



## Article

# Deep Distillation Recursive Network for Remote Sensing Imagery Super-Resolution

Kui Jiang <sup>1</sup>, Zhongyuan Wang <sup>1,\*</sup>, Peng Yi <sup>1</sup>, Junjun Jiang <sup>2</sup>, Jing Xiao <sup>1</sup> and Yuan Yao <sup>3</sup>

<sup>1</sup> National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, Wuhan 430072, China; 2017282110506@whu.edu.cn (K.J.); 2017202110008@whu.edu.cn (P.Y.); jing@whu.edu.cn (J.X.)

<sup>2</sup> School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China; junjun0595@163.com

<sup>3</sup> State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; whyaoyuan@163.com

\* Correspondence: wzy\_hope@163.com; Tel.: +86-136-2865-2051

Received: 22 September 2018; Accepted: 24 October 2018; Published: 29 October 2018



**Abstract:** Deep convolutional neural networks (CNNs) have been widely used and achieved state-of-the-art performance in many image or video processing and analysis tasks. In particular, for image super-resolution (SR) processing, previous CNN-based methods have led to significant improvements, when compared with shallow learning-based methods. However, previous CNN-based algorithms with simple direct or skip connections are of poor performance when applied to remote sensing satellite images SR. In this study, a simple but effective CNN framework, namely deep distillation recursive network (DDRN), is presented for video satellite image SR. DDRN includes a group of ultra-dense residual blocks (UDB), a multi-scale purification unit (MSPU), and a reconstruction module. In particular, through the addition of rich interactive links in and between multiple-path units in each UDB, features extracted from multiple parallel convolution layers can be shared effectively. Compared with classical dense-connection-based models, DDRN possesses the following main properties. (1) DDRN contains more linking nodes with the same convolution layers. (2) A distillation and compensation mechanism, which performs feature distillation and compensation in different stages of the network, is also constructed. In particular, the high-frequency components lost during information propagation can be compensated in MSPU. (3) The final SR image can benefit from the feature maps extracted from UDB and the compensated components obtained from MSPU. Experiments on Kaggle Open Source Dataset and Jilin-1 video satellite images illustrate that DDRN outperforms the conventional CNN-based baselines and some state-of-the-art feature extraction approaches.

**Keywords:** remote sensing imagery; super-resolution; ultra-dense connection; feature distillation; video satellite; compensation unit

## 1. Introduction

In recent years, remote sensing imaging technology is developing rapidly and provides extensive applications, such as object matching and detection [1–4], land cover classification [5,6], assessment of urban economic levels, resource exploration [7], etc. [8,9]. In these applications, high-quality or high-resolution (HR) imageries are usually desired for remote sensing image analysis and processing procedure. The most technologically advanced satellites are able to discern spatial within a squared meter on the Earth surface. However, due to the high cost of launch and maintenance, the spatial resolution of these satellite imageries in ordinary civilian applications is often low-resolution (LR). Therefore, it is very useful to construct HR remote sensing images from existing LR observed images [10].

Compared with the general images, the quality of satellite imageries can be subject to additional factors, such as ultra-distanced imaging, atmospheric disturbance, as well as relative motion. All these factors can impair the spatial resolution or clarity of the satellite images, but video satellite imageries are more severely affected due to the over-compression. More specifically, for the video satellite, since it captures continuous dynamic video, in order to improve the temporal resolution, the optical imaging system has to sacrifice spatial resolution. At present, the original data volume of the video satellite has reached to the  $Gb/s$  level, but the channel transmission capacity of the spaceborne communication system is only in  $Mb/s$  level. To adapt to the transmission capacity of the satellite channel, the video acquisition system has to increase the compression ratio or reduce the spatial sampling resolution. For example, taking the video imagery taken by “Jilin No. 1” launched in China in 2015 as an example, although its frame rate reaches 25 fps, the resolution is only in  $2048 \times 960$  pixels (equivalent to 1080P), and hence the imagery looks very blurred. Therefore, the loss of high-frequency details caused by excessive compression is a special concern for video satellite imagery SR.

To address the above mentioned problems, a series of SR techniques for the restoration of HR remote sensing images have been proposed [10–14]. For example, Merino et al. proposed the super-resolution with variable-pixel linear reconstruction algorithm, named SRVPLR [15], which recombines a set of LR images in a linear nonuniform optimum manner. In [16], a hidden Markov tree model is proposed to establish a prior model in the wavelet domain to regularize the ill-conditioned problem for remote sensing image SR restoration. To fully use prior knowledge from a given LR image, Gou et al. [17] presented a non-local pairwise dictionary learning (NPDL) based model. In this model, the photometric, geometric, and feature information of the given LR image can be considered to improve the quality of reconstruction.

However, these shallow learning-based frameworks, show poor reconstruction performance when a high object resolution is required in practical applications. Recently, given the strength of deep CNNs, many CNN-based methods have evolved to deal with complex tasks in various applications [18–20], such as medical imaging, satellite imaging and video surveillance [21,22]. In particular, these effective architectures have achieved very good performance in general image SR reconstruction. For example, Dong et al. [23] introduced a three-layer CNN into single image SR (SISR) and achieved considerable improvement. Then, Kim et al. [24] proposed a residual network, called VDSR by using adaptive gradient clipping and skip connection to alleviate training difficulty. More recently, Sheng et al. [25] proposed the deep laplacian pyramid super-resolution network (LapSRN) to reconstruct the sub-band residuals of HR images at multiple pyramid levels. In LapSRN, a weight-sharing mechanism is implemented in the same structure, thus considerably reducing large quantity of parameters. However, the incremental depth in a deep CNN framework causes loss of information, thus weakening the continuity of information propagation. Moreover, these conventional CNN-based or residual-learning-based structures fail to restore fine texture details with simply direct or skip connections under complex imaging conditions. In particular, remote sensing satellite imageries have a complicated degradation process, low ground object resolution, and weak textures, thus posing considerable challenges for SR reconstruction.

Recently, Huang et al. [26] introduced the dense convolutional network (DenseNet) to strengthen feature propagation and encourage feature reuse by connecting each layer to every other layer in a feed-forward manner. Furthermore, in [27], the feature maps of each layer are propagated into all subsequent layers, thus providing an effective method of combining the low- and high-level features to boost reconstruction performance. Tai et al. [28] proposed memory blocks to build MemNet by heavily using long-term dense connections in MemNet to recover more high-frequency information. Although these methods can enforce information propagation by increasing nodes between layers with skip or dense connections, the features are fused in the network with a concatenated manner and will lead to large computational burden and high memory consumption.

Following the idea of sharing weights among recursive nodes, recursive learning networks have been recently used to reduce redundancy parameters of the network. For example, Kim et al. [29]

presented to use more layers to increase the receptive field of the network. It proposes a very deep recursive layer to avoid excessive parameters. In addition, a skip-connection manner is used to mitigate the training difficulty. Tai et al. [30] proposed a deep recursive residual network to address the problems of model parameters and accuracy, which recursively learns the residual unit in a multi-path model. More recently, Yang et al. [31] used the LR image and its edge map to infer sharp edge details of an HR image during the recurrent recovery process. However, the simple-connection manner used in these models [29,30] extremely limits the SR reconstruction performance.

In this study, a novel ultra-dense-connection manner is proposed to improve the reconstruction performance along with recursive strategy to mitigate memory consumption. Compared with the conventional skip- and dense-connection-based networks [24,26], the proposed UDB contains approximately twice as many short and long paths as the conventional dense block given the same convolution layers. Therefore, this will greatly enhance the representational power of the network. In addition, parameters sharing strategy between UDBs can extremely release the memory burden. We also find feature distillation in different stages leads to better accuracy for deep SR networks. Thus, we distill the feature maps by partly choosing output (with a special ratio) in different stages yet retain its integrity. After getting feature maps in different UDBs, we aggregate these components for gaining more abundant and efficient information in a multi-scale purification unit.

The strategy of feature distillation and compensation is obviously different from the knowledge distillation in these studies [32,33]. They compacted deep networks by letting a small simple network learn from a large complex network. In [34], the authors distilled a multi-model complex network by retaining the necessary network knowledge while keeping close performance. In [35], Pinteau et al. showed substantially reduced parameters by recasting multiple residual layers in the large network into a single recurrent simple layer. However, our proposed distillation and compensation strategy is mainly used to compensate for the high-frequency details lost during information propagation rather than model compression.

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

1. We propose a novel deep distillation recursive network DDRN for remote sensing satellite image SR reconstruction in a convenient and effective end-to-end training manner.
2. We propose a novel multiple-path residual block UDB, which provides additional possibilities for feature extraction through ultra-dense connections, quite agreeing with the uneven complexity of image content.
3. We construct a distillation and compensation mechanism to compensate for the high-frequency details lost during information propagation through the network with a special distillation ratio.

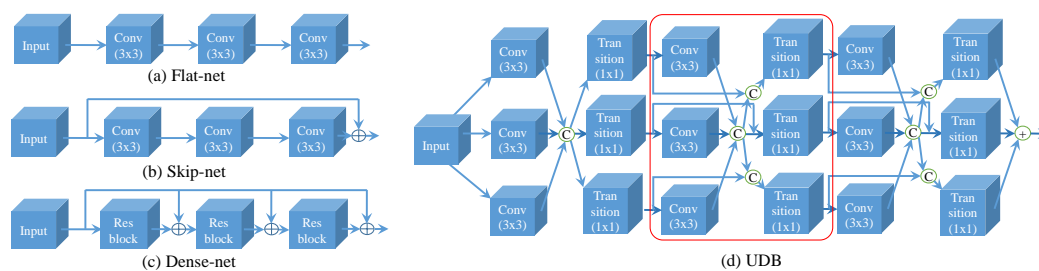
The remainder of this paper is organized as follows. In Section 2, we introduce previous works on CNN-based SR reconstruction algorithms, particularly network structures for feature extraction. Section 3 particularly presents the framework of the proposed DDRN. Section 4 individually presents the design of each key module under the proposed DDRN framework in details, including UDB, MSPU, resolution lifting, and loss function. Experimental results are given in Section 5, and the conclusions of this study are given in Section 6.

## 2. Related Work

We briefly review previously related works on structure-efficient networks [25,29,36–38], from which our network draws inspiration. These previous deep networks are committed to learning fine detail textures by designing a sophisticated structure. In this section, we focus on recent skip- and dense-connection-based methods.

**Skip connection:** A skip connection that directly connects input to output through an identity map, as shown in Figure 1b, was pioneered for SISR by Kim et al. [24]. They proposed a 20-layer CNN model known as VDSR. Instead of learning the actual pixel values, VDSR harnesses the global residual learning paradigm to predict the differences between ground truth and bicubic interpolated image.

This learning strategy makes the feature maps very sparse, enabling easy training and convergence. Compared with the traditional methods [39–42], this learning strategy on the benchmark datasets shows a significant superiority on reconstruction performance in terms of visual and quantitative indicators. In addition, DRCN [29] constructs a recursive-supervision structure to alleviate the difficulty in training a deep residual network further. Recently, Sheng et al. [25] proposed a deep Laplacian pyramid super-resolution network (LapSRN) to reconstruct the sub-band residuals of HR images at multiple pyramid levels with skip connection.



**Figure 1.** Frameworks of the CNN-based modules. (a) Flat-net (e.g., SRCNN [23] and FSRCNN [43]): Direct connections are commonly used to learn the features. (b) Skip-net (e.g., VDSR [24]): An identity map with connecting input to the output is pioneered for SISR. (c) Dense-net (e.g., DenseNet [26] and SRDenseNet [27]): The feature maps are directly passed from the preceding layers to the current layers through the identity function with much richer connections. (d) UDB: Interacted multiple-path units are embedded for extracting local feature maps with a richer ultra-dense connections. “C” and “+” denote the concatenation and adding operation, respectively.

**Dense connection:** Enlightened by previous works, Huang et al. [26] recently represented an intensive skip connection called dense connection. As shown in Figure 1c, the feature maps of the current layer are connected to every subsequent layer in a feed-forward manner. With rich local dense connections, the current layer can aggregate the information from all of the preceding layers within the dense block for further selection and fusion. These strategies effectively address the vanishing-gradient problem and enhance information propagation, thus strengthening the feature expression and boosting the convergence. Subsequently, Tong et al. [27] proposed an enhancement version called SRDenseNet. In SRDenseNet, the feature maps obtained from each dense block are propagated into the deconvolution layers to reconstruct SR images, providing an effective way to combine the low-level and high-level features, which further boosts the reconstruction performance. In addition, the dense skip connections in the network enable short paths to be built directly linking to the output from each layer, thus mitigating the vanishing-gradient problem. While considering the research on feature extraction and fusion, the earlier work of Gao et al. [38] is also noteworthy. They proposed a technique called multi-scale dense network for resource-efficient image classification. Their main idea is to train multiple classifiers in different stages using a two-dimensional multi-scale architecture, enabling them to preserve the coarse-and-fine level features all throughout the network.

**Ultra-dense connection:** These above mentioned strategies have been proven effective in addressing vanishing-gradient problem, guaranteeing accurate feature extraction and fusion. However, the directly concatenated operation on all layers in previous works [27,38] have led to high memory consumption and computation burden. In addition, conventional dense-connection-based networks have to construct a deeper network the more the skip paths required. Moreover, the increasing computational burden and memory consumption are unacceptable.

As shown in Figure 1d, on the basis of the dense network [26], we propose a multiple-path residual block called UDB. Compared with conventional skip or dense networks [24,26,27,29], UDB contains richer short and long paths with the same convolution layers. In particular, given the multiple-path units and transition layer, the feature channels becomes shallower, extremely reducing the parameters and decreasing the computational burden and memory consumption.

### 3. Network Architecture

As shown in Figure 2, the proposed model is a deep recursive neural network that can be roughly partitioned into three substructures, namely, local feature extraction and fusion, feature distillation, and feature compensation and SR reconstruction. Except for the upsampling operation, motivated by previous works on SISR [24,25,27,43], the entire process of local feature extraction and fusion is in the LR space.  $I_{LR}$  and  $I_{SR}$  are considered the LR input and HR output of the proposed DDRN, respectively.  $F_i$  and  $B_j$  refer to the output in the  $i_{th}$  layer and the  $j_{th}$  block, respectively. In this work, the LR RGB images are directly fed into the network and processed with the initial convolutional layers (two layers with  $3 \times 3$  kernel) to extract features as follows:

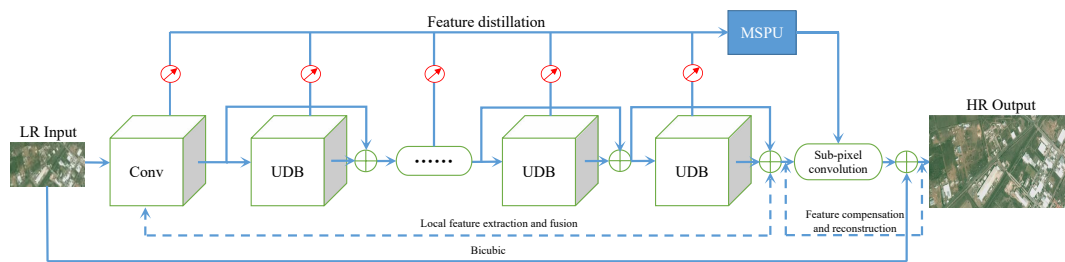
$$F_1 = H(I_{LR}), \quad (1)$$

$$F_2 = H(F_1), \quad (2)$$

where  $H(\cdot)$  denotes the convolution operation.  $F_1$  and  $F_2$  represent the shallow feature maps extracted through the initial convolutional layers, served as the input of the UDB. Moreover, the proposed residual block UDB is used as a basic module for local feature extraction in DDRN. For each UDB, the information cannot only be shared among layers and multiple-path units but also be used as the input for the subsequent residual blocks with ultra-dense connections. These strategies enforce information propagation and lead to fine feature expression by combining the multi-scale coarse-and-fine features in different stages. The operation can be defined as follows:

$$B_i = H_{block,i}(B_{i-1}) + B_{i-1}, \quad (3)$$

where  $H_{block,i}$  denotes the entire convolution operation in the  $i_{th}$  UDB and  $B_{i-1}$  refers to the extracted feature maps from the  $(i-1)_{th}$  UDB. As shown in Figure 1, compared with the conventional CNN-based modules [24–26,29,30], whose commonly used residual block contains the simply direct or skip connections between layers, the proposed UDB module is composed of several interactive multiple-path units and parametric rectified linear units (PReLU). The dedicated architecture for UDB enjoys more linking paths in the same layers and provides more possibilities for feature extraction than do these previous strategies, thus matching the uneven content complexity of remote sensing imagery. Specifically, the simple links are adapted to smooth areas, whereas complex connections are suited for high-frequency texture details.

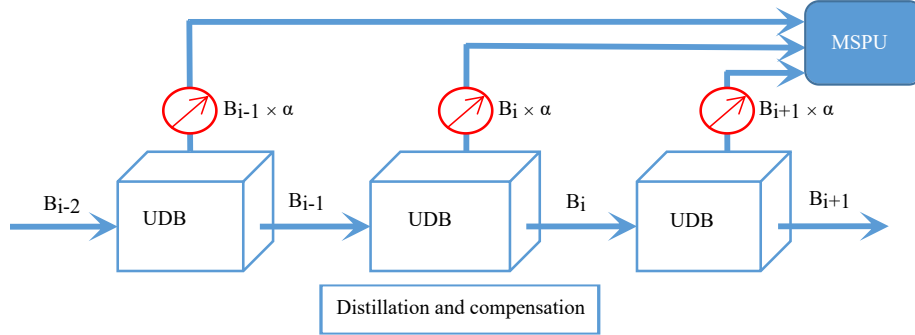


**Figure 2.** Outline of the proposed deep distillation recursive network (DDRN). The red distillation symbol followed the UDB represents the distillation operation with a special distilled ratio of  $\alpha$ .

According to previous SISR algorithms [24,27,29,30], the output of the current stage is directly transmitted to the next stage. Then the final residual maps are obtained at the top layer for SR reconstruction. However, information loss is inevitable during its propagation in the network, thereby weakening the continuity of information propagation. Previous works add a set of nodes to shorten the transmission distance, thus boosting information propagation and reducing information loss during propagation, so-called skip connections [24,29]. However, increasing the nodes between the input and the output cannot only deepen the network but also increase computational burden



and memory consumption. Differently, we facilitate information propagation with the multiple-path residual module UDB. Furthermore, we also present a distillation and compensation strategy for fine feature expression by compensating for extra-high-frequency details. As shown in Figure 3, unlike the traditional network, whose output in each block is directly transmitted to the subsequent part, our proposed method can adaptively distill and preserve the feature maps by partly choosing information from the current output yet retain its integrity. Then, these feature maps collected from different stages are aggregated and purified in MSPU to infer and compensate for the high-frequency details before the reconstruction operation.



**Figure 3.** The distillation and compensation mechanism. The red components indicate that the distilled feature maps  $B_i \times \alpha$  in current UDB are adaptively preserved.  $\alpha$  denotes the distillation ratio for current UDB output  $B_i$ . MSPU refers to the further purification operation.

In this study, we denote the preserved part from  $B_i$  as the distillation unit (DU) with the ratio of  $\alpha$ . At the same time,  $B_i$  is used as the input to the subsequent residual block for further extraction. This process can be formulated as follows:

$$DU_i = S(B_i, \alpha), \quad (4)$$

where  $\alpha$  refers to the distillation ratio, which indicates that the feature maps in each stage with the ratio of  $\alpha$  will be distilled and preserved. In our experiments, we set  $\alpha$  to  $\{0.0, 0.125, 0.25, 0.5\}$ .  $S(\cdot)$  represents the distillation operation, and  $DU_i$  denotes the distilled information from the  $i_{th}$  residual block  $B_i$ .

In addition, the reserved feature maps  $DU_i$  in different stages are aggregated through a concatenation operation, and then they are fed into the purified unit MSPU, where the HR components lost in the previous blocks are reactivated as a compensation for SR reconstruction. In Equation (5),  $H_C(\cdot)$  denotes the concatenation operation adopted to collect the distillation information and  $M(\cdot)$  refers to the MSPU. Through the distillation and compensation mechanism, the high-frequency components compensated from MSPU can further promote reconstruction performance.

$$P = M(H_C(DU_0, \dots, DU_i, \dots, DU_n)), \quad (5)$$

At the end of the network, the feature maps extracted from the top UDB and the compensated high-frequency details purified from MSPU are combined to infer and restore the HR components by a transition layer with  $3 \times 3$  kernel. Then, a sub-pixel upsampling operation is used to project these features into HR space to obtain the residual image. The detailed operation is expressed as follows:

$$I_{SR} = PS(H_S(D_n, P)) + I_B, \quad (6)$$

where  $D_n$  and  $P$  represent the feature maps extracted from the top UDB and the compensated details from MSPU, respectively.  $H_S$  denotes a transition function that contains a  $3 \times 3$  convolution layer to fuse features and infer HR components, adaptively.  $I_B$  refers to the bicubic interpolated image.

$PS(\cdot)$  represents the reconstruction operation performing a sub-pixel amplification to obtain the HR residual image in the ending part of the network.

#### 4. Feature Extraction and Distillation

In this section, we present the design of each key module under our DDRN framework in details, including UDB, MSPU, and Resolution Lifting.

##### 4.1. Ultra-Dense Residual Block (UDB)

It is acknowledged that rich dense connections can promote feature expression [26,27]. Therefore, we design a dense connection module for feature extraction. In this study, a multiple-path residual block UDB is constructed to enforce the correlation among layers and blocks with rich dense connections. Compared with existing skip- or dense-connection-based methods, UDB considers diverse short and long linking paths (the multiple-path structure) and exhibits effective information-sharing capability among the layers. Therefore, our network provides additional possibilities for feature extraction, quite agreeing with the uneven complexity of image content. More precisely, simple links are adapted to smooth areas, whereas complex connections are suited for high-frequency texture details. As shown in Figure 1d, UDB includes several interactive multiple-path units, which can fuse the feature maps extracted from parallel multiple convolution paths. The information-sharing mechanism aggregates features in different levels to ensure a rich feature representation further. The function of the  $i_{th}$  unit can be formulated as follows:

$$y_i = H_C([F_{i,0}(x_0), F_{i,1}(x_1), \dots, F_{i,n}(x_n)]), \quad (7)$$

$$s_{i,n} = H_1(H_C(y_i, s_{i-1,n})). \quad (8)$$

Equations (7) and (8) formally show the operation process in a multiple-path unit. In Equation (7),  $F_{i,n}(x_n)$  and  $H_C([F_{i,0}(x_0), F_{i,1}(x_1), \dots, F_{i,n}(x_n)])$  refer to the single convolution operation and the feature congregation of multiple convolution layers in each unit, respectively. In Equation (8),  $y_i$  denotes feature concatenation in the current unit.  $s_{i,n}$  indicates the transition output in the  $n_{th}$  path of the  $i_{th}$  unit, and  $s_{i-1,n}$  represents the output from the  $n_{th}$  path of the  $(i-1)_{th}$  unit. Functionally, a group of skip connections is used to enforce the correlation among the input and output feature maps, where the transition layers represented as  $H_1$  are embedded to reduce feature channels with  $1 \times 1$  convolution kernel.

Unlike skip- or dense-connection-based algorithms [26–28], the proposed multiple-path ultra-dense connection block can simultaneously explore and infer local and global features. In particular, the feature maps in the multiple-path unit cannot only be shared among the layers in the current unit through aggregation and dense connections but also be used as the input of other units with skip connections. Given the simplicity, effectiveness, and robustness of this strategy, local features can be well expressed through numerous short and long paths. Furthermore, owing to the effective structure for feature extraction in UDB, the network can become shallow in the channels but wide for the convolution paths, which extremely reduces the parameters and simultaneously boosts the reconstruction performance.

##### 4.2. Multi-Scale Purification Unit (MSPU)

In [44], the authors focused on channels and proposed a novel architectural unit termed “squeeze-and-excitation” (SE) block to recalibrate channel-wise feature responses adaptively by explicitly modeling the interdependencies between channels. The SE block can learn to use global information to emphasise informative features and suppress less useful features selectively. This model won the first place in the classification contest ILSVRC2017 [45].

In this study, we adopt the SE module because of its promising efficiency and efficacy. On the basis of this finding, we propose an applicable module MSPU for information compensation. The basic structure of MSPU building unit is illustrated in Figure 4. Contrary to the squeeze-and-excitation

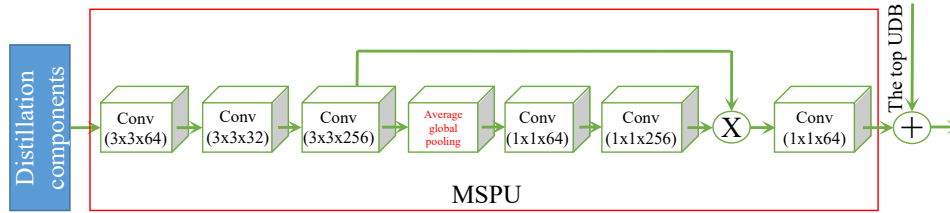
network (SEN) [44], the redundant residual connections between SE blocks used for features transmission are removed. In addition, given that the full connection layer can destroy the internal structure of the image, we therefore replace it with a  $1 \times 1$  convolution layer. Moreover, we adopt a robust activation function, e.g., parametric rectified linear unit (PReLU), to replace the previous version rectified linear unit (ReLU).

On the basis of MSPU process, we further propose a distillation and compensation strategy to compensate for lost details. By partially distilling the components from  $B_i$  with the distillation ratio of  $\alpha$ , as shown in Figure 3, we can obtain feature maps originating from UDB in different stages. Then, these features are aggregated into MSPU to purify and gain more abundant and efficient information. The extraction functions can be defined as follows:

$$MS = H(x), \quad (9)$$

$$P = \sigma(H_1(A_P(MS))) \times MS. \quad (10)$$

In Equation (9), the input  $x$  denotes the concatenation of the distilled components in different stages, equivalent to  $H_C(DU_0, \dots, DU_i, \dots, DU_n)$  in Equation (5), and  $H(\cdot)$  represents a group of convolutional operations (with  $3 \times 3$  kernel) that is adopted to fuse the features distilled from different levels. As expressed in Equation (10),  $A_P$  denotes the global average pooling,  $H_1$  refers to the group of transition layers that comprises the bottleneck structure, and  $\sigma$  represents the sigmoid function.



**Figure 4.** The Multi-scale feature purification unit (MSPU). The distillation components preserved from the different stages are used to obtain compensation information lost during the information delivery. X denotes the matrix multiplication.

#### 4.3. Resolution Lifting

To project a single LR image into HR space, the resolution of LR image must be increased to match that of the HR image at a certain point. Osendorfer et al. [46] presented a computationally efficient architecture for image SR by leveraging the fast approximate inference to increase the image resolution in the middle of the network gradually. Another well-known approach can also achieve spatial resolution enhancement by linear interpolation [23,24]. They obtained the same image resolution by directly using the common bicubic interpolation before loading the dataset into the network.

In addition, the early work of Shi et al. [47] is noteworthy when considering the upsampling operation. Contrary to authors of previous works, the researchers proposed an efficient sub-pixel convolution layer to increase the image resolution only at the final layer, eliminating the need to perform most of the SR operations in the large HR space. Compared with the transposed convolution and bicubic interpolation, sub-pixel magnification [47] is actually a realignment of feature maps without extra parameters, thus quite decreasing memory consumption and computational cost. These reasons enable the network go deeper and be trained easily.

As expressed in Equation (11),  $PS$  is a shuffling operator that rearranges the elements of a  $H \times W \times C \cdot r^2$  tensor acquired in the top layer into a  $rH \times rW \times C$  tensor (where  $r$  is the magnification factor of the network, and  $C$  refers to the feature channels of the input image). Mathematically, the upsampling function can be expressed as follows:

$$PS(T)_{x,y,c} = T_{\lfloor x/r \rfloor, \lfloor y/r \rfloor, c}(\text{mod}(x, r), \text{mod}(y, r)), \quad (11)$$



where  $T$  indicates the output from the final layer with the size of  $W \times H \times Cr^2$ ,  $(x, y)$  denotes the output pixel coordinate in the HR space,  $(x/r, y/r)$  represents the pixel area of  $r \times r$  in the sub-pixel space, and  $(\text{mod}(x, r), \text{mod}(y, r))$  refers to the pixel coordinate in LR space. The  $Cr^2$  channels of each pixel in the same location in the LR space is rearranged into a region of  $1r \times 1r \times C$ , which corresponds to a subblock in an HR image, and the feature image is rearranged into an HR image of  $rW \times rH \times C$ .

In this work, as in many CNN-based SISR methods [25,47,48], we adopt the sub-pixel upsampling strategy to reconstruct the HR image at the top layer because of its promising efficiency and efficacy.

#### 4.4. Loss Function

It is well known that SISR is an ill-posed problem whose solution from the reconstruction constraint is not unique because of the insufficient number of LR images, ill-conditioned registration, and unknown degradation process. In previous works, the loss function is commonly used to fit the real target image by minimizing the distance between the reconstructed HR image and the ground truth. The commonly used distance measurements include pixel-based  $l_1$ -norm [25] and  $l_2$ -norm [23,24,29], and cosine distance based on feature level.

Most of the previous works [23,27,29] constrain the reconstruction image by minimizing the mean squared error (MSE) or maximizing the peak signal to noise ratio (PSNR), which is a common measure used to evaluate SR algorithms [49]. However, the capability of MSE to capture perceptually relevant components, such as high-frequency texture details, is insufficient because they are defined on basis of pixel-wise image differences [50]. For example, the previous works [23,29,43] use MSE loss as the cost function and produce overly smooth reconstruction results that are inconsistent with human vision. In [25,51], the authors proposed a novel optimal function charbonnier loss based on the  $l_1$ -norm, which can recover a large amount of realistic details, more faithful to the ground truth. In our work, we therefore introduce the charbonnier penalty function to penalize the deviation of the prediction from the residuals of ground truth. The loss function can be expressed as follows:

$$\text{Loss}(I_{SR}, I_{HR}, \theta) = \arg \min_{\theta} \sum \rho(I_{HR} - f(I_{LR}, \theta)), \quad (12)$$

where  $\theta$  denotes a set of model parameters to be optimized and  $\rho(x) = \sqrt{x^2 + \varepsilon^2}$  represents the charbonnier penalty function (a differentiable variant of  $l_1$ -norm). We empirically set the compensation parameter  $\varepsilon$  of  $10^{-3}$ .  $I_{SR}$  and  $I_{HR}$  refer to the predicted HR image and the ground truth.

## 5. Experimental Results and Analysis

In this section, first, we describe the experimental settings, including the data collection and model parameters. Then, we assess the effect of the distillation ratio  $\alpha$  and the network depth  $m$  on the reconstruction performance. Subsequently, we compare our results with these state-of-the-art techniques and provide a thorough analysis. We retrain the comparison algorithms with our training dataset to ensure a fair comparison, including SRCNN [23] and VDSR [24]. Moreover, we directly apply the original models [23–25] trained with general image datasets, as the anchors.

### 5.1. Data Collection

For general image SR, a large quantity of public training and assessing datasets, such as DIV2K [52], BSD500 [53] and Yang291 [39], are used to evaluate the results. However, few available datasets can be used as the training samples for satellite imagery SR because of the special requirements of ground target resolution. We use two available satellite image datasets, namely, *Kaggle Open Source Dataset* and *Jilin-1* video satellite imagery, to train and evaluate the proposed DDRN method.

1. The first imagery dataset is the *Kaggle Open Source Dataset* (<https://www.kaggle.com/c/draper-satellite-image-chronology/data>), which contains more than 1000 HR images of aerial photographs captured in southern California. The photographs were taken from a plane and

meant as a reasonable facsimile for satellite images. The images are grouped into five sets, each of which having the same setId. Each scenario in a set contains five images captured on different days (not necessarily at the same time each day). The images for each set cover approximately the same area but are not exactly aligned. Images are named according to the convention (setId-day). In this dataset, the scene has  $3099 \times 2329$  pixels and 324 different scenarios. A total of 1720 satellite images cover agriculture, airplane, buildings, golf course, forest, freeway, parking lot, tennis court, storage tanks, and harbor. In this study, 30 different categories are selected for the test and 10 for the evaluation. Meanwhile, a total of 350 images are used for the training. Regarding the training dataset, the entire images are cropped into many batches with  $720 \times 720$  pixels, but only the central area of the testing images with size of  $720 \times 720$  pixels is cropped for testing and evaluation.

2. The second satellite dataset is from *Jilin-1* video satellite imagery. In 2015, the Changchun Institute of Optics, Fine Mechanics, and Physics successfully launched the *Jilin-1* video satellite which had 1.12 m resolution. To cover the duration of video sequences, we select one for every five frames from each video and crop the central part with the size of  $480 \times 204$  as test samples. We select several areas in different countries with certain typical surface coverage types, including vegetation, harbor, and a variety of buildings as the test images.

## 5.2. Model Parameters and Experiment Setup

In our experiments, we use an NVIDIA GTX1080Ti GPU and an Intel I7-8700K CPU for training and testing, respectively. Our model is implemented on TensorFlow with Python3 under Windows10, CUDA8.0, and CUDNN5.1 systems. We mainly focus on the up-scaling factor of 4, which is usually the most challenging case in image SR.

The original HR images are downsized by bicubic interpolation to generate LR images for training. We augment the training patches by horizontal or vertical flipping and rotating  $90^\circ$ . By following the settings presented in [54], we send one batch consisting of 16 LR RGB patches with the size of  $32 \times 32$  from the training datasets to our network each time. The learning rate is initialized to  $10^{-3}$  for all layers and halved for every  $10^4$  steps up to  $10^{-5}$ . In our model, each convolution layer contains 64 filters, followed by PReLU. We empirically set the distillation ratio  $\alpha$  to  $\{0.0, 0.125, 0.25, 0.5\}$  and the number of parallel convolution layers  $n$  in each multiple-path unit to 3. For the basic module DDRN, the depth of UDB is 15. In our experiments, training a basic module consumes approximately 20 h under the previously presented experimental settings.

## 5.3. Quantitative Indicators (QI)

Similar to many previous representative works [23,24,28,29], we also select two commonly used evaluation metrics, i.e., PSNR and structural similarity (SSIM), to evaluate the model performance. These evaluation metrics differ in terms of visual perception but involve reference images for comparison. However, in real SR scenes, we have only LR images to be super-resolved, without the corresponding HR reference image. Therefore, we need to introduce quantitative non-reference image quality assessment methods. Quality with no reference (QNR) [55,56], generalized quality with no reference (GQNR) [57] and average gradient (AG) [58] are commonly used image quality evaluation algorithms without reference, which can reasonably assess the clarity of reconstructed image. Nevertheless, QNR and GQNR are used for multispectral or hyperspectral images rather than ordinary RGB images, which needs to calculate the spectral distortion index and spatial distortion index. Thus, in this study, we propose to alternatively use AG for objective evaluation without reference. This process can be expressed as follows:

$$G(x, y) = dx_{(i,j)} + dy_{(i,j)}, \quad (13)$$

$$dx_{(i,j)} = I_{(i+1,j)} - I_{(i,j)}, \quad (14)$$

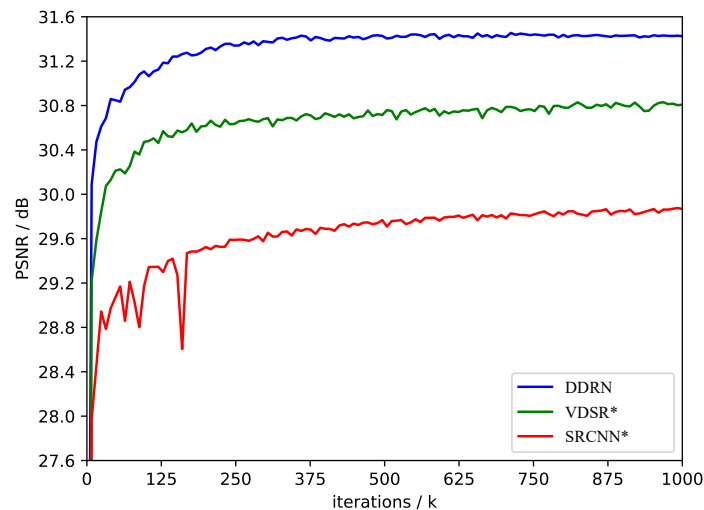
$$dy_{(i,j)} = I_{(i,j+1)} - I_{(i,j)}, \quad (15)$$

where  $dx$  and  $dy$  refer to the horizontal and vertical gradients, respectively, and  $I_{(i,j)}$  denotes the pixel value corresponding to the coordinate of  $(i, j)$ .

The indicator of the AG can reasonably assess image clarity because it sensitively reflects content sharpness, detail contrast, and texture diversity. Generally, the larger the AG, the richer the details. Thus, the AG can be used to evaluate the reconstruction quality of satellite imagery in real-world scenes, such as *Jilin-1* video satellite imageries.

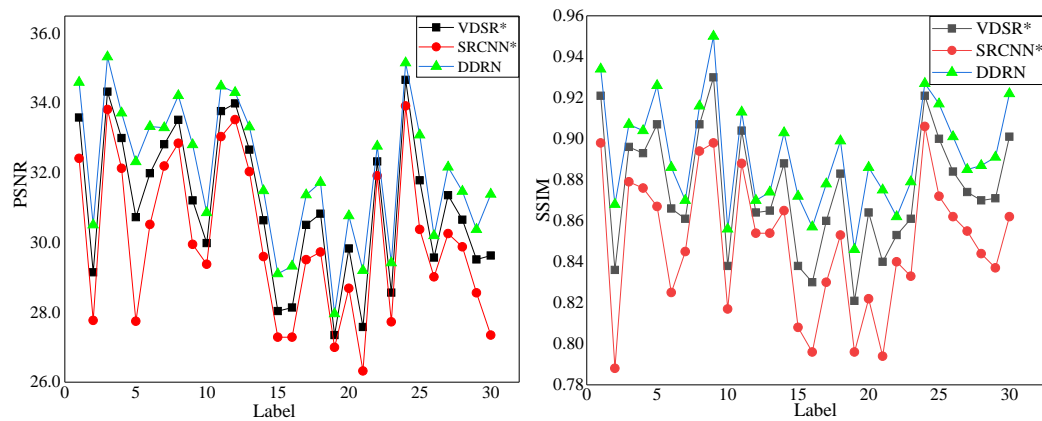
#### 5.4. Validation of the Ultra-Dense Residual Block

We examine the effectiveness of the proposed deep recursive CNN network DDRN and the multiple-path UDB. Given that SRCNN [23] and VDSR [24] are the most representative and most effective deep-learning-based SR methods, in our experiments, we retrain these two models by using the same training datasets and label them as SRCNN\* and VDSR\*. Figure 5 shows the comparison results according to the iterations of DDRN, SRCNN, and VDSR. Comparatively, our DDRN exhibits faster convergence and higher scores than do direct-connection-based SRCNN and skip-connection-based VDSR. This superiority can be mainly attributed to the proposed multiple-path ultra-dense connections which can readily capture local features. Thus, our framework significantly boosts the SR efficacy of remote sensing imagery.



**Figure 5.** Training process for different models with the scale of 4. On the top, the blue line denotes the convergence process of the basic module DDRN with depth of 15 while the green and the red lines at the bottom refer to the VDSR and SRCNN. The competitive algorithms marked by \* denote the retrained versions with our dataset.

In Figure 6, we show the evaluation results of the proposed DDRN method and the comparison algorithms on the *Kaggle Open Source Dataset* to verify the usefulness of the ultra-dense connections strategy further. The test set contains 30 different scenarios, which are labeled 1 to 30 in Figure 6. The figure shows that by using ultra-dense connections, we obtain better reconstruction results than do the conventional CNN-based methods, i.e., SRCNN [23] and VDSR [24]. For the average PSNR, our DDRN shows substantial improvements, surpassing VDSR by 0.92 dB, and SRCNN by 1.94 dB. Similarly, SSIM is also considerably improved.



**Figure 6.** The SR performance comparisons for 30 different scenarios (denoted by label) from *Kaggle Open Source Dataset*. The competitive algorithms marked by \* denote the retrained versions with our dataset.

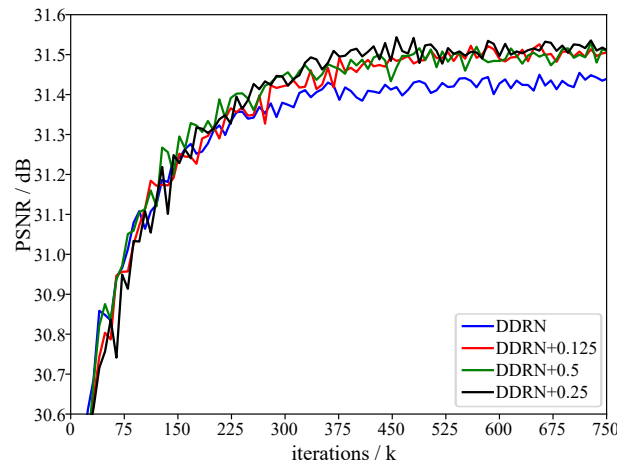
In summary, the proposed residual block UDB effectively captures realistic detail textures. Although SRCNN and VDSR are effective, the well-designed deep recursive framework DDRN is more suitable for satellite image SR reconstruction.

#### 5.5. Influence of Parameters $\alpha$ and $m$

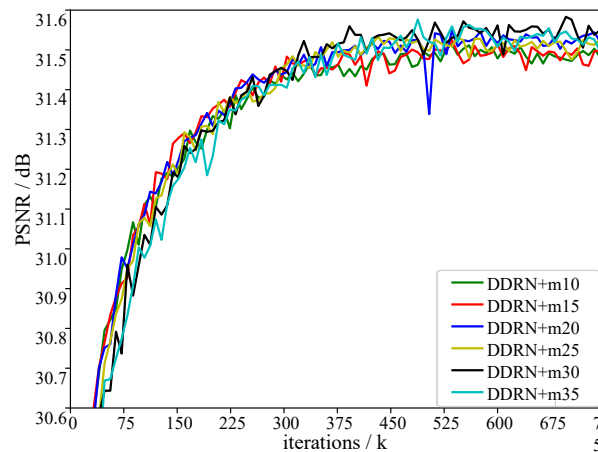
On the basis of the basic module DDRN, we implement a distillation and compensation mechanism to compensate for the HR components lost during information propagation to infer and restore more realistic high-frequency details. The improved model with MSPU embedment is called  $\text{DDRN}^+$ . In particular, a couple of comparison simulation experiments are conducted to analyze the influences of (i) the hyperparameter  $\alpha$  in Equation (4) for partial feature maps distillation and preservation, (ii) the depth value  $m$  of UDB on the reconstruction performance.

We report the training process of the proposed  $\text{DDRN}^+$  with respect to different distillation ratios to verify the necessity of the proposed distillation and compensation mechanism. When  $\alpha$  is set to 0, no components are distilled in the current stage, whereas MSPU does not function. Figure 7 shows the comparison results of the training process under different distillation ratios. From the figure, we learn that the proposed  $\text{DDRN}^+$  exhibits better training performance than the basic module DDRN. In addition, we observe that, with an increase in the distillation ratio  $\alpha$ , the module exhibits robust and fast convergence. This result can be attributed to the increasing compensated high-frequency details from the MSPU by an increased distillation ratio. However, we also observe that the performance starts to decline when  $\alpha$  is set to a large value, e.g., 0.5. This result can be mainly attributed to the large distillation rate, which may result in information redundancy. In addition, excessive parameters might lead to overfitting. All of these results indicate that the proposed distillation and compensation mechanism show substantial improvements by compensating for high-frequency details. Therefore, embedding MSPU into the basic module for satellite image SR reconstruction is an effective and reliable choice.

In light of the observations in these previous works [26–28], fine features can be well inferred from a deep CNN framework. Thus, we gradually increase the depth of the network by simply adding the number of the UDB (i.e.,  $m$  is set to 10, 15, 20, 25, 30, and 35). We assess the performance of different values of  $m$ . In Figure 8, we show the training details of the proposed  $\text{DDRN}^+$  method with different depths. When simply increasing the value of  $m$  to 30, the improvement gradually increases and surpasses the basic module by approximately 0.22 dB in the scale of 4. By contrast, the performance declines when we continue to increase  $m$  to 35 and the network exhibits slow convergence. This result can be mainly attributed to the overfitting, and the convergence of the network becomes more difficult in such a depth.



**Figure 7.** Training process for different distillation ratios by the scale of 4.  $\text{DDRN}^+$  represents the improved module with MSPU embedded at different ratios on the basis of the basic module. In particular, DDRN denotes the improved module with the distillation ratio  $\alpha$  of 0, which is actually the basic module.



**Figure 8.** Training process for different depths of  $\text{DDRN}^+$  with scale of 4 and the distillation ratio  $\alpha$  of 0.25. We set UDB number  $m$  to 10, 15, 20, 25, 30 and 35 while keeping other parameters consistent.

On the basis of the experiments, we can obtain the optimal distillation ratio  $\alpha$  and UDB depth  $m$  for satellite image SR reconstruction, which are set to 0.25 and 30, respectively.

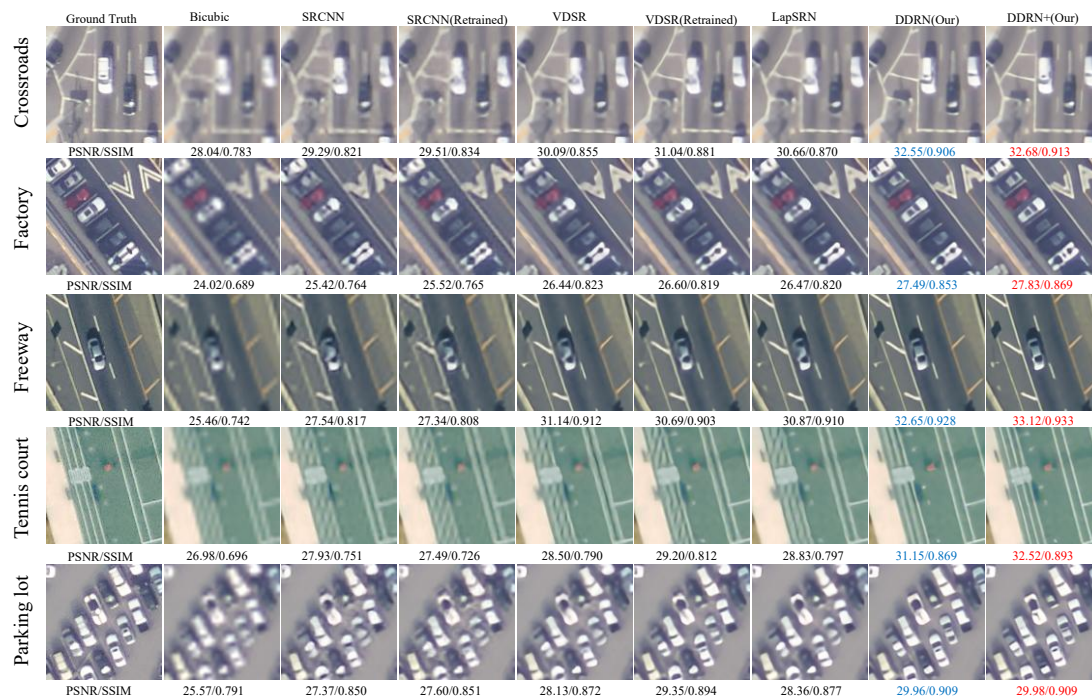
### 5.6. Comparison Results with the State-of-the-Art

We compare our basic model DDRN and the improved version  $\text{DDRN}^+$  ( $\alpha = 0.25$ ,  $m = 30$ ) with other SISR algorithms, including Bicubic, SRCNN [23], VDSR [24], and LapSRN [25], by the scaling factors of  $\times 2$ ,  $\times 3$ , and  $\times 4$ . The implementations of these anchor methods have been released online and can thus be conducted on the same test datasets.

The reconstruction results obtained with above mentioned *Kaggle Open Source Dataset* for the proposed approaches and the comparison methods are shown in Figure 9. We select several different but representative scenarios (i.e., crossroads, factory, freeway, tennis court, and parking lot) to produce a visual presentation. Experimentally, we crop these representative scenarios into a sub-batch with the size of  $120 \times 120$  pixels from each reconstructed SR image and compute PSNR and SSIM. Notably, the proposed method DDRN and its improved version  $\text{DDRN}^+$  surpass these state-of-the-art methods by a large margin. Moreover, the modules that we propose exhibit the most accurate and



realistic image details from the visual effect. Most of the comparison methods produce noticeable artifacts and blurred edges, whereas the proposed  $\text{DDR}^{\text{N}^+}$  can recover sharper and clearer edges because of successful feature extraction and fusion, more faithful to the ground truth. For example, as shown in Figure 9, only our proposed modules restore the clear court boundary in the tennis court scenario and the accurate and credible car outline in the four other scenarios. Therefore, all of the proposed models exhibit solid performance improvements compared with the conventional direct- or skip-connection-based algorithms [23–25].



**Figure 9.** The reconstruction results on *Kaggle Open Source Dataset* and by the scale of 4. We select several different but representative scenarios, i.e., crossroads, factory, freeway, tennis court and parking lot, and then crop them into small image batches in size of  $120 \times 120$  for demonstration. Red and blue indicate the best and the second best performance, respectively.

Objectively, Tables 1, 2 and 3 tabulate the detailed evaluating results in terms of PSNR, SSIM and AG with the magnification scales of  $\times 2$ ,  $\times 3$ , and  $\times 4$ , respectively. From these records, we learn that raw CNN-based or skip connection methods, such as SRCNN [23] and VDSR [24], exhibit lower scores than do  $\text{DDR}^{\text{N}}$ -based methods (i.e., in terms of PSNR, the proposed  $\text{DDR}^{\text{N}^+}$  surpasses SRCNN and VDSR retrained by approximately 2.16 and 1.14 dB with the scale of 4 in the first test dataset, respectively). Among these comparison methods, the basic module  $\text{DDR}^{\text{N}}$  shows the best performances because of its ultra-dense-connection-based effective framework for local spatial information extraction. In addition, through the compensated high-frequency details obtained from the MSPU, the improved version  $\text{DDR}^{\text{N}^+}$  can produce fine detail textures. With regard to PSNR and SSIM, Figure 6 shows an more intuitive result that the proposed modules outperform these state-of-the-art methods [23–25] by a large margin. For the metric AG, the proposed  $\text{DDR}^{\text{N}}$  and  $\text{DDR}^{\text{N}^+}$  are also better than previous works on average. In particular, in the comparison results shown in the three tables, our methods exhibit remarkable advantages when the upsampling factor is large, as reported at the bottom of the three tables. These results indicate the advantages of the proposed ultra-dense-connection manner in modeling the relationship between LR and HR images with larger magnification factors.

**Table 1.** Quantitative evaluation of the proposed DDRN approach and its improved version DDRN<sup>+</sup> against some state-of-the-art SISR algorithms on *Kaggle Open Source Dataset* with 30 different scenarios for the scale factor of  $\times 2$ . Bold indicates the best performance. Particularly, \* refers to the modules retrained by us with *Kaggle Open Source Dataset*.

Labels	Methods Scale	Bicubic PSNR/SSIM/AG	SRCNN [23] PSNR/SSIM/AG	SRCNN * PSNR/SSIM/AG	VDSR [24] PSNR/SSIM/AG	VDSR * PSNR/SSIM/AG	LapSRN [25] PSNR/SSIM/AG	DDRN (Our) PSNR/SSIM/AG	DDRN <sup>+</sup> (Our) PSNR/SSIM/AG
(1)	2	36.77/0.960/3.468	39.17/0.973/3.878	39.49/0.974/3.849	40.52/0.978/3.887	40.83/0.979/3.881	40.65/0.979/3.881	41.33/0.980/3.958	<b>41.38/0.981/3.958</b>
(2)	2	31.97/0.919/4.729	35.27/0.953/5.848	35.21/0.951/5.791	35.99/0.959/5.907	36.58/0.962/5.838	35.95/0.959/5.879	37.10/0.965/5.915	<b>37.23/0.965/5.920</b>
(3)	2	37.42/0.945/2.700	39.20/0.959/3.176	39.31/0.960/3.100	40.20/0.965/3.197	40.36/0.966/3.170	40.34/0.966/3.177	40.77/0.968/3.216	<b>40.82/0.968/3.220</b>
(4)	2	36.78/0.953/3.698	39.07/0.968/4.172	38.97/0.966/4.119	39.47/0.969/4.174	39.70/0.970/4.147	39.61/0.970/4.158	39.95/0.971/4.194	<b>40.05/0.971/4.201</b>
(5)	2	31.97/0.948/6.149	35.63/0.970/6.808	35.54/0.969/6.846	36.75/0.974/6.821	37.23/0.976/6.860	36.84/0.975/6.808	37.16/0.977/6.921	<b>37.66/0.978/6.903</b>
(6)	2	33.81/0.913/3.614	35.78/0.936/4.238	35.91/0.935/4.192	37.16/0.944/4.346	37.26/0.945/4.269	37.20/0.945/4.310	37.57/0.946/4.357	<b>37.73/0.947/4.373</b>
(7)	2	35.80/0.924/3.474	37.26/0.941/4.020	37.10/0.939/3.908	37.50/0.943/4.037	37.56/0.944/3.986	37.59/0.944/4.022	37.72/0.945/4.054	<b>37.79/0.945/4.061</b>
(8)	2	36.66/0.953/2.538	39.05/0.968/3.050	38.88/0.966/3.022	40.00/0.971/3.067	40.02/0.971/3.048	39.96/0.971/3.041	40.66/0.973/3.097	<b>40.74/0.973/3.104</b>
(9)	2	33.39/0.962/5.090	37.62/0.982/5.652	38.29/0.982/5.785	39.77/0.987/5.604	40.02/0.988/5.705	39.72/0.987/5.576	40.81/0.989/5.748	<b>41.10/0.990/5.737</b>
(10)	2	32.91/0.922/3.573	35.15/0.950/4.470	35.35/0.950/4.440	36.29/0.957/4.540	36.90/0.960/4.525	36.25/0.957/4.499	<b>37.96/0.964/4.622</b>	37.88/0.964/4.608
(11)	2	37.05/0.951/2.866	39.42/0.966/3.352	39.27/0.964/3.290	39.81/0.967/3.353	40.07/0.968/3.304	39.92/0.968/3.325	40.35/0.969/3.356	<b>40.38/0.969/3.360</b>
(12)	2	38.34/0.949/2.916	40.53/0.967/3.486	40.40/0.966/3.422	40.91/0.968/3.510	41.04/0.970/3.499	41.06/0.969/3.497	41.24/0.970/3.543	<b>41.31/0.971/3.548</b>
(13)	2	36.20/0.941/3.775	38.55/0.959/4.353	38.51/0.958/4.306	38.93/0.960/4.364	39.07/0.962/4.368	38.99/0.961/4.355	39.33/0.963/4.399	<b>39.36/0.963/4.405</b>
(14)	2	33.84/0.945/4.742	36.50/0.964/5.355	36.43/0.963/5.305	37.18/0.967/5.349	37.64/0.969/5.348	37.44/0.968/5.333	38.15/0.970/5.424	<b>38.17/0.970/5.427</b>
(15)	2	31.83/0.936/6.327	35.17/0.966/7.572	35.60/0.967/7.550	36.46/0.972/7.548	37.21/0.975/7.534	36.72/0.974/7.532	38.02/0.978/7.652	<b>38.08/0.978/7.660</b>
(16)	2	31.26/0.920/5.463	34.63/0.956/6.625	34.61/0.955/6.569	36.14/0.964/6.717	36.44/0.966/6.648	35.99/0.964/6.701	37.19/0.969/6.740	<b>37.35/0.969/6.747</b>
(17)	2	33.78/0.933/4.433	36.88/0.959/5.294	36.82/0.958/5.199	37.56/0.963/5.300	37.86/0.964/5.247	37.69/0.964/5.270	38.22/0.966/5.355	<b>38.35/0.966/5.362</b>
(18)	2	34.00/0.944/4.304	37.17/0.967/5.066	37.15/0.966/4.986	38.28/0.972/5.065	38.51/0.973/5.022	38.40/0.973/5.029	39.04/0.975/5.085	<b>39.21/0.975/5.085</b>
(19)	2	31.33/0.924/6.328	34.07/0.957/7.558	33.80/0.954/7.620	34.77/0.963/7.567	34.76/0.963/7.619	34.72/0.963/7.518	35.15/0.966/7.659	<b>35.32/0.967/7.664</b>
(20)	2	32.37/0.926/4.947	35.42/0.956/5.800	35.75/0.959/5.739	37.20/0.968/5.867	37.56/0.970/5.786	37.17/0.968/5.828	38.07/0.972/5.890	<b>38.19/0.972/5.897</b>
(21)	2	29.57/0.905/5.137	32.84/0.945/6.269	32.72/0.944/6.186	34.62/0.959/6.412	35.03/0.961/6.308	34.29/0.958/6.324	<b>36.10/0.967/6.434</b>	36.08/0.967/6.430
(22)	2	35.46/0.931/3.450	37.54/0.954/4.091	37.45/0.952/3.990	38.33/0.959/4.103	38.41/0.960/4.065	38.34/0.959/4.082	38.72/0.962/4.153	<b>38.80/0.962/4.157</b>
(23)	2	31.57/0.934/5.460	35.06/0.964/6.485	34.96/0.963/6.549	36.23/0.970/6.487	36.63/0.972/6.499	36.32/0.971/6.445	37.32/0.975/6.553	<b>37.47/0.975/6.560</b>
(24)	2	38.26/0.965/3.085	40.78/0.976/3.375	40.47/0.974/3.367	41.25/0.977/3.362	41.17/0.977/3.391	41.44/0.978/3.358	41.63/0.978/3.437	<b>41.69/0.979/3.437</b>
(25)	2	34.75/0.948/3.281	37.61/0.968/3.958	37.70/0.967/3.946	39.04/0.974/4.018	39.37/0.975/4.014	39.12/0.974/3.991	<b>40.31/0.978/4.097</b>	40.20/0.978/4.098
(26)	2	32.86/0.946/3.699	34.87/0.966/4.312	34.75/0.964/4.242	36.62/0.976/4.480	37.29/0.978/4.310	36.98/0.977/4.314	39.34/0.983/4.514	<b>39.86/0.984/4.535</b>
(27)	2	34.43/0.944/3.425	37.35/0.965/4.132	37.36/0.963/4.092	38.35/0.967/4.176	38.80/0.969/4.143	38.37/0.968/4.136	39.14/0.970/4.188	<b>39.21/0.970/4.193</b>
(28)	2	33.36/0.930/4.591	36.51/0.959/5.420	36.25/0.957/5.359	37.56/0.965/5.411	37.73/0.967/5.364	37.57/0.966/5.367	38.13/0.968/5.417	<b>38.19/0.969/5.422</b>
(29)	2	32.19/0.929/4.881	35.30/0.959/5.782	35.53/0.959/5.813	36.52/0.967/5.819	37.07/0.969/5.804	36.59/0.967/5.778	37.61/0.971/5.881	<b>37.70/0.972/5.886</b>
(30)	2	31.26/0.941/5.749	34.69/0.964/6.484	34.43/0.963/6.527	35.69/0.969/6.513	36.54/0.971/6.559	35.76/0.970/6.486	37.10/0.974/6.605	<b>37.23/0.974/6.604</b>
Avg	2	34.04/0.938/4.263	36.80/0.961/5.004	36.80/0.960/4.970	37.83/0.966/5.033	38.15/0.968/5.008	37.90/0.967/5.000	38.70/0.970/5.080	<b>38.81/0.970/5.085</b>

**Table 2.** Quantitative evaluation of the proposed DDRN approach and its improved version DDRN<sup>+</sup> against some state-of-the-art SISR algorithms on *Kaggle Open Source Dataset* with 30 different scenarios for the scale factor of  $\times 3$ . Bold indicates the best performance. Particularly, \* refers to the modules retrained by us with *Kaggle Open Source Dataset*.

Labels	Methods Scale	Bicubic PSNR/SSIM/AG	SRCNN [23] PSNR/SSIM/AG	SRCNN * PSNR/SSIM/AG	VDSR [24] PSNR/SSIM/AG	VDSR * PSNR/SSIM/AG	DDRN (Our) PSNR/SSIM/AG	DDRN <sup>+</sup> (Our) PSNR/SSIM/AG
(1)	3	33.06/0.915/3.021	34.58/0.935/3.579	35.07/0.940/3.586	36.22/0.953/3.616	36.65/0.955/3.599	37.63/0.961/3.653	<b>37.70/0.962/3.663</b>
(2)	3	28.25/0.821/3.717	30.10/0.871/4.911	30.12/0.870/4.941	31.68/0.903/5.274	31.86/0.903/5.047	32.97/0.919/5.333	<b>33.12/0.920/5.346</b>
(3)	3	34.30/0.897/2.256	35.38/0.913/2.796	35.76/0.918/2.737	36.56/0.930/2.812	36.57/0.930/2.798	37.51/0.938/ <b>2.868</b>	<b>37.68/0.939/2.864</b>
(4)	3	32.88/0.901/3.197	34.68/0.926/3.825	34.89/0.925/3.780	35.58/0.934/3.796	35.71/0.935/3.766	36.40/0.940/3.881	<b>36.49/0.941/3.884</b>
(5)	3	27.86/0.885/5.309	30.35/0.921/6.439	30.87/0.923/6.403	32.11/0.940/6.345	33.31/0.945/6.531	<b>34.68/0.953/6.502</b>	34.43/0.953/6.452
(6)	3	31.17/0.852/3.031	32.44/0.880/3.686	32.38/0.879/3.728	33.72/0.901/3.803	34.03/0.902/3.731	35.13/0.912/3.843	<b>35.27/0.914/3.870</b>
(7)	3	32.92/0.870/2.979	34.18/0.893/3.548	34.15/0.891/3.474	34.83/0.902/3.546	34.67/0.900/3.519	35.11/0.905/3.589	<b>35.25/0.906/3.609</b>
(8)	3	33.32/0.907/2.070	34.90/0.930/2.593	34.93/0.929/2.566	35.89/0.939/2.611	35.74/0.938/2.573	36.48/0.943/2.668	<b>36.70/0.944/2.680</b>
(9)	3	29.16/0.897/4.537	32.80/0.945/5.666	32.79/0.943/5.643	34.18/0.964/5.508	34.90/0.966/5.432	36.07/0.974/5.563	<b>36.29/0.974/5.578</b>
(10)	3	29.89/0.843/2.766	31.04/0.878/3.594	31.04/0.877/3.579	31.79/0.894/3.743	32.21/0.897/3.710	<b>33.06/0.906/3.803</b>	33.05/0.907/3.793
(11)	3	33.43/0.903/2.384	35.52/0.930/2.971	35.44/0.928/2.961	36.16/0.937/2.988	36.23/0.937/2.927	36.87/0.942/3.050	<b>37.01/0.943/3.062</b>
(12)	3	34.62/0.888/2.311	35.95/0.913/2.991	35.97/0.912/2.937	36.47/0.919/3.007	36.50/0.919/2.972	36.86/0.923/3.070	<b>36.93/0.923/3.078</b>
(13)	3	32.70/0.881/3.189	34.14/0.908/3.837	34.22/0.908/3.804	35.17/0.916/3.871	35.35/0.917/3.860	35.81/0.920/3.927	<b>35.88/0.921/3.940</b>
(14)	3	30.20/0.888/4.099	32.03/0.919/4.961	32.24/0.919/4.888	33.07/0.931/4.898	33.50/0.933/4.900	34.37/0.940/4.947	<b>34.58/0.942/4.966</b>
(15)	3	27.84/0.844/5.081	29.70/0.893/7.098	30.40/0.907/6.934	31.27/0.924/7.059	31.69/0.927/7.010	32.46/0.942/ <b>7.138</b>	<b>33.49/0.948/7.104</b>
(16)	3	27.70/0.822/4.472	29.21/0.872/5.548	29.29/0.872/5.605	30.99/0.902/5.901	31.20/0.904/5.744	32.33/0.916/5.947	<b>32.60/0.918/5.969</b>
(17)	3	29.95/0.854/3.684	31.87/0.896/4.683	32.18/0.897/4.620	33.26/0.916/4.722	33.22/0.915/4.644	33.98/0.924/4.793	<b>34.13/0.925/4.807</b>
(18)	3	30.15/0.875/3.615	32.14/0.913/4.622	32.36/0.913/4.540	33.61/0.933/4.625	33.66/0.931/4.550	34.52/0.940/4.699	<b>34.68/0.941/4.703</b>
(19)	3	27.82/0.829/5.063	29.49/0.886/6.717	29.46/0.884/6.934	30.36/0.907/6.705	30.07/0.901/6.727	30.79/0.915/6.951	<b>30.97/0.918/6.957</b>
(20)	3	28.97/0.842/4.186	30.99/0.891/5.177	31.20/0.895/5.132	32.50/0.921/5.324	32.68/0.923/5.276	33.62/0.934/5.392	<b>33.85/0.936/5.397</b>
(21)	3	26.45/0.808/4.169	28.29/0.865/5.253	28.30/0.863/5.204	29.87/0.900/5.555	30.04/0.901/5.449	31.57/0.920/5.623	<b>31.59/0.921/5.637</b>
(22)	3	32.43/0.866/2.898	33.84/0.895/3.522	33.77/0.893/3.453	34.45/0.905/3.524	34.38/0.904/3.488	34.81/0.909/3.572	<b>34.91/0.911/3.586</b>
(23)	3	27.88/0.852/4.529	30.12/0.900/5.799	30.11/0.898/5.763	31.24/0.919/5.785	31.40/0.920/5.731	32.41/0.932/5.829	<b>32.53/0.932/5.833</b>
(24)	3	34.58/0.927/2.702	36.80/0.948/3.215	36.72/0.947/3.185	37.82/0.956/3.164	37.66/0.955/3.191	38.46/0.960/3.199	<b>38.64/0.961/3.210</b>
(25)	3	31.05/0.891/2.640	32.75/0.921/3.440	33.19/0.923/3.448	34.36/0.938/3.549	34.60/0.938/3.467	35.75/0.947/3.575	<b>35.97/0.949/3.592</b>
(26)	3	29.70/0.884/3.075	30.66/0.912/3.843	30.90/0.913/3.724	31.01/0.923/3.967	31.55/0.928/3.747	32.31/0.940/3.991	<b>33.16/0.948/4.009</b>
(27)	3	30.89/0.880/2.747	32.44/0.909/3.499	32.70/0.909/3.452	33.80/0.922/3.584	34.07/0.923/3.538	34.83/0.929/3.623	<b>34.92/0.929/3.632</b>
(28)	3	30.14/0.862/3.958	32.37/0.905/4.850	32.09/0.900/4.838	33.41/0.923/4.877	33.21/0.920/4.801	34.02/0.930/4.731	<b>34.18/0.932/4.747</b>
(29)	3	28.67/0.851/4.072	30.74/0.899/5.104	30.92/0.900/5.130	32.09/0.924/5.207	32.33/0.926/5.162	33.23/0.936/5.277	<b>33.41/0.938/5.294</b>
(30)	3	27.62/0.876/4.913	29.80/0.916/5.971	29.86/0.915/5.927	31.03/0.933/5.969	31.91/0.937/6.018	<b>33.74/0.949/6.097</b>	33.49/0.948/6.065
Avg	3	30.52/0.870/3.555	32.31/0.906/4.457	32.44/0.906/4.430	33.48/0.923/4.511	33.69/0.924/4.463	34.59/0.933/4.571	<b>34.76/0.935/4.577</b>

**Table 3.** Comparison results of the proposed DDRN approach and its improved version DDRN<sup>+</sup> with some state-of-the-art algorithms on *Kaggle Open Source Dataset* for the scale factor of 4. Bold indicates the best performance. Particularly, \* refers to the modules retrained by us with *Kaggle Open Source Dataset*.

Labels	Methods Scale	Bicubic PSNR/SSIM/AG	SRCNN [23] PSNR/SSIM/AG	SRCNN* PSNR/SSIM/AG	VDSR [24] PSNR/SSIM/AG	VDSR* PSNR/SSIM/AG	LapSRN [25] PSNR/SSIM/AG	DDRN (Our) PSNR/SSIM/AG	DDRN <sup>+</sup> (Our) PSNR/SSIM/AG
(1)	4	30.84/0.867/2.664	32.15/0.892/3.234	32.41/0.897/3.150	33.29/0.917/3.281	33.69/0.922/3.311	33.50/0.920/3.263	34.60/0.934/3.343	<b>34.74/0.936/3.354</b>
(2)	4	26.41/0.739/2.997	27.80/0.792/4.129	27.77/0.788/3.924	28.92/0.834/4.438	29.27/0.839/4.290	28.96/0.835/4.328	30.51/0.868/4.547	<b>30.67/0.872/4.680</b>
(3)	4	32.35/0.852/1.929	33.31/0.873/2.475	33.81/0.878/2.349	34.49/0.895/2.502	34.58/0.898/2.474	34.72/0.899/2.469	35.33/0.907/2.467	<b>35.67/0.910/2.551</b>
(4)	4	30.52/0.847/2.809	32.19/0.880/3.432	32.13/0.875/3.385	32.94/0.893/3.452	33.11/0.894/3.387	33.06/0.895/3.404	33.72/0.904/3.362	<b>33.95/0.906/3.365</b>
(5)	4	25.31/0.816/4.522	27.18/0.861/5.814	27.75/0.867/5.655	28.30/0.888/5.735	30.92/0.909/6.079	28.39/0.890/5.738	<b>32.32/0.926/6.258</b>	31.94/0.924/6.364
(6)	4	29.57/0.799/2.610	30.85/0.836/3.258	30.51/0.825/3.169	31.41/0.856/3.440	32.04/0.865/3.393	32.03/0.865/3.331	33.33/0.886/3.401	<b>33.55/0.888/3.485</b>
(7)	4	31.03/0.822/2.635	32.32/0.849/3.171	32.20/0.845/3.105	32.95/0.864/3.205	32.89/0.862/3.177	33.07/0.865/3.173	33.30/0.870/3.137	<b>33.45/0.873/3.276</b>
(8)	4	31.58/0.871/1.771	32.81/0.895/2.196	32.84/0.894/2.138	33.66/0.908/2.208	33.63/0.908/2.177	33.76/0.909/2.188	34.22/0.916/2.187	<b>34.45/0.918/2.189</b>
(9)	4	26.90/0.831/4.097	30.23/0.904/5.256	29.94/0.898/5.059	31.51/0.935/4.986	31.42/0.933/4.864	31.69/0.938/4.957	32.81/0.950/5.094	<b>33.18/0.954/5.228</b>
(10)	4	28.47/0.783/2.246	29.26/0.816/2.947	29.37/0.817/2.864	29.77/0.835/3.061	30.09/0.839/3.004	29.71/0.834/3.013	30.86/0.856/3.153	<b>30.88/0.856/3.182</b>
(11)	4	31.32/0.858/2.041	33.19/0.891/2.586	33.03/0.887/2.518	33.81/0.903/2.637	33.87/0.904/2.549	33.94/0.905/2.605	34.49/0.913/2.554	<b>34.71/0.915/2.552</b>
(12)	4	32.49/0.830/1.867	33.53/0.856/2.479	33.53/0.854/2.360	33.99/0.866/2.529	34.05/0.865/2.554	34.01/0.867/2.481	34.31/0.870/2.408	<b>34.52/0.873/2.394</b>
(13)	4	30.75/0.822/2.745	31.95/0.853/3.404	32.04/0.853/3.320	32.59/0.865/3.412	32.68/0.865/3.344	32.62/0.866/3.358	33.32/0.874/3.340	<b>33.55/0.876/3.333</b>
(14)	4	27.94/0.830/3.570	29.52/0.868/4.509	29.60/0.865/4.340	30.29/0.884/4.413	30.72/0.889/4.437	30.51/0.888/4.414	31.49/0.903/4.555	<b>31.79/0.907/4.601</b>
(15)	4	25.70/0.744/4.120	27.11/0.805/6.071	27.29/0.808/5.721	28.16/0.842/6.114	28.27/0.845/6.140	28.33/0.850/6.032	29.11/0.872/6.271	<b>29.37/0.875/6.326</b>
(16)	4	25.98/0.738/3.809	27.29/0.797/4.809	27.29/0.796/4.712	28.00/0.827/4.964	28.15/0.830/4.890	28.02/0.828/4.894	29.33/0.857/5.089	<b>29.84/0.863/5.152</b>
(17)	4	27.91/0.784/3.139	29.49/0.832/4.069	29.51/0.930/3.904	30.56/0.862/4.130	30.60/0.862/3.997	30.63/0.864/4.101	31.37/0.878/4.083	<b>31.61/0.881/4.100</b>
(18)	4	28.10/0.810/3.091	29.65/0.855/4.082	29.72/0.853/3.910	30.81/0.884/4.117	30.93/0.885/4.172	30.90/0.886/4.072	31.73/0.899/4.083	<b>31.89/0.902/4.108</b>
(19)	4	25.79/0.734/4.064	27.01/0.802/5.738	27.00/0.796/5.497	27.55/0.827/5.677	27.48/0.823/5.552	27.57/0.829/5.684	27.96/0.846/5.910	<b>28.17/0.852/5.999</b>
(20)	4	27.06/0.766/3.612	28.50/0.818/4.575	28.69/0.821/4.411	29.57/0.854/4.643	29.90/0.864/4.606	29.64/0.858/4.621	30.77/0.886/4.753	<b>31.18/0.893/4.787</b>
(21)	4	24.87/0.733/3.487	26.18/0.792/4.517	26.32/0.794/4.429	27.12/0.831/4.737	27.70/0.841/4.798	27.07/0.832/4.684	<b>29.20/0.875/4.943</b>	28.96/0.873/5.042
(22)	4	30.72/0.811/2.541	31.96/0.844/0.050	31.91/0.840/2.941	32.37/0.855/3.066	32.42/0.854/3.003	32.41/0.855/3.025	32.77/0.862/2.945	<b>32.87/0.864/2.935</b>
(23)	4	25.85/0.779/3.834	27.52/0.832/5.104	27.73/0.833/4.950	28.47/0.860/5.037	28.62/0.862/4.988	28.60/0.862/4.991	29.42/0.879/5.049	<b>29.67/0.882/5.120</b>
(24)	4	32.16/0.883/2.382	34.09/0.912/2.929	33.91/0.906/2.800	35.09/0.926/2.902	34.82/0.922/2.872	35.18/0.927/2.877	35.16/0.927/2.904	<b>35.86/0.934/2.909</b>
(25)	4	29.08/0.839/2.174	30.34/0.872/2.898	30.38/0.871/2.819	31.50/0.898/3.091	31.97/0.901/3.140	31.59/0.901/3.075	33.09/0.917/3.149	<b>33.39/0.919/3.173</b>
(26)	4	27.96/0.824/2.586	28.89/0.864/3.433	29.01/0.861/3.194	28.89/0.877/3.787	29.78/0.888/3.629	29.13/0.879/3.489	30.20/0.901/3.626	<b>30.82/0.910/3.629</b>
(27)	4	29.06/0.824/2.275	30.30/0.855/2.943	30.25/0.854/2.885	31.15/0.873/3.054	31.45/0.875/2.987	31.23/0.874/2.995	32.17/0.885/3.012	<b>32.35/0.887/3.001</b>
(28)	4	28.33/0.800/3.517	30.02/0.850/4.333	29.87/0.844/4.237	30.93/0.874/4.340	30.74/0.871/4.310	31.02/0.876/4.289	31.47/0.887/4.352	<b>31.72/0.891/4.357</b>
(29)	4	26.80/0.783/3.483	28.46/0.837/4.489	28.56/0.836/4.439	29.41/0.868/4.537	29.61/0.872/4.440	29.47/0.870/4.509	30.38/0.891/4.610	<b>30.65/0.895/4.709</b>
(30)	4	25.46/0.810/4.213	27.40/0.863/5.412	27.34/0.862/5.313	28.18/0.888/5.406	30.00/0.904/5.599	28.26/0.891/5.428	<b>31.39/0.922/5.756</b>	31.26/0.920/5.729
Avg	4	28.54/0.808/3.028	30.01/0.850/3.911	30.06/0.848/3.783	30.86/0.873/3.963	31.08/0.875/3.60	30.97/0.875/3.916	32.00/0.892/4.018	<b>32.22/0.895/4.064</b>



Another group of comparison experiments are conducted with the *Jilin-1* satellite imagery to illustrate the effectiveness and applicability of the proposed ultra-dense strategy and distillation and compensation mechanism further. Compared with the first dataset *Kaggle Open Source Dataset*, the test images obtained from *Jilin-1* show lower quality (small ground objects and weak textures) but more realistic satellite imagery characteristics. Unlike the images in the training dataset, the test images have completely different imaging conditions, including ultra-high imaging distance, atmospheric scattering, relative motion between satellite and moving ground targets, and compression distortion. These severe imaging conditions pose substantial demands to SR networks.

With an operation similar to the previously presented preprocessing of the testing images, we crop the test images with the size of  $480 \times 204$ . The reconstruction results obtained from our proposed approaches and the comparison methods are shown in Figure 10. For the first and second images, most of the comparison methods produce noticeable artifacts and blurred edges. By contrast, the proposed DDRN and DDRN<sup>+</sup> can recover sharp and clear edges because of fine feature expression that is faithful to the ground truth. At the bottom of the figure, only our proposed modules can reconstruct a clear outline of the warships and dock, whereas the other conventional methods fail to restore the realistic details. These results further indicate the effectiveness of the proposed method.

Furthermore, we perform a set of realistic SR reconstruction experiments for the unknown real degradation process (i.e., using the observed LR images instead of the downscaled LR images as input). These test images are randomly selected from *Jilin-1* satellite imagery using the same preprocessing to acquire the test images with the size of  $480 \times 204$ . Then, the processed images used as the LR input are directly transmitted to the network to obtain the reconstructed HR images. The comparison results with other state-of-the-art algorithms are shown in Figure 11 (we show only one example due to space constraints). Evidently, most of compared methods [23,24] produce noticeable artifacts and blurred building outlines, whereas the proposed DDRN and DDRN<sup>+</sup> yield better results with fewer jagged lines and ringing artifacts. Instead of the commonly used evaluation metrics PSNR and SSIM (because the original HR images are unavailable), we introduce the AG to measure the sharpness of the SR results. As shown in Figure 11, the proposed modules DDRN and DDRN<sup>+</sup> enjoy the second and first highest AG scores, respectively. The results for real video satellite imagery indicate that our model is more robust than the comparison methods in super-resolving the image with unknown degradation process.

In brief, the SR reconstruction experiments on different test datasets and magnification scales show the advantages of feature expression and indicate the robustness of our modules against images of unknown degradation models.

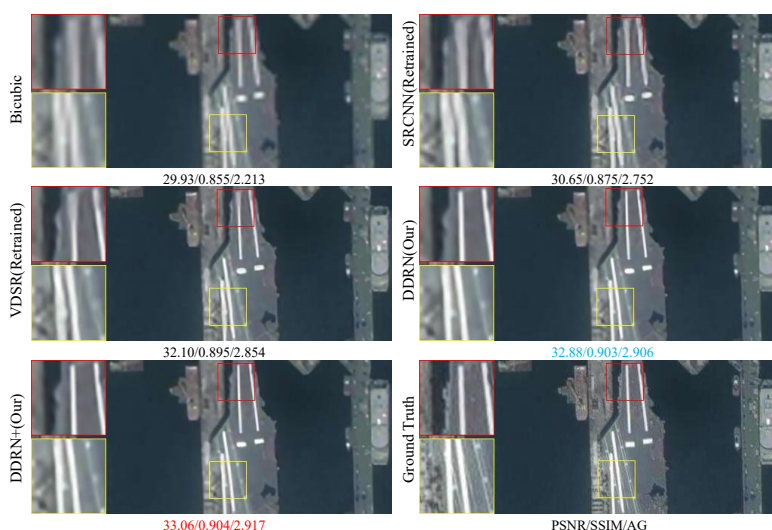
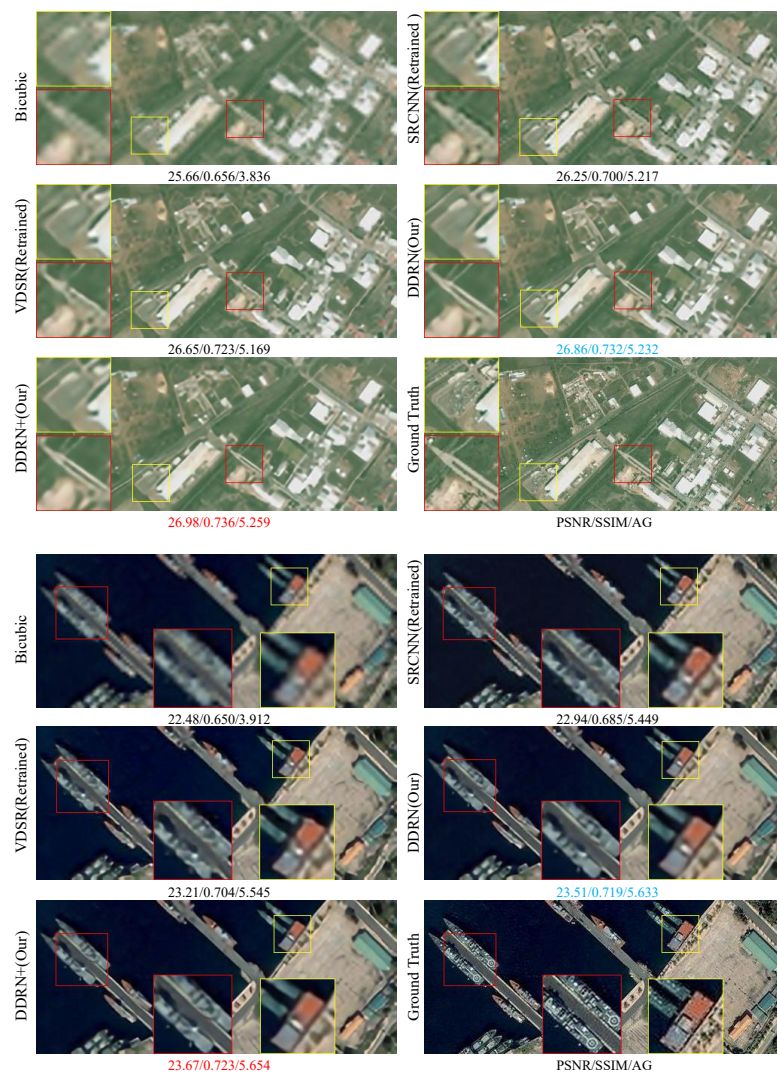
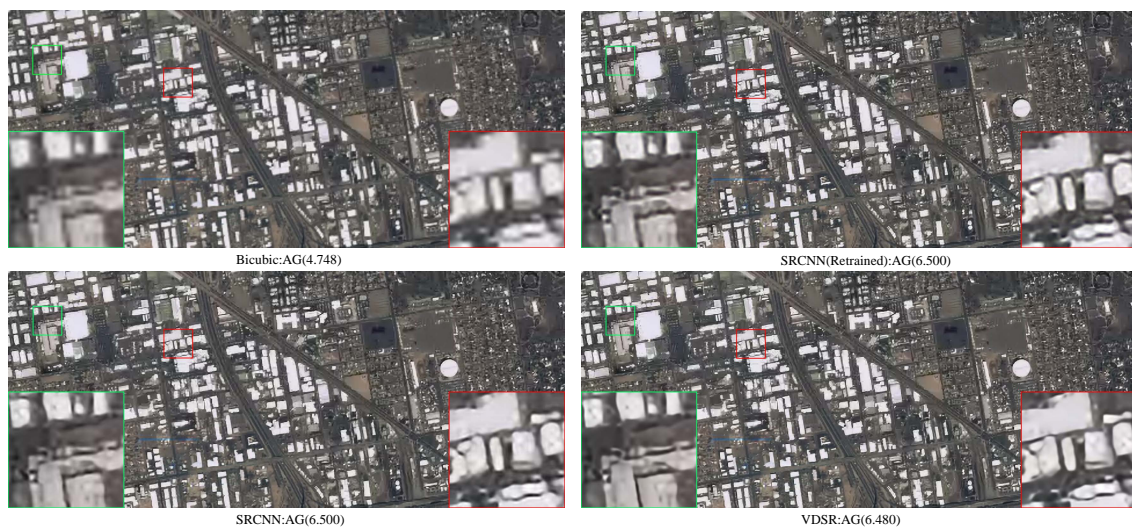


Figure 10. Cont.





**Figure 10.** The reconstruction results on *Jilin-1* dataset with the scale of 4. We select several different but representative scenarios, i.e., aircraft carrier, city suburb, and military harbour to make comparisons. Red and blue indicate the best and the second best performance, respectively.



**Figure 11.** Cont.



**Figure 11.** An example for the reconstruction results on *Jilin-1* imagery by the scale of 4. The experiment is performed with real low satellite images rather than simulation degradation. Red and blue respectively indicate the first and the second best performance in terms of AG. Note that the enlarged details are shown in the boxes on the bottom left and bottom right in each image.

## 6. Conclusions

In this study, we propose a simple but very effective technique for remote sensing image SR reconstruction. In particular, we present a multiple-path UDB for local feature extraction and fusion. Unlike in the conventional methods, rich dense connections between layers and units promote information interaction and improve reutilization. In addition, we further promote feature expression by advocating a distillation and compensation mechanism. The feature maps distilled from different stages with a special distillation ratio  $\alpha$  are aggregated to compensate for the high-frequency details lost during information propagation in MSPU. Extensive experiments on the test datasets indicate that the proposed DDRN and its improved version DDRN<sup>+</sup> outperform existing state-of-the-art feature extraction techniques, including conventional direct- and skip-connection-based methods. In particular, when the image degradation model is unknown, the proposed algorithm can still obtain competitive reconstruction results compared with the comparison algorithms.

**Author Contributions:** Conceptualization, K.J. and Z.W.; Data curation, K.J. and P.Y.; Formal analysis, P.Y. and J.J.; Funding acquisition, Z.W.; Investigation, J.X.; Methodology, K.J. and Z.W.; Project administration, K.J.; Resources, Z.W.; Software, P.Y.; Supervision, Y.Y.; Validation, K.J., Z.W. and J.J.; Visualization, P.Y.; Draft Preparation, K.J.; Writing, review and editing, Z.W. and J.J.

**Funding:** This research was funded by [National Natural Science Foundation of China] grant number [61671332, 41771452, 41771454, 61501413], [National Key R&D Project] grant number [2016YFE0202300], and [Hubei Province Technological Innovation Major Project] grant number [2017AAA123]. (Corresponding author: Zhongyuan Wang).

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

## Abbreviations

The following abbreviations are used in this manuscript:

CNNs	Convolutional neural networks
SR	Super-resolution
SISR	Single image super-resolution
LR	Low resolution
HR	High resolution
DDRN	Deep distillation recursive network



UDB      Ultra-dense residual block  
 MSPU    Multi-scale purification unit

## References

1. Xu, Y.; Zhu, M.; Li, S.; Feng, H.; Ma, S.; Che, J. End-to-end airport detection in remote sensing images combining cascade region proposal networks and multi-threshold detection networks. *Remote Sens.* **2018**, *10*, 1516. [\[CrossRef\]](#)
2. Shao, Z.; Wu, W.; Wang, Z.; Du, W.; Li, C. SeaShips: A large-scale precisely annotated dataset for ship detection. *IEEE Trans. Multimed.* **2018**, *20*, 2593–2604. [\[CrossRef\]](#)
3. Ma, J.; Zhao, J.; Tian, J.; Yuille, A.L.; Tu, Z. Robust point matching via vector field consensus. *IEEE Trans. Image Process.* **2014**, *23*, 1706–1721.
4. Ma, J.; Jiang, J.; Zhou, H.; Zhao, J.; Guo, X. Guided locality preserving feature matching for remote sensing image registration. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4435–4447. [\[CrossRef\]](#)
5. Jiang, J.; Ma, J.; Chen, C.; Wang, Z.; Cai, Z.; Wang, L. SuperPCA: A superpixelwise PCA approach for unsupervised feature extraction of hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4581–4593. [\[CrossRef\]](#)
6. Jiang, J.; Ma, J.; Wang, Z.; Chen, C.; Liu, X. Hyperspectral image classification in the presence of noisy labels. *IEEE Trans. Geosci. Remote Sens.* **2018**. [\[CrossRef\]](#)
7. Li, C.; Ma, Y.; Mei, X.; Liu, C.; Ma, J. Hyperspectral unmixing with robust collaborative sparse regression. *Remote Sens.* **2016**, *8*, 588. [\[CrossRef\]](#)
8. Ma, J.; Ma, Y.; Li, C. Infrared and visible image fusion methods and applications: A survey. *Inf. Fus.* **2019**, *45*, 153–178. [\[CrossRef\]](#)
9. Ma, J.; Yu, W.; Liang, P.; Li, C.; Jiang, J. FusionGAN: A generative adversarial network for infrared and visible image fusion. *Inf. Fus.* **2019**, *48*, 11–26. [\[CrossRef\]](#)
10. Dong, W.; Fu, F.; Shi, G.; Cao, X.; Wu, J.; Li, G.; Li, X. Hyperspectral image super-resolution via non-negative structured sparse representation. *IEEE Trans. Image Process.* **2016**, *25*, 2337–2352. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Yang, S.; Sun, F.; Wang, M.; Liu, Z.; Jiao, L. Novel super resolution restoration of remote sensing images based on compressive sensing and example patches-aided dictionary learning. In Proceedings of the International Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping, Xiamen, China, 10–12 January 2011; pp. 1–6.
12. Jiang, K.; Wang, Z.; Yi, P.; Jiang, J. A progressively enhanced network for video satellite imagery superresolution. *IEEE Signal Process. Lett.* **2018**, *25*, 1630–1634. [\[CrossRef\]](#)
13. Luo, Y.; Zhou, L.; Wang, S.; Wang, Z. Video satellite imagery super resolution via convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2398–2402. [\[CrossRef\]](#)
14. Xiao, A.; Wang, Z.; Wang, L.; Ren, Y. Super-resolution for “Jilin-1” satellite video imagery via a convolutional network. *Sensors* **2018**, *18*, 1194. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Merino, M.T.; Nunez, J. Super-resolution of remotely sensed images with variable-pixel linear reconstruction. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 1446–1457. [\[CrossRef\]](#)
16. Li, F.; Jia, X.; Fraser, D. Universal HMT based super resolution for remote sensing images. In Proceedings of the 15th IEEE Conference on ICIP, San Diego, CA, USA, 12–15 October 2008; pp. 333–336.
17. Gou, S.; Liu, S.; Yang, S.; Jiao, L. Remote sensing image super-resolution reconstruction based on nonlocal pairwise dictionaries and double regularization. *IEEE J. Sel. Top. Appl. Earth Observ.* **2014**, *7*, 4784–4792. [\[CrossRef\]](#)
18. Li, Y.; Zhang, Y.; Huang, X.; Zhu, H.; Ma, J. Large-scale remote sensing image retrieval by deep hashing neural networks. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 950–965. [\[CrossRef\]](#)
19. Shao, Z.; Cai, J. Remote sensing image fusion with deep convolutional neural network. *IEEE J. Sel. Top. Appl. Earth Observ.* **2018**, *11*, 1656–1669. [\[CrossRef\]](#)
20. Zhou, W.; Newsam, S.; Li, C.; Shao, Z. Learning low dimensional convolutional neural networks for high-resolution remote sensing image retrieval. *Remote Sens.* **2017**, *9*, 489–508. [\[CrossRef\]](#)
21. Lu, T.; Ming, D.; Lin, X.; Hong, Z.; Bai, X.; Fang, J. Detecting building edges from high spatial resolution remote sensing imagery using richer convolution features network. *Remote Sens.* **2018**, *10*, 1496. [\[CrossRef\]](#)

22. Zhang, W.; Witharana, C.; Liljedahl, A.K.; Kanevskiy, M. Deep convolutional neural networks for automated characterization of arctic ice-wedge polygons in very high spatial resolution aerial imagery. *Remote Sens.* **2018**, *10*, 1487. [[CrossRef](#)]
23. 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](#)]
24. Kim, J.; Lee, J.K.; Lee, K.M. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE Conference on CVPR, Las Vegas, NV, USA, 27–30 June 2016; pp. 1646–1654.
25. Lai, W.S.; Huang, J.B.; Ahuja, N.; Yang, M.H. Deep laplacian pyramid networks for fast and accurate super-resolution. In Proceedings of the IEEE Conference on CVPR, Honolulu, HI, USA, 21–26 July 2017; pp. 5835–5843.
26. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on CVPR, Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
27. Tong, T.; Li, G.; Liu, X.; Gao, Q. Image super-resolution using dense skip connections. In Proceedings of the IEEE Conference on ICCV, Venice, Italy, 22–29 October 2017; pp. 4809–4817.
28. Tai, Y.; Yang, J.; Liu, X.; Xu, C. MemNet: A persistent memory network for image restoration. In Proceedings of the IEEE Conference on ICCV, Venice, Italy, 22–29 October 2017; pp. 4549–4557.
29. Kim, J.; Lee, J.K.; Lee, K.M. Deeply-recursive convolutional network for image super-resolution. In Proceedings of the IEEE Conference on CVPR, Las Vegas, NV, USA, 27–30 June 2016; pp. 1637–1645.
30. Tai, Y.; Yang, J.; Liu, X. Image super-resolution via deep recursive residual network. In Proceedings of the IEEE Conference on CVPR, Honolulu, HI, USA, 21–26 July 2017; pp. 2790–2798.
31. Yang, W.; Feng, J.; Yang, J.; Zhao, F.; Liu, J.; Guo, Z.; Yan, S. Deep edge guided recurrent residual learning for image super-resolution. *IEEE Trans. Image Process.* **2017**, *26*, 5895–5907. [[CrossRef](#)] [[PubMed](#)]
32. Yim, J.; Joo, D.; Bae, J.; Kim, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In Proceedings of the IEEE Conference on CVPR, Honolulu, HI, USA, 21–26 July 2017; pp. 7130–7138.
33. Gupta, S.; Hoffman, J.; Malik, J. Cross modal distillation for supervision transfer. In Proceedings of the IEEE Conference on CVPR, Las Vegas, NV, USA, 27–30 June 2016; pp. 2827–2836.
34. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *Comput. Sci.* **2015**, *14*, 38–39.
35. Pintea, S.L.; Liu, Y.; van Gemert, J.C. Recurrent knowledge distillation. In Proceedings of the 25th IEEE Conference on ICIP, Athens, Greece, 7–10 October 2018; pp. 3393–3397.
36. Zhao, P.; Liu, K.; Zou, H.; Zhen, X. Multi-stream convolutional neural network for SAR automatic target recognition. *Remote Sens.* **2018**, *10*, 1473. [[CrossRef](#)]
37. Gu, S.; Zuo, W.; Xie, Q.; Meng, D.; Feng, X.; Zhang, L. Convolutional sparse coding for image super-resolution. In Proceedings of the IEEE Conference on ICCV, Santiago, Chile, 7–13 December 2015; pp. 1823–1831.
38. Huang, G.; Chen, D.; Li, T.; Wu, F.; van der Maaten, L.; Weinberger, K.Q. Multi-scale dense networks for resource efficient image classification. *arXiv* **2018**, arXiv:1703.09844.
39. Yang, J.; Wright, J.; Huang, T.S.; Ma, Y. Image super-resolution via sparse representation. *IEEE Trans. Image Process.* **2010**, *19*, 2861–2873. [[CrossRef](#)] [[PubMed](#)]
40. Timofte, R.; De, V.; Gool, L.V. Anchored neighborhood regression for fast example-based super-resolution. In Proceedings of the IEEE Conference on ICCV, Sydney, Australia, 8 April 2013; pp. 1920–1927.
41. Yang, J.; Wright, J.; Huang, T.; Ma, Y. Image super-resolution as sparse representation of raw image patches. In Proceedings of the IEEE Conference on CVPR, Anchorage, Alaska, 23–28 June 2008; pp. 1–8.
42. Yang, C.Y.; Yang, M.H. Fast direct super-resolution by simple functions. In Proceedings of the IEEE Conference on ICCV, Sydney Australia, 8 April 2013; pp. 561–568.
43. Dong, C.; Chen, C.L.; Tang, X. Accelerating the super-resolution convolutional neural network. In Proceedings of the IEEE Conference on ECCV, Amsterdam, The Netherlands, 8–16 October 2016; pp. 391–407.
44. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. *arXiv* **2018**, arXiv:1709.0150.
45. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [[CrossRef](#)]
46. Osendorfer, C.; Soyer, H.; Smagt, P.V.D. Image super-resolution with fast approximate convolutional sparse coding. In Proceedings of the International Conference on Neural Information Processing, Kuching, Malaysia, 3–6 November 2014; pp. 250–257.

47. 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 CVPR, Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.
48. Lim, B.; Son, S.; Kim, H.; Nah, S.; Lee, K.M. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE Conference on CVPRW, Honolulu, HI, USA, 21–26 July 2017; pp. 1132–1140.
49. Yang, C.Y.; Ma, C.; Yang, M.H. Single-image super-resolution: A nenchmark. *Lect. Notes Comput. Sci.* **2014**, *8692*, 372–386.
50. 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](#)]
51. Lai, W.; Huang, J.; Ahuja, N.; Yang, M. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *1*. [[CrossRef](#)] [[PubMed](#)]
52. Timofte, R.; Agustsson, E.; Gool, L.V.; Yang, M.H.; Zhang, L.; Limb, B.; Som, S.; Kim, H.; Nah, S.; Lee, K.M.; et al. NTIRE 2017 challenge on single image super-resolution: Methods and results. In Proceedings of the IEEE Conference on CVPRW, Honolulu, HI, USA, 21–26 July 2017; pp. 1110–1121.
53. Huang, J.B.; Singh, A.; Ahuja, N. Single image super-resolution from transformed self-exemplars. In Proceedings of the IEEE Conference on CVPR, Boston, MA, USA, 7–12 June 2015; pp. 5197–5206.
54. Tao, X.; Gao, H.; Liao, R.; Wang, J.; Jia, J. Detail-revealing deep video super-resolution. In Proceedings of the International Conference on ICCV, Venice, Italy, 22–29 October 2017; pp. 4482–4490.
55. Loncan, L.; de Almeida, L.B.; Bioucas-Dias, J.M.; Briottet, X.; Chanussot, J.; Dobigeon, N.; Fabre, S.; Liao, W.; Licciardi, G.A.; Simoes, M.; et al. Hyperspectral pansharpening: A review. *IEEE Geosc. Remote Sens. Mag.* **2015**, *3*, 27–46. [[CrossRef](#)]
56. Vivone, G.; Alparone, L.; Chanussot, J.; Mura, M.D.; Garzelli, A.; Licciardi, G.A.; Restaino, R.; Wald, L. A critical comparison among pansharpening algorithms. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 2565–2586. [[CrossRef](#)]
57. Kwan, C.; Budavari, B.; Bovik, A.C.; Marchisio, G. Blind quality assessment of fused WorldView-3 images by using the combinations of pansharpening and hypersharpening paradigms. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1835–1839. [[CrossRef](#)]
58. Chen, A.; Chen, B.; Chai, X.; Bian, R.; Li, H. A novel stochastic stratified average gradient method: Convergence rate and its complexity. *arXiv* **2017**, arXiv:1710.07783.



© 2018 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 (<http://creativecommons.org/licenses/by/4.0/>).