Chapter 5

Development of Deep Learning Joint Single Image Super-resolution and Deblurring Network for Remote Sensing Images

5.1 Introduction

In the recent years, DL-based techniques have gained remarkable progress in the fields of image processing, computer vision, and its related areas; it achieves state-of-the-art performance for the SISR of natural RGB RS images. DL-based SR methods have become emerging research topic in RS applications. Due to the automated extraction of high-level and complex features besides low-level features at different abstraction levels, it becomes very effective for RS imagery because they frequently have complex structures and detailed textures at multiple resolutions, which cannot be represented well by the hand-crafted features alone.

State-of-the-art DL-based SR techniques can be divided mainly into two categories: generative adversarial networks (GAN) [52, 59, 110] and convolutional neural networks (CNN) [18, 22, 45, 53, 97, 125]. Although GAN-based SR networks approximate the original HR images to produce more realistic and perceptually enhanced HR images, their main limitation is that the reconstructed HR images obtained by these networks have a large structural difference from the original HR images. Furthermore, GAN network training is a challenging task as it often suffers from training instability, leading to problems, like mode collapse or oscillation that pose significant practical constraints [88]. CNN-based SR methods CNN-based SR methods [18, 22, 45, 53, 125, 132] use advanced network architectures, residual learning, and attention mechanisms to minimize the structural error between reconstructed HR and original images by learning the intricate relationships between LR and HR images. These CNN-based methods aim to capture and restore the missing

high-frequency details and structural information in the LR images, bringing the reconstructed HR images closer to the original ones. They achieve this by training on a large dataset of LR-HR image pairs and optimizing network parameters. However, existing CNN-based RS super-resolution approaches still have the following limitations: (1) there is scope for further improvement of the existing networks performance by extracting the relevant features from the input LR using deep neural networks. However, vanishing or exploding gradient problems are still prevalent in deeper models (2) Existing RS super-resolution models treat spatial and channel features equally, limiting their flexibility in dealing with diverse kinds of information, and their ability to discriminate for the extraction of more meaningful features; (3) They only perform the bicubic kernel-based downsampling operation for the LR image generation within the image degradation model. However, in real-world image degradation, mixed processes, such as downsampling due to imaging sensor's resolution and Gaussian blurring due to finite aperture of the optical lens and atmospheric turbulence are used to mimic the natural process of conversion from HR to LR images. In contrast, traditional CNN-based methods are unable to efficiently reconstruct HR images from the blurred and downsampled LR images.

In this chapter, we focus on developing a hybrid dual-branch CNN network that performs both SR and image deblurring tasks concurrently in order to recover clear and sharp HR images from blurry LR remote sensing images. The feature extraction step is divided into two task-independent branches, including deblurring and SR feature extraction, and then an attention-based gated module is used to adaptively fuse the features from both the branches, allowing the dual-branch CNN network to perform both SR and deblurring tasks at the same time. A residual spatial and channel squeeze-and-excitation (RSCSE) module is designed to extract SR features; specifically, a concurrent spatial and channel squeeze-and-excitation (SCSE) module is used in a residual network. By recalibrating the feature maps simultaneously, this concurrent SCSE module is capable of making feature maps more representational. Furthermore, to adaptively retain local features, each RSCSE module utilises the local feature fusion (LFF) concept. Similarly, the deblurring module is designed in such a way that extracts sharp features from the blurry LR image using a simple

SCSE-based encoder-decoder CNN structure. The proposed joint DL network is evaluated on publicly available RS as well as real MS image datasets, outperforming state-of-the-art methods both in terms of visual analysis and objective criteria.

5.1.1 Main contributions of the chapter

The main contributions of the proposed work are summarized as follows:

(i) We propose a joint dual-branch CNN network for the SR of RS images, which have undergone both blurring and downsampling in the process of its acquisition. By addressing the dual problems of SR and deblurring simultaneously, the proposed method restores clear and sharp HR images from a blurred LR image. To achieve this, we have designed an attention-based gated DL module that effectively combines the features extracted from both the SR and deblurring networks.

(ii) We propose an SR feature extraction module, which adopts a RSCSE module to extract SR features efficiently by applying spatial and channel squeeze-and-excitation (SCSE) concurrently with local feature fusion (LFF) concept to increase its representational ability. The proposed deblurring module is developed based on SCSE-based encoder-decoder CNN structure for extracting sharp features from the blurry LR images.

(iii) Extensive simulations are performed on publicly available RGB and real MS remote sensing datasets for various zooming factors. Training datasets for the proposed model are obtained by selecting more informative band images based on entropy and variance values. To preserve the spectral information in these MS datasets, we train and test SR models on individual spectral bands.

The rest of the chapter is organized as follows: Section 5.2 discusses prior art, including residual learning and upsampling methods. Section 5.3 describes the proposed method in detail. Experimental datasets and simulation results are discussed

in Section 5.4. Finally, Section 5.5 draws a few conclusions on the current work done.

5.2 Prior Art

5.2.1 Residual learning

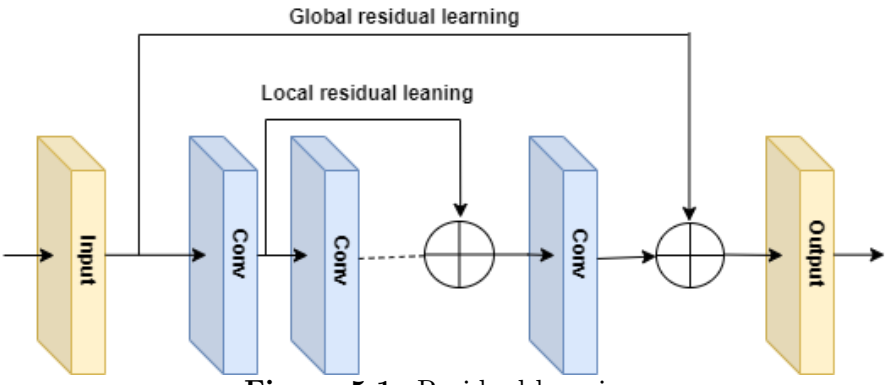


Figure 5.1: Residual learning.

It is observed that SR method widely uses residual learning [85, 99], which is shown in Fig. 5.1. When the DL network becomes deeper and complex, this learning strategy is employed mostly to mitigate vanishing gradients. They can be broadly divided into two categories: global and local residual learning.

(i) **Global residual learning:** Since image SR is an image-to-image translation problem in which the input image is closely correlated with the output image, global residual learning is used to learn only the residuals between them. Learning a complex transformation from one complete image to another is avoided in favor of learning only a residual map to recover the high-frequency information that are missing. The model complexity and learning difficulties are substantially reduced since the residuals are almost zero in the majority of regions. Therefore, it is extensively used by SR models [40, 45, 95].

(ii) **Local residual learning:** The local residual learning is very analogous to the residual learning that is used in ResNet [35]. It is utilized to mitigate the

degradation issue [96] caused by steadily growing network depths, minimise training difficulty, and increase learning ability. It is also often used for SR [57, 66, 132].

Practically, shortcut connections with element-wise addition are used to implement both techniques. The primary difference between the two approaches is that the former directly connects the input and output images, whereas the latter typically introduces multiple shortcuts between layers at different depths within the network.

5.2.2 Upsampling methods

5.2.2.1 Sub-pixel layer

Sub-pixel layer [87] is an end-to-end learnable upsampling layer used in image SR to increase the spatial resolution of an image by a factor of n (where n is the scaling factor). The technique involves performing convolution on an input image to produce feature maps that are equal to n^2 times channels in total, followed by rearranging the resulting feature map in a way that effectively upscales the image, as shown in Fig. 5.2. Suppose, the input feature map size is $h \times w$ (Fig. 5.2a) and the output size after convolution will be of $h \times w \times n^2 c$ (Fig. 5.2b). Using a reshaping operation, the final output will be of $nh \times nw \times c$ obtained from the convolved feature map. (Fig. 5.2c). The rearrangement is achieved by interleaving the values of the feature map in such a way that they form a grid of size $n \times n$. For example, if $n = 2$, the values in the feature map are interleaved such that every other pixel corresponds to a new pixel in the output image. By having a larger receptive filed, sub-pixel layer can incorporate more contextual information, which results in generating more realistic details.

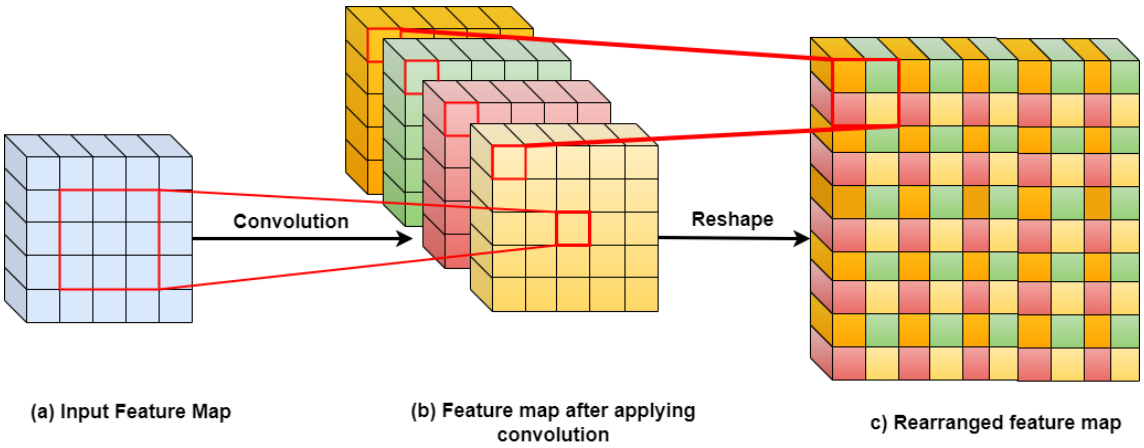


Figure 5.2: Sub-pixel layer.

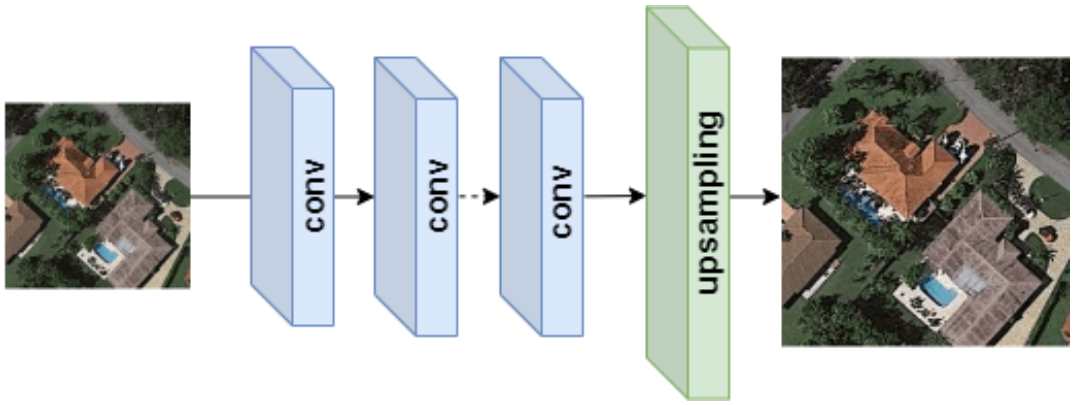


Figure 5.3: Post-upsampling SR.

5.2.2.2 Post-Upsampling SR

To restore high-quality details, the pre-upsampling SR technique [22], [45] first uses interpolation methods to upsample LR images to the target HR image, which is subsequently refined using DL networks. However, the cost and complexity of the pre-upsampling approach is very high because DL network is being applied on HR space. To address the limitations of pre-upsampling SR, a post-upsampling strategy is employed to fully use the DL technology to increase the spatial resolution automatically by introducing end-to-end learnable upsampling layers at the end of the DL models. The LR input images are directly fed into DL networks for feature extraction without enhancing resolution, and sub-pixel layers are added to upsample feature maps at the network's end. Since the computationally expensive feature extraction procedure only occurs in low-dimensional space and the resolution increases only at the end, computation and spatial complexity are considerably re-

duced. Therefore, this method has become one of the most commonly used methods in SR [52, 61, 101].

5.3 Proposed method

In order to reconstruct a clear and sharp HR remote sensing image, a joint dual-branch SR and deblur network (JSRDNet) is developed and applied on the blurred LR remote sensing image. The schematic diagram of the proposed JSRDNet is shown in Fig. 5.4. It consists of four main modules: (i) the SCSE-based SR feature extraction module for extracting highly representative feature maps; (ii) the deblurring module for obtaining deblurring feature maps; (iii) the SCSE-based gated fusion module to adaptively fuse the SR and deblurring feature maps by recalibrating them, and (iv) the upscaling and reconstruction module to reconstruct the final super-resolved image. In the following sections, each stage is elaborated in details:

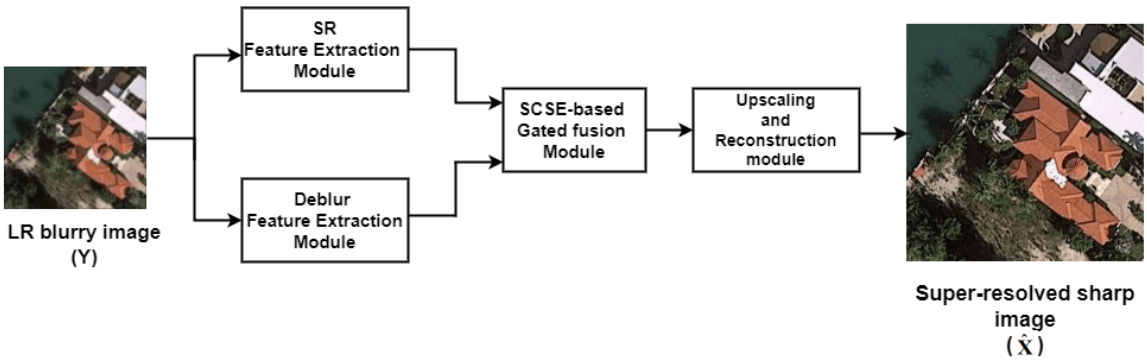


Figure 5.4: The schematic diagram of the proposed JSRDNet

5.3.1 SR feature extraction module

In this section, a fully trainable feature extraction module for SR is designed for RS images, which mainly consists of two major parts: shallow feature extraction module $\mathcal{F}_{SFE}$ and deep feature extraction module $\mathcal{F}^{n}_{RSCSE}$. Given a blurred LR image $\mathbf{Y} \in \mathbb{R}^{h \times w \times n}$ (n= 1 or 3, represents the number of bands), in our formulation, the SR feature map $\mathbf{F}_{SR}$ is obtained by the following steps:

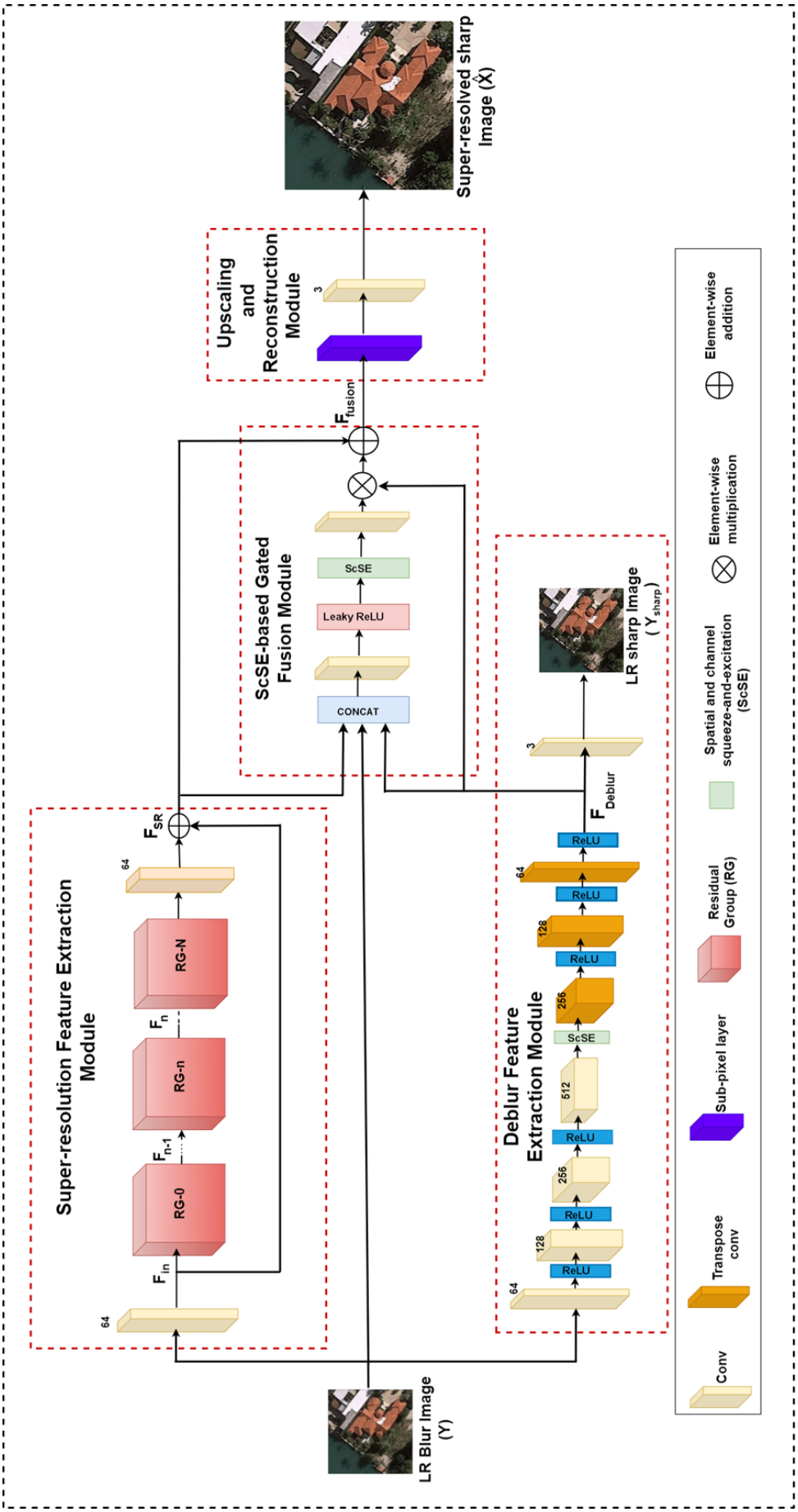


Figure 5.5: Framework of the proposed JSRDNet.

5.3.1.1 Shallow feature extraction (SFE)

Initially, a 3×3 convolution layer is used for extracting the shallow features $\mathbf{F}_{in}$ from the input $\mathbf{Y}$, as follows:

$$\mathbf{F}_{in} = \mathcal{F}_{SFE}(\mathbf{Y}) \in \mathbb{R}^{h \times w \times c}, \tag{5.1}$$

where $\mathcal{F}_{SFE}(\cdot)$ performs the convolution (conv) operation and c is the number of feature maps or channels.

5.3.1.2 Deep feature extraction

The extracted shallow feature $\mathbf{F_{in}}$ is fed to the input of a stacked RSCSE modules for extracting the deeper SR feature map $\mathbf{F_{SR}}$. DL super-resolution models are designed to learn hierarchical representations of the LR input images. The shallow feature module is used as an initial stage in this hierarchy for extracting basic patterns and structures, which may lack discriminative power. Given that RS images exhibit highly detailed and complex structures, the deeper feature extractor module is employed to extract more abstract and complex features. This hierarchical approach allows the SR model to effectively increase the representation of the input LR images, thereby enhancing discriminative power. The process is described as follows:

$$\mathbf{F}_{SR} = \mathcal{F}_{RIR}(\mathbf{F}_{in}) \in \mathbb{R}^{h \times w \times c}, \tag{5.2}$$

where $\mathcal{F}_{RIR}(\cdot)$ is a deep residual-in-residual (RIR) [132] structure and $\mathbf{F}_{SR}$ is the target SR feature map. As shown in Fig. 5.6, the network architecture realizing $\mathcal{F}_{RIR}$ function consists of N RG blocks and a residual long skip connection (LSC). Therefore, Eq. 5.2 is rewritten as follows after using LSC:

$$\begin{aligned} \mathbf{F}_{SR} &= \mathbf{W_{RG}} * (\mathcal{F}^{n}_{RG}(\mathcal{F}^{n-1}_{RG}(\cdots \mathcal{F}^{0}_{RG}(\mathbf{F}_{in}) \cdots))) + \mathbf{F}_{in} \\ &= \mathbf{W}_{RG} * \mathcal{F}^{N}_{RG}(\mathbf{F}_{in}) + \mathbf{F}_{in}, \end{aligned} \tag{5.3}$$

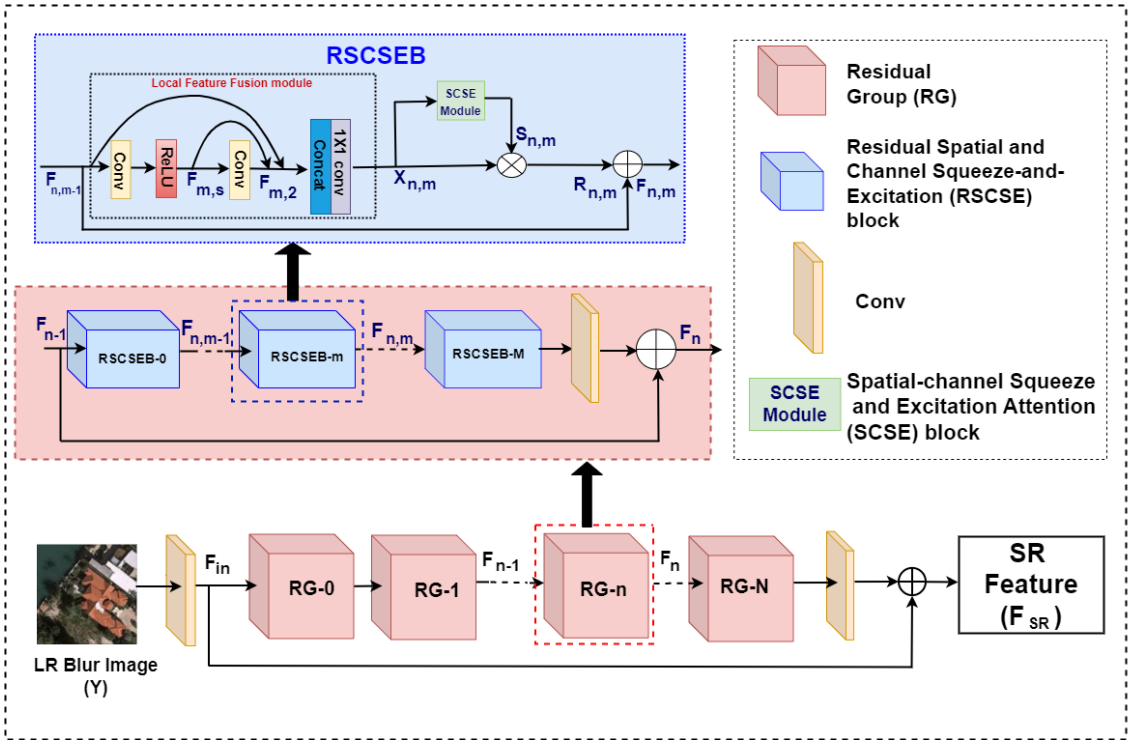


Figure 5.6: Proposed SR feature extraction module.

where $\mathcal{F}_{RIR}(\mathbf{F}_{in}) = \mathbf{W}_{RG}\mathcal{F}^{N}_{RG}(\mathbf{F}_{in}) + \mathbf{F}_{in}$, and $\mathbf{W}_{RG}$ corresponds to the weight matrix of a 3×3 conv layer that is applied to the feature map of the N^{th} RG block. Further, as shown in Fig. 5.6, each RG consists of M numbers of RSCSE blocks (RBs) with a short skip connection (SSC). The m^{th} RB block with SSC connection in n^{th} RG can be expressed as:

$$\begin{aligned}\mathbf{F}_n &= \mathbf{W}_{RB} * \mathcal{F}^{n}_{RB}(\mathcal{F}^{n,m-1}_{RB}(\cdots\mathcal{F}^{n,0}_{RB}(\mathbf{F}_{n-1})) + \mathbf{F_{n-1}} \\ &= \mathbf{W}_{RB} * \mathcal{F}^{M}_{RB}(\mathbf{F}_{n-1}) + \mathbf{F}_{n-1},\end{aligned} \tag{5.4}$$

where $\mathbf{F}_n$ and $\mathbf{F}_{n-1}$ represent the output and input of the n^{th} RG, respectively. The function of m^{th} RB is $\mathcal{F}^{m}_{RB}$ and the conv layer at the end of the n^{th} RG is represented by the weight matrix $\mathbf{W}_{RB}$. The local feature fusion (LFF) concept [135] is applied to the RB module for preserving the high-frequency information adaptively by using local dense features. The formulation of the m^{th} RB combined with the LFF and the SCSE modules in m^{th} RG can be expressed by:

$$\begin{aligned}\mathbf{F}_{n,m} &= \mathbf{F}_{n,m-1} + \mathcal{F}_{LFF\_SCSE}(\mathbf{F}_{n,m-1}) \\ &= \mathbf{F}_{n,m-1} + \mathbf{R}_{n,m},\end{aligned} \tag{5.5}$$

where $\mathcal{F}_{LFF\_SCSE}(\cdot)$ is the function for the combination of the LFF and the SCSE modules. $\mathbf{F}_{n,m}$ and $\mathbf{F}_{n,m-1}$ denote the output and input of the m^{th} RB in the n^{th} RG, respectively. Feature maps $\mathbf{R}_{n,m}$ are obtained as follows:

$$\begin{aligned}\mathbf{R}_{n,m} &= \mathbf{X}_{n,m} \cdot \mathbf{S}_{n,m} \\ &= \mathcal{F}_{LFF}(\mathbf{F}_{n,m-1}) \cdot \mathcal{F}_{SCSE}(\mathbf{X}_{n,m}),\end{aligned} \tag{5.6}$$

In LFF, first, the feature maps from the $(m-1)^{th}$ RB are introduced to the m^{th} RB through concatenation. Next, The 1×1 conv layer serves as a gating layer that reduces the dimensionality of the concatenated feature maps. It achieves this by applying a conv operation with a 1×1 kernel size. This operation effectively reduces the number of features in the concatenated feature maps, while preserving spatial information. This LFF operation is formulated as follows:

$$\mathbf{X}_{n,m} = \mathcal{H}_{m,LFF}(\mathbf{F}_{n,m-1}; \mathbf{F}_{m,s}; \mathbf{F}_{m,2}), \tag{5.7}$$

where $\mathcal{H}_{m,LFF}$ represents the operations of the 1×1 conv layer in the m^{th} RB. $\mathbf{F}_{n,m-1}$, $\mathbf{F}_{m,s}$, and $\mathbf{F}_{m,2}$ denote the outputs of the $(m-1)^{th}$ RB, feature component yield by ReLU function after first conv, and the second conv layers in the m^{th} RB, respectively.

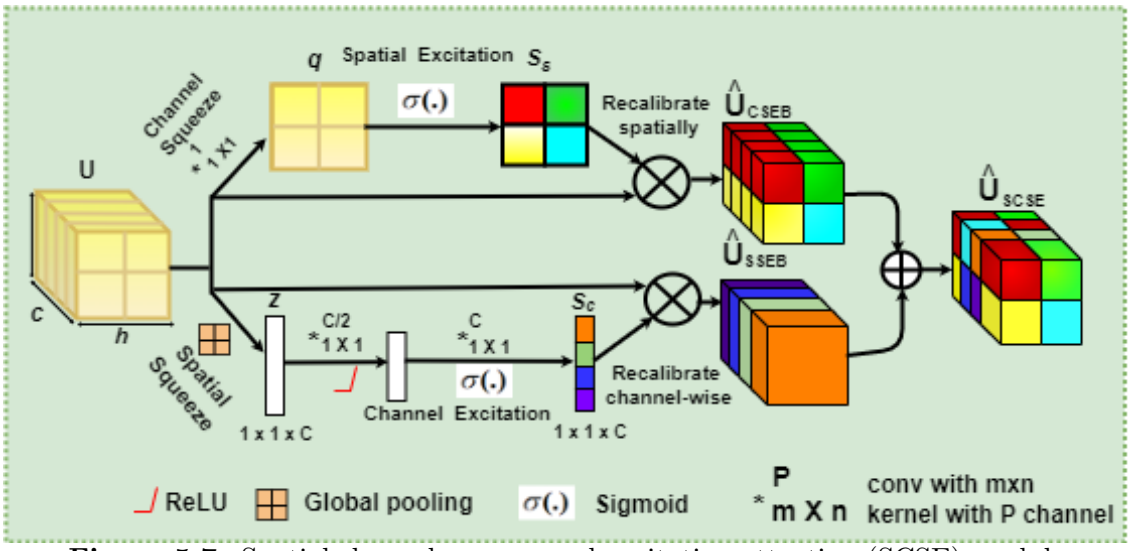


Figure 5.7: Spatial-channel squeeze and excitation attention (SCSE) module.

5.3.1.2.1 Spatial-channel squeeze and excitation attention (SCSE) block

Deep CNN has the ability to generate the features containing different kind of information both spatially and channel-wise, each of which contributes to recovery of the high-frequency details. Learning more significant features would improve the network's performance and representation power, leading the network to become more sensitive to higher contributing features. In view of this, a concurrent spatial and channel squeeze and excitation block (SCSE) [78] is introduced into the residual block (RB) by leveraging the interdependencies across channel and spatial features in order to adaptively recalibrate the representations of features. The architecture of the concurrent SCSE is illustrated in Fig. 5.7, which consists of two main components: spatial squeeze and channel excitation block (SSEB) and channel squeeze and spatial excitation block (CSEB). Let $\mathbf{U} \in \mathbb{R}^{h \times w \times c}$ be an input feature map is being applied to SCSE module, $\mathcal{F}_{SCSE}$ to generate output feature map, $\hat{\mathbf{U}}^{SCSE} \in \mathbb{R}^{h \times w \times c}$.

In SSEB module, the input feature maps $\mathbf{U} = [u_1, u_2, \cdot\cdot, u_i, \cdot\cdot, u_c] \in \mathbb{R}^{h \times w \times c}$, where $u_i \in \mathbb{R}^{h \times w}$ is transformed into recalibrated form by squeezing both spatially and exciting channel-wise. A spatial squeeze process $\mathcal{F}_{sq}$ is performed via global average pooling to obtain a scaler descriptor vector $\mathbf{z} = [z_1, z_2, \cdot, \cdot, \cdot, z_c] \in \mathbb{R}^{1 \times 1 \times c}$ with its c^{th} element, as follows:

$$z_c = \mathcal{F}_{sq}(u_c) = \frac{1}{h \times w} \sum_{i}^{h} \sum_{j}^{w} u_c(i, j), \tag{5.8}$$

By performing this process, the global spatial information is incorporated into the vector $\mathbf{z}$. In the channel excitation process $\mathcal{F}_{ce}$, the network learns channel dependencies to adaptively determine excitation or scaling factors $\mathbf{s}$. This operation is a gating mechanism that performs channel attention by employing two fully connected layers, ReLU operation $\delta(\cdot)$, followed by a sigmoid activation $\sigma(\cdot)$:

$$\mathbf{s}_c = \mathcal{F}_{ce}(z) = \sigma(\mathbf{W}_1 \delta(\mathbf{W}_2 \mathbf{z})), \tag{5.9}$$

where $\mathbf{W}_1 \in \mathbb{R}^{c \times \frac{c}{2}}$ and $\mathbf{W}_2 \in \mathbb{R}^{\frac{c}{2} \times c}$ represent weights of the two fully connected

layers. The activation functions are adaptively tuned by the network as it learns to abandon less significant channels and prioritize relevant ones. Finally, the output feature maps $\hat{\mathbf{U}}_{\mathbf{SSEB}}$ of SSEB module is obtained by mulitplying $\mathbf{U}$ by $\mathbf{s}_c$:

$$\hat{\mathbf{U}}_{SSEB} = \mathcal{F}_{SSEB}(U) = \mathbf{U} \cdot \mathcal{F}_{ce}F_{sq}(U) = \mathbf{U} \cdot \mathbf{s}_c, \tag{5.10}$$

In CSEB module, the feature map $\mathbf{U} = [u_{1,1}, u_{1,2}, .., u_{i,j}, .., u_{h,w}]$, where $u_{i,j} \in \mathbb{R}^{1\times1\times c}$ is recalibrated by squeezing channel-wise and exciting spatially. A channel squeeze operation ($\mathcal{F}_{cq}$) is achieved by employing 1×1 conv layer on the input feature map $\mathbf{U}$, i.e.: $q = \mathcal{F}_{cq}(U) = \mathbf{W}_{cq} * \mathbf{U}$. Here, $\mathbf{W}_{cq} \in \mathbb{R}^{1\times1\times c}$ is the weight and $q \in \mathbb{R}^{h\times w}$ is a projection tensor. The linear combination of all the channels c at each spatial location (i,j) is represented by each $q_{i,j}$. In the spatial excitation process $\mathcal{F}_{se}$, the sigmoid function $\sigma(\cdot)$ is used to excite $\mathbf{U}$ spatially by rescaling activations to [0, 1] i.e. $\mathbf{s}_s = \mathcal{F}_{se}(q) = \sigma(q)$. These activations aid in learning the network to pay attention on more important spatial locations while ignoring irrelevant ones. The final output feature map $\hat{\mathbf{U}}_{CSEB}$ of CSEB module is achieved by multiplying the input feature map $\mathbf{U}$ element-wise with $\mathbf{s}_s$ as shown:

$$\hat{\mathbf{U}}_{CSEB} = \mathcal{F}_{CSEB}(U) = \mathbf{U} \cdot \mathcal{F}_{cq}F_{se}(U) = \mathbf{U} \cdot \mathbf{s}_s, \tag{5.11}$$

Finally, the outputs of SSEB and CSEB modules are combined to obtain the target feature map $\hat{\mathbf{U}}_{SCSE}$ of the concurrent SCSE module $\mathcal{F}_{SCSE}$, as follows:

$$\hat{\mathbf{U}}_{SCSE} = \mathcal{F}_{SCSE}(U) = \hat{\mathbf{U}}_{SSEB} + \hat{\mathbf{U}}_{CSEB}. \tag{5.12}$$

Both channel and spatial rescalings increase the activation of a specific location (i, j, c) on the input feature map $\mathbf{U}$. With this recalibration, the network is compelled to learn feature maps that are more significant both spatially and channel-wise.

5.3.2 Deblurring module

A deblurring module is developed based on the encoder-decoder CNN structure to extract sharp features from the blurred LR image $\mathbf{Y}$. The architecture of the deblurring feature extraction module is shown in Fig. 5.5. It consists of an encoder, a SCSE module and a decoder, as follows:

$$\mathbf{F}_{Deblur} = \mathcal{F}_{decoder}(\mathcal{F}_{SCSE}(\mathcal{F}_{encoder}(\mathbf{Y}))) \tag{5.13}$$

The encoder is composed of four convolutional layers with increasing numbers of filters, followed by a SCSE module. The SCSE module is incorporated in order to make encoder feature maps more informative. The number of filters in the convolutional layers are 64, 128, 256, and 512. The spatial dimensions of the input are reduced by a factor of 2 in each layer due to the 3×3 kernel size and 1 padding of each convolution layer. The decoder uses three transposed convolutional layers with decreasing numbers of filters. The number of filters in the transposed convolutional layers in the decoder are 256, 128, and 64, respectively. As the kernel size of each transposed convolutional layer is 3×3, stride 1, and padding 1, which upsample the spatial dimensions of the encoded feature maps. The final layer of the decoder is a transposed convolutional layer with kernel size 3×3, stride 1, and padding 1, which produces the deblurred image with the same size as $\mathbf{Y}$. The activation function utilized throughout the model is ReLU for non-linearity.

5.3.3 SCSE-based gated fusion module

In the dual branch feature extraction process, the features extracted by the SR feature extraction module try to recover spatial details that are lost due to downsampling of the original scene information, while the features extracted by the deblur feature extraction module are especially the high-frequency features lost due to the blurring of the imaging sensor. A gating mechanism is used to ensure the significance of SR and deblur feature maps and accumulate the most relevant information

accordingly. The SCSE attention mechanism is integrated into a gated fusion module, which dynamically evaluates and selectively merges features from both modules. This approach not only preserves local and contextual information but also enhances the representational power of the fused feature map. Essentially, the SCSE-based gated module G_{scse} consists of three layers and the SCSE module: one concatenation layer, a 3×3 convolution layer ($conv_{3\times3}$), a Leaky ReLU, a SCSE module ($\mathcal{F}_{SCSE}$) and a 1×1 convolution layer ($conv_{1\times1}$), as shown in Fig. 5.8. A set of SR feature maps ($\mathbf{F}_{SR}$), deblur feature maps ($\mathbf{F}_{Deblur}$) and blurry LR input ($\mathbf{Y}$) are applied to G_{scse}. Subsequently, G_{scse} produces representational weight map to merge $\mathbf{F}_{SR}$ and $\mathbf{F}_{Deblur}$ effectively. The fused feature maps $\mathbf{F}_{\mathbf{fusion}}$ can be expressed by:

$$\begin{aligned}\mathbf{F}_{fusion} &= G_{scse}(\mathbf{F}_{SR}, \mathbf{F}_{Deblur}, Y) \otimes \mathbf{F}_{Deblur} + \mathbf{F}_{SR} \\ &= \mathbf{F}_G \otimes \mathbf{F}_{Deblur} + \mathbf{F}_{SR},\end{aligned} \tag{5.14}$$

where $\mathbf{F}_G = Conv_{1\times1}(\mathcal{F}_{SCSE}(LeakyReLU(conv_{3\times3}(concat(\mathbf{F}_{SR}, \mathbf{F}_{Deblur}, \mathbf{Y})))))$.

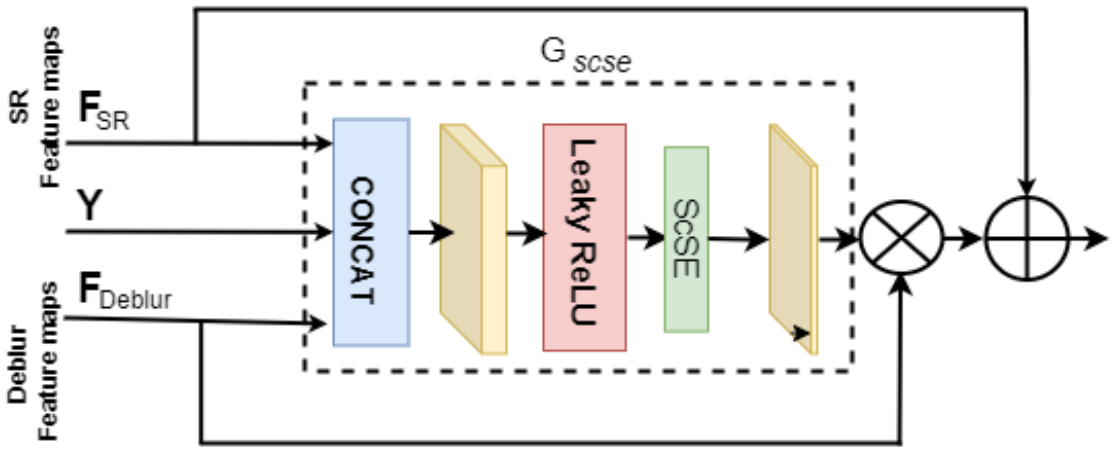


Figure 5.8: SCSE-based gated fusion module.

5.3.4 Upscaling and reconstruction module

The final reconstructed SR output $\hat{\mathbf{X}}$ is obtained as follows:

$$\hat{\mathbf{X}} = \mathcal{F}_{UP}(\mathcal{F}_{smooth}(\mathbf{F}_{fusion}) \in \mathbb{R}^{Uh\times Uw\times n}, \tag{5.15}$$

where n may be 3 or 1. $\mathcal{F}_{UP}(\cdot)$ denotes the upscaling and reconstruction module which consists of one sub-pixel conv followed by single conv layer with kernel size 3×3. Additionally, $\mathcal{F}_{smooth}(\cdot)$ represents a 3×3 conv operation that is applied on $\mathbf{F}_{fusion}$ for smoothing the refined fusion features.

5.3.5 Loss function

Our training data consists of N ground truth HR images $\{\mathbf{X}\}_{i=1}^{N}$ with corresponding blurred LR images $\{\mathbf{Y}\}_{i=1}^{N}$ as well as N interpolated LR images $\{\mathbf{Y}_{target}\}_{i=1}^{N}$, which are obtained by bicubic interpolation of $\mathbf{X}$. In order to train the proposed JSRDNet network, SR loss ($\mathcal{L}_{SR_{REC}}$) and deblurring loss ($\mathcal{L}_{Deblur}$) are jointly optimized, as follows:

$$\min_{\{\theta_1, \theta_2\}} \mathcal{L}_{SR_{REC}}(\hat{\mathbf{X}}, \mathbf{X}) + \alpha \mathcal{L}_{Deblur}(\mathbf{Y}_{sharp}, \mathbf{Y}_{target}), \tag{5.16}$$

Here, $\hat{\mathbf{X}}$ and $\mathbf{Y}_{sharp}$ are the predicted HR and LR images, respectively. α is used as a weight for balancing the two loss terms. The pixel-wise L_1 loss function is used optimized both $\mathcal{L}_{SR_{REC}}$ and $\mathcal{L}_{Deblur}$. These losses are defined as:

$$\mathcal{L}_{SR_{REC}}(\theta_1) = \frac{1}{N} \sum_{i}^{N} \left\| \hat{\mathbf{X}}_i - \mathbf{X}_i \right\|_1. \tag{5.17}$$

$$\mathcal{L}_{Deblur}(\theta_2) = \frac{1}{N} \sum_{i}^{N} \left\| \mathbf{Y}_{sharp_i} - \mathbf{Y}_{target_i} \right\|_1. \tag{5.18}$$

5.4 Results and Discussions

5.4.1 Dataset preparation

Simulations are done using RS images obtained from two publicly available databases, namely, PatternNet[1] and AID[2], as well as two real MS remote sensing datasets, LISS-IV[3] and LISS-III[3], collected from the NRSC data center. The training datasets from the PatternNet and AID include approximately 80% of the total data in each of the databases, selected randomly, having both textural and structural information. While the validation is carried out on another 10% of the total data selected randomly and not considered at all during the training.

In order to maintain uniformity in terms of number of images and size with the PatternNet images, an equal number of Region of Interest (RoI) images of size 256×256 are chosen from the original images (of size ≈ 10000×10000) from the LISS-III and LISS-IV datasets. For the preparation of training datasets for DL-based methods using LISS-III and LISS-IV, care has been taken to include images from all the spectral bands; the individual bands are selected based on their entropy and variance values. By computing the entropy and variance of each band image and ranking the bands accordingly, the top to medium-ranked entropy and variance values are chosen for the training dataset, while the ones with the lowest values are discarded. This approach ensures that the most informative band images are selected, which leads to more accurate and robust training of DL models compared to random selection. For testing purposes, a few images from the aforementioned datasets are randomly selected ensuring that they are neither part of the training set nor the validation, as shown in Fig. 5.9.

[1]PatternNet data: `https://sites.google.com/view/zhouwx/dataset`
[2]AID data: `https://captain-whu.github.io/AID/`
[3]NRSC Data Center: `https://uops.nrsc.gov.in/ImgeosUops/land.html`

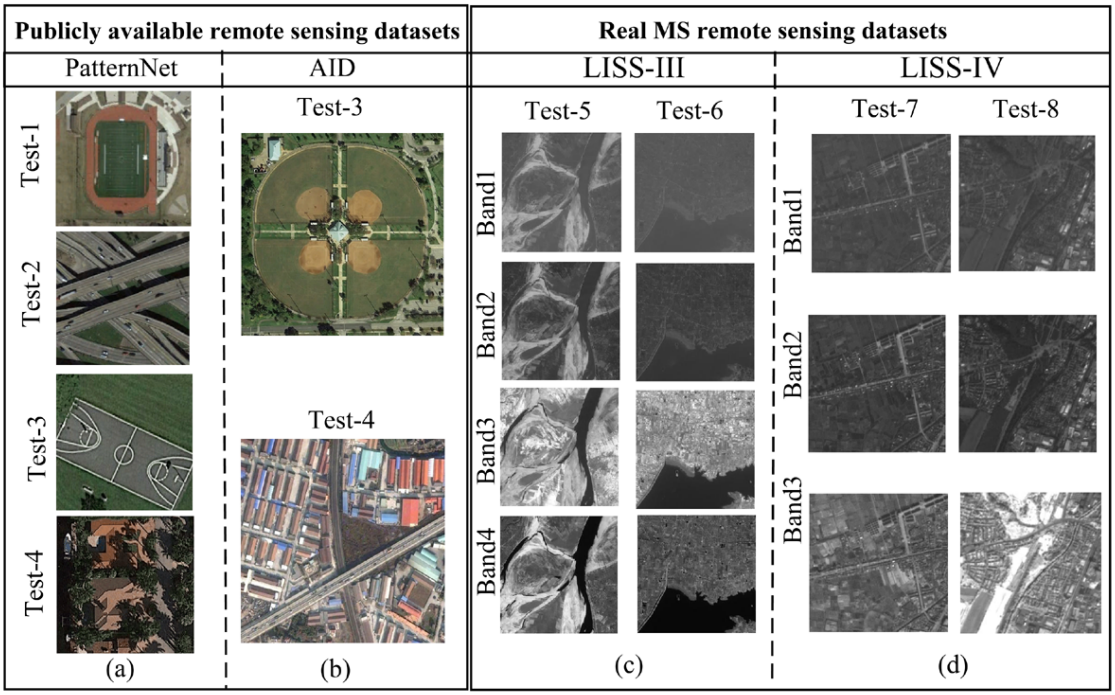


Figure 5.9: Column 1-2 from left to right: Publicly available remote sensing test images of (a) PatternNet,(b) AID; Column 3-4 from left to right: Real MS remote sensing test images of (c) LISS-III and (d) LISS-IV datasets for different bands.

5.4.2 Degradation method

For simulation of the proposed JSRDnet, blurred LR images ($\mathbf{Y}$) are generated from their corresponding HR images ($\mathbf{X}$). First, the HR image is blurred using a 3×3 Gaussian kernel (sigma=0.6). Next, the blurred image is downsampled by different scaling factors: 2,3 and 4. In order to train the deblurring network, we generate sharp downsampled (resized) images by applying bicubic kernel to the HR images $\mathbf{X}$.

5.4.3 Experimental settings

The training dataset is subjected to data augmentation, which includes rotation by 90°, horizontal and vertical flipping, random inversion, and channel shuffling. In each training batch, the proposed network takes 8 LR patches, each with dimensions of 96×96, as inputs. The SR module consists of 10 RG units, where each unit is

Table 5.1: Ablation study conducted on 'Test-1' image for 2× and 4× zooming factors.

Component	×2					×4				
Baseline (Channel attention)	✓					✓				
SCSE module	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓
LFF module	✗	✗	✓	✓	✓	✗	✗	✓	✓	✓
Deblur module	✗	✗	✗	✓	✓	✗	✗	✗	✓	✓
Gated module	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗
Gated module with SCSE	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓
PSNR (dB)/SSIM	33.76/ 0.944	34.05/ 0.945	34.16/ 0.946	35.17/ 0.955	**35.26/ 0.956**	28.34/ 0.768	28.53/ 0.772	28.62/ 0.779	28.90/ 0.790	**28.95/ 0.791**

composed of 20 RSCSE blocks and one convolutional layer. To enhance the network's capacity, each layer (excluding the input and upsampling layers) is allocated 64 channels. During the training process, the initial learning rate is set to 1×10^{-4} and is reduced to half for every 20% of the total iterations. The ADAM optimizer, with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-6}$, is employed to optimize the network's performance. The activation function and α used in the gated module with SCSE is the leaky rectified linear unit with a negative slope of 0.2 and 0.5, respectively. The implementation of the proposed network utilizes PyTorch 1.9.0 and CUDA toolkit 11.1, executed on a Linux server with an Intel Xeon CPU, Ubuntu 16.04 OS, 128 GB RAM, and NVIDIA Tesla V100 GP-GPU hardware. The model is trained for 15,000 iterations over approximately 2 days.

5.4.4 Ablation studies

In this section, the effectiveness of the main components of our proposed JSRDnet network is shown by conducting a series of ablation studies on PatternNet dataset using 'Test-1' as the testing image for 2× and 4× factors, as shown in Table 5.1. We have used the RCAN model as the baseline model for SR, which solely employed the channel attention (CA) module. When we replace CA with the SCSE module, the PSNR increases by 0.29 dB for 2× and 0.19 dB for 4× factors, respectively. It is evident that the features obtained from the SCSE module boost performance. Furthermore, as shown in Fig. 5.6, the LFF module is connected to the SCSE in such a way that they increase PSNR by 0.11 dB and 0.09 dB for 2× and 4× factors, respectively. We further explore the combined effect of debluring and SR

modules when their features are jointly learned using a gated module. It is observed from Table 5.1 that compared to only using the SR module, their combined effect improves PSNR by 1.01 dB for 2× factor, and by 0.26 dB for 4× factor. This comparison evidently shows the effectiveness of the joint deblurring and SR modules with learned gated module (JDSRGN) on the performance. Finally, when SCSE module is integrated into the gated module, the PSNR increases by 0.09 dB and 0.05 dB over JDSRGN for 2× and 4× factors, respectively.

Another ablation study is conducted as indicated in Eq. 5.16, by omitting the deblurring loss ($\mathcal{L}_{Deblur}$). In this configuration, both the SR feature extraction module and the deblur module are trained solely using super-resolution loss ($\mathcal{L}_{SR_{REC}}$). This resulted in a notable performance decrease, with PSNR values of 'Test-1' decreasing from 35.26 to 34.91 in the joint deblurring and SR task. This highlights the importance of incorporating $\mathcal{L}_{Deblur}$ to guide feature extraction, improving model performance.

5.4.5 Comparison with the state-of-the-art

To show the competitiveness of our proposed method with other state-art-the-art DL-based methods on the PatternNet and AID remote sensing datasets, the proposed method is compared with SRCNN [22], VDSR [45], SAN [18], MHAN [125], CFSRCNN [97], HSENet [53], RCAN-it [62], GFN [129] and DASR [107] and SRFormer [139]. SRFormer is transformer-based SR model for natural images. These networks are re-trained using the same datasets and environment as used by the proposed approach for fair comparison. Among them, GFN and DASR are designed for the joint SR and deblurring problem for natural scenes. MHAN and HSENet methods are the SR model for remote sensing images and the rest are excellent methods for natural images. The SR results obtained by the various methods are quantitatively evaluated using six metrics, including PSNR, SSIM, ERGAS [105], SAM [121], Q-index [113] and sCC [137]. Better reconstructed image quality is indicated by higher values of PSNR, SSIM, UIQI, sCC, and lower values of ERGAS and

SAM. The quantitative evaluation results on test images of PatternNet and AID using ×2, ×3 and ×3 upscaling factors are shown in Table 5.2 and 5.3. The results in the table are highlighted in bold, which denotes the best-performing methods. It is observed that the proposed JSRDNet always outperforms the other state-of-the art SR models on all scales. The proposed method achieves the maximum average PSNR for PatternNet test images when 'Test-1' and 'Test-2' are taken into account, with improvements of 1.72–4.8 dB, 0.72–3.62 dB, and 0.62–2.89 dB for ×2, ×3, and ×4 upscaling factors, respectively. In case of AID, JSRDNet has the highest

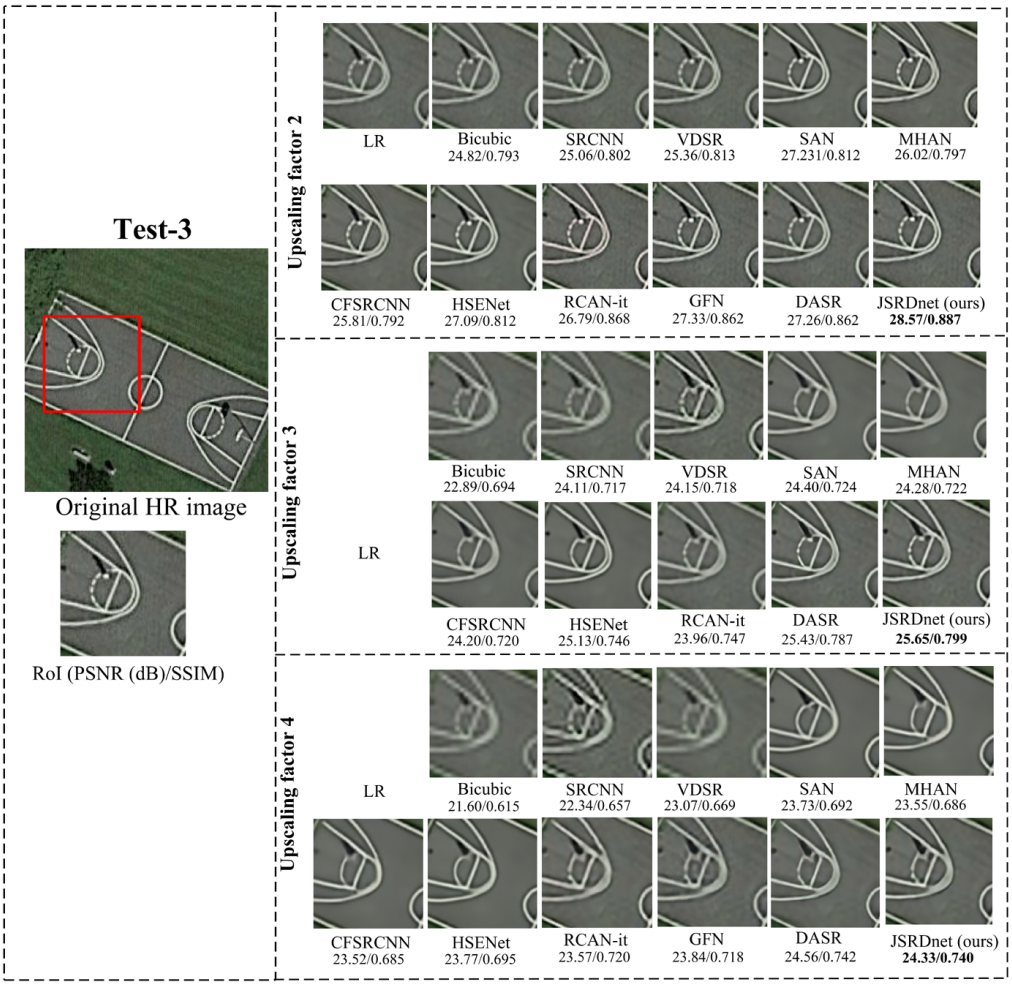


Figure 5.10: Visual comparisons of SR results for different methods on 'Test-3' for ×2, ×3 and ×4.

PSNR for 'Test-3' and 'Test-4' images, resulting in gains of 1.04–4.91 dB, 0.43–2.49 dB, and 0.25–2.23 dB in comparison to other methods for 2×, 3×, and 4× factors, respectively. Additionally, JSRDNet yields the highest scores for SSIM among all SR models for both datasets. Furthermore, the proposed method exhibits superior

Table 5.2: Quantitative comparison of test images of PattenNet dataset with different methods for different zooming factors. The best results are in bold.

Methods	Scale	Test-1						Test-2					
		PSNR (dB)↑	SSIM↑	ERGAS↓	SAM↓	UIQI↑	sCC↑	PSNR (dB)↑	SSIM↑	ERGAS↓	SAM↓	UIQI↑	sCC↑
Bicubic	×2	30.57	0.875	4.621	0.0840	0.8405	0.9787	33.60	0.935	2.625	0.0480	0.8584	0.9912
SRCNN [22]		30.85	0.885	4.480	0.0795	0.8544	0.9805	33.83	0.938	2.569	0.0449	0.8664	0.9922
VDSR [45]		31.35	0.896	4.230	0.0752	0.8705	0.9825	33.83	0.938	2.569	0.0450	0.8664	0.9922
SAN [18]		31.82	0.892	4.006	0.0729	0.8374	0.9840	34.53	0.933	2.367	0.0434	0.7783	0.9928
MHAN [125]		31.62	0.890	4.050	0.0735	0.8259	0.9835	34.24	0.935	2.445	0.0445	0.7940	0.9924
CFSRCNN [97]		31.53	0.884	4.196	0.0740	0.8193	0.9832	33.77	0.932	2.579	0.0469	0.7849	0.9915
HSENet [53]		31.80	0.888	4.016	0.0727	0.8292	0.9837	34.31	0.932	2.426	0.0440	0.7720	0.9926
RCAN-it [62]		33.76	0.944	3.198	0.0581	0.9252	0.9898	34.80	0.966	2.239	0.0420	0.9076	0.9934
SRFormer [139]		33.64	0.941	3.245	0.0573	0.9219	0.9896	36.18	0.966	1.947	0.0346	0.9107	0.9952
GFN [129]		33.91	0.939	3.150	0.0572	0.9158	0.9899	36.36	0.963	1.901	0.0347	0.8882	0.9953
DASR [107]		33.88	0.942	3.157	0.0573	0.9238	0.9900	36.44	0.970	1.886	0.0346	0.9171	0.9954
Proposed		**35.26**	**0.956**	**2.691**	**0.0562**	**0.9446**	**0.9926**	**38.51**	**0.972**	**1.492**	**0.0343**	**0.9360**	**0.9971**
Bicubic	×3	27.72	0.755	6.404	0.1168	0.6856	0.9580	30.56	0.8698	3.730	0.0684	0.7403	0.9822
SRCNN [22]		28.37	0.758	5.951	0.1084	0.6259	0.9635	30.82	0.870	3.621	0.0663	0.6132	0.9831
VDSR [45]		28.42	0.759	5.920	0.1078	0.6277	0.9639	30.90	0.870	3.590	0.0657	0.6149	0.9834
SAN [18]		28.72	0.760	5.719	0.1043	0.6410	0.9662	31.39	0.874	3.393	0.0622	0.6294	0.9851
MHAN [125]		28.57	0.756	5.819	0.1060	0.6342	0.9650	31.14	0.872	3.493	0.0640	0.6208	0.9843
CFSRCNN [97]		28.48	0.754	5.876	0.1071	0.6303	0.9644	31.00	0.871	3.546	0.0649	0.6175	0.9838
HSENet [53]		29.35	0.808	5.322	0.0966	0.7149	0.9710	31.81	0.882	3.236	0.0589	0.6438	0.9867
RCAN-it [62]		28.87	0.818	5.611	0.1019	0.7541	0.9683	31.14	0.903	3.474	0.0636	0.7688	0.9845
DASR [107]		30.52	0.853	4.411	0.0796	0.8285	0.9608	33.30	0.885	2.967	0.0541	0.8023	0.9615
Proposed		**31.08**	**0.876**	**4.350**	**0.0768**	**0.8412**	**0.9814**	**34.18**	**0.925**	**2.456**	**0.0556**	**0.8275**	**0.9922**
Bicubic	×4	25.94	0.651	7.871	0.1437	0.5445	0.9359	28.60	0.808	4.677	0.0858	0.619	0.9719
SRCNN [22]		26.26	0.696	7.591	0.1387	0.6254	0.9436	28.99	0.830	4.479	0.0811	0.6744	0.9760
VDSR [45]		26.79	0.685	7.146	0.1304	0.5231	0.9469	29.10	0.830	4.479	0.0811	0.6744	0.9760
SAN [18]		27.10	0.703	6.890	0.1258	0.5523	0.9507	30.19	0.840	3.896	0.0715	0.5457	0.9803
MHAN [125]		27.03	0.698	6.947	0.1268	0.5447	0.9498	29.96	0.838	4.001	0.0734	0.5518	0.9792
CFSRCNN [97]		27.02	0.697	6.958	0.1270	0.5425	0.9496	29.88	0.837	4.037	0.0741	0.5485	0.9789
HSENet [53]		27.46	0.717	6.619	0.1311	0.5734	0.9547	30.14	0.842	3.924	0.0718	0.5478	0.9802
RCAN-it [62]		28.34	0.763	5.971	0.1305	0.6915	0.9630	30.82	0.864	3.623	0.0661	0.7035	0.9835
SRFormer [139]		28.75	0.781	5.572	0.1245	0.7257	0.9695	31.21	0.876	3.367	0.0705	0.7245	0.9845
GFN [129]		28.31	0.762	5.991	0.1292	0.6764	0.9629	30.24	0.849	3.867	0.0708	0.6659	0.9807
DASR [107]		28.23	0.771	6.048	0.1280	0.6819	0.9626	30.96	**0.878**	3.499	0.0733	0.7111	0.9795
Proposed		**28.95**	**0.791**	**5.564**	**0.1230**	**0.7212**	**0.9680**	**31.49**	0.875	**3.352**	**0.0700**	**0.7255**	**0.9854**

Table 5.3: Quantitative comparison of test images of AID dataset with different methods for different zooming factors. The best results are in bold.

Methods	Scale	Test-3						Test-4					
		PSNR (dB)↑	SSIM↑	ERGAS↓	SAM↓	UIQI↑	sCC↑	PSNR (dB)↑	SSIM↑	ERGAS↓	SAM↓	UIQI↑	sCC↑
Bicubic	×2	30.88	0.950	4.077	0.0732	0.8470	0.9805	30.32	0.977	2.909	0.0547	0.9036	0.9864
SRCNN [22]		30.97	0.949	4.052	0.0725	0.7201	0.9806	29.99	0.959	3.023	0.0569	0.8679	0.9850
VDSR [45]		31.15	0.951	3.978	0.0711	0.7281	0.9812	30.26	0.961	2.929	0.0552	0.8735	0.9859
SAN [18]		32.13	0.961	3.555	0.0635	0.7571	0.9850	31.38	0.971	2.576	0.0485	0.8951	0.9890
MHAN [125]		31.86	0.960	3.667	0.0654	0.7542	0.9842	31.07	0.968	2.670	0.0502	0.8890	0.9883
CFSRCNN [97]		31.63	0.956	3.764	0.0672	0.7427	0.9832	30.74	0.965	2.772	0.0522	0.8829	0.9873
HSENet [53]		34.06	0.990	2.814	0.0507	0.9120	0.9905	32.72	0.990	2.205	0.0414	0.9507	0.9920
RCAN-it [62]		33.68	0.994	2.917	0.0516	0.9158	0.9905	32.35	0.995	2.286	0.0433	0.9566	0.9914
GFN [129]		29.19	0.912	4.979	0.0889	0.6478	0.9708	28.15	0.934	3.735	0.0704	0.8287	0.9770
DASR [107]		34.78	0.993	2.601	0.0467	0.9194	0.9920	34.15	0.994	1.871	0.0352	0.9579	0.9943
Proposed		**35.67**	**0.996**	**2.349**	**0.0422**	**0.9317**	**0.9934**	**35.35**	**0.997**	**1.630**	**0.0307**	**0.9674**	**0.9956**
Bicubic	×3	28.09	0.917	5.617	0.1011	0.7053	0.9622	27.72	0.925	6.404	0.1168	0.6856	0.9580
SRCNN [22]		28.43	0.925	5.407	0.0973	0.7262	0.9650	27.47	0.924	4.040	0.0761	0.8099	0.9732
VDSR [45]		29.01	0.912	5.083	0.0910	0.6043	0.9692	28.07	0.924	3.771	0.0710	0.7937	0.9764
SAN [18]		29.45	0.950	4.786	0.0864	0.7871	0.9765	28.07	0.957	3.768	0.0710	0.8719	0.9851
MHAN [125]		29.27	0.949	4.881	0.0882	0.7847	0.9710	27.78	0.955	3.893	0.0734	0.8691	0.9749
CFSRCNN [97]		29.20	0.950	4.915	0.0889	0.7863	0.9706	27.63	0.955	3.963	0.0747	0.8684	0.9740
HSENet [53]		29.95	0.952	4.530	0.0815	0.7921	0.9754	29.04	0.960	0.960	0.0635	0.8776	0.9812
RCAN-it [62]		29.43	0.943	4.815	0.0865	0.7634	0.9726	28.69	0.948	3.509	0.0659	0.8547	0.9802
DASR [107]		30.27	0.952	4.276	0.0768	0.8060	0.9781	29.67	0.945	3.047	0.0576	0.8871	0.9819
Proposed		**30.61**	**0.959**	**4.211**	**0.0757**	**0.8171**	**0.9788**	**30.19**	**0.967**	**2.953**	**0.0556**	**0.8943**	**0.9856**
Bicubic	×4	26.39	0.861	6.826	0.1232	0.5705	0.9434	25.06	0.844	5.333	0.1005	0.6711	0.9531
SRCNN [22]		26.51	0.838	6.741	0.1212	0.4347	0.9450	24.91	0.828	5.423	0.1022	0.6328	0.9510
VDSR [45]		26.82	0.848	6.508	0.1170	0.4506	0.9487	25.33	0.845	5.171	0.0975	0.6564	0.9553
SAN [18]		27.19	0.858	6.244	0.1124	0.4620	0.9527	25.85	0.865	4.866	0.0918	0.6895	0.9605
MHAN [125]		27.08	0.859	6.324	0.1138	0.4717	0.9515	25.76	0.863	4.922	0.0928	0.6857	0.9596
CFSRCNN [97]		27.01	0.855	6.374	0.1147	0.4644	0.9507	25.63	0.857	4.996	0.0942	0.6764	0.9583
HSENet [53]		27.52	0.914	5.966	0.1081	0.6667	0.9568	25.89	0.906	4.844	0.0913	0.7697	0.9609
RCAN-it [62]		27.99	0.911	5.682	0.1022	0.6694	0.9611	25.88	0.895	4.844	0.0894	0.7518	0.9633
GFN [129]		27.74	0.916	5.827	0.1054	0.6718	0.9588	26.18	0.907	4.687	0.0883	0.7707	0.9634
DASR [107]		28.26	0.917	5.504	0.0992	0.6760	0.9633	27.14	0.915	4.194	0.0790	0.7853	0.9710
Proposed		**28.43**	**0.918**	**5.405**	**0.0973**	**0.6896**	**0.9647**	**27.48**	**0.923**	**4.035**	**0.0761**	**0.8024**	**0.9730**

performance in ERGAS, SAM, Q-Index, and sCC values for 2×, 3×, and 4× zooming on both datasets by considering the same test images when compared to other methods. Although the proposed method clearly outperforms DASR at a zooming factor of 2, however it does not significantly outperform DASR at 3×, and 4× zooming factors. Since the DASR network is designed specifically to deal with more complex and unknown degradation model in a unsupervised way, it is thus expected to perform good at higher zooming factors when applied with less complex mixed degradations such as bicubic interpolation and the Gaussian kernel. Although the proposed method performs well in most of the cases, it is not particularly designed for handling complex and unknown degradations that can occur in real-world scenarios. This limitation can be overcome by making the proposed network degradation aware by incorporating mechanisms to learn and adopt the complex degradation characteristics including noise and anisotropic Gaussian kernels. Table 5.2 further reveals that that the quantitative metrics for the SRFormer network are lower than those achieved by our proposed method for 2× zooming, it delivers comparable results for 4× zooming. The competitive performance of SRFormer at 4× zooming may be due to its effective use of transformer-based architectures, which excel at maintaining intricate spatial information needed for higher magnification factors.

A visual comparison of SR results of different methods on 'Test-2' for ×2, ×3 and ×4 are shown on Fig. 5.10. From the original HR and SR images of the different models, a RoI is chosen and zoomed in to provide better visual comparison. While both SRCNN and VDSR tend to produce blurry SR images due to their limited use of features. Similarly, SAN, MHAN, and CFSRCNN, generate unnatural artifacts and smoothing effects in the SR results, causing a significant discrepancy between ground-truth HR and reconstructed SR images. HSENet and RCAN-it are able to restore texture details to some extent, but they are still blurry. Both GFN and DASR are capable of recovering realistic textures with less blurriness in the images. The proposed method outperforms both. As compared to other models, JSRDNet produces SR with more clarity and sharpness, which is closer to the desired HR image. The result reveal that JSRDNet effectively reconstructs the white line and textural information of the basketball court in the RoI, while the other methods

produce blurry images.

5.4.6 Results on real RS data

Here, we employ some real MS images captured by the LISS-III and LISS-IV satellite sensors to further validate the reconstruction quality of the proposed method. In order to preserve the spectral information, we apply the SR models to each spectral bands separately. Since most of the approaches are designed for processing 3-channel RGB images, the proposed method as well as other DL methods are modified to process the band images separately. The dataset preparation for the training are already discussed in Section 5.4.1. Here, JSRDNet is compared with some of the best performing SR methods, i.e. SAN, CFSRCNN, MHAN, HSENet, RCAN-it, GFN and DASR are compared with the proposed method on LISS-III and LISS-IV datasets. Table 5.4 shows the average results calculated over all the band images for each of the test images. On an average, the proposed method improves the PSNR of 'Test-5' images by 1.14–1.58 dB for a 2× upscaling and by 0.21–0.63 dB for a 4× upscaling. In the case of 'Test-7' , the proposed method offers the highest PSNR when compared to previous methods, with improvements of 1.51–4.64 dB and 0.28–2.03 dB for 2× and 4× factors, respectively. In some cases, the proposed method encounters substantial competition from DASR in terms of SSIM, ERGAS, SAM, UIQI and sCC. The proposed method specifically focuses on fixed Gaussian blur in such a way that the dedicated modules and mechanisms are designed to handle this degradation, and quantitative results show its competitiveness compared to DASR specifically in the case of Gaussian blur. The reason for DASR giving substantial competition is already mentioned in Section 5.4.5. While the proposed method may not have demonstrated significant improvements over DASR in terms of PSNR and SSIM metrics, a detailed error analysis was conducted to better understand these performance differences. The error statistics are calculated by comparing original and reconstructed images for both DASR and the proposed method at 2× and 4× upscaling factors. As shown in Fig. 5.11, the proposed method consistently exhibits lower error rates, particularly in mean absolute error (MAE), despite the PSNR

Table 5.4: Quantitative comparison of test images of LISS-III and LISS-IV dataset with different methods for ×2 and ×4 zooming factors. The best results are in bold.

LISS-III													
Methods	**Scale**	**Test-5**						**Test-6**					
		PSNR (dB)↑	**SSIM↑**	**ERGAS↓**	**SAM↓**	**UIQI↑**	**sCC↑**	**PSNR (dB)↑**	**SSIM↑**	**ERGAS↓**	**SAM↓**	**UIQI↑**	**sCC↑**
SAN [18]	×2	34.27	0.868	3.716	0.0680	0.6850	0.9514	33.39	0.830	3.422	0.0673	0.6552	0.8643
MHAN [125]		33.63	0.856	3.877	0.0705	0.6017	0.9429	33.63	0.856	3.877	0.0705	0.6017	0.9429
CFSRCNN [97]		33.55	0.857	3.912	0.0705	0.6030	0.9445	32.76	0.818	3.607	0.0698	0.6365	0.8479
HSENet [53]		33.63	0.856	3.877	0.0705	0.6017	0.9429	32.84	0.817	3.573	0.0697	0.6205	0.8447
RCAN-it [62]		34.32	0.866	3.696	0.0677	0.6760	0.9519	33.47	0.830	3.395	0.0669	0.6537	0.8666
GFN [129]		35.56	0.910	3.247	0.0577	0.7791	0.9709	34.87	0.889	2.933	0.0563	0.8272	0.9154
DASR [107]		34.71	0.910	3.354	0.0614	0.8275	0.9543	34.12	0.887	3.051	0.0600	0.8184	0.8858
Proposed		**35.85**	**0.918**	**3.143**	**0.0574**	**0.8428**	**0.9683**	**35.06**	**0.898**	**2.859**	**0.0563**	**0.8368**	**0.9104**
SAN [18]	×4	31.39	0.737	5.242	0.0960	0.3638	0.9057	30.60	0.668	4.774	0.0941	0.3160	0.7239
MHAN [125]		31.18	0.724	5.468	0.1000	0.3479	0.9020	30.48	0.658	4.885	0.0962	0.3402	0.7150
CFSRCNN [97]		31.09	0.723	5.496	0.1006	0.3448	0.8996	30.40	0.656	4.913	0.0968	0.3360	0.7081
HSENet [53]		31.15	0.733	5.313	0.0968	0.3459	0.9000	30.42	0.664	4.833	0.0945	0.2960	0.7144
RCAN-it [62]		30.88	0.755	5.345	0.0975	0.4603	0.8926	30.28	0.698	4.808	0.0944	0.4501	0.7077
GFN [129]		31.83	0.766	5.024	0.0909	0.4889	0.9245	31.08	0.708	4.539	0.0886	0.4973	0.7722
DASR [107]		31.81	**0.776**	**4.948**	0.0904	**0.5753**	0.9168	31.04	**0.719**	4.516	0.0888	**0.5097**	0.7584
Proposed		**32.02**	0.773	4.954	**0.0903**	0.5623	**0.9216**	**31.26**	0.716	**4.470**	**0.0881**	0.5051	**0.7690**
LISS-IV													
Methods		**Test-7**						**Test-8**					
SAN [18]	×2	35.29	0.905	2.783	0.0545	0.7754	0.9529	34.84	0.922	2.620	0.0503	0.8370	0.9799
MHAN [125]		35.35	0.903	2.762	0.0542	0.7734	0.9532	34.90	0.920	2.603	0.0501	0.8347	0.9797
CFSRCNN [97]		35.44	0.910	2.723	0.0526	0.7849	0.9557	34.87	0.924	2.603	0.0494	0.8415	0.9804
HSENet [53]		38.72	0.964	1.861	0.0344	0.9159	0.9814	38.25	0.972	1.784	0.0322	0.9414	0.9921
RCAN-it [62]		38.72	0.964	1.861	0.0344	0.9159	0.9814	38.25	0.972	1.784	0.0322	0.9414	0.9921
GFN [129]		38.90	0.962	1.827	0.0338	0.9208	0.9824	38.40	0.969	1.743	0.0318	0.9425	0.9922
DASR [107]		38.42	0.965	1.922	0.0376	0.9189	0.9770	37.12	0.972	1.981	0.0380	0.9463	0.9885
Proposed		**39.93**	**0.967**	**1.631**	**0.0320**	**0.9249**	**0.9839**	**39.60**	**0.976**	**1.523**	**0.0292**	**0.9541**	**0.9933**
SAN [18]	×4	30.51	0.750	4.843	0.0947	0.4361	0.8503	29.40	0.741	4.894	0.0936	0.5115	0.9309
MHAN [125]		30.52	0.752	4.834	0.0949	0.4461	0.8490	29.48	0.745	4.841	0.0930	0.5239	0.9319
CFSRCNN [97]		30.42	0.750	4.887	0.0959	0.4391	0.8454	29.33	0.741	4.919	0.0945	0.5161	0.9296
HSENet [53]		30.79	0.757	4.693	0.0913	0.4601	0.8625	29.51	0.748	4.813	0.0917	0.5298	0.9340
RCAN-it [62]		32.24	0.829	3.998	0.0769	0.6822	0.9074	30.87	0.820	4.140	0.0782	0.7271	0.9538
GFN [129]		32.11	0.813	4.064	0.0794	0.6567	0.9001	30.34	0.794	4.396	0.0842	0.6911	0.9460
DASR [107]		32.26	0.827	3.986	0.0780	**0.6798**	0.9040	30.87	**0.820**	4.132	0.0792	**0.7264**	0.9522
Proposed		**32.54**	**0.828**	**3.877**	**0.0760**	0.6762	**0.9086**	**31.07**	0.819	**4.058**	**0.0780**	0.7227	**0.9536**

and SSIM values not differing significantly. Furthermore, while direct comparison metrics such as PSNR and SSIM are commonly used, they are not able to fully capture all aspects of image quality. This error analysis offers deeper insights into the performance disparities between the proposed method and DASR.

Fig. 5.11 shows the visual comparison of reconstructed band 3 images of 'Test-8' using different methods for two upscaling factors ×2 and ×4. JSRDNet can generate comparable SR results with HR images (see the textures and edges of the 'Test-6' of LISS-IV in Fig. 5.11) as well as restore complex spatial information quite well. Fig. 5.12 also provides the visual results of 'Test-8' by stacking all bands in a false color representation. By considering the visual results of JSRDNet, we can conclude that it performs well while dealing with the complex real-world MS remote sensing data.

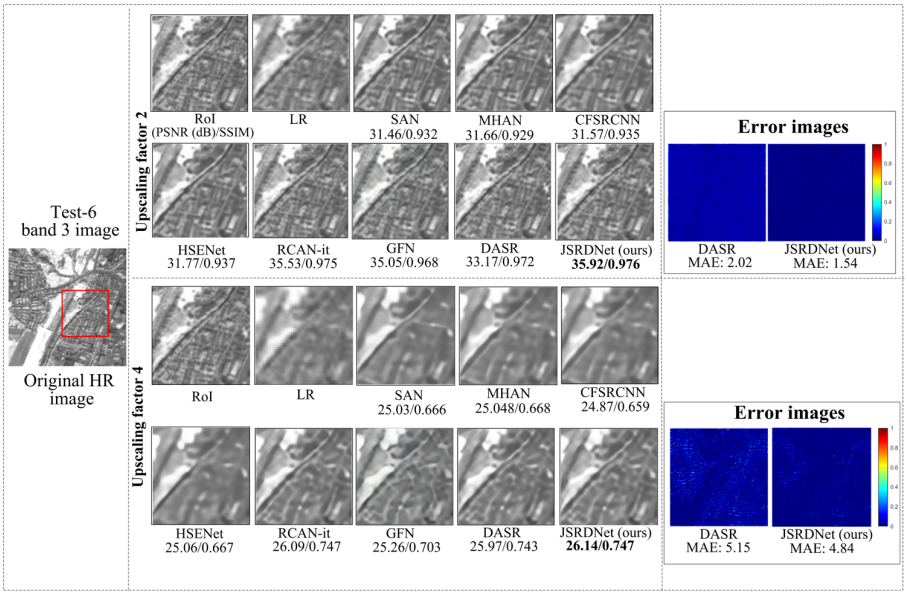


Figure 5.11: Visual results of different methods on band3 of Test-8 (LISS-IV) for $\times 2$ and $\times 4$ upscaling factors.

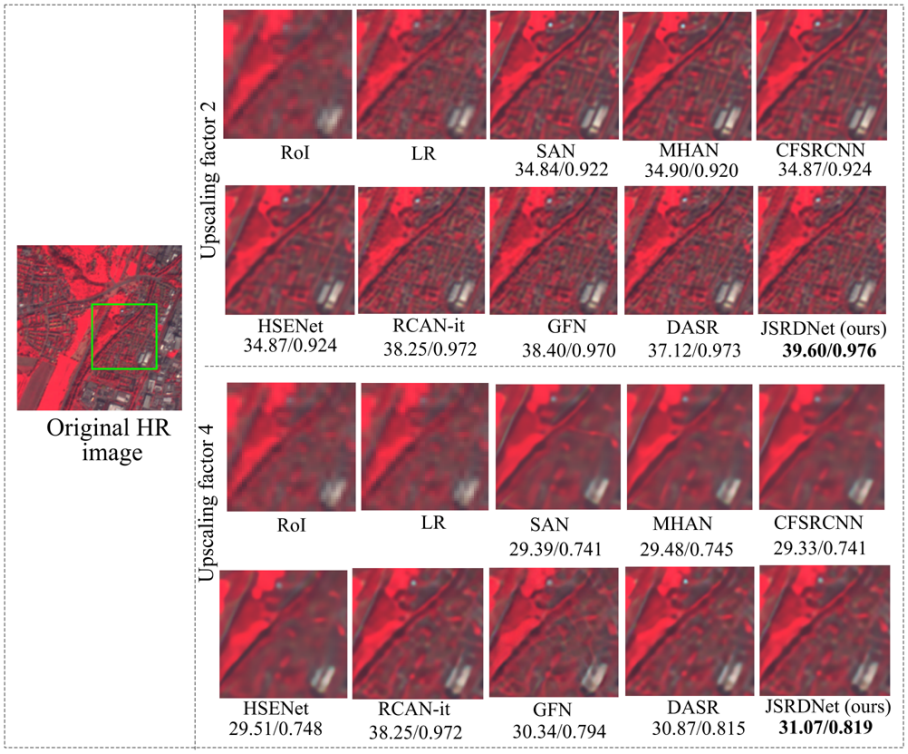


Figure 5.12: Visual results of different methods on Test-8 (LISS-IV) for $\times 2$ and $\times 4$ upscaling factors. Visual results are shown in false color RGB composition.

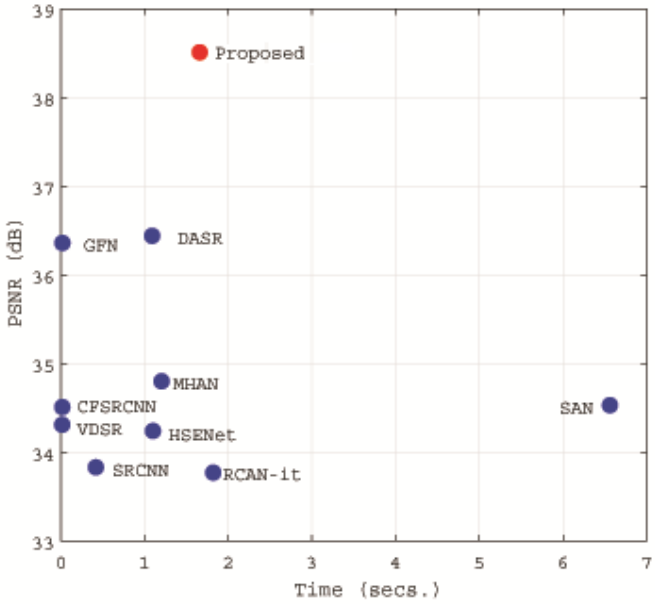


Figure 5.13: Comparisons of performance vs inference time.

5.4.7 Comparison on model size

DL model complexity is often estimated roughly based on the number of model parameters. We have conducted an in-depth analysis regarding the parameter count and its correlation with performance improvements. We consider the various configurations of RGs and RSCSE blocks, aiming to strike a balance between model complexity and reconstruction quality. Table 5.5 shows the results of these configurations for 'Test-1'. Selecting 15 RGs and 30 RSCSE blocks results in 41 million (M) parameters. However, a more optimized approach, such as employing 10 RGs and 20 RSCSE blocks, significantly reduces the parameter count to 21 M, while maintaining a commendable level of performance. Further reduction to 5 RGs and 10 RSCSE blocks yields a lighter model with 7M parameters. Although this choice slightly compromises the SR reconstruction quality, it offers a notable reduction in computational cost. Crucially, even with a moderate parameter count of 7 million, our proposed JSRDNet network consistently outperforms other methods in terms of PSNR for 'Test-2', as depicted in Table 5.2. We chose to stick with the configuration of 5 RGs and 10 RSCSE blocks, totaling 7M parameters, which is only slightly higher than DASR, as shown in Table 5.6. This increase is justified by the substantial performance improvement over DASR, balancing model complexity with image

reconstruction quality. The inference time taken by the proposed method is only 1.67 seconds, which is quite fast and practical for real-world applications. Moreover, the inference time tradeoff is a reasonable compromise for attaining high-quality reconstruction. Fig. 5.13 shows comparisons of inference time (in the GPU mode) and performance of different methods on the PatternNet test images for upscaling factor 2.

Table 5.5: Performance and parameters under different combinations of RGs and RSCSE blocks for 'Test-1'.

Number of RGs	Number of RSCSE blocks	Parameters (M)	PSNR (dB)
5	10	7	34.95
10	20	21	35.26
15	30	41	35.30

Table 5.6: Comparison of model efficiency of different SR methods for 'Test-2'.

Methods	SRCNN	VDSR	SAN	MHAN	CFSRCNN	RCAN-it	GFN	DASR	Proposed
Parameters	57K	667K	15.7M	13.8M	1.2M	16M	10M	5M	7M
PSNR	33.83	34.51	34.53	34.24	33.78	34.80	36.37	36.45	38.32

5.4.8 Application: Land cover classification

In order to interpret and analyse the areas included in the remote sensing image, land cover classification can be performed as a post processing step on the reconstructed SR image. To evaluate the effectiveness of the proposed JSRDNet, we perform supervised classification on SR results of different methods. We apply the support vector machine (SVM) algorithm to conduct supervised classification on the SR results of various methods, using the 'Test-2' and 'Test-8' for zooming factor 2. The classification and analysis of the results are performed using Envi classic 5.1. The RoI of 'Test-2' are divided into four categories: tree (red), building (green), water body(blue) and bare land(blue). Similarly, we divide RoI of 'Test-8' into three classes: building (red), bare land (green), road (blue). The proposed method has the most similar classified regions with the original image when compared to other SR methods, as shown in Table 5.7. The pixel counts per class for different methods are computed for both the images. The total pixel counts in both the images is 65,536. Results of different methods on 'Test-2' and 'Test-8' test images for zooming factor

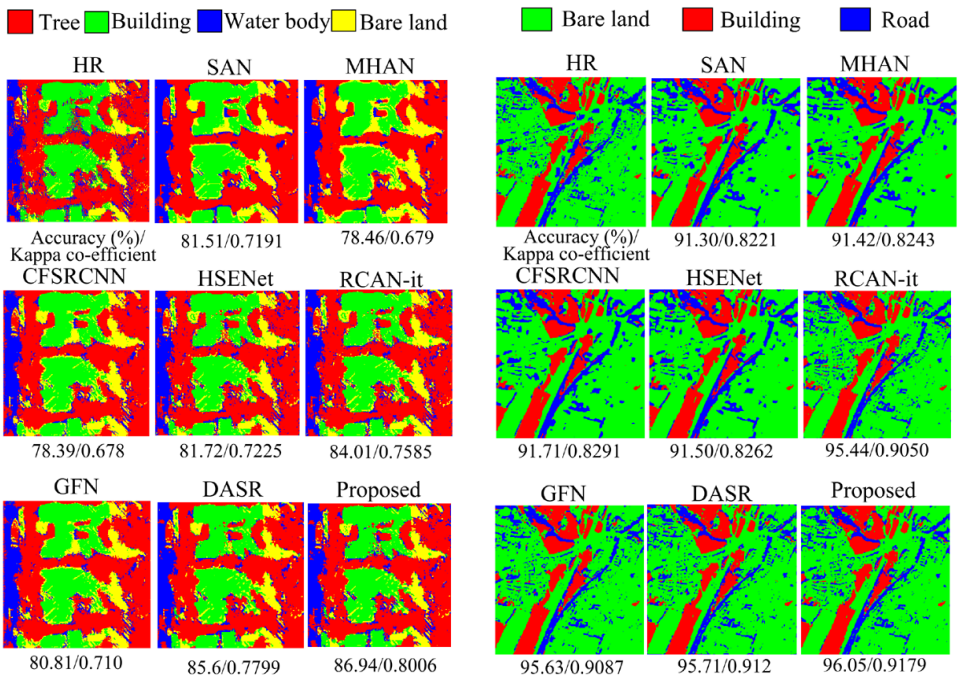
Table 5.7: Pixel count of each class of Test Image using unsupervised classification for different methods.

Patternnet 'Test-2'									
Classes	**HR**	**SAN**	**MHAN**	**CFSRCNN**	**HSENET**	**RCAN-it**	**GFN**	**DASR**	**Proposed**
	Pixels	Pixels	Pixels	Pixels	Pixels	Pixels	Pixels	Pixels	Pixels
Tree (Red)	32,954	32,142	30,463	30,369	31,851	31,139	32,395	31,247	32,181
Building (Green)	17,095	16,068	15,597	15,703	16,488	16,622	16,779	16,098	16,830
Water body (Blue)	9,931	8,766	8,415	8,433	8,977	9,742	8,625	8,570	9,768
Bare land (yellow)	9,931	8,766	8,415	8,433	8,977	9,762	8,625	8,570	9,768
LISS-IV 'Test-8'									
Classes	**HR**	**SAN**	**MHAN**	**CFSRCNN**	**HSENET**	**RCAN-it**	**GFN**	**DASR**	**Proposed**
	Pixels	Pixels	Pixels	Pixels	Pixels	Pixels	Pixels	Pixels	Pixels
Bare land (Green)	11,165	11,142	11,070	10,782	11,146	11,412	11,230	11,200	11,145
Building (red)	45,104	43,711	43,814	44,201	43,687	43,687	44,896	45,004	44,099
Road (blue)	9,267	10,683	10,652	10,553	10,703	9,776	9,410	9,332	9,307

2 are shown in Fig. 5.14, along with the overall accuracy and kappa coefficient. Here, the accuracy is referred as the percentage of correctly classified images out of the total number of images in a dataset and the kappa co-efficient is a statistical measure of agreement or performance for labelling images using classification models. It is used to measure how well the predicted labels from a classification model fit the actual labels. The ideal value of the kappa coefficient is 1, indicating perfect agreement. The HR image classification map is used to represent the ground truth, while the kappa coefficient is measured individually for the SR classification map generated by other approaches. It is observed that the proposed method exhibits superior performance in land cover classification compared to all other methods.

5.5 Conclusion

In this chapter, we have developed a joint dual-branch CNN network for recovering the sharp and clear HR images from LR remote sensing images degraded with Gaussian blur. The proposed network utilizes an attention-based gate module for fusing features adaptively from SR and deblurring feature extraction modules, allowing the network to handle deblurring and SR tasks jointly. We developed a RSCSE module to extract SR features efficiently by adopting SCSE and LFF modules in residual blocks in order to increase the representation ability of the proposed network. Further, deblurring module uses a simple SCSE-based encoder-decoder CNN module to extract sharp features for LR. Extensive simulations demonstrate that the pro-



(a) Test-2 of PatternNet **(b)** Test-8 of LISS-IV

Figure 5.14: Classification results of various methods on (a) PatternNet 'Test-2' image, and (b) LISS-IV 'Test-8' image. Overall accuracy and kappa co-efficient of each methods provided.

posed network outperforms other state-of-the-art approaches in terms of both visual analysis and objective criteria when recovering RS images. Furthermore, the proposed method also provides promising outcomes for land-cover classification, which is significant for RS applications.