Chapter 5

JAMiSR- Joint Adaptive Multispectral Image Super-resolution via Sparse Representations and its Applications

5.1 Introduction

The fundamental work on natural image SR using sparse representation was proposed by Yang *et al.* [115], which is known as sparse coding SR (ScSR). In traditional sparse coding, individual image patches are reconstructed separately assuming that they are independent and uncorrelated due to which instabilities arise among the sparse solutions. Although the patch-based methods have achieved success in improving the image quality both visually and quantitatively, yet they lack in minimizing the serrated edges and the ringing artifacts as demonstrated in Fig. 5.1.

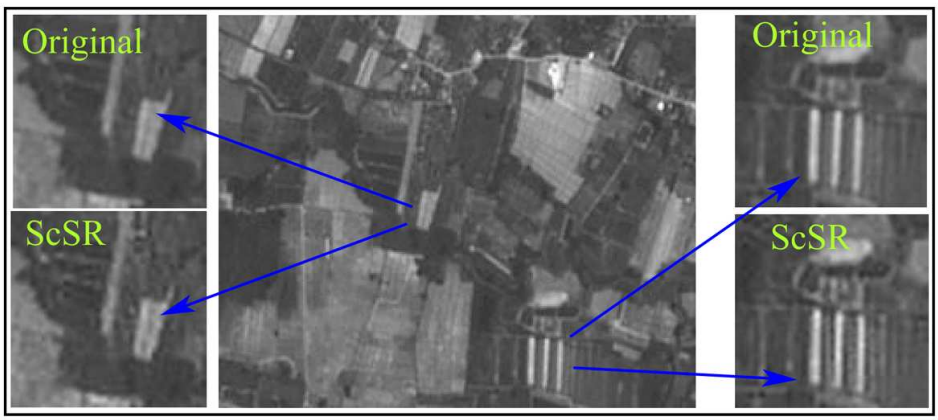


Figure 5.1: Pictorial demonstration of jaggy edge and ragged surface generated in the patch-based SR methods.

Different research works suggested different approaches to improve the results of ScSR either by learning a more effective dictionary or extracting features of impact or considering novel *a priori* information in the reconstruction problem

[53, 57, 82, 130]. Currently, focus is on the sparse representation of groups of similar patches to maintain a consistency among the recovered patches in image restoration problems [62, 113, 117, 123, 126]. A graphical example is shown in Fig. 5.2, which depicts the superiority of group-based sparse coding over single patch-based sparse coding in terms of consistency and stability of the estimated sparse coefficients for a prescribed number of non-zeros coefficients in the sparse matrix (i.e. the sparsity pattern). The group-based sparse coding is well balanced for the given sparsity pattern.

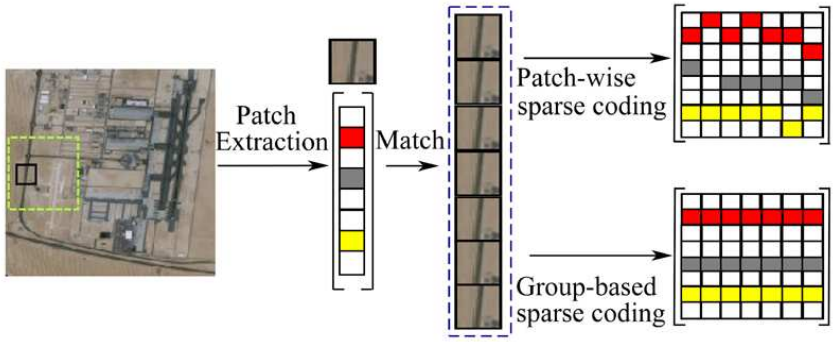


Figure 5.2: Example of a group-based sparse coding showing more stability/consistency in coefficients' estimation (source: [117], Fig. 1)

In this work, we focus on developing a joint sparse representation-based SR algorithm by integrating the concepts of both patch and group sparse coding together in a single framework. It is observed that similar patches may occur at multiple locations (non-locally) within a single image irrespective of any scaling [45]. Although different SR works are available combing the non-local self-similarity (NLSS) with patch-based sparse representation [17, 75], only a few works are reported exploiting the group sparse representation (GSR) technique for remote sensing image SR.

5.1.1 Patch sparse representation (PSR)

As discussed in Chapter 3 and 4, in traditional patch-based sparse representation, an image patch $\mathbf{x}_i \in \mathbb{R}^{n\times 1}, \forall i = 1, 2, ..., N$ can be exactly represented or approximated by linear combination of a few basis functions or atoms/columns of an overcomplete dictionary $\mathbf{D} \in \mathbb{R}^{n\times K}$ where $n \ll K$. The dictionary atoms are selected based on

the sparse coefficient vector $\boldsymbol{\alpha}_i \in \mathbb{R}^{K\times 1}$ Fig. 5.3 shows the sparse representation of an image patch using only a few selected dictionary atoms.

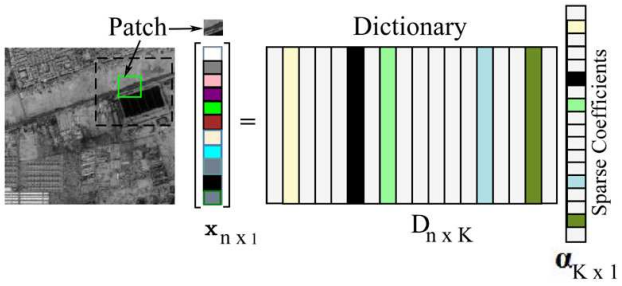


Figure 5.3: Schematic representation of patch-based sparse representation

Mathematically, the patch-based sparse representation problem is given by Eq. 1.14 in Chapter 1 and Eq. 3.10 in Chapter 3. In computational imaging, PSR has been widely used for SR-based image restoration [35, 133]. As reported in previous chapters, in SISR, a coupled overcomplete dictionary $\mathbf{D}_C$ consisting of both low- and high-resolution dictionaries, $\mathbf{D}_\ell$ and $\mathbf{D}_h$, respectively are usually learned using training sample patches extracted from both the given LR and similar available HR image datasets [115]. Usually, $\mathbf{D}_C$ can be efficiently learned using the K-SVD algorithm reported in [111]. As shown in Fig. 5.1, sparsity prior based SR images are always not able to provide sufficient improvements in terms of visual or quantitative evaluations. This can be either due to the lack of accurate dictionary training or inconsistent sparse coefficients.

5.1.2 Group sparse representation (GSR)

A patch group may be represented by a matrix with columns formed by nonlocal image patch vectors having the most identical features in them, and is the basic unit of group sparse coding. A given image $\mathbf{X}$ is first subdivided into N numbers of overlapping patches of size $\sqrt{n} \times \sqrt{n}$. Next, as shown in Fig. 5.4, for each example patch $\mathbf{x}_i$, the closest k similar patches are selected from a search window of size $W \times W$. Here, the similarity is measured in terms of the Euclidean distance between

the target patch $\mathbf{x}'$ and the candidate patch $\mathbf{x}$ as follows:

$$d_i = \sqrt{\sum_{i=1}^{\sqrt{n}} \sum_{j=1}^{\sqrt{n}} (\mathbf{x}_{i,j} - \mathbf{x'}_{i,j})^2}, \tag{5.1}$$

where $\mathbf{x}_{i,j}$ and $\acute{\mathbf{x}}_{i,j}$ represent the pixel values of the target and candidate patches, respectively, while d_i is the distance between them. Now, similar patches within the search window $S_{\mathbf{x}_i}$ are vectorized and stacked as columns of the group matrix $\mathbf{X}_{g_i} = \{\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,k}\} \in \mathbb{R}^{n \times k}$, where $\mathbf{x}_{i,j}$ represents the j-th patch in the i-th group.

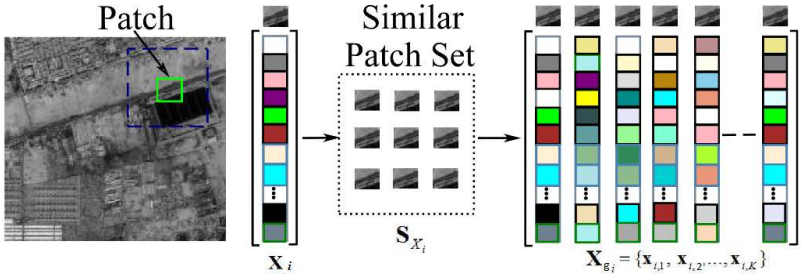


Figure 5.4: Formation of patch-group matrix (source: [126], Fig. 1).

Similar to patch-wise sparse coding, assuming a group dictionary $\mathbf{D}_{g_i} \in \mathbb{R}^{n \times M}$ having M atoms, we can carry out sparse coding for each patch-group $\mathbf{X}_{g_i}$ corresponding to the i-th patch over $\mathbf{D}_{g_i}$ as follows:

$$\hat{\boldsymbol{\beta}}_{g_i} = \arg\min_{\boldsymbol{\beta}_{g_i}} \left(\frac{1}{2} \left\| \mathbf{X}_{g_i} - \mathbf{D}_{g_i} \boldsymbol{\beta}_{g_i} \right\|_2^2 + \lambda \left\| \boldsymbol{\beta}_{g_i} \right\|_1 \right), \tag{5.2}$$

where $\boldsymbol{\beta}_{g_i} \in \mathbb{R}^{M \times k}$ are the corresponding group sparse coefficients matrices. If we put all the patch-groups extracted from N patch locations of the given image $\mathbf{X}$ together as $\mathbf{X}_G = [\mathbf{X}_{g_1}, \mathbf{X}_{g_2}, \ldots, \mathbf{X}_{g_N}] \in \mathbb{R}^{n \times kN}$, we can use a dictionary $\mathbf{D}_G = [\mathbf{D}_{g_1}, \mathbf{D}_{g_2}, \ldots, \mathbf{D}_{g_N}] \in \mathbb{R}^{n \times MN}$ to rewrite the Eq. 5.2 as follows:

$$\hat{\boldsymbol{\beta}}_G = \arg\min_{\boldsymbol{\beta}_G} \left(\frac{1}{2} \left\| \mathbf{X}_G - \mathbf{D}_G \boldsymbol{\beta}_G \right\|_2^2 + \rho \left\| \boldsymbol{\beta}_G \right\|_1 \right), \tag{5.3}$$

where ρ is the regularization parameter and $\boldsymbol{\beta}_G \in \mathbb{R}^{MN \times kN}$ is an expanded version of $\boldsymbol{\beta}_{g_i}$. Split Bregman iteration (SBI) [46] or accelerated SBI method (ASBM) [43] are successfully utilized to solve the above minimization problem in image restorations

[62, 126]. Incorporating GSR into SR problem can help in enhancing the non-local information in an image. However, it is found that with higher upscale ratios, SR reconstructed images suffer from oversmoothing. GSR alone is not adequate to restore the high-frequency information in the target HR image which might be obtained by PSR.

5.2 Joint Sparse Representation (JSR)

Joint sparse representation (JSR) model aims to mitigate the associated drawbacks of PSR and GSR mentioned in Subsections 5.1.1 and 5.1.2, by fitting them within a joint model that integrates both the local sparsity (from patch) with the non-local self-similarity (from patch groups) of an image. In JSR, a joint regularization problem is formulated by combining the sub-problems of PSR and GSR, respectively. Since different patch-groups $\mathbf{X}_{g_i}$ of $\mathbf{X}_G$ would contain the same patches available in the patch dataset $\mathbf{X}$ of the input image, we can also use the patch-based dictionary $\mathbf{D}$ to represent the patch-group $\mathbf{X}_G$ i.e.

$$\mathbf{X}_G = \mathbf{D}\boldsymbol{\alpha}_\mathbf{G}, \tag{5.4}$$

where $\boldsymbol{\alpha}_G \in \mathbb{R}^{K \times kN}$ is a matrix consisting of sparse representations α of N groups as its columns. The corresponding unconstrained optimization problem is:

$$\hat{\boldsymbol{\alpha}}_G = \arg\min_{\boldsymbol{\alpha}_G} \left(\frac{1}{2} \left\| \mathbf{X}_G - \mathbf{D}\boldsymbol{\alpha}_G \right\|_2^2 + \lambda \left\| \boldsymbol{\alpha}_G \right\|_1 \right). \tag{5.5}$$

This suggests that we can combine $\hat{\boldsymbol{\beta}}_G$ and $\hat{\boldsymbol{\alpha}}_G$ of Eqs. 5.3 and 5.5, respectively to estimate $\mathbf{X}_G$ using the JSR model as follows:

$$\hat{\mathbf{C}} = \arg\min_{\mathbf{C}} \frac{1}{2} \left\| \mathbf{X}_G - \mathbf{U}\mathbf{C} \right\|_2^2 + \lambda \left\| \boldsymbol{\alpha}_G \right\|_1 + \rho \left\| \boldsymbol{\beta}_G \right\|_1, \tag{5.6}$$

where $\mathbf{U} = [\mathbf{D}\ \mathbf{D}_G]$ and $\mathbf{C} = \begin{bmatrix} \boldsymbol{\alpha}_G \\ \boldsymbol{\beta}_G \end{bmatrix}$. Eq. 5.6 can be solved by splitting it into two unconstrained minimization sub-problems related to $\hat{\boldsymbol{\alpha}}_G$ and $\hat{\boldsymbol{\beta}}_G$. To solve such regularization problems, methods, like ADMM [10] or SBI [46] are successfully applied. The patch sparsity prior term associated with $\boldsymbol{\alpha}$ helps in reducing the over-smoothing effect by inducing more local information, while the group sparsity prior term associated with $\boldsymbol{\beta}$ subdues the visual artifacts by maintaining nonlocal uniformity. An example of how combining PSR and GSR provides enhanced result in terms of contrast and PSNR for a MS band image is shown in Fig. 5.5. Recently, the JSR models are also being successfully exploited by researchers to solve other image restoration problems (e.g. image inpainting) [122].

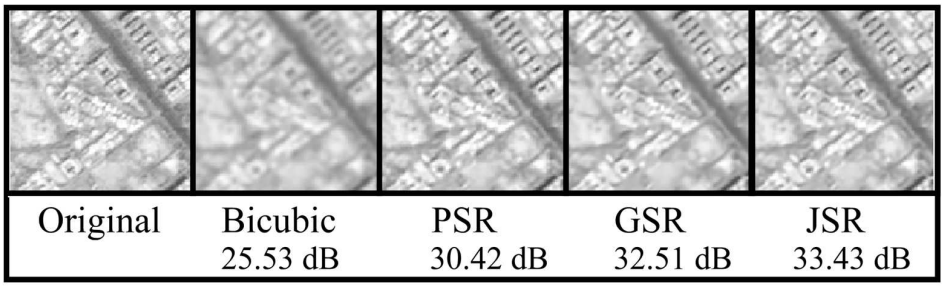


Figure 5.5: Example of improved performance by the JSR approach over the PSR and GSR corresponding to 2 times upscaling.

5.3 Adaptive versus global dictionary learning

The performance of a sparse representation-based technique highly relies on the selection of effective dictionary learning method. Traditional global dictionary learning methods used in image SR can not yield an efficient dictionary for MS satellite imagery as the individual bands inherently carries low spatial information and sometimes they represent no significant information. The global dictionaries are potentially unstable and too many atoms are found to be irrelevant (i.e. low correlation) to the given patch. Also, due to high computational time, the global dictionaries are learned offline, which itself is a limitation for practical remote sensing applications.

On the other hand, an adaptive dictionary can be trained online from the given LR test image itself during SR reconstruction [49, 112]. Here, the dictionary atoms are locally adopted to the given signal, which increases the stability and accuracy of sparse approximation [35]. A dictionary $\mathbf{D}$ can be learned by applying the sparse coding approach (e.g. K-SVD) using a set of example patches taken from the given MS image, called the patch dataset $\mathbf{X}_T = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$. Mathematically, the adaptive dictionary learning problem may be defined as follows:

$$\left(\hat{\mathbf{D}}, \hat{\boldsymbol{\alpha}}\right) = \arg \min_{\{\mathbf{D}, \boldsymbol{\alpha}\}} \left(\sum_{k=1}^{N} \tfrac{1}{2} \left\| \mathbf{x}_k - \mathbf{D}\boldsymbol{\alpha}_k \right\|_2^2 + \lambda \left\| \boldsymbol{\alpha}_k \right\|_1 \right) \quad \text{such that } \left\| \boldsymbol{\alpha}_k \right\|_2^2 \leq 1,\ k = 1, 2, ..., N \tag{5.7}$$

Here, the sparse coefficients $\boldsymbol{\alpha}$ and the dictionary $\mathbf{D}$ are jointly optimized by fixing either of them and iteratively updating the other until the cost function reaches the convergence.

5.4 Proposed method

The proposed algorithm restores the desired HR image by solving two sparse coding sub-problems- the patch sparse representation (PSR) and the group sparse representation (GSR) for individual image patches and patch-groups, respectively. The above two sub-problems may be combined and called it as the joint sparse representation (JSR) problem, which can be fitted within the ADMM framework for the final estimation of the SR image. We named it as the joint sparse representation-based adaptive multispectral image SR (JAMiSR). A graphical representation of the proposed method is given in Fig. 5.6. Details of adaptive dictionary learning and joint sparse reconstruction techniques are given in the following subsections.

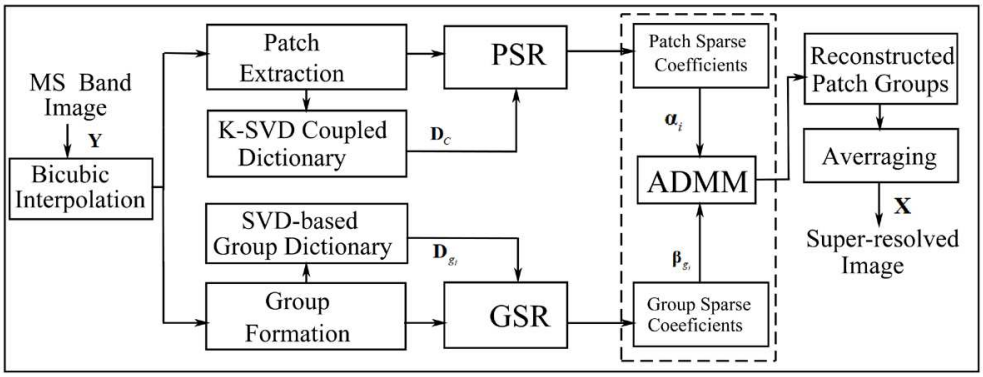


Figure 5.6: Schematic overview of the proposed JAMiSR method.

5.4.1 Self-adaptive group dictionary training

We now recall the traditional SISR problem, i.e. recovering the HR version $\mathbf{X}$ of an image from the observed LR image $\mathbf{Y}$ following the imaging model defined as

$$\mathbf{Y} = SH\mathbf{X} + \mathbf{n}, \tag{5.8}$$

where $\mathbf{n}$ is the additive white Gaussian noise or representation error, which is normally taken care of by the fidelity term, while solving for sparse approximation. First, we learn a coupled overcomplete dictionary $\mathbf{D}_C$ from the HR-LR combined patch dataset $\mathbf{Y}_C = (\mathbf{Y}_h; \mathbf{Y}_\ell)$. In line with the discussions on LR-HR dataset preparation for global dictionary learning (in Subsections 3.3.2, 4.4.1), in the previous chapters, we adopt the similar strategy for self-adaptive dictionary learning as well. Here, HR patches having the most relevant high-frequency information are directly extracted from the input image itself and stored as columns in the matrix $\mathbf{Y}_h$. On the other hand, for the corresponding LR feature patch, the input image is first blurred, downsampled, and then resized to the original image size by using bicubic interpolation. In order to extract high-frequency features and their orientations from the LR patch, we apply four 1-D Sobel filters in x-, y- and two diagonal directions ($+45$ deg and -45 deg) on each patch. We prefer the Sobel filters here since they are simple to use and capable of generating effective gradient maps, which essentially import details, like edge, contour, etc., for accurate sparse representation. Now, an LR feature vector is formed corresponding to each patch by concatenating

four gradient maps resulting from the preceding step. Finally, the feature matrix $\mathbf{Y}_\ell$ is obtained by stacking feature vectors corresponding to every patch in the input image as columns. Assuming that low- and high-resolution dictionaries $\mathbf{D}_h$ and $\mathbf{D}_\ell$ share the same sparse coefficients matrix $\mathbf{A}$, the coupled dictionary $\mathbf{D}_C = (\mathbf{D}_h; \mathbf{D}_\ell)$ can now be jointly learned from $\mathbf{Y_C}$. Mathematically, this is given as:

$$\min_{\{\mathbf{D}_C, \mathbf{A}\}} \|\mathbf{Y}_C - \mathbf{D}_C\mathbf{A}\|_2^2 + \lambda \left(\tfrac{1}{p} + \tfrac{1}{q}\right) \|\mathbf{A}\|_1 , \tag{5.9}$$

where p and q are the dimensions of high- and low-resolution patch vectors, respectively and λ is the regularization parameter. Eq. 5.9 is a simultaneous sparse coding and dictionary updating problem, which is efficiently solved by the coupled K-SVD algorithm [111].

Next, for the GSR, we learn self-adaptive group dictionaries $\mathbf{D}_{g_i}$ for individual patch-groups $\mathbf{Y}_{g_i}$ rather than learning single overcomplete dictionary $\mathbf{D}_g$ from all available patch-groups because learning a single dictionary by joint sparse coding of all available patch-groups at a time is computationally expensive. Adaptive group dictionaries $\mathbf{D}_{g_i}$ $(i = 1, 2, ..., M)$ are directly learned from the given image patch-groups $\mathbf{Y}_{g_i}$ $(i = 1, 2, ..., M)$ using the singular value decomposition (SVD) technique to obtain its low-rank estimate $\mathbf{Y}_{g_r}$, i.e.

$$\mathbf{Y}_{g_r} = \mathbf{U}_{g_i} \sum\nolimits_{g_i} \mathbf{V}_{g_i}^T = \sum_{j=1}^{r} \sigma_{g_i \otimes j} \left(\mathbf{u}_{g_i \otimes j} \mathbf{v}_{g_i \otimes j}^T\right), \tag{5.10}$$

where $\mathbf{U}_{g_r}$ and $\mathbf{V}_{g_r}$ are orthogonal matrices consisting of the left- and right-singular vectors of $\mathbf{Y}_{g_r}$, respectively and $\sum_{g_r} = \text{diag}\left(\gamma_{Y_{g_i \otimes j}}\right)$ is a diagonal matrix consisting of r non-zero singular values of $\mathbf{Y}_{g_i}$ represented by $\gamma_{Y_{g_i \otimes j}} = [\sigma_{g_i \otimes 1}; \sigma_{g_i \otimes 2}; ...; \sigma_{g_i \otimes r}]$ on its principal diagonal in the descending order, i.e. $\sigma_{g_i \otimes 1} \geq \sigma_{g_i \otimes 2} \geq ... \geq \sigma_{g_i \otimes r}$. Now, if columns of $\mathbf{U}_{g_i}$ and $\mathbf{V}_{g_i}$ are denoted by $\mathbf{u}_{g_i}$ and $\mathbf{v}_{g_i}$, respectively then each submatrix $\mathbf{d}_{g_i} \in \mathbb{R}^{m \times k}$ of the group-dictionary $\mathbf{D}_{g_i}$ is obtained as follows:

$$\mathbf{d}_{g_i \otimes j} = \mathbf{u}_{g_i \otimes j} \mathbf{v}_{g_i \otimes j}^T, \quad {}_{j=1,2,...,r}. \tag{5.11}$$

Finally, the adaptively learned dictionary for the group $\mathbf{Y}_{g_i}$ can be obtained by concatenating all the r submatrices obtained from above as follows:

$$\mathbf{D}_{g_i} = \left[\mathbf{d}_{g_i \otimes 1}, \mathbf{d}_{g_i \otimes 2}, ..., \mathbf{d}_{g_i \otimes r}\right]. \tag{5.12}$$

The proposed self-adaptive group dictionary training is more robust and effective as it assures that all the patches in each group $\mathbf{Y}_{g_i}$ are using the same dictionary $\mathbf{D}_{g_i}$ and sharing the same dictionary atoms. The learning is also very efficient as it needs to solve only one SVD problem for each group.

5.4.2 Joint sparse reconstruction-based SR

The proposed JSR-based reconstruction, first, resizes the LR image $\mathbf{Y}$ according to the upscale ratio using bicubic interpolation. Then a joint regularization problem incorporating a priori terms of the PSR and GSR subproblems is solved using the ADMM. Mathematically, the proposed reconstruction model is given as follows:

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \|\mathbf{Y} - SH\mathbf{X}\|_2^2 + \lambda \Re_{JSR}(\mathbf{X}), \tag{5.13}$$

where $\Re_{JSR}(\mathbf{X})$ represents the JSR regularization, which can be further split into PSR and GSR regularizations, i.e.

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \|\mathbf{Y} - SH\mathbf{X}\|_2^2 + \lambda_1 \Re_{PSR}(\mathbf{X}) + \lambda_2 \Re_{GSR}(\mathbf{X}), \tag{5.14}$$

where λ_1 and λ_2 are the regularization parameters. Here, the first regularization term $\Re_{PSR}(\mathbf{X})$ solves a patch-wise optimization problem with the input bicubic upscaled image $\mathbf{Y}$ and pre-trained LR and HR dictionaries of Eq. 5.9 as follows.

$$\Re_{PSR}(\mathbf{X}) = \sum_i \frac{1}{2} \left\| \tilde{\mathbf{y}}_i - \tilde{\mathbf{D}}\boldsymbol{\alpha_i} \right\|_2^2 + \lambda_1 \|\boldsymbol{\alpha_i}\|_1, \tag{5.15}$$

where $\tilde{\mathbf{D}} = \begin{pmatrix} \mathbf{D}_l \\ E_p\mathbf{D}_h \end{pmatrix}$, $\tilde{\mathbf{y}}_i = \begin{pmatrix} \mathbf{y}_i \\ \mathbf{w} \end{pmatrix}$ and $\mathbf{y}_i = E_p(SH\mathbf{X})$ gives the i-th feature patch extracted by operator E_p and $\mathbf{w}$ contains the overlapping pixels between the present HR patch and previously generated HR image. In this work, an efficient ℓ_1-minimization method known as the features sign search algorithm [56] is adopted to solve the PSR-based regularization subproblem.

Algorithm 3: Proposed JSRMiSR Algorithm

Input: $\mathbf{Y}$, S, H, E_p, Γ, $\mathbf{D}_\ell$, $\mathbf{D}_h$, $\mathbf{D}_{g_i}$

Initialization: $t \leftarrow 0$, $\delta \leftarrow 10^{-4}$, λ_1, λ_2, μ_1, μ_2

1: **while** not converge **do**

2: $k \leftarrow k+1$

3: $\boldsymbol{\alpha}_i{}^t \leftarrow \arg\min_{\boldsymbol{\alpha}} \sum_i \left\| E_p(SH\mathbf{X}) - \tilde{\mathbf{D}}\boldsymbol{\alpha}_i \right\|_2^2 + \lambda_1 \|\boldsymbol{\alpha}_i\|_1$

4: $\boldsymbol{\beta}_{g_i}^t \leftarrow \arg\min_{\boldsymbol{\beta}} \sum_i \left\|\Gamma_i(SH\mathbf{X}) - \mathbf{D}_{g_i}\boldsymbol{\beta}_{g_i}\right\|_2^2 + \lambda_2 \left\|\boldsymbol{\beta}_{g_i}\right\|_1$

5: $\mathbf{X}^t \leftarrow \frac{(SH)^t\mathbf{Y} + \mu_1 \sum_i (E_pSH)^t E_p\tilde{\mathbf{D}}\boldsymbol{\alpha}_i + \mu_2 \sum_j (\Gamma_j SH)^t \mathbf{D}_{g_i}\beta_{g_i}}{(SH)^tSH + \mu_1 \sum_i (PSH)^t E_pSH + \mu_2 \sum_i (\Gamma_i SH)^t \mathbf{D}_{g_i}\boldsymbol{\beta}_{g_i}}$

6: check convergence :$\|\mathbf{X}^t - \mathbf{X}^{t-1}\| / \|\mathbf{X}^t\| \leq \delta$

7: **end while**

Output: $\mathbf{X}^* \leftarrow \mathbf{X}^t$

The second regularization term $\Re_{GSR}(\mathbf{X})$ solves a group sparse coding problem for the bicubic upscaled image $\mathbf{Y}$ by forming groups of similar patches. It finds sparse representation of the i-th group $\mathbf{Y}_{g_i}$ involving the dictionary $\mathbf{D}_{g_i}$, i.e. $\boldsymbol{\beta}_{g_i}$ as follows:

$$\Re_{GSR}(\mathbf{X}) = \sum_i \frac{1}{2} \left\|\mathbf{Y}_{g_i} - \mathbf{D}_{g_i}\boldsymbol{\beta}_{g_i}\right\|_2^2 + \lambda_2 \left\|\boldsymbol{\beta}_{g_i}\right\|_1, \tag{5.16}$$

where $\mathbf{Y}_{g_i} = \Gamma_i(SH\mathbf{X})$ is a patch-group extracted using the operator Γ from the i-th search window. The above GSR regularization subproblem is efficiently solved using the accelerated split Bregman algorithm [43].

Now, after solving the above two subproblems, their results are used to solve the main problem in Eq. 5.14 to find the target image $\hat{\mathbf{X}}$. Mathematically, we can

write it as follows:

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \|\mathbf{Y} - SH\mathbf{X}\|_2^2 + \mu_1 \sum_i \left\| E_p(SH\mathbf{X}) - \tilde{\mathbf{D}}\boldsymbol{\alpha}_i \right\|_2^2 + \mu_2 \sum_i \left\| \Gamma_i(SH\mathbf{X}) - \mathbf{D}_{g_i}\boldsymbol{\beta}_{g_i} \right\|_2^2, \tag{5.17}$$

where μ_1 and μ_2 are small positive regularization parameters. Minimization problem in Eq. 5.17 is strictly convex and a closed-form solution can be obtained by setting its gradient w.r.t. $\mathbf{X}$ to zero. The solution is shown in step 5 of Algorithm 1. In this work, we jointly solve both PSR ($\boldsymbol{\alpha}_i$) and GSR ($\boldsymbol{\beta}_{g_i}$) within the ADMM framework to estimate the final result $\mathbf{X}$. The proposed algorithm proves to be very crucial as the results seem to improve significantly.

5.5 Results and discussions

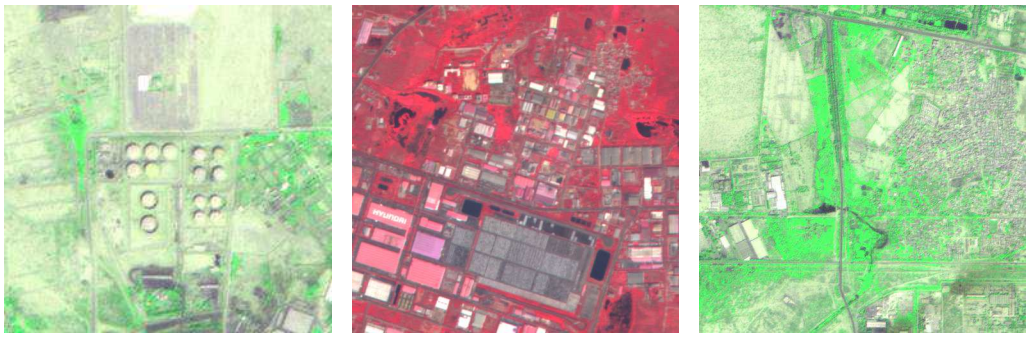
The proposed SR algorithm is tested on MS images captured by ResourceSat-2 satellite developed and launched by Indian Space Research Organization (ISRO) and collected from National Remote Sensing Center (NRSC), ISRO, Hyderabad, India.

5.5.1 Performance evaluation and comparisons

5.5.1.1 Dataset preparation

We have considered MS images acquired by the Linear Imaging Self Scanner IV (LISS-IV) sensor of the ResourceSat-2. It consists of three spectral bands out of which two are visible bands, i.e. Green (B2: 0.52-0.59 μm) and Red (B3: 0.62-0.68 μm) and the third one is the near infrared band i.e. NIR (B4: 0.77-0.86 μm). Each band has a spatial resolution of 5.8 m and swath coverage of 25 Km. As shown in Fig. 5.7, three representative images (Test1: 256×256, Test2: 512×512, and Test3: 510×510) are taken from the dataset for evaluating the algorithm;

'Test1' and 'Test3' have Date of Pass 07-OCT-2014, Latitude $25°04'03''02S$ and Longitude $66°95'44''83E$, while they are respectively 06-FEB-2017, $13°44'16''52N$ and $79°58'50''52E$ for 'Test2'. The corresponding LR test images are obtained by blurring and downsampling of these test images.



(a) Test1 **(b)** Test2 **(c)** Test3

Figure 5.7: LISS IV original test images. From left to right: Test1, Test2 and Test3

5.5.1.2 Experimental settings

In this work, Gaussian blur filters of sizes 5×5, 7×7 and 9×9 with standard deviations 1.2, 1.6 and 2.0 are applied on each MS band, and then downsampled by factors 2, 3, and 4, respectively to generate the LR image bands for simulations. The

Table 5.1: Selection of different parameters for the proposed algorithm

Parameters	**Values**
Gaussian blur filter**:** For upscale =2	kernel size = 5×5; standard deviation = 1.2
For upscale =3	kernel size = 7×7; standard deviation = 1.6
For upscale =4	kernel size = 9×9; standard deviation = 2.0
Patch size	5×5
Patch searching window size	20×20
Overlapping pixels between adjacent patches	4
Number of patches in each group	60
Regularization parameters	$\lambda_1 = 0.15$; $\lambda_2 = 0.2$
Regularization constants	$\mu_1 = 0.00005$; $\mu_2 = 0.005$; δ =10^-4
Dictionary sizes	PSR: $\mathbf{D}_h$ = 25×256; $\mathbf{D}_\ell$ = 100×256; GSR: $\mathbf{D}_{g_i}$ = 25×60

other parameters for this experiment are set as shown in the Table 5.1. We have considered six state-of-the-art sparse representation and learning-based SR methods, namely, ScSR [115], Moustafa [71], CRNS [17], GSRGSiSR [113], RAISR [85] and CDLSR [94] for comparisons with the proposed method. Results are compared

both visually and quantitatively. Moreover, for fair comparisons, we have considered five state-of-the-art deep learning-based methods and evaluated their results on three publicly available standard remote sensing datasets along with a few MS datasets collected from ISRO. Additionally, remote sensing image analysis are also being conducted in terms of MS image classifications and spectral graph-based evaluation techniques. All simulations are done in the MATLAB environment running on a Workstation equipped with 128 GB of RAM and Windows 10 OS.

5.5.1.3 Evaluation parameters

For validation of the proposed algorithm, we have computed several quantitative metrics e.g. peak signal-to-noise ratio (PSNR), structural similarity (SSIM), erreur relative globale adimensionnelle de synthese (ERGAS), spectral angle mapper (SAM), Q-index, spatial correlation coefficient (sCC), etc., along with three no-reference based parameters, namely, natural image quality evaluator (NIQE), entropy (EN), and enhanced measure evaluation (EME) are used for comparison of results and validation. For high quality reconstruction with rich information, PSNR, SSIM, Q-index, EME and EN should increase, while that of ERGAS, SAM and NIQE should decrease.

5.5.1.4 Visual interpretation of reconstructed images

SR reconstruction of each MS band is performed separately and then results are combined to achieve the desired HR MS image. We perform reconstruction of SR images corresponding to upscale ratios 2 and 4 for the images Test1 and Test2, while for Test3 we choose to perform SR by ratios 2 and 3, respectively. The reconstructed false color RGB images for different methods are shown in Fig. 5.8 for all the test images.

Here, results for 3× upscaling of Test1 and Test2 are not included because their sizes are not exactly divisible by 3 and an additional interpolation before

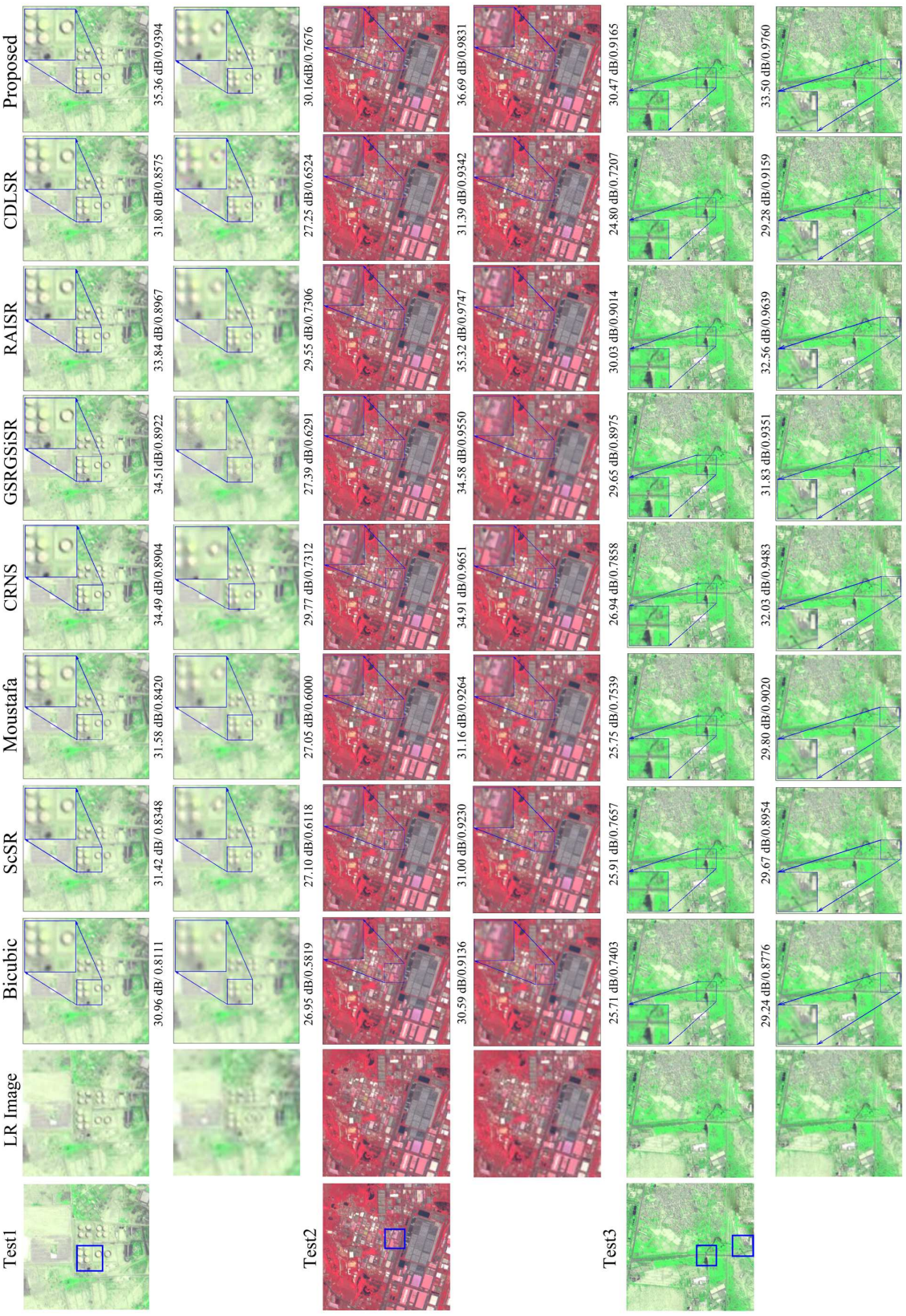


Figure 5.8: Visual comparisons of the SR outputs (row wise): rows 1–2 for 'Test1', rows 3–4 for 'Test2' images for upscale ratios 2 and 4, and rows 5–6 for 'Test3' image for upscale ratios 2 and 3 using different methods. The PSNR and SSIM measures for all the visual outputs are provided below each image for better interpretation.

downsampling by 3 may change the pixel properties with respect to the ground-truth besides adding to extra computation. While, it is also possible to select a suitable RoI of proper size, but for fair comparisons, we have not carried out that in our simulations. In all the visual comparisons, a small region of the reconstructed image is selected and then its zoomed in version is superimposed over it to highlight the quality of reconstruction. From visual inspection, it is clear that the proposed method provides the best perceptual quality of reconstructed images. It is able to restore the texture as well as structure components better than other methods. Although methods, like the CRNS, the GSRGSiSR and the RAISR are slightly closer in terms of PSNR, however, finer details in the reconstructed images tend to be smoothed. GSRGSiSR lacks in recovering the edges and sharp details, when subjected to higher upscalings e.g. 3 and 4. Also, it is noticed that other methods, like ScSR, CDLSR, etc., which do not utilize blurring in practice and simply use downsampling for LR image generation, fail to maintain similar performances for the same test images. Furthermore, error images between original and reconstructed images of Test1 and Test2 are shown in Fig. 5.9 for a better interpretation of visual results. Images in the first row of Fig. 5.9 from left to right (a-h) show the error images of Test1 for upscale ratio 2, where the intensity of error pixels are more (i.e. ~ 65) in case of bicubic interpolation, while it is the least visible in the proposed method. Similarly, error images of Test2 corresponding to upscale ratio 4 are shown in 5.9(j-q) and it demonstrate the least error produced by the proposed method.

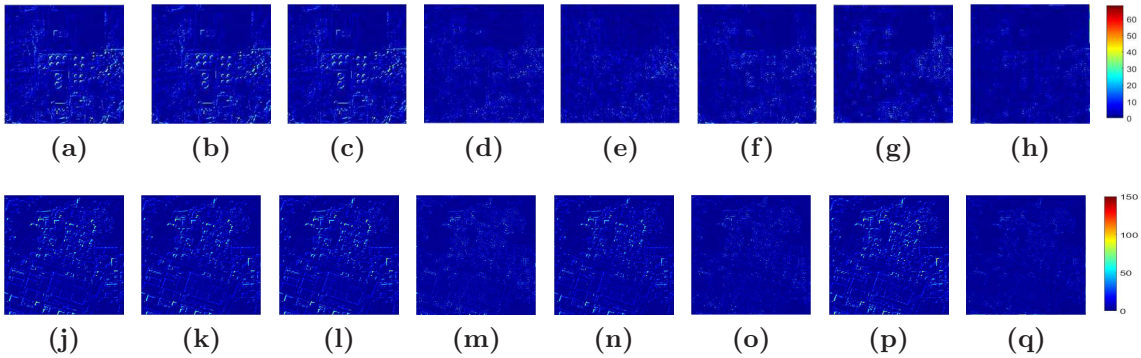


Figure 5.9: Error between original and reconstructed images: first row (a-h)- Test1-band2 with upscale ratio 2, second row (j-q)- Test-band3 with upscale ratio 4. In each row, from left to right it shows error images by: Bicubic, ScSR, Moustafa, CRNS, GSRGSiSR, RAISR, CDLSR and the proposed method, respectively.

Table 5.2: Quantitative performance evaluation of different sparse representation based SR algorithms tested for the LISS-IV datasets, 'Test1', 'Test2', and 'Test3' corresponding to upscale ratios 2, 4, and 3 respectively.

Image	Parameters	Bicubic	ScSR[115]	Moustafa[71]	GSRGSiSR[113]	CRNS[17]	RAISR[85]	CDLSR[94]	Proposed
Test1 (zoom = 2)	PSNR (dB)	30.96	31.42	31.58	34.49	34.51	33.84	31.80	**35.36**
	SSIM	0.8111	0.8348	0.8420	0.8904	0.8922	0.8967	0.8575	**0.9394**
	ERGAS	1.845	1.750	1.718	1.234	1.228	1.319	1.676	**1.125**
	SAM	0.5542	0.5311	0.5241	0.4805	0.3731	0.3527	0.5077	**0.2817**
	Q-index	0.7316	0.7665	0.7777	0.8303	0.8360	0.8511	0.8017	**0.9160**
	sCC	0.9753	0.9778	0.9786	0.9891	0.9890	0.9874	0.9796	**0.9909**
	EME	7.100	7.520	7.663	9.079	8.303	8.439	8.504	**9.641**
	EN	6.929	6.934	6.936	6.961	6.951	6.954	6.953	**6.970**
	NIQE	24.17	22.08	21.92	**15.35**	28.06	23.16	19.55	18.00
Test2 (zoom = 4)	PSNR (dB)	25.71	25.91	25.75	26.94	29.65	30.03	24.80	**30.47**
	SSIM	0.7403	0.7657	0.7539	0.7858	0.8975	0.9014	0.7207	**0.9165**
	ERGAS	2.539	2.482	2.527	2.202	1.633	1.544	3.756	**1.465**
	SAM	2.039	2.003	2.012	1.878	1.190	1.252	2.411	**1.147**
	Q-index	0.4645	0.5141	0.4928	0.5258	0.6832	0.6982	0.4712	**0.7320**
	sCC	0.9569	0.9588	0.9574	0.9678	**0.9864**	0.9844	0.9472	0.9859
	EME	41.15	45.32	44.52	43.25	49.46	52.56	54.54	**56.17**
	EN	7.341	7.354	7.352	7.344	7.365	7.376	7.377	**7.382**
	NIQE	33.44	28.39	27.61	**16.09**	27.30	27.84	23.85	22.44
Test3 (zoom = 3)	PSNR (dB)	26.86	27.34	27.13	28.49	29.68	29.94	26.29	**30.90**
	SSIM	0.7701	0.7953	0.7780	0.8321	0.8807	0.9106	0.7610	**0.9221**
	ERGAS	2.244	2.175	2.273	1.900	1.661	**1.610**	2.448	1.681
	SAM	1.258	1.232	1.246	1.154	0.8047	0.7565	1.400	**0.7166**
	Q-index	0.5313	0.5795	0.5479	0.6449	0.6928	0.7504	0.5551	**0.7762**
	sCC	0.9634	0.9657	0.9640	0.9739	0.9802	**0.9814**	0.9568	0.9796
	EME	33.85	36.08	34.67	41.40	37.67	41.88	44.80	**46.70**
	EN	7.315	7.321	7.317	7.336	7.336	7.343	7.350	**7.352**
	NIQE	23.58	15.02	20.72	**8.827**	21.26	18.62	20.06	13.68

5.5.1.5 Quantitative analysis

We also evaluate eight quantitative parameters mentioned above and shown in Table 5.2. We observe that average PSNR is the highest in case of the proposed method that is 35.36 dB, with an average improvement of 1-4 dB over other techniques. Similarly, SSIM is also the highest in case of the proposed method, while ERGAS and SAM values are the least. Parameters, like sCC and EME are also far better than other methods. Although we observe smaller NIQE values for the GSRGSiSR method (which may be due to the over smoothing effect) compared to the proposed method, it does make little impact as the proposed method comfortably outperforms it in terms of all other metrics. PSNR values of different test images are also graphically compared in Fig. 5.10, which reiterate the superiority of the proposed method. The remaining quantitative measures of Test1 with 4× upscaling, Test2 with 2× upscaling and Test3 with 2× upscaling are graphically depicted in Fig. 5.11. It can be observed that almost in case of every parameter the proposed method is able to achieve the best performance, when compared to others.

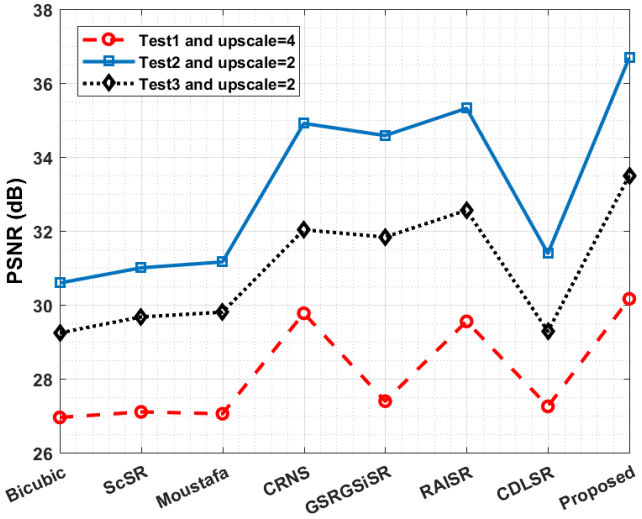


Figure 5.10: Comparison of PSNR values of Test1, Test2 and Test3 images with respect to upscale ratios 4, 2 and 2, respectively

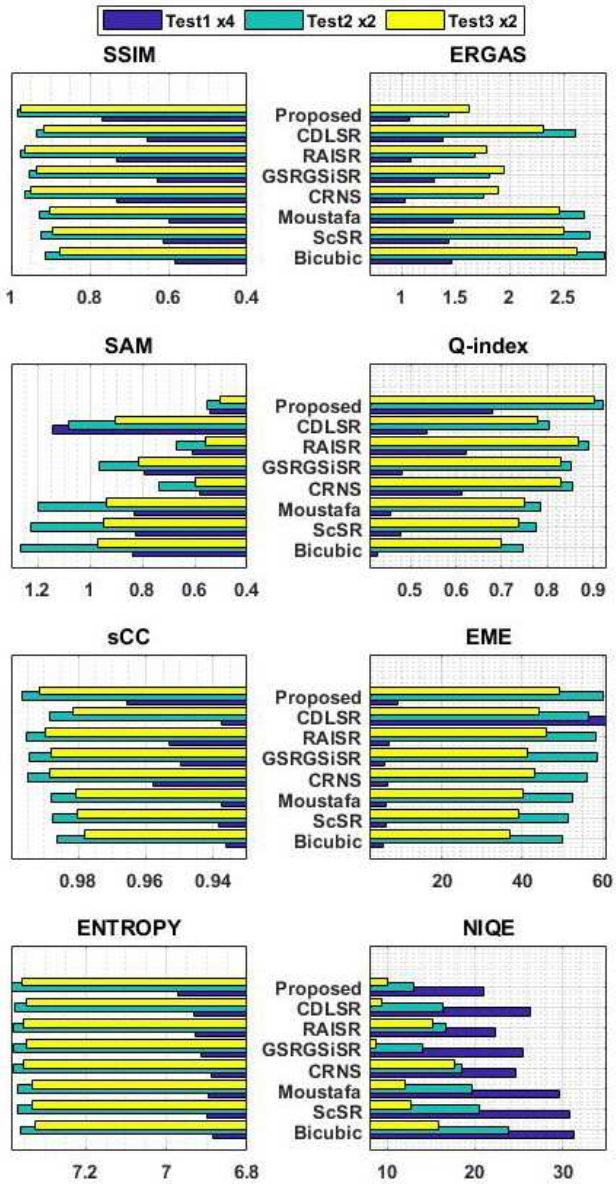
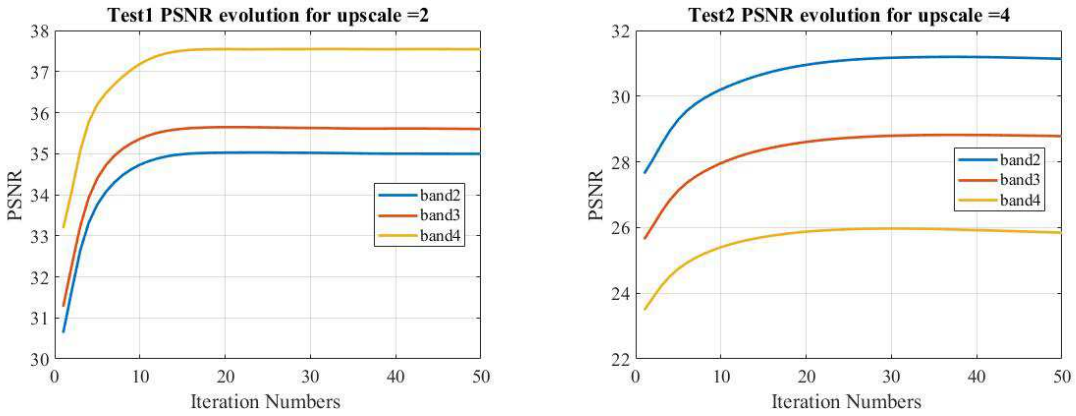


Figure 5.11: Comparison of SSIM, ERGAS, SAM, Q-index, sCC, EME, Entopy and NIQE values for Test1, Test2 and Test3 images corresponding to upscale ratios 4, 2 and 2, respectively

5.5.1.6 Convergence test of the proposed algorithm

An empirical test is done by plotting the PSNR values of the reconstructed images with varying iteration numbers for the algorithm. Figs. 5.12a-5.12b show the convergence plot of all the three bands i.e. band2, band3 and band4 of the LISS-IV Test1 and Test2 images, respectively. It can be observed that in all the bands, the PSNR value is increasing gradually until it becomes constant after a certain number of iterations. The convergence behavior of Test3 image with respect to upscale values 2, 3 and 4 are also very stable. We consider maximum iteration number as 30 to reduce the computation time of the algorithm as it is observed that the PSNR increases up to 20-30 iterations and after that it becomes nearly flat.



(a) (b)

Figure 5.12: Convergence analysis of the proposed method shown in terms of progression of PSNR values with respect to the iteration number. (a) for 'Test1' image with upscale ratio 2; (b) for 'Test2' image with upscale ratio 4

5.5.1.7 Computational complexity and runtime comparison

Major computation comes from two aspects- firstly, learning the K-SVD dictionary and solving the PSR problem, and secondly solving the GSR problem. If K is the number of atoms in the dictionary then PSR needs $\mathcal{O}(K^3)$ operations. If N is the total number of pixels, t_s is the time required for searching similar patches for any given patch, the complexity of SVD dictionary learning from each group is $\mathcal{O}(m \times k)$. Therefore, the overall complexity of the GSR part can be given as $\mathcal{O}(N(m \times k + t_s))$.

The overall computational cost of the JSR problem is $\mathcal{O}(K^3 + N(m \times k + t_s))$. The execution time of the proposed method is graphically compared with those of other methods in Fig. 5.13. In this work, since we have performed band-wise SR, we compute the execution time for the reconstruction of a test image as the average time over all of its bands. Here, the first plot (top) compares the average SR reconstruction time taken by different methods for equal number of test images randomly taken from each of the dataset. Also, to get an idea of the underlying experimentation on different test images, we have shown a comparison of average execution times against the three test images of LISS-IV for different methods (bottom part).

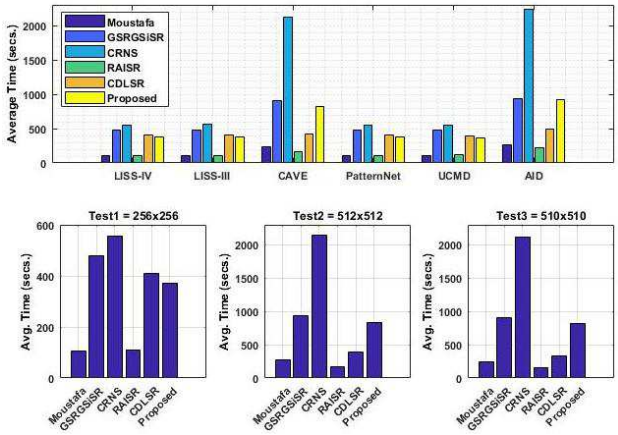


Figure 5.13: Average reconstruction time of different methods (corresponding to upscale ratio 2) for equal number of randomly selected images from different datasets (top), and average band-wise reconstruction time of different methods for the three test images of LISS-IV dataset.

It can be noticed that although the execution time of the proposed method is little higher, it is faster than its closest competitors: GSRGSiSR, CRNS and CDLSR. RAISR outperforms the proposed method in terms of execution time, but its reconstruction quality is poor. This is because the objective of the former is only to achieve rapid reconstruction using a simple mapping from LR-to-HR without targeting high accuracy in SR reconstruction. We have not compared the proposed method with the ScSR as the dictionary training in the latter is done globally from similar external HR datasets. The coupled dictionary training in ScSR takes 2-3 hours, while in CDLSR it takes 30-35 minutes [94]. Since, the results of the proposed SR method are highly promising, in future, parallel processing-based approach using

general-purpose graphics processing unit (GP-GPU) may be explored for achieving speed up in execution.

5.5.2 Comparison with deep learning-based SR methods

In this work, a few DL-based methods are also considered for comparing their performances with the proposed method and establish the merits of the proposed method over the DL-based methods, particularly, in remote sensing. Since, DL-based methods require large datasets for network learning, we consider few publicly available land-use datasets, namely, PatternNet[1], UC Merced Dataset (UCMD)[2] and the Aerial Image Database (AID)[3]. PatternNet consists of 38 classes of Google Earth images of size 256×256 and spatial resolution ranging from 0.062 m to 4.693 m[132]; UCMD has 21 classes and images are of size 256×256 with 0.3 m spatial resolution[119]; AID imagery have total 30 classes and measuring 600×600 pixels and spatial resolution in the range 0.5 m to 0.8 m[110].

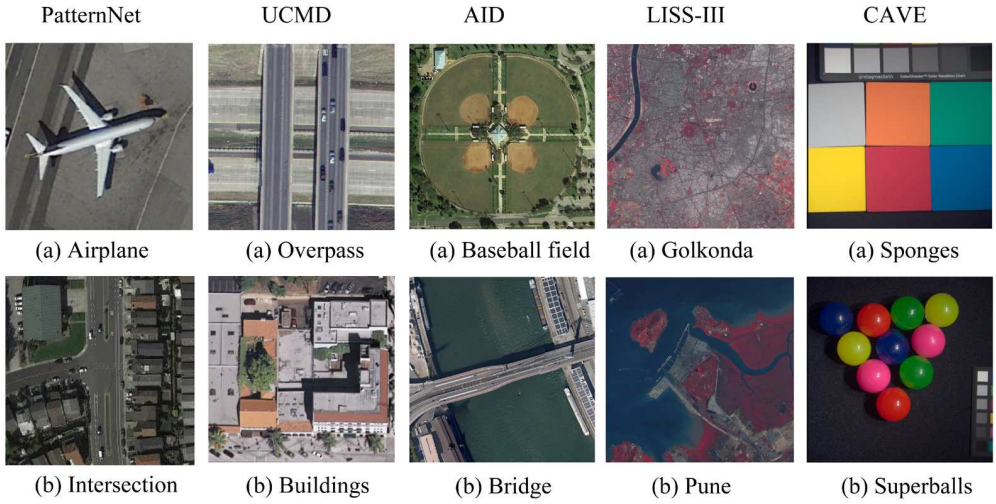


Figure 5.14: Example images of different test datasets considered for validation of this proposed work and the DL-based methods, column-wise from left to right: PatternNet, UC Merced, AID, LISS-III and CAVE.

We have also tested the two best performing DL-based and the sparse represen-

[1]PatternNet is available at `https://sites.google.com/view/zhouwx/dataset`
[2]UCMD is available at `http://weegee.vision.ucmerced.edu/datasets/landuse.html`
[3]AID is available at `https://captain-whu.github.io/AID/`

tation based methods of above experiments on the CAVE[4] MS dataset along with self-procured LISS-III and LISS-IV datasets. The LISS-III imagery have four spectral bands namely, Green (B2: 0.52-0.59 μm), Red (B3: 0.62-0.68 μm , near infrared i.e. NIR (B4: 0.77-0.86 μm) and the mid infrared (MIR) (B5: 1.55-1.70 μm), respectively. While, the CAVE MS dataset consists 32 scenes and each image includes 31 spectral bands (400 nm -700 nm, with 10 nm steps) each having a size of 512×512 [120]. The CAVE data is commonly used for validations of spectral reconstruction in MS image SR [65, 120]. Figure 5.14 shows two representational images from each category considered for this comparison. The LISS-III test images 'Golconda' and 'Pune' are of size 256×256 and the LR versions corresponding to downsapmling factor 2 are generated from them using a blurring operator as discussed in subsection 5.5.1.2. Also, the LR versions of all the test images are generated using the same approach.

5.5.2.1 Evaluation of DL-based methods

A few state-of-the-art DL-based SR methods, namely, CFSRCNN[102], MHAN[125], SAN[28], VDSR[55] and SRCNN[34] are simulated to get a fair idea which among them is performing better on the publicly available standard remote sensing datasets. Since these images are in RGB format, we first transform them into YCbCr to apply SISR algorithm on the Y-channel. First, we have re-trained the respective DL models by selecting 80-90% images of each dataset and about 10% images for validation. We select a few test images from the above datasets randomly such that they are not included in the training. To provide a fair comparison with the non-DL methods, LR test images for evaluating the DL-based methods are generated by the same procedure. While preparing the training dataset for LISS-IV, equal number of RoI images of size 256×256 are selected from the original images (of size, say 10000×10000). RoI images selected from individual bands are based on their entropy and variance values. Next, we have tested the methods for 5-8 images

[4]CAVE data is available at `https://www.cs.columbia.edu/CAVE/databases/multispectral/`

from each dataset and shown the average results. Here, we have calculated only the PSNR and SSIM values for simplicity. Additionally, in another experiment, we have selected the five best performing methods from the DL, and sparse representation categories and results are compared with the proposed method corresponding to LISS-III, LISS-IV and the CAVE datasets.

The target HR image of each MS LR-band is individually reconstructed using the proposed JAMiSR algorithm, and finally individual bands are merged to obtain a false colour RGB image for visual representation. The visual outputs by different DL-based methods are compared with that of the proposed method in Fig. 5.15. From the visual interpretation of the reconstructed images, it is observed that the DL-based methods can not overcome the effects of image degradations i.e blurring and downsampling, which are considered in SR works. However, the proposed method is able to reduce these effects to a large extent through adaptive dictionary learning and ADMM-based JSR algorithm. Magnified versions of RoI sub-images of the test images, Airplane, Buildings, and Bridge are shown in Fig. 5.15 for better visual interpretation. The reconstructed images are also sharper in case of the proposed method presenting the fine details and edges in a less artifact manner and reduced haziness in backgrounds. For quantitative evaluation, two metrics, PSNR and SSIM are calculated in each case and displayed below the reconstructed images of each method in Fig. 5.15. The PSNR and SSIM comparisons for the other test images of datasets, PatternNet, UC Merced, and AID is done and shown in Table 5.3. We can observe that the average PSNR and SSIM of the proposed method is high for all the test images of the three public datasets. The improvements in PSNR are in the range of 2-3 dB, while SSIM values are in the range of 0.1-0.15 approximately.

Table 5.3: Performance evaluation of deep learning based methods and the proposed method for publicly available land cover datasets, PatternNet, UC Merced and AID images. Here, PSNR (in dB) and SSIM measures are compared for upscale ratios 2, 3 and 4 respectively.

	Test Images	SRCNN[34]			VDSR[55]			SAN[28]			CFSRCNN[102]			MHAN[125]			Proposed		
		×2	×3	×4	×2	×3	×4	×2	×3	×4	×2	×3	×4	×2	×3	×4	×2	×3	×4
		PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM	PSNR/ SSIM
PatternNet	Airplane	31.85 0.856	30.07 0.796	28.01 0.738	33.43 0.885	30.42 0.821	28.51 0.770	33.64 0.908	31.31 0.856	29.33 0.810	32.16 0.873	30.04 0.815	29.29 0.791	33.23 0.901	30.65 0.844	28.96 0.797	**39.34** **0.950**	**34.07** **0.881**	**30.83** **0.859**
	Cemetery	27.06 0.611	26.09 0.549	25.32 0.481	27.94 0.686	**26.31** 0.555	**25.57** 0.493	27.62 0.674	26.08 0.547	25.51 0.494	27.31 0.658	25.75 0.513	25.47 0.484	27.55 0.670	26.05 0.541	25.45 0.491	**28.36** **0.894**	24.92 **0.765**	23.45 **0.595**
	Ferry terminal	25.45 0.776	23.58 0.683	21.73 0.603	25.71 0.813	23.12 0.699	21.64 0.623	25.84 0.826	23.64 0.744	21.95 0.672	25.46 0.807	22.95 0.708	21.99 0.663	25.67 0.821	23.39 0.732	21.81 0.664	**29.05** **0.907**	**24.63** **0.790**	**22.22** **0.764**
	Intersection	26.55 0.697	25.34 0.602	23.97 0.507	27.21 0.708	25.56 0.599	**24.43** 0.514	27.12 0.712	25.58 0.609	24.38 0.521	26.81 0.693	25.04 0.559	24.36 0.507	27.07 0.708	25.51 0.599	24.32 0.511	**28.31** **0.788**	**25.72** **0.641**	24.26 **0.621**
	Mobilehomepark	27.46 0.793	25.51 0.696	23.34 0.615	29.28 0.855	25.91 0.732	23.97 0.633	29.07 0.863	26.24 0.762	24.47 0.681	28.11 0.836	24.98 0.716	24.48 0.663	28.82 0.859	25.97 0.754	24.12 0.662	**34.28** **0.963**	**29.70** **0.907**	**26.17** **0.815**
	Nursing home	24.84 0.681	23.47 0.599	21.91 0.489	25.41 0.721	23.52 0.606	22.34 0.517	25.64 0.737	23.72 0.645	22.52 0.568	25.17 0.712	23.07 0.593	**22.53** 0.544	25.46 0.731	23.55 0.631	22.29 0.547	**27.29** **0.848**	**24.14** **0.734**	22.34 **0.657**
	Wastewaterplant	30.49 0.811	28.67 0.730	26.55 0.638	32.38 0.863	29.17 0.751	27.31 0.671	32.57 0.878	29.34 0.784	**27.38** 0.706	31.51 0.855	28.29 0.731	27.24 0.688	32.37 0.876	29.06 0.775	27.23 0.699	**36.55** **0.958**	**32.97** **0.879**	26.55 **0.829**
	Average	27.67 0.747	26.10 0.665	24.40 0.581	28.76 0.790	26.28 0.680	24.82 0.603	28.78 0.799	26.55 0.706	25.07 0.636	28.07 0.776	25.73 0.662	25.05 0.620	28.60 0.795	26.31 0.696	24.88 0.624	**31.88** **0.901**	**28.02** **0.796**	**25.11** **0.737**
UC Merced	Denseresidential	25.01 0.726	23.69 0.639	22.46 0.562	25.82 0.765	24.23 0.671	23.05 0.594	26.45 0.801	24.64 0.711	23.36 0.641	25.42 0.735	24.07 0.661	23.38 0.627	26.26 0.792	24.46 0.703	23.25 0.635	**29.28** **0.946**	**26.30** **0.899**	**24.16** **0.758**
	Mediumresidential	28.43 0.821	26.43 0.719	24.73 0.622	29.63 0.847	27.30 0.752	25.53 0.657	30.33 0.875	27.59 0.774	25.97 0.693	29.23 0.835	26.87 0.727	26.15 0.684	30.13 0.869	27.42 0.768	25.78 0.684	**34.95** **0.962**	**29.77** **0.894**	**27.05** **0.797**
	Mobilehomepark	20.15 0.601	18.62 0.447	17.27 0.317	19.93 0.603	18.42 0.444	17.37 0.326	21.18 0.681	**19.21** 0.518	17.99 0.404	20.19 0.614	18.80 0.461	17.96 0.397	20.89 0.665	19.04 0.502	17.86 0.393	**22.49** **0.833**	19.10 **0.646**	**18.11** **0.491**
	Overpass	27.41 0.721	25.71 0.641	24.27 0.544	28.11 0.762	26.21 0.654	24.79 0.564	28.78 0.781	**26.98** 0.681	**25.68** 0.617	28.19 0.763	26.05 0.636	25.41 0.581	28.63 0.776	26.79 0.676	25.15 0.583	**30.81** **0.868**	26.93 **0.724**	24.86 **0.652**
	River	23.54 0.655	22.14 0.521	21.23 0.442	23.82 0.667	22.43 0.544	21.58 0.465	23.98 0.687	22.39 0.549	21.51 0.472	23.54 0.658	22.05 0.515	**21.59** 0.475	23.89 0.681	22.34 0.547	21.45 0.467	**25.93** **0.838**	**22.76** **0.654**	21.34 **0.592**
	Average	25.23 0.717	23.69 0.612	22.31 0.518	25.83 0.741	24.08 0.630	22.82 0.543	26.64 0.780	24.63 0.666	23.35 0.588	25.68 0.730	23.95 0.618	23.32 0.574	26.44 0.771	24.45 0.659	23.13 0.576	**29.20** **0.900**	**25.42** **0.785**	**23.41** **0.675**
AID	Baseballfield	29.72 0.822	28.09 0.754	26.39 0.689	30.34 0.832	28.22 0.757	26.73 0.696	30.92 0.854	28.48 0.774	27.09 0.718	30.31 0.831	28.27 0.761	26.93 0.706	30.63 0.846	28.32 0.767	26.94 0.710	**34.83** **0.972**	**29.77** **0.931**	**27.51** **0.902**
	Bridge	28.33 0.801	26.98 0.752	25.45 0.703	28.99 0.811	27.17 0.753	25.64 0.696	29.33 0.827	27.33 0.767	26.02 0.720	28.56 0.803	27.07 0.755	25.80 0.709	28.97 0.819	27.05 0.760	25.82 0.713	**31.71** **0.930**	**28.85** **0.911**	**26.97** **0.877**
	Church	28.15 0.796	26.21 0.710	24.23 0.635	28.79 0.809	26.35 0.718	24.42 0.655	29.44 0.865	26.84 0.780	25.16 0.707	28.69 0.828	26.54 0.756	24.90 0.685	29.14 0.855	26.62 0.768	24.88 0.691	**32.85** **0.967**	**28.35** **0.915**	**25.81** **0.891**
	Commercial	24.75 0.773	22.84 0.654	20.69 0.528	25.51 0.785	23.07 0.654	21.03 0.497	25.71 0.777	23.33 0.657	21.72 0.550	25.12 0.741	23.05 0.626	21.44 0.526	25.45 0.770	23.06 0.637	21.43 0.529	**28.42** **0.900**	**24.28** **0.868**	**21.83** **0.773**
	Industrial	28.67 0.816	26.87 0.730	24.89 0.637	29.43 0.831	27.01 0.729	25.37 0.624	30.01 0.850	27.20 0.730	25.49 0.638	29.45 0.825	26.93 0.714	25.32 0.629	29.89 0.843	27.06 0.722	25.38 0.633	**35.21** **0.965**	**30.08** **0.912**	**27.05** **0.893**
	Mediumresidential	24.98 0.685	23.39 0.544	21.96 0.428	25.49 0.691	23.49 0.536	22.09 0.408	25.69 0.682	23.63 0.533	**22.53** 0.446	25.15 0.648	23.47 0.514	22.39 0.428	25.41 0.668	23.46 0.520	22.34 0.432	**28.21** **0.893**	24.11 **0.874**	22.23 **0.763**
	Resort	24.51 0.699	23.32 0.595	22.11 0.509	24.88 0.706	23.35 0.590	22.25 0.489	24.95 0.706	23.42 0.595	**22.46** 0.518	24.62 0.678	23.32 0.582	22.42 0.509	24.77 0.692	23.31 0.586	22.38 0.510	**26.23** **0.880**	**23.47** **0.910**	22.07 **0.837**
	School	28.76 0.816	26.83 0.736	24.55 0.637	29.39 0.826	27.01 0.737	24.94 0.643	30.11 0.865	27.30 0.762	**25.52** 0.675	29.74 0.842	27.16 0.750	25.32 0.664	30.02 0.858	27.23 0.757	25.37 0.668	**35.31** **0.957**	**29.75** **0.963**	22.62 **0.913**
	Average	27.23 0.775	25.69 0.684	23.78 0.595	27.85 0.786	25.70 0.684	24.06 0.588	28.27 0.803	25.94 0.699	24.49 0.621	29.70 0.774	25.72 0.682	24.31 0.607	28.03 0.793	25.76 0.689	24.31 0.610	**31.22** **0.933**	**27.33** **0.906**	**24.51** **0.860**

In a second set of experiments, we have selected five best performing DL- and sparse representation-based methods and compared with the proposed method using LISS-III, LISS-IV and CAVE MS datasets. The reconstructed images by different methods are shown in Fig. 5.16 for the three MS image datasets. From the visual results, we can see that the proposed method is able to provide best visual results with higher PSNR and SSIM values. The RoIs selected contain clear structural information of urban areas, industrial and other important places. It is because the proposed method can reproduce local information the best along with the maximum removal of undesired artifacts. In some experiments, the results of the GSRSiSR method performs closer to the proposed method, but this is only when the test image is either having less structures or a high uniform background e.g. in LISS-III second image and the LISS-IV first image. Moreover, the GSRSiSR method induces higher smoothing and fails to bring out small objects distinctly as compared to the proposed method. For LISS-IV test images 'Test1' and 'Test3', the performances of all the three DL-based methods are poor as we can see distorted objects, highly suppressed by the smoothing effects. In case of the CAVE dataset, the edges of the Sponge image generated by the proposed method is sharply visible, while a uniform background is also being restored.

The objective evaluation of the reconstructed images are carried out using PSNR and SSIM and displayed below the output images of each method in Fig. 5.16. Few other quantitative metrics like, ERGAS, SAM, Q-index, and Entropy are also evaluated and compared graphically in Fig. 5.17 for the two test images of LISS-III dataset. The spectral information of all the images also remain intact in case of the proposed method without suffering any color changes. In another experiment, we have performed an additional comparison of the sparse-dictionary models with DL-models on a common test set. For MS datasets, we apply the SR algorithm on every LR band image separately. Here in Fig. 5.18, SR reconstruction results are shown for band2 and band3 of the Test1 and Test2 images (as shown in Fig. 5.7) with upscaling factors 2 and 4, respectively. From visual inspection, it is clear that the proposed method provides the best perceptual quality of reconstructed images. Moreover, the PSNR and SSIM values of the proposed method shown in respective

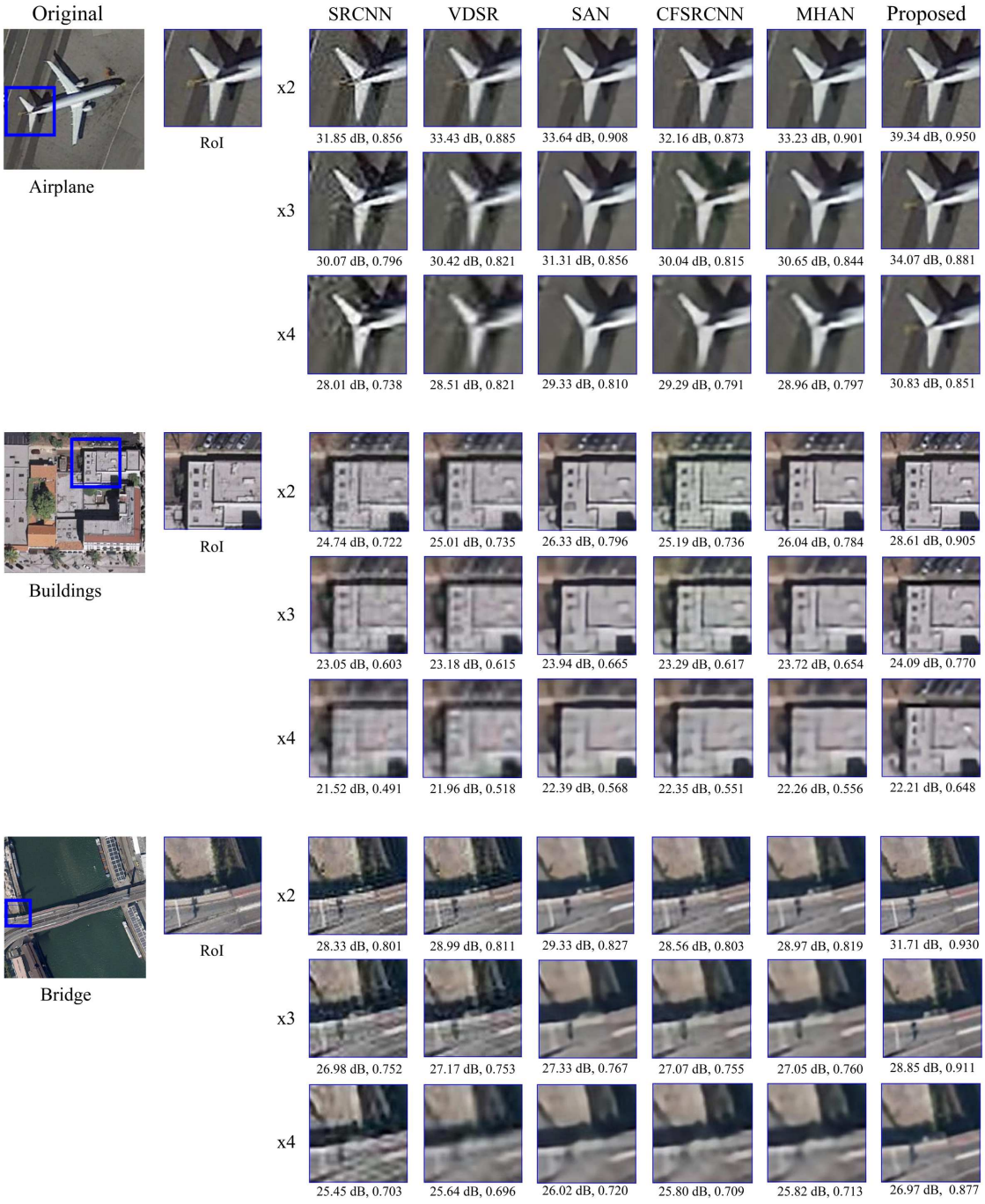


Figure 5.15: Comparison of reconstructed images of DL-based methods with the proposed method. From top to bottom: first three rows are for PatternNet (Airplane), next three rows are for UC Merced (Buildings) and the last three rows are for AID image (Bridge), respectively corresponding to upscaling factors 2, 3, and 4. Column-wise from left to right: original image, region of interest (RoI) image, results of different methods: SRCNN, VDSR, SAN, CFSRCNN, MHAN, and proposed method, respectively.

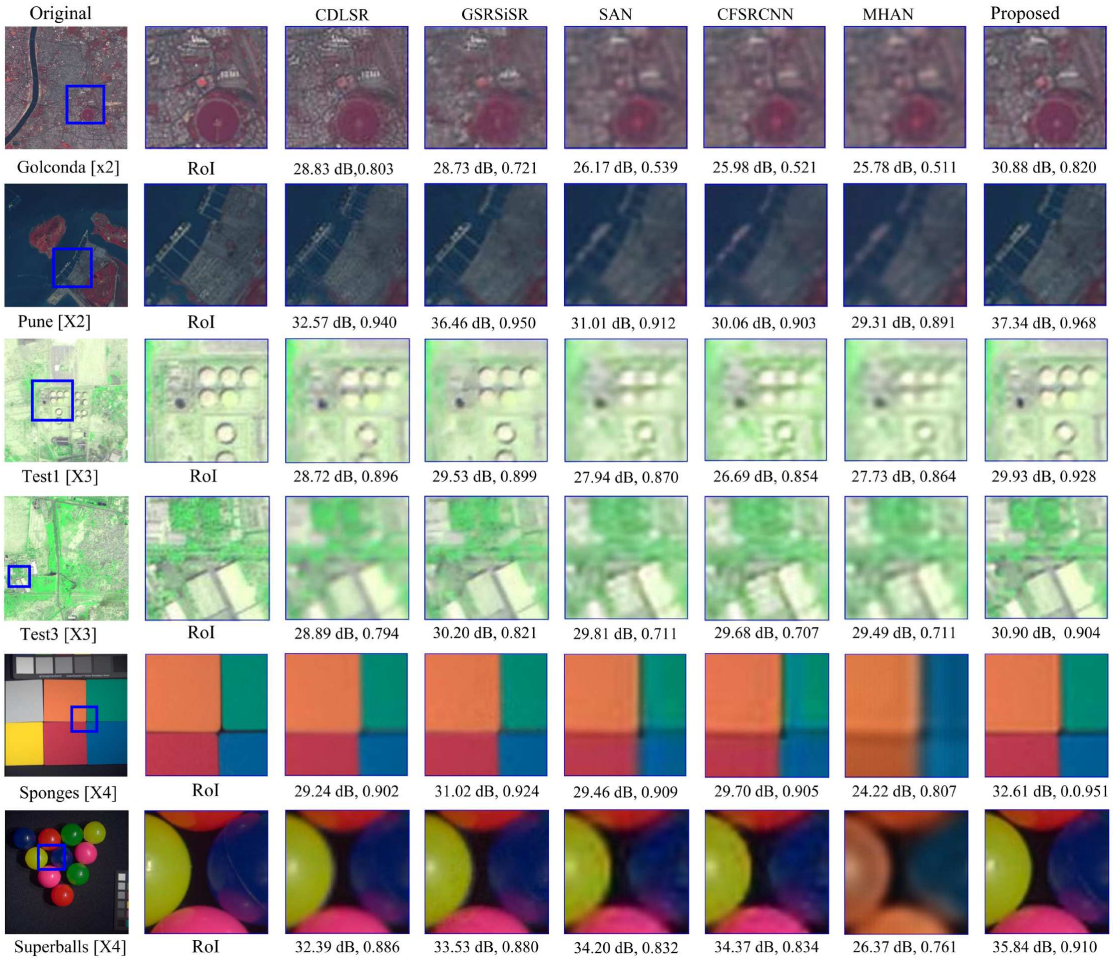


Figure 5.16: Visual results of LISS-III, LISS-IV, and CAVE MS images shown for upscale ratios 2, 3 and 4 respectively. Column-wise from left to right: original, RoI, reconstructed images using CDLSR, GSRGSiSR, SAN, CFSRCNN, MHAN and the proposed method, respectively. Row-wise from top to bottom: first two rows represent LISS-III images corresponding to upscale ratio 2, next two rows represent LISS-IV images corresponding to upscale ratio 3, and finally the last two rows represent the CAVE images corresponding to upscale ratio 4.

figures indicate its superiority over the other methods. The reasons behind the poor performances of the DL-based methods in this work are, firstly, consideration of the blurring in image degradation model, secondly, smaller datasets with comparatively less (approximately 100) training images. Moreover, the proposed JAMiSR is capable of reconstructing an HR image from the given input image only through self-adaptive group dictionary learning and optimization-based reconstruction, which is very less time consuming ($\approx$ 100 seconds) for training compared to the training of DL-based methods ($\approx$ 5-8 hours). However, it may be mentioned here that the DL-based image reconstruction alone can be very fast; it takes only few seconds

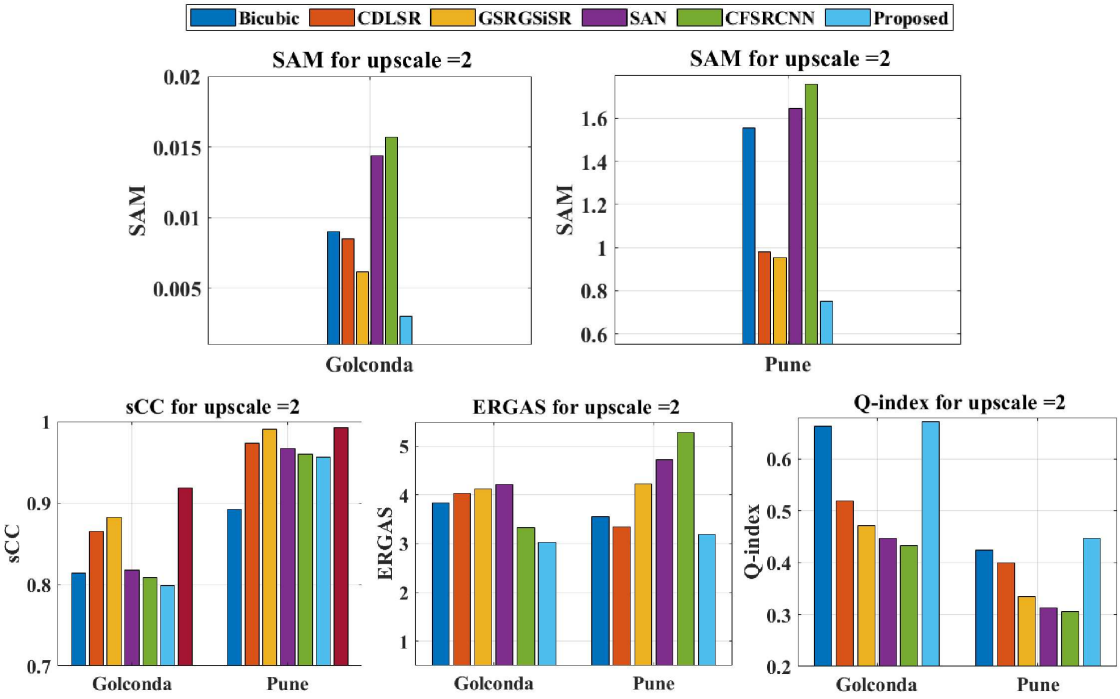


Figure 5.17: Comparison of quantitative metrics of DL-based and sparse representation methods for SR of LISS-III test images 'Golconda' and 'Pune' respectively w.r.t. upscale ratio 2.

provided sufficient numbers of DL-based super-resolution models are being trained offline from large number of available datasets. An estimate of average inference time for the DL-models to perform 2× SR reconstruction on a single band image of size 256×256 is given in Table 5.4. A graphical comparison of sequential time of the proposed method with respect to other sparse dictionary-based methods is also shown in Fig. 5.13, where it is clearly indicated that the former is able to surpass other methods. In CPU, it takes ≈ 350–400 seconds for learning dictionaries as well as SR image reconstruction. However, the run-time of the parallel version of the proposed algorithm is reasonably less (≈ 15 seconds) and comparable to the DL inference time. However, the proposed method has distinct advantage over other sparse- and DL-models in terms of output quality.

Table 5.4: Comparison of inference time of the DL-methods and the proposed method.

Methods	CFSRCNN	VDSR	SAN	CFSRCNN	MHAN	Proposed (Sequential)	Proposed (Parallel)
Time (in sec)	0.4175	0.0170	5.821	0.0172	1.754	384	14.25

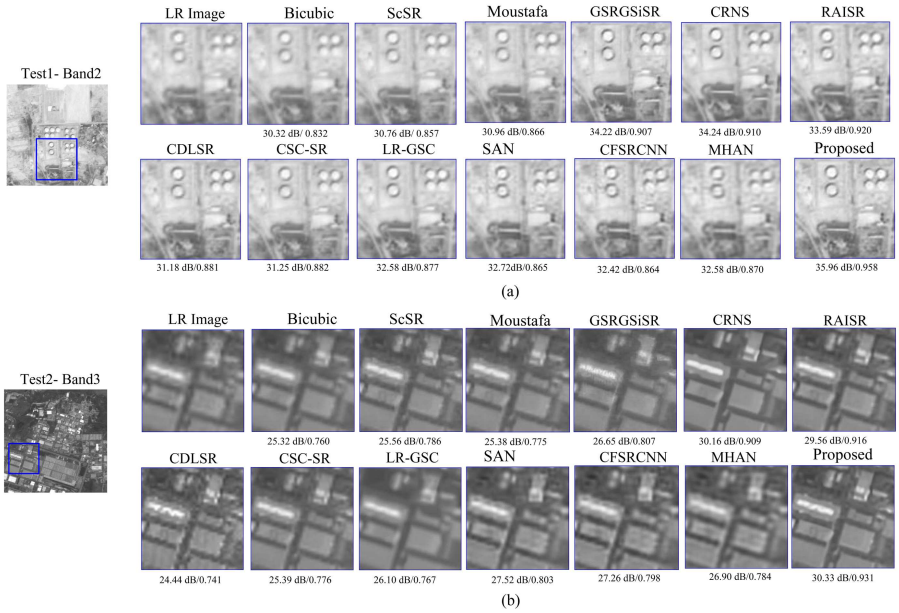


Figure 5.18: Visual comparison of the SR outputs for LISS-IV dataset. (a) Rows 1–4 "Test1" band2. (b) Rows 1–2 "Test2" band3 images for upscale ratios 2 and 4 using different sparse dictionary- and DL-based methods. The PSNR and SSIM measures for all the visual outputs are provided at the bottom of image for better interpretation.

5.5.3 Remote sensing applications of the proposed method

5.5.3.1 Spectral graph comparisons

In a MS image, for the same area of view, different bands exhibit different reflectance values. Therefore, different bands convey information of different features/objects in the image e.g. blue band specifies deep water (50 m), atmosphere, etc.; green band specifies vegetation, deep water (30 m), etc.; red band specifies the man-made objects, soil, deep water (9 m), etc., and NIR band specifies geological features, forest, fires, etc..

Table 5.5: Imaging features of MS image bands

Band	Imaging
Blue	Deep water (50 m), atmosphere, etc.
Green	Vegetation, deep water (30 m), etc.
Red	Man-made objects, soil, deep water (9 m), etc.
NIR	Geological features, forest, fires, etc.

A spectral profile characterizes properties of different band images; it is a plot of pixel intensity values corresponding to the wavelength of each band. Fig. 5.19 shows the spectral differences between the original and reconstructed images of different sparse representation-based and the proposed methods for the LISS-IV 'Test1' and LISS-III 'Pune' images. From the spectral profiles of all the reconstructed images, it is observed that the proposed method is able to regenerate the HR image better by providing the closest approximation of the spectral profile with respect to the original. On the other hand, methods, like ScSR and CRNS tend to deviate from the spectral graphs of the original images across all the wavelengths indicating their limitation in providing better spectral estimations. Therefore, characterization of terrestrial objects for remote sensing analysis using spectral profiling is likely to be more benefitted from the HR images generated by the proposed method.

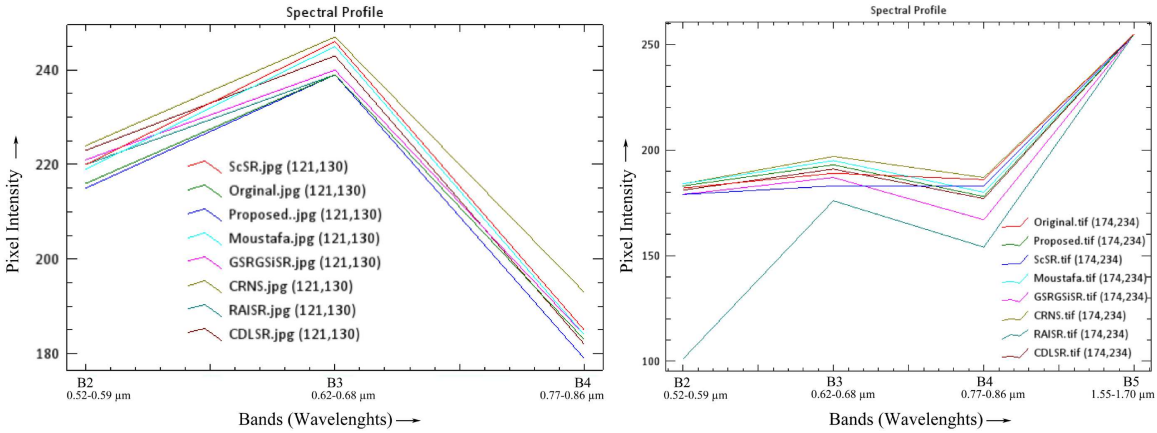
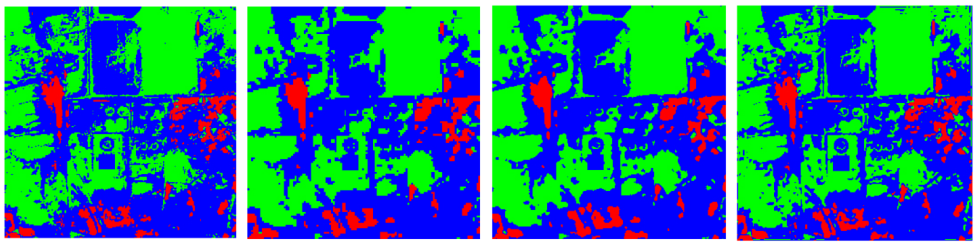


Figure 5.19: Spectral differences between the original and reconstructed images by different methods

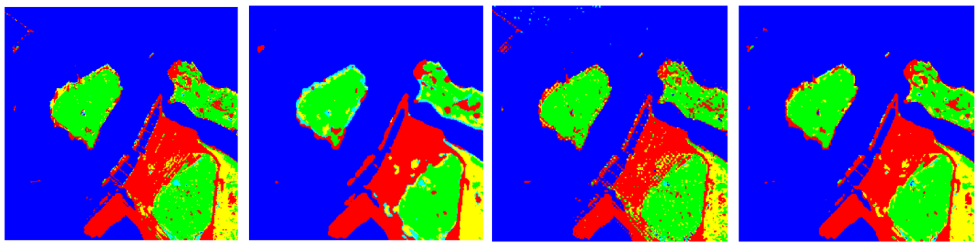
5.5.3.2 Classification performance analysis

We have performed classification on the HR reconstructed images obtained from different methods to interpret the image contents and analyze the effects of SR algorithm on a given LR MS image. ENVI classic 5.1 is used for classification and analyzing the results. With the help of expert defined training classes, supervised classification, like support vector machine (SVM) can cluster pixels from the test images. We have conducted the experiment with two test images having different



Classes	Original			LR Image			ScSR			Proposed		
	Pixels	Area (Sq. Km)	Acc. (%)	Pixels	Area (Sq. Km)	Acc. (%)	Pixels	Area (Sq. Km)	Acc. (%)	Pixels	Area (Sq. Km)	Acc. (%)
Bare Land	27,243	0.9164	--	26,916	0.9054	98.79	27,731	0.9328	98.20	27,276	0.9175	99.87
Buildings	33,689	1.1332	--	33,872	1.1394	99.45	34,219	1.1511	98.42	33,639	1.1316	99.85
Vegetation	4,604	0.1548	--	4,748	0.1596	96.87	4,586	0.1542	99.60	4,611	0.1550	96.84
Average Accuracy						98.37%			98.74%			99.85%

(a) LISS-IV 'Test1'



Classes	Original			LR Image			ScSR			Proposed		
	Pixels	Area (Sq. Km)	Acc. (%)	Pixels	Area (Sq. Km)	Acc. (%)	Pixels	Area (Sq. Km)	Acc. (%)	Pixels	Area (Sq. Km)	Acc. (%)
Infastructure	8,915	4.923	--	8,998	4.968	99.07	8,957	4.946	99.53	9,144	5.049	97.49
Forest	10,339	5.709	--	8,913	4.921	86.20	10,755	5.939	96.13	10,062	5.556	97.32
Sea water	42,800	23.63	--	42,806	23.63	99.98	42,762	26.37	99.91	42,842	23.65	99.90
Vegetation	3,165	1.747	--	3,406	1.880	92.92	2864	1.581	90.48	3,274	1.807	96.67
Water body	317	0.175	--	1,413	0.780	22.43	198	0.109	62.46	214	0.118	67.50
Average Accuracy						80.12%			89.70%			91.77%

(b) LISS-III 'Pune'

Figure 5.20: Results of supervised classification performed on LISS-IV 'Test1' image (a) and LISS-III 'Pune' image (b). From left to right: original image, LR image, and reconstructed images of ScSR and the proposed method.

number of classes. In the first test image, RoIs are labeled as bare land (green), buildings (blue) and vegetation (red), while in the second image they are labeled as infrastructure (red), forest (green), sea water (blue), vegetation (yellow), and water body (cyan). Multiple RoIs are selected under each class for training with the test images. Figs. 5.20(a) and 5.20(b) show the classification results obtained for the two MS test images from LISS-IV and LISS-III, respectively.

It is significantly noticeable that the proposed method has the most similar classified regions to that of the LR image. On the other hand, some regions are not distinctly classified in the conventional sparse representation method, like ScSR, when

compared to the proposed method. For quantitative evaluation, the corresponding pixel counts for each class are also shown in the figure. An average accuracy of pixel counts is also computed across different classes for both the images. It is observed that maximum accuracy is obtained in case of the proposed method, while in case of the ScSR, the pixel count is quite different from both the original and LR images.

We also calculate the land cover area under each classified region from their pixel counts and standard per-pixel resolution (PR) of the sensor. Since, LISS-IV has 5.8 m spatial resolution, so each pixel covers an area of 33.64 m^2 and thus Test1 image will have a coverage of 2.2046 km^2, which is spread over the three classes. Similarly, LISS-III has 23.5 m spatial resolution, resulting into a pixel area of 552.25 m^2 and total image coverage of 36.19 km^2, distributed among the five land cover classes i.e. infrastructure, forest, sea water, vegetation and water body as depicted in Fig. 5.20b. This further shows that the land cover area classification resulting from the proposed method is more accurate and at par with the original images.

5.5.4 Parallel implementation using GPGPU hardware

The computation time of the proposed JAMiSR appears to be more because the coupled dictionary training, patch sparse coding of PSR as well as non-local similar patch searching and group sparse coding via SVD of GSR increases the computational cost. Moreover, the ADMM algorithm needs to solve both the PSR and GSR subproblems in each iteration. Therefore, in the proposed algorithm, we have used GPGPU hardware to accelerate the execution time of the proposed JSR-based MS image SR using the CUDA programming model in the MATLAB environment.

An analysis of the CPU execution time is done to make an assessment of the potentially parallelizable sections of the proposed algorithm. As shown in Table 5.6, the major time consuming parts in the sequential program are found to be PSR (K-SVD dictionary training and patch sparse optimization) and GSR (block matching, group sparse coding via SVD).

Table 5.6: CPU run-time for different sections of the proposed SR algorithm.

Function	Run time (in secs)
GSR	167.93
PSR	33.20
ℓ_1-feature-sign-search	108.67
non-local patch search	35.67
mean	4.025

The parallel computing toolbox of MATLAB allows to run computations on NVIDIA CUDA-enabled GPU(s) in workstations using built-in functions or generating CUDA codes from MATLAB. Computationally heavy functions may be converted into MEX file that contains CUDA code and execute them in GPU by calling through MATLAB. A CPU-GPU hybrid environment is utilized to speed-up the code executions in this work. Here, the PSR is implemented using CUDA-mex and the other functions are implemented using C++ mex functions in the MATLAB parallel computing environment.

We use GPU coder to convert the existing sequential functions by duly modifying them. Since, in CPU all the variables are defined in the global memory, so while we convert them using GPU coder it can not generate the MEX file. To run in multiple blocks/threads, the variables are needed to be declared in shared memory and accordingly the code is modified to properly return the outputs back to CPU. GPU coder allows to automatically define the input variables types and dimensions before generating the MEX file by running the main program which calls the function to be converted. By manually setting the variable ranges to a possible maximum value will allow to use the generated MEX files for any image sizes within that limit. In this work while configuring MEX-setup we use Microsoft Visual C++ 2015 for C language compilation. The algorithm is then run on a workstation equipped with Intel Xeon Processor, Windows 10 OS, 128 GB RAM, and NVIDIA's QUADRO P5000 GPU card (Compute capability = 6.1, CUDA Toolkit = 10.1, memory = 16 GB).

In sequential version of JAMiSR, the PSR and GSR subproblems in CPU requires 4.82 seconds and 5.82 seconds, respectively in an inner loop, which in turn

Table 5.7: Comparison of proposed method's sequential and parallel implementation time for MS datasets of varying sizes.

Dataset	Input size	Output size	Seq. time (in secsonds)	Parallel time (in seconds)
LISS-III	128×128×4	256×256×4	1365	57
LISS-IV	256×256×3	512×512×3	3443	261
QuickBird	512×512×4	1024×1024×4	26463	787
LISS-IV	1024×1024×3	2048×2048×3	44478	1178

are called in an outer loop of the ADMM algorithm. Through parallelization, the proposed MEX-CUDA based implementation of the PSR takes approx. 0.70 seconds, while the GSR takes approx 0.36 seconds for each patch. Hence, the overall execution time of the algorithm is accelerated by several times. We have compared the run time of the proposed algorithm for different test MS images of sizes starting from 128×128×4 to 1024×1024×3. A time comparison for sequential vs parallel implementations of the proposed algorithm is shown in Table 5.7. It can be observed that to reconstruct a four channel MS image of 256×256, the parallel algorithm requires about 57 seconds, while its sequential counterpart takes approximately 1365 seconds. Here, we have additionally used the MATLAB parfor instruction to simultaneously process all the bands of the MS images, where each image will be reconstructed through the proposed MEX-CUDA based parallelization. A speed-up in the order of 15-35 times is achieved through the parallelization of the algorithm. The graphical comparison of band-wise average execution times in CPU vs GPU shown in Fig. 5.21 also clearly indicates the reduction of computational time by the parallel implementation of the proposed algorithm.

5.5.5 Conclusion

In this chapter, we have presented an efficient image SR method based on JSR and adaptive dictionary learning for MS remote sensing images. The non-local self similarity information of different patch-groups provide improved sparse representations over learned dictionaries. The proposed MS image SR algorithm learns two overcom-

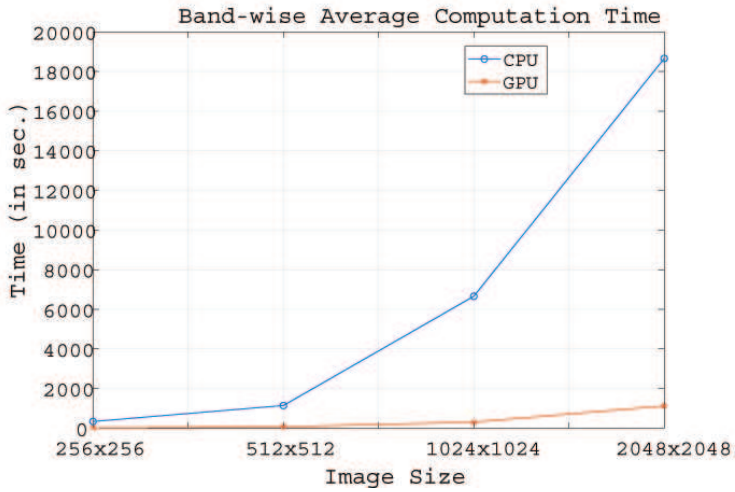


Figure 5.21: Comparison of average execution time for band-wise super-resolution of different sized images.

plete dictionaries: an adaptive group dictionary for every patch groups present in the image and an adaptive patch-based dictionary. Reconstructed images with group sparsity are enriched with both the local and non-local information. Therefore, the proposed JAMiSR method is able to show better performances, while reducing the high-computation costs of global dictionary training. Results are demonstrated both visually and quantitatively, where significant improvements are achieved over several state-of-the-art methods. Extensive simulations are also carried out for the DL-based methods using both publicly available datasets as well as some self-procured MS image datasets. On an average, the proposed method performs better in most of the experiments. Also, the high classification accuracy of the reconstructed MS images indicate that the proposed algorithm is able to improve the resolution of input LR MS images required for practical remote sensing applications.