Abstract

This work studies Hyperspectral image (HSI) super-resolution (SR). HSI SR is characterized by high-dimensional data and a limited amount of training examples. This raises challenges for training deep neural networks that are known to be data hungry. This work addresses this issue with two contributions. First, we observe that HSI SR and RGB image SR are correlated and develop a novel multi-tasking network to train them jointly so that the auxiliary task RGB image SR can provide additional supervision and regulate the network training. Second, we extend the network to a semi-supervised setting so that it can learn from datasets containing only low-resolution HSIs. With these contributions, our method is able to learn hyperspectral image super-resolution from heterogeneous datasets and lifts the requirement for having a large amount of high resolution (HR) HSI training samples. Extensive experiments on three standard datasets show that our method outperforms existing methods significantly and underpin the relevance of our contributions. Our code can be found at https://github.com/kli8996/HSISR.git.

1. Introduction

Hyperspectral imaging acquires images across many intervals of the electromagnetic spectrum. It has been applied to numerous areas such as medical diagnosis [37], food quality and safety control [23], remote sensing [22] and object detection [39]. All these applications benefit from analyzing the spectral information coming with HSIs. One obstacle in the way of further unleashing this potential is data acquisition. Acquiring HSIs of high spatial and high spectral resolution at a high frame rate is still a grand challenge. There is still no camera to achieve these three goals at the same time. Cameras for a compromise setting – high spectral but low spatial resolution – are quite common by now, though still expensive. As a result, increasing efforts have been made to advance HSI super-resolution (SR).

While numerous deep learning methods have been developed for improving the resolution of RGB images (RGBIs), methods for HSI SR are fewer. One of the main reasons is the lack of large-scale HSI datasets featuring high-resolution (HR) HSIs. As known, supervised deep learning methods need an enormous amount of training data. This situation, unfortunately, will not be improved in the foreseeable future due to the challenges hyperspectral imaging faces. In this work, we choose a different route and propose to learn HSI SR with an auxiliary task. We find that while it is difficult to collect HR HSIs, it is very easy to collect HR RGB images. It is thus appealing to have a HSI SR method which can learn from the two heterogeneous sources – RGB images and hyperspectral images – for hyperspectral ISR. Our method is designed for this aim.

Although the data distribution is not the same between RGBIs and HSIs, the two SR tasks do share some common goals in integrating information from neighboring spatial regions during the learning. We embrace this observation and formulate both tasks into the same learning framework such that the parameter distribution induced by the RGBI SR task can serve as an effective regularization for our HSI SR task. The challenge lies in the difference in spectral band numbers, e.g. three in RGBIs vs. e.g. 31 or 128 in HSIs. To tackle this problem, we decompose the HSI SR and RGBI SR into a commonly-shared spatial super-resolution task and two specific spectral refinement tasks, and propose a novel spatial-spectral neural network to solve them in a multi-tasking framework. This way, the spatial super-resolution network is shared between the two tasks to increase the total amount of supervision. It is in a similar spirit to other multi-tasking learning methods [46].

While the aforementioned contribution can yield state-of-the-art performance for HSI SR already, we extend the method further to learn from ‘unlabeled’ low-resolution HSI images as well. Semi-supervised learning (SSL) exploits unlabeled data to reduce over-fitting to the limited amount of labeled data [16, 32, 45, 49, 25]. While good progress has been made, the strategies are mainly designed for high-level recognition tasks. Their applicability to a
low-level dense regression task such as HSI SR has yet to be verified. In this work, we again leverage the success of RGB image (RGBI) SR and propose a cross-model consistency that favors functions giving consistent outputs between super-resolved RGBIs and super-resolved HSIs. Basically, we convert LR HSIs into LR RGB images and pass those through the trained RGB SR network. In the meanwhile, we pass the LR HSIs through our HSI SR network to get the super-resolved HSIs and convert them to RGBIs with a standard camera response function. We enforce the consistency between the two versions of super-resolved RGBIs. This way, supervision is transferred from the better-trained RGB SR network to our HSI SR network via a second route.

To summarize, this work makes two contributions: 1) a multi-tasking HSI SR method to learn together with an auxiliary RGBI SR task, and 2) A SSL method to learn also from `unlabeled` LR HSIs. With these contributions, our method sets the new state of the art for hyperspectral image super-resolution.

2. Related Work

**Hyperspectral Image Super-Resolution.** HSI SR can be grouped into three categories according to their settings: 1) HSI SR from only RGBIs or HR multispectral images (MSIs); 2) Single HSI SR from LR HSIs; and 3) HSI SR from both HR RGBIs (or MSIs) and LR HSIs of the same scene. Our method belongs to the second group.

HSI SR from only RGBIs is a highly ill-posed problem. However, it has gained great traction in recent years due to its simple setup and the well-organized workshop challenges [8]. Similar to other computer vision topics, the trend has shifted from ‘conventional’ methods such as radial basis functions [40] and sparse coding [2] to deep neural networks [21, 44, 8]. This trend highlights the need for bigger training datasets. Due to the challenge of reconstructing HSI from RGBIs that contain only three bands, there emerges research applying MSIs containing 3-8 bands to reconstruct HSIs [14].

Single image SR aims to model the relationship between the LR images and HR ones by learning from a collection of examples consisting of pairs of HR images and LR images. Single RGBI SR has achieved remarkable results in the last years. Since the first work of using neural networks for the task [18], progress has been made in making networks deeper and the connections denser [28, 57], using feature pyramids [31], employing GAN losses [34], and modeling real-world degradation effects [24]. As to single HSI SR, there has been great early work [3, 53] as well. However, that is also surpassed by deep learning methods. For instance, Yuan et al. [54] trained a single-band SR method on natural image datasets, and applied it to HSIs in a band-wise manner to explore spatial information. The spectral information is explored via matrix factorization afterwards. In order to explore both spatial and spectral correlation at the same time, methods based on 3D Convolutional Networks [38, 35] have been developed. Although 3D CNNs sound like a perfect solution, the computational complexity is very high. To alleviate this, Grouped Convolutions (GCs) with shared parameters have been recently used in [36, 27]. The backbone network of our method is also based on GCs.

Fusion-based methods use HR RGBIs (or MSIs) of the same scene as references to improve the spatial resolution of the LR HSIs [11, 53, 51]. This stream of methods has received more research attention than the former two. Many learning techniques have been applied to this data fusion task including Bayesian inference [5, 6, 50], matrix factorization [33, 17], sparse representation [4, 19], and deep neural networks [41, 50]. The common goal of these methods is to learn to propagate the detailed information in the HR RGBIs (or MSIs) to the target HSIs and fuse them with the fundamental spectral information from LR HSIs. Despite the plethora of fusion algorithms developed, they all assume that the LR HSIs and the HR RGBIs (or MSIs) are very well co-registered [27]. This data registration is a challenge on its own and registration errors will lead to degraded SR results [13, 59].

**Learning with Auxiliary Tasks.** It is quite a common practice to borrow additional supervision from related auxiliary tasks, when there is insufficient data to learn a task. The common strategy is to learn all the tasks together so that the auxiliary tasks can regularize the optimization. There are normally two assumptions: (1) we only care about the performance of the main task and (2) the supervision for the auxiliary tasks is easier to obtain than that of the main task. Previous work has employed various kinds of self-supervised methods as auxiliary tasks for the main supervised task in a semi-supervised setting [30, 10, 42]. For instance, generative approaches have been explored in [30] and predicting the orientation of image patches is used in [10]. Another related setting is multi-task learning (MTL) [47]. In MTL, the goal is to reach high performance on multiple tasks simultaneously, so all tasks are main tasks and all tasks are auxiliary tasks. While the goal is different, many strategies in MTL such as parameter sharing [9], task consistency [55], and loss balance [15] are useful for learning with auxiliary tasks.

3. Approach

HSIs provide tens of narrow bands and RGB images have three bands. In order to let them share a large part of the overall network, we decouple both the HSI SR task and the RGBI SR task into a spatial super-resolution task and a spectral refinement task. The spatial super-resolution task is designed to enhance the spatial resolution of a general single-channel image, regardless of the spectral frequency
of that band. This way, it can be used and shared by both of the SR tasks. The spectral refinement networks are task-specific – one to refine the spectral signals of the three channels of RGB images and the other to refine the spectral signals of tens of bands for HSIs.

In this work, we assume that the relationships between low/high-resolution HSIs and low/high-resolution RGBIs are correlated, so they should be trained together so that RGBI SR can provide additional supervision for HSI SR. This way, the HSI SR method can enjoy training samples of a much more diverse set of scenes especially those that cannot be captured well by current hyperspectral imaging devices such as moving objects. In order to share the spectral super-resolution network by all single bands from the two tasks, we use a grouped convolutional network with group size of 1. The grouping strategy is to divide input HSIs and RGB images into single bands. The architecture of the network is shown in Fig. 1.

3.1. HSI SR with an Auxiliary RGBI SR Task

Given two SR tasks $T_{HS}$ and $T_{RGB}$, we aim to help improve the learning of a model for $T_{HS}$ by using the knowledge contained in $T_{RGB}$. In the supervised setting, each task is accompanied by a training dataset consisting of $N$ training samples, i.e., $\mathcal{D}_{HS} = \{x_{HS}, y_{HS}\}^N_{i=1}$ and $\mathcal{D}_{RGB} = \{x_{RGB}, y_{RGB}\}^N_{i=1}$, where $x_{HS} \in \mathbb{R}^{h \times w \times C}$, $y_{HS} \in \mathbb{R}^{h \times w \times 1}$, $x_{RGB} \in \mathbb{R}^{h_2 \times w_2 \times Z}$, and $y_{RGB} \in \mathbb{R}^{h_2 \times w_2 \times 3}$. We denote low-resolution (LR) images by $x$, high-resolution (HR) images by $X$, the number of bands of HSIs by $C$, the number of bands in RGB images by $Z$ (3 here), and the size of the images by $h, w, H$ and $W$. Given a scaling factor $\tau$, we have $H_i = \tau h_i$ and $W_i = \tau w_i$ for both tasks.

The goal is to train a neural network $\Phi_{HS}$ to predict the HR HSI for a given LR HSI: $X_{HS} = \Phi_{HS}(x_{HS})$. Different from previous methods, which have a single network for the whole task, our method consists of three blocks: an encoder which is shared by the two SR tasks, and two task-specific decoders to output the final outputs. More specifically, $\Phi_{HS} = (\Phi^{En}_{HS}, \Phi^{De}_{HS})$ and $\Phi_{RGB} = (\Phi^{En}_{RGB}, \Phi^{De}_{RGB})$.

In order to share the same encoder between the two SR tasks to enhance the spatial resolution of all single-band images, we divide $x_{HS}$ into $C$ single bands and $x_{RGB}$ into 3 single bands. For both tasks, the encoder network $\Phi^{En}$ takes one low-resolution single-band image as input and generates one high-resolution single-band image as output, regardless of the spectral frequency of that band. The outputs of all the bands of $x_{HS}$ are then concatenated according to their original spectral band position to assemble a high-resolution HSI $X_{HS} \in \mathbb{R}^{H \times W \times C}$. Similarly, we can assemble a high-resolution RGB image $X_{RGB} \in \mathbb{R}^{H_2 \times W_2 \times 3}$. There are two upsampling layers to upscale the size of the input to the desired size in a progressive manner. This progressive upsampling has proven useful for both RGBI SR [31] and HSI SR [27]. The reconstructed $X_{HS}$ is then fed into the decoder network $\Phi^{De}_{HS}$ as a whole for spectral refinement in order to generate the final output $\hat{X}_{HS}$, which is then compared to the ground truth $X_{HS}$ to compute the

Figure 1: The architecture of our network consisting of a shared encoder and two specific decoders for the two SR tasks.
loss for HSI SR. Likewise, the spatially enhanced $\tilde{X}_{\text{RGB}}$ is fed into the decoder network $\Phi_{\text{De}}$, to generate the final HR RGB estimate $\hat{X}_{\text{RGB}}$, which is then compared to the ground truth $X_{\text{RGB}}$ to compute the loss for RGBI SR. For the refinement decoders, all the bands are fed directly to learn both short-range and long-range spectral correlations to refine the results.

In order to have a modular design, the three sub-networks have the same basic architecture. The encoder network for spatial super-resolution is composed of a sequence of Spatial Block (SB) modules. The SB module has two identical basic cells connected in a sequence, each consists of one $3 \times 3$ convolutional layer, followed by a Relu and another $3 \times 3$ convolutional layer. There is also skip connection for each of this basic cell. Please see the bottom-left panel of Fig. [1] for its structure. The two decoders designed for spectral refinement of the two SR tasks have the same architecture and each consist of a sequence of three spectral-spatial block (SSB) modules. The SSB module was proposed in [27] as a basic building block for their spectral-spatial block (SSB) modules. The SSB module is consists of one convolutional layer, followed by a Relu and another $3 \times 3$ convolutional layer. There is also skip connection for each of this basic cell. Please see the bottom-left panel of Fig. [1] for its structure. The two decoders designed for spectral refinement of the two SR tasks have the same architecture and each consist of a sequence of three spectral-spatial block (SSB) modules. The SSB module was proposed in [27] as a basic building block for their

3.2. Semi-Supervised HSI SR

While training with auxiliary RGB SR task can greatly improve the performance, it is still interesting to investigate whether the method can also learn from an additional collection of low-resolution hyperspectral images. This setting is interesting because capturing low-resolution hyperspectral images is much easier than capturing high-resolution hyperspectral images. This is especially true as modern snapshot HS cameras that captures LR HSIs at high frame rate are becoming more and more accessible. This means that methods that can learn further with low-resolution hyperspectral images are practically very useful. In the literature, there has been a diverse sets of methods developed for semi-supervised learning (SSL) based on techniques such as entropy minimization and pseudo-labels generation. However, they are mostly designed for high-level recognition tasks and cannot be applied to HSI SR directly.

In this work, we propose a new SSL method specifically for HSI SR. For this purpose, we again leverage the fact that RGBI SR is a better-addressed problem, given that it has a large amount of training data and it predicts only three channels. The method works as follows: given an image $x_{\text{HS}}$, we convert it to an RGB image $\tilde{x}_{\text{RGB}}$ with the camera response function of a standard RGB camera:

$$f : \tilde{x}_{\text{RGB}}^{(i,j)} = f \ast \hat{x}_{\text{HS}}^{(i,j)}, \quad (1)$$

where $\ast$ is a convolution operation. The operation is to integrate the spectra signatures into R, G, and B channels with the response function of a standard camera. Note that this conversion is widely used in the literature of spectral image super-resolution [20]. The response function of Canon 1D Mark 3 [26] is used in this work but the method works with the response functions of other cameras.

The original HSI $x_{\text{HS}}$ and the converted RGB image $\tilde{x}_{\text{RGB}}$ are then fed into the HSI SR network $\Phi_{\text{HS}}$ and the RGBI SR network $\Phi_{\text{RGB}}$, respectively, to generate the super-resolved results:

$$\hat{X}_{\text{HS}} = \Phi_{\text{HS}}(x_{\text{HS}}), \quad (2)$$

and

$$\hat{X}_{\text{RGB}} = \Phi_{\text{RGB}}(\tilde{x}_{\text{RGB}}). \quad (3)$$

$\hat{X}_{\text{HS}}$ is then converted to an RGB image by using the same camera response function:

$$\tilde{\hat{x}}_{\text{RGB}}^{(i,j)} = f \ast \hat{X}_{\text{HS}}^{(i,j)}. \quad (4)$$

Finally, a consistency loss $L_{\text{ssl}}(\tilde{\hat{x}}_{\text{RGB}}, \tilde{x}_{\text{RGB}})$ is computed between the two HR RGB results. This consistency makes a good use of low-resolution HSIs and high-resolution RGB images. It transfers supervision from the RGB side to the HSI side. The diagram of this method is shown in Fig. [2].

3.3. Loss Function

The overall loss for our SR tasks is:

$$L_{\text{Total}} = L_{\text{HS}}(X_{\text{HS}}, \hat{X}_{\text{HS}}) + L_{\text{RGB}}(X_{\text{RGB}}, \hat{X}_{\text{RGB}}) + L_{\text{ssl}}(\tilde{\hat{x}}_{\text{RGB}}, \tilde{x}_{\text{RGB}}). \quad (5)$$

The main loss is augmented by the two auxiliary losses which are optional but highly beneficial.
In order to capture both spatial and spectral correlation of the SR results, we follow [27] and combine the L1 loss and the spatial-spectral total variation (SSTV) loss [1]. SSTV is used to encourage smooth results in both spatial domain and spectral domain and it is defined as:

\[
\mathcal{L}_{\text{SSTV}} = \frac{1}{N} \sum_{n=1}^{N} (\|\nabla_h \mathbf{X}^n\|_1 + \|\nabla_w \mathbf{X}^n\|_1 + \|\nabla_c \mathbf{X}^n\|_1),
\]

where \(\nabla_h\), \(\nabla_w\), and \(\nabla_c\) compute gradient along the horizontal, vertical and spectral directions, respectively. The loss is:

\[
\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_{\text{SSTV}}.
\]

A joint training with all losses together works well in principle by stacking multiple types of data samples in a single mini-batch. However, that will heavily limits the size of the training data for each loss. In this work, we adopt an alternating training strategy; that is to train with each of the three losses in turn in every iteration. In our implementation, the weights for all losses are set to 1. The contributions of different terms are balanced or controlled by altering the number of mini-batches for that loss in each iteration. The influence of these numbers are studied in Sec. 4.2.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate our method on three public datasets: CA VE dataset [52], Harvard dataset [12], and NTIRE 2020 dataset [8]. Images in CA VE and NTIRE 2020 dataset have 31 bands ranging from 400 nm to 700 nm at a step of 10 nm. Images in Harvard dataset contain 31 bands as well but range from 420 nm to 720 nm. The CA VE dataset contains 32 images of 512 x 512 pixels. We use 20 images for training and 10 images for testing. We evaluated in a supervised setting and a semi-supervised setting. For our semi-supervised setting, only 5 high-resolution training images are used and for the remaining 15 images, only their downsampled low-resolution images are available. For the Harvard dataset, there are 50 images in total. We use 40 for training and 10 for test. For the semi-supervised setting, 6 high-resolution images and the down-sampled low-resolution images of the remaining 34 images are used for training. For NTIRE 2020, there are 480 images. We use 400 images for training and 80 images for test. For the semi-supervised case, we further split the 400 images into 100 and 300; for the former, their high-resolution images can be used and for the latter only the downsampled low-resolution images are available. For the auxiliary RGBI SR task, we adopt the DIV2K Dataset [2]. Because the resolution of DIV2K is much higher than our HSIs, we first downsample them by a factor of \(\times 2\) and take these downsampled images as our HR RGB images. After cropping, it leads to 137,430 image patches of 64 \(\times\) 64 pixels. This is about 34,10, and 6 times larger than CAVE, Harvard and NTIRE datasets, respectively.

Methods. We compare the proposed method to four state-of-the-art HSI SR methods: GDRRN [36], 3DFCNN [38], SSPSR [27], and MCNet [35]. We use the same training data for all methods and use the default training settings given by the authors of these methods. Bicubic interpolation is also introduced as a baseline.

Evaluation Metrics. We follow the literature and evaluate the performance of all methods under three standard metrics [27]. They are root mean squared error (RMSE), erreur relative globale adimensionnelle de synthèse (ERGAS) [48], and peak signal-to-noise ratio (PSNR). For PSNR of the reconstructed HSIs, their mean values of all spectral bands are reported as MPSNR. ERGAS are widely used in HSI fusion task.

Parameters. In this work, we focus on scaling factor \(\times 4\) and \(\times 8\). For the case of \(\times 4\), we crop the images into patches of 64 \(\times\) 64 pixels without overlapping to collect the training data. For \(\times 8\), we use patches of 128 \(\times\) 128 pixels. Those patches are then downsampled via Bicubic interpolation to obtain the corresponding LR HSI patches. The choice of value for other parameters are studied in Sec. 4.2.

Training Details. We use ADAM optimizer [29] and train all variants of our method for 20 epoches. This is a small number compared to the ones used by the comparison methods. For instance, GDRRN [36] trains for 30 epoches, 3DFCNN [38] trains for 200 epoches, SSPSR [27] for 40 epoches, and MCNet [35] for 200 epoches. We find that 20 epoches are sufficient to give good results for our method, and believe a larger number probably can further push the numbers up. The initial learning rate of all our methods is set to \(10^{-4}\) and is reduced by a factor of 0.3 after every 3 epoches. As to the batch size, 16 is used for all experiments except for the case when the SSL loss is added. For that 8 is used due to the limit of GPU memory.

4.2. Ablation Study

We analyze the parameter choices of our method in this section. Experiments are conducted on the CA VE dataset in the semi-supervised setting.

Amount of RGB data. The number of mini-batches for
The good performance of our base model is mainly due to the use of decoupled spatial-spectral networks for this task. Due to the design, each sub-network can just focus on one task: spatial enhancement or spectral refinement. The network of \cite{38} is quite shallow, probably because 3D convolution based methods are computationally heavy in general. We believe that this is the reason why their method does not give top results. When compared to the very recent method MCNet \cite{35}, our base model also performs better in almost all cases.

The proposed contributions, namely training with the auxiliary task RGBI SR and the semi-supervised learning method based on cross-model consistency, both contribute significantly to the final results. Learning with the auxiliary task RGBI SR can work on its own. The SSL component needs to be used together with the auxiliary task RGBI SR. The results show that our SSL method can provide further improvement on top of the auxiliary RGBI SR method, and when the two components are combined together, we get the best performance.

When more supervision is given such as in the fully-supervised setting, the conclusions we drawn in the semi-supervised setting hold as shown in Table \ref{tab:all}. The results show that the proposed components are very effective and can be applied to situations with varying amount of HR HSIs. We show visual results of our method and other competing methods in Fig. \ref{fig:visualization1} and Fig. \ref{fig:visualization2}. The figure shows that our method generate few errors than all competing methods. When more supervision is given such as in the fully-supervised setting, the conclusions we drawn in the semi-supervised setting hold as shown in Table \ref{tab:all}. The results show that the proposed components are very effective and can be applied to situations with varying amount of HR HSIs. We show visual results of our method and other competing methods in Fig. \ref{fig:visualization1} and Fig. \ref{fig:visualization2}. The figure shows that our method generate few errors than all competing methods.

### 4.3. Main Results

We first present the results in the semi-supervised setting. The results of all competing methods and all variants of our method on the CAVE, Harvard, and NTIRE dataset are shown in Table \ref{tab:all} and in Table \ref{tab:all_supervised}. The results in this table and other results in supplemental material show that our method outperforms all other state-of-the-art methods significantly and consistently over all datasets and under all evaluation metrics.

The good performance of our base model is mainly due to the use of decoupled spatial-spectral networks for this task. Due to the design, each sub-network can just focus on one task: spatial enhancement or spectral refinement. The network of \cite{38} is quite shallow, probably because 3D convolution based methods are computationally heavy in general. We believe that this is the reason why their method does not give top results. When compared to the very recent method MCNet \cite{35}, our base model also performs better in almost all cases.

The proposed contributions, namely training with the auxiliary task RGBI SR and the semi-supervised learning method based on cross-model consistency, both contribute significantly to the final results. Learning with the auxiliary task RGBI SR can work on its own. The SSL component needs to be used together with the auxiliary task RGBI SR. The results show that our SSL method can provide further improvement on top of the auxiliary RGBI SR method, and when the two components are combined together, we get the best performance.

When more supervision is given such as in the fully-supervised setting, the conclusions we drawn in the semi-supervised setting hold as shown in Table \ref{tab:all}. The results show that the proposed components are very effective and can be applied to situations with varying amount of HR HSIs. We show visual results of our method and other competing methods in Fig. \ref{fig:visualization1} and Fig. \ref{fig:visualization2}. The figure shows that our method generate few errors than all competing methods.

### 4.4. Discussion

The superior performance shows that our method is able to learn from heterogeneous datasets rather than only from pairs of low-resolution and high-resolution HR HSIs. This greatly increases the amount of training data that can be used for HSI SR and can also include training samples for scenes such as moving objects that cannot be captured easily with the current hyperspectral imaging devices. We would like to point out that while the general concepts of learning with auxiliary tasks and semi-supervised learning are well known, the challenge and novelty lie in defin-

<table>
<thead>
<tr>
<th>#(Mini-Batches)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>8</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE ↓</td>
<td>0.01451</td>
<td>0.01357</td>
<td>0.01329</td>
<td>0.01309</td>
<td>0.01305</td>
<td>0.01315</td>
<td>0.01315</td>
<td>0.01317</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Performance as a function of the number of mini-batches for RGBI SR loss.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Components</th>
<th>CAVE</th>
<th>Harvard</th>
<th>NTIRE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RGBSR SSL</td>
<td>RMSE ↓</td>
<td>MPSNR ↑</td>
<td>ERGAS ↓</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td>0.0144</td>
<td>40.8385</td>
<td>4.0345</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>0.0118</td>
<td>42.3575</td>
<td>3.0128</td>
</tr>
<tr>
<td>Ours</td>
<td>✓ ✓</td>
<td>0.0114</td>
<td>42.7645</td>
<td>3.3346</td>
</tr>
<tr>
<td>Bicubic</td>
<td></td>
<td>0.0185</td>
<td>38.7380</td>
<td>5.2719</td>
</tr>
<tr>
<td>GDRRN</td>
<td>[36]</td>
<td>0.0246</td>
<td>36.2775</td>
<td>7.0043</td>
</tr>
<tr>
<td>3DFCNN</td>
<td>[38]</td>
<td>0.0173</td>
<td>38.3928</td>
<td>6.7055</td>
</tr>
<tr>
<td>SSPSR</td>
<td>[27]</td>
<td>0.0144</td>
<td>40.9131</td>
<td>4.0406</td>
</tr>
<tr>
<td>MCNet</td>
<td>[35]</td>
<td>0.0146</td>
<td>40.7385</td>
<td>4.1659</td>
</tr>
</tbody>
</table>

Table 2: Results of all methods on the CAVE, Harvard, and NTIRE datasets in the semi-supervised setting for the ×4 case.
### Table 3: Results of all methods on the CAVE, Harvard, and NTIRE datasets in the semi-supervised setting for the ×8 case.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Components</th>
<th>CAVE</th>
<th>Harvard</th>
<th>NTIRE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RGBSR SSL</td>
<td>RMSE</td>
<td>MPSNR</td>
<td>ERGAS</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td>0.0241</td>
<td>35.8976</td>
<td>7.1154</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>0.0215</td>
<td>37.1387</td>
<td>6.1442</td>
</tr>
<tr>
<td>Ours</td>
<td>✓ ✓</td>
<td>0.0206</td>
<td>37.3532</td>
<td>6.0027</td>
</tr>
</tbody>
</table>

### Table 4: Results of all methods on the CAVE and Harvard datasets in the fully-supervised setting for the ×4 case.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Components</th>
<th>CAVE</th>
<th>Harvard</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RGBSR SSL</td>
<td>RMSE</td>
<td>MPSNR</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td>0.0118</td>
<td>42.3836</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>0.0105</td>
<td>43.3242</td>
</tr>
<tr>
<td>GDRRN</td>
<td>[36]</td>
<td>0.0162</td>
<td>39.7470</td>
</tr>
<tr>
<td>3DFCNN</td>
<td>[38]</td>
<td>0.0158</td>
<td>39.2178</td>
</tr>
<tr>
<td>SSPSR</td>
<td>[27]</td>
<td>0.0124</td>
<td>42.1378</td>
</tr>
<tr>
<td>MCNet</td>
<td>[35]</td>
<td>0.0124</td>
<td>42.2597</td>
</tr>
</tbody>
</table>

Figure 3: Exemplar results of our method and two competing methods trained in the semi-supervised setting on the CAVE dataset: top row for the super-resolved results and bottom row for the error maps.

Table 4: Results of all methods on the CAVE and Harvard datasets in the fully-supervised setting for the ×4 case.
5. Conclusion

In this paper, we have proposed a new method for hyperspectral image (HSI) super-resolution (SR). We build a deep convolutional network that decouple the task HSI SR into two sub-tasks: spatial super-resolution of single spectral band and joint spectral refinement over all bands. The method yields the state-of-the-art results. To further improve it, we have proposed two more contributions. First, we extend the network such that the HSI SR task can be trained together with an auxiliary RGB image SR task to gain more supervision. Second, the network is extended to also learn from datasets with LR HSIs only. The contributions greatly increase the amount of training data that HSI SR methods can use. Extensive experiments show that all the contributions are useful for the performance.
References


