn deterministic mapping from HR to LR, ignoring the stochastic nature of image degradation.

2.3. Normalizing Flow

Normalizing Flow [18] is a branch of generative model approaches that has received less attention. It aims to parametrize a complex distribution $p_{\mathbf{y}}(\mathbf{y})$ into a simple distribution $p_{\mathbf{z}}(\mathbf{z})$ (e.g., standard Gaussian) using an invertible neural networks f_{θ} , and it maps the samples from $\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$ using the reverse forward step $\mathbf{y} = f_{\theta}^{-1}(\mathbf{z})$ during inference. Thus, the network can be trained directly by minimizing the negative log-likelihood $-\log p_{\mathbf{y}}(\mathbf{y})$ with any back-propagation algorithms. In recent years, various works [19–21] have been proposed to improve the performance of normalized flow on image generation tasks, but there are few works on image degradation learning. In this paper, we will learn the image degradation via conditional normalizing flow in a supervised scheme from existing real-world SR datasets.

3. Materials and Methods

3.1. Image Degradation Formulation

Given an HR image $\mathbf{I}_{HR}$, the corresponding LR observation $\mathbf{I}_{LR}$ is generally assumed to follow the image degradation as follows:

$$\mathbf{I}_{LR} = (\mathbf{I}_{HR} * \mathbf{k}) \downarrow_s + \mathbf{n}, \quad (1)$$

where $\downarrow_s$ denotes a down-sampling operation, $*$ denotes a convolution operation, and $\mathbf{k}$ and $\mathbf{n}$ denote blur kernel and random noise, respectively. However, most existing SR methods assume a simple and uniform degradation, which utilizes a fixed blur kernel (e.g., Bicubic kernel) to generate HR-LR images for SR model training. In real-world scenarios, it is a challenge to manually model the image degradation process, which is usually susceptible to a combination of sensor noise and post-processing artifacts. Thus, the degradation process can be expressed as

$$\mathbf{y}_n = \mathbf{y} + \mathbf{n} = f_b(\mathbf{x}) + \mathbf{n}, \quad (2)$$

where $\mathbf{y}_n$ and $\mathbf{y}$ denote the corrupted noisy LR image and its clean version, $\mathbf{x}$ denote the HR image, and f_b denotes the blur degradation function.

Considering the stochasticity of image degradation, we take IDM as the problem of learning a conditional probability distribution of degraded LR images given the HR input. We propose a conditional normalizing flow based method named IDFlow to model image degradation and learn the distribution $p_{\mathbf{y}_n|\mathbf{x}}(\mathbf{y}_n|\mathbf{x})$ from existing real-world SR datasets. As shown in Figure 1, we decompose the IDM process into two steps: blur degradation modeling (BDM) and real-world noise modeling (RNM). Blur degradation modeling aims to model the degradation process from $\mathbf{x}$ to $\mathbf{y}$, while real-world noise modeling aims to model the noise addition process.

Figure 1. Overall architecture of IDFlow for image degradation modeling. The blue solid lines correspond to the training procedure, and the blue dotted lines correspond to the inference procedure.

3.2. Image Degradation Modeling via Conditional Normalizing Flow

For blur degradation modeling, we parametrize the conditional probability distribution $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x})$ of $\mathbf{y}$ corresponding to $\mathbf{x}$ with an invertible neural network $f_{b,\theta}$. The image space of HR-LR image pairs is mapped via $f_{b,\theta}$ into a latent space:

$$\mathbf{z}_b = f_{b,\theta}(\mathbf{y}; \mathbf{x}), \quad (3)$$

where $\mathbf{z}_b \sim p_{\mathbf{z}_b}(\mathbf{z}_b)$ denotes the latent variable. Since $f_{b,\theta}$ is required to be invertible, LR image $\mathbf{y}$ can be generated from a sampled latent variable $\mathbf{z}_b$ once given an HR image $\mathbf{x}$ as

$$\mathbf{y} = f_{b,\theta}^{-1}(\mathbf{z}_b; \mathbf{x}). \quad (4)$$

The distribution of latent space $p_{\mathbf{z}_b}(\mathbf{z}_b)$ can be simply assumed to follow a Gaussian distribution $\mathbf{z}_b \sim \mathcal{N}(0, \mathbf{I})$. Thus, the probability density $p_{\mathbf{y}|\mathbf{x}}$ can be explicitly computed as:

$$p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}, \theta) = p_{\mathbf{z}_b}(f_{b,\theta}(\mathbf{y}; \mathbf{x})) \left| \det \frac{\partial f_{b,\theta}}{\partial \mathbf{y}}(\mathbf{y}; \mathbf{x}) \right|. \quad (5)$$

It is obtained by applying a variable change formula to the density, where the second term is the final volume scaling given by the determinant of Jacobian $\frac{\partial f_{b,\theta}}{\partial \mathbf{y}}$. The Equation (5)

allows to train the network and optimize the parameters by minimizing the negative log-likelihood(NLL) loss:

$$\mathcal{L}(\theta; \mathbf{x}, \mathbf{y}) = -\log p_{\mathbf{z}_b}(f_{\theta}(\mathbf{y}; \mathbf{x})) - \log \left| \det \frac{\partial f_{b,\theta}}{\partial \mathbf{y}}(\mathbf{y}; \mathbf{x}) \right|. \quad (6)$$

Once the distribution is learnt, diverse realistic LR images can be generated by applying the inverse flow $\mathbf{y} = f_{b,\theta}^{-1}(\mathbf{z}_b; \mathbf{x})$. Furthermore, to ease the expression of the second term, the network $f_{b,\theta}$ is decomposed into a combination of N cascaded invertible flow layers. The output of the n -th layer can be expressed as:

$$\mathbf{h}^{n+1} = f_{\theta}^n(\mathbf{h}^n; \mathbf{u}), \quad (7)$$

where $\mathbf{u} = g_{hr}(\mathbf{x})$ is encoding feature of the input HR image $\mathbf{x}$ generated by an HR image encoder $f_{encoder}$. Specially, $\mathbf{h}^0 = \mathbf{y}$ and $\mathbf{z}_b = \mathbf{h}^N$. Then the Equation (6) can be expressed as:

$$\mathcal{L}(\theta; \mathbf{x}, \mathbf{y}) = -\log p_{\mathbf{z}_b}(\mathbf{z}_b) - \sum_{n=0}^{N-1} \log \left| \det \frac{\partial f_{\theta}^n}{\partial \mathbf{h}^n}(\mathbf{h}^n; \mathbf{u}) \right|. \quad (8)$$

The second term can be efficiently computed by summing the log-determinant of Jacobian $\frac{\partial f_{\theta}^n}{\partial \mathbf{h}^n}$ in each layer. Moreover, each layer is required to be invertible and allow a tractable Jacobian determinant.

For real-world noise modeling, another invertible neural network $f_{n,\theta}$ is utilized. Following [10], we collect noise patches from real-world images by setting max noise variance to a certain range and use the patches to train $f_{n,\theta}$, the objective function can be expressed as:

$$\mathcal{L}(\theta; \mathbf{n}) = -\log p_{\mathbf{z}_n}(\mathbf{z}_n) - \sum_{n=0}^{N-1} \log \left| \det \frac{\partial f_{n,\theta}^n}{\partial \mathbf{h}^n}(\mathbf{h}^n; \mathbf{u}_n) \right|, \quad (9)$$

where $\mathbf{u}_n = \bar{\mathbf{n}}$ is the global mean of input noise patch $\mathbf{n}$ and is extended to have the same spatial resolution as the input, and $\mathbf{z}_n$ is assumed to follow $\mathbf{z}_n \sim \mathcal{N}(0, I)$ as well.

The training procedure is illustrated by the blue solid lines in Figure 1. For blur degradation modeling, HR images are first encoded by the network of the HR encoder. Clean LR images along with HR encoding features are then processed through a series of flow steps in BDM and transformed to the corresponding latent variables $\mathbf{z}_b$, as shown in Figure 1. For real-world noise modeling, the noise patches are taken as an input of RNM to produce the latent variables of noise $\mathbf{z}_n$. The variables are later used to calculate NLL losses for model optimization.

3.3. Degraded Image Generation

Given one HR image $\mathbf{x}^{(i)}$, the generation of a noisy degraded LR image $\mathbf{y}_n^{(i)}$ during inference is indicated by the dotted blue lines in Figure 1. IDFlow first samples two latent variables $\mathbf{z}_b^{(i)}$ and $\mathbf{z}_n^{(i)}$ from the Gaussian distribution. It then takes $\mathbf{z}_b^{(i)}$ and $\mathbf{x}^{(i)}$ as input and uses the reverse flow of BDM to generate synthetic clean LR image $\mathbf{y}^{(i)}$. The global average pooling is performed on $\mathbf{y}^{(i)}$ as the RNM condition feature $\mathbf{u}_n^{(i)}$. $\mathbf{z}_n^{(i)}$ and $\mathbf{u}_n^{(i)}$ are taken as input via the reverse flow of RNM to generate synthetic noise output $\mathbf{n}^{(i)}$. At last, the noisy LR image $\mathbf{y}_n^{(i)}$ is obtained by adding $\mathbf{n}^{(i)}$ to $\mathbf{y}^{(i)}$.

Thus, IDFlow can explore the learnt conditional distribution $p_{\mathbf{y}_n|\mathbf{x}}(\mathbf{y}_n|\mathbf{x})$ by sampling various degraded LR images via $\mathbf{y}_n^{(i)} = f_{IDFlow}^{-1}(\mathbf{z}_b^{(i)}, \mathbf{z}_n^{(i)}; \mathbf{x}), \mathbf{z}_b^{(i)}, \mathbf{z}_n^{(i)} \sim p_{\mathbf{z}}$ given one HR image $\mathbf{x}$. Here, $f_{IDFlow}^{-1}(\cdot)$ is the reverse function of IDFlow. We simply utilize a Gaussian distribution $\mathcal{N}(0, \tau)$ with variance τ , which is also called sampling temperature. As discussed in [21], the better results are achieved when sampling with a slightly lower variance.

3.4. Model Architecture

Following the previous works [20,21], we utilize a multi-level architecture of normalizing flow for proposed IDFlow. As shown in Figure 1, we use a three-level flow model for blur degradation modeling and a 1-level flow model for real-world noise modeling. In each level, there are 12 flow steps with a *Squeeze* layer at the head and a *Split* layer at the tail. The *Squeeze* layer is a space-to-depth operator that converts a tensor with a size of $2H \times 2W \times C$ into a tensor with size of $H \times W \times 4C$. The *Split* layer just splits tensor equally in the channel dimension. Each flow step consists of four different flow layers: conditional affine coupling (CAC), an affine injector (AI), invertible 1×1 convolution (invConv 1×1), and Actnorm. More details of these flow layers can be found in [20,21].

Conditional Affine Coupling is an extended version of the affine coupling layer used in Glow [21]. The input feature $\mathbf{h}^n$ is split equally in channel dimension into $(\mathbf{h}_1^n, \mathbf{h}_2^n)$ at first, the $\mathbf{h}_1^n$ as well as condition feature $\mathbf{u}$ is taken into a neural network to generate a scaling factor and a bias factor. The two factors are applied to normalize $\mathbf{h}_2^n$. This layer can be expressed as

$$\mathbf{h}_2^{n+1} = f_{\theta,s}^n(\mathbf{h}_1^n; \mathbf{u}) \cdot \mathbf{h}_2^n + f_{\theta,b}^n(\mathbf{h}_1^n; \mathbf{u}), \mathbf{h}_1^{n+1} = \mathbf{h}_1^n, \mathbf{h}^{n+1} = (\mathbf{h}_1^{n+1}, \mathbf{h}_2^{n+1}), \quad (10)$$

where $f_{\theta,s}^n$ and $f_{\theta,b}^n$ denote an arbitrary CNN network.

The Affine Injector [20] takes condition feature $\mathbf{u}$ to generate scaling and bias factors to normalize the input feature as $\mathbf{h}^{n+1} = f_{\theta,s}^n(\mathbf{u}) \cdot \mathbf{h}^n + f_{\theta,b}^n(\mathbf{u})$.

Invertible 1×1 convolution scales feature with an invertible matrix as $\mathbf{h}^{n+1} = W * \mathbf{h}^n$. Here we use the standard non-factorized formulation as in [21].

Actnorm [21] normalizes input features in a channel dimension with a learnable scaling factor and bias factor.

For the HR image encoder, a CNN-based network is utilized with no need to be invertible. The details of network architectures are shown in Figure 1. The RRDB denotes the residual-in-residual dense block used in [4]. We set the number of HR encoder output channel to be 64. The 3×3 convolution layer is used as default. Moreover, for the networks of $f_{\theta,s}^n$ and $f_{\theta,b}^n$, we use a structure of two 3×3 convolution layers with an ReLU activation layer.

3.5. Datasets

To conduct experiments on IDFlow, we chose three different image datasets: RealSR [8], DF2K, and DPED [22]. Moreover, we split these datasets used for training and testing. The summarized information about the datasets are reported in Table 1.

RealSR: The RealSR [8] dataset contains 559 real-world paired HR-LR images taken by DSLR cameras in real scenarios. We used the third version of this dataset, which contains HR and LR images at different resolutions with a scale factor of $\times 4$. Following [8], we split the dataset into two parts: 459 images for training, denoted as the RealSR-train, and 100 images only for testing, denoted as the RealSR-Test. RealSR-Train is used for the training of blur degradation modeling in IDFlow.

DF2K: The DF2K dataset combines both the DIV2K [13] and Flickr2K [4] datasets, totally containing 3450 high-resolution images of diverse scenes. We used the DF2K dataset as HR source images to generate HR-LR training pairs for SR model training.

DPED: The DPED [22] dataset is a real-world image dataset that has 5614 images directly taken by an iPhone 3 mobile phone, denoted as DPED-Train. Moreover, it has a testing set that contains 100 cropped images, denoted as DPEP-Test. The images have sensor noise, blur, and unknown artifacts. We collected about 3500 noise patches from DPED-Train with a size of 256×256 for the training of real-world noise modeling in IDFlow, denoted as DPED-noises. Specially, the DPED dataset has no HR ground-truth available.

Table 1. Summarized information about image datasets.

Datasets	Number	Avg. Resolution		Format
		HR	LR	
DF2K	3450	(1446, 1942)	-	PNG
RealSR-Train	459	(1312, 1527)	(328, 381)	PNG
RealSR-Test	100	(1247, 1569)	(312, 392)	PNG
DPED-Train	5614	-	(768, 1024)	PNG
DPED-Test	100	-	(256, 511)	PNG

3.6. Evaluation Metrics

For any image with ground truth available, we introduced two reference-based image quality assessment (IQA) metrics, the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) [23], to evaluate image quality. They can measure the similarity and fidelity between input image and reference image. We also introduced another reference-based IQA metric LPIPS [24], which focuses on image perceptual quality. It is a learned evaluation metric based on a fine-tuned AlexNet, which has been shown to be more consistent with human perception. For images without ground truth, we conducted an image quality evaluation based on no-reference IQA metrics NIQE [25], BRISQUE [26], and PIQE [27]. All the metrics were calculated on RGB images.

3.7. Training Results

We trained the proposed IDFlow with only the NLL loss as mentioned above for 60 k iterations. During each training batch, we sampled 16 RGB HR-LR pairs randomly cropped from the original training images in RealSR-Train for blur degradation modeling. The cropped HR patch size was set to 128×128 . For real-world noise modeling, we used 16 noise patches with a size of 128×128 randomly cropped from noisy image patches in DPED noises. Data augmentation was performed during training, including random rotation by 90° , as well as horizontal and vertical flipping. We utilized the Adam optimizer [28] to optimize IDFlow training with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\varepsilon = 10^{-8}$. The learning rate was first set to 2×10^{-4} and decreased to half after 30 k, 40 k, and 50 k training iterations.

To illustrate that IDFlow can help promote the performance of existing SR methods in real-world scenarios, two known SR models RRDBNet [4] and ESRGAN [4] were used for $\times 4$ super resolution in subsequent experiments. For the model RRDBNet, we used L1 loss to train RRDBNet for 60 k iterations. The learning rate was set to 2×10^{-4} and decreased to 1×10^{-4} after 50 k training iterations. Moreover, ten images were selected from RealSR-Test as the validation set to test the model SR performance after every 2 k iterations. The model with the best PSNR results was used for experiments. For the model ESRGAN, we trained it for 60 k iterations, and, based on the training code provided by [10], the losses for training included L1 loss, perceptual loss, and GAN loss [4]. The final trained model was used for experiments. The learning rate was initialized as 1×10^{-4} and decreased to half after 5 k, 10 k, 20 k, and 30 k training iterations. Both RRDBNet and ESRGAN training were optimized with the Adam optimizer with the default setting. We implemented IDFlow as well as SR models by using the PyTorch framework [29] on a desktop workstation equipped with an NVIDIA 1080Ti GPU.

4. Experimental Results and Discussion

4.1. Evaluation on Image Degradation Stochasticity

In this section, we conduct analyses on the image degradation stochasticity of IDFlow. Figure 2 shows the multiple degraded LR examples generated by IDFlow with blur degradation modeling given the same HR image with $4\times$ down-sampling. For each HR image, the first row lists the zoomed details of five degraded LR images. The second row presents the first LR image as well as absolute pixel-value difference maps between the first LR image and the other four images, to highlight the differences between LR images.

From these maps, it can be seen that both of these synthetic LR images retain the overall structural information similar to the original HR image. Meanwhile, the LR images are different in the areas with more high-frequency information, which is consistent with the real-world image degradation. This shows that the image degradation realized by IDFlow has the characteristics of diversity and stochasticity.

Figure 2. Random samples generated by IDFlow with $\times 4$ down-sampling using a sampling temperature $\tau = 0.7$. For each HR image in the first column, the first row presents zoomed local regions of five synthetic LR examples. The second row presents the absolute pixel-value difference maps between the first LR image and the rest of the images.

Furthermore, by adjusting the value of sampling temperature τ , we can control the diversity and stochasticity of IDFlow. We conduct experiments on RealSR-Test. For each τ , IDFlow takes an HR image as input and generates one synthetic LR image. The synthetic LR images are used to calculate IQA metrics PSNR, SSIM, and LPIPS with the LR images in RealSR-Test. IQA metrics can reflect the degradation performance of IDFlow. To illustrate the diversity of IDFlow, we let IDFlow generate 10 LR images for each testing HR image, and we then compute the pixel standard deviation of these LR images for each HR image. The average of the pixel standard deviations of all tested images in RealSR-Test can be regarded as the diversity. If the diversity is larger, the difference between any two LR images becomes larger. The results are reported in Figure 3. We also introduce the results of

two learning-based image degradation modeling methods KernelGAN [11] and DSGAN [9] for comparison. They are trained with Real-Train and then used for testing.

Since KernelGAN and DSGAN can only produce one LR image given an HR image, the diversity of these methods are zero, and the results of the IQA metrics will not change with the sampling temperature. As for IDFlow, when sampling temperature τ decreases, the generated results are pixel-wisely closer to the original images, the curve of the PSNR shows an increasing trend, and the diversity of the generated results decreases accordingly. At the same time, the structural similarity and perceptual similarity of the generated results have not changed significantly. Specially, when $\tau = 0$, the diversity decreases to zero, which means each HR image could only have one LR image generated by IDFlow, which is the same as KernelGAN and DSGAN. Thus, we set the $\tau = 0.7$ as a tradeoff between fidelity and diversity.

Figure 3. The effect of different τ on degradation performance of IDFlow. PSNR, diversity, SSIM, and LPIPS results are reported. The results of DSGAN and KernelGAN are introduced as a reference.

4.2. Evaluation on Image Degradation Modeling

To validate the effectiveness of the proposed IDFlow, we compared IDFlow with existing IDM methods. Firstly, since Bicubic is the most widely used degradation method for SR, we used it as the baseline. Moreover, learning-based methods KernelGAN [11] and DSGAN [9] were introduced for comparison. We also introduced a CNN-based IDM method named baselineCNN, which utilizes a CNN that has the same structure as the HR encoder of IDFlow with an additional convolution layer at the tail. baselineCNN tries to learn the direct mapping from HR to LR in a supervised manner. All the compared methods were trained on RealSR-Train and tested on RealSR-Test with $4\times$ down-sampling. During testing, the IDM methods take HR images in RealSR-Test as input to generate synthetic LR images. Subsequently, these images were used to calculate IQA metrics with LR images in RealSR-Test. The IQA results measure the similarity between the synthetic results and ground-truth LR images and thus can be used to evaluate image degradation performance.

Table 2 shows the degradation performance results of the compared methods on RealSR-Test with $4\times$ down-sampling. It can be seen that IDFlow outperforms Bicubic,

KernelGAN, and DSGAN on all the IQA metrics, demonstrating that IDFlow can effectively model the image degradation process. Benefiting from pixel-wisely supervision, baselineCNN achieves better degradation performance than IDFlow.

Table 2. Degradation performance of different image degradation modeling methods on RealSR-Test with $4\times$ down-sampling. $\uparrow$ indicates that the higher the value, the better the result. $\downarrow$ indicates that the lower the value, the better the result.

Method	PSNR (dB) $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Bicubic	29.16	0.901	0.150
KernelGAN	30.01	0.928	0.077
DSGAN	32.23	0.953	0.049
baselineCNN	34.00	0.963	0.038
IDFlow	33.44	0.959	0.042

The main purpose of image degradation modeling for SR is to generate realistic training datasets to improve SR performance. Thus, we introduced these compared IDM methods into $4\times$ SR model training to verify their effects on SR. The experimental scheme is shown in Figure 4. Different image degradation methods were first applied to generate synthetic 3450 HR-LR training image pairs based on the DF2K dataset with $4\times$ down-sampling. Afterwards, different synthetic datasets were used to train a known SR model RRDBNet [4] with a scale of $4\times$ with the same training settings. SR models trained with Bicubic, DSGAN, baselineCNN, and IDFlow are named Bicubic-SR, DSGAN-SR, baselineCNN-SR, and IDFlow-SR. Specially, since KernelGAN needs no training process, it will generate SR images using ZSSR [17] during testing, denoted as K-ZSSR. After training, these trained SR models were tested on RealSR-Test. They took the LR images as input and generated $4\times$ SR results to compute IQA metrics PSNR/SSIM/LPIPS with HR ground truth. The SR performance results are reported in Table 3.

Figure 4. The scheme of SR model training and testing with different image degradation modeling methods.

Table 3. SR performance of different SR models on RealSR-Test with a scale factor of $4\times$.

SR Model	PSNR (dB) $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Bicubic-SR	25.99	0.735	0.443
K-ZSSR	23.07	0.645	0.352
DSGAN-SR	24.29	0.644	0.353
baselineCNN-SR	23.96	0.650	0.437
IDFlow-SR	26.90	0.783	0.282

As shown in Table 3, K-ZSSR and DSGAN-SR achieved better results in terms of LPIPS than the baseline SR model Bicubic-SR, but obtained an obviously lower performance in terms of PSNR and SSIM. Even though baselineCNN learned a direct mapping from HR to LR, the corresponding SR model baselineCNN-SR could not achieve SR performance gains as expected. On the contrary, IDFlow-SR achieved the best SR performance with a large margin, obtaining 2.94 dB/0.133/0.155 and 0.91 dB/0.048/0.161 better than baselineCNN-SR and Bicubic-SR in terms of PSNR/SSIM/LPIPS. IDFlow models the image degradation by learning the HR-LR conditional distribution, so it can generate more realistic HR-LR image pairs that help to promote SR performance on real-world images.

Visual comparisons of these SR models trained with different image synthesis approaches are shown in Figures 5 and 6. The results generated by Bicubic-SR are blurry, with no difference in visual perception when compared with the input. Though DSGAN and K-ZSSR help to generate sharper results, there are obvious ringing artifacts around image edges in the SR results. In flat regions, these methods tend to amplify the noises, resulting in tiny irregular textures, which can be observed in Figure 6. Although baselineCNN supervisedly learns the degradation process, the model trained with baselineCNN produces the SR results with similar patterns as DSGAN and K-ZSSR. Both of these three methods utilized a CNN-based network. Once the parameters of convolution layers are fixed, the CNN only obtains a certain fixed degradation mode, without maintaining the stochastic nature of image degradation. Thus, they could not help the SR model to obtain convincing SR images, which also means that the image degradation process cannot be effectively modeled only with a single CNN model. In contrast, IDFlow-SR generates SR output that has clearer image patterns and better perception quality with no visible artifacts. By learning the HR-LR conditional distribution is more efficient than a deterministic mapping, demonstrating the superiority of IDFlow over the compared methods above.

Figure 5. Visual comparisons of SR models on image “Nikon007” with a scale factor of $4\times$. Local regions of the results are cropped and zoomed for a better visual comparison.

Figure 6. Visual comparisons of SR models on images “Nikon040” and “Nikon017” with a scale factor of $4\times$. Local regions of the results are cropped and zoomed for a better visual comparison.

4.3. Evaluation on Real-World Super Resolution

The main challenge of super-resolution is a real-world application. Thus, we evaluated the effectiveness of the proposed IDFlow on DPED-Test in which all the images are blurry or noisy taken from different real scenarios, much more challenging for SR. We used standard IDFlow to generate realistic noisy LR training images based on the DF2K dataset. Some examples are shown in Figure 7. We then used the synthetic training dataset to re-train ESRGAN [4] for comparison for a better perceptual reconstruction. The trained model was named as IDFlow-GAN. The state-of-the-art SR methods being compared include K-ZSSR [11], DSGAN [9], SRFlow [20], the standard RRDBNet [4], and ESRGAN [4]. We also trained an ESRGAN with a RealSR-Train dataset added with synthetic noises generated by IDFlow for comparison, denoted as noisyRealSR.

(a) Examples of the DPED dataset.

(b) Examples generated by IDFlow.

Figure 7. Examples of realistic noisy image patches generated by IDFlow. The images in (a) and (b) have similarly noisy visual patterns.

Since there is no ground truth of testing LR images, we first conducted an evaluation on the SR images generated by different SR methods with no-reference IQA metrics. The results are reported in Table 4. We then made a visual comparison by presenting the local image regions of different SR results. As shown in Figure 8, SRFlow, ESRGAN, and RRDBNet cannot eliminate noises inside the images since they are all trained on clean Bicubic datasets, resulting in visible artifacts and abnormal textures. In contrast, the SR results of IDFlow-GAN have richer textures and clearer patterns, and there are fewer artifacts and less noise than the results of DSGAN as well as K-ZSSR. noisyRealSR is able to restore the sharper image edges and maintain cleaner patterns than DSGAN. This demonstrates that IDFlow realizes effective real-world noise modeling, so that noisyRealSR can eliminate the visible noises in testing images. However, it fails to generate clear tiny textures such as the tree branches shown in the first row of Figure 8. The main reason is that the number and diversity of the RealSR dataset is limited, causing a lack of enough high-frequency information to improve the SR performance on image details. This also proves from the opposite side that IDFlow alleviates the current deficiencies in existing real-world SR datasets. A more robust SR model can be trained. Therefore, IDFlow-GAN achieves better SR performance than noisyRealSR on real-world images.

Table 4. Comparison with different state-of-the-art SR methods on the DPED-Test dataset with a scale of $4\times$.

SR Method	NIQE ↓	BRISQUE ↓	PIQE ↓
RRDBNet	7.01	55.99	77.54
SRFlow	4.20	21.76	17.71
ESRGAN	4.02	19.63	16.09
K-ZSSR	8.15	58.73	80.67
DSGAN	3.5	11.34	12.06
noisyRealSR	5.39	32.09	42.85
IDFlow-GAN	3.76	7.95	14.69

Both qualitative and quantitative experiments indicate that IDFlow can accurately model image degradation and generate realistic noisy images. Benefiting from IDFlow, the performance of the existing SR methods in real scenarios can be significantly promoted.

Figure 8. Visual results of different SR methods on DPED-Test real-world images with a scale of $4\times$. Local regions of the results are cropped and zoomed for a better visual comparison.

5. Conclusions

In this study, we investigated image degradation modeling to promote the performance of SR models in real-world scenarios. We took image degradation modeling as the problem of learning a conditional probability distribution of degraded LR images given the HR input. The conditional-normalizing-flow-based method IDFlow was proposed to learn the distribution from existing real-world SR datasets. Given one HR image, IDFlow can generate a series of realistic LR images by transforming latent variables sampling from a simple distribution. Large-scale synthetic HR-LR pairs can be generated, alleviating the problem of an insufficient number and a diversity of existing real-world SR datasets. Quantitative and qualitative experiments show that not only can IDFlow produce realistic degraded LR images close to real-world images, but it also improves the generalization ability of SR models. SR models trained with IDFlow can produce richer image textures and clearer patterns in SR results without visible noises, resulting in a better SR performance than state-of-the-art SR methods on real-world noisy images. Therefore, the superiority and effectiveness of the proposed IDFlow is well demonstrated.

Author Contributions: W.X. proposed the research idea of this paper and was responsible for writing, conceptualization, and methodology. Q.Z. conducted literature research and data curation, and verified the experimental results. F.L. was responsible for the visualization of experimental data and chart making. The paper was mainly written by W.X., with the participation of Q.Z. and F.L. Supervision and review of the manuscript as well as funding acquisition were completed by R.C. 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, grant number 51675265, the Advantage Discipline Construction Project Funding of University in Jiangsu Province grant number PAPD, and the Independent Research Funding of the State Key Laboratory of Mechanics and Control of Mechanical Structures, grant number 0515K01.

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

References

1. 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. [\[CrossRef\]](#)
2. Kim, J.; Lee, J.K.; Lee, K.M. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
3. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
4. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Change, Loy, C. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2018; pp. 63–79.
5. 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.
6. Zhang, K.; Zuo, W.; Zhang, L. Learning a single convolutional super-resolution network for multiple degradations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3262–3271.
7. Chen, C.; Xiong, Z.; Tian, X.; Zha, Z.J.; Wu, F. Camera lens super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
8. Cai, J.; Zeng, H.; Yong, H.; Cao, Z.; Zhang, L. Toward real-world single image super-resolution: A new benchmark and a new model. In Proceedings of the IEEE International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 3086–3095.
9. Fritsche, M.; Gu, S.; Timofte, R. Frequency separation for real-world super-resolution. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea, 27–28 October 2019; pp. 3599–3608.
10. Ji, X.; Cao, Y.; Tai, Y.; Wang, C.; Li, J.; Huang, F. Real-world super-resolution via kernel estimation and noise injection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Seattle, WA, USA, 13–19 June 2020.
11. Bell-Kligler, S.; Shocher, A.; Irani, M. Blind super-resolution kernel estimation using an internal-gan. In *Advances in Neural Information Processing Systems*; 2019; pp. 284–293.
12. Bulat, A.; Yang, J.; Tzimiropoulos, G. To learn image super-resolution, use a gan to learn how to do image degradation first. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 185–200.

13. Lim, B.; Son, S.; Kim, H.; Nah, S.; Mu Lee, K. Enhanced Deep Residual Networks for Single Image Super-Resolution. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Computer Society, Honolulu, HI, USA, 21–26 July 2017; pp. 1132–1140.
14. Wang, X.; Yu, K.; Dong, C.; Loy, C.C. Recovering realistic texture in image super-resolution by deep spatial feature transform. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
15. Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv* **2016**, arXiv:1609.04802.
16. Xu, Y.S.; Tseng, S.Y.R.; Tseng, Y.; Kuo, H.K.; Tsai, Y.M. Unified dynamic convolutional network for super-resolution with variational degradations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 12496–12505.
17. Shocher, A.; Cohen, N.; Irani, M. Zero-Shot Super-Resolution Using Deep Internal Learning. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
18. Dinh, L.; Krueger, D.; Bengio, Y. Nice: Non-linear independent components estimation. *arXiv* **2014**, arXiv:1410.8516.
19. Dinh, L.; Sohl-Dickstein, J.; Bengio, S. Density estimation using real NVP. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
20. Lugmayr, A.; Danelljan, M.; Van Gool, L.; Timofte, R. Srfflow: Learning the super-resolution space with normalizing flow. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020.
21. Kingma, D.P.; Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, Montreal, QC, Canada, 3–8 December 2018; pp. 10236–10245.
22. Ignatov, A.; Kobyshev, N.; Timofte, R.; Vanhoey, K.; Van Gool, L. DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; Volume 2017.
23. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
24. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
25. Mittal, A.; Soundararajan, R.; Bovik, A.C. Making a ‘Completely Blind’ Image Quality Analyze. *IEEE Signal Process. Lett.* **2013**, *20*, 209–212. [[CrossRef](#)]
26. Mittal, A.; Moorthy, A.K.; Bovik, A.C. Referenceless image spatial quality evaluation engine. In Proceedings of the 45th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 6–9 November 2011; Volume 38, pp. 53–54.
27. Venkatanath, N.; Praneeth, D.; Bh, M.C.; Channappayya, S.S.; Medasani, S.S. Blind image quality evaluation using perception based features. In Proceedings of the 2015 Twenty First National Conference on Communications (NCC), Mumbai, India, 27 February–1 March 2015; pp. 1–6.
28. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
29. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in PyTorch. In Proceedings of the Adv. Neural Information Processing Systems Workshop Autodiff, Long Beach, CA, USA, 9 December 2015; pp. 1–4.