Chapter 1

Introduction

1.1 Image super-resolution

An image sensor resolution is fixed by the pixel size or pixel count; it is defined as the ability of the sensor to detect the smallest object with sharp edge/line details and provide accurate scene reconstruction. Depending upon their spatial resolution, images can be either high-resolution (HR) or low-resolution (LR) images. HR images have the higher pixel density than LR images within an image, and capable of offering more details and information. Image sensors such as charge-coupled device (CCD) and CMOS are generally used for capturing the digital images. Although a number of imaging applications are being done with these sensors, their demand will outgrow due to the continuous advancement of modern digital display technology and related hardware equipment, and development of new applications of HR sensors [75]. Therefore, improving the spatial resolution of digital images has become an active research problem in image processing that will fit well and perform better with the state-of-the-art imaging systems.

Reducing pixel size, or increasing the number of pixels per unit area, is the most straightforward way to improve spatial resolution in sensors. However, when pixel size decreases, the amount of incoming light lowers, causing shot noise and increased diffraction effects in the sensor, resulting in significant image quality degradation. Another way to improve the spatial resolution is to enlarge the chip size that increases capacitance [47], which in turn hinders the speed of the charge transfer rate. Additionally, the cost and complexity of digital display systems are dependent on the sensor resolution, therefore the hardware cost becomes a restriction for many applications. To overcome these limitations, algorithmic-based post-processing techniques are quite preferable since they allow image resolution to be improved beyond

the sensor's physical limits. One promising algorithmic-based and cost-effective approach is image super-resolution (SR) which can recover HR images from LR input, making it a more affordable solution for many commercial applications. Recently, SR becomes a demanding research area in image processing and computer vision due to the abundance of high-resolution display systems in many vision-based applications, such as object detection, surveillance and monitoring, medical imaging, and remote sensing imaging, etc.

The remaining of the chapter is structured as follows: Section 1.2 provides an overview of single image Super-Resolution (SISR). In Section 1.3, the focus shifts to remote sensing SISR, covering topics such as the image degradation model, remote sensing (RS) data processing for SISR approaches, various SISR approaches in RS, and the challenges associated with SISR in RS. Section 1.4 explains the motivation behind the current work, while Section 1.5 describes on the scope of this study. Section 1.6 outlines the contributions of the thesis. Finally, Section 1.7 concludes the chapter by presenting an outline of the thesis.

1.2 Single Image Super-Resolution (SISR)

Generally, image SR is broadly grouped into two categories based on the number of LR input images: single-image SR (SISR) and multi-image SR (MISR). MISR obtains a sequence HR images or a single HR image from several LR input images acquired with different positions of the imaging sensor (camera) for the same scene. In MISR approach, it is possible to reconstruct a HR image or sequence of HR images if it is capable of taking multiple images of the same scene with sub-pixel misalignment. Since it is difficult to acquire such LR input images for the reconstruction, we can instead recover the HR image using only a single input LR image, which is know as SISR. Another limitation of MISR approaches is that they are time intensive since they need a complex registration procedure involving sub-pixel alignment of multiple LR input images. Therefore, the SISR technique is more practical for many applications, such as remote sensing [54, 125], video streaming [112] and

medical imaging [75].

1.3 Remote Sensing (RS) SISR

The need for remote sensing (RS) applications such as object identification, target recognition, military surveillance and land cover classification has been growing rapidly; HR images with fine features have become crucial in the recent years. However, RS images are often LR owing to limitations in optical sensors (due to limited aperture and diffraction, high speed imaging) and communication bandwidth, which are unsuitable for real-world image analysis. However, even though the most advanced satellite sensors are capable of acquiring high spatial resolution images, these sensors are very expensive. It is also difficult to upgrade embedded LR imaging sensors once the RS satellite have been deployed in the orbit. Since SR is a software-based solution for retaining HR images, it can be easily deployed in the ground station to post-process LR images obtained from the satellite. In RS, acquiring multiple images of the same scene for MISR is also a challenge due to cloud coverage, moving objects and other atmospheric disturbances, etc. Since the SISR technique offers a generic scheme to super-resolve any image sensor without the requirement for a satellite constellation, it is widely used in RS [20].

1.3.1 Image Degradation Model

In the real-life scenarios, an observed image is considered as an LR image ($\mathbf{Y}$) obtained from the natural HR counterpart ($\mathbf{X}$) after it encounters the following three degradations while being acquired: (i) the blurring (H), which is produced by atmospheric turbulence and optical sensor; (ii) the down-sampling (S) that sub samples the blurred image as per the resolution of the sensor and (iii) the additive noise (N). The basic image degradation model/process of the SISR framework is

shown in Fig. 1.1. Mathematically, it can be represented as follows:

$$\mathbf{Y} = SH\mathbf{X} + N \tag{1.1}$$

In most of the SISR models, the additive noise, N is considered ideally zero or has negligible effect (less severe) compared to the other two degradations. Therefore, the final image degradation process of $\mathbf{Y}$ can be expressed as: $\mathbf{Y} = SH\mathbf{X}$.

In RS domain, the blurring operator is referred to as the point spread function (PSF). Due to certain factors, such as the finite aperture of the lens, diffraction occurs, and the PSF deviates from a perfect impulse function and may be represented by a Bessel function. However, due to lens aberration or atmospheric turbulence, this function is well approximated by the Gaussian distribution, as follows:

$$G(x) = \frac{1}{\sigma\sqrt{(2\pi)}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \tag{1.2}$$

where σ and μ denote the standard deviation and mean, respectively. In order to generate an LR image with greater blurring effect, a larger kernel with higher σ and down-sampling factor is applied on the HR image. To provide an example, in Fig. 1.1, the LR image is obtained by applying a Gaussian kernel of size 7×7 with standard deviation 1.6 and then down scaled by the scaling factor of 3 on HR image. Ideally, the Gaussian mean is taken as zero.

The goal of SISR, as shown in Fig. 1.1, is to reverse the degradation process caused by the image degradation model in order to produce a super-resolved HR output image ($\mathbf{X}$) from its LR image counterpart ($\mathbf{Y}$). However, this process of solving Eq. 1.1 is ill-posed inverse problem because many HR solution are possible to obtain the same LR image. To tackle this ill-posed problem, most SISR algorithms should exploit additional information such as exemplar images or relevant a priori information to estimate the target super-resolved image.

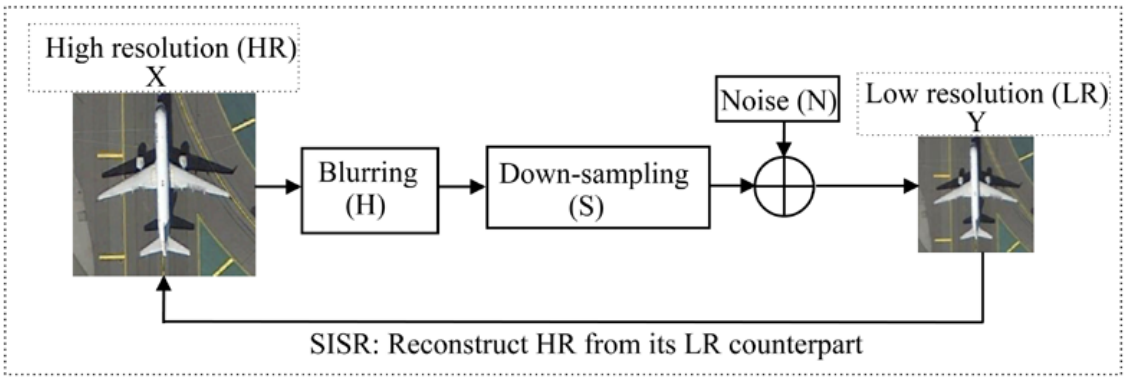


Figure 1.1: Image degradation and reconstruction framework of RS SISR.

1.3.2 RS Data Process for SISR Approach

The conventional SISR algorithms are designed to enhance LR grayscale images. In case of RGB RS images, first RGB image is transformed into the YC_bC_r space and then SISR algorithm is applied on the luminance channel Y only, which contains the high-frequency information. While, the two color channels C_b and C_r are upscaled to the target resolution by using the bicubic interpolation technique. Once the SR of the Y channel is done, they are again converted back to the RGB space as shown in Fig. 1.2. While real-world multispectral (MS) RS images comprise of several

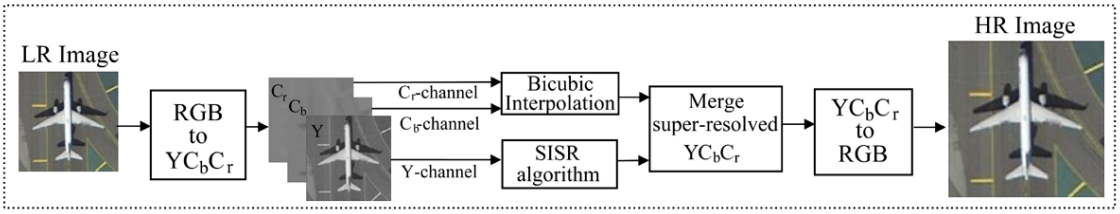
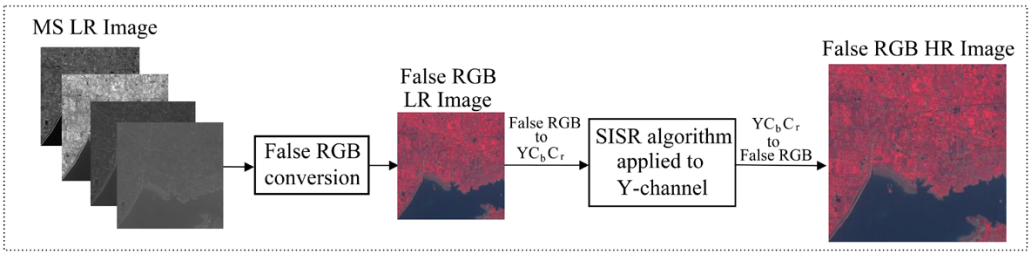
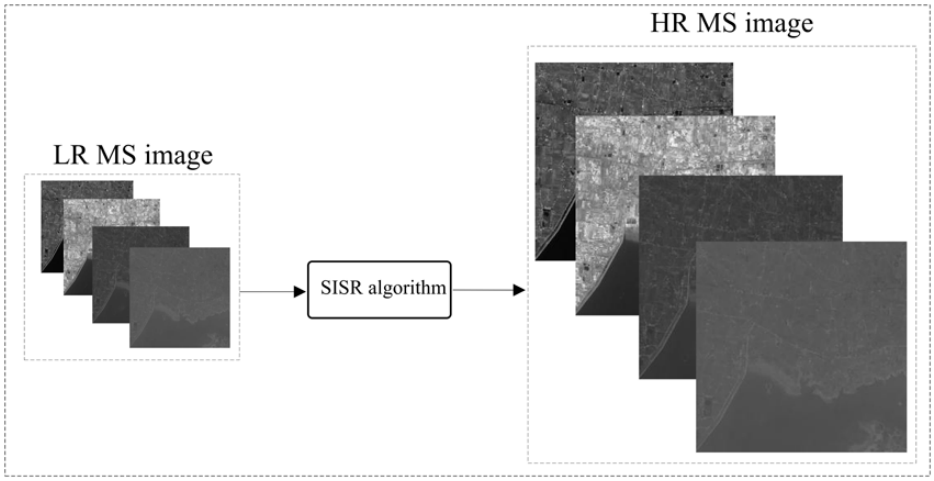


Figure 1.2: RGB RS data process for SISR approach.

bands (3–10). Two approaches can be adopted for super-resolving the MS images. One common approach is to apply SISR algorithm to the luminance channel by converting the MS image to false color RGB image [31], as shown in Fig. 1.3a. However, this approach is not effective if spectral information of MS images need to be maintained. Therefore, the simplest and most straightforward way is to apply the SISR algorithm on each band separately, in order to preserve their spectral properties, as shown in Fig. 1.3b.



(a)



(b)

Figure 1.3: Real-world MS remote sensing data process for SISR approach.

1.3.3 SISR approaches in RS

SISR problem can be addressed by using different approaches, which are broadly classified into three categories: interpolation-, reconstruction- and learning-based methods. Although interpolation-based SISR approaches are the fastest and the most simplest ones, they fail miserably to reconstruct non-smooth regions such as edges and textures, resulting in blurring and other ringing artifacts. In reconstruction-based approaches, prior information or constraints, such as gradient sparsity [92], total-variation (TV) sparsity [5, 70], and nonlocal sparsity [127], are explicitly incorporated to regularize the ill-posed inverse problem of SISR. However, these methods are often computationally intensive, and their efficacy gradually decreases as the upscaling factor increases.

In recent years, learning-based approaches have shown excellent performance over the aforementioned methods in terms of reconstruction quality as well as per-

ceptual accuracy. These approaches can effectively recover missing high-frequency information by learning the map between LR image patches and their corresponding HR counterparts. Example-based learning methods, such as the neighbor embedding [12], the random forest [85], the anchored neighborhood regression [98, 99], the sparse coding [25, 118, 122], and the deep learning (DL) [22, 45, 46, 52, 61, 132] are the most popular as they can predict the target HR patch by learning the correspondence between HR and LR patches using an external ideal dataset. More recently, sparse coding and DL-based approaches have obtained remarkable progress, which achieves state-of-the-art performance in the field of SISR. They have become emerging research topics in RS applications. Because of their extraction of high-level and complex features, these methods become very convenient for RS imagery. The basic mathematical formulation for SISR is detailed below:

1.3.3.1 Sparse Representation model

A signal is to be sparse if there are only a few non-zero elements present in the signal. In the conventional sparse representation model, as shown in Fig. 1.4, a signal $\mathbf{x} \in R^{N\times 1}$ can be represented as a sparse linear combination of K "atoms" from the overcomplete dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_K] \in R^{N\times K}$ with $N \ll K$, as follows:

$$\mathbf{x} = \mathbf{D}\boldsymbol{\alpha}, \tag{1.3}$$

where $\boldsymbol{\alpha} \in R^{K\times 1}$ is the sparse vector with a few non-zero weighting coefficients. The signal can be called as "s-sparse" if only s ($s \ll N$) non-zero entities are present in the column vector $\boldsymbol{\alpha} \in R^{K\times 1}$. The recovery of $\boldsymbol{\alpha}$ from $\mathbf{x}$ is an ill-posed problem that cannot provide an unique solution. By imposing *a priori* information or an appropriate regularizer, this problem can be accurately solved by obtaining sparse representation with the following ℓ_0-norm minimization problem:

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \qquad \text{such} \quad \text{that} \qquad \mathbf{x} = \mathbf{D}\boldsymbol{\alpha}, \tag{1.4}$$

where $\|.\|_0$ refers to the number of non-zero elements present in the vector. The optimization problem of Eq. 1.4 is a non-deterministic polynomial-time hard (NP-hard) problem and approximation of its solution is very difficult [6]. In the above optimization problem, replacing ℓ_0 norm by ℓ_1-norm provides a sufficiently sparse solution which is equivalent to solution obtained by ℓ_0-norm minimization provided atoms of the dictionary $\mathbf{D}$ are sufficiently incoherent [10]. Moreover, there are highly accurate off-the-shelf solvers for the ℓ_1-norm minimization. Therefore, ℓ_1-norm is used instead of ℓ_0-norm that converts the non-convex problem into a convex optimization problem, as follows:

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \qquad \text{such} \quad \text{that} \qquad \|\mathbf{D}\boldsymbol{\alpha} - \mathbf{x}\|_2^2 \leq \epsilon. \tag{1.5}$$

This basis pursuit denoising (BPDN) problem can be efficiently solved by using the recently developed fast ℓ_1-minimization algorithms such as the Fast Iterative Shrinkage Algorithm (FISTA) [7], LASSO [27], etc. The major challenge in a sparse

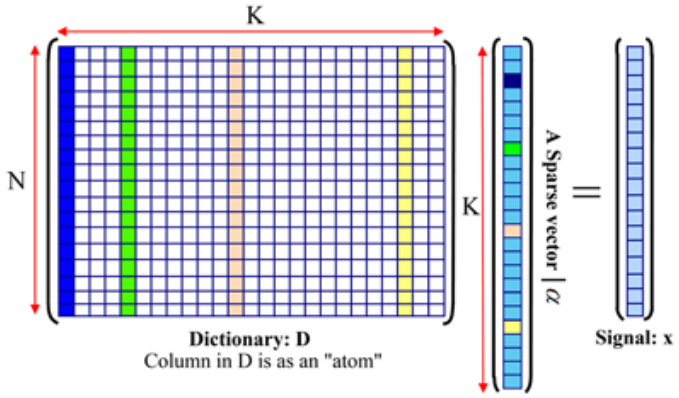


Figure 1.4: Sparse representation model.

representation-based model is how to choose the dictionary. Many pre-defined dictionaries, defined by transforms such as, the Gabor, the Fourier, the wavelet, and the discrete cosine transform (DCT) are present in the literature. These non-adaptive dictionaries, despite their simplicity and ease of computation, are incapable of sparsifying a given class of signals sufficiently. To overcome this issue, dictionary learning has received a lot of attention in the recent years [79, 102]. This approach, called the sparsity-based dictionary, involves using a few training signals from the signal class of interest to learn a dictionary. A dictionary learning algorithm utilizes the

training data matrix, $\mathbf{X}_h = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$ having N signals from the given signal class of interest to estimate the dictionary, $\mathbf{D}$, which can sufficiently sparsify all the training signals. Typically, a dictionary learning algorithm solves the following optimization problem:

$$\min_{\mathbf{A},\mathbf{D}} \|\mathbf{X}_h - \mathbf{D}\mathbf{A}\|_F^2, \tag{1.6}$$

where $\mathbf{A}$ indicates the sparse matrix obtained by concatenating the sparse representation vectors $\boldsymbol{\alpha}$ corresponding to each patch $\mathbf{x}_i$, i.e. $\mathbf{A} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_N]$ and $\|.\|_F$ indicates the Frobenius norm where it enforces unit ℓ_2-norm constraints to the columns of $\mathbf{D}$. Generally, the above optimization problem can be solved iteratively by performing a two-steps in each iteration. The dictionary can be randomly initialized at first, and then the following two processes are repeated many times [2, 79]:

(i) Step 1: Sparse representation:

$$\mathbf{A}^{(k+1)} = \arg\min_{\mathbf{A}} \|\mathbf{X}_h - \mathbf{D}^{(k)}\mathbf{A}^{(k)}\|_F^2 \tag{1.7}$$

(ii) Step 2: Dictionary update:

$$\mathbf{D}^{(k+1)} = \arg\min_{\mathbf{D}} \|\mathbf{X}_h - \mathbf{D}^{(k)}\mathbf{A}^{(k)}\|_F^2, \tag{1.8}$$

where k is the iteration number. Step 1 is the simple sparse representations problem, which can be carried out using a variety of sparse coding algorithms [103]. The dictionary is updated in step 2 to mitigate the representation error of step 1. Many dictionary learning algorithms, such as the method of optimal directions (MOD) [28], the K-singular value decomposition (K-SVD) [2] and simultaneous codeword optimization (SimCO) [19] can be used to perform step 2.

1.3.3.2 Sparse representation formulation for SISR:

In the sparse coding approach for SISR, image patches can be represented by sparse linear combination of elements from an appropriately chosen over-complete dictio-

nary. By considering this observation, patch-wise sparsity prior regularization is used to solve the ill-posed problem of Eq. 1.1. This process can be divided into two following steps: (i) Dictionary learning phase and (ii) Reconstruction phase.

(i) **Dictionary learning phase:** In order to learn the coupled dictionary $\mathbf{D}_c = [\mathbf{D}_h; \mathbf{D}_\ell]$, assume that there are N spatially correlated HR patches denoted by $\mathbf{X}_h = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ and N LR observed patches denoted by $\mathbf{Y}_\ell = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]$. We can extract high-frequency features vector corresponding to each patch vector $\mathbf{y}_i$ in $\mathbf{Y}_\ell$ by applying, for example, first-order gradient operators in x- and y-directions to the LR image $\mathbf{Y}$ before patch extraction. Since $\mathbf{Y}_\ell$ and $\mathbf{X}_h$ would share a common sparse coefficients matrix $\mathbf{A}$, a joint dictionary training approach can be adopted by enforcing the sparsity $\mathbf{A}$ on the concatenated data $\mathbf{Y_c} = [\mathbf{X}_h; \mathbf{Y}_\ell]$ using the following optimization problem:

$$\{\mathbf{D}_c, \mathbf{A}^*\} = \arg \min_{\{\mathbf{D}_c, \mathbf{A}\}} \|\mathbf{Y}_c - \mathbf{D}_c\mathbf{A}\|_F^2 \text{ subject to } \|\mathbf{a}_j\|_1 \leq T\,, \tag{1.9}$$

where a_j is the j^{th} column of $\mathbf{A}$ and T indicates the sparsity level. Different dictionary learning strategies such as [2, 118] and its variations are used for solving Eq. 3.5. One of the most frequently used technique is the K-SVD [2] approach that has the advantages of both simplicity and efficiency over other approaches [118].

(ii) **Reconstruction phase:** In order to reconstruct the HR image $\hat{\mathbf{X}}$, the sparse co-efficients can be obtained by using a LR dictionary ($\mathbf{D}_\ell$), which is trained using high-frequency feature patches extracted from the LR training images as mentioned in the sparse representation stage. To compute the sparse co-efficient of each LR image patch $\mathbf{y}_i = \mathbf{D}_\ell\boldsymbol{\alpha}_i$, the following minimization problem can be used:

$$\min_{\boldsymbol{\alpha}_i} \|\boldsymbol{\alpha}_i\|_1 \qquad \text{subject} \quad \text{to} \qquad \|\mathbf{D}_\ell\boldsymbol{\alpha}_i - \mathbf{y}_i\|_2^2 \leq \epsilon, \tag{1.10}$$

By using Lagrange multipliers, the above optimization problem can be refor-

mulated to:

$$\min_{\boldsymbol{\alpha}_i} \|\mathbf{D}_\ell \boldsymbol{\alpha}_i - \mathbf{y}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1, \tag{1.11}$$

where the regularization parameter λ is used to trade-off between sparsity of the solution and accuracy of the estimated output. The above optimization problem can be solved using the basic pursuit denoising problem [7, 27]. Since LR and HR patches share the same sparse coefficient, the desired HR image patch $\mathbf{x}_i$ can be obtained by:

$$\mathbf{x}_i = \mathbf{D}_h \boldsymbol{\alpha}_i \,, \tag{1.12}$$

Eventually, the HR image $\mathbf{X}_0$ is then reconstructed by tiling of all the reconstructed HR image patches. The basic sparse representation and dictionary learned-based SISR model for RS image is shown in Fig. 1.5. However, $\mathbf{X}_0$ may not exactly project onto the assumed image acquisition model i.e. $\mathbf{Y} = SH\mathbf{X}$. In order fit into the assumed imaging model, a global reconstruction constraint is applied on $\mathbf{X}_0$ by solving the following quadratic optimization problem using gradient descent method:

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \|SH\mathbf{X} - \mathbf{Y}\|_2^2 + C\|\mathbf{X} - \mathbf{X}_0\|_2^2, \tag{1.13}$$

where C is the regularization parameter. C is typically fixed experimentally and it indicates the trade-off between the fidelity of the reconstructed image $\hat{\mathbf{X}}$ and the proximity to the initial approximation $\mathbf{X}_0$. While still adhering to the reconstruction constraints, this image is as similar as possible to the initial SR $\mathbf{X}_0$ provided by sparsity.

1.3.3.3 Basic deep learning architecture

The recent years have witnessed significant advancements in the field of deep learning (DL). They are representation-learning methods with several layers of representations obtained by combining basic but non-linear modules that transform the net-

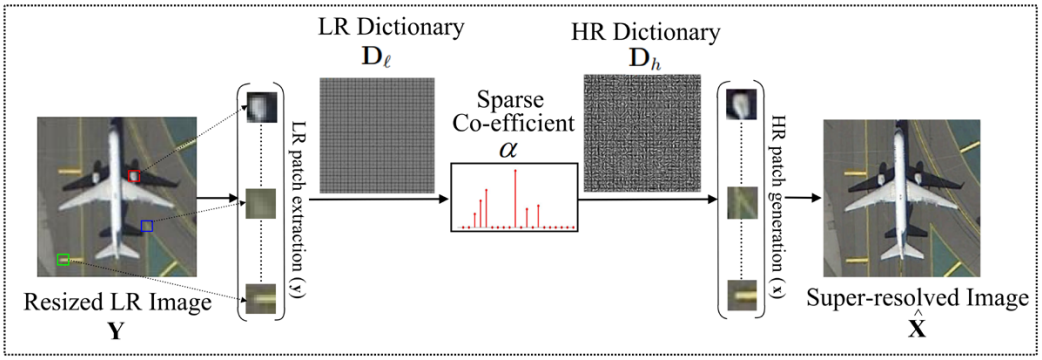


Figure 1.5: Sparse representation-based SISR model for RS image.

work's representation power from a simple to a higher and more abstract level [51]. In contrast to traditional learning-based methods, DL network does not employ any hand-crafted features. These models are capable of learning an optimal and informative set of features automatically through the training process for each task and dataset. In general, the basic feature extraction, hierarchical high-level feature extraction and decision making are all performed concurrently in a single DL-model training procedure. The fundamental process of deep learning was explained by LeCun *et al.* as follows: "Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction...Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer" [51]. The process of backpropagation involves determining the gradient of the error between the output and expected pattern of scores with regard to a multilayer stack of module weights. [51]. The chain rule is being used for backpropagation, as shown in Fig. 1.6.

The basic unit of DL network is the neural network (NN), that has the ability to estimate any continuous function [1]. In general, the N-layer NN consists of an input layer and $N-1$ hidden layers. In fully connected layers, neurons of a specific layer are fully connected to its adjacent layer, but not to each other. A fully-connected layer's forward pass involves a matrix multiplication, a bias offset, and an activation function. The most common activation functions are: tanh, sigmoid, and the rectified linear unit (ReLU) [33]. The initialization of parameters for the

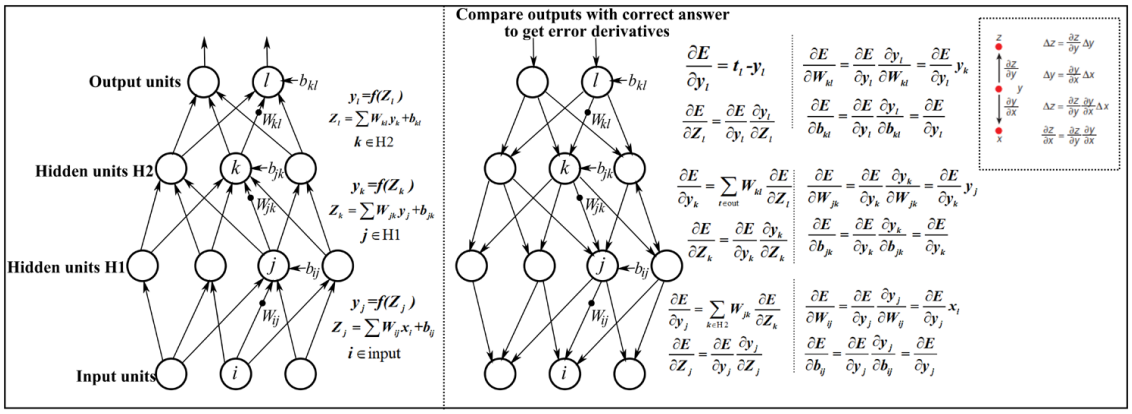


Figure 1.6: Backpropagation procedure.

first forward pass of a NN can be done in many ways, such as zero weights, random weights, etc. Let us consider a NN with one layer, the corresponding output $\mathbf{y}_j$ being obtained by differentiating with respect to the weights and bias parameters during backpropagation. Assume that we have a weight matrix $\mathbf{W}_{ij}$ of size m×n and an input vector $\mathbf{x}_i$ of size n, where m denotes the number of neurons in the layer. The bias vector $\mathbf{b}_{ij}$ has a length of m. The first layer's output can be described as follows:

$$\mathbf{z}_j = \mathbf{W}_{ij}\mathbf{x}_i + \mathbf{b}_{ij}; \;\; \mathbf{y}_j = f(\mathbf{z}_j), \tag{1.14}$$

The activation function is denoted by $f(\cdot)$, and it is applied to the vector $\mathbf{z}_j$ in a element-wise. Now, differentiating with respect to (w.r.t) the weights ($\mathbf{W}_{ij}$):

$$\frac{\partial \mathbf{y}_j}{\partial \mathbf{W}_{ij}} = \frac{\partial \mathbf{y}_j}{\partial \mathbf{z}_j}\frac{\partial \mathbf{z}_j}{\partial \mathbf{W}_{ij}}, \tag{1.15}$$

$\frac{\partial \mathbf{y}_j}{\partial \mathbf{z}_j}$ is the activation function's derivative with respect to $\mathbf{z}_j$, which is given by: $\frac{\partial \mathbf{y}_j}{\partial \mathbf{z}_j} = f'(\mathbf{y}_j)$. $\frac{\partial \mathbf{z}_j}{\partial \mathbf{W}_{ij}}$ can be computed as follows: $\frac{\partial \mathbf{z}_j}{\partial \mathbf{W}_{ij}} = \mathbf{x}_i$ (since $\mathbf{W}_{ij}\mathbf{x}_i$ is a linear operation). Therefore, the differentiation with respect to $\mathbf{W}_{ij}$ is:

$$\frac{\partial \mathbf{y}_j}{\partial \mathbf{W}_{ij}} = \frac{\partial \mathbf{y}_j}{\partial \mathbf{z}_j}\frac{\partial \mathbf{z}_j}{\partial \mathbf{W}_{ij}} = f'(\mathbf{y}_j)\,\mathbf{x}_i, \tag{1.16}$$

Next, $\mathbf{y}_j$ is differentiated w.r.t the bias ($\mathbf{b}_{ij}$):

$$\frac{\partial \mathbf{y}_j}{\partial \mathbf{b}_{ij}} = \frac{\partial \mathbf{y}_j}{\partial \mathbf{z}_j}\frac{\partial \mathbf{z}_j}{\partial \mathbf{b}_{ij}}, \tag{1.17}$$

$\frac{\partial \mathbf{z}_j}{\partial \mathbf{b}_{ij}}$ can be computed as follows: $\frac{\partial \mathbf{z}_j}{\partial \mathbf{b}_{ij}} = 1$ (since $\mathbf{b}_{ij}$ is added element-wise to the linear operation).

Next, the error signal or loss function is estimated by comparing the results of each forward pass to the ground truth. The network is learned by backpropagating this error signal and computing gradient to recalculate parameter weights for the following forward pass. The backpropagation algorithm is used to estimate the parameters (weights and biases) using local gradients with gradient descent. Backpropagation allows us to compute the gradients of the loss function with regard to the network parameters and then update the parameters in the reverse direction of these gradients to minimize the loss. Assuming that loss function of the network as $\mathbf{E}$. This loss function needs to be minimized w.r.t the network parameters. For each training set, the network's output ($\mathbf{y}_j$) is first calculated for a given input ($\mathbf{x}_i$). The local gradients, which represent the partial derivatives of the loss w.r.t each parameter, are then computed, as follows:

1. **Compute the output of the network:** For a given input $\mathbf{x}_i$, perform a forward pass through the network to determine the predicted output $\mathbf{y}_j$ against an expected output $\mathbf{t}_j$. The loss function is calculated by: $\mathbf{E} = \mathbf{t}_j - \mathbf{y}_j$.

2. **Calculate the local gradients using backpropagation:**

   (a) The partial derivative of the loss is calculated w.r.t the net input $\mathbf{z}_j$ of the output layer: $\frac{\partial \mathbf{E}}{\partial \mathbf{z}_j} = \frac{\partial \mathbf{E}}{\partial \mathbf{y}_j} \frac{\partial \mathbf{y}_j}{\partial \mathbf{z}_j}$.

   (b) The partial derivatives of the loss is computed w.r.t the weights $\mathbf{W}_{ij}$ and biases $\mathbf{b}_j$: $\frac{\partial \mathbf{E}}{\partial \mathbf{W}_{ij}} = \frac{\partial \mathbf{E}}{\partial \mathbf{z}_j} \frac{\partial \mathbf{z}_j}{\partial \mathbf{W}_{ij}}$ and $\frac{\partial \mathbf{E}}{\partial \mathbf{b}_{ij}} = \frac{\partial \mathbf{E}}{\partial \mathbf{z}_j} \frac{\partial \mathbf{z}_j}{\partial \mathbf{b}_{ij}}$.

3. **Update the parameters using gradient descent:** Each parameter's value is updated in the reverse direction of the gradient by subtracting the learning rate (ϵ) from the gradient: $\mathbf{W}_{ij} = \mathbf{W}_{ij} - \epsilon \frac{\partial \mathbf{E}}{\partial \mathbf{W}_{ij}}$ and $\mathbf{b}_{ij} = \mathbf{b}_{ij} - \epsilon \frac{\partial \mathbf{E}}{\partial \mathbf{b}_{ij}}$.

These three steps are repeated until the loss function converges, where the optimized parameters should correctly classify all the remaining test cases [51]. Both gradient

descent and backpropagation have their drawbacks: (i) NN suffers by vanishing and exploding gradients; (ii) computational complexity increases with the size of the network; (iii) convergence is slowed by local minima and plateaus; (iv) sensitivity of hyperparameters and overfitting. Here, vanishing gradients may occur in deep NNs when the gradients become small during backward propagation. This may hinder learning and affecting the updates of early layers. Contrary to that, exploding gradients occur when gradients become excessively large, leading to unstable training. Both can affect NN optimization, which can be handled by using techniques, like proper initialization, regularization techniques, or gradient clipping.

The convolutional neural network (CNN), a subset of deep NN is the simple deep feedforward network and can be trained and generalized with great ease. However, current CNNs suffer challenges such as as vulnerability to adversarial attacks and a lack of labelled training data, highlighting the need of ongoing research and improvement in these areas. They have showed excellent results in a broad range of machine learning problems [51]. The architecture of a CNN is hierarchical, where the output u_i of the subsequent layer is calculated for a given input signal u, as follows:

$$u_i = \rho \mathbf{W_i} u_{i-1}, \tag{1.18}$$

Here $\mathbf{W}_i$ and ρ indicate the weights and the non-linearity or the activation, respectively. $\mathbf{W}_i$ and the convolution layers are considered as a set of convolutional filters and filter maps, respectively. Therefore, each layer can be expressed as a sum of previous layer's convolutions:

$$u_i(m, n_i) = \Big(\rho \sum_{n=1} (u_{i-1}(., n) * \mathbf{W}_{i,n_i}(., n)(m))\Big), \tag{1.19}$$

Here $*$ is the discrete convolution operator. The mathematical formulation of convolution operation is given as follows:

$$(f * g)(x) = \sum_{k=-\infty}^{\infty} (f(k)g(x-k)), \tag{1.20}$$

A CNN defines an extremely non-convex optimization problem. Therefore, the weights $\mathbf{W}_i$ are commonly learned by stochastic gradient descent and the backpropagation algorithm. The basic architecture of CNN is shown in Fig. 1.7.

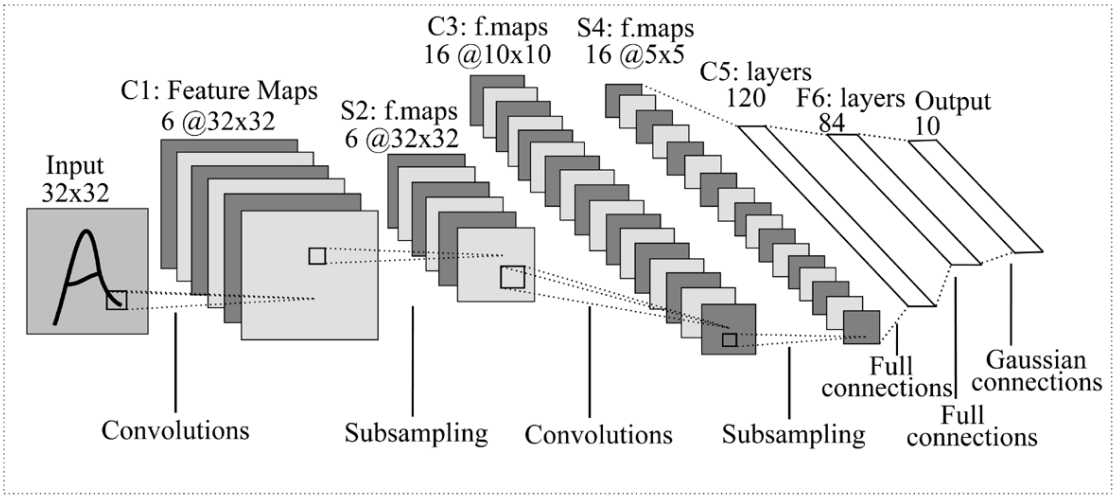


Figure 1.7: The basic architecture of CNN [50].

1.3.3.4 Deep learning architecture for SISR:

Due to the rapid increase in computational power, and availability of big data, DL-based networks are getting unprecedented attention and success in various image processing and computer vision applications, such as image classification [34, 48], image denoising [55], face recognition [93] and object detection [73]. In order to solve SISR problems that are ill-posed and non-convex, traditional techniques, such as the neighbor embedding [12], the random forest [85], the anchored neighborhood regression [98, 99] the sparse coding [25, 118, 122] are typically used. However, these methods use handcrafted features, have the problem of utilizing approximations, like convexity, which prevents them from having an optimal solution. With the emergence of deep learning, features are automatically learned from the raw image data, and effectively learn an end-to-end non-linear mapping between the original HR and degraded LR images, which are considered the prior information. CNN have been shown promising results in image processing and computer vision domain. In SISR problem, deep CNN aims to extract the most informative features in order to minimize the loss between the estimated SR image and the original (natural) HR,

and therefore, obtain remarkable results. An HR approximation $\hat{\mathbf{X}}$ of the original HR image $\mathbf{X}$ can be recovered from the observed LR image $\mathbf{Y}$ by using the following equation:

$$\hat{\mathbf{X}} = \mathcal{F}(\mathbf{Y}; \theta), \tag{1.21}$$

where θ stands for the parameters of $\mathcal{F}$, which is the SR model. The objective of SR model is as follows:

$$\theta = \underset{\theta}{\operatorname{argmin}} \mathcal{L}(\hat{\mathbf{X}}, \mathbf{X}) + \lambda\Phi(\theta). \tag{1.22}$$

where $\mathcal{L}(\hat{\mathbf{X}}, \mathbf{X})$ is the loss function between $\hat{\mathbf{X}}$ and $\mathbf{X}$. $\Phi(\theta)$ and θ are the regularization term and tradeoff parameter, respectively. Several loss functions are commonly used in SISR networks. Here are some types of loss functions used in SISR networks:

(i) **Mean-squared error (MSE):** MSE is a commonly used loss function in CNN-based SISR network. It calculates the average squared difference between the predicted HR image and the ground truth. MSE loss encourages the network to reduce pixel-wise differences and can result in smooth outputs. However, it may fail to capture perceptual details and textures.

(ii) **Charbonnier loss (ℓ_1 loss with perceptual blur):** It is an ℓ_1-based loss function that provides perceptual blur to the image. It helps to reduce artifacts and preserves high-frequency features in the reconstructed images.

(iii) **Adversarial loss (GAN loss):** Adversarial loss is a commonly used loss function in a generative adversarial network (GAN) network. A discriminator network differentiates generated and actual HR images in adversarial loss. The generator network generates images that the discriminator cannot distinguish from real ones. The network is encouraged to produce realistic and visually appealing outputs through adversarial loss.

The selection of the loss function is determined by the desired SR output characteristics and the task-specific requirements. Due to its simplicity and ease of optimization, MSE loss is frequently utilised, however it may result in over-smoothing.

The preservation of details and textures is improved using perceptual loss functions, like Charbonnier loss and adversarial loss, which encourage visually pleasing results. Adversarial loss can improve the visual quality even further by aligning the generated images to the distribution of real HR images.

The impact of the loss function on convergence varies depending on the problem and network architecture. The network is guided by different loss functions to optimize different aspects of the SR problem. The impact of the loss function on convergence is multi-fold, affecting the optimization direction, gradient calculation, and model behavior. The loss function needs to be chosen carefully to enhance convergence, model performance, and alignment with SR image reconstruction quality.

1.3.4 Challenges of SISR in RS

The SISR technique has the ability to overcome the physical limitation of the sensor during image acquisition as well as challenges in the reconstruction of the super-resolved image at the subpixel level. However, there are certain challenges that need to be overcome when working with remotely sensed images. These challenges are outlined below:

(i) **Modeling of SISR problem**: During the acquisition process, RS images are frequently vulnerable to several degradations. So, the SISR model needs to be flexible and well defined.

(ii) **Selection of dataset:** The selection of dataset is very much essential as distinct image features need to be exploited from the dataset at the time of dictionary and DL network training. Small databases are often inadequate for exploiting the relevant image features, while large databases increase computational time.

(iii) **Selection of regularization parameters:** SISR approaches such as sparse representation-based approach heavily rely on the selection of regularization

parameters. It has significant impact on the algorithm performance and is dependent on image types. The proper regularization parameters need to be selected for RS images.

(iv) **Computational complexity:** Sparse representation-based SR methods often involve addressing optimization problems such as sparse coding or dictionary learning, which may be computationally costly. As the size of the input image increases, so does its computational complexity, making real-time processing of large-scale RS images highly complex.

(v) **Handling complex edge structures:** RS images may contain complex textural and structural features, such as edges, or non-linear boundaries. Sparse representation-based SR algorithms often encounter difficulties in effectively preserving and reconstructing these complex edge structures, as they may require more sophisticated regularization terms or edge preserving priors. Incorporating additional priors specifically used for preserving complex edge information is necessary to achieve accurate HR reconstructions.

(vi) **Efficient architecture design:** Architectures of DL models for SISR needs to be computationally efficient to process RS images in a reasonable time frame. It is critical to design architectures that effectively trade off computational complexity and performance.

(vii) **Trade-off between computational complexity and performance:** DL-based SISR models with more parameters and complex architectures tend to achieve better performance. However, this comes at the cost of increased computational complexity and resource requirements. It might be difficult to find an optimal trade-off between model complexity and performance, especially when deploying in resource-constrained environments.

(viii) **Interpretable representations:** DL models often lack interpretability, making it challenging to understand the learned representations. Interpretable feature extraction methods that provide insights into the underlying spatial characteristics of RS images are desirable. Attention mechanisms can play a

crucial role in guiding the feature extraction process in DL-based SISR methods for RS images. They can help focus on relevant regions or spatial structures in the images, enhancing the model's ability to capture discriminative information.

(ix) **Need of different quantitative metric:** Traditional quantitative performance criteria, such as peak signal-to-noise ratio (PSNR) and structural similarity (SSIM), are insufficient to evaluate SISR results for RS images. To evaluate the effectiveness of RS super-resolved images, metrics that correlate RS image properties should be explored.

1.4 Motivation of the present work

RS satellites, particularly MS sensors often have low spatial resolution, which fail to deliver high-quality images for many practical RS applications. The development of efficient SISR methods is crucial for producing HR images that are useful for RS image analysis and applications. Learning-based SISR approaches have gained in popularity in RS due to their high performance and the fact that they synthesize the required information directly from the test image itself [77]. The aim of learning-based SISR approaches is to establish an end-to-end mapping between LR and their corresponding HR image patches. High-frequency feature extraction from LR remote sensing images are essential to enhance learning-based SISR methods. Sparse representation is one of the most successful learning-based methods for SISR that shows effective result for RS images. However, the reconstruction quality of LR image is largely dependent on how good the trained dictionary, as well as the effective handcrafted feature extraction strategy required for the efficient dictionary training. The dictionary learning and regularization operations involved in the sparse representation-based SR methods are also time consuming. Moreover, MS images consist of several spectral bands leading to big data volume. Therefore, it is computationally exhaustive to restore HR images from volumetric data. In order to solve these problems, parallel computing using the general purpose graphics

processing unit (GPGPU) can be exploited for real-time SISR of RS images. These aspects motivate to develop a fast sparse representation-based SISR technique using an efficient dictionary learning method with improved feature extraction for RS images.

DL is the most popular trend in image processing that achieves state-of-the-art performance in the SISR of natural images. Because DL-based SISR has the ability to automatically extract high-level and complex features, it is especially useful of RS images, which has a highly complex and detailed structure. CNN-based DL networks, in particular, would automatically learn and represent high-level features, outperforming traditional learning-based approaches that depend only on low-level feature representations. It is crucial in SISR approaches to enhance network depth for improved performance. However, the depth of the network affects the representational power of deep networks. As a result, attention-based modules are widely used in deep CNN architecture to yield adaptively learnable, highly informative feature maps. This will help the network in boosting its discriminative learning capacity as well as its representation power for better outcomes. However, remote sensing SR for real-time may be difficult in DL-based approach because of its limitations, such as higher computation overhead, memory and requirement of a large amount of data for training and validation. GPU can be used for accelerating the training time of DL to handle this computational overhead effectively. These motivate us to develop a fast deep CNN-based SISR network with an attention module to enhance the network's representational power and accuracy for RS images.

1.5 Scope of the work

The application value of data products is determined directly by the resolution of RS images. Learning-based SISR algorithms have superior image reconstruction ability and are suitable for a wider range of RS applications. Although this method obtains better reconstruction quality, it is computationally exhaustive. The developed sparse representation-based algorithm not only obtains the high-quality RS

images, but also computationally less heavy, which has a wide scope for real-time applications. Advancements in DL technology are more significant due to the state-of-the-art architectural design and super-resolving abilities when compared to sparse representation-based approaches. The proposed CNN-based SR method is expected to further enhance the resolution and quality of RS imagery, providing more accurate and detailed images for a wide range of applications. Furthermore, it has a low inference time, making it fast and beneficial for real-world RS applications. Overall, it can be said that the proposed works in this thesis have a broad scope of potential applications in real-world sensing scenarios.

It is important to note that the thesis also outlines certain limitations outside its scope. Firstly, it does not assess other types of SISR algorithms, such as multi-level/scale networks, GANs, transformers or unsupervised learning methods. The thesis focus remains to be sparse representation-based and supervised DL-based methods. Additionally, the thesis does not deal with practical aspects of deploying and integrating the proposed SISR algorithms into existing RS systems or workflows. The primary focus remains on algorithmic developments and performance evaluations, rather than implementation specifics and system-level issues.

1.6 Contributions from the Thesis

The research works carried out in the thesis resulted in the following contributions:

i) Development of a new method for enhancing the resolution of MS remote sensing images using sparse coding and adaptive dictionary learning. Unlike traditional methods that rely on external HR images, this approach uses the LR multispectral remote sensing image itself to learn dictionaries based on sparse representations. The method also incorporates a new feature extraction technique using difference of Gaussians (DoG), Sobel, and fast Fourier transform (FFT) filter, which helps in efficient dictionary learning and image reconstruction. Proposed a new parallel implementation of the orthogonal matching pursuit (OMP) algo-

rithm in adaptive dictionary learning using compute unified device architecture (CUDA) programming on GPU. The reconstruction algorithm has also been accelerated with the help of CUDA programming model using the GPU. The method is capable of processing real MS remote sensing images band wise with sizes up to 2048×2048 within a few seconds for different upscaling factors using a parallel framework.

ii) A novel technique for overcomplete dictionary learning in remote sensing SR that utilize both keypoints and non-keypoints features. To effectively preserve structural and textural features, multiple coupled dictionaries are learned from an external RS database. These dictionaries include SIFT-based keypoints and non-keypoints patch-based pairs, which are more effective at preserving high frequency information than traditional patch-based dictionaries alone. A joint sparse reconstruction model is proposed that combines SIFT-driven keypoints and NLTV regularization priors, and solves different sub-problems iteratively using the alternating direction method of multipliers (ADMM). To accelerate dictionary learning and reconstruction tasks using GPU, hybrid CPU-GPU algorithms are developed based on the CUDA programming model with GPU and OpenCV. The proposed parallel framework is able to process real-time RS images up to 2048×2048 in a matter of seconds for various upscaling factors.

iii) A novel joint dual-branch CNN network to recover the sharp and clear HR images from LR remote sensing images with Gaussian blur. The feature extraction stage is decomposed into two task-independent branches, namely, deblurring and SR feature extraction stages, and then train an attention-based gate module for fusing the features from these branches adaptively, making this dual-branch CNN network to handle SR and deblur tasks jointly. In order to extract SR features, a residual spatial and channel squeeze-and-excitation (RSCSE) module is developed, where a concurrent spatial and channel squeeze-and-excitation (SCSE) module is employed in residual blocks. The SCSE module is capable of making the feature maps more informative by recalibrating feature maps separately, spatially and channel wise, and then combining them both. Each RSCSE module employs the local feature fusion (LFF) concept to adaptively preserve

local features. Further, a deblurring module is developed that uses a simple SCSE-based encoder-decoder CNN architecture to extract sharp features from blurry LR images.

List of publications:

A. *Journals:*

(i) Trishna Barman, Bhabesh Deka and Helal Uddin Mullah, "Edge-Preserving Single Remote-Sensing Image Super-resolution Using Sparse Representations," SN Computer Science, Springer, vol. 4, no. 3, 2023. Doi: 10.1007/s42979-023-01764-7

(ii) Bhabesh Deka, Helal Uddin Mullah, Trishna Barman and A.V.V. Prasad, "Joint Sparse Representation-based Single Image Super-Resolution for Remote Sensing Applications," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 16, pp. 2352-2365, 2023. Doi: 10.1109/JSTARS.2023.3244069

(iii) Trishna Barman and Bhabesh Deka, "A Deep Learning-based Joint Image Super-resolution and Deblurring Framework," IEEE Transactions on Artificial Intelligence, 2023. Doi: 10.1109/TAI.2023.3343319

B. *Book Chapters:*

(i) Helal Uddin Mullah, Bhabesh Deka, Trishna Barman, and A.V.V. Prasad, "Sparsity Regularization Based Spatial-Spectral Super-Resolution of Multispectral Imagery," Pattern Recognition and Machine Intelligence (PReMI-19), Lecture Notes in Computer Science, Springer, vol. 11941, pp. 523 531, November 2019. Doi: 10.1007/978-3-030-34869-4_57

C. *Conferences:*

(i) Trishna Barman and Bhabesh Deka, "Attention-based Residual Network for Single Image Remote Sensing Super-resolution," 2022 International

Conference on Computing, Communication, and Intelligent Systems (ICCCIS), pp. 569-574, 2022. Doi: 10.1109/ICCCIS56430.2022.10037720

(ii) Trishna Barman, Bhabesh Deka and A. V. V. Prasad, "GPU-Accelerated Adaptive Dictionary Learning and Sparse Representations for Multispectral Image Super-resolution," 2021 IEEE 18th India Council International Conference (INDICON), 2021, pp. 1-7, December 2021. Doi: 10.1109/INDICON52576.2021.9691521

1.7 Thesis outline

The thesis is organized into six chapters. In the following, a brief introduction to each chapter is given:

Chapter 1:

This chapter introduces the fundamentals of image SISR, formulation of SISR, RS SISR, concept of sparse representation- and deep-learning-based image SR.

Chapter 2:

This chapter gives an overview of traditional and current SR methods, along with relevant literature around them. An in-depth review of sparse-based and DL-based SISR techniques, along with various approaches related to these methods for obtaining the HR image in RS images, is provided. It also discusses the brief background of CUDA-enabled GPU hardware for parallel processing, its implementation in SR and the related literature of CUDA-GPU-based SR works. Additionally, evaluation parameters and details of dataset used in the thesis are explained. Lastly, the chapter concludes with a summary and highlights few research issues on this topic.

Chapter 3:

In this chapter, we developed a novel framework for the SISR of RS images using sparse coding and self-example-based dictionary learning. Instead of training from external HR images, coupled dictionaries based on sparse representations are learnt from the given RS LR image itself. A feature extraction step based on DoG, Sobel

and FFT filters is introduced for efficient HR dictionary learning as well as sparse reconstruction of HR images. Further, we designed highly parallelized algorithms for the orthogonal matching pursuit (OMP) in adaptive dictionary learning and reconstruction module; hardware acceleration with NVIDIA P100 GP-GPU hardware is achieved using the CUDA programming model. The parallel framework for SR can process real MSRS images up to 2048×2048 within a few seconds for different upscaling factors. Simulations are carried out using real MS remote sensing images acquired by the Indian Satellite- Linear Imaging Self Scanner (LISS-IV). Results are evaluated in terms of visual quality and objective fidelity criteria, besides computational time and compared with the state-of-the-art.

Chapter 4:

In this chapter, we have proposed a parallel SISR framework based on edge preserving dictionary learning and sparse representations on CUDA-enabled GPU platform. To recover edges, multiple coupled dictionaries, namely, the scale-invariant feature transform (SIFT) keypoints and non-keypoints patch-based dictionaries are learned. In particular, a sparse reconstruction model with SIFT-driven and non-local total variation regularization priors is presented. These sub-problems are solved iteratively using the alternating direction method of multipliers (ADMM). We proposed hybrid CPU-GPU algorithms based on CUDA programming model for dictionary learning and sparse reconstruction using hardware acceleration with NVIDIA P100 GP-GPU hardware. CUDA-based implementation of the ADMM technique is also done to solve the above joint problem. Extensive simulations are demonstrated on two publicly available RS and two real MS remote sensing datasets for various scaling factors to show that the proposed method outperforms state-of-the-art techniques both visually and quantitatively. Also, proposed parallel SR framework obtained remarkable speedup in comparison to CPU counterparts, implying a great potential for real-time applications.

Chapter 5:

In this chapter, we designed a joint dual-branch CNN network for image deblurring and SR. The feature extraction network is divided into two task-independent branches, i.e. deblurring and SR; features from these two branches are adaptively

fused by learning a gate module with attention to generate a clear HR from LR remote sensing images with Gaussian blur. We developed a residual spatial and channel squeeze-and-excitation (RSCSE) module to extract SR features of RS images; the SCSE module is adopted in residual blocks. Further, local feature fusion (LFF) concept is used in each RSCSE block for preserving the local features adaptively. Further, a deblurring module is developed that uses a simple SCSE-based encoder-decoder CNN architecture to extract sharp features from blurry LR images. The proposed network is evaluated on two publicly available RS datasets and two real MS image datasets. Results obtained by the proposed network achieves better reconstruction of RS images in terms of visual analysis and objective criteria.

Chapter 6:

This chapter concludes this thesis by providing a brief summary of the work done in the previous chapters and outlines potential future research directions in the field.