TV-loss and Smooth L1-loss for Hyperspectral Image Reconstruction in SSR-NET

Author:
Maria Jose Rueda Montes

Supervisor:
Hao Wang
Guoxia Xu

***Abstract*—Hyperspectral image reconstruction is a technique that consists of combining information from High Resolution Multispectral Images (HR-MSI) and Low Resolution Hyperspectral Images (LR-HSI) to obtain High Resolution Hyperspectral Images (HR-HSI). This fusion method is very popular since hyperspectral camera sensors sacrifice image resolution to obtain a wider range of spectral information, generating undesirable noise in the final image. Several image fusion techniques have been developed to achieve HR-HSI. However, recent deep learning methods, particularly those based on Convolutional Neural Networks (CNNs), are achieving groundbreaking results in hyperspectral image reconstruction. This article studies the implementation of two denoising algorithms, Total Variation loss function (TV-loss) and Smooth L1-loss, as a proposal to improve the quality of the final image. To demonstrate the improvements of our implementations, three different deep learning models have been created, using SSR-NET as a reference with the following variations: 1) TV-loss loss; 2) Smooth L1-loss; 3) TV-loss combined with Smooth L1-loss. The proposed models are evaluated against the baseline established by the original SSR-NET by using five HSI datasets (Botswana, Urban, Pavia Center, Pavia University, and Kennedy Space Center), and four quality metrics (PSNR, ERGAS, SAM, and RMSE). Our results reveal that the TV-loss model is more effective for lower-resolution datasets, while the combined TV-loss and Smooth L1-loss model yields better reconstructions for higher-resolution datasets. Both mentioned models outperform the original SSR-NET in terms of image quality, as demonstrated by our evaluations.**

***Index Terms*—Convolutional neural network (CNN), Hyperspectral Image (HSI), Multispectral Image (MSI), image reconstruction, Total Variation (TV) loss function, Smooth L1-loss function.**

1 Introduction

Hyperspectral images (HSI) are characterized by the hundreds of spectral bands obtained from a scene. The high spectral resolution of these images allows good results in a large number of applications such as remote sensing and computer vision tasks [1]. The reason behind the successful results of hyperspectral imaging is related to its accurate identification of materials since each material has a different reflectance at each wavelength. Images captured with high spectral resolution and a wide spectral range offer an advantage in discriminating between different materials in a scene. [2].However, due to limitations of imaging sensors, capturing hyperspectral images often requires long exposures. This leads to images with low spatial resolution (LR-HSI). On the other hand, imaging sensors can acquire images with higher spatial resolution but with fewer spectral bands, such as multispectral images (MSI). To address this limitation and improve the quality of hyperspectral data, the concept of hyperspectral and multispectral image fusion emerged. In this way, the high spatial resolution of multispectral images (HR-MSI) can be used to reconstruct HSIs with high spatial resolution (HR-HSI). There are several approaches to HSI and MSI fusion, including pan-sharpening-based methods, matrix factorization-based methods, and tensor-based methods [2]. However, compared to these traditional methods, new research based on deep learning using convolutional neural networks (CNN) is achieving excellent results in improving the quality and performance of the final image.

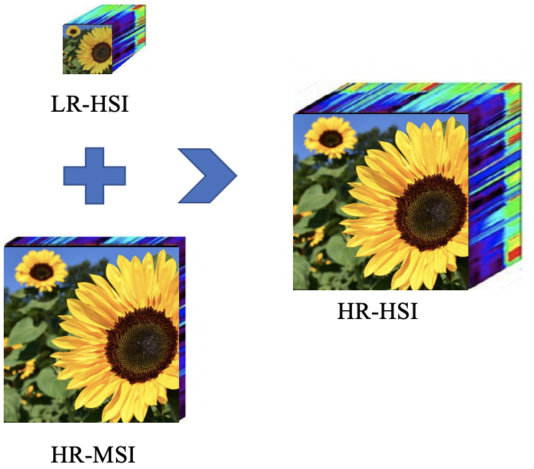


Figure 1: LR-HSI and HR-MSI fusion for obtaining HR-HSI.

The CNN models for LR-HSI and HR-MSI fusion can be divided into input-level fusion models and feature-level fusion models. In feature-level fusion approaches, first, the spatial features are extracted from HR-MSI and the spectral features from LR-HSI. Both features are fused to reconstruct the HR-HSI. In [3], Shao *et al.* proposed a model with two branches network to extract the HR-MSI and LR-HSI information separately. In [4], Yang *et al.*, use two branches for extracting the spectral features of each pixel in LR-HSI, and its correspondent spatial neighborhood in HR-MSI, to exploit the spatial correlation of the image. Then they fuse the information in a fully connected layer. In [5], Han *et al.* use a multi-scale CNN system, which gradually reduces the HR images and increases the feature sizes of LR. On the other hand, for input-level fusion models, the LR-HSI and HR-MSI images are fused before passing through the neural network to obtain the final fused HR-HSI. A frequent technique before the HR-MSI and LR-HSI fusion is to interpolate the LR-HSI to generate the same HR-MSI dimensions. The studies of Masi *et al.* [6] and Palsson *et al.* [7], uses input-level fusion, where the preliminary fused LR-HSI and HR-MSI are used as input for a super-resolution CNN, SRCNN, and PCA prior for reducing the dimensions of the fusion. In [8], Dian *et al.* present a DHSIS method to reconstruct HR-HSI, where they map the first fused HR-HSI to the reference HR-HSI using the priors learned from a deep CNN-based residual learning to regularize the fusion. A recent study by Zhang *et al.* [9] demonstrates significant HSI reconstruction improvements with their proposed SSR-NET model, which incorporates three different modules: Cross-Mode Message Inserting (CMMI), Spatial Reconstruction Network (SpatRN), and Spectral Reconstruction Network (SpecRN). The main objective of the CMMI module is to produce a preliminary concatenated image that combines the spatial and spectral information of MSI and HSI, the SpatRN module allows the reconstruction of spatial information with the implementation of spatial edge loss ($\mathcal{L}_{spat}$), and finally the SpecRN module focus on the reconstruction of spectral information using spectral edge loss ($\mathcal{L}_{spec}$). This network achieves groundbreaking results, demonstrating statistically significant improvements on seven benchmark methods (CNMF, LTTR, MSDCNN, TFNet, ResTFNet, SSFCNN, and ConSSFCNN). Given the satisfactory results of this model, in this paper, we are using SSR-NET as a baseline to improve the quality of the results when generating HSIs-HR.

Since the low resolution of HSIs is mainly influenced by the noise generated in the different bands, it is logical to think about the possible use of methods that focus on solving this problem. Total Variation (TV) regularization is one important technique used for denoising in HSIs, this method of regularization demonstrates superior performance on image processing, being an important algorithm that achieves denoise since it exploits the spatial correlation in each band and it works on the preservation of edge information and spatial smoothness [10]. He *et al.* [10] implemented total variation regularization for the hyperspectral image restoration, where they propose a mixed-noise removal method that integrates TV regularization combined with the low-rank matrix factorization model. They use the low-rank model to study spectral correlations, while TV regularization uses the piecewise smooth of HSI spatial information. In their work [11], Aggarwal *et al.* introduce spatio-spectral total variation to develop an algorithm that reduces mixed noise. In this study, they use total variation for the spatial dimension along the height and width of the image, and total variation for the spectral information. To improve Video Super Resolution (VSR), [12] Chadha *et al.* used a generative adversarial network (GANs) with a space-time approach, incorporating a four-fold (MSE, perceptual, adversarial and TV) loss function to capture the fine details of the image, choosing TV-loss as denoising loss function. Given the importance of TV loss in reducing noise in HSI images, and since it has been used successfully for this purpose in previous research, we have decided to use TV-loss together with SSR-NET model.

Since the quality of the final image depends on the choice of the loss function, in this paper we have also studied the incorporation of Smooth L1-Loss as a substitute for MSE-loss. MSE-loss function is one of the most used methods. This algorithm achieves good results when evaluated with the popular PNSR metric. However, this metric is not representative of the spatial features of the image [12]. Smooth L1-loss, also known as Huber loss, is less susceptible to outliers compared to MSE loss. This characteristic makes it particularly effective in reducing noise while processing images. In their study [13], Chlewicki *et al.* applied Smooth L1-loss in image reconstruction. This implementation effectively reduced image noise while largely preserving image contrast. The effectiveness of Smooth L1-loss stems from its property of prioritizing smaller differences between neighboring pixel values. This focus minimizes noise while preserving larger variations that often represent image edges [13].

In summary in this paper are proposed the following contributions:

1) We incorporate the TV-loss function for the first time in the CNN SSR-NET to improve the quality of the generated HSI.
2) We study the implementation of Smooth L1-loss compared to MSE-loss.
3) Our implementation incorporating TV-loss and Smooth L1-loss demonstrates improvements over the baseline SSR-NET results across all five datasets: Botswana, Urban, Pavia Center, Pavia University, and Kennedy Space Center.

2 Methodology

To understand how the proposed implementations interact with the existing architecture, this section summarizes the basics of the SSR-Net architecture. This network consists of three differentiated modules: Cross-Mode Message Inserting, Spatial Reconstruction Network with Spatial Edge Loss, and Spectral Reconstruction Network with Spectral Edge Loss. This section will also provide an explanation of the Total Variation Loss (TV-loss) and Smooth L1-loss algorithm implementations.

A. Cross-Mode Message Inserting

The objective of this module is to process the HR-MSI and LR-HSI images that are going to serve as input to the network, since in this case, the network is of the input-level fusion type. For this purpose, HR-MSI is unsampled until reaching the size of LR-HSI, and then merging both images using bilinear interpolation. In this way a hypermultiple spectral image (HMSI) is obtained, $\mathbf{Z}_{pre} \in \mathbb{R}^{H \times W \times L}$. Thus, this new set of images has the spatial and spectral information of LR-HSI and HR-MSI. Then a 3×3 and with 1 stride convolutional layer is applied to the images in order to combine spectral and spatial information.

B. Spatial Module

In this module, based on the reconstruction of spatial information, a new convolutional layer, SpatRN, with 3×3 and 1 stride characteristics is applied to $\mathbf{Z}_{pre}$, obtaining $\mathbf{Z}_{spat}$. This layer generate the edge maps of HMSI, utilizing $\mathcal{L}_{spat}$ which use the spectral information to update the weights. $\mathcal{L}_{spat}$ operates as follows

$$\mathbf{E}_{spat\_height}(i, j, k) = \mathbf{Z}_{spat}(i + 1, j, k) - \mathbf{Z}_{spat}(i, j, k), \tag{1}$$

$$\mathbf{E}_{spat\_width}(i, j + 1, k) = \mathbf{Z}_{spat}(i, j, k) - \mathbf{Z}_{spat}(i, j, k), \tag{2}$$

where i, j and k correspond to the dimensions of the data set, being the height, width, and spectral bands, respectively. For this first pair of equations, in Equation 1 it is calculate the edge map in the height dimension (i), $\mathbf{E}_{spat\_height} \in \mathbb{R}^{(H-1) \times W \times L}$, while Equation 2 obtains he edge map in the width dimension (j), $\mathbf{E}_{spat\_width} \in$

$\mathbb{R}^{H\times(W-1)\times L}$. Both equations use HMSI, $\mathbf{Z}_{spat}$ as input. Moreover, the same operations are carried out for the reference images, HR-HSI, denoted as $\mathbf{Z}$.

$$\bar{\mathbf{E}}_{spat\_height}(i,j,k) = \mathbf{Z}(i+1,j,k) - \mathbf{Z}(i,j,k), \tag{3}$$

$$\bar{\mathbf{E}}_{spat\_width}(i,j+1,k) = \mathbf{Z}(i,j,k) - \mathbf{Z}(i,j,k), \tag{4}$$

After obtaining the results of the four equations, the HMSI edge map and the reference edge map are compared by mean squared error function (MSE)

$$\mathcal{L}_{spat\_height} = \frac{\sum_{k=1}^{L}\sum_{i=1}^{(H-1)}\sum_{j=1}^{W}(\mathbf{E}_{spat\_height}(i,j,k) - \bar{\mathbf{E}}_{spat\_height}(i,j,k))^2}{2WL(H-1)}, \tag{5}$$

$$\mathcal{L}_{spat\_width} = \frac{\sum_{k=1}^{L}\sum_{i=1}^{H}\sum_{j=1}^{(W-1)}(\mathbf{E}_{spat\_width}(i,j,k) - \bar{\mathbf{E}}_{spat\_height}(i,j,k))^2}{2HL(W-1)}, \tag{6}$$

$\mathcal{L}_{spat\_height}$ and $\mathcal{L}_{spat\_width}$ represent the loss function with respect to height and width respectively, which are combined into $\mathcal{L}_{spat}$ by the following equation

$$\mathcal{L}_{spat} = \mathcal{L}_{spat\_height} * 0.5 + \mathcal{L}_{spat\_width} * 0.5, \tag{7}$$

C. Spectral Module

This module takes as input the output of the previous module, $\mathbf{Z}_{spat}$ which it applies a convolutional layer SpecRN with the same characteristics as the one used in the spatial module, kernel set in 3×3 and 1 step, where is obtained the edge map of $\mathbf{Z}_{spat}$, $\mathbf{E}_{spec} \in \mathbb{R}^{H\times W\times(L-1)}$. In this part attention is paid to reconstructing the spectral information, for which a similar loss function is used as in SpatRN, $\mathcal{L}_{spec}$ in this occasion.

$$\mathbf{E}_{spec}(i,j,k) = \mathbf{Z}_{spec}(i,j,k+1) - \mathbf{Z}_{spec}(i,j,k), \tag{8}$$

$$\bar{\mathbf{E}}_{spec}(i,j,k) = \mathbf{Z}(i,j,k+1) - \mathbf{Z}(i,j,k), \tag{9}$$

$$\mathcal{L}_{spec} = \frac{\sum_{k=1}^{(L-1)}\sum_{i=1}^{H}\sum_{j=1}^{W}(\mathbf{E}_{spec}(i,j,k) - \bar{\mathbf{E}}_{spec}(i,j,k))^2}{2HW(L-1)}, \tag{10}$$

Where $\mathbf{E}_{spec} \in \mathbb{R}^{H\times W\times(L-1)}$ is the edge map from $\mathbf{Z}$. On this occasion, since we move across one dimension, the spectral bands (k), only one loss function is calculated. At the end of the process, MSE function is used again between the reference HMSI and the HR-HSI edge maps.

Finally, $\mathbf{Z}_{spec}$ correspond with the estimated HR-HSI, which is optimized with the following loss function

$$\mathcal{L}_{opti} = \frac{\sum_{k=1}^{L}\sum_{i=1}^{H}\sum_{j=1}^{W}(\mathbf{Z}_{spec}(i,j,k) - \bar{\mathbf{Z}}(i,j,k))^2}{2HWL}, \tag{11}$$

where $\mathbf{Z}$ is the map edge of reference. The final loss function is represented as

$$\mathcal{L} = \mathcal{L}_{spat} + \mathcal{L}_{spec} + \mathcal{L}_{opti} \tag{12}$$

D. Total Variation (TV) loss function

TV-loss was proposed first time by [14], which describe the sum of the absolute differences between neighboring pixels, both horizontally and vertically. In this way TV-loss is able to attenuate the noise of the output image, since it measures the input noise and produces smoothness throughout the entire spatial dimension of the output image [12]. This algorithm studies the vertical and horizontal differences at the same time, unlike in the SSR-NET model, which uses two different functions to calculate the height and width differences independently, as it appears in the Equation 5 and Equation 6. For our experiments, TV-loss takes as input the output of the Spatial Module. TV-loss function is implemented as follows:

$$TV Loss = 2c \cdot \frac{1}{WH} \cdot \left(\frac{\sum_{ij}(x_{i,j+1} - x_{i,j})^2}{H} + \frac{\sum_{ij}(x_{i+1,j} - x_{i,j})^2}{W} \right) \tag{13}$$

where c is the TV-loss weight, x represent the pixel and H and W are the dimensions, of height and width respectively.

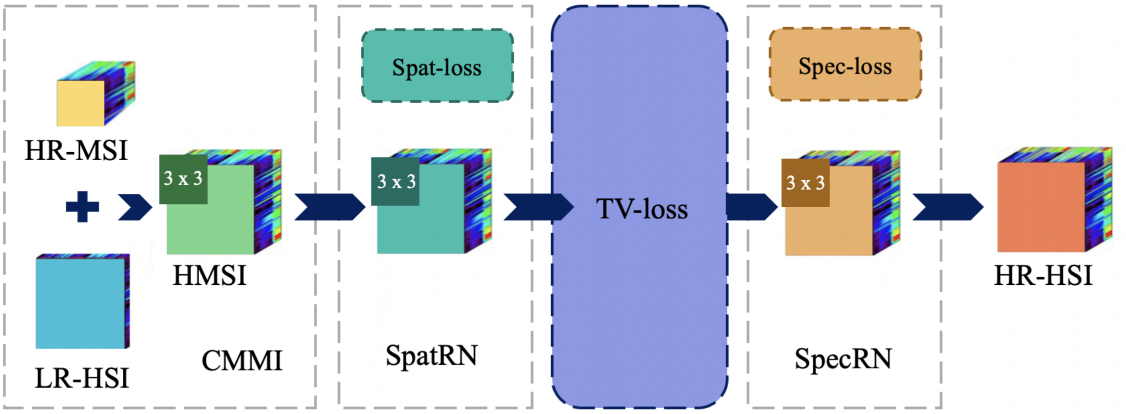


Figure 2: Framework used in the experiments to check the efficiency of TV-loss. TV-loss uses the output of the SpacRN module as input. The 3×3 squares indicate the convolutional layers that are applied to the input images. CMMI, SpatRN and SpecRN show the Cross-Mode Message Inserting, Spatial, and Spectral module respectively.

E. Smooth L1-loss function

As shown in the spatial module and spectral module of SSR-NET explained above, to compare the edge maps generated by the module with the reference edge maps, they use MSE. In the experiments in this paper, we replace MSE function with Smooth L1-loss, and see how it affects the network. Smooth L1-loss algorithm uses a criterion that when the absolute error of the elements is less than beta applies the square, otherwise apply L1 term [15]. This function is described as follows:

$$loss(x, y) = \frac{1}{n} \sum_i z_i \tag{14}$$

where z_i:

$$z_i = \begin{cases} 0.5(x_i - y_i)^2/beta, & if|x_i - y_i| < beta \\ |x_i - y_i| - 0.5 * beta, & otherwise \end{cases} \tag{15}$$

beta is a modifiable parameter that by default is equal to 1, which is the parameter to be used for the experiments in the experiments.

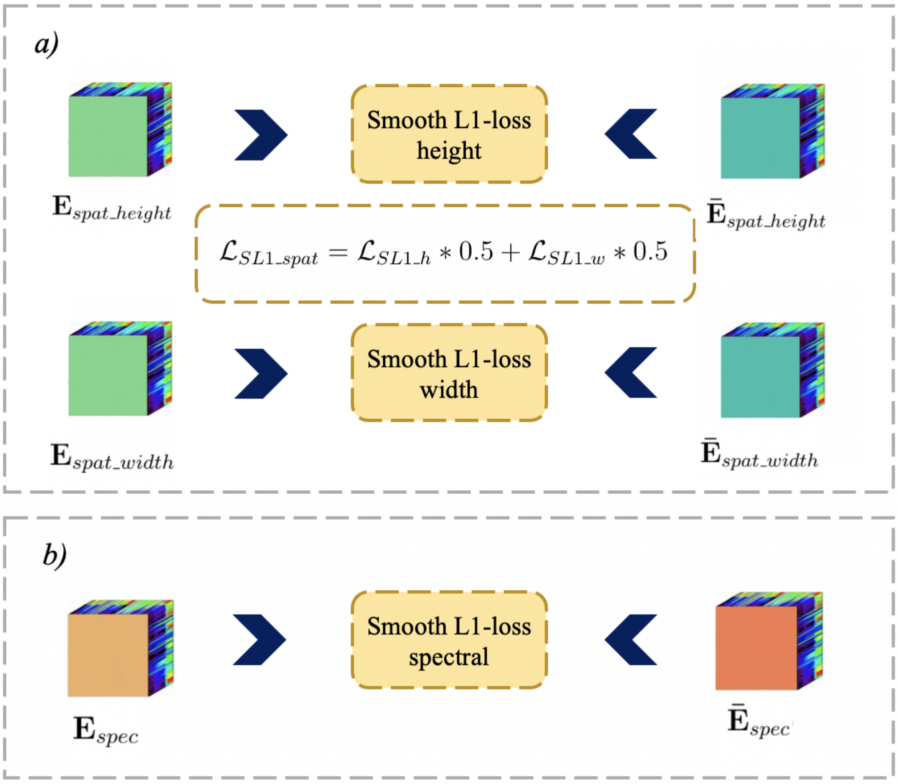

a)
Smooth L1-loss height
$\mathbf{E}_{spat\_height}$
$\bar{\mathbf{E}}_{spat\_height}$
$\mathcal{L}_{SL1\_spat} = \mathcal{L}_{SL1\_h} * 0.5 + \mathcal{L}_{SL1\_w} * 0.5$
Smooth L1-loss width
$\mathbf{E}_{spat\_width}$
$\bar{\mathbf{E}}_{spat\_width}$
b)
Smooth L1-loss spectral
$\mathbf{E}_{spec}$
$\bar{\mathbf{E}}_{spec}$

Figure 3: Smooth L1-loss implementation. a) Smooth L1-loss takes as input the features map estimated after the spatial module ($\mathbf{E}_{spat\_height}$ and $\mathbf{E}_{spat\_width}$), and the reference map of HR-HSI ($\bar{\mathbf{E}}_{spat\_height}$ and $\bar{\mathbf{E}}_{spat}$) to calculate the loss. To obtain the final loss, the results are used after applying the Smooth L1-loss function along the height and width. b) In the spectral module, the features maps of the estimated images ($\mathbf{E}_{spec}$) are utilized with respect to the reference ($\bar{\mathbf{E}}_{spec}$) ones along the spectral dimension.

3 Experiments

This section begins by outlining the five datasets used for experimentation and the evaluation metrics employed to assess the results. We then delve into the experimental setup, followed by a presentation of the implementation outcomes compared to the original SSR-NET.

3.1 Data Sets

The databases that are going to be exposed below are available as an open resource [16], in addition, the images have their corresponding ground truth.

1) Botswana: this database was captured in 2001-2004 by the Hyperion sensor of the NASA EO-1 satellite over the Okavango Delta. The data consists of 147 x 256 pixels with a spatial resolution of 30 m. Spectral bands cover wavelengths from 400 to 2500 nm. Removing the uncalibrated and noisy bands of water absorption features 145 bands left.

2) Urban: urban dataset was collected in 1995 over Copperas Cove, TX, USA. There are 307 x 307 pixels with 2 m as spatial resolution. It has 210 bands in total in a range from 400 to 2500 nm with an interval of 10 nm. Urban is made up of captures of buildings, architectural structures, or urban landscapes. When removed bands of dense water vapor and atmospheric, are obtained 162 bands.

3) Pavia Center: the Pavia Center database was collected with the same optical sensor used to capture the images from Pavia University, thus having the same spatial resolution, 1.3 m. However, in this set, each band has 1096 x 1096 pixels.

4) Pavia University: database obtained in 2003 by ROSIS-3 optical airborne sensor over the area of the University of Pavia, Italy. This dataset has 610 x 610 pixels and 1.3 m resolution. The bands are 115 with a spectral range of 430 to 860 nm within an interval of 10 nm.

5) Kennedy Space Center (KSC): This database was collected with the AVIRIS optical sensor from NASA, on March 23, 1996, over the Kennedy Space Center (KSC) in Florida. The images have 224 bands with a range of 400 to 2500 nm in 10 nm steps and a spatial resolution of 18 m. After removing the water absorption bands, 176 bands remain.

3.2 Evaluation Metrics

We evaluate the performance of our implementations using four quality metrics applied to each dataset. These metrics assess both the spatial and spectral fidelity of the reconstructed images compared to their corresponding ground truth data.

1) Peak Signal-to-Noise Ratio (PSNR): the peak SNR (PSNR) is a very popular quality metric, used to evaluate the spatial quality of the bands in the reconstructed HR-HSI. This metric measures the similarities between the fused image and the reference image.

$$PSNR = 10\log_{10}\left(\frac{\max(\mathbf{R}_k)^2}{\frac{1}{HW} \parallel \mathbf{R}_k - \mathbf{Z}_k \parallel {}_2^2}\right), \tag{16}$$

where R and Z are the *kth* band of the reference and estimated reconstructed image, respectively. The final result is the average of all the bands, where a higher result indicates a better spatial quality of the final image.

2) Erreur Relative Globale Adimensionelle de Synthèse (ERGAS): the ERGAS measures the quality of the fused image in a global statistical way, where the ideal value would be 0.

$$ERGAS = \frac{100}{r}\sqrt{\frac{1}{L}\sum_{k=1}^{L}\frac{\parallel \mathbf{R}_k - \mathbf{Z}_k \parallel {}_2^2}{\mu^2(\mathbf{R}_k)}}, \tag{17}$$

where r is the ratio of the spatial resolution from HR-MSI to LR-HSI. R_k and Z_k denotes the *kth* bands of the reference and fused image, accordingly. Moreover, $\mu(R_k)$ represents the mean value of the *kth* band in the reference image.

3) Spectral Angle Mapper (SAM): this metric evaluates the spectral information preservation at each pixel. It is useful to estimate the spectral distortion.

$$SAM = \arccos\left(\frac{\langle \mathbf{R}(i,j), \mathbf{Z}(i,j)\rangle}{\| \mathbf{R}(i,j) \|_2 \| \mathbf{Z}(i,j) \|_2}\right), \tag{18}$$

where $\mathbf{R}(i,j)$ and $\mathbf{Z}(i,j)$ represent the spectral vectors at the spatial pixel position (i,j) in the reference and estimated fused image respectively. In addition, $\langle \mathbf{R}(i,j), \mathbf{Z}(i,j)\rangle$ is the two vector inner product. The final result is achieved by averaging the SAM metric for all pixels. A better spatial quality is obtained for SAM values close to 0.

4) Root-Mean-Squared Error (RMSE): the RMSE measures the difference between the estimated and reference image, to compare the prediction errors.

$$RMSE = \sqrt{\frac{\sum_{k=1}^{L}\sum_{i=1}^{H}\sum_{j=1}^{W}(\mathbf{R}_k(i,j) - \mathbf{Z}_k(i,j))^2}{HWL}}, \tag{19}$$

The best results are obtained with smaller values for the RMSE.

3.3 Experimental settings

Our experimental setup mirrors the one presented in [9]. In this dataset, the central region of the images is chosen as a test set, cropped to fit 128x128 pixels. Another randomly cropped region of the same size is used as the training set.

To assess the impact of the proposed algorithms, we conducted experiments on three different network configurations: one incorporating TV-loss, another utilizing Smooth L1-loss, and a final network combining both functions. During the training stage, all models were optimized using the Adam optimizer for 10,000 iterations.

For the experiment setup, we used Python 2.7.17. The setup was a computer Alienware Aurora r8, Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz, with RAM of 32GB, and the GPU GeForce GTX 2080, 11GB.

3.4 Comparison with Data Sets

A) Botswana Data Set: our implementation combining TV-loss and Smooth L1-loss achieves excellent ERGAS scores on this dataset, demonstrating a significant improvement over the baseline SSR-Net. This emphasizes the effectiveness of our approach in capturing global information, as ERGAS specifically targets this aspect.

TV-loss and Smooth L1-loss effectively address the limitations of separate spatial and spectral edge loss functions, which tend to prioritize local features and lead to poorer performance in global evaluations as discussed in [9]. Notably, we achieve these improvements in ERGAS while maintaining similar PSNR values to SSR-Net and even surpassing it in the SAM metric. Figure 4 visually confirms these findings, showcasing the good results in ERGAS without compromising PSNR during training.

		Botswana		
Method	PSNR	RMSE	ERGAS	SAM
TV-loss	35.6858243	0.5984603	9.8286364	2.9243915
Smooth L1-loss	35.3552605	0.6216752	9.4089469	2.8217714
TV + Smooth L1	35.9895400	0.5778959	7.8465301	2.6916665
SSR-Net	36.0953531	0.5708985	9.0349448	2.7552987

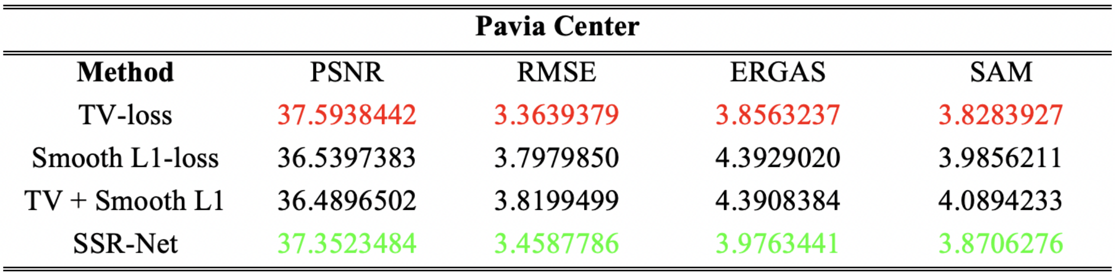
Table 1: Quantitative comparison of TV-loss, Smooth L1-loss and both added implementations, with respect to the original SSR-Net [9]. The best mark appears in RED, and the second best mark in GREEN.

B) Urban Data Set: On the Urban dataset, TV-loss takes the lead on most metrics, with Smooth L1-loss following closely behind. However, for the SAM metric, Smooth L1-loss gets worse results and is surpassed by SSR-NET. This suggests Smooth L1-loss might be badly suited for preserving spectral information, as evidenced by its bad performance on SAM in this case. Interestingly, combining TV-loss and Smooth L1-loss in the Urban dataset yields results that fall short of the baseline SSR-Net.

		Urban		
Method	PSNR	RMSE	ERGAS	SAM
TV-loss	36.5380089	2.9326284	1.4765417	2.7778814
Smooth L1-loss	36.3140211	3.0092373	1.5204431	2.8409947
TV + Smooth L1	33.2532634	4.2804980	2.0310636	2.8240672
SSR-Net	35.5324254	3.2925792	1.6043167	2.7424033

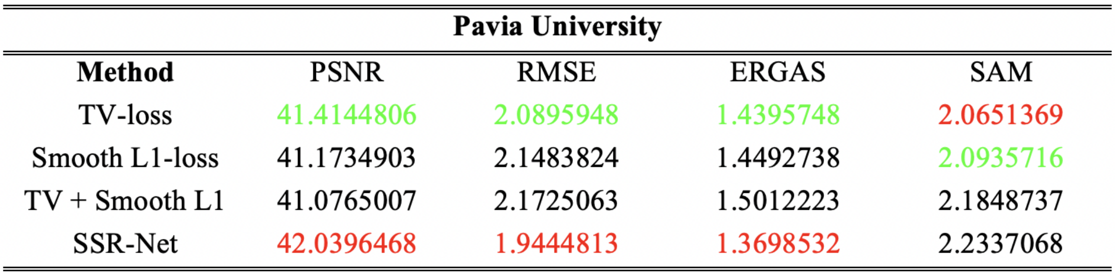
Table 2: Quantitative comparison of TV-loss, Smooth L1-loss and both added implementations, with respect to the original SSR-Net [9]. The best mark appears in RED, and the second best mark in GREEN.

C) Pavia Center Set and Pavia University Data Set: given the similarities between the Pavia Center and Pavia University datasets, we obtain similar results. TV-loss consistently achieves the best performance across all metrics for Pavia Center. For Pavia University, TV-loss excels in the SAM metric, indicating its strength in spectral information preservation. While competitive, Smooth L1-loss and the combined TV-loss and Smooth L1-loss implementations generally do not surpass the baseline SSR-Net results on these datasets.



Pavia Center				
Method	PSNR	RMSE	ERGAS	SAM
TV-loss	37.5938442	3.3639379	3.8563237	3.8283927
Smooth L1-loss	36.5397383	3.7979850	4.3929020	3.9856211
TV + Smooth L1	36.4896502	3.8199499	4.3908384	4.0894233
SSR-Net	37.3523484	3.4587786	3.9763441	3.8706276

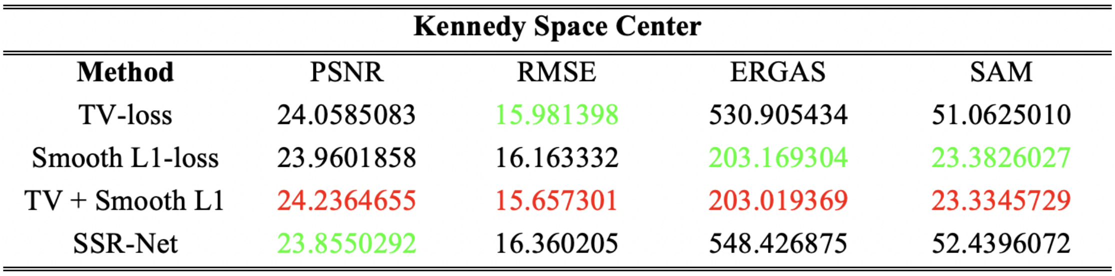
Table 3: Quantitative comparison of TV-loss, Smooth L1-loss and both added implementations, with respect to the original SSR-Net [9]. The best mark appears in RED, and the second best mark in GREEN.



Pavia University				
Method	PSNR	RMSE	ERGAS	SAM
TV-loss	41.4144806	2.0895948	1.4395748	2.0651369
Smooth L1-loss	41.1734903	2.1483824	1.4492738	2.0935716
TV + Smooth L1	41.0765007	2.1725063	1.5012223	2.1848737
SSR-Net	42.0396468	1.9444813	1.3698532	2.2337068

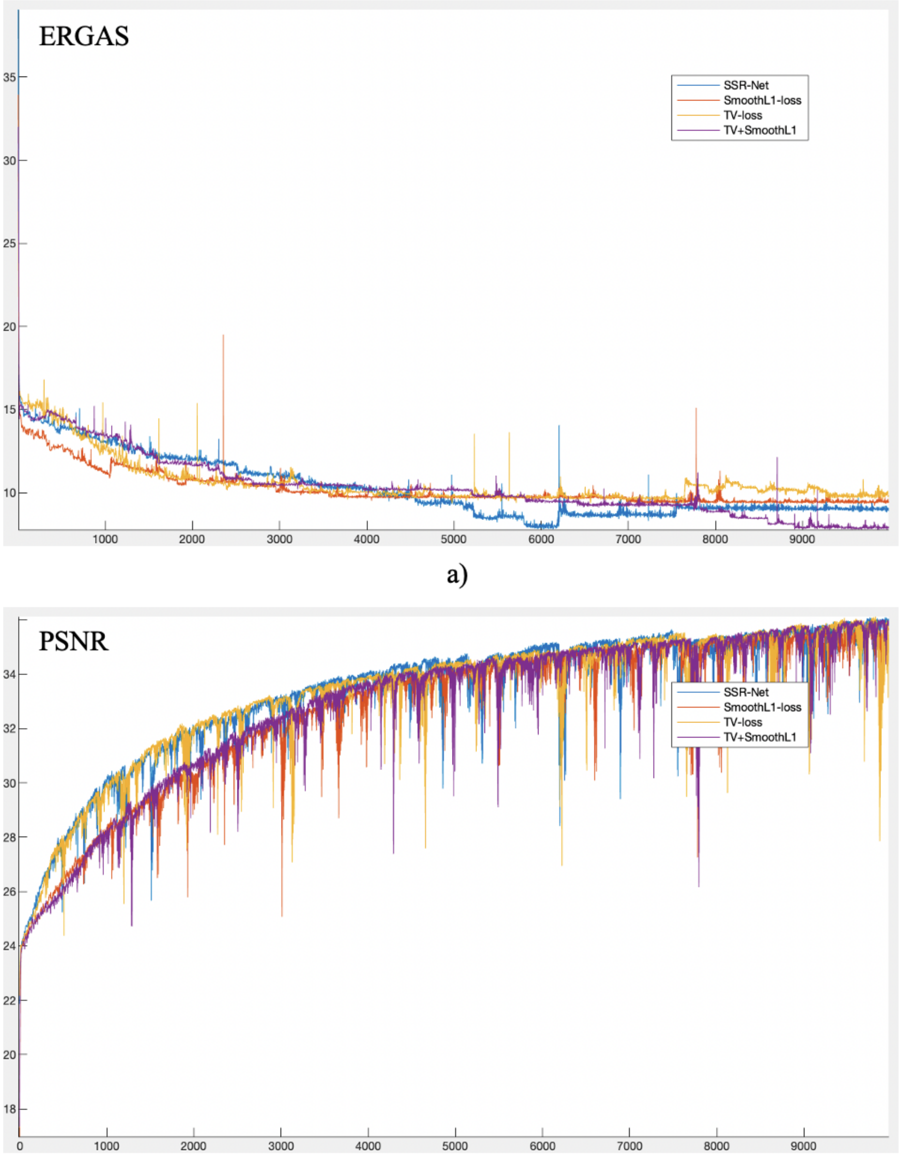
Table 4: Quantitative comparison of TV-loss, Smooth L1-loss and both added implementations, with respect to the original SSR-Net [9]. The best mark appears in RED, and the second best mark in GREEN.

D) Kennedy Space Center Data Set (KSC): the experiments in this dataset reaffirm the dominance of the combined TV-loss and Smooth L1-loss approach. Similar to the Botswana dataset, this combination achieves top marks across all metrics for KSC. This trend suggests the combined TV-loss and Smooth L1-loss might be particularly effective for high-resolution imagery datasets, which is a characteristic shared by both KSC and Botswana. Furthermore, Smooth L1-loss also achieves good results, having the second-best scores for ERGAS and SAM metrics. Besides, we can observe that SSR-Net achieves a good PSNR, likely due to its use of MSE-loss, which usually improves PSNR values.



Kennedy Space Center				
Method	PSNR	RMSE	ERGAS	SAM
TV-loss	24.0585083	15.981398	530.905434	51.0625010
Smooth L1-loss	23.9601858	16.163332	203.169304	23.3826027
TV + Smooth L1	24.2364655	15.657301	203.019369	23.3345729
SSR-Net	23.8550292	16.360205	548.426875	52.4396072

Table 5: Quantitative comparison of TV-loss, Smooth L1-loss and both added implementations, with respect to the original SSR-Net [9]. The best mark appears in RED, and the second best mark in GREEN.



a)

b)

Figure 4: Metrics (a) ERGAS and (b) PSNR during the training stage for each of the implementations and for the original SSR-Net [9]. The x-axis indicates the number of iterations.

4 Conclusion

In this paper, the implementations of TV-loss, Smooth L1-loss and both loss functions combined, are proposed to achieve improved results in multispectral and hyperspectral image fusion. After the experiments in the five databases selected, we observe the implementations represent an improvement in the results with respect to the original SSR-Net chosen as reference. The implementation of TV-loss plus Smooth L1-function manages to improve the fusion for the Botswana and Kennedy Space Center databases, indicating that TV-loss plus Smooth L1-loss improves extraction of complex spatial features, since the two databases have a high spatial resolution per pixel, 30 m and 18 m respectively. We especially get excellent results for the metric that assesses fusion quality globally (ERGAS). This implementation manages to provide a global restoration in the final HR-HSI without excessively damaging the PSNR, solving the problem that spatial edge loss and spectral edge loss only focus on the restoration of local features. On the other hand, for databases with less resolution, such as Pavia Center, Pavia University or Urban, we obtain the TV-loss is the best of the implementations, getting the best results for all the metrics in both Urban and Pavia Center data sets. The reason why implementation of the two loss functions combined worsens the results in these databases, it may be because both algorithms together overly softens the spatial features of the images with less spatial resolution, since the two algorithms focus on denoising.

Summing up, the implementations improve the results, being an interesting proposal to continue investigating in the different combinations between the TV-loss and Smooth L1-loss functions to improve the hyperspectral image restoration. As future works, it can be tested with TV-loss along the spectral dimension, since in our article TV-loss focuses on spatial features. After the good results obtained from the TV-loss implementation, another important future work would be the incorporation of a graph total variation (GTV) that supports our TV-loss function.

5 Acknowledgements

Above all, thanks to my supervisors Hao Wang and Guoxia Xu, for having guided and solved the problems that have arisen throughout this project, as well as providing the necessary knowledge to be able to successfully complete the results. And also thanks to the NTNU Computer Science department for supplying the necessary GPU and computer to carry out the experiments, especially to Binu Melit Devassy the person who has managed the access to the machine.

Bibliography

[1] R. Dian, S. Li, A. Guo and L. Fang, ‘Deep hyperspectral image sharpening’, *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–11, 2018.

[2] R. Dian, S. Li, B. Sun and A. Guo, ‘Recent advances and new guidelines on hyperspectral and multispectral image fusion’, *Information Fusion*, 2020.

[3] Z. Shao and J. Cai, ‘Remote sensing image fusion with deep convolutional neural network’, *IEEE journal of selected topics in applied earth observations and remote sensing*, vol. 11, no. 5, pp. 1656–1669, 2018.

[4] J. Yang, Y.-Q. Zhao and J. C.-W. Chan, ‘Hyperspectral and multispectral image fusion via deep two-branches convolutional neural network’, *Remote Sensing*, vol. 10, no. 5, p. 800, 2018.

[5] X.-H. Han, Y. Zheng and Y.-W. Chen, ‘Multi-level and multi-scale spatial and spectral fusion cnn for hyperspectral image super-resolution’, in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[6] G. Masi, D. Cozzolino, L. Verdoliva and G. Scarpa, ‘Pansharpening by convolutional neural networks’, *Remote Sensing*, vol. 8, no. 7, p. 594, 2016.

[7] F. Palsson, J. R. Sveinsson and M. O. Ulfarsson, ‘Multispectral and hyperspectral image fusion using a 3-d-convolutional neural network’, *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 639–643, 2017.

[8] R. Dian, S. Li, A. Guo and L. Fang, ‘Deep hyperspectral image sharpening’, *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–11, 2018.

[9] X. Zhang, W. Huang, Q. Wang and X. Li, ‘Ssr-net: Spatial-spectral reconstruction network for hyperspectral and multispectral image fusion’, *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[10] W. He, H. Zhang, L. Zhang and H. Shen, ‘Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration’, *IEEE transactions on geoscience and remote sensing*, vol. 54, no. 1, pp. 178–188, 2015.

[11] H. K. Aggarwal and A. Majumdar, ‘Hyperspectral image denoising using spatio-spectral total variation’, *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 3, pp. 442–446, 2016.

[12] A. Chadha, J. Britto and M. M. Roja, ‘Iseebetter: Spatio-temporal video super-resolution using recurrent generative back-projection networks’, *Computational Visual Media*, vol. 6, no. 3, pp. 307–317, 2020.

[13] W. Chlewicki, F. Hermansen and S. Hansen, ‘Noise reduction and convergence of bayesian algorithms with blobs based on the huber function and median root prior’, *Physics in Medicine & Biology*, vol. 49, no. 20, p. 4717, 2004.

[14] H. A. Aly and E. Dubois, ‘Image up-sampling using total-variation regularization with a new observation model’, *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1647–1659, 2005.

[15] SmoothL1Loss. ‘Smoothl1loss - pytorch 1.7.0 documentation’. (), [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.SmoothL1Loss.html#torch.nn.SmoothL1Loss (visited on 07/12/2020).

[16] H. R. S. Scenes. ‘Hyperspectral remote sensing scenes - grupo de inteligencia computacional (gic)’. (), [Online]. Available: http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes (visited on 07/12/2020).